Universidad Nacional de San Luis

Diseño de Algoritmos de Aprendizajes de Máquinas y Visión Computacional para la Medición Autónoma de Variables Vitícolas

POR

Ing. Diego Sebastián Pérez

DIRECTOR DE TESIS: DR. FACUNDO BROMBERG Co-Director de Tesis: Dr. Marcelo Luis Errecalde

Índice general

1.	Intr	roducción	4
	1.1.	Antecedentes	6
		1.1.1. Agricultura de Precisión	6
		1.1.2. Viticultura de Precisión	8
		1.1.3. Inteligencia Artificial en el Agro	9
		1.1.4. Búsqueda visual activa de objetos	10
	1.2.	Objetivos de Investigación	11
	1.3.		16
	1.4.		17
2.	Con	nparación de Algoritmos de Interpolación Espacial con Evalua-	
	ción	del Impacto en la Cantidad de Datos Disponibles	8
	2.1.	Introducción	19
	2.2.	Elementos de Trabajo y Metodología	20
			22
		2.2.2. Kriging	23
		2.2.3. Máquinas de vector soporte para regresión	25
		2.2.4. Máquinas de vector soporte local	27
	2.3.	Resultados experimentales	29
		2.3.1. Evaluación de los métodos según las medidas de calidad 3	30
		2.3.2. Evaluación de la capacidad de MVS para modelar correlaciones	
		espaciales en variables vitícolas	34
	2.4.	Discusión	40
	2.5.	Conclusiones	43
3.	Seg	mentación de Imágenes en Viñedos para la Medición Autónoma	
	de V	Variables Vitícolas 4	15
	3.1.	Introducción	46
		3.1.1. Trabajos relacionados	47
	3.2.	Elementos de Trabajo y Metodología	49
		3.2.1. Segmentación de Imágenes	49
			50
			51
			52
			54
			56

		3.2.7. Máquinas de vector soporte para clasificación	57
	3.3.	Resultados experimentales	59
		3.3.1. Reclasificación por área	62
		3.3.2. Complejidad computacional de MG y MVS	63
	3.4.	Discusión	64
	3.5.	Aplicación: Medición de Diámetro de Tronco	67
		3.5.1. Medición del Diámetro en Píxeles	69
		3.5.2. Medición por Imágenes con Calibración Automática	71
		3.5.3. Discusión sobre las aplicaciones	78
	3.6.	Conclusiones y Trabajos Futuros	78
4.	Cla	sificación de Imágenes para la Detección de Yemas de Vid median-	
		Algoritmos de Visión Computacional y Aprendizaje de Máquinas	
	4.1.	Introducción	
		4.1.1. Trabajos relacionados	
	4.2.		86
		4.2.1. Corpus de parches etiquetados	87
		4.2.2. De píxeles a descriptores visuales	90
		4.2.3. Algoritmo de clasificación	95
	4.3.	1	97
		4.3.1. Evaluación de clasificador sobre los parches no-yema	99
		4.3.2. Evaluación del clasificador para parches típicamente extraídos	100
		de un proceso de escaneo-ventana	
	4.4	4.3.3. Análisis para los histogramas producidos por BoF	
	4.4.	Discusión	
	4.5.	Conclusiones	113
5.		cia un Algoritmo de Búsqueda Visual Activa para Yemas de Vid	
	5.1.		
	5.2.	Elementos de trabajo y metodología	
		5.2.1. Construcción del escenario de evaluación	
		5.2.2. Creación de un nuevo corpus de parches	128
		5.2.3. Planificación de vistas y construcción del conocimiento de sentido común	190
	5.3.	Resultados experimentales	
	5.4.	Discusión	
	5.5.		
C			
υ.	Cor	nclusiones Generales y Trabajos Futuros	152

Capítulo 1

Introducción

Con el objetivo de asistir en la toma de decisiones agrarias y en particular en la industria vitícola, la investigación descrita en este documento se orienta al estudio de técnicas de aprendizaje de máquinas y visión computacional aplicadas a la medición autónoma de procesos vitícolas a partir de imágenes del cultivo y su contexto tomadas en condiciones naturales de campo.

En la práctica, la mayoría de las decisiones que toma el agricultor dependen directamente de información obtenida a través de observaciones y mediciones del cultivo y su entorno. En algunos casos la información disponible es suficiente para que la certeza en las decisiones sea muy alta. Sin embargo, adquirir información del cultivo es un proceso que se encuentra generalmente limitado por la disponibilidad de recursos económicos y de tiempo. Con el objetivo de reducir los recursos que demanda la adquisición de información, los agricultores suelen emplear técnicas de muestreo estadístico para reducir la cantidad de mediciones a realizar, y a partir de estas mediciones estimar los valores faltantes mediante técnicas de interpolación espacial. Un problema que surge durante este proceso es que se pierde información de grano fino respecto a la variabilidad espacial del cultivo, lo que implica que las decisiones del agricultor dependerán de información que presenta incertidumbre, producto de la imposibilidad para observar de manera directa todas las mediciones necesarias. A la vista de estos problemas, se hace evidente la necesidad de mejorar la capacidad de los instrumentos de sensado y medición de variables agrícolas, a fin de aumentar la resolución espacial de los datos y reducir los recursos necesarios. Con la necesidad de atender estos y otros desafíos y requerimientos tecnológicos de la actividad agrícola, en las últimas décadas se ha gestado una comunidad científica multidisciplinar conocida como agricultura de precisión, que busca gestionar la variabilidad de los suelos y cultivos a partir de la observación, medición y manejo de las plantaciones mediante el uso de tecnologías de la información, con el objetivo de optimizar la rentabilidad y el rendimiento a la vez

que se preservan los recursos naturales [25]. Uno de los cultivos donde la agricultura de precisión se implementó rápidamente fue en viñedos para vinificación, adoptando el nombre de viticultura de precisión [7, 57]. Concretamente la viticultura de precisión es la gestión de la variabilidad del cultivo de la vid que utiliza las herramientas de la agricultura de precisión junto al conocimiento sobre el comportamiento de esta especie para la optimización de la producción de uvas de calidad superior.

Por otro lado, la naturaleza intrínsecamente compleja, dinámica y no lineal de los sistemas agrícolas requiere tecnologías avanzadas para proporcionar soluciones apropiadas a los problemas del agro [27]. En este sentido, lo avances logrados en el área de la inteligencia artificial y muchas de sus disciplinas colocan a estas tecnologías como soluciones potenciales para muchos problemas complejos de la agricultura. Particularmente, el aprendizaje de máquinas es una de las disciplinas de la inteligencia artificial que ha sido exitosamente aplicada a diversos problemas agrícolas, como ser interpolación espacial y regresión para estimar valores faltantes, modelado de variables mutuamente correlacionadas, estimación de calidad y producción del cultivo, predicción de heladas con mayor granularidad espacial, y más [27, 122]. Otra disciplina de la inteligencia artificial que se destaca por sus importantes aportes en aplicaciones de agricultura de precisión es la visión computacional, mejorando tareas que requieren alguna forma de inspección y evaluación visual de variables, como ser sensado remoto, detección de las partes de una planta, localización de frutos, detección de malezas, y más [154, 172].

Concretamente, el desarrollo de esta tesis se realizó en el ámbito general de la viticultura de precisión, y en particular en el problema de medición autónoma de variables vitícolas para mejorar la calidad y rendimiento de los viñedos, a través de disciplinas propias de la inteligencia artificial, como ser aprendizaje de máquinas y visión computacional. El aporte principal de esta tesis consiste en el estudio, desarrollo e implementación de algoritmos de segmentación, clasificación y detección de objetos en imágenes con aplicación en problemas de medición de variables vitícolas, y que además establecen las bases para la construcción preliminar de un sistema de búsqueda visual activa (BVA) para yemas de vid en condiciones naturales de campo durante la temporada de invierno, cuya validez es verificada en un escenario que simplifica algunos aspectos de la práctica. Este aporte se justifica en la necesidad de mejorar los instrumentos de medición de variables agrícolas a través de su automatización, lo que requiere a su vez entidades autónomas inteligentes que toman decisiones eficientes sobre cómo explorar el entorno para encontrar los objetos de interés.

1.1. Antecedentes

En las últimas décadas el sector agrícola ha producido un cambio fundamental sobre la forma en la que se administran los cultivos alrededor del mundo, recolectando, almacenando y procesando grandes cantidades de información detallada de las plantaciones y su entorno. Estos cambios sustanciales han sido fuertemente estimulados por los avances en las tecnologías de la información y la comunicación (TICs) y su aplicación en la agricultura, originando nuevas comunidades dedicadas al estudio e investigación de métodos y técnicas que mejoren la administración de los cultivos, como ser la agricultura de precisión y la viticultura de precisión.

1.1.1. Agricultura de Precisión

La agricultura de precisión (AP) ofrece la posibilidad de realizar un tratamiento diferencial del cultivo en distintos sectores del terreno, aumentando su productividad y rendimiento, mientras que reduce el impacto ambiental negativo sobre los recursos naturales o, idealmente, lo suprime [25]. La AP surge en el mundo hacia fines de la década de 1980, como respuesta a la necesidad de entender, modelar y controlar las condiciones heterogéneas en la producción de cultivos [173]. Generalmente, los avances tecnológicos en términos de AP se han producido en los países más industrializados y desarrollados, como ser Estados Unidos, Australia y algunos países de Europa [25, 126]. Sin embargo, en Argentina el Instituto Nacional de Tecnología Agropecuaria (INTA) comenzó a evaluar y difundir técnicas de AP a finales de la década de 1990, realizando principalmente experiencias con monitores de rendimiento [108]. Desde entonces, la AP ha ganado popularidad y se ha comenzado a implementar en diferentes niveles para los cultivos más importantes del país, como ser: soja, maíz, trigo, cebada, sorgo, arroz, girasol, maní, entre otros [22, 111].

La variabilidad espacial y temporal del cultivo, el suelo y su entorno, ha sido observada desde la antigüedad, cuando los primeros agricultores comenzaron a cultivar la tierra [143]. Esta propiedad variante está determinada por diversos factores intrínsecamente relacionados entre sí, tales como: (i) variabilidad espacial y temporal del suelo y su topografía; (ii) condiciones macro-climáticas y micro-climáticas de la región; (iii) variabilidad en las propiedades de las plantas individuales; y (iv) malezas, plagas y enfermedades que afectan al cultivo. Si bien históricamente los agricultores han comprendido muchos de estos fenómenos, las limitaciones en los métodos y técnicas agrícolas solo permitieron una gestión homogénea del cultivo sobre todo el campo sembrado. En la actualidad, con el advenimiento de la AP y el uso de las TICs, la determinación de la variabilidad espacial y temporal de ciertos aspectos del cultivo le

han permitido a los agricultores comenzar a administrar de forma diferencial sectores que presentan cierto grado de homogeneidad interna, cuyo tamaño puede variar de unas pocas hectáreas hasta unos pocos metros cuadrados. Con esta tecnología, cada sector del campo puede ser potencialmente sometido a una forma específica de gestión significativamente diferente a la del sector adyacente.

En la práctica, la aplicación de la AP se basa en una combinación de tecnologías fundamentales y sistemas innovadores interactuando en un flujo de trabajo cíclico [173, 148, 25], que puede ser resumido en tres pasos generales:

1. Captura de información sobre la variabilidad del cultivo y su entorno.

En este paso se involucran tecnologías de sensado, observación y medición para cultivos, suelos y variables climáticas, como ser: sistemas de posicionamiento global y banderilleros satelitales; sensores remotos terrestres, aéreos y satelitales; sensores in-situ de rendimiento, biomasa y respuesta de los cultivos; estaciones meteorológicas en red; sensores de suelo; redes de sensores inalámbricos; y más [142, 25]. Cada tecnología empleada para observar y medir variables del cultivo y su entorno a menudo produce un conjunto de datos masivo que codifica información importante acerca de los procesos agrícolas, como ser variaciones espaciales y temporales correlacionadas intra-variable; y correlaciones, lineales o no, inter-variables [142]. Además, dependiendo de las limitaciones de la tecnología y los recursos disponibles, las observaciones y mediciones en campo pueden ser realizadas mediante muestreo estadístico, implicando que los datos obtenidos presenten diferentes resoluciones espaciales. En este escenario de información tan heterogénea y altamente correlacionada es imprescindible el uso de sistemas de procesamiento que faciliten el análisis y la toma de decisiones en el contexto global.

2. Procesamiento y análisis de información para evaluar la variabilidad.

Durante años los geoestadísticos han investigando formas de describir y representar los datos espaciales y temporales a partir de los datos en bruto. Históricamente, estas tareas se han realizado a partir de conjuntos de datos espacialmente ralos o dispersos [148]. Cuando se aumenta la resolución espacial de los datos, esta tarea suele ser cada vez más compleja, requiriendo tanto sistemas de procesamiento de información avanzados para modelar las relaciones intra-variable e inter-variables que afectan a los procesos agrícolas, como sistemas de soporte a las decisiones que asistan a los agricultores en la gestión de sus cultivos. Algunos sistemas pioneros incluyen software para mapeo y visualización de datos de diferentes fuentes y sistemas de información geográfica específicos para la agri-

cultura [25, 173]. Una vez procesada la información disponible, solo queda llevar a la práctica las decisiones tomadas por el agricultor.

3. Aplicación de los resultados en la gestión diferencial de los insumos. La tecnología empleada en este paso es esencialmente un problema de ingeniería. Debido a su potencial comercial, gran parte de este desarrollo ingenieril está siendo impulsado por el sector privado [148]. Ejemplos típicos de estas tecnologías son: equipos de dosificación variable para fertilizantes y pesticidas; equipos de guía automática; sistemas de siembra autónomo; entre otros. En este punto, ya se puede comenzar con un nuevo ciclo del flujo de trabajo de la AP, observando y midiendo cómo ha respondido el cultivo y su entorno al tratamiento diferencial que se aplicó.

Al repasar las tecnologías involucradas en el flujo de trabajo de la AP se distinguen aportes que provienen desde áreas muy variadas de la ciencia, como ser: agricultura, electrónica, química, física, geoestadística, aeroespacial, electromecánica, computación y sistemas de información. En conjunto, la comunidad de científicos e ingenieros de la AP ha desarrollado con éxito complejas plataformas y dispositivos para la solución de problemas específicos [31, 173, 86]. Sin embargo, aún quedan muchos desafíos por resolver. Para lograr un régimen de gestión de sitio específico a nivel de planta individual, todavía es necesario mejorar la capacidad para medir la variabilidad, aumentar la resolución espacial de los datos y disminuir el costo de la tecnología [149, 148]. Por otro lado, la disponibilidad de información masiva y novedosa ha generado nuevos retos para la AP, requiriendo sistemas de extracción de conocimiento y minería de datos para descubrir relaciones y patrones desconocidos y potencialmente útiles [50]. Muchos de los esfuerzos de las investigaciones actuales en la comunidad de AP están siendo enfocadas en esta dirección.

1.1.2. Viticultura de Precisión

De manera similar a la AP, la viticultura de precisión (VP) permite a los viticultores y agrónomos adquirir y gestionar información geo-referenciada detallada sobre las variables involucradas en el cultivo de la vid, afirmando el proceso de toma de decisiones económicas, ambientales, y agronómicas, mejorando el rendimiento del cultivo, la calidad del fruto y la rentabilidad del producto [7, 102]. El objetivo general de la VP coincide, en esencia, con el objetivo general de la AP, pero además presenta objetivos específicos, como ser: predicción de la calidad de la uva; estimación del rendimiento del cultivo; detección de zonas de productividad alta, media y baja; detección y predicción de enfermedades de la vid; aplicación diferencial de fertilizantes; aplicación diferencial

de pesticidas en hojas y racimos; distribución precisa del riego; maximización de la eficiencia de los insumos; minimización del impacto ambiental negativo.

Debido a la estrecha relación que existe entre la AP y la VP es esperable que los avances producidos en una disciplina impacten y beneficien a la otra. Sin embargo, todos los desafíos presentes en la AP, también son desafíos para la VP. Por lo tanto, con el mismo espíritu de la comunidad de AP, muchos de los esfuerzos de la comunidad de VP están siendo dirigidos hacia la solución de estos desafíos compartidos.

1.1.3. Inteligencia Artificial en el Agro

En la búsqueda por hacer frente a los desafíos actuales de la AP, la comunidad ha comenzado a emplear métodos y técnicas avanzadas propias de la inteligencia artificial (IA), disciplina que está tomando un rol cada vez más importante en la actividad agrícola [122]. Por lo tanto, no es extraño encontrar entre las tecnologías de la AP algoritmos de aprendizaje de máquinas (AM) para el procesamiento de información y la generación asistida o automatizada de modelos a partir de grandes conjuntos de datos agrícolas. AM se ocupa de la detección y reconocimiento automático de regularidades o patrones en los datos mediante el uso de algoritmos computacionales [19]. La literatura presenta importantes contribuciones de AM en problemas agrícolas que requieren clasificación, regresión, interpolación espacial, modelado de variables correlacionadas espacialmente, extracción de conocimiento, minería de datos, y más [58, 150, 138, 50, 82, 130, 13, 59], basados en técnicas propias de la IA, como ser algoritmos de agrupamiento, redes neuronales artificiales, máquinas de vector soporte, redes bayesianas, redes markovianas, y más [160, 19, 66, 52]. Otra disciplina dentro de las ciencias de la IA que ha sido empleada en problemas de la AP es visión computacional (VC). El objetivo de los algoritmos de VC es obtener información útil acerca de los objetos y escenas reales capturadas en imágenes digitales [134, 144]. La ventaja de los sistemas de VC es que no sólo reconocen tamaño, forma, color y textura de los objetos, sino que también pueden proporcionar atributos numéricos de los objetos y escenas que se han fotografiado. Existen variadas aplicaciones en AP que utilizan la tecnología de VC, como ser: estimación de la producción; sensado remoto (terrestre y aéreo) para la evaluación de recursos naturales; interpolación espacial de variables agrícolas; estimación de la calidad del producto; segmentación de las partes de una planta; dosificación variable para fertilizantes y pesticidas; localización de frutos; reconocimiento de defectos en la superficie del producto; detección y control de malezas; obtención de índices de vegetación; estimación de la geometría de los frutos; y más [77, 37, 21, 89, 161, 145, 105, 26, 60, 94, 117, 154, 172, 87, 129, 131].

Con todo esto, existe suficiente evidencia para afirmar que los avances en las tec-

nologías de IA, AM y VC, se presentan como alternativas efectivas y económicas para la estimación de muchas variables de interés agrícola. Entre estas tecnologías prometedoras también se encuentran los sistemas de búsqueda visual activa.

1.1.4. Búsqueda visual activa de objetos

La búsqueda visual activa (BVA) de objetos se define como la tarea de buscar eficientemente las instancias de un objeto determinado en un entorno generalmente desconocido a través de agentes autónomos equipados con una cámara y la inteligencia necesaria para realizar dicha tarea [171, 10, 9]. El concepto de búsqueda visual asume la existencia de un agente con la capacidad de sensar el entorno a través de imágenes y la habilidad para determinar si el objeto buscado está presente en el campo de visión del sensor. Por otro lado, el término activo se refiere a la necesidad de cambiar activamente, bajo demanda, los parámetros del sensor (posición y ángulos) hasta que el objeto buscado aparezca en su campo de visión y sea distinguible del resto de la escena. Además, en un espacio de búsqueda desconocido y de gran escala el agente debe encontrar una estrategia eficaz para localizar el objeto a fin de reducir el costo total de la búsqueda, dado que explorar exhaustivamente todo el espacio no es viable al ser un problema NP-completo [171]. En este espacio, el agente de búsqueda puede ejecutar acciones de movimiento y acciones de sensado. Cuando el sensor es una cámara digital, las acciones de sensado están caracterizadas por la pose del agente, los parámetros de la cámara y un algoritmo de detección y/o reconocimiento de objetos. Por otro lado, el costo de la búsqueda depende del problema, pero generalmente suele estar asociada a la minimización de ciertos recursos, como ser el tiempo que se tarda en completar la tarea, la cantidad de acciones ejecutadas, la distancia total recorrida, la cantidad de cómputo requerida, y más. Concretamente, el problema de BVA requiere un algoritmo de planificación que decida qué secuencia de acciones ejecutar para localizar todas las instancias de un objeto de interés, a la vez que se minimiza el costo total de la búsqueda. Este algoritmo es el proceso medular de cualquier algoritmo de BVA, ya que permite realizar la búsqueda de forma eficiente a partir de la incorporación del conocimiento de sentido común relevante para el problema de búsqueda [137]. De la misma manera que el sentido común le permite a un ser humano establecer relaciones entre los elementos del espacio de búsqueda que lo guían en la tarea de encontrar un objeto específico, la integración de este conocimiento dentro del algoritmo le permite al agente establecer relaciones entre los conceptos del entorno y el objeto buscado. Por ejemplo, para el problema particular de buscar yemas de vid en entornos desconocidos, se podría utilizar parte del conocimiento de sentido común que indica que las yemas se desarrollan sobre los nudos de las ramas, las ramas se encuentran unidas a la

estructura de la planta, las plantas están dispuestas sobre las hileras del viñedo. Con esta información, cualquier agente colocado en un entorno de viñedo con la habilidad de reconocer y detectar hileras, plantas, ramas y yemas, debería poder realizar la búsqueda de yemas eficientemente sin mayores complicaciones, de manera similar a como lo haría un humano. Tal como se ilustra en este ejemplo, los algoritmos de BVA se presentan como un enfoque viable en el desarrollo de sistemas autónomos para la medición de variables vitícolas, de alta precisión, operacionalmente factible y de bajo costo. En la práctica, un sistema de estas características permitiría obtener mayor cantidad de información, para un gran número de variables a un costo equivalente (o potencialmente menor) en comparación con los métodos manuales de medición utilizados en la práctica.

1.2. Objetivos de Investigación

La investigación presentada en este documento se enmarca en el problema general de la VP, y en particular en la mejora de la calidad y el rendimiento de los viñedos a través de algoritmos de aprendizaje de máquinas y visión computacional para la captura de datos, análisis y modelado inteligente de los factores involucrados en la producción vitícola del país. Con la visión de contribuir en el campo de la VP, el **objetivo general** de este trabajo se especifica como sigue:

Estudio, aplicación y desarrollo de algoritmos de aprendizaje de máquinas y visión computacional para la medición autónoma de variables vitícolas requeridas en el modelado de patrones de comportamiento espacial asociados a la productividad del viñedo y la calidad de la uva.

En la agricultura, los patrones de comportamiento espacial habitualmente son ajustados mediante técnicas de interpolación espacial [91]. En la literatura existe una gran variedad de técnicas de interpolación estadística (siendo ponderación de distancia inversa y kriging las más utilizadas en la práctica), e incluso técnicas propias de AM (e.g. máquinas de vector soporte y redes neuronales artificiales), todas ellas aplicadas en problemas de diversas disciplinas, como ser geoestadística, hidrología, mineralogía, climatología, agricultura, y más [58, 92, 91]. Los métodos de interpolación se presentan principalmente como una solución efectiva y económica para suplir la falta de datos. En general, las técnicas de interpolación espacial toman un conjunto de mediciones conocidas, obtenidas por muestreo estadístico, y generan modelos para estimar valores desconocidos en otras ubicaciones no observadas del espacio [43]. Idealmente, si la distribución espacial de la muestra es lo suficientemente fina y está uniformemente

distribuida a lo largo del área de estudio, las estimaciones serán muy fiables. Por el contrario, si la muestra presenta baja densidad con datos agrupados en ciertas zonas las estimaciones serán muy poco fiables [74]. Existen muchos otros factores que influyen en la calidad de las estimaciones de los modelos de interpolación, como ser el diseño del muestreo, la distribución espacial de la muestra, la calidad de los datos, la correlación entre las variables primarias y secundarias, y la interacción entre estos factores [163]. Además, dada las limitaciones de los métodos e instrumentos de medición actuales empleados en la agricultura, las técnicas de muestreo estadístico pueden presentar diferentes resoluciones espaciales, que van desde metros hasta kilómetros. En muchos casos, esta resolución espacial puede no ser lo suficientemente "fina" para que una técnica de interpolación logre modelar completamente los patrones de variabilidad espacial del cultivo. De esta manera, las técnicas de interpolación tienen un límite en la precisión de sus modelos que está determinado principalmente por el diseño del muestreo y la distribución espacial de la muestra, i.e. la densidad de las muestras tomadas y las correlaciones espaciales que pueden codificar estas muestras [157]. A fin de evaluar los límites y la calidad de los métodos de interpolación espacial sobre variables de interés vitícola, en el capítulo 2 se presenta un estudio para técnicas de interpolación espacial, tanto estadísticas como de AM, para variables vitícolas espacialmente correlacionadas, a partir de mediciones geoespaciales. La evaluación de cada técnica se realiza sobre la base de la calidad de los modelos de interpolación obtenidos y las limitaciones en el modelado de los patrones espaciales que sufren frente a la falta de datos.

Por otro lado, uno de los objetivos principales de la AP es lograr un régimen de gestión de sitio específico a nivel de planta individual [104]. Cuando las técnicas de interpolación no pueden generalizar adecuadamente los patrones de correlación espacial, la gestión precisa de sitio específico no es viable. Una baja capacidad de generalización en los métodos de interpolación habitualmente se asocia a la falta de datos durante la generación de los modelos, i.e. procesos de muestreo de resolución insuficiente. Por lo tanto, uno de los desafíos actuales de la AP consiste en aumentar la resolución espacial en la adquisición de datos y reducir el costo de la tecnología de medición, a fin de mejorar los modelos de interpolación de variables que intentan predecir la productividad y calidad del cultivo [84]. Para hacer frente a estos desafíos, en las últimas décadas la AP ha dirigido muchos de sus esfuerzos a la investigación y desarrollo de métodos autónomos o semiautónomos para realizar tareas de inspección, medición y evaluación de variables agrícolas de forma más objetiva, rápida y económica. En este sentido, es importante notar que en la actualidad existen trabajos innovadores e inspiradores que presentan avances en muchas de estas tareas. Por ejemplo, en Aquino et

al. (2015) [5] se presenta una aplicación para smartphones, basada en técnicas de VC, que permite contar de forma automática, no invasiva, y directamente en el viñedo, el número de flores en un racimo en periodo de inflorescencia. La aplicación, desarrollada para Android disponible en Google Play¹, requiere colocar un tablero negro detrás del racimo antes de capturar la imagen con la cámara del teléfono. En Wang et al. (2013) [159] se presenta un sistema basado en VC para detectar y contabilizar manzanas a partir de imágenes, que utiliza iluminación artificial durante la noche, montado en un vehículo terrestre que se mueve de forma autónoma entre las hileras de los frutales. Las imágenes son adquiridas de manera secuencial a distancias homogéneas de la planta, asumiendo que esa distancia es suficiente para detectar los frutos. Sin embargo el sistema no toma decisiones sobre cómo explorar el entorno para mejorar el proceso de detección. En Das et al. (2015) [45] utilizan un conjunto de sensores para determinar la morfología (volumen de la canopia, índice de área foliar y recuento de fruta) en plantas vid, naranja y manzana, que consta de un escáner láser, cámaras multiespectrales, una cámara de imagen térmica y sensores de navegación. Presentan además el uso del sistema en dos situaciones prácticas: montado en un vehículo aéreo no tripulado (VANT) multirrotor y montado en un arnés sobre el hombro de una persona. Si bien el sistema contiene sensores de navegación dedicados, no se dan detalles de cómo se utilizan para navegar el entorno en el caso del VANT. Esta información tiene el potencial de ser aprovechada por un algoritmo de planificación de vistas que permita mejorar la calidad de las mediciones realizadas por el sistema. Si bien estos trabajos representan avances importantes en el área, en la práctica, estos métodos no resultan atractivos para los agricultores debido a los altos costos del equipamiento requerido (cámaras hiperespectrales y térmicas, escáneres láser, sensores, etc.), como así también ciertos inconvenientes operacionales en campo (entornos controlados, simplificación del problema, vehículos terrestres, operarios humanos, fondos artificiales, etc.). Por otro lado, en Roy et al. (2015) [129] proponen un enfoque para la estimación del rendimiento en manzanos, en condiciones naturales de campo, mediante la utilización de un VANT equipado con un par de cámaras RGB de baja resolución (800x600 pixels) y un sistema de visión estéreo para detectar manzanas y estimar su diámetro. Este trabajo incorpora una estrategia de planificación de vistas elemental asumiendo que los árboles se encuentran típicamente en filas. Esta estructura permite establecer una estrategia de planificación fuera de línea, la cual se limita a recorrer el espacio a través de una trayectoria en zigzag respecto al plano de la hilera. Más allá de su simpleza, esta estrategia permite alcanzar resultados muy prometedores, resaltando los beneficios de realizar un proceso de planificación de vistas durante la tarea

¹https://play.google.com/store/apps/details?id=com.ur.android.vitisflowerbeta

de medición de variables. Los resultados presentados en este último trabajo, además sugieren fuertemente que muchas tareas de medición pueden lograrse sin recurrir a sensores costosos y condiciones de entorno controladas. Con todo esto se considera que existe suficiente evidencia para afirmar que los avances en las áreas de AM y VC se presentan como alternativas efectivas y económicas para la estimación de muchas variables de interés agrícola. En la misma línea, los esfuerzos de esta investigación se han volcado hacia el estudio y desarrollo de métodos autónomos para la inspección y medición autónoma de variables vitícolas, con aplicación en el desarrollo de un sistema de búsqueda visual activa de yemas de vid.

En este contexto, uno de los problemas inspiradores para esta investigación consiste en desarrollar la tecnología necesaria para implementar una plataforma autónoma de medición y evaluación de variables vitícolas que sean susceptibles de ser obtenidas por inspección visual a partir de algoritmos propios de AM y VC. Esta tecnología puede proporcionar a los viticultores un mecanismo operacionalmente factible, de alta precisión y bajo costo, para obtener cantidades masivas de información (como ser localización, características geométricas y rasgos visuales) de los elementos de una planta de vid que se emplean en la estimación de variables de interés vitícola. Particularmente, este trabajo se enfoca en el desarrollo de algoritmos para el problema de búsqueda autónoma de yemas de vid en condiciones naturales de campo durante la temporada de invierno. Una tecnología efectiva para la búsqueda autónoma de yemas basada en algoritmos avanzados de AM y VC presenta importantes oportunidades en dos aspectos principales: (i) aumentar la resolución espacial de los procesos de medición de variables vitícolas (e.g. conteo de yemas, conteo de yemas brotadas y no brotadas, brotes de yemas francas, estado de desarrollo de la yema, y más); y (ii) desarrollar nuevas aplicaciones con impacto en la industria, como ser poda de vid autónoma, fenotipado de plantas de vid, reconstrucción 3D de la estructura de la planta y sus componentes, entre otros. Además, es esperable que una vez resuelto este problema para yemas, el desarrollo de algoritmos de búsqueda para otros elementos de la vid requiera considerablemente menos esfuerzos. Con esto, el objetivo específico de la tesis se enuncia como sigue:

Estudio, implementación y desarrollo de algoritmos de segmentación, clasificación y detección de objetos en imágenes con aplicación al problema de detección de yemas de vid, que establecen las bases para la construcción de un sistema de búsqueda visual activa (BVA) que cumple con los siguientes requerimientos: (i) las imágenes son tomadas mediante una cámara RGB en condiciones naturales de campo, durante la temporada de invierno, cuando las plantas han perdido sus hojas y frutos, a fin de evitar la oclusión de las yemas; (ii) el proceso de

búsqueda comienza en vistas donde las yemas tienen una resolución menor a 20×20 píxeles, lo que corresponde a una distancia aproximada a la yema de 75 centímetros, y continúa realizando acercamientos hacia las zonas donde es probable que se encuentren yemas, hasta que la distancia entre la cámara y la yema sea suficiente para ser detectada de manera precisa y poder estimar de forma autónoma sus características geométricas y visuales; (iii) la elección de la siguiente vista a explorar se determina por un proceso de planificación de vistas sobre la base del conocimiento de sentido común del problema codificado en el algoritmo; (iv) la navegación de las vistas está restringida sólo a movimientos lineales respecto al plano que forma la hilera del viñedo, i.e. la cámara que captura las imágenes solo posee 3 grados de libertad (movimientos horizontales, verticales y acercamientos).

Vale la pena resaltar que el desarrollo de un algoritmo de planificación de vistas para un sistema de búsqueda visual activa de yemas de vid presenta varios aspectos desafiantes: (i) el espacio de búsqueda corresponde a escenas típicas de viñedos, en exterior, sin alterar las condiciones naturales del entorno; (ii) dado que la yema es un elemento de la naturaleza, las características visuales de cada instancia de yema son diferentes; (iii) los atributos de forma, color y tamaño de la yema respecto al resto de los elementos de la escena requiere vistas tomadas a una distancia muy cercana (12 centímetros aproximadamente) para que la detección sea efectiva; y (iv) el problema requiere encontrar todas las instancias del objeto yema en el espacio, siendo desconocida la cantidad exacta de yemas. Frente a estos desafíos, en imprescindible integrar algún conocimiento del sentido común dentro del algoritmo de planificación de vistas, tal como se explicó en la sección 1.1.4. Para el problema de búsqueda de yemas presentado en el objetivo específico, es relevante conocer que las yemas se encuentran en lugares particulares de la estructura de la planta y con una alta correlación espacial con otros elementos de la vid, tal como se ejemplifica en la figura 1.1: las yemas se desarrollan en las ramas (o sarmientos); las ramas pertenecen a la clase de material madera de la vid; el resto de los elementos de la escena pueden ser omitidos durante la búsqueda. Esta información, codificada a partir del conocimiento del sentido común, es crucial para limitar el espacio de búsqueda y mejorar la eficiencia de la detección y exploración en cualquier entorno natural de viñedo desconocido, a la vez que permite hacer suposiciones de las zonas del espacio que potencialmente contienen instancias de una yema, información útil cuando las yemas son tan pequeñas que no sería posible detectarlas por sí solas. Es evidente entonces que un algoritmo de planificación de vistas que integra el conocimiento de sentido común del problema se construye a partir del desarrollo de diferentes herramientas, como ser segmentación de madera, de-

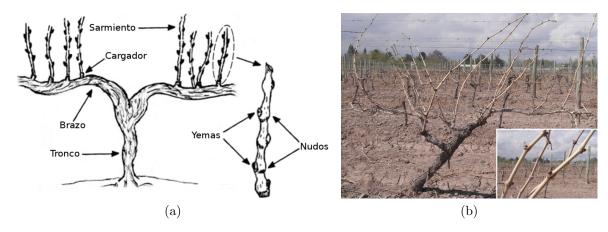


Figura 1.1: (a) Esquema de una planta de vid en invierno. (b) Planta de vid en su entorno natural, durante la temporada de invierno.

tección de ramas y detección de yemas de la vid. Sin embargo estas herramientas por sí solas tienen potenciales aplicaciones para la medición autónoma de otras variables vitícolas, como ser: peso de madera nueva (i.e. ramas desarrolladas en la temporada anterior), longitud de entrenudos, longitud de ramas, conteo de yemas, diámetro de tronco, diámetro de ramas, reconstrucción 3D de la estructura de la planta, fenotipado autónomo de plantas, poda autónoma de la vid, y más. Por lo tanto, la tesis procede explicando en los capítulos 3 y 4 dos de los algoritmos requeridos para construir el conocimiento de sentido común, i.e. segmentación de madera y clasificación de imágenes para detección de yemas, y se discuten sus resultados independientemente sobre dos aplicaciones concretas para la medición autónoma de variables vitícolas: diámetro de tronco y detección de yemas. Finalmente, los detalles del algoritmo de detección de ramas se posterga hasta el capítulo 5, donde además se ensamblan todas estas herramientas para resolver un nuevo problema específico: la búsqueda visual activa de yemas de vid.

1.3. Organización del Documento

Este documento se organiza en 6 capítulos. Seguido de la Introducción, el capítulo 2 presenta un estudio sobre técnicas de interpolación espacial para variables vitícolas mediante algoritmos de aprendizaje de máquinas y técnicas de interpolación estadística, que justifica el desarrollo de nuevas herramientas que permitan mejorar los instrumentos de medición de variables agrícolas. En el capítulo 3 se presentan diferentes enfoques para la segmentación de madera en imágenes de vid (i.e. tronco, sarmientos, brotes), basado en algoritmos de aprendizaje de máquinas y visión computacional, a partir de imágenes tomadas en condiciones naturales de campo, y se presentan dos

aplicaciones prácticas concretas para evaluar el impacto del enfoque. El capítulo 4 presenta un algoritmo de clasificación de imágenes para la detección de yemas de plantas de vid, también basado en técnicas de visión computacional y aprendizaje de máquinas, y se discuten sus resultados para el problema específico de la detección autónoma de yemas. En el capitulo 5 se introduce un enfoque para la búsqueda visual activa de yemas a partir del conocimiento del sentido común construido mediante las herramientas anteriores y presenta un algoritmo para detección de ramas requerido durante la construcción de este conocimiento. Finalmente, en el capítulo 6 se presentan las conclusiones principales de esta investigación.

1.4. Publicaciones Relevantes

En esta sección se detallan brevemente las publicaciones relevantes para la tesis desarrollada es este documento. Los resultados que inspiraron el capítulo 2 fueron presentados inicialmente en el trabajo de conferencia titulado Interpolación espacial mediante Aprendizaje Inductivo en viñedos de la provincia de Mendoza, Argentina, enviado al Workshop de Agentes y Sistemas Inteligentes (WASI), durante el XVIII Congreso Argentino de Ciencias de la Computación 2012 (CACIC), realizado en la ciudad de Bahía Blanca, Buenos Aires, Argentina [26]. De la misma manera, el capítulo 3 fue elaborado a partir del trabajo titulado Segmentación de Imágenes en Viñedos para la Medición Autónoma de Variables Vitícolas, presentado en el Argentine Symposium on Artificial Intelligence (ASAI), durante la edición 41 de las Jornadas Argentinas de Informática 2012 (JAIIO), llevadas a cabo en la ciudad de La Plata, Buenos Aires [120]. Además, el desarrollo de la aplicación para diámetro de tronco presentada en este capítulo se encuentra disponible como un reporte técnico titulado Computer Vision Approach for Low Cost, High Precision Measurement of Grapevine Trunk Diameter in Outdoor Conditions, disponible en el repositorio de publicaciones científicas arXiv [118]. Finalmente, el artículo titulado Image classification for detection of winter grapevine buds in natural conditions using scale-invariant features transform, bag of features and support vector machines, actualmente en prensa en la revista científica Computers and Electronics in Agriculture, dio forma al capítulo 4 [119].

Capítulo 2

Comparación de Algoritmos de Interpolación Espacial con Evaluación del Impacto en la Cantidad de Datos Disponibles

Como se explicó en el capítulo 1, en la agricultura los patrones espaciales del cultivo suelen ser modelados a través de técnicas de *interpolación espacial*. Estos métodos se presentan como una herramienta efectiva y económica que permite la estimación de valores faltantes a partir de modelos generados desde un conjunto de mediciones conocidas, las cuales son obtenidas por un proceso de muestreo estadístico espacial. Sin embargo, los métodos de interpolación espacial tienen un límite en la precisión de sus estimaciones que está determinada, entre otros factores, por la densidad espacial de las muestras capturadas y las correlaciones espaciales que pueden codificar estas muestras. Un método de interpolación que produce estimaciones poco precisas presenta una capacidad de generalización muy baja, problema que es habitualmente asociado a la falta de datos durante la generación de los modelos, i.e. procesos de muestreo de resolución insuficiente. De esta manera, una forma directa de aumentar la calidad de los métodos de interpolación consiste en elevar la resolución espacial durante la adquisición de datos y reducir el costo de la tecnología de medición a través del desarrollo de herramientas autónomas de inspección y medición de variables agrícolas.

Concretamente, en este capítulo se evalúan los modelos de interpolación generados mediante dos algoritmos de aprendizaje de máquinas, como ser máquinas de vector soporte y máquinas de vector soporte local. Los resultados de estas técnicas son comparados con los dos métodos de interpolación estadística de uso más extendido empleadas en agricultura de precisión para el modelado de datos agrícolas: ponderación

de distancia inversa y kriging. Los resultados reportados en este trabajo muestran, en general, que los algoritmos de aprendizaje obtienen resultados de mayor calidad. Además se presenta un análisis basado en curvas de aprendizaje, una herramienta de diagnóstico que pone de manifiesto las limitaciones de los métodos de AM para el problema de interpolación en algunas variables vitícolas presentadas en este capítulo. Esta herramienta permite verificar de manera empírica si para una variable en particular es necesario aumentar la cantidad de datos durante el modelado o utilizar modelos más complejos para mejorar la calidad de la interpolación espacial.

2.1. Introducción

En la actividad agrícola, las decisiones que debe tomar el agricultor muchas veces dependen directamente de información obtenida a través de mediciones de variables que pueden ser muy costosas de obtener. Una posible solución a este problema consiste en tomar una muestra aleatoria pequeña de la población total y utilizar técnicas de interpolación espacial para inferir los valores restantes. De esta manera se pueden obtener modelos de predicción de productividad y calidad ante la falta de datos [4]. Sin embargo, la calidad de los modelos interpolados depende fuertemente de las correlaciones espaciales propias de la variable en estudio y de la densidad de muestras tomadas en campo. La interpolación es una técnica que permite calcular valores de variables en aquellos lugares donde no se pudieron realizar mediciones. La interpolación espacial asume que estas variables son continuas en el espacio. Esto permite la estimación del valor de la variable en cualquier lugar dentro de los límites de los datos disponibles. Otra suposición es que la variable es espacialmente dependiente, indicando que valores cercanos son más propensos a ser similares que valores lejanos. El objetivo de la interpolación espacial es crear una superficie que pretende representar mejor la realidad empírica [11]. Desafortunadamente, la agricultura de precisión (AP) se ha visto limitada al intentar modelar e interpolar la información medida utilizando exclusivamente las técnicas estadísticas tradicionales. La información provista por la AP no solo presenta variaciones espaciales de pequeña escala y con gran correlación espacial, sino que además presenta correlaciones fuertes, muchas veces no lineales, entre valores de las distintas variables medidas. El primero de estos problemas puede ser mitigado elevando la densidad espacial de las muestras tomadas, aunque en la práctica esto podría ser inviable producto del aumento del costo derivado de tomar mayor cantidad de muestras a partir de las herramientas de medición actuales. Mientras que el segundo problema requiere de técnicas avanzadas de procesamiento de datos multivariables y reconocimiento de patrones no-lineales, como ser algunos algoritmos propios

del aprendizaje de máquinas. Es por eso que, entre el conjunto de tecnologías que conforman la AP se pueden hallar técnicas de aprendizaje de máquinas (AM) para la generación asistida o automatizada de modelos agrícolas [91].

Este capítulo está orientado al estudio, implementación y análisis de dos técnicas propias de AM, como ser máquinas de vector soporte (MVS) [28, 19] originalmente propuesto para interpolación espacial por Kanevski et al. (2000) [81], y máquinas de vector soporte local (MVSL) [24, 58], aplicadas al problema de interpolación de variables vitícolas a partir de mediciones geo-espaciales. La calidad de estos algoritmos es comparada con dos técnicas de interpolación espacial estadística muy empleadas por la AP en el modelado de datos agrícolas: ponderación de distancia inversa (PDI) [135] y kriging [42, 43]. Tanto PDI como kriging presentan limitaciones importantes que las técnicas de aprendizaje no poseen. Es posible demostrar teóricamente que kriging es mejor interpolador que PDI [43], pero es automatizable solo parcialmente ya que requiere ajustar una función de interacción espacial llamada variograma [103, 42], proceso que en la práctica requiere de intervención humana. Otro problema con kriging es que asume una media espacial nula. Esto puede resolverse aplicando kriging al resultado de restarle a los valores medidos la media espacial. Esta solución requiere sin embargo conocer esta media espacial, lo que a su vez requiere de un modelo del proceso subyacente. En contraste, las técnicas de aprendizaje permiten la generación automatizada de modelos de interpolación a partir de los datos. MVS se hizo popular para resolver problemas de clasificación y luego se extendió a problemas de regresión. Una revisión completa del tema se puede encontrar en [19] y [28], así como su aplicación a problemas de interpolación espacial [81]. Más tarde se propuso el algoritmo MVSL [24] como una mejora para problemas en los que los datos no se encuentran distribuidos uniformemente. Este método puede mejorar significativamente los resultados de interpolación mediante la construcción de múltiples modelos locales en lugar de un único modelo global [58].

2.2. Elementos de Trabajo y Metodología

En esta sección se presentan detalles sobre los datos vitícolas específicos empleados durante la experimentación, así como también se introducen brevemente los métodos de interpolación espacial y los detalles de implementación de los algoritmos de AM usados para interpolación, i.e. MVS y MVSL. Finalmente, se presentan las medidas de calidad utilizadas para evaluar la respuesta de las cuatro técnicas de interpolación.

Todos los algoritmos empleados en este trabajo tienen por objetivo realizar una interpolación para estimar los valores faltantes de variables con valores espacialmente

Variable	Descripción
BCW	Peso de Racimo
BUDS	Yemas no Brotadas
BWH	Peso de Bayas
BXB	Cantidad de Bayas
DDT	Diámetro de Tronco
LEN	Longitud de Entrenudo
MAD	Longitud de Primer Alambre Descubierto
PCLA	Plantas Por Claro
PDP	Peso de Poda
RDP	Riqueza de Poda
SHC	Brotes Totales
SHOOT	Brotes de Yemas Francas
SXL	Longitud de Brote

Tabla 2.1: Descripción de las variables vitícolas abordadas.

dependiente, i.e. estimar aquellos puntos de la superficie en donde no están disponibles las mediciones, pero sí se cuenta con mediciones en su entorno espacial cercano. Las técnicas estadísticas evaluadas en este capítulo incluyen PDI, un método de interpolación determinista, y kriging, un método de interpolación estocástico. Cada uno presenta algunas ventajas frente al otro que se explican en los apartados 2.2.1 y 2.2.2. Para el caso de los algoritmos de AM, los métodos de interpolación espacial propuestos se basan en un enfoque de aprendizaje supervisado para obtener modelos de regresión a partir de los datos disponibles para cada variable vitícola, lo que implica las siguientes tareas: (i) procesamiento del conjunto de datos para las etapas de entrenamiento y evaluación de los algoritmos; (ii) aprendizaje de los modelos de regresión; y (iii) evaluación de la calidad de los modelos sobre el conjunto de datos. Estas etapas se explican en detalle más adelante en esta sección.

Por otro lado, los datos necesarios para llevar a cabo la experimentación propuesta en este capítulo fueron suministrados en el año 2011 por la empresa Xoomcode S.A., una empresa local especializada en servicios de geolocalización. Esta información se encuentra distribuida en diferentes conjuntos de datos compuestos de mediciones tomadas en viñedos de importantes bodegas mendocinas (e.g. Agrícola Altamira, Cobos, Foster, Las Divas, Serenade, Teresa Mosso y Vistalba), sobre un total de 13 variables que intervienen en procesos vitícolas de distintas variedades de uva, como ser Malbec, Cabernet Sauvignon y Chardonnay. En la tabla 2.1 se puede consultar una breve descripción de las variables abordadas en este trabajo.

Por otro lado, los conjuntos de datos empleados en la experimentación están conformados por diferentes variables medidas en 7 fincas para 3 variedades de uva. Para

Conjuntos	D	escript	ores E	stadíst	icos
\det° de datos	N	$ ilde{X}$	S^2	Max	Min
BCW_{C-CS}	141	90,1	24,2	170	27
BCW_{C-M}	333	64,7	32,1	185	3
$BCW#1_{F-C}$	58	125,6	28,9	190	45
$BCW\#2_{F-C}$	39	129,1	31,9	190	45
BCW_{S-M}	90	88,7	50,1	325	14
$BUDS_{V-M}$	80	10,7	8,75	66	2
BWH_{S-M}	90	83,8	47,7	310	13
BXB_{S-M}	90	55,2	28,3	177	5
DDT_{V-M}	30	0,034	0,012	0,07	0,02
LEN_{V-M}	80	0,034	0,011	0,056	0,014
MAD_{V-M}	130	2,85	1,16	$5,\!52$	$0,\!34$
$PCLA_{F-C}$	71	5,06	0,843	6	3
PDP_{AA-M}	153	655,1	260,9	1490	220
RDP_{AA-M}	299	76,9	8,44	98	52
RDP_{F-C}	39	90,1	13,8	118	51
RDP_{LD-M}	386	120,8	12,1	163	45
RDP_{TM-M}	108	72,1	14,3	107	27
RDP_{V-M}	130	63,7	18,8	106	9
$RDP#1_{C-CS}$	142	77,3	11,5	112	52
$RDP#1_{C-M}$	257	67,4	22,1	132	8
$RDP\#2_{C-CS}$	142	82,4	14,6	134	54
$RDP\#2_{C-M}$	239	83,5	24,3	152	15
SHC_{V-M}	80	65,9	19,1	123	16
$SHOOT_{V-M}$	80	52,5	14,4	91	15
SXL_{V-M}	80	0,031	0,014	0,08	0,01

Tabla 2.2: Descriptores estadísticos simples para cada conjuntos de datos, donde N es el tamaño del conjunto de datos, \tilde{X} la media aritmética de la variable, S^2 su desviación estándar, Max su valor máximo y Min su valor mínimo.

diferenciar una misma variable que ha sido medida en diferentes fincas, o en la misma finca pero para distintos tipos de uva, se hace uso de subíndices que acompañan al nombre de la variable, donde la primer letra corresponde a la finca y la segunda al tipo de uva. Por ejemplo, C-CS para Cobos-Cabernet Sauvignon, F-C para Foster - Chardonnay, AA-M para Agrícola Altamira - Malbec, y más. En la tabla 2.2 se presentan los 25 conjuntos de datos disponibles durante la experimentación, junto con algunos descriptores estadísticos simples que caracterizan a cada uno.

2.2.1. Ponderación de distancia inversa

PDI se basa en el supuesto de que los valores cercanos contribuyen en mayor medida a los valores interpolados que valores más alejados. En otras palabras, para este

método la influencia de un punto conocido es inversamente proporcional a la distancia desde la ubicación del punto que se está estimando. En problemas de interpolación, la forma general para obtener un valor interpolado u en un punto dado $\mathbf{x_0}$, basado en un conjunto de valores $u_i = u(\mathbf{x}_i)$ conocidos, con i = 1, 2, ..., N, está dada por:

$$u(\mathbf{x}_0) = \begin{cases} \sum_{i=1}^{N} w_i(\mathbf{x}_0, \mathbf{x}_i) u_i \\ \sum_{i=1}^{N} w_i(\mathbf{x}_0, \mathbf{x}_i) \end{cases}, \quad \text{si } d(\mathbf{x}_0, \mathbf{x}_i) \neq 0 \text{ para todo } i \\ u_i, \qquad \qquad \text{si } d(\mathbf{x}_0, \mathbf{x}_i) = 0 \text{ para algún } i \end{cases}$$
(2.1)

donde \mathbf{x}_0 denota un punto arbitrario de valor no conocido, \mathbf{x}_i es un punto muestreado con valor u_i conocido, $d(\mathbf{x}_0, \mathbf{x}_i)$ es la distancia de \mathbf{x}_0 a \mathbf{x}_i , N es el tamaño de la muestra, y $w_i(\mathbf{x}_0, \mathbf{x}_i)$ es la función de interpolación. La forma más simple de PDI utiliza una función de interpolación definida por:

$$w_i(\mathbf{x}_0, \mathbf{x}_i) = \frac{1}{d(\mathbf{x}_0, \mathbf{x}_i)^p}$$
(2.2)

donde p es un número real positivo arbitrario llamado parámetro de potencia. Generalmente la distancia se define como:

$$d_i(\mathbf{x}_0, \mathbf{x}_i) = |\mathbf{x}_0 - \mathbf{x}_i| \tag{2.3}$$

En la ecuación 2.2 los valores de $w_i(\mathbf{x}_0, \mathbf{x}_i)$ disminuyen a medida que aumenta la distancia entre \mathbf{x}_0 y \mathbf{x}_i . Además, valores grandes de p dan mayor influencia a los puntos más cercanos al punto interpolado. Habitualmente en la práctica, el parámetro de potencia se establece en p=2, siendo este valor el utilizado en este trabajo para todas las variables disponibles. Por otro lado, según la ecuación 2.1, el valor u_i de cada punto \mathbf{x}_i conocido debe mantenerse sin alteraciones, tal como fue medido durante el muestreo. Para conocer más detalles sobre PDI, el lector interesado puede consultar varios trabajos disponibles en la literatura [135, 35].

2.2.2. Kriging

Al igual que en PDI, kriging utiliza una ponderación que asigna más influencia para aquellos puntos conocidos más cercanos al punto que se está interpolando. Sin embargo kriging no es determinista, ya que extiende la forma general de PDI para incluir componentes aleatorios, donde la localización exacta del punto no es conocida por la

función. De esta manera, kriging depende de las relaciones espaciales y estadísticas para calcular la superficie [43]. El objetivo de kriging es estimar el valor de una función real desconocida Z en un punto \mathbf{x}_0 , dados los valores de la función en su *vecindario*, i.e. los \mathbf{x}_i puntos más cercanos a \mathbf{x}_0 , para i=1,2,...,N. Un estimador kriging es lineal ya que para obtener una estimación de $Z(\mathbf{x}_0)$ se realiza una combinación lineal de los valores $Z(\mathbf{x}_i)$:

$$\hat{Z}(\mathbf{x}_0) = \sum_{i=1}^{N} \lambda_i Z(\mathbf{x}_i)$$
(2.4)

donde los parámetros λ_i son pesos asignados al valor de cada punto \mathbf{x}_i dependiendo de su ubicación relativa al punto \mathbf{x}_0 . Los pesos λ_i son calculados de acuerdo a dos requerimientos particulares: (i) conseguir una estimación de mínima varianza, lo que implica que la dispersión del error entre los valores estimados y los valores reales debe ser mínima, ya que a mayor dispersión más impreciso será el estimador; y (ii) evitar el sesgo causado por eventuales agrupamientos espaciales de los datos muestreados, lo que implica que la media de las estimaciones debe ser igual a la media de los valores reales. Por lo tanto, suponiendo que los datos muestreados \mathbf{x}_i son el resultado de un proceso aleatorio $Z(\mathbf{x})$, el objetivo es determinar los pesos λ_i que minimizan la varianza del error de predicción $\varepsilon(\mathbf{x}_0)$

$$\sigma_{\varepsilon(\mathbf{x}_0)}^2 = Var \left[\hat{Z}(\mathbf{x}_0) - Z(\mathbf{x}_0) \right]$$
 (2.5)

bajo el requerimiento de insesgamiento

$$E[\hat{Z}(\mathbf{x}_0) - Z(\mathbf{x}_0)] = 0$$
(2.6)

Durante el proceso de minimización de la varianza, los pesos λ_i son calculados como soluciones de un sistema de ecuaciones lineales. Además, para estimar la varianza del error de predicción $\sigma_{\varepsilon(\mathbf{x}_0)}^2$ se requiere conocer la función de covarianza del proceso, obtenida mediante un modelo especial llamado variograma. Un variograma es una herramienta que permite analizar el comportamiento espacial de una variable sobre un área definida [90]. Para obtener un variograma es necesario ajustar empíricamente los datos muestreados a una curva a partir de una familia de curvas propuestas, lo que requiere de intervención humana. Por lo tanto, la creación de un modelo de variograma requiere de un experto humano con amplio conocimiento experimental del problema. El desarrollo análitico del proceso de minimización de la varianza y el proceso para obtener un modelo de variograma quedan fuera del alcance de este trabajo. Para conocer más detalles sobre estos aspectos se puede consultar diferentes fuentes disponibles en la literatura [43, 61, 39, 90].

Es posible demostrar analíticamente que kriging es mejor interpolador que PDI, sin embargo su desventaja principal es que requiere intervención humana y un buen conocimiento experimental de la variable para calcular e interpretar empíricamente los datos seleccionados, ajustarlos a un modelo teórico y validar el modelo. Además, el hecho de que kriging es un estimador lineal, representa una desventaja cuando los patrones de correlación espacial que se están intentando modelar no son lineales

Por último, existen diversos tipos de modelos teóricos que permiten ajustar y validar variogramas según los datos muestreados. En este trabajo, los modelos de variograma generados para kriging han sido del tipo *exponencial*, *esférico*, *gaussiano* y *lineal* [43], dejando todos sus parámetros por defecto según la librería utilizada¹.

2.2.3. Máquinas de vector soporte para regresión

Las máquinas de vectores soporte (MVS) fueron introducidas en los años 90 por Vapnik y sus colaboradores [153]. Aunque originariamente las MVS fueron pensadas para resolver problemas de clasificación binaria, pueden adaptarse para resolver problemas de regresión.

A continuación se presentan las intuiciones básicas de MVS para regresión extraídas principalmente de Smola y Schölkopf (2004) [140].

Dado un conjunto de datos de entrenamiento $S = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_N, y_N)\}$, donde $\mathbf{x}_i \in \mathbb{R}^d$ e $y_i \in \mathbb{R}$, para i = 1, 2, ..., N, se asume que los valores y_i de todos los ejemplos de S se pueden ajustar mediante una función lineal. El objetivo del algoritmo es encontrar una función $f(\mathbf{x})$ que comete como máximo un error ε entre el valor predicho $f(\mathbf{x}_i)$ y el valor real de y_i , y al mismo tiempo es lo más plana posible. En otras palabras, no se le da importancia a los errores, siempre y cuando estos sean menores que ε , pero no se acepta cualquier desviación mayor que ésta. Para encontrar esta función, es necesario primero descubrir los parámetros $\mathbf{w} = (w_1, ..., w_d)$ tal que:

$$f(\mathbf{x}_i) = \langle \mathbf{w}, \mathbf{x} \rangle + b \tag{2.7}$$

con $\mathbf{w} \in \mathbb{R}^d$ y $b \in \mathbb{R}$, donde $\langle \cdot, \cdot \rangle$ denota el producto punto en \mathbb{R}^d . El requerimiento de *planicidad* en el caso de la ecuación 2.7, significa que se buscan valores pequeños para los \mathbf{w}_i . Una manera de asegurar esto es minimizar la norma del vector \mathbf{w} , i.e. $\|\mathbf{w}\|^2$. Formalmente se puede escribir este problema como un problema de optimización

https://cran.r-project.org/web/packages/gstat/index.html

convexa que requiere:

minimizar
$$\frac{1}{2} \|\mathbf{w}\|^2$$

sujeto a
$$\begin{cases} y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle - b & \leq \varepsilon \\ \langle \mathbf{w}, \mathbf{x}_i \rangle + b - y_i & \leq \varepsilon \end{cases}$$
(2.8)

La principal razón para elegir esta función es la de permitir cierta dispersión en la función solución, de tal forma que todos los ejemplos que quedan confinados en la región definida por $\pm \varepsilon$ no serán considerados vectores soporte. De esta forma se reducirá significativamente el número de éstos. Dado que en la práctica es muy difícil que los ejemplos de entrenamiento se ajusten al modelo lineal con un error de predicción igual a cero, se recurre al concepto de margen blando. Para ello, se definen dos variables de holgura, ξ_i^+ y ξ_i^- que permiten cuantificar la magnitud de dicho error. De esta manera, la variable $\xi_i^+ > 0$ cuando $f(\mathbf{x}_i) - y_i > \varepsilon$, y en otro caso su valor es cero. De forma similar, la variable $\xi_i^- > 0$ cuando $y_i - f(\mathbf{x}_i) > \varepsilon$, y en otro caso es cero. Al introducir las dos variables de holgura en la ecuación 2.8 es posible llegar a la siguiente formulación del problema de optimización [152]:

minimizar
$$\frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{N} (\xi_i^+ + \xi_i^-)$$
sujeto a
$$\begin{cases} y_i - \langle \mathbf{w}, \mathbf{x}_i \rangle - b & \leq \varepsilon + \xi_i^- \\ \langle \mathbf{w}, \mathbf{x}_i \rangle + b - y_i & \leq \varepsilon + \xi_i^+ \\ \xi_i^+, \xi_i^- & \geq 0 \end{cases}$$
(2.9)

La constante C>0 determina el punto de equilibrio entre planicidad de f y la cantidad de desviaciones mayores que ε se tolera. Para esto, se utiliza la denominada función de pérdida ε -insensible, caracterizada por ser una función lineal con una zona insensible de anchura ε en la que el error es nulo (zona sombreada en la figura 2.1), y está definida por:

$$|\xi|_{\varepsilon} := \begin{cases} 0 & \text{si } |\xi| \le \varepsilon \\ |\xi| - \varepsilon & \text{en otro caso} \end{cases}$$
 (2.10)

La figura 2.1 representa la situación gráficamente. Sólo los puntos fuera de la región sombreada contribuyen al costo, penalizando estas desviaciones de forma lineal.

El problema de optimización de la ecuación 2.9 se puede resolver más fácilmente a través de su formulación dual. Además, esta formulación proporciona la clave para extender el algoritmo a funciones no lineales. En el caso de que los datos no sean

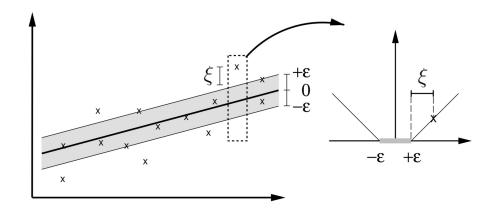


Figura 2.1: MVS con margen blando: se muestra la relación entre las variables de holgura ξ_i^+ y ξ_i^- , asociadas a ejemplos que quedan fuera de la zona ε -insensible, y la función de pérdida $|\xi|_{\varepsilon}$. Imagen extraída de Smola y Schölkopf (2004) [140].

linealmente separables en su propio espacio, el algoritmo realiza una transformación que los proyecta a un espacio de dimensionalidad superior en el que si son linealmente separables. Esta transformación se realiza mediante una función kernel. Tanto el tipo de transformación como los resultados obtenidos por el algoritmo dependen del kernel utilizado, siendo los kernel de tipo gaussiano, polinomial, radial y lineal los más comunes en la práctica. El desarrollo de la formulación dual y los detalles sobre las funciones kernel disponibles queda fuera del alcance de este trabajo. Para una revisión completa de estos temas y del algoritmo MVS en general el lector interesado puede consultar la literatura disponible [140, 19, 28, 152]. En resumen, para resolver problemas de regresión mediante MVS es necesario seleccionar el kernel más adecuado (en el caso de regresión no lineal) y establecer los valores para los parámetros ε y C. Ambos parámetros afectan a la complejidad del modelo. Para seleccionar el valor de parámetro más adecuado se usa normalmente un enfoque de validación cruzada.

2.2.4. Máquinas de vector soporte local

Según lo propuesto por Bottou y Vapnik (1992) [24], cuando los datos se encuentran espacialmente distribuidos de manera no uniforme, es posible mejorar los resultados de la predicción de los algoritmos de aprendizaje mediante la construcción de varios modelos locales en lugar de un solo modelo global. En particular, MVSL propone construir un modelo MVS por cada punto a ser estimado tomando solo aquellos puntos cercanos del conjunto de entrenamiento a fin de evitar influencias ruidosas de puntos más alejados. El subconjunto de punto cercanos es elegido sobre la base de la distancia euclídea entre el punto a ser estimado y los puntos de entrenamiento, aprendiendo el modelo MVS solamente con aquellos puntos del conjunto de entrenamiento que se

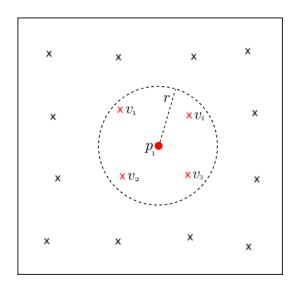


Figura 2.2: Esquema general para el algoritmo MVSL, siendo p_1 el punto a estimar, r_{mvsl} el radio y v_1, v_2, v_3, v_4 sus vecinos.

encuentran dentro de cierto radio definido por el usuario con centro en el punto a estimar [58]. En la figura 2.2 se puede observar el comportamiento general del algoritmo, donde p_1 es el punto a estimar y $\{v_1, v_2, v_3, v_4\}$, que pertenecen al conjunto de entrenamiento, son sus puntos vecinos cuya distancia euclídea resulta menor a un radio r_{mvsl} , los cuales se usan para entrenar el modelo MVS local para p_1 .

En este trabajo, tanto MVS como MVSL, se han utilizado como regresores de tipo ν -regression [133], empleando como kernel una función de tipo radial basis [28, 19]. Cada modelo de regresión ha sido entrenado para establecer la relación entre la posición (X,Y) de un punto y el valor de la variable en ese punto, sin utilizar otra información. El conjunto de parámetros que definen MVS y MVSL son variados en tipo y número y la comprensión total de los mismo requiere de un profundo entendimiento teórico sobre los algoritmos y el problema, por lo que invitamos al lector a seguir la referencias para conocer mayores detalles [133, 28, 19]. Los valores para estos parámetros han sido obtenidos a través de la técnica de validación cruzada de 5 iteraciones, independientemente para el entrenamiento y la evaluación, eligiendo la combinación de valores de los parámetros que logran el menor error de validación. En la tabla 2.3 se pueden consultar los parámetros utilizados, una breve descripción de los mismos, y el rango de valores discretos considerados para cada uno durante el proceso de validación cruzada.

Parámetro	Valores	Descripción
ρ	0.0005, 0.001, 0.01, 0.1, 1, 10, 100, 1000	Representa un criterio de tolerancia para la terminación del algoritmo.
C	1.0	Parámetro de penalización por violar una restricción. Es una constante en el término de regularización en la formulación de Lagrange.
ν	0.01, 0.03, 0.05, 0.1, 0.3, 0.5, 0.75, 0.9	Constante usada para controlar el número máximo de errores permitidos, así como también la cantidad mínima de vectores soporte.
r_{mvsl}	0.1, 0.15, 0.2, 0.25, 0.3, 0.4, 0.5, 0.75, 1	Radio de inclusión de vecinos en MVSL. Representa una proporción que se multiplica por la distancia mayor entre dos puntos del dataset.

Tabla 2.3: Parámetros para los algoritmos MVS y MVSL, ambos usados como regresores de tipo ν -regression con un kernel radial basis functions. La segunda columna indica el rango de valores considerado en la validación cruzada.

2.3. Resultados experimentales

En esta sección se comparan los resultados obtenidos por los algoritmos propuestos mediante las medidas de calidad introducidas en las ecuaciones 2.11, 2.12 y 2.13, como así también se presenta un análisis sobre la capacidad de MVS para modelar las correlaciones espaciales a partir de los datos disponibles de las variables bajo estudio.

Como se adelantó en el apartado 2.2.2, en este trabajo los modelos de variograma generados para kriging han sido del tipo exponencial, esférico, gaussiano y lineal, obteniendo un modelo de interpolación diferente para cada variograma utilizado. Dado que en este trabajo kriging se presenta como un algoritmo competidor para los algoritmos de aprendizaje, solo se reportan los resultados del modelo que genera el menor error absoluto en la interpolación. Por lo tanto, en la experimentación presentada a continuación, se reportan los resultados para cuatro algoritmos: PDI, el mejor resultado obtenido por kriging (de ahora en adelante KRG*), MVS y MVSL.

Para comparar la calidad de los algoritmos presentados en este capítulo, se presentan tres medidas de calidad que evalúan el error en la estimación de los valores interpolados: error absoluto medio (EAM), desviación estándar del error absoluto (DEEA)

y raíz del error cuadrático medio (RECM), calculados como:

$$EAM = \frac{\sum_{i=1}^{N} |y_i - y_i'|}{N}, \qquad (2.11)$$

$$DEEA = \sqrt{\frac{\sum_{i=1}^{N} (y_i' - EAM)^2}{N}}, \qquad (2.12)$$

$$RECM = \sqrt{\frac{\sum_{i=1}^{N} (y_i - y_i')^2}{N}},$$
 (2.13)

donde N es el número de ejemplos del conjunto de prueba, y_i es el valor real de la variable e y_i' es el valor estimado para el ejemplo i de prueba. En este punto cabe destacar que la DEEA es una medida de la variabilidad espacial del error, lo que supone que para valores de DEEA elevados el error cometido en algunos puntos a estimar es muy grande. Atendiendo al objetivo que tienen los mapas de interpolación en la AP, una gran variabilidad en los errores de estimación implica mayor incertidumbre en la toma de decisiones basadas en esta información.

Cada uno de los 25 conjuntos de datos disponibles (c.f. tabla 2.2) fue dividido en dos conjuntos disjuntos a través de un proceso de selección aleatoria, llamados conjunto de entrenamiento y conjunto de prueba. El primero, con el 75 % de los ejemplos tomados al azar de manera uniforme del conjunto de datos original, está reservado para propósitos de generación de los modelos de interpolación, mientras que el restante 25 % de los ejemplos se utiliza para evaluar la calidad de los mismos sobre las medidas de calidad propuestas.

2.3.1. Evaluación de los métodos según las medidas de calidad

En la tabla 2.4 se reportan los valores para la media del error absoluto (EAM), desviación estándar del error absoluto (DEEA) y la raíz del error cuadrático medio (RECM) obtenidos por los algoritmos PDI, KRG*, MVS y MVSL. Los valores resaltados en negrita representan el menor error obtenido por los cuatros algoritmos para esa variable. En esta tabla se puede observar que los algoritmos de aprendizaje, en general, obtuvieron mejor desempeño frente a sus competidores directos (PDI y KRG*). Sobre un total de 25 conjuntos de datos MVSL se destaca como el mejor en un 68 % de los conjuntos de datos según el EAM, 56 % según la DEEA y 64 % según la RECM. Esto se puede ver resumido en la figura 2.3, que muestra la proporción de ocasiones donde cada algoritmo obtuvo la mejor calidad para cada una de las medidas propuestas. Además, agrupando los resultados de los algoritmos de aprendizaje de máquina, i.e. cantidad de conjuntos de datos en donde MVSL y MVS se destacan como los mejores,

15,47 13,63

Tabla 2.4: EAM, DEEA, RECM obtenidos por los algoritmos PDI, KRG*, MVS y MVSL para distintas variables vitícolas (valores menores implican mejora calidad).

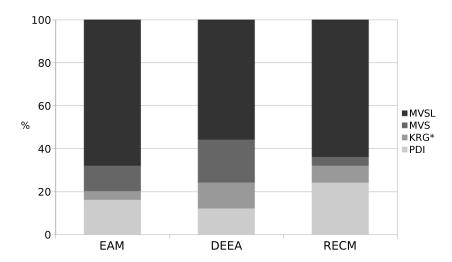


Figura 2.3: Proporción de ocasiones donde cada técnica obtuvo la mejor calidad para cada una de las medidas propuestas (EAM, DEEA y RECM).

estos porcentajes aumentan a 80 %, 76 % y 68 % según el EAM, DEEA y RECM respectivamente. Es importante resaltar que los algoritmos de aprendizaje, en general, también reducen la DEEA y la RECM, situación deseable en problemas de interpolación. La DEEA es una medida de la dispersión de los errores cometidos, por lo que en este trabajo se interpreta como una medida del grado de precisión en la predicción del algoritmo. Valores bajos para la DEEA indican estimaciones más precisas para los datos del conjunto de prueba. De forma similar, la RECM es una medida de desempeño cuantitativa utilizada comúnmente para evaluar estimadores. Dado que esta medida consiste en la raíz cuadrada de la sumatoria de los errores cuadráticos, amplifica y penaliza con mayor fuerza aquellos errores de mayor magnitud. Un valor de RECM menor generalmente, aunque no siempre, implica un mejor estimador. Por último, en la tabla también se hace evidente el caso en el que kriging no converge a una solución (celdas con valor ND, i.e. no disponible), situación que, como se explicó anteriormente, requiere de intervención humana para ajustar manualmente algunos parámetros en el variograma de forma que pueda completar el proceso de interpolación.

A modo ilustrativo, la figura 2.4 muestra los mapas que se obtienen luego de la interpolación para cada uno de los algoritmos propuestos sobre la variable RDP de la finca Teresa Mosso, para el tipo de uva Malbec (RDP_{TM-M}) . Las cruces indican los puntos del conjunto de prueba sobre el que se realizó la evaluación, mientras que las burbujas indican la magnitud del error absoluto entre el valor real y el estimado para cada punto de prueba. Para esta variable la técnica MVSL obtiene los valores más pequeños para cada una de las medidas de calidad propuestas (i.e. EAM, DEEA y RECM), situación que se ve reflejada en las burbujas de la figura 2.4d. Además la figura resalta que tanto kriging como MVS generan mapas de regiones suaves a costa

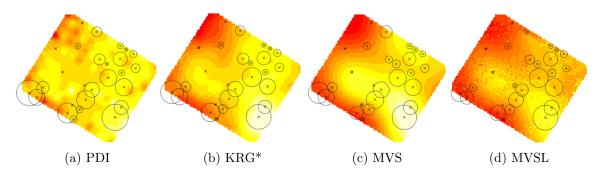


Figura 2.4: Mapas de interpolación, puntos de prueba (cruces) y errores absolutos (burbujas) para la variable RDP perteneciente a la finca Teresa Mosso, tipo de uva Malbec (RDP_{TM-M}) .

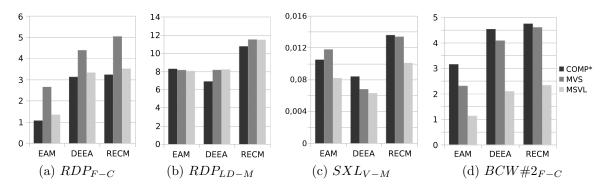


Figura 2.5: Comparativa de las medidas de calidad entre el mejor valor obtenido por PDI o KRG* (etiquetado como COMP*), MVS y MVSL para cuatros variables (valores menores implican mejora calidad).

de una menor calidad de interpolación, mientras que para MVSL se pueden observar regiones más granularizadas, producto de la construcción de múltiples modelos locales (uno por cada punto a estimar) en lugar de un único modelo global.

Por último, en la figura 2.5 se observan comparativas de las medidas de calidad para cuatros variables (RDP_{F-C} , RDP_{LD-M} , $BCW\#2_{F-C}$, SXL_{V-M}). Ya que el objetivo de este trabajo es comparar MVS y MVSL contra sus competidores PDI y kriging, se han resumido los mejores resultados de los dos últimos bajo la etiqueta $COMP^*$ a fin de hacer más fácil el análisis de los resultados. De esta manera, si PDI obtuvo mejor calidad que kriging se toma ese valor para COMP* y viceversa. Las comparativas en la figura 2.5a y 2.5b son casos para los que PDI o KRG* alcanzaron mejores calidades (etiquetado como COMP*), mientras que las figuras 2.5c y 2.5d son casos para los que MVS y MVSL lograron una calidad de predicción superior. Como se puede apreciar en 2.5a y 2.5b los resultados de MVSL se encuentran muy cerca del valor obtenido por COMP*, mientras que las mejoras de MVSL frente a COMP* son aproximadas al 40 % en 2.5c y cercanas al 50 % en 2.5d. Observando la tabla 2.4 se

puede corroborar que en la mayoría de los casos en los que COMP* obtiene mejores resultados, los algoritmos de aprendizaje obtienen resultados muy cercanos.

2.3.2. Evaluación de la capacidad de MVS para modelar correlaciones espaciales en variables vitícolas

En este apartado se presenta un análisis basado en una herramienta de análisis conocida como curvas de aprendizaje, que pone de manifiesto las limitaciones de los métodos de AM para interpolación espacial sobre las variables vitícolas presentadas en este capítulo, limitaciones que pueden aparecer ya sea por la falta de datos de entrenamiento o porque el modelo no se ajusta correctamente a estos datos. Concretamente, el interés principal de este análisis es demostrar que para ciertos conjuntos de datos la disponibilidad de una mayor cantidad de muestras durante el entrenamiento permitiría mejorar los resultados de interpolación espacial logrados por las técnicas de aprendizaje, particularmente para el algoritmo MVSL y asumiendo que también sería el caso para MVS dada la relación que existe entre ambos métodos.

Cuando un algoritmo de aprendizaje no funciona tan bien como se espera, generalmente es porque puede existir un problema de sesqo alto o de varianza alta, i.e. un problema de sub-ajuste o de sobre-ajuste respectivamente [56]. El fenómeno de sobre-ajuste se da cuando el modelo aprendido se ajusta muy bien a los datos de entrenamiento, pero tiene un rendimiento muy pobre para predecir nuevos resultados, i.e. el error de entrenamiento es bajo pero el de generalización es alto. En estos casos se dice que el modelo es muy complejo y sufre de varianza alta. En el otro extremo, el sub-ajuste se da cuando el modelo aprendido es demasiado simple y su rendimiento es pobre tanto en el conjunto de entrenamiento como en el de prueba, i.e. tanto el error de entrenamiento como el de generalización convergen a un valor alto. El modelo en sí está sesgado y se dice que sufre de sesgo alto. Esto indica que cuando un modelo aprendido con técnicas de AM se evalúa solamente sobre la base del error de generalización que comete sobre el conjunto de testeo, es imposible distinguir el caso de sesgo alto del de varianza alta, ya que ambos se expresan de igual manera con errores altos de generalización. Si además se compara el error de entrenamiento podríamos distinguir entre ambos casos ya que en el primero el error de entrenamiento es alto, y en el segundo es bajo. Conocer cuál de ellos ocurre proporciona un indicador de las posibles formas útiles y prometedoras para mejorar el rendimiento del algoritmo. Un algoritmo que aprende un modelo que ajusta correctamente los datos debería lograr un error de generalización bajo, sólo un poco más alto que el error de entrenamiento. En la figura 2.6 se puede observar el comportamiento de ambos errores en función de

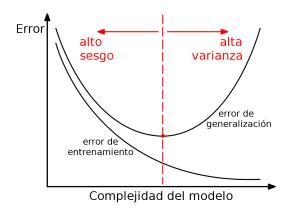


Figura 2.6: Error de entrenamiento y error de generalización en función de la complejidad del modelo (el tamaño del conjunto de entrenamiento se mantiene fijo). La zona destacada a la izquierda corresponde al fenómeno de sesgo alto, la de la derecha a varianza alta, y el punto destacado sobre el error de generalización indica la complejidad óptima del modelo.

la complejidad del modelo, donde se destacan los fenómenos de sesgo alto y varianza alta como así también la complejidad óptima del modelo.

Por otro lado, una curva de aprendizaje es una gráfica que presenta el error de generalización y el error de entrenamiento de un modelo (de cierta complejidad arbitraria) en función del número M de ejemplos de entrenamiento [121]. En general, el proceso consiste en reducir artificialmente el tamaño de M, para volver a entrenar el modelo sobre este subconjunto de datos y calcular los errores de generalización y entrenamiento. Es importante resaltar que el error de entrenamiento se calcula utilizando el subconjunto que se usó para entrenamiento, y no el conjunto de entrenamiento completo. Estas curvas son herramientas que permiten realizar un diagnóstico de los problemas que puede estar sufriendo un algoritmo de aprendizaje. Si el modelo aprendido por el algoritmo tiene sesgo alto, sus errores se comportarán de manera parecida a las curvas de la figura 2.7a. En esta situación, entregarle más datos de entrenamiento al algoritmo no ayudará a disminuir el error de generalización, ya que el problema es la forma subvacente en la que se modelan los datos. Para lidiar con este fenómeno o bien se utilizan modelos más complejos, o se añaden más características al conjunto de entrenamiento, como ser otras variables correlacionadas a la que se está intentado modelar. En caso que el algoritmo presente varianza alta, los errores de entrenamiento y generalización se comportarán de manera similar a las curvas presentadas en la figura 2.7b. Dado que el modelo está sobre-ajustando los datos, el error de generalización se mantendrá alto, aún si se cuenta con un número moderado de ejemplos de entrenamiento. El hecho de existir una gran brecha entre los dos errores es evidencia suficiente para sospechar que existe un problema de varianza alta. En estos casos se

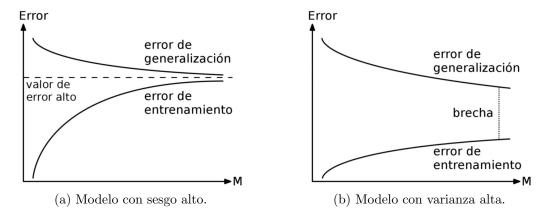


Figura 2.7: Curvas de aprendizaje estilizadas para los fenómenos de sesgo alto y varianza alta. Se grafica el error de generalización y el error de entrenamiento de un modelo (de cierta complejidad arbitraria) en función del número M de ejemplos de entrenamiento.

puede esperar que al añadir más ejemplos de entrenamiento, el error de generalización continúe bajando, y efectivamente sea de ayuda.

En este trabajo, el error de generalización corresponde a la media del error relativo sobre el conjunto de prueba (MER_G) , mientras que el error de entrenamiento corresponde a la media del error relativo sobre el conjunto de entrenamiento (MER_E) . Se optó por utilizar el error relativo para poder comparar entre conjuntos de datos que poseen valores de variable en escalas muy diferentes (c.f tabla 2.2). Concretamente, las curvas para MER_G y MER_E se calculan como:

$$MER_G(N) = \frac{\sum_{i=1}^{N} \left| \frac{y_G^{(i)} - y_G^{\prime(i)}}{y_G^{(i)}} \right|}{N}, \qquad (2.14)$$

$$MER_E(M) = \frac{\sum_{j=1}^{M} \left| \frac{y_E^{(j)} - y_E^{\prime(j)}}{y_E^{(j)}} \right|}{M}, \qquad (2.15)$$

donde N es el número de ejemplos del conjunto de prueba, $y_G^{(i)}$ es el valor real e $y_G^{\prime(i)}$ es el valor estimado para el ejemplo i de prueba, M es el número de ejemplos del conjunto de entrenamiento, $y_E^{(j)}$ es el valor real e $y_E^{\prime(j)}$ es el valor estimado para el ejemplo j de entrenamiento. Por simplicidad de aquí en adelante se omiten los argumentos N y M de sus respectivas funciones de curva.

Para obtener cada punto de la gráfica se generaron subconjuntos de entrenamiento tomando M muestras aleatorias del conjuntos de entrenamiento completo. Sea M_{max} el tamaño del conjunto de entrenamiento completo, se toman valores en el rango $M = 5, 6, ..., M_{max}$. El menor valor corresponde a M = 5 debido a que durante el

entrenamiento de los modelos se realiza un procedimiento de validación cruzada de 5 iteraciones, lo que implica disponer de al menos 5 ejemplos de entrenamiento. Por otro lado, el conjunto de prueba siempre es el mismo ya que se mantiene fijo. A fin de suavizar las curvas de los errores, cada punto de la gráfica se obtiene promediando los resultados sobre 10 iteraciones completas del proceso, i.e. para cada punto de MER_G y MER_E se entrenan 10 modelos, se computan sus correspondientes errores y luego se reporta su promedio. El análisis por curvas de aprendizaje se realiza solamente para el algoritmo MVSL, y dado que éste es una variación de MSV las conclusiones que se desprenden en el análisis son válidas para MVS también. Por otro lado, dada la naturaleza estadística de los métodos PDI y kriging no se aborda su análisis mediante esta herramienta. Además, por razones de brevedad y espacio, no se presentan la totalidad de las gráficas para los 25 conjuntos de datos presentados en la tabla 2.4, sino que se toman sólo aquellos conjuntos que se consideran más interesantes desde el punto de vista de su análisis y conclusiones derivadas.

A continuación se presenta el análisis por curvas de aprendizaje para algunos modelos aprendidos por MVSL sobre la variable RDP y tipo de uva malbec. La sigla RDP hace referencia a la variable riqueza de poda (c.f. tabla 2.1). La riqueza de poda se refiere a la cantidad de nudos o yemas principales que se dejan en una planta, independientemente del tipo de poda que se realice, por lo que está asociada al balance o equilibrio de la planta entre crecimiento vegetativo y producción [3]. El balance de la planta y su riqueza de poda se ven afectadas tanto por factores internos (variedad, edad, estado sanitario) como externos (heladas, granizo, radiación solar, temperaturas, relieve del terreno, suelo, riego, fertilización, poda, y más). En al figura 2.8 se presentan las curvas para cuatro conjuntos de datos de diferentes bodegas, ordenadas de mayor a menor (de izquierda a derecha y de arriba a abajo) según el tamaño del conjunto de entrenamiento completo, i.e. valor máximo de M. Para el caso de las fincas Las Divas y Agrícola Altamira (gráficas superiores), se observa que el comportamiento de sus curvas es bastante parecido: las curvas MER_G muestran tendencias descendentes mientras que las curvas MER_E presentan tendencias crecientes; cuando M es mínimo la brecha entre las curvas es aproximadamente 0,1, mientras que para M > 80 la brecha se reduce a menos de 0,01; y por último, todas las curvas convergen alrededor de un valor de MER cercano a 0,07. Sobre la base de esta evidencia y dado que un error relativo medio de 0,07 puede ser admisible, es razonable afirmar que ambos modelos de interpolación poseen sesgo bajo y varianza baja, y por tanto la interpolación producida es fiable. Examinando el caso de la finca Cobos (gráfica inferior izquierda), se puede ver que la curva MER_G decrece levemente tendiendo a un valor de MER cercano a 0,29, mientras que la curva MER_E crece hasta un valor

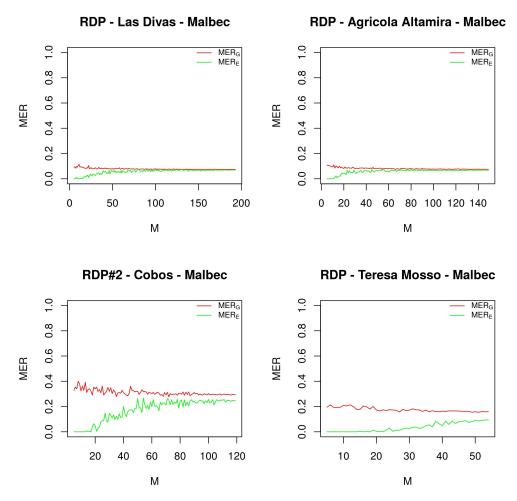


Figura 2.8: Curvas de aprendizaje para los conjuntos de datos de la variable RDP, tipo de uva malbec. En el título de cada gráfica se indica la bodega a la que pertenece.

de MER próximo a 0,24. Si bien ambas curvas comienzan a converger para M>80, la brecha entre ellas es de aproximadamente 0,05. Teniendo en cuenta este análisis es posible diagnosticar que este modelo sufre el problema de sesgo alto, ya que presenta errores de generalización y entrenamiento muy elevados y por lo tanto es sensato esperar que estos resultados mejoren modificando al complejidad del modelo o agregando más características al conjunto de entrenamiento. Finalmente, para la finca Teresa Mosso (gráfica inferior derecha) la curva MER_G muestra una leve tendencia descendente mientras que la curva MER_E presenta una tendencia creciente más marcada. En este caso, para el valor máximo M=54 aún no existe suficiente evidencia para asegurar que ambas curvas convergen a un valor específico. Sin embargo, si se agregan más datos durante el entrenamiento sería razonable esperar que converjan para algún valor aproximado entre 0,16 y 0,1 (valores de MER_G y MER_E en M=54 respectivamente), con la esperanza de que MER_G todavía descienda un poco más y MER_E aumente hasta que la brecha de 0,06 entre ambas curvas sea aún más pequeña. Sobre

la base de este análisis y sustentado además por el valor M de convergencia de los tres casos anteriores, se puede afirmar que este modelo sufre levemente de varianza alta por lo que mayor cantidad de datos durante el entrenamiento ayudará a reducir el error de generalización, aunque en este caso pareciera ser que la mejora esperada no será muy significativa ya que el valor 0.1 de MER_E es una cota inferior para MER_G.

Una situación llamativa del análisis presentado anteriormente es que para ciertos conjuntos de datos MVSL presenta sesgo alto, en otros varianza alta y en otros ninguno de ellos. Esta diferencia en la magnitud de los problemas que sufre cada modelo puede parecer contradictorio considerando que el mismo algoritmo aprende modelos de interpolación para la misma variable. Sin embargo, como se explicó anteriormente, la variable RDP depende del balance de la planta, la cual se ve fuertemente influenciada por factores internos y externos, alterando el comportamiento de sus relaciones espacial. Aún cuando se trata de la misma variable, los resultados de cada modelo depende de la complejidad y resolución espacial en la que se producen estos patrones espaciales para un contexto geográfico particular. Si estos patrones espaciales son muy complejos, además de requerir modelos complejos y más características para disminuir el sesgo alto, es posible que también requieran una resolución de datos adecuada para capturar las relaciones complejas durante el entrenamiento.

A continuación se presenta el análisis por curvas de aprendizaje para otros cuatro casos presentados en la figura 2.9 que corresponden a la variable BCW y RDP para el tipo de uva chardonnay y DDT para el tipo de uva malbec. Las siglas BCW se refieren a la variable peso de racimo, la cual es un indicador directo de la productividad de la planta. Por otro lado, la sigla DDT se refiere al diámetro de tronco, variable correlacionada a la capacidad de la planta para sostener el crecimiento y maduración de los brotes, hojas y frutos. En las cuatro gráficas es posible observar que tanto las curvas de MER_G como las de MER_E no convergen a un valor específico para el valor máximo de M, a la vez que la brecha entre ellas en ese punto es muy grande: 0,11 para BCW#1; 0,18 para BCW#2; 0,12 para RDP; y 0,47 DDT. Con esta evidencia sumado al hecho de que los datos disponibles para entrenamiento son realmente pocos, es razonable diagnosticar que estos modelos de interpolación sufren de varianza alta. Por lo tanto, se espera que aumentando la cantidad de datos de entrenamiento se reduzca el error de generalización, a la vez que las curvas MER_G y MER_E convergen alrededor de un mismo valor de MER.

Finalmente, sobre el total de 25 modelos aprendidos para los conjuntos de datos, el análisis por curvas de aprendizaje arrojó los siguientes diagnósticos: 5 modelos presentan sesgo y varianza baja; 3 modelos sufren exclusivamente de sesgo alto; 7 modelos padecen únicamente varianza alta; los 10 modelos restantes presentan una

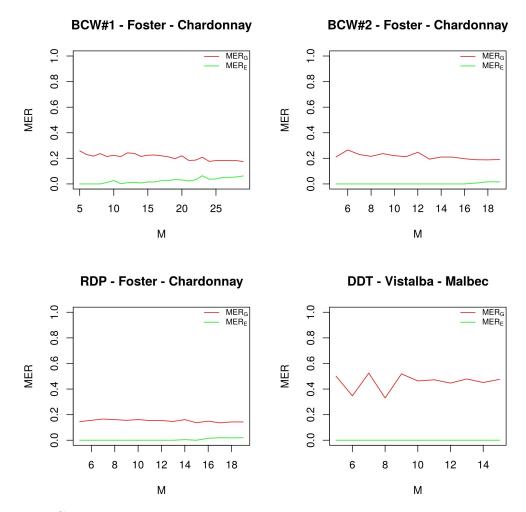


Figura 2.9: Curvas de aprendizaje de los modelos aprendidos sobre las variables BCW y RDP para el tipo de uva chardonnay y DDT para el tipo de uva malbec. En el título de cada gráfica se indica la bodega a la que pertenece.

combinación de sesgo alto con varianza alta. A la luz de los resultados obtenidos es posible verificar que los modelos de interpolación espacial basados en MVSL en ocasiones presentan un límite inherente en la precisión de sus modelos, ya sea por varianza alta, sesgo alto o una combinación de ellos, determinado tanto por la granularidad espacial de las muestras tomadas en campo, como por su capacidad para modelar las correlaciones espaciales que codifican estas muestras.

2.4. Discusión

En esta sección se discuten los alcances y limitaciones de los resultados obtenidos en la evaluación de los cuatro métodos de interpolación espacial para variables vitícolas, interpretando sus resultados en el contexto general de los problemas de interpolación, e incluyendo posibles maneras de mejorarlo y eventuales trabajos futuros.

Comenzando por los métodos de interpolación espacial estadística, la ventaja principal de PDI es que resulta en un método intuitivo y generalmente muy eficaz. En general, este interpolador funciona mejor con puntos uniformemente distribuidos, por lo que PDI es sensible a valores atípicos [11]. Por otra parte, una distribución espacial desigual de los datos también perjudica la calidad de la interpolación obtenida. Por otro lado, kriging tiene la ventaja de que permite incorporar interdependencia variable y además se puede demostrar que es mejor interpolador que PDI [90]. La desventaja principal es que requiere intervención humana y un buen conocimiento experimental de la variable para su correcta modelización, como así también necesita mucho más tiempo de cómputo. Respecto a las técnicas de aprendizaje de máquinas, su principal ventaja es que permiten la generación automatizada de modelos de interpolación a partir de los datos. Además, en la práctica, estos métodos pueden mejorar significativamente los resultados de interpolación mediante la construcción de múltiples modelos locales en lugar de un único modelo global [58]. Como se señaló en el apartado 2.2.4, MVSL propone la utilización de varios modelos MVS locales de interpolación. A la luz de los resultados experimentales presentados en la sección 2.3.1 se puede verificar que esta técnica basada en modelos locales realiza, generalmente, mejores predicciones para problemas de interpolación de variables vitícolas. Este comportamiento es coherente con otros resultados teóricos y experimentales que se pueden encontrar en la literatura sobre diferentes variables [91].

Tal como se explicó en el capítulo 1, debido a la variabilidad espacial inherente de la tierra y del cultivo, uno de los objetivos que persigue la AP es la zonificación del cultivo y su tratamiento particular, ya que la administración individual de pequeñas áreas o zonas con rendimiento similar es altamente preferible frente a la administración homogénea de grandes campos [25]. En este trabajo se prioriza la calidad del modelo aprendido mediante la minimización del error, para la cual MVSL resulta en los mejores modelos. En la práctica es importante incorporar otras medidas de error no locales como ser, por ejemplo, cantidad de zonas generadas y suavizamiento de zonas. Tal como es esperado en algoritmos que no contemplan este tipo de restricción, MVSL podría resultar en modelos muy granularizados, i.e. un número excesivo de zonas, a pesar de minimizar el error.

A partir del análisis por curvas de aprendizaje sobre los conjuntos de datos vitícolas disponibles en este trabajo, se obtuvieron algunas conclusiones importantes acerca de la capacidad del algoritmo MVSL para modelar correlaciones espaciales en variables vitícolas. En la sección 2.3.2 se examinaron varios modelos para determinar si sufrían algún problema de sesgo alto, varianza alta o un poco de cada uno. Para aquellos casos en los que MVSL aprendió modelos de interpolación que presentan evidencias de sesgo

alto, existen en general dos mejoras posibles: (i) aprender modelos más elaborados que respondan a la complejidad de los patrones espaciales subyacentes; y (ii) aumentar la cantidad de características durante el entrenamiento con información de factores internos y externos que afectan al cultivo. En el caso de MVSL la mejora en (i) se puede lograr mediante el uso de funciones kernel específicas para problemas de interpolación espacial, como así también disminuyendo el efecto de la regularización, mientras que la mejora en (ii) requiere la implementación de MVSL como un método de interpolación multivariado, donde se añaden características mutuamente correlacionadas a la variable que se está modelando. Por otro lado, cuando los modelos aprendidos por MVSL muestran evidencias de varianza alta, en general la mejora consiste en aumentar la cantidad de datos durante el proceso de entrenamiento del algoritmo. Esta mejora se encuentra en línea con el argumento presentado en la sección 1.2, donde se explica que las técnicas de interpolación espacial tienen un límite inherente en la precisión de sus modelos, determinado por la granularidad espacial de las muestras tomadas en campo y las correlaciones espaciales que pueden codificar estas muestras [163, 157]. Es posible entonces asociar una baja capacidad de generalización de los métodos de interpolación a la falta de datos durante la generación de los modelos. Sin embargo, conseguir la cantidad de datos suficientes para garantizar una cierta calidad mínima en la interpolación, en la práctica puede ser un problema difícil de resolver. Más cantidad de datos para entrenamiento implica invertir más recursos para la captura de datos en campo, siendo los costos de mano obra una de las limitantes principales [17, 101]. Por lo tanto, aumentar la resolución espacial en la adquisición de datos y reducir el costo de la tecnología de medición tiene un impacto positivo en la calidad de los modelos de interpolación. Es por esto que muchos de los esfuerzos actuales de la AP se dirigen a la investigación y desarrollo de métodos autónomos o semiautónomos para realizar tareas de inspección, medición y evaluación de variables agrícolas de forma más objetiva, rápida y económica. En este contexto, los algoritmos de aprendizaje de máquinas y visión computacional se presentan como alternativas viables para el desarrollo de tales sistemas [58, 150, 138, 50, 130, 13, 37, 154, 172, 87, 129, 131]. Por otro lado, los resultados presentados en los capítulos 3, 4 y 5 también representan esfuerzos dirigidos al desarrollo de herramientas de medición autónoma de variables vitícolas.

Finalmente, algunas líneas posibles para mejorar aún más la calidad de los modelos de interpolación a partir de algoritmos de aprendizaje, consisten en aplicar y adaptar técnicas de modelización univariadas a problemas multivariados, donde se espera que existan fuertes correlaciones, lineales no, entre las diferentes variables del cultivo. De esta manera, la información medida (o estimada) para una variables en un punto, ayuda a estimar los valores de otras variables en ese mismo punto, e incluso otros

puntos vecinos. Por otro lado, en la literatura existen varios trabajos que mejoran el rendimiento del algoritmo MVS como regresor [170, 107, 124], asignando pesos a diferentes puntos de datos de manera que el algoritmo produzca un modelo de acuerdo a la importancia relativa de estos puntos. La idea de utilizar una ponderación que asigna más influencia a ciertos puntos en el conjunto de entrenamiento es análoga a la idea de PDI y kriging de dar más importancia a los puntos conocidos más cercanos al punto que se esta interpolando. De esta manera, MVSL podría ponderar los puntos de entrenamiento de cada modelo local a partir de la distancia al punto de interpolación, resultando en mejores modelos de interpolación. Por último, también podrían elaborarse enfoques híbridos que vinculen las técnicas geoestadsiticas y de aprendizaje de máquinas. Como se explicó en la sección 2.1 un problema con kriging es que asume una media espacial nula. Esta situación podría resolverse aplicando kriging al resultado de restarle a los valores medidos la media espacial, solución que requiere sin embargo conocer esta media espacial, lo que a su vez requiere de un modelo del proceso subyacente. Este modelo podría ser obtenido desde los datos mediante MVS para estimar la media espacial y aplicar finalmente kriging, logrando automatizar el proceso completo.

2.5. Conclusiones

En este capítulo se presenta un estudio para la evaluación de cuatro métodos de interpolación espacial sobre diferentes variables vitícolas medidas en diversos viñedos de la provincia de Mendoza, cuyos datos fueron suministradas por la empresa Xoomcode S.A., especializada en servicios de geolocalización. Entre los métodos de interpolación evaluados se encuentran: PDI v kriging, dos técnicas geoestadística utilizadas frecuentemente en la práctica; MVS, un algoritmo propio de aprendizaje de máquinas, en este estudio empleado como regresor; y MVSL, una variación de MVS para problemas de interpolación espacial. La calidad de cada método se evaluó en base a tres medidas de error: EAM, DEEA y RECM. Sobre un total de 25 conjuntos de datos MVSL se destacó como el de mejor calidad en un 68 % de los casos según el EAM. Además MVSL también reducen la DEEA y la RECM, situación deseable en problemas de interpolación, logrando la mejor calidad en el 56% de los casos según la DEEA y 64%según la RECM. Estos resultados son coherentes con estudios experimentales y teóricos de la literatura en donde se sostiene que las técnicas basadas en modelos locales, en general, logran mejores resultados que las técnicas que producen un único modelo global para problemas de interpolación. Por lo tanto, los resultados experimentales presentados en este capítulo, junto al hecho de que las técnicas de aprendizaje de máquinas permiten la generación automatizada de modelos de interpolación, destacan a los algoritmos MVS y MVSL como alternativas viables para la generación de modelos de interpolación de variables vitícolas en la práctica. También, se presentó un análisis basado en curvas de aprendizaje que pone de manifiesto las limitaciones de los métodos de AM para interpolación espacial sobre diferentes variables vitícolas. Según el análisis realizado sobre los 25 conjuntos de datos disponibles, 7 de estos modelos podrían mejorar directamente su error de generalización aumentando la cantidad de datos, justificando la necesidad de desarrollar mejores instrumentos de medición para variables de interés agrícola. Por último se ofrecieron varias vías de mejoras y trabajos futuros para mitigar estos problemas.

Capítulo 3

Segmentación de Imágenes en Viñedos para la Medición Autónoma de Variables Vitícolas

En este capítulo se presenta un estudio sobre algoritmos de segmentación de imágenes para discriminar la madera de una planta de vid del resto de los elementos de una escena natural de viñedo. Los resultados obtenidos en este capítulo son utilizados más tarde durante el capítulo 5 como una herramienta para la codificación del conocimiento de sentido común necesario para el problema de búsqueda visual activa de yemas de vid. Una construcción importante de este sentido común consiste en reconocer madera del resto del entorno. Sin embargo, en este capítulo se analizan los resultados de estos algoritmos independientemente del enfoque propuesto en el capítulo 5, ejemplificando su uso para otras problemáticas de la medición autónoma de variables vitícolas, diferentes al problema de búsqueda visual activa de yemas de vid.

Particularmente, la elección de segmentar el elemento madera del resto de la escena se encuentra motivada por dos razones: (i) es un componente de la vid que, frente a otros elementos del fondo, se muestra desafiante para los algoritmos de segmentación dada la similitud en sus características visuales particulares; y (ii) el elemento madera incluye varios componentes de la vid (i.e. tronco, brazos, cargadores, sarmientos y brotes) que constituyen la estructura principal sobre la cual crecen y se desarrollan las yemas, hojas, zarcillos y racimos de la planta (c.f. figura 1.1). Contar con información digital de la estructura de la planta tiene variadas aplicaciones en la práctica, por ejemplo: estimación de variables de interés vitícola (como ser peso de madera podada, longitud de entrenudos, longitud de brotes, conteo de yemas, y más) para la predicción de la productividad y calidad; sistema autónomo de poda; reconstrucción 3D de la estructura de la planta; entre otros. Además, en términos de algoritmos de

visión computacional, segmentar la madera de la escena tiene el potencial de mejorar la respuesta de la clasificación, segmentación y detección de otras partes de la vid. Por ejemplo, intentar detectar las yemas de una planta de vid en una escena de campo natural es una tarea muy compleja, sin embargo, dado que las yemas siempre se encuentran sobre los brotes y sarmientos de la vid, mediante un proceso de segmentación de madera robusto es posible restringir la búsqueda de yemas solo a la zonas donde existe madera.

En este capítulo se consideran tres algoritmos de segmentación, los cuales son k-means, mezcla de gaussianas y máquinas de vector soporte, y se introduce un enfoque sencillo para incluir información espacial local que mejora los resultados de los algoritmos originales. Los resultados obtenidos por el mejor algoritmo presentado aquí para la clase madera fueron de 0,96 para precisión (c.f. ecuación 3.2) y 0,681 para sensibilidad (c.f. ecuación 3.3). Estos resultados tienen impacto en el desarrollo de técnicas de medición autónoma para variables de interés vitícola, lo que a su vez contribuye a uno de los desafíos actuales de la agricultura de precisión, i.e. aumentar la resolución espacial de los datos y reducir el costo de la tecnología de medición con el objetivo de mejorar los modelos para la predicción de la productividad y calidad del cultivo. Por lo tanto, los resultados obtenidos en este capítulo son contrastados sobre un caso de uso potencial de estas tecnologías: medición del diámetro de tronco en vides a partir de imágenes.

3.1. Introducción

Como se explicó en el capítulo 1 y 2, uno de los desafíos actuales de la agricultura de precisión (AP) es mejorar la disponibilidad, eficiencia y precisión en la medición de variables agrícola para mejorar así los modelos de predicción de productividad y calidad de los cultivos [25]. Para hacer frente a estos desafíos, en las últimas décadas las ciencias agrícolas se han nutrido de diversas disciplinas, entre las que se destacan la robótica y la inteligencia artificial [122, 20]. Por esto no es extraño encontrar entre las tecnologías aplicadas al agro diversos algoritmos de aprendizaje de máquinas (AM) y visión computacional (VC), como ser: campos aleatorios de Markov; campos aleatorios condicionales; técnicas de agrupamiento; redes neuronales artificiales; extracción de características visuales; detección de bordes; mezcla de modelos; visión estéreo; máquinas de vector soporte; y otros [19, 144, 73]. Estos algoritmos son cada vez más utilizados en la industria agrícola para realizar tareas de inspección, medición y evaluación de forma más objetiva, rápida y económica. En el capítulo 1 se presenta una lista de trabajos del estado del arte que resuelven muchos de estos problemas.

Particularmente, en viticultura, el desarrollo de algoritmos de segmentación para las partes individuales de una planta de vid, como ser yemas, hojas, flores, brotes, frutos y tronco, se presenta como un enfoque viable para el desarrollo de sistemas autónomos de medición en muchas tareas de las prácticas agrícolas, de alta precisión, operacionalmente factible y de bajo costo. En la práctica, un sistema de medición de estas características permitiría obtener mayor cantidad de información en la misma unidad de tiempo, para un gran número de variables, y a un costo equivalente o potencialmente menor en comparación con los métodos (muchas veces manuales) de medición utilizados en la actualidad, permitiendo aumentar de esta manera la resolución espacial de los datos.

3.1.1. Trabajos relacionados

En la actualidad existen trabajos basados en tecnologías de visión computacional y procesamiento de imágenes para la segmentación autónoma de elementos de interés en problemas de la actividad agrícola [154]. A continuación se presentan algunos ejemplos que puede resultar inspiradores sobre el potencial uso de estas tecnologías. En Klodt et al. (2015) [87] se introduce un método de segmentación multiclase para discriminar hojas, tallos, uvas y fondo (i.e. el resto de los elementos de la escena). Este enfoque se basa en tecnologías de visión estéreo a partir de imágenes RGB tomadas en viñedos en condiciones naturales de campo. La eliminación del fondo se logra mediante la reconstrucción de la profundidad, descartando aquellos píxeles que están más lejos en la escena. Una vez eliminado el fondo, la información de color se utiliza, para distinguir entre hojas, tallos y uvas. Los mejores resultados se observaron para las clases fondo (95 % de píxeles clasificados correctamente) y hoja (87 %). La segmentación de la clase tallo muestra la tasa clasificación incorrecta más alta, con el 47 % de píxeles clasificados como hoja. Una desventaja del método es que requiere técnicas de visión estéreo para descartar el fondo de manera robusta. En Aquino et al. (2015) [6] se presenta una técnica para la segmentación de flores de vid en periodo de inflorescencia a partir de imágenes digitales. Este enfoque requiere un procesamiento morfológico de la imagen basada en una descomposición piramidal de la misma, que puede ser utilizado sin alterar la escena natural de campo con fondos artificiales. El algoritmo logra valores de precisión y sensibilidad de 83 % y 85 % respectivamente (c.f. ecuaciones 3.2 y 3.3). En Diago et al. (2012) [49] se introduce un algoritmo de agrupamiento supervisado, basado en la distancia de Mahalanobis, para segmentar el follage y los racimos en plantas de vid, a fin de evaluar el área foliar y el rendimiento de la planta mediante el uso de imágenes RGB. El enfoque procesa automáticamente las imágenes y calcula las áreas (cantidad de píxeles) que corresponden a cada clase, i.e. uvas, madera, fondo y cuatro

clases de la hoja, según su etapa de crecimiento. Los resultados de la segmentación mostraron una exactitud (c.f. ecuación 3.1) de 92 % para hojas y 98 % para racimos. Sin embargo, para cada nuevo grupo de imágenes el enfoque requiere una interacción mínima con el usuario, seleccionando una muestra representativa de píxeles de cada clase para inicializar el algoritmo de agrupamiento.

Por lo tanto, los resultados presentados en estos trabajos constituyen una buena base de evidencia sobre dos aspectos importantes de la AP: (i) muchas tareas de medición pueden automatizarse a través de técnicas de VC sin recurrir a sensores costosos y condiciones de entorno controladas; y (ii) estos trabajos tienen la potencialidad de aumentar efectivamente la resolución espacial de los datos medidos, aumentando la cantidad de muestras tomadas en campo, lo que generalmente está limitado por el tiempo disponible para realizar la campaña de medición. En este contexto, dentro de la comunidad de AP cada vez gana más fuerza la idea de sistemas de medición montados en plataformas móviles autónomas, ya sean terrestres [14, 88, 109] o aéreas [159, 129, 131], que puedan proporcionar un mecanismo efectivo para obtener cantidades masivas de datos, de alta precisión, operacionalmente factible y de bajo costo. En estas plataformas autónomas, la capacidad de cómputo y almacenamiento y el consumo de energía suelen ser recursos escasos. Al respecto, los trabajos del estado del arte nombrados aquí presentan algunas limitaciones importantes. En el caso de Klodt et al. [87], la necesidad de utilizar técnicas de visión estéreo para eliminar el fondo aumenta los requerimientos de cómputo y almacenamiento. De la misma manera, en Aquino et al. [6], el procesamiento morfológico de la imagen basada en la descomposición piramidal de la misma es una técnica que puede necesitar una gran carga de trabajo si las imágenes son de alta resolución, tal como señalan los autores. Por otro lado, en Diago et al. [49] se espera que los requerimientos de cómputo sean bajos, debido a que la segmentación se realiza mediante un algoritmo de agrupamiento. Sin embargo el método requiere interacción con el usuario durante el proceso de segmentación, lo que es una limitación importante si se quiere lograr un sistema de medición totalmente autónomo. Los enfoques propuestos en este capítulo son afines al enfoque propuesto por Diago et al. [49], en el sentido que se utilizan como algoritmos de segmentación supervisados que presentan un elevado costo computacional durante la etapa de entrenamiento del modelo, pero requieren mucha menos capacidad de cómputo durante la etapa de segmentación de las imágenes.

3.2. Elementos de Trabajo y Metodología

En esta sección se introducen brevemente algunas nociones básicas sobre procesamiento y segmentación de imágenes y transformaciones en espacios de color, se presentan detalles sobre las imágenes empleadas durante la experimentación, y se describen las medidas de calidad utilizadas para evaluar la respuesta de los algoritmos de segmentación. Finalmente se dan los detalles de implementación de los algoritmos utilizados: k-means, máquinas de vector soporte y mezcla de gaussianas.

Para todos los enfoques presentados en este capítulo, la segmentación de las imágenes se lleva a cabo sobre la base de un algoritmo de clasificación de píxeles, i.e. el algoritmo predice la clase de cada píxel de una imagen entre una cantidad K de clases. Cada uno de estos algoritmos de clasificación presentan características y parámetros que los hacen muy diferentes entre sí: k-means agrupa datos no etiquetados en K grupos (apartado 3.2.5); a partir de regiones pre-etiquetadas manualmente, MG aprende K distribuciones de probabilidad sobre los píxeles de la escena, cada una representando la probabilidad del píxel de pertenecer a la k-ésima región (apartado 3.2.6); también a partir de regiones pre-etiquetadas, MVS aprende un clasificador que determinar a cuál de las K clases pertenece cada píxel de la escena (apartado 3.2.7). Por lo tanto, a diferencia de k-means, MG y MVS se basan en un enfoque de aprendizaje supervisado para crear un modelo de clasificación de píxeles, por lo que requieren ser entrenados a partir de un conjunto de ejemplos representativos cuya clase es conocida de antemano. Si bien la naturaleza de "algoritmo no supervisado" de k-means puede ser una ventaja en muchas situaciones, en problemas de clasificación de imágenes sufre una limitación similar que la propuesta Diago et al. (2015) [49], i.e. una vez segmentada la imagen todavía es necesario una mínima interacción por parte del usuario para determinar las clases de cada grupo de píxeles.

A continuación se repasan brevemente algunos conceptos propios del procesamiento y segmentación de imágenes, se explica el procedimiento para capturar las imágenes RGB y generar el conjunto de datos de experimentación, se describen los algoritmos de segmentación empleados en la experimentación, y se detalla la metodología de evaluación propuesta.

3.2.1. Segmentación de Imágenes

Segmentar una imagen consiste en particionar la misma en un conjunto de regiones no superpuestas que tienen una semántica coherente con respecto a una aplicación en particular [63]. Idealmente se espera que las regiones representen un objeto o partes del mismo. El proceso de extraer desde el dominio de la imagen una o más regiones

que satisfacen un criterio de uniformidad u homogeneidad se basa principalmente en las características derivadas del espacio de color. Este proceso puede ser aumentado por algún conocimiento adicional sobre los objetos de la escena, tanto como propiedades geométricas y ópticas [99], como también atributos de correlación espaciales con otros objetos [125]. Formalmente, se define segmentación como un método para particionar una imagen I en R_k regiones, con k = 1, ..., K, tal que cada región R_k es un objeto candidato [75]. Una región es un subconjunto de píxeles de una imagen I. Una segmentación es el agrupamiento de píxeles en regiones, tal que:

- Las regiones R_k forman una partición, es decir, abarcan toda la imagen y no se superponen entre sí.
- Cada región R_k satisface un criterio de uniformidad (todos los píxeles de la región tienen una o más propiedades en común).
- Los píxeles pertenecientes a regiones adyacentes, cuando se toman en conjunto, no satisfacen el criterio de uniformidad.

El criterio de uniformidad es el elemento más importante de este modelo, y puede ser tan simple como la uniformidad en la intensidad de los componentes del color, aunque generalmente es más complejo dependiendo de la aplicación.

3.2.2. Espacios de Color

En este problema, cada píxel p_{ij} de una imagen I es un punto en el espacio de 3 dimensiones que comprende las intensidades de los canales rojo, verde y azul (R, G, B) [75, 144]. RGB es usado comúnmente para visualización de color, pero no es un espacio de color adecuado para análisis y segmentación de imágenes debido a la alta correlación que existe entre las intensidades de los tres componentes. Además, es imposible evaluar la similaridad de dos colores desde su distancia euclídea en el espacio RGB, debido a que los valores de sus componentes no representan las diferencias de color en una escala uniforme [38]. Por lo tanto, las imágenes RGB originales son preprocesadas y convertidas mediante una transformación no lineal al espacio de color $L^*u^*v^*$ (L=luminescence, u=saturation, v=hue angle) de la CIE (Commision Internationale de l'Eclairage), escrito generalmente como CIELUV o solamente LUV [75, 144]. Este espacio tiene varias ventajas sobre RGB: separa la información de color (componentes UV) e intensidad (componente L) haciéndolos más independientes (útil en los casos en los que la iluminación varía de píxel a píxel); la escala de las componentes U y V son uniformes, con lo que tiene sentido usar la distancia euclídea como medida de similaridad; es eficiente midiendo pequeñas diferencia entre dos colores; y

por último reduce la dimensionalidad del problema ya que solo se utilizan los componentes UV que, idealmente, representan el color puro sin la influencia de la intensidad o iluminación L [38].

3.2.3. Corpus de imágenes para la experimentación

Las imágenes necesarias para llevar a cabo la experimentación propuesta pertenecen a viñedos con un sistema de conducción de tipo espaldero, pertenecientes a la Facultad de Ciencias Agrarias, Universidad Nacional de Cuyo (Luján de Cuyo, Mendoza, Argentina). Las fotografías fueron realizadas con una cámara compacta NIKON COOLPIX L16, en formato JPEG con una resolución de 2034 × 3072. Sin embargo las imágenes fueron recortadas (no escaladas) a una resolución de 1024 × 1420 para reducir los requerimientos de memoria y procesamiento de los algoritmos. Tres de las cinco imágenes empleadas en este trabajo se pueden apreciar en la figura 3.1a. Las mismas fueron tomadas los primeros días del mes de Mayo de 2012, entre las 14:00 y las 15:00 horas, sin alterar la escena natural y bajo condiciones de clima y luminosidad realistas. Los únicos elementos artificiales agregados en la escena corresponden a la cinta azul que envuelve el tronco de la vid a unos 20 cm de la cruz (lugar en donde se bifurca el tronco) y a la marca azul clavada sobre el tronco a unos 15 cm de la cruz. Estas marcas fueron colocadas con propósitos de calibrar las distancias, aunque estos resultados no han sido considerados en el presente trabajo.

Pese a las diferencias inherentes de cada algoritmo, la necesidad de indicar externamente el parámetro K que define la cantidad de regiones en las que se particiona la imagen es común a todos, y es sobre la base de este parámetro que se realiza la evaluación y comparación de los mismos. La cantidad de regiones se establece en K=3, correspondientes a madera, no-madera, y marcas azules, lo que define la cantidad de grupos en k-means, distribuciones en MG y clases en MVS. Si bien se busca agrupar en dos regiones distintas todos aquellos píxeles que pertenecen a madera de aquellos que pertenecen al no-madera, se agrega una región extra para aislar los píxeles que pertenecen a las marcas azules introducidas en la escena. El total de píxeles disponibles para entrenamiento de los algoritmos es de 5 675 199, extraídos de un conjunto de imágenes de entrenamiento diferentes a las imágenes de prueba de la figura 3.1a, de los cuales 1646961 pertenecen a madera, 3981693 pertenecen al no-madera y 46545 pertenecen a las marcas de referencia. Estos píxeles corresponden a zonas parciales de las distintas imágenes de entrenamiento que fueron demarcadas de forma aproximada en una o más zonas de madera, no-madera y marcas, dejando sin etiquetar aquellos píxeles cercanos a la frontera de diferentes materiales ya que, debido al comportamiento del algoritmo JPEG, los píxeles en las fronteras se presentan de forma difusa. De

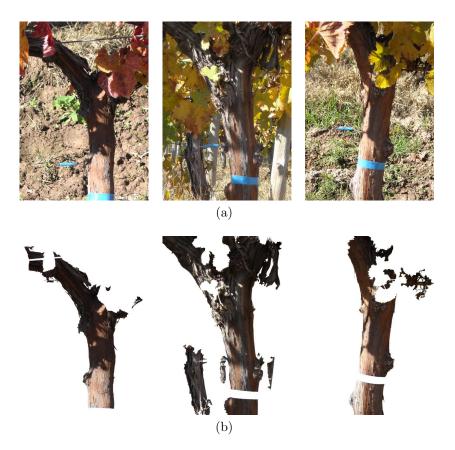


Figura 3.1: (a) Imágenes originales; (b) Imágenes de referencia.

esta manera se descartan píxeles dudosos en el entrenamiento de los algoritmos. Esta sección prosigue con la explicación de los métodos empleados en la evaluación de los algoritmos.

3.2.4. Medidas de calidad para evaluar la segmentación lograda

La evaluación de los algoritmos se realiza mediante métodos de discrepancia que comparan la imagen segmentada con una imagen de referencia, usando la diferencia para valorar el rendimiento [174]. La segmentación se considera un proceso de clasificación de píxeles con lo que se propone como medida de discrepancia diferentes índices sobre el número de píxeles correctamente/erróneamente clasificados. Las imágenes de referencia (figura 3.1b) fueron obtenidas mediante un proceso de segmentación manual realizado por los autores.

En la tabla 3.1 se presentan las medidas de calidad para la comparación de los algoritmos utilizadas típicamente en problemas de clasificación [141, 53]: exactitud (ecuación 3.1), denominado a veces "accuracy" por su término en inglés; precisión (ecuación 3.2), también conocido como "valor predictivo positivo" o "precision" por

Ecuación		Descripción
$exactitud = \frac{vp_k + vn_k}{vp_k + vn_k + fp_k + fn_k}$	(3.1)	Proporción de píxeles correctamente clasificados (positivos y negativos) para una clase, i.e., evalúa la efectividad global del algoritmo para la k-ésima clase.
$precisi\'on = rac{vp_k}{vp_k + fp_k}$	(3.2)	Fracción de píxeles correctamente clasificados como positivos para una clase, i.e., evalúa la capacidad de predicción del algoritmo para la késima clase.
$sensibilidad = \frac{vp_k}{vp_k + fn_k}$	(3.3)	Proporción de píxeles positivos reales que son clasificados correctamente como tal, i.e., evalúa la eficacia del algoritmo para los positivos de la k-ésima clase.
$medida\text{-}F = 2 \times \frac{precisi\'{o}n \times sensibilidad}{precisi\'{o}n + sensibilidad}$	(3.4)	Medida compuesta que re- laciona precisión y sensibili- dad, calculado como la <i>media</i> armónica entre ellos, i.e, la in- versa de la media aritmética.

Tabla 3.1: Ecuaciones y descripción breve de cada medida de calidad implementada.

su término en inglés; sensibilidad (ecuación 3.3), nombrado igualmente como "tasa de verdaderos positivos" o "recall" en inglés; y medida-F (ecuación 3.4), llamada a veces "valor-F" o "f-measure" por su término en inglés. Los términos verdadero positivo (vp), verdadero negativo (vn), falso positivo (fp) y falso negativo (fn) corresponden a contadores que se utilizan para contrastar los resultados de la segmentación con la imagen de referencia. Las expresiones positivos y negativos se refieren a la clasificación producida por el algoritmo, mientras que verdadero y falso se refieren a si esa clasificación concuerda con el valor de la etiqueta (positivos o negativos) del píxel en la imagen de referencia (ver tabla 3.2). Los resultados de la evaluación se analizan en la sección 3.3. De las cinco medidas propuestas se prestará especial atención a precisión y sensibilidad. En el contexto de este problema, un valor de precisión= 1,0 significa que todos los píxeles clasificados como positivos son verdaderamente positivos en la imagen de referencia, pero no dice nada acerca de si todos los píxeles positivos de la imagen de referencia fueron encontrados y etiquetados como tal. Por otro lado, un valor de sensibilidad= 1,0 significa que todo píxel positivo de la imagen de referencia

		Imagen de referencia		
		verdadero	falso	
Imagen	positivo	vp	fp	
segmentada	negativo	fn	vn	

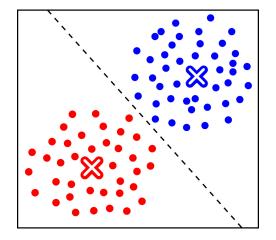
Tabla 3.2: Matriz de confusión.

cia fue clasificado como tal, pero no dice nada acerca de los píxeles incorrectamente clasificados como positivos. Por lo tanto, un algoritmo que clasifique a todos los elementos como parte de una sola clase obtendría un valor de sensibilidad= 1,0, pero valores de precisión extremadamente bajos para esa clase, mientras que un algoritmo que clasifique correctamente uno y solo un elemento de la clase lograría un valor de precisión= 1,0 y un valor de sensibilidad cercano a cero. Esta relación inversa comienza a balancearse a medida que la clasificación se acerca a la clasificación real haciendo que la precisión y la sensibilidad tiendan a 1,0. Es por esto que, generalmente, estas magnitudes no se tratan de forma aislada. En su lugar, o bien los valores para una medida se comparan para un nivel fijo en la otra medida (por ejemplo, precisión a un nivel de sensibilidad= 0,8) o ambos se combinan en una sola medida, tal como la medida-F.

3.2.5. K-means

K-means es un método de agrupamiento iterativo descendiente usado para agrupar datos no etiquetados en un espacio multidimensional (figura 3.2). Suponiendo que se cuenta con un conjunto de datos $\{\mathbf{x}_1,...,\mathbf{x}_N\}$ que consta de N observaciones de una variable aleatoria d-dimensional \mathbf{x} en el espacio Euclídeo, el objetivo es particionar el dataset en algún número K dado de grupos. Se puede pensar en un grupo como un conjunto de puntos cuyas distancias entre ellos son pequeñas comparadas con las distancias a los puntos fuera del grupo. Para formalizar esta idea, en primer lugar se introduce un conjunto de vectores d-dimensionales μ_k , con k=1,...,K, donde μ_k es el vector de valores medios asociado con el grupo k-ésimo y se puede interpretar como el centro de masa del grupo. El objetivo del algoritmo entonces es encontrar una asignación de puntos a los K grupos, así como un conjunto de vectores μ_k , tal que la suma de las distancias o medida de disimilaridad de cada punto al vector μ_k más cercano sea mínima. Dado un conjunto inicial de centros μ_k , el algoritmo avanza en dos pasos:

1. para cada centro μ_k se identifica el subconjunto de puntos (su grupo) que están más cercanos a él que a cualquier otro,



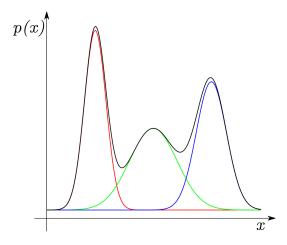


Figura 3.2: Agrupamiento en dos grupos realizado por k-means. La cruces indican el centro de cada grupo.

Figura 3.3: Distribución p(x) modelada como una MG (linea negra) con tres componetes (rojo, verde y azul).

2. se recalculan las medias μ_k de cada grupo.

Estos pasos se repiten iterativamente hasta que el algoritmo converge, i.e., cuando los centros finalmente se mantienen iguales de una iteración a la siguiente. Generalmente se toman K puntos de forma aleatoria como conjunto inicial de centros μ_k . El algoritmo está pensado para situaciones en las que todas las variables son de tipo cuantitativo y se elige como medida de disimilaridad la distancia euclídea al cuadrado,

$$d(\mathbf{x}_{i}, \mathbf{x}_{i'}) = \sum_{j=1}^{P} (\mathbf{x}_{ij} - \mathbf{x}_{i'j})^{2} = \|\mathbf{x}_{i} - \mathbf{x}_{i'}\|^{2}$$
(3.5)

donde \mathbf{x}_{ij} son mediciones para i=1,2,...,N, sobre las variables (o atributos) j=1,2,...,P. Dado que se utiliza la Eq. 3.5 como medida de disimilaridad, k-means no sólo está limitado a variables cuantitativas, sino que además no es robusto a valores atípicos en los datos. El desarrollo completo y los algoritmos para k-means se puede encontrar en [19, 66].

En este punto es necesario aclarar que la decisión de incluir k-means en la experimentación está fundamentada en la utilidad que tiene como línea base para la comparación del resto de los algoritmos presentados, ya que éste no es enfoque particularmente sofisticado para el problema de segmentación de imágenes y en general es extremadamente difícil enfrentarlo con esta técnica, siendo propenso a errores cuando la escena está compuesta por objetos que presentan superficies no homogéneas, superficies texturadas y/u objetos que se ven afectados por cambios de luminosidad o sombras. Estos problemas se hacen evidentes en las imágenes de la figura 3.1a. Las mismas exhiben una variedad de materiales, tonalidades y texturas diversas, perte-

necientes a madera, follage, uva, tierra, malezas, alambre y otros elementos, mientras que la cantidad de grupos K=3 deseados como salida es bastante menor. Sumado a esto, la medida de disimilaridad basada en información del color hace a la imagen extremadamente ruidosa, aún cuando el algoritmo opera en el espacio de color CIELUV sobre los componentes UV.

3.2.6. Mezcla de gausianas

Una MG es un modelo compuesto por una superposición lineal simple de distribuciones gaussianas que tiene por objeto proveer una clase multimodal de modelos de densidad, frente a la clase uni-modal de una única gaussiana (ver figura 3.3). Tal superposición puede ser formulada como un modelo probabilístico definiendo la MG para una variable aleatoria d-dimensional \mathbf{x} como una superposición de C componentes gaussianos de la forma

$$p(\mathbf{x}) = \sum_{c=1}^{C} \pi_c \mathcal{N}(\mathbf{x} | \mu_c, \Sigma_c),$$
 (3.6)

donde cada densidad $\mathcal{N}(\mathbf{x}|\mu_c, \Sigma_c)$ es un componente C del modelo con su propio vector de medias μ_c y matriz de covarianza Σ_c , y el parámetro π_c es el coeficiente de mezcla. Para que la Eq. 3.6 satisfaga los requerimientos de probabilidad, y siendo que cada $\mathcal{N}(\mathbf{x}|\mu_c, \Sigma_c) \geqslant 0$, se debe cumplir que

$$\sum_{c=1}^{C} \pi_c = 1 \tag{3.7}$$

$$0 \leqslant \pi_c \leqslant 1. \tag{3.8}$$

Por lo tanto, la forma de una MG está gobernada por los parámetros π_k , μ_c y Σ_c , con c=1,...,C. Un método elegante y poderoso para establecer los valores de estos parámetros consiste en encontrar la máxima verosimilitud de los datos mediante el algoritmo de esperanza-maximización (EM) [106]. En [19, 66] se explica de forma detallada cómo calcular la máxima verosimilitud junto con el desarrollo del algoritmo EM.

El enfoque de clasificación propuesto aquí consiste en encontrar una MG que modele la distribución de probabilidad de los píxeles de un material en particular sobre los componentes UV del espacio de color CIELUV. Una vez entrenados los modelos se evalúa la probabilidad de cada píxel de la imagen a segmentar en cada distribución de probabilidad correspondiente a cada material, clasificando el píxel con la etiqueta de la distribución que mayor probabilidad entregó. En el proceso de entrenamiento fueron empleados la totalidad los píxeles disponibles, i.e., 5 675 199 divididos en 1 646 961

de madera, 3 981 693 de no-madera y 46 545 de marcas. Por último, el número C de componentes de la MG no es el mismo en cada modelo y es un parámetro provisto externamente al algoritmo: la distribución de píxeles pertenecientes a madera fue entrenada con C=4; la distribución de píxeles que pertenecen a no-madera (conformado por un conjunto de materiales) con C=6 y la distribución de los píxeles que pertenecen a las marcas de referencia con C=1 (lo que es igual a una distribución gaussiana simple). Para establecer el número de componentes C de cada modelo se entrenaron distintas MG para cada material y luego se optó por la que mejor rendimiento obtuvo sobre las imágenes de evaluación. El rango de valores propuesto fue de C=1,...,8 para cada material.

3.2.7. Máquinas de vector soporte para clasificación

La nociones básicas de este algoritmo fueron introducidas anteriormente en la sección 2.2.3, cuando se presentó MVS para problemas de regresión en el contexto de interpolación espacial. Por lo tanto, en este apartado se explican brevemente los principios del algoritmo MVS para clasificación. Nuevamente, para una revisión completa del mismo puede dirigirse a [19, 66, 28, 152]. MVS pertenece a la familia de algoritmos de clasificación que requieren recordar ejemplos de entrenamiento, sin embargo sólo necesita memorizar un conjunto disperso de ejemplos, llamados vectores soporte. La intuición principal de este algoritmo consiste en seleccionar un hiperplano de separación que equidista de los ejemplos más cercanos de cada clase para conseguir lo que se denomina un margen máximo a cada lado del hiperplano. Al definir el hiperplano sólo se consideran los ejemplos de entrenamiento de cada clase que caen justo en la frontera de dichos márgenes, i.e. los vectores soporte. En caso que los datos no sean linealmente separables en su propio espacio, éstos pueden ser proyectados a un espacio de dimensionalidad superior usando una transformación producida por una función kernel. Intuitivamente, una buena separación que maximice el poder de generalización del algoritmo a ejemplos con clasificación desconocida se consigue mediante el hiperplano que logra maximizar el margen, definido como la distancia perpendicular entre los límites de decisión (figura 3.4). Una de sus mayores limitaciones radica en que es un clasificador binario, y el problema de clasificación multiclase es todavía una cuestión abierta, con varios enfoques que lo resuelven parcialmente. En contraste, las ventajas de esta técnica para problemas de clasificación son múltiples: la determinación de los parámetros del modelo es un problema de optimización convexa, por lo que cualquier solución local es también un óptimo global, reduciendo y mejorando considerablemente la calidad de la optimización; la proyección a otro espacio de características permite una representación basada en la función kernel, lo que reduce

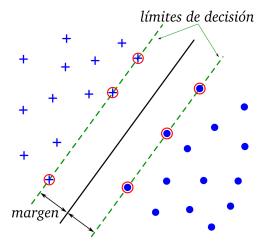


Figura 3.4: Ejemplo de margen y límites de decisión en MVS para un problema de clasificación binaria. Los vectores soporte son aquellos puntos y cruces destacados con círculos rojos.

aún más la cantidad de cómputo necesario; y por último minimiza el sobre ajuste del modelo a los datos de entrenamiento asegurando una minimización del error de generalización, independientemente de la dimensionalidad de los datos.

El conjunto de parámetros que definen al algoritmo son variados en tipo y número y la comprensión total de los mismos requiere de un profundo entendimiento teórico sobre el algoritmo y el problema, por lo que se invita al lector a seguir la referencias para conocer mayores detalles [19, 66, 28]. Dado que MVS es por naturaleza un clasificador binario, para lograr un clasificador multiclase se utilizó el enfoque uno contra el resto [19], cuya implementación fue suministrada por la librería LIBSVM¹ [34]. La función kernel utilizada durante el entrenamiento del algoritmo fue de tipo radial (RBF). Los parámetros involucrados en el algoritmo son tres: γ , la distancia perpendicular entre los límites de decisión; D, la dimensionalidad del conjunto de datos; y C, una constante de penalización por cada punto que viola los límites de decisión. En este trabajo los parámetros fueron establecidos en $\gamma = 1/D$ y C = 1. A diferencia de MG, durante el proceso de entrenamiento del modelo no se empleó la totalidad de píxeles disponibles debido a los altos requerimientos de cómputo y memoria que hubiese demandado tal operación. Por lo tanto se tomó una muestra aleatoria de píxeles de cada material a segmentar, a saber, 10 % de los 1646 961 píxeles de madera, 5 % de los 3 981 693 píxeles de no-madera y 100 % de los 46 545 píxeles de marcas, lo que hace un total de 410325 píxeles disponibles para entrenamiento. Por último, con el objetivo de entregarle más información a MVS y mejorar su clasificación, el dataset de entrenamiento se aumentó con información de los componentes UV de los vecinos

¹https://www.csie.ntu.edu.tw/~cjlin/libsvm/

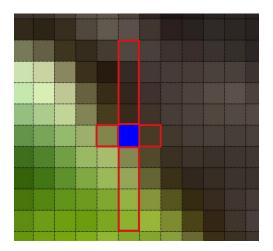


Figura 3.5: Vecindario del píxel p_{ij} . Los recuadros rojos encierran a los vecinos del píxel remarcado en azul.

más próximos de cada píxel. Se define el *vecindario* de un píxel p_{ij} como la región encerrada por una elipse vertical con centro en p_{ij} , semieje mayor a=4 (sobre las abscisas) y semieje menor b=1 (sobre las ordenadas). De esta manera el vecindario de p_{ij} queda conformado por diez vecinos (figura 3.5), lo que aporta veinte nuevas variables al dataset de entrenamiento (cada píxel está definido por sus componentes UV) elevando la dimensionalidad del mismo a D=22. El diseño del aspecto del vecindario no es caprichoso y está fundamentado en las proporciones que mantiene el tronco de la vid en relación al resto de los objetos de la escena.

3.3. Resultados experimentales

En esta sección se presentan los detalles de la experimentación realizada y se comparan los resultados obtenidos por los algoritmos de segmentación propuestos para evaluar su calidad. Además se introduce un enfoque para mejorar los resultados obtenidos tanto por MG como por MVS.

En la primer fila de la tabla 3.3a se puede consultar los resultados obtenidos para madera, donde se ve que si bien obtiene uno de los valores más altos de sensibilidad, el puntaje de precisión es demasiado bajo, lo que perjudica notablemente el valor de la medida-F. Por lo tanto, k-means es bueno clasificando píxeles de madera, pero clasifica incorrectamente demasiados píxeles no-madera como clase madera, produciendo un deterioro del rendimiento. Los resultados que se observan en la tabla 3.3b para no-madera muestran el comportamiento inverso, es decir, valores altos para precisión y bajos para sensibilidad. También es posible contemplar este comportamiento realizando una comparación por inspección visual de los resultados de k-means frente a

	exactitud	precisión	sensibilidad	medida-F
k-means	0,644	0,446	0,814	0,567
MG	0,831	0,666	$0,\!819$	0,727
MVS	0,852	0,718	0,806	0,751
MG- RA	0,898	0,960	0,681	$\boldsymbol{0,791}$
MVS-RA	0,884	0,923	0,661	0,760

(a) Resultados para la región madera

	exactitud	precisión	sensibilidad	medida-F
k-means	0,652	0,909	0,565	0,671
MG	0,830	0,914	0,838	0,873
MVS	0,847	0,904	$0,\!876$	0,888
MG-RA	0,897	0,880	0,986	$\boldsymbol{0,929}$
MVS-RA	0,877	0,864	0,978	0,916

(b) Resultados para la región no-madera

Tabla 3.3: Valores promedio para cada medida de calidad propuesta sobre cinco imágenes segmentadas obtenidos por cada algoritmo. Los valores destacados en negrita corresponden al mejor resultado para cada medida.

la imagen de referencia presentada en la figura 3.6a. En la figura 3.6b se muestran los resultados de la segmentación obtenida con k-means para K=3 grupos. En la figura se puede observar que se han conservado muchos píxeles que corresponden a la clase madera, sin embargo también se han incluido demasiados píxeles no-madera y prácticamente todos los píxeles de la clase marcas (cintas azules en la escena).

El mismo análisis presentado para k-means se puede realizar para MG y MVS. Ambos algoritmos mejoran la calidad global de k-means para la clase madera (tabla 3.3a), obteniendo valores aproximados de sensibilidad pero muy superiores de precisión, lo que eleva el valor de la medida-F. Además es notable el incremento de la exactitud, que es de aproximadamente un $29\,\%$ para MG y de $32\,\%$ para MVS por sobre k-means. Comparando directamente MG y MVS a través de sus valores de medida-F para la clase madera, el segundo es superior por un margen pequeño. Examinando la tabla 3.3b correspondiente a la clase no-madera, es posible observar que MVS vuelve a superar a MG según la medida-F, esta vez por una diferencia más ajustada. Si bien estos resultados muestran que la calidad de la segmentación de MVS es superior a MG, la diferencias entre ambos son pequeñas, lo que los hace prácticamente equivalentes. Las segmentaciones obtenidas por ambos algoritmos se encuentran en las imágenes de la figura 3.6c y 3.6b respectivamente. Es posible observar que, a diferencia de k-means, los píxeles correspondientes a las cintas azules (i.e. clase marcas) han sido correctamente descartado de la clase madera, a la vez que las imágenes muestran una menor cantidad de píxeles no-madera. Esto se corresponde con los resultados de precisión y

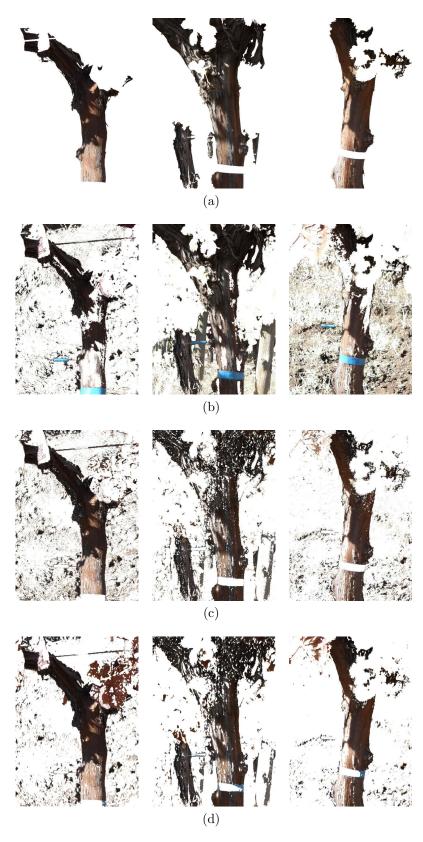


Figura 3.6: Píxeles clasificados como madera. (a) Imágenes de referencia; (b) segmentación realizada con k-means; (c) segmentación obtenida mediante MG; y (d) segmentación lograda con MVS.

3.3.1. Reclasificación por área

Si bien MG y MVS mejoran notablemente los resultados frente a la segmentación realizada por k-means, ambos algoritmos todavía confunden ciertos píxeles de las escena que pertenecen a madera con píxeles que pertenecen a no-madera y viceversa. Sin embargo, en general, este fenómeno se produce en pequeñas zonas de la imagen y de forma diseminada, generando una segmentación "ruidosa" con grupos de píxeles dispersos, como muestran las imágenes de la figura 3.6c y 3.6d sobre la segmentación obtenida para madera. Con este problema en mente se elaboró un mecanismo que realiza una reclasificación de los píxeles segmentados por MG y MVS. Este post-procesamiento, llamado reclasificación por área (RA), mejora los resultados de la segmentación para MG y MVS, denominados MG-RA y MVS-RA, como se puede apreciar por simple inspección visual en las imágenes de la figura 3.7a y 3.7b respectivamente. Este enfoque ha sido originalmente propuesto por [169] usando MG para segmentación de piel humana en imágenes y video a color. El mecanismo consiste en definir un área circular A_{ij} de radio r_{ra} con centro en el píxel p_{ij} a reclasificar y un porcentaje P que determina el umbral de reclasificación. En este trabajo, si la proporción de píxeles madera en A_{ij} con $r_{ra} = 30$ es mayor a P = 0,3, el píxel p_{ij} es reclasificado como madera, si no el píxel es reclasificado como no-madera. Para establecer el valor de r_{ra} se evaluó MG-RA y MVS-RA en el rango $r_{ra} = \{5, 10, 15, 20, 30, 45, 60, 90\},$ siendo $r_{ra} = 30$ el caso en el que mejor rendimiento se obtuvo teniendo en cuenta la medida de calidad de exactitud. Por otro lado, el valor de P se tomó directamente de [169].

El análisis sobre MG-RA y MVS-RA revela mejoras considerables sobre las técnicas anteriores para madera y no-madera. En el caso de madera, si bien los valores de sensibilidad caen apreciablemente, el aumento en precisión es tan importante que la medida-F termina superando a sus competidores anteriores. Para el caso de no-madera, las mejoras son aún más notables, logrando valores de sensibilidad cercanos a 0,98 y valores de precisión aproximados a 0,87, lo que provoca que la medida-F trepe hasta valores próximos a 0,92. Estas mejoras se pueden observar en la figura 3.7a y 3.7b.

Por último, comparando los resultados de la tabla 3.3a con la tabla 3.3b se ve que MG-RA es el algoritmo que mejores resultados de segmentación obtiene para las métricas planteadas en el presente trabajo. Además, en general, MG necesita menos capacidad de cómputo y requerimientos de memoria que MVS, por lo que lo hace preferible frente a éste. Sin embargo MG necesita un procesamiento previo, empírico o analítico, de los datos para definir el número de componentes C de cada material a



Figura 3.7: RA aplicado a MG y MVS con $r_{ra}=30$ y P=0,3. (a) Segmentación obtenida por MG-RA para madera; y (b) segmentación lograda mediante MVS-RA.

segmentar, mientras que MVS no lo requiere.

3.3.2. Complejidad computacional de MG y MVS

En términos de complejidad computacional es posible verificar que MG es más conveniente que MVS. Tal como se comentó en la sección 3.2.6, el entrenamiento de un modelo MG se realiza aplicando de manera iterativa el algoritmo EM (esperanzaminimización), el cual presenta una complejidad computacional $O(idN + iCN^2)$ [128, 66], donde i es la cantidad de iteraciones requeridas por EM para converger, d es la dimensionalidad del conjunto de datos, N es la cantidad de datos de entrenamiento y C es la cantidad de componentes de la MG. Luego, durante la etapa de predicción de clase, calcular la probabilidad para un píxel dado a partir de la distribución modelada

con MG presenta una complejidad computacional O(dC). Por otro lado, cuando MVS se utiliza con una función kernel de tipo RBF es posible esperar que la complejidad computacional durante el entrenamiento sea $O(N^3)$ [23, 1], siendo N la cantidad de datos del conjunto de entrenamiento. Durante la etapa de predicción de clase, la complejidad computacional para clasificar un píxel típicamente será $O(n_{VS}d)$, donde n_{SV} es el número de vectores de soporte y d la dimensionalidad del dataset. Sin embargo existen algunas aproximaciones que reducen la complejidad a $O(d^2)$ [40]. Por lo tanto, MG presenta una complejidad computacional de entrenamiento menor que MVS, sin embargo durante esta etapa MG también requiere un procesamiento previo de los datos por parte del usuario para definir el número de componentes C de cada mezcla de gaussianas, mientras que MVS no lo requiere ninguna intervención. Por otro lado, durante la etapa clasificación de píxeles, MG tendrá menor complejidad computacional que MVS siempre que $C < n_{VS}$, situación que generalmente se dará en la práctica.

Finalmente, dado que el proceso de reclasificación por área se aplica sobre el resultado de cada clasificador, naturalmente MG-RA seguirá siendo más conveniente que MVS-RA tanto en tiempo de entrenamiento de los modelos como durante la etapa de segmentación de nuevas imágenes. Más allá de esto, la complejidad computacional de la reclasificación por área para un píxel es O(A), donde A es la cantidad de píxeles vecinos que pertenecen al área definida por el radio r_{ra} con centro en el píxel a reclasificar.

3.4. Discusión

En esta sección se discuten los alcances y limitaciones de los algoritmos segmentación de madera en escenas naturales de viñedos respecto a su utilidad para codificar el conocimiento de sentido común dentro de un sistema de detección de yemas de plantas de vid, como así también su impacto en el desarrollo de técnicas de medición autónoma para variables de interés vitícola.

En primer lugar, los resultados de segmentación automática obtenidos para los diferentes algoritmos, muestran que k-means no es un enfoque robusto para alcanzar resultados de segmentación útiles. Además, k-means requiere una mínima interacción por parte del usuario para determinar las clases de cada grupo de píxeles una vez realiza la clasificación. Por otro lado, tanto MG como MVS obtienen resultados de segmentación equivalentes y superiores a k-means. Respecto al mecanismo de reclasificación por área, MG-RA y MVS-RA presentan mejoras considerables sobre las técnicas MG y MVS teniendo en cuenta las medidas de calidad tanto para madera

como para no-madera, siendo MG-RA el algoritmo que mejores resultados de segmentación obtiene para las métricas planteadas. Por lo tanto, predecir la clase de un píxel teniendo en cuenta información de píxeles en un vecindario cercano mejora notablemente los resultados de segmentación, haciendo evidente que la información espacial local influye de manera positiva en la respuesta de los algoritmos de segmentación de imágenes. Esta situación se encuentra en línea con varios trabajo de la literatura donde se muestra que incluir información espacial del contexto local en algoritmos de clasificación puede aumentar la calidad de la segmentación producida [36, 165, 125]. Además, es importante notar que la reclasificación por área ofrece un mecanismo eficaz para modificar los valores de precisión y sensibilidad para las clases bajo estudio, lo que permitiría ajustar estas medidas a los requerimientos del problema que se está intentando resolver, donde algunos pueden demandar alta precisión y alta sensibilidad, pero también podrían resignar la calidad de alguna de estas medidas. Por ejemplo, si el problema requiere medir el área de la imagen que corresponde a la clase madera estimada a partir de la cantidad de píxeles clasificados como tal, el algoritmo de clasificación debería lograr valores de precisión y sensibilidad cercanos a 1, va que de otra manera la estimación no sería fiable. Otro problema interesante, por ejemplo, es aquel donde se requiere información de las zonas donde existe madera a fin de restringir la búsqueda de elementos que se encuentran exclusivamente sobre ella o en sus cercanías. En este caso se podría renunciar a valores altos de precisión, i.e. segmentar incorrectamente zonas de la imagen que no pertenecen a madera, pero se esperaría que la sensibilidad se mantenga lo más elevada posible, i.e. sin descartar zonas que son madera, a fin de realizar una detección fiable sobre las zonas segmentadas como madera.

Un algoritmo de segmentación de madera con alta sensibilidad, tal como fue presentado en el último ejemplo al final del párrafo anterior, tiene la capacidad para ayudar a mejorar la eficiencia de un algoritmo de detección de yemas de vid, ya que ofrece información útil respecto a las potenciales ubicaciones de las yemas y nudos de la planta, dado el hecho de que las yemas siempre se desarrollan en un nudo, sobre los brotes y sarmientos de la vid. De esta manera, un algoritmo de segmentación de madera puede ser útil para codificar el conocimiento de sentido común en un sistema de detección de yemas de plantas de vid. Este conocimiento permite establecer propiedades y relaciones entre los elementos de una escena que hace más eficiente la tarea de encontrar el objeto deseado (c.f. sección 1.2). Si el algoritmo de detección de yemas fija su espacio de búsqueda exclusivamente sobre los resultados de segmentación de madera producidos por los algoritmos MG-RA y MVS-RA, es lógico esperar que la detección no sea fiable ya que presentan valores de sensibilidad bajos, mientras que para MG y MVS

se esperaría que la detección efectivamente sea más fiable. Por otro lado, una particularidad notable de MG-RA y MVS-RA frente a MG y MVS, es que la reclasificación produce imágenes más "limpias" respecto a la clase no-madera y marcas, a la vez que rellena mucho de las "lagunas" que se dan sobre la clase madera, tal como se puede observar en los resultados de segmentación presentados en las figuras 3.6 y 3.7. Esta propiedad se logra a costas de una sensibilidad menor, i.e. clasificar incorrectamente píxeles de la clase madera en otro clase, situación que sucede principalmente sobre los bordes del tronco. Sin embargo, existe una clase de algoritmos de detección donde esta propiedad es una particularidad que puede ser explotada con cierta independencia del valor de sensibilidad que obtenga el clasificador para la clase madera. Estos algoritmos son conocidos como detección de tipo escaneo-ventana. El enfoque de estos algoritmos es bastante simple: una ventana de tamaño fijo se desplaza con cierta superposición a lo largo de toda la imagen (i.e. dos ventanas consecutivas pueden compartir píxeles de la imagen), clasificando las sub-imágenes de cada ventana según contenga o no el objeto que se quiere detectar (en el capítulo 4 se dan más detalles sobre estos algoritmos). En este sentido, un algoritmo de tipo escaneo-ventana para yemas, podría descartar todas aquellas ventanas en donde la cantidad de píxeles de madera es cero o menor a cierto umbral. Siguiendo este enfoque, una clasificación "ruidosa", como la que producen los algoritmos MG y MVS, podría resultar en una gran cantidad de ventanas a escanear. Sin embargo, utilizando las segmentaciones "limpias" como las producidas por MG-RA y MVS-RA es posible evitar operaciones innecesarias de extracción y clasificación de ventanas en aquellas zonas donde no existen píxeles de la clase madera. Además, dado que MG-RA y MVS-RA comenten errores principalmente en los bordes del tronco, un algoritmo de escaneo-ventana puede compensar estos errores ampliando los límites de extracción de ventanas, permitiendo escanear sobre los bordes de las regiones segmentadas como madera con cierta cantidad de píxeles no-madera. Vale la pena resaltar que un enfoque de detección de escaneo-ventana como el ejemplificado anteriormente implica tomar decisiones específicas del problema durante el diseño e implementación de los algoritmos, a fin de encontrar el equilibrio adecuado entre todos los factores involucrados en el proceso, como ser: tamaño del área de reclasificación, umbral de reclasificación, tamaño de ventana, desplazamiento del escaneo, umbral para descartar una ventana, y más.

Una ventaja de los algoritmos de clasificación MG y MVS es que permiten obtener la probabilidad de clase de un píxel. De esta manera es posible alterar los umbrales de clasificación para cada clase modificando la respuesta del algoritmo en términos de precisión y sensibilidad, según lo requiera el problema tal como se explicó anteriormente. Este análisis por umbral de clasificación se realiza generalmente a través de

una herramienta conocida como curvas de precisión-sensibilidad [46]. La capacidad de clasificación probabilista de los algoritmos MG y MVS también se puede trasladar fácilmente a los algoritmos MG-RA y MVS-RA, alterando la probabilidad de clase de un píxel según la probabilidad de los píxeles cercanos. Por otro lado, un algoritmo de clasificación probabilista abre la posibilidad de realizar diferentes tipos de posprocesamiento a fin de conseguir segmentaciones más robustas que las logradas para el caso de una clasificación categórica, tal como se demuestra en varios trabajos del estado del arte [62, 158, 54].

3.5. Aplicación: Medición de Diámetro de Tronco

Si bien los algoritmos de clasificación introducidos en este capítulo fueron presentados en el contexto de un sistema de detección de yemas de vid, sus resultados también son contrastados para un caso de uso potencial de estas tecnologías. De esta manera, como aplicación directa de los resultados obtenidos se presenta la medición del diámetro de tronco. El diámetro de tronco es una variable de interés agrícola en la producción de árboles frutales en general, ya que el tronco cumple con la función vital de transportar agua y nutrientes desde las raíces hasta las áreas de consumo (i.e. brotes, hojas y frutos), así como también acumular las sustancias de reserva de la planta [69, 100]. Además, el diámetro del tronco está correlacionado con la capacidad de la planta de sostener el crecimiento y maduración de los brotes, hojas y frutos, y es un buen estimador de otros indicadores del potencial productivo de la planta, como ser biomasa y área foliar [33, 112]. Existen diversas relaciones alométricas reportadas en la literatura para estimar el área foliar en diferentes árboles usando el diámetro o área de una sección transversal del tronco [32, 113].

En la práctica, el diámetro del tronco generalmente es medido manualmente usando calibres de Vernier o cintas métricas. Aunque estos instrumentos pueden ser considerados de alta precisión (0,05 mm para un calibre y 0,5 mm para una cinta), son también muy propensos a errores en el proceso de medición. Existen errores humanos que pueden ocurrir durante las extensas campañas de medición en campo y en el procesamiento posterior de los datos obtenidos. Estos errores pueden ser causados principalmente por la imperfección de los sentidos del operador, potencialmente alterados por las condiciones del medio ambiente, apuro, fatiga o simplemente descuido (errores del I al IV en la tabla 3.4). Además, de acuerdo a conversaciones privadas con ingenieros agrónomos especializados en viticultura, en la práctica generalmente se trabaja con una escala de 5 mm, logrando una precisión de $\pm 2,5$ mm, aplicando redondeo a un decimal. La razón de reducción en la precisión se justifica por la necesidad de

Tipo	Descripción del Error
I	Lectura del instrumento de medición
II	Transcripción de valores medidos a planillas
III	Registro de valores para plantas equivocadas
IV	Transcripción de planillas a la computadora
V	Errores de los métodos de interpolación para los valores obtenidos

Tabla 3.4: Errores que pueden ocurrir durante todo el proceso de medición manual. Los errores de tipo I a IV pueden ser ocasionados por operarios humanos, mientras que el error de V es propio del método de procesamiento de datos empleado.

reducir los tiempos de la tarea y maximizar la cantidad de mediciones efectivas. Por otro lado, es usual que las campañas de medición se diseñen mediante un muestreo estadístico. Si bien es posible disminuir la cantidad de mediciones gracias al uso de métodos de interpolación (consultar capítulo 2), tales como kriging [42], en la práctica, todavía puede ser prohibitivo realizar el número mínimo de medidas requerido por estos métodos para obtener estimaciones precisas, dado los costos y tiempos asociados al proceso de medición manual. Este tipo de errores, cometidos por falta de datos durante el proceso de interpolación espacial, son referidos como error de interpolación, y se lista en la tabla 3.4 junto a los demás errores.

Es posible encontrar en la literatura diversos trabajos que intentan mitigar los errores cometidos por operadores humanos durante el proceso de medición del diámetro de tronco (errores del I al IV en la tabla 3.4). La lista de trabajos presentada a continuación sirve para la ilustrar los avances logrados en esta dirección, sin embargo no pretende ser exhaustiva. Jutila et al. (2007) [78] presenta un método diseñado para cosechadoras forestales, que consiste en un sensor láser 2D montado en un vehículo todo terreno. Logra una buena precisión en bosques de pinos, con un error medio de 6 mm y desviación estándar de 22 mm. Una de las limitaciones para la adopción de esta técnica en la producción de árboles frutales es el alto costo de los equipos necesarios para su implementación. En Omasa et al. (2008) [116] emplean la tecnología de escaneo LIDAR (acrónimo del inglés Light Detection and Ranging) en combinación con datos procedentes de sensores aerotransportados y datos obtenidos con sensores portátiles a nivel del suelo. Este sistema logra errores menores a 3 mm. Sin embargo, el elevado costo de los sensores LIDAR aéreos y portátiles hace que sea inalcanzable para la mayoría de los productores de árboles frutales. Kan et al. (2008) [80] presenta un método basado en algoritmos de visión computacional para medir diámetros de tronco y ramas a partir de imágenes tomadas con una cámara digital convencional. Un tablero de calibración se incluye en la escena junto al tronco a fotografiar. Una vez segmentado el tronco y el tablero, las medidas reales en milímetros se obtienen

multiplicando el diámetro en píxeles del tronco por el tamaño en milímetros de un píxel, obtenido gracias al tablero de calibración. El error medio reportado es de 6,7 mm y la desviación estándar de 17,3 mm. Sin embargo, cuando el fondo de la escena es complejo (e.g. otros árboles en el fondo), el tablero de calibración o el tronco puede no ser detectado y el método falla para obtener los diámetros. Los resultados reportados muestran una tasa de éxito del 90 % en la detección del tablero y tronco. A diferencia de los trabajos anteriores, este método es accesible para cualquier productor de árboles frutales.

Las técnicas descritas anteriormente contribuyen de manera significativa en el proceso de medición del diámetro de tronco. Estos métodos presentan diferentes ventajas para el manejo del cultivo: (i) son susceptibles de ser operados por personal no especializado, con menores costos operativos; (ii) reduce los niveles de fatiga evitando el deterioro de la calidad de las mediciones en el tiempo; y (iii) permiten desarrollar sistemas de medición autónomos, a fin de aumentar la cantidad de mediciones por unidad de tiempo. Sin embargo, teniendo en cuenta la relación costo/precisión, estas soluciones se encuentran desequilibradas, ya que algunas son de bajo costo pero de baja precisión, y otros son de alta precisión pero de alto costo, dejando la medición manual mediante calibres o cintas como la única opción para la mayoría de las situaciones prácticas.

A continuación se presentan dos aplicaciones derivadas del trabajo desarrollado en este capítulo, con la intención de mostrar el potencial de estos algoritmos como herramientas de medición de bajo costo y alta precisión.

3.5.1. Medición del Diámetro en Píxeles

En este apartado se presentan los resultados obtenidos para estimar el diámetro de tronco en píxeles mediante el algoritmo MG-RA. Este método aún requiere una calibración manual para estimar el diámetro en milímetros, dejando el problema de calibrado automático para el siguiente apartado. Estos resultados se presentan como medidas del diámetro en diferentes posiciones horizontales (filas de la imagen), correspondientes a las diferentes posiciones donde puede medirse el diámetro.

La figura 3.8a corresponde la imagen segmentada por MG-RA, y requiere de una breve explicación: las regiones de color rojo representan píxeles correctamente clasificados como madera (verdaderos positivos); las de color blanco píxeles clasificados correctamente como no-madera (verdaderos negativos); las regiones de color azul representan píxeles incorrectamente clasificados como madera (falsos positivos); y finalmente las regiones de color verde representan píxeles que pertenecen a madera y no han sido clasificados como tal (falso negativos). Por otro lado, las flechas grises en la

figura 3.8a señalan la zona del tronco totalmente vertical que comienza aproximadamente a 5 cm de la cruz del tronco y finaliza a 20 cm de la misma, y que aquí se nombra región de medición. Dado que en cada fila de la imagen los errores cometidos son de dos tipos (falsos positivos y falsos negativos), los resultados se reportan en términos de medida-F, precisión y sensibilidad. Analizando las posiciones horizontales dentro de la región de medición, en la figura 3.8c se nota que la mayoría de los valores de precisión y sensibilidad son mayores a 0,9, mientras que en la figura 3.8b la medida-F alcanza valores mayores a 0,95. Estos valores representan un error menor a 3,4 mm (15 píxeles en esta imagen) en el diámetro del tronco con respecto al diámetro calculado desde la imagen segmentada manualmente. Comparado con la precisión que generalmente se emplea en la práctica cuando la medición es manual, se ve una diferencia de 0,9 mm. Si bien estos resultados parecen prometedores, errores mayores a 3 mm aún pueden ser graves.

Una de las ventajas de este enfoque es la simpleza del mismo. Una vez entrenados el algoritmo de segmentación, lo único que se requiere es fotografiar los troncos y procesar las imágenes. Sin embargo, el enfoque no pretende ser una solución factible para la medición del diámetro de tronco, sino que intenta mostrar la efectividad del método de segmentación en imágenes en un entorno natural, simplificando muchos de los aspectos que debería resolver en la práctica un método de medición verdadero. Por lo tanto, este enfoque aún tiene varias limitaciones importantes. En primer lugar, el tronco que se va a medir tiene que ser claramente discriminado de otros troncos que

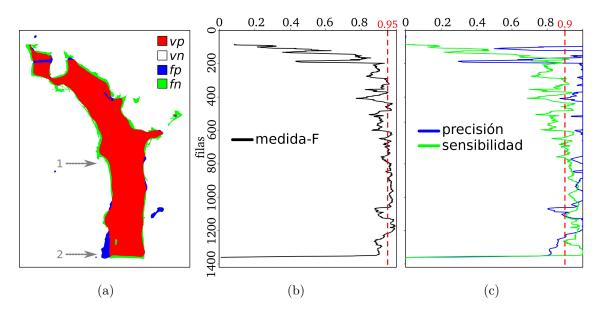


Figura 3.8: Valor de medida-F (b), precisión y sensibilidad (c) para cada fila de píxeles de la imagen segmentada por MG-RA (a).

pueden aparecer en la escena (ver la imágen del centro de la figura 3.1a). En segundo lugar, se debe detectar la dirección de la medición, ya que aquí se asume que el tronco se encuentra totalmente vertical. En tercer lugar, se debe detectar la posición preferida de medición a lo largo de todo el tronco segmentado. En cuarto lugar, es necesario medir la distancia en píxeles entre los bordes del tronco en la dirección de la medición determinada, y realizar un proceso de calibrado de la medición de píxeles a milímetros. En el próximo apartado se presenta un enfoque que hace frente a estas limitaciones.

3.5.2. Medición por Imágenes con Calibración Automática

El enfoque presentado aquí hace uso del algoritmo MG como técnica de segmentación (descrita en el apartado 3.2.6) en vez del algoritmo MG-RA (apartado 3.3.1), pero además incorpora a la escena un artefacto para manejar todas las dificultades nombradas en el apartado anterior: cada tronco a ser fotografiado es sujetado previamente con una pinza, como se puede apreciar en la figura 3.9a y 3.9b. Es importante resaltar que el uso de la pinza en este trabajo está justificado por su simplicidad operacional en la etapa de experimentación, así como también por la disponibilidad universal del producto, por lo que es posible reproducir estos resultados sin mayores dificultades. Más allá de su simplicidad operacional, en la práctica el uso de la pinza todavía requiere mucha intervención de un operario humano, similar al requerido por un calibre de Vernier o una cinta métrica. Sin embargo, en la práctica la pinza podría ser fácilmente reemplazada por otros mecanismos más efectivos y baratos. Por ejemplo, un tablero de calibración colocado justo por detrás del tronco al momento de capturar la imagen. Sin embargo, el diseño de estas alternativas es un problema técnico que está fuera del alcance de este trabajo.

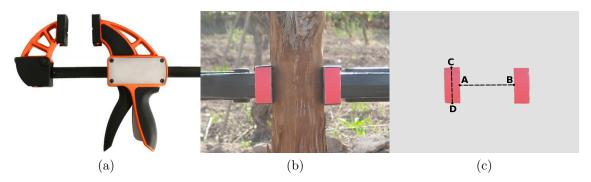


Figura 3.9: De izquierda a derecha: (a) pinza; (b) imagen original con la pinza modificada sujetando al tronco; (c) segmentación de las marcas mediante el algoritmo MG para la imagen original y estimación en píxeles del diámetro de tronco (distancia entre los puntos A-B) y altura de las almohadillas (distancia entre los puntos C-D).

Implementación del Método por Imágenes con Calibración Automática

En este trabajo, el objetivo de la segmentación es discriminar los píxeles que pertenecen a las almohadillas de la pinza, y no los píxeles del tronco. El problema de segmentación se simplifica modificando el color de las almohadillas a rojo, un color muy poco probable en escenas de viñedos. Con esta modificación es posible realizar una segmentación muy poco ruidosa y muy precisa de las almohadillas de la pinza, y es por esta razón que ya no es necesario utilizar el algoritmo MG-RA para lidiar con este problema. Durante la etapa de entrenamiento del algoritmo MG se emplea un conjunto de imágenes de entrenamiento segmentadas manualmente. En este trabajo se utilizaron imágenes capturadas viñedos naturales, sin alterar las condiciones de luminosidad, tal como se ejemplifica en la figura 3.10. A partir de este conjunto de entrenamiento se estiman dos distribuciones de probabilidad (consultar el apartado 3.2.6): una distribución para las almohadillas y otra distribución para el no-madera. Estas distribuciones modelan la probabilidad de que un píxel pertenezca o no a la clase. Una vez obtenidas ambas distribuciones, una nueva imagen extraída del conjunto de prueba (ver figura 3.11) puede ser procesada para clasificar sus píxeles en alguna de las dos clases. La decisión de clasificación se toma según la probabilidad de cada píxel en cada distribución: el píxel se asigna a la clase con mayor probabilidad.

Una vez que se han segmentado las almohadillas y estimado los puntos A, B, C y D, tal como se ilustra en la figura 3.9c, todas las dificultades del enfoque anterior



Figura 3.10: Ejemplo de imágenes de entrenamiento para obtener las distribuciones de probabilidad para las clases *almohadilla* y no-madera. Las imágenes fueron tomadas en condiciones naturales de campo, e incluyen fotos tomadas tanto en días soleados como días nublados.



Figura 3.11: Ejemplo de imágenes de prueba que fueron procesadas para segmentar las clases *almohadilla* y no-madera. Las imágenes fueron tomadas en condiciones naturales de campo, e incluye fotos tomadas tanto en días soleados como días nublados. La cuerda que rodea el tronco indica la posición en la que deben colocarse tanto la pinza como el calibre en cada tronco.

son fácilmente manejadas: el tronco a medir se encuentra exactamente entre las dos almohadillas segmentadas; la dirección de medición es perpendicular al segmento C-D; la posición de la pinza en el tronco determina la posición preferida de medición; el diámetro del tronco es la distancia en píxeles entre los bordes internos de las almohadillas segmentadas (puntos A-B de la figura 3.9c); y finalmente la calibración de píxeles a milímetros es realizada utilizando el conocimiento previo de la altura de las almohadillas en milímetros y su medida en píxeles de la segmentación actual (puntos C-D de la figura 3.9c). Vale la pena aclarar que en la práctica todavía pueden ocurrir errores de segmentación en los bordes de las almohadillas, por lo tanto se toman varios puntos A_n - B_n y C_m - D_m a lo largo de los bordes y se promedian su distancias.

Evaluación del Enfoque

En este apartado se describe el procedimiento empleado para evaluar la validez y robustez del método por *imágenes con calibración automática (ICA)*. Concretamente, se llevaron a cabo dos experimentos para evaluar este enfoque.

Errores de operador. El primer experimento tuvo por objetivo contrastar los resultados del método ICA con los resultados obtenidos por el método manual basado en calibre (MBC), usualmente empleado en la práctica. El protocolo del método MBC es el siguiente: (i) se obtiene la magnitud del diámetro del tronco manipulando el cali-

bre; (ii) se registra la medida en una planilla de papel; (iii) al finalizar la campaña de medición se cargan todas las medidas en planillas electrónicas para su procesamiento posterior. Este experimento se diseñó para reproducir las condiciones tal y como se presentan en la práctica, i.e. en campo, sin alteraciones de la escena y con luminosidad natural (considerando días soleados y nublados). El mismo se llevó a cabo sobre un conjunto de 30 plantas de vid, en viñedos reales con sistema de conducción espaldero, las cuales fueron fotografiadas en condiciones de luminosidad natural (ver figura 3.11), mientras que las mediciones manuales se realizaron con un calibre estándar, con precisión de 0,5mm. Cuatro operadores fueron elegidos y se les instruyó para ser extremadamente cuidadosos al realizar las mediciones, especialmente para el método MBC, a fin de minimizar la posibilidad de errores humanos. La posición de medición siempre fue la misma en cada tronco, tanto para la pinza como para el calibre (ver figura 3.11). Cada operador realizó siete rondas de medición para cada método sobre las 30 plantas. La comparación entre ambos métodos se realizó en base a sus errores de medición. Para calcular estos errores se requiere un valor referencia del diámetro de cada tronco. Para obtener este valor se decidió utilizar un método no estándar, pero muy preciso, que en este trabajo se denominó método manual basado en imágenes (MBI). Este método coincide exactamente con el método ICA, con la única diferencia que la segmentación de las almohadillas en cada imagen se realiza por un operario de forma manual, mediante el uso de un software de edición de imágenes, y con extremo cuidado y precisión. De esta manera, los errores en la medición del diámetro real se remiten a errores de redondeo ocasionados por la resolución de la cámara. En este trabajo, un píxel corresponde aproximadamente a 0, 1 mm, equivalente a los 0, 05 mm de precisión del calibre de Vernier. Para el caso del método ICA, este resulta en un error muy conveniente que mide exactamente el error de la segmentación obtenida. En el caso del método MBC, este error pone de manifiesto los posibles errores manuales de tipo I, II y IV (cf. tabla 3.4) cometidos por un operador.

Condiciones de luminosidad. El segundo experimento tuvo por objetivo verificar la robustez del método ICA ante los inevitables cambios naturales de luminosidad. La luz recibida por las plantas se ve fuertemente afectada por las condiciones climáticas, i.e. cielos nublados o despejados, que pueden cambiar incluso en el mismo día que se está realizando la campaña de medición. Para evaluar la robustez del enfoque frente a estos cambios se manipularon los conjuntos de entrenamiento y prueba agrupando imágenes según las condiciones de luminosidad, i.e. soleado o nublado. De esta manera, el experimento consistió en entrenar el algoritmo MG utilizando sólo imágenes tomadas para una condición (por ejemplo, soleado), y probar su rendimiento en imágenes tomadas en la otra condición (por ejemplo, nublado). Concretamente, el experimen-

to se llevó a cabo sobre dos grupos de 20 imágenes cada uno, el conjunto soleado y el conjunto nublado. De cada grupo se seleccionaron aleatoriamente 4 imágenes para el entrenamiento de los dos modelos MG, llamados $MG_{soleado}$ y $MG_{nublado}$ respectivamente, dejando un total de 16 imágenes en cada conjunto para la evaluación del algoritmo, denotados $Test_{soleado}$ y $Test_{nublado}$ respectivamente. Los dos modelos y los dos conjuntos de prueba se utilizaron en dos evaluaciones cruzadas. Por un lado las 16 imágenes del conjunto soleado fueron segmentados utilizando el modelo nublado, caso denotado como $Test_{soleado}|MG_{nublado}$, y por otro lado las 16 imágenes del conjunto nublado fueron segmentados utilizando el modelo soleado, caso denotado como $Test_{nublado}|MG_{soleado}$. Para evaluar la variabilidad de estas segmentaciones, cada caso se contrastó con el caso de igual condición, es decir, el conjunto soleado contra el modelo soleado, denotado como $Test_{soleado}|MG_{soleado}$, y el conjunto nublado sobre el modelo nublado, denotado como $Test_{nublado}|MG_{soleado}$, y el conjunto nublado sobre el modelo nublado, denotado como $Test_{nublado}|MG_{nublado}$. En resumen: $Test_{soleado}|MG_{nublado}$ se evaluó contra $Test_{soleado}|MG_{soleado}$; mientras que $Test_{nublado}|MG_{soleado}$ se evaluó contra $Test_{nublado}|MG_{nublado}$.

Resultados

En este apartado se informan y discuten los resultados obtenidos para los dos experimentos. En ambos casos se reporta alguna agregación de los errores para los métodos ICA y MBC.

Errores de operador. En este experimento se evaluó la media y la desviación estándar del error absoluto de las 840 mediciones de los operadores. Los resultados muestran que el error medio es menor para ICA, con 1,05 mm contra los 1,34 mm de MBC. La misma tendencia se mantiene para las desviaciones estándar, con 0,95 mm de ICA en contra de 1,44 mm para MBC. Una desviación estándar más pequeña implica que es más probable que ICA tenga errores más cercanos a la media, siendo más preciso que MBC. Estos resultados demuestran, al menos empíricamente, que el método ICA es equivalente (aunque ligeramente mejor) que el método MBC. También se realizó un análisis más detallado de los valores atípicos (outliers) del método ICA y MBC. Para ello, se ha calculado un histograma en base al número de errores que produce cada método. Para cada valor de error posible, que va desde 0 mm hasta 18 mm (el mayor error cometido), el histograma indica la cantidad de mediciones que dieron lugar a un error en un rango específico (véase la figura 3.12). Primero se observa que de las 840 mediciones, el 62,2 % de los errores de MBC son más pequeños que la media máxima de 1,34 mm (i.e. la media de MBC), mientras que ICA tiene 7,2 % más casos en ese intervalo, es decir 69,4 %. Además, todos los errores del método ICA son menores a 6,2 mm, mientras que el 98,6 % de las mediciones de MBC están dentro de

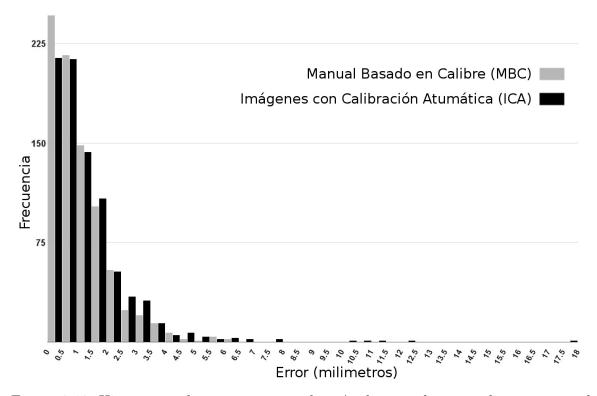


Figura 3.12: Histograma de errores para cada método en cada rango de error para el análisis de valores atípicos.

ese intervalo, con el restante 1,4% alcanzando errores de hasta 18 mm. El 1,4% de las 840 mediciones representa 12 medidas, cuya causa más probable es un error humano.

Por último, se analizaron estos resultados con más de detalle para cada una de las 28 rondas de medición individuales llevadas a cabo por los operadores para cada método, donde se reportó el error medio y desviación estándar sobre las 30 mediciones de cada ronda. Estos resultados se muestran en la figura 3.13, con una curva por método (y el área debajo de cada una). Los resultados fueron ordenados de mayor a menor por error medio. La curva gris oscuro corresponde a ICA y la gris claro a MBC. Las barras en cada ronda muestran la magnitud de la desviación estándar. Teniendo en cuenta el ordenamiento impuesto, es importante tener notar que las rondas para los métodos ICA y MBC con el mismo número no son directamente comparables entre sí. Sin embargo, el gráfico muestra una clara mejora de ICA sobre MBC: el área bajo la curva de ICA es menor que el área bajo la curva de MBC, i.e. la posibilidad de errores es menor para las rondas medidas con ICA que las medidos con MBC. Por ejemplo, la ronda con el error medio más pequeño utilizando MBC (la número 28), es mayor que el error medio de la ronda 17 obtenido por ICA, y por lo tanto mayor que las últimos 11 rondas de ICA.

Condiciones de luminosidad. Los resultados para este experimento se resumen

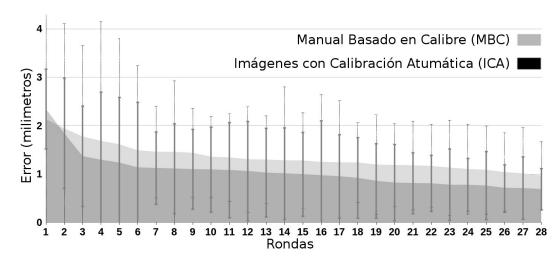


Figura 3.13: Error medio y desviación estándar para cada una de las 28 rondas de medición para las 30 plantas, para ambos métodos, ordenados por error medio de mayor a menor.

en la tabla 3.5. La tabla muestra la media y la desviación estándar de la diferencia de las mediciones obtenidas por el método ICA entre $Test_{soleado}|MG_{nublado}$ vs. $Test_{soleado}|MG_{soleado}$ (fila superior) y $Test_{nublado}|MG_{soleado}$ vs. $Test_{nublado}|MG_{nublado}$ (fila inferior), para cada una de las 16 imágenes de los conjuntos de prueba. Ambos casos muestran una media cercana a 0,45 mm, una discrepancia muy pequeña teniendo en cuenta que el error medio que comete el método ICA (como se muestra en el experimento anterior) es aproximadamente de 1 mm. Por lo tanto, esta evaluación empírica permite concluir que el algoritmo de segmentación es robusto a cambios en la luminosidad causadas por un cielo nublado. Esto justifica el uso de un solo modelo MG entrenado con imágenes en condiciones de luminosidad mixta, como fue el caso del primer experimento.

Comparación cruzada	Media	Desviación estándar
$Test_{soleado} MG_{nublado}$ vs. $Test_{soleado} MG_{soleado} $	0.421	0.328
$Test_{nublado} MG_{soleado} $ vs. $Test_{nublado} MG_{nublado} $	0.476	0.26

Tabla 3.5: Comparación de los casos nublados y soleados para el método ICA. Para cada medición obtenida sobre las 16 imágenes de prueba, se muestra la media y desviación estándar (en milímetros) de sus diferencias (se omiten las medidas). En la fila superior, se muestran los resultados que comparan el caso $Test_{soleado}|MG_{nublado}$ vs. $Test_{soleado}|MG_{soleado}$. En la fila inferior, los resultados para el caso $Test_{nublado}|MG_{soleado}$ vs. $Test_{nublado}|MG_{nublado}$.

3.5.3. Discusión sobre las aplicaciones

Diferentes métodos de medición autónomas se han introducido en la literatura con la intención de mitigar los errores que se resumen en la tabla 3.4. En general, esto métodos buscan mitigar el efecto de los errores manuales más que aumentar la precisión de las mediciones individuales, ya que los métodos manuales de medición ya logran la precisión suficiente requerida en la práctica. Un sistema de medición autónomo tiene el potencial de reducir la intervención humana durante el proceso de medición. Los métodos presentados en las secciones 3.5.1 y 3.5.2, si bien todavía requieren de un operador, su intervención en el proceso completo se reduce en gran parte. Por otro lado, en la práctica la pinza empleada en la segunda aplicación presentada podría ser fácilmente reemplazada por otros mecanismos, como ser un tablero de calibración (como ya se explicó anteriormente). Además, es posible transferir los métodos presentados a otros tipos de árboles frutales como ser manzana, pera, durazno, ciruela, y más; con dificultades técnicas menores asociados a la construcción de una abrazadera adecuada a las características del nuevo cultivo. Si bien no se ha evaluado esta propuesta, se espera que los resultados sean tan exitosos como los que se presentan en la sección anterior. Por otro lado, los resultados obtenidos por estos métodos asientan los primeros pasos hacia un sistema embebido dentro de un sistema móvil que mide de forma autónoma un gran número de plantas (por ejemplo, vehículos no tripulados de tierra o aire). La automatización completa podría lograrse, por ejemplo, mediante el uso de abrazaderas baratas permanentes unidos a los troncos de vid. Este sistema móvil autónomo debe ser capaz de identificar la abrazadera en la escena, la captura de la imagen perpendicularmente al plano de las pastillas de pala, y, finalmente, realizar el proceso de segmentación y de calibración. Un sistema de este tipo permitiría mediciones masivas de toda la población de las plantas, con los consiguientes beneficios en las operaciones de manejo de cultivos. Por último, vale la pena señalar algunos problemas abiertos alrededor de estos enfoque. Este método es todavía propenso a errores de tipo III (c.f. tabla 3.4), ya que las plantas no se identifican de forma autónoma, por lo que requiere un operario humano para asociar la foto con la planta. Sin embargo, esto no es difícil de automatizar. Por ejemplo, una solución consiste en incluir códigos QR en la escena para identificar automáticamente la planta [95].

3.6. Conclusiones y Trabajos Futuros

En este capítulo se han presentado varios enfoques de segmentación de madera sobre imágenes de viñedos, basados en algoritmos de clasificación a nivel de píxel, tales como: k-means, mezcla de gaussianas (MG) y máquinas de vector soporte (MVS).

Además se introduce un enfoque sencillo que mejora los resultados de los algoritmos MG y MVS mediante un proceso de reclasificación basado en información espacial local, llamados MG-RA y MVS-RA. Dado que k-means no es enfoque particularmente sofisticado para el problema de segmentación de imágenes, sus resultados se incluyen como línea base para la comparación del resto de los algoritmos presentados. Para la clase madera, k-means logra una exactitud de 0,644, precisión de 0,446 y sensibilidad de 0,814, mientras que los algoritmos MG y MVS mejoran la calidad global de k-means, obteniendo valores aproximados de sensibilidad pero muy superiores de precisión, lo que eleva el valor de la medida-F. Además es notable el incremento de la exactitud, que es de aproximadamente un 29 % para MG y de 32 % para MVS por sobre kmeans. Comparando directamente MG y MVS se puede observar que en general MVS obtiene mejores resultados para la clase madera, con exactitud de 0,852, precisión de 0,718, sensibilidad de 0,806, lo que produce un valor de medida-F de 0,751, superior al valor de medida-F de MG por un margen pequeño de 0,024. El análisis sobre MG-RA y MVS-RA revela mejoras considerables sobre las técnicas anteriores para madera. Particularmente, MG-RA logra una exactitud de 0,898, precisión de 0,96 y sensibilidad de 0,681 lo que provoca un valor de medida-F de 0,791. Comparando los resultados de las medidas de calidad para la clase madera y no-madera se observa que MG-RA es el algoritmo que mejores resultados de segmentación obtiene para las métricas planteadas en el presente trabajo. Por último, estos resultados se discuten en el contexto de los algoritmos de detección y su utilidad para codificar el conocimiento de sentido común para el problema específico de detección de yemas de vid en imágenes naturales de viñedos. Además, los resultados obtenidos en este capítulo son también contrastados sobre un caso de uso potencial de estas tecnologías: medición del diámetro de tronco en vides a partir de imágenes.

Respecto a las aplicaciones, los resultados obtenidos por los métodos basados en imágenes se presentan como potencialmente superadores a los métodos manuales, donde generalmente el diámetro del tronco se mide usando calibres Vernier. El método manual conlleva errores humanos que pueden ocurrir durante las campañas de medición debido a fatiga, prisa o descuido del operario. Hay varios estudios que abordan el problema de la automatización de la medición del diámetro de tronco en árboles. Sin embargo, estas soluciones no son óptimas: algunos son de bajo costo pero de baja precisión, y otros son de alta precisión, pero de alto costo. El análisis experimental para el método de medición basado en imágenes con calibración automática, muestra que es ligeramente más exacto que el método manual basado en calibre. Una ventaja importante sobre otros autores citados en este trabajo, es que la tecnología necesaria para implementar este método es barato y probablemente cualquier agrónomo ya cuen-

ta con los elementos principales: una computadora de escritorio y una cámara digital (3 megapíxeles). Además, el método basado en imágenes con calibración automática tiene dos ventajas sobre el método manual: evita errores I, II, y IV, que se resumen en la tabla 3.4; y es susceptible de ser totalmente automatizado, con el potencial de reducir los errores III y V.

Por último, los resultados obtenidos en este trabajo incentivan el estudio y la investigación de otras posibles aplicaciones en viticultura. De acuerdo a comunicaciones privadas con ingenieros agrónomos, otras variables de interés vitícola susceptibles de ser medidas por algoritmos de AM y VC son: área foliar, índice verde, conteo de yemas, conteo de flores, conteo de racimos por planta, conteo de racimos por brote, entre otros [69].

Capítulo 4

Clasificación de Imágenes para la Detección de Yemas de Vid mediante Algoritmos de Visión Computacional y Aprendizaje de Máquinas

En viticultura existen muchas aplicaciones en donde reconocer y detectar yemas de vid es una tarea necesaria que se realiza por observación e inspección visual. Por tanto, la detección de yemas es una tarea susceptible de ser automatizada mediante el uso de métodos de visión computacional. En este capítulo se presenta un enfoque de clasificación de imágenes para la detección de yemas de vid. Al igual que para el capítulo anterior, los resultados obtenidos aquí son utilizados como una herramienta para la construcción del conocimiento de sentido común necesario para el problema de búsqueda visual activa de yemas de vid presentado en el capítulo 5.

Concretamente, en este capítulo se presenta un algoritmo de clasificación de imágenes para yemas de vid con el objetivo de ser embebido en un proceso de detección de tipo escaneo-ventana para imágenes tomadas en condiciones naturales de campo durante la temporada de invierno. Esta técnica de detección engloba a una familia de algoritmos que se destacan por su eficacia y son usualmente empleados en la práctica. Su enfoque es realmente simple: una ventana de tamaño (generalmente) fijo se desplaza a lo largo de toda la imagen extrayendo sub-imágenes o parches, donde cada parche es clasificado en clase positiva o negativa según el objeto que se quiere detectar. Por lo tanto, la simplicidad de estos algoritmos tiene su aspecto más desafiante en la fase de clasificación. El método de clasificación de parches propuesto aquí utiliza un conjunto

de tecnologías propias de visión computacional: características invariantes en escala transformada para el cálculo de características de bajo nivel, bolsa de características para la construcción de un descriptor de imagen, y máquinas de vector soporte para el entrenamiento de un clasificador de sub-imágenes. Este enfoque logra una sensibilidad de clasificación superior a 0,97, para imágenes que contienen entre 90 % y un 100 % de los píxeles de una yema, y estos píxeles representan entre 20 % y un 30 % de la imagen, i.e. imágenes de tres a cinco veces más grande que la yema que la contienen casi en su totalidad. Por último, los resultados obtenidos en este capítulo se discuten en el contexto de un sistema de detección de tipo escaneo-ventana para yemas de vid.

4.1. Introducción

En la viticultura de precisión los algoritmos de visión computacional son usados generalmente para adquirir información sobre los viñedos [164], como ser detección de frutos y racimos [114], estimación del tamaño y peso del fruto [146], detección de yemas [167], indices de área foliar y estimación de la producción [49], fenotipado de plantas [68], pulverización selectiva autónoma [15], y más[147]. Entre todas estas aplicaciones, la clasificación y detección de yemas de vid en su entorno natural es una aplicación que se presenta desafiante para los algoritmos de visión computacional, mientras que tiene un gran impacto como posibles solución a muchos problemas en la viticultura. Los algoritmos de detección de objetos están diseñados para detectar todas las instancias de objetos de una clase conocida en una imagen. Por lo general, sólo un pequeño número de estas instancias de objetos están presentes en la imagen, pero hay un gran número de posibles ubicaciones y escalas en las que pueden aparecer [73]. Los algoritmos de detección escaneo-ventana son una familia popular de algoritmos elegidos en la práctica cuando se requiere un sistema de reconocimiento de alto rendimiento [51, 162]. En este tipo de algoritmos, cada imagen es escaneada densamente de un extremo a otro (por ejemplo, desde la esquina superior izquierda hasta la esquina inferior derecha) mediante una ventana rectangular deslizante en diferentes escalas, con el objetivo de extraer sub-imágenes o parches de la imagen original. Para cada parche se computan diferentes características visuales, como ser bordes, regiones, texturas, gradientes, y más, y se alimenta a un clasificador previamente entrenado usando parches etiquetados en sus respectivas clases. Algunos ejemplos típicos de algoritmos para clasificación son máquinas de vector soporte, árboles de decisión, redes neuronales artificiales, bosques aleatorios, clasificación estadística, y más. El clasificador discrimina los nuevos parches que contienen el objeto deseado como casos positivos, y el resto como casos negativos. Finalmente, la detección de un objeto en particular se

produce cuando un número de parches en alguna región de la imagen se clasificaron positivos. En este esquema de detección escaneo-ventana, la función del algoritmo de clasificación es esencial, relegando el resto de los detalles de implementación, como ser el tamaño y la forma de la ventana, el desplazamiento y la escala, a las decisiones de diseño particulares al problema de detección que se intenta resolver.

Por lo tanto, en este capítulo se presenta un algoritmo de clasificación binaria para parches de yema de vid que van de 100 a 1600 píxeles de diámetro, tomadas en condiciones naturales de campo en invierno (es decir, sin racimos, muy pocas hojas y con yemas no brotadas), que cumpla los requisitos de un algoritmo de detección escaneoventana. Este enfoque fue construido usando algoritmos de visión computacional conocidos, como ser detección de características invariantes en escala transformada [98], bolsa de características [44] y máquinas de vector soporte [153]. El enfoque asume que los parches contienen como máximo una yema. El mismo es robusto a parches que contienen al menos un 60 % de los píxeles de la yema, mientras que la cantidad de píxeles de la yema en el parche representa entre un 20 % y un 80 % en comparación con los píxeles no-yema (es decir, partes del entorno natural del viñedo que no son la yema). Como se muestra en la sección 4.3, los resultados obtenidos por el algoritmo de clasificación son lo suficientemente robustos para su uso dentro de un algoritmo de detección de yemas de tipo escaneo-ventana.

Vale la pena resaltar que una tecnología de detección de yemas efectiva presenta importantes oportunidades de aplicación en viticultura, tales como: medición de variables de interés vitícola (como ser conteo de yemas), fenotipado autónomo de yemas, poda autónoma de la vid, reconstrucción 3D de la estructura de la planta y sus componentes, entre otros. A continuación se discuten brevemente algunas aplicaciones para ayudar a evaluar mejor el impacto potencial de esta contribución. La poda de la vid es una tarea clave en la gestión de viñedos que busca el equilibrio entre la productividad de la planta y la calidad de los frutos [83]. La operación manual es el principal método de poda, requiriendo trabajo físico durante largas jornadas laborales. La necesidad de operarios calificados hace a la poda una de las tareas más costosas en la administración de viñedos [17]. Sin embargo, es posible reducir los costos de poda mediante la mecanización total de esta tarea [12], o mediante la aplicación de sistemas expertos de poda para ayudar a operadores menos calificados [41]. Tal sistema debe ser capaz de encontrar las diferentes yemas de una planta, lo que supone clasificación, detección, y otras operaciones complejas. Hasta la fecha, se han encontrado en la literatura algunos estudios preliminares que ponen su atención en este problema [167, 41, 55], pero estas son soluciones aproximadas, es decir, bajo condiciones controladas o en entornos simulados computacionalmente. Otra aplicación potencial para la detección de yemas

es el fenotipado de plantas, una herramienta cada vez más utilizada en la viticultura. A través de la fenotipificación es posible llevar a cabo una evaluación cuantitativa de las características de las plantas, para obtener su descripción anatómica y fisiológica [16]. En general, el fenotipado se lleva a cabo mediante la inspección visual directa de la planta, por lo que está fuertemente limitada por el tiempo, los costos y la subjetividad del operador. En la actualidad, existen plataformas de fenotipado de alto rendimiento basadas en técnicas de análisis y procesamiento de imágenes, generalmente limitadas a obtener una caracterización inicial de alguna parte específica de la planta [85, 87, 156, 65]. Para la vid, la etapa de desarrollo de las yemas durante la brotación es uno de sus rasgos fenotípicos. Por lo tanto, el fenotipado de alto rendimiento se acerca más a la realidad cuando la clasificación y detección autónoma de yemas es lo suficientemente madura para distinguir diferentes niveles de desarrollo. Una tercera aplicación posible para la detección de yemas es la reconstrucción 3D de la estructura de la planta, es decir, la forma, la ubicación y la orientación de todos sus componentes. La ubicación 3D de los componentes de una planta real puede ser útil para mejorar los procesos actuales para la simulación de respuestas fisiológicas y procesos físicos de la vid, tales como las variaciones en la intercepción de la luz y su distribución sobre la estructura de la vid, que afecta su productividad, rendimiento y calidad [139]. En los últimos años han surgido diferentes métodos para simular la estructura 3D de diversas plantas [72, 97]. Para adquirir modelos 3D más realistas, estas simulaciones pueden combinarse con información obtenida a partir de mediciones en 2D o 3D sobre la estructura real de la planta [136]. La detección de la posición 3D de las yemas es útil en la reconstrucción 3D, ya que la posición está en correlación directa con otros componentes de la planta debido a que las yemas aparecen junto a los nudos, muy cerca del lugar donde crecen hojas, racimos y zarcillos. Por ejemplo, la detección de yemas en invierno, cuando las plantas han perdido la mayor parte de las hojas, racimos y zarcillos, se puede utilizar para mejorar la simulación 3D de una planta real según la estructura que presentó el verano anterior. Estas mediciones se pueden obtener de forma autónoma a través de algoritmos de visión computacional.

4.1.1. Trabajos relacionados

En esta sección se discuten algunos trabajos relacionados disponibles en la literatura, tanto aquellos que presentan soluciones alternativas al problema de clasificación y detección de yemas, así como trabajos que presentan soluciones a problemas similares.

Luego de realizar una profunda revisión bibliográfica, no se ha podido encontrar en la literatura una solución práctica al problema de clasificación yemas de vid. En cambio, sí se encontraron dos trabajos del estado del arte que resuelven el problema

de la detección de vemas de vid. En Xu et al. (2014) [167] se presenta un algoritmo de detección de yemas de vid, cuya motivación principal es establecer las bases para un sistema autónomo de poda de vid en invierno. La detección de yemas se realiza a partir de imágenes RGB (su resolución no se informa), capturadas con una cámara CCD industrial¹, en interior, con un fondo artificial blanco e iluminación controlada. Para discriminar los píxeles de la planta del los píxeles del fondo, se aplica un filtro basado en umbral que resulta en una imagen binaria, que luego se somete a un proceso de adelgazamiento para la obtención de un esqueleto de alambre de la planta. Bajo el supuesto de que las características morfológicas de las yemas son similares a las esquinas, se aplica el algoritmo de Harris [64] a la imagen de esqueleto para la detección de esquinas. Este método consigue una sensibilidad (o tasa de verdaderos positivos) de 0,702. Mientras que los autores explican posibles mejoras para aumentar la tasa de detección, la limitación más importante de este enfoque es la necesidad de obtener imágenes en condiciones controladas de laboratorio. Un segundo trabajo para la detección de yemas es presentado por Herzog et al. (2015) [68]. En este trabajo se introducen tres métodos para la detección de yemas en estadio de desarrollo avanzado (i.e. BBCH² 10) llevado a cabo en viñedos experimentales. Los resultados más relevantes se consiguen mediante el método semi-automático, que requiere la interacción del usuario para validar la calidad de los resultados. La detección está basada en imágenes RGB de 3456×2304 píxeles de resolución, donde la escena es alterada con un fondo negro artificial, produciendo un sensibilidad de 0,94. Los autores argumentan que el sensibilidad de este enfoque es lo suficientemente buena para satisfacer las necesidades del fenotipado de plantas en esta etapa de desarrollo. Sin embargo, como señalan los mismos autores, estos resultados se explican por la morfología y color de las yemas al momento de capturar las imágenes. A principios de BBCH 10 las yemas ya están empezando a brotar. Estos brotes son en su mayoría pequeños, crecen cerca de la rama de manera no uniforme y tienen colores similares (amarillo, verde claro, beige o marrón) a otros componentes de la planta. Por lo tanto, las imágenes son capturadas unos pocos días después de la primera identificación de BBCH 10, cuando los brotes de las yemas están visiblemente verdes y su tamaño promedio es alrededor de 2 cm, lo que resulta en una morfología más distinguible de otros componentes de la planta. Lamentablemente no se proporcionan los detalles de implementación de algoritmos de procesamiento de imágenes. Estos trabajos presentan avances en aplicaciones específicas para la detección de yemas, sin embargo, sufren alguna de las siguientes

¹Los sensores CCD (Charge-Coupled Device) suelen ser más caros que los sensores CMOS (Complementary Metal Oxide Semiconductor), más comunes en las cámaras compactas.

²En biología, la escala BBCH (Biologische Bundesanstalt, Bundessortenamt und Chemische Industrie) para viñedos es un modelo que describe el desarrollo fenológico de la vid [96].

limitaciones que no existen en el enfoque introducido en este capítulo: (i) uso de un fondo artificial; (ii) iluminación controlada en interior; (iii) necesidad de interacción del usuario; o (iv) detectar yemas en fase muy avanzada de desarrollo. Estas limitaciones y la sensibilidad reportada en estos trabajos representan una barrera para el desarrollo de soluciones a los problemas mencionadas en la introducción.

En la literatura también se pueden encontrar trabajos basados en algoritmos de visión computacional para la detección de diferentes componentes de la vid además de yemas. Por ejemplo, existen muchos enfoques para realizar la detección de uvas y racimos a partir de imágenes RGB, con variadas aplicaciones que van desde fumigación autónoma hasta estimación de la producción. Nuske et al. (2011) [114] presenta un enfoque para detectar uvas verdes basado en características de forma y textura, sin oclusión, manteniendo el fondo natural, alcanzando una precisión de 0,98 y un sensibilidad de 0,64. En Reis et al. (2012) [127] se introduce un detector de racimos de uvas rojas y blancas utilizando técnicas de procesamiento de imágenes muy simples, como ser conteo de píxeles, umbral y asignación de colores, con un sensibilidad de 0,91 para las uvas blancas y 0,97 para las rojas. Otra aplicación novedosa de algoritmos de detección en viticultura es la evaluación del número de flores por inflorescencia, siendo una de las principales fuentes de información para la estimación temprana el rendimiento de la vid. Diago et al. (2014) [48] desarrolla un método sencillo, económico, rápido, preciso y robusto basado en algoritmos de procesamiento de imágenes para estimar de forma autónoma el número de flores por inflorescencia. Este enfoque se aplica a las imágenes RGB capturadas en condiciones de campo, logrando una precisión de 0,93 y un sensibilidad de 0,74. Si bien existen soluciones a diferentes niveles para la detección de partes de la vid, la clasificación y detección de yemas sigue siendo un problema abierto que se encuentra actualmente en estudio por las comunidades de viticultura de precisión y visión computacional. Por otra parte, se considera que los trabajos aquí descritos constituyen una sólida base de evidencia sobre la importancia de los algoritmos de detección de objetos y de clasificación de imágenes en las ciencias agrícolas y la viticultura de precisión.

4.2. Elementos de trabajo y metodología

En esta sección se presenta las tecnologías empleadas para construir el modelo de clasificación, junto al procedimiento para capturar las imágenes RGB y generar los parches de entrenamiento, y se dan los detalles de implementación del algoritmo de clasificación.

El algoritmo presentado en este trabajo tiene como objetivo resolver una de las eta-

pas más complejas durante la ejecución de un algoritmo de detección escaneo-ventana para la detección de yemas de vid: la clasificación de parches de la imagen en clase yema (i.e. parches que contiene una yema) o clase no-yema (parches que no contienen píxeles de yema). Esta propuesta se basa en un enfoque de aprendizaje supervisado para aprender el modelo de clasificación, utilizando como entrada parches etiquetados como yema y no-yema, obtenidos mediante un proceso de escaneo-ventana, sobre imágenes de viñedos representativas tomadas en condiciones naturales de campo, ejemplificado en la imagen de la figura 4.1. El método propuesto se puede separar en tres partes principales: generación del corpus de parches etiquetados, cálculo del descriptor visual de cada uno de esos parches, y aprendizaje del modelo de clasificación. Para generar el corpus etiquetado, en primer lugar se capturaron las imágenes en campo usando cámaras digitales compactas. A continuación, se generó un corpus de parches de acuerdo a los requerimientos de los algoritmos de detección escaneo-ventana, y finalmente se etiquetaron estos parches manualmente como yema o no-yema. Para calcular el descriptor visual de cada parche, el algoritmo extrae un conjunto de características visuales de bajo nivel y construye un descriptor representativo del parche, lo que resulta en un nuevo corpus de descriptores etiquetados, con un elemento en el corpus por parche. Por último, el aprendizaje del modelo de clasificación de yema/no-yema consistió en el entrenamiento de un clasificador binario supervisado tomando como entrada un subconjunto de este corpus etiquetado, denominado en adelante conjunto de entrenamiento. El resto de los parches etiquetados, llamadas en adelante conjunto de prueba, se utilizan más tarde en la sección 4.3 para la evaluación del modelo de clasificación. En la siguiente subsección se explican cada una de estas etapas con mayor detalle.

4.2.1. Corpus de parches etiquetados

El corpus de parches etiquetados empleados en este trabajo ha sido generado con el requerimiento de aprender un clasificador que discrimina el concepto visual de yema frente al resto de los elementos de la escena. La clase no-yema es todo lo que no es una yema, por ejemplo, tallos, hojas, troncos, alambres, suelo, cielo, y más. En este escenario, el enfoque propuesto tiene como objetivo clasificar parches extraídos de imágenes de alta resolución. Estas imágenes de alta resolución fueron capturados en viñedos de la Facultad de Ciencias Agrarias, Universidad Nacional de Cuyo, y en los viñedos del Instituto Nacional de Tecnología Agropecuaria (INTA), sede Luján de Cuyo, Mendoza, Argentina. Estas imágenes fueron tomadas con diferentes dispositivos: (1) cámara compacta Nikon Coolpix P530; (2) cámara reflex compacta Samsung NX1000; y (3) cámara Samsung SM-G920I (celular Samsung s6). Todas las imágenes



Figura 4.1: Izquierda: imagen de alta resolución capturada en campo. Derecha: parche yema extraído de la imagen de la izquierda (marcada por el recuadro rojo).

fueron capturadas en formato JPEG con una resolución que varía de 12 megapíxeles a 20 megapíxeles, dependiendo del dispositivo utilizado. Las imágenes fueron capturadas en condiciones normales de campo, sin alterar la escena con elementos artificiales y manteniendo las condiciones de iluminación natural. Sólo para algunos casos se activó el flash de la cámara a fin de producir un conjunto de imágenes con una variabilidad visual mayor. En total fueron capturadas 760 imágenes, conteniendo una o más yemas por imagen, junto con otros elementos del viñedo y su entorno natural (ver figura 4.1). Las imágenes fueron capturadas entre las 10hs y las 16hs, en diferentes campañas realizadas entre el 10 de julio y el 24 de septiembre de 2015 (segunda mitad del invierno de Argentina), cuando las hojas de las plantas están secas o se han caído, pero antes de que comenzaron a brotar nuevamente.

Todos los escenarios considerados aquí fueron diseñados con el objetivo de abarcar todos los escenarios posibles que un algoritmo de detección escaneo-ventana podría enfrentar. En todos los casos, se requiere la clasificación de una porción rectangular de la imagen. Usando las imágenes originales de alta resolución, se han generado parches como los que normalmente producen los algoritmos de detección escaneo-ventana, donde la proporción de píxeles de yema y no-yema varían de una muestra a otra, dependiendo de su tamaño, así como su ubicación en la imagen original. Para el entrenamiento del clasificador, se consideran dos categorías diferentes de parches, una para cada concepto, llamados parches yema y parches no-yema respectivamente (véase la figura 4.2). Los parches yema son regiones rectangulares que circunscriben a la yema, lo que resulta en parches que contienen diferentes proporciones de píxeles yema y no-yema según la forma de la yema, como se ilustra en las imágenes de la primera fila de la figura 4.2. Por el contrario, parches no-yema consisten en regiones



Figura 4.2: Corpus de parches utilizados en este trabajo. La primera fila corresponde a parches yema. La segunda y tercera fila corresponden a parches no-yema. La fila inferior muestra los parches yema que fueron desechados por estar fuera de foco, sobreexpuestos o subexpuestos.

rectangulares que no poseen píxeles de yema, correspondiente al fondo y otras partes de la planta, como se ilustra en las imágenes de la segunda y tercera fila de la figura 4.2. La motivación principal para generar este corpus con parches de yemas circunscritas es conseguir parches yema "ideales", tratando de maximizar la proporción de píxeles de yema. Debido a la forma no rectangular de las yemas, es inevitable incorporar píxeles no-yema cuando se extrae un parche rectangular que contiene a toda la yema. En promedio, los parches yema contienen un 25 % de píxeles no-yema, y por lo general se agrupan en áreas cercanas a las esquinas del parche. Para construir este corpus, fueron extraídos parches de diferentes geometrías a partir de las 760 imágenes de viñedos de alta resolución. Estos parches, además de rectangulares, son de tamaño variable, con un rango de resoluciones desde 0,1 hasta 2,56 megapíxeles (correspondientes a los parches que van desde 100×100 a 1600×1600 píxeles aproximadamente). El corpus de parches resultante contiene un total de 790 parches yema y más de 25 200 parches no-yema³. El proceso para la toma de muestras de parches yema y no-yema se realizó mediante una herramienta de selección a mano alzada, que forma parte del software de

³Estas imágenes se encuentran disponibles en http://dharma.frm.utn.edu.ar/vise/bc/

manipulación de imágenes desarrollado por los autores⁴. El protocolo utilizado para extraer estos parches fue el siguiente:

- (I) Parches yema: cada yema fue segmentada manualmente por un operador utilizando la herramienta de selección a mano alzada. El operador fue instruido para realizar la segmentación de la forma más precisa y más cuidadosa posible. La segmentación manual tiene un doble propósito, i.e. para encajar la yema en su totalidad en un parche rectangular y para crear una máscara de la yema, la cual permite calcular exactamente la cantidad de píxeles yema y no-yema en cada parche (información utilizada más tarde durante el evaluación del enfoque propuesto). Los parches yema fueron procesados manualmente para desechar aquellos que presentan una imagen fuera de foco, sobreexpuesta o subexpuesta (ver ejemplos en la fila inferior de la figura 4.2), dejando un total de 500 parches yema. Descartar estos casos no viola la viabilidad del enfoque, ya que en la práctica es viable evitar este tipo de distorsiones en la imagen, ya sea por simple inspección visual, o de forma autónoma como se ejemplifica en las obras de [71, 93].
- (II) Parches no-yema: estos parches también fueron extraídos a través de la herramienta de selección a mano alzada mediante un proceso semi-automático. El operador selecciona una región de la imagen que pertenece a esta clase. Entonces el software toma varias muestras de parches rectangulares, todas ellas dentro de la región seleccionada, con un desplazamiento y tamaño de parche pre-establecido. Este procedimiento funciona de manera similar a un algoritmo escaneo-ventana, pero limitado a escanear en una región restringida de la imagen entera. Con este procedimiento se podrían obtener una gran cantidad de parches no-yema, órdenes de magnitud más que para los parches yema.

4.2.2. De píxeles a descriptores visuales

El siguiente paso consiste en calcular los descriptores visuales para cada parche, lo que resulta en un nuevo corpus que contiene un descriptor visual por parche. Este proceso consiste en método de dos pasos:

(I) Dado un parche, el primer paso consiste en calcular su conjunto de características visuales locales de bajo nivel. Estas características se calculan utilizando el algoritmo características invariantes en escala transformada (SIFT por sus siglas en Inglés) [98], llamadas características SIFT de aquí en adelante.

⁴El código fuente de esta herramienta está disponible en http://dharma.frm.utn.edu.ar/vise/bc/

(II) A continuación, el descriptor visual del parche se construye mediante la asignación de sus características SIFT de bajo nivel a un vector de tamaño constante definido por el usuario. Este vector, considerado el descriptor de alto nivel del parche, es obtenido mediante el algoritmo bolsa de características (BoF por sus siglas en Inglés) [44], denominado descriptor BoF a partir de ahora. Este descriptor es la información utilizada en las fases de entrenamiento y clasificación.

En el siguiente apartado se presentan algunas ideas del algoritmo SIFT y BoF, junto con algunas decisiones de implementación particulares para este trabajo.

Características Invariantes en Escala Transformada (SIFT)

SIFT es uno de las algoritmos más populares para la extracción de características visuales y cómputo de descriptores en imágenes. En líneas generales, el algoritmo toma una imagen y la transforma en un conjunto de vectores de características locales que son invariantes a escala, rotación o traslación de la imagen. La explicación detallada del algoritmo queda fuera del alcance de este capítulo, pero se puede consultar el trabajo original de Lowe (2004) [98] para conocer todos los detalles. A continuación se presenta la intuición del algoritmo SIFT resumida en cuatro pasos:

- (I) Detección de extremos en espacio de escalas. Los objetos del mundo real sólo tienen sentido en una cierta escala que depende del tamaño del objeto y del punto de vista del observador. Esta propiedad multi-escala de los objetos es muy común en la naturaleza, y es posible aproximar este concepto en imágenes digitales a partir de funciones especiales, conocidas como funciones en espacios de escalas. SIFT propone una función que toma una imagen original y produce una sucesión de imágenes con diferentes niveles de borrosidad. Primero obtiene imágenes de menor resolución que la original mediante cambios en escala de la imagen, y a continuación convoluciona cada píxel de cada imagen con una gaussiana centrada en ese píxel. Para identificar los potenciales puntos de interés, i.e. lugares que son identificables a diferentes escalas del mismo objeto, se aplica una Diferencia de Gaussianas (DG) entre dos imágenes sucesivas de la misma escala para obtener una imagen DG. Los puntos de interés potenciales son tomados como los máximos y mínimos locales de la imagen DG que se producen en las diferentes escalas. Para detectar estos máximos y mínimos locales, cada píxel se compara con sus 8 vecinos en la imagen DG, y sus 9 vecinos de la imagen DG anterior y posterior de la misma escala.
- (II) Localización de punto de interés. El paso anterior produce una gran cantidad de potenciales puntos de interés, donde algunos se encuentran a lo largo de un borde

o no tienen suficiente contraste. Este tipo de puntos de interés resultan inestables y no son útiles como características, por lo que es necesario un refinamiento para obtener resultados más precisos. Para tomar la decisión de descartar o mantener un punto de interés, el algoritmo calcula la curvatura de la imagen DG en dos direcciones sobre ese punto. Se considera que existe un borde si la curvatura en una dirección es grande (perpendicular al borde) y la otro pequeña (a lo largo del borde), y por lo tanto se rechaza ese punto. Si ambas curvaturas son lo suficientemente grandes, por encima de un umbral, se considera que existe una esquina y el punto se mantiene. Si ambas curvaturas son pequeñas, el punto se encuentra sobre una región plana y se mantiene. Para descartar puntos de bajo contraste simplemente se comprueba que su intensidad se encuentre por encima de cierto umbral.

- (III) Asignación de orientaciones. Hasta aquí se han calculado un conjunto de puntos de interés estables, invariantes a la escala. Ahora es necesario asignarle a estos puntos una orientación consistente basada en las características locales de la imagen. Este paso es clave para lograr invarianza a la rotación de la imagen. Para esto se toma un vecindario alrededor de cada punto dependiendo de la escala en la que se detectó (a mayor escala más grande es la región) y se calculan las magnitudes y direcciones de los gradientes en cada píxel. Se crea un histograma de orientaciones con 36 bins que abarcan los 360 grados. El bin más prominente del histograma se toma como gradiente del punto. Para cualquier otro bin que supere el 80% del bin más prominente, se crea un nuevo punto de interés y se le asigna ese gradiente, i.e. se pueden crear puntos de interés con la misma ubicación y escala, pero con diferentes direcciones.
- (IV) Descriptor del punto de interés. En este paso, ya existe un conjunto de puntos de interés invariantes a escala con una orientación asignada. El último paso consiste en crear un descriptor para cada punto que sea invariante a rotaciones de la imagen. Para esto se toma un vecindario de 16 × 16 píxeles alrededor del punto de interés. A su vez, este vecindario se divide en 16 regiones de 4×4 píxeles. Para cada región, se calculan las magnitudes y direcciones de los gradientes en cada píxel. Se calcula un histograma de direcciones de 8 bines y se asigna a cada región la dirección correspondiente al bin más alto. Para que este descriptor tenga la propiedad de ser invariante a rotación, se resta las direcciones del gradiente de cada región a la dirección del gradiente del punto de interés, de manera que ahora cada gradiente de región es relativo al gradiente del punto. Al finalizar este proceso se obtiene un vector de características SIFT de 16 * 8 = 128 valores

que forma el descriptor del punto de interés, invariante a escala, rotaciones y traslaciones.

Una vez que se han calculado los vectores de características SIFT para diferentes imágenes, se cuenta con un número variable de vectores por imagen, con el número de vectores en función de las características visuales particulares de cada imagen. Sin embargo, un algoritmo de clasificación requiere información de entrada de tamaño constante, i.e. un descriptor de imagen que consiste en un conjunto de variables asignadas con el mismo tamaño para todas las imágenes del corpus. Para obtener un descriptor de tamaño homogéneo dado un número variable de vectores SIFT, se utiliza el algoritmo de BoF, el cual se describe en la siguiente sección.

Bolsa de Características (BoF)

BoF es un descriptor visual muy popular utilizado en clasificación de imágenes. Básicamente produce una representación compacta y de tamaño constante de una imagen a partir de una colección de características locales de bajo nivel [44]. El cómputo de estas características es independiente de BoF, pudiendo ser computadas a partir de cualquier algoritmo de extracción de características locales disponible en la literatura, siendo SIFT uno de los más populares. BoF está inspirado en un concepto llamado bolsa de palabras originalmente empleado para la clasificación de documentos. Una bolsa de palabras es un vector que mantiene un contador de ocurrencias para cada palabra o término interesante en el documento, i.e. un histograma sobre el vocabulario empleado. Siguiendo este principio, la idea es tratar de entender las imágenes como una colección de palabras visuales sobre puntos de interés, como lo son las características de bajo nivel SIFT. Como se explicó en el apartado anterior, los puntos de interés son pequeñas regiones en la imagen que pueden tener cualquier tipo de información local interesante en un espacio de características específicas, como ser colores, texturas, líneas, formas y mucho más. Por lo general, estos espacios de características tienen alta dimensionalidad y valores continuos, lo que implica un número enorme y variable de los puntos de interés. Por lo tanto es necesario agrupar los puntos de interés que presentan características comunes y que pueden ser considerados como la misma palabra visual. Al tratar cada grupo como una palabra visual que representa un patrón local específico compartido por los puntos de ese grupo, se cuenta con un vocabulario de palabras visuales que describen diferentes tipos de patrones locales en una imagen. Si todas las imágenes se refieren al mismo vocabulario (es decir, conjunto de palabras visuales), entonces todas las imágenes pueden hablar el mismo idioma (o lenguaje visual). Luego, una vez que todos sus puntos fueron asignados a palabras visuales de un vocabulario,

una imagen puede ser representada como una bolsa de características visuales [168]. Con la agrupación de puntos de interés en palabras visuales ahora es posible describir imágenes mediante un histograma de puntos sobre el vocabulario, matemáticamente modelado como un vector sobre las palabras visuales que, para una imagen dada, mantiene un contador de ocurrencias de cada palabra [144]. BoF implica dos pasos principales:

- (I) El primer paso es un proceso fuera de línea que consiste en construir un vocabulario de características visuales a partir de un conjunto de entrenamiento de parches (más adelante en esta sección se explica cómo se creó este conjunto). Para lograr una representación compacta, el vocabulario se genera por la agrupación de vectores en el espacio de características visuales a través de un algoritmo de agrupamiento, como ser k-means. Con SIFT, BoF realiza el agrupamiento de acuerdo al vector 128-dimensional SIFT para cada punto de interés calculado para las imágenes del conjunto de entrenamiento. Luego, cada grupo se trata como una palabra visual única en ese vocabulario visual. El tamaño del vocabulario, siendo el número de grupos definido en el algoritmo de agrupamiento, es una decisión de diseño que se establece externamente por el usuario.
- (II) El segundo paso de BoF comprende el cálculo de un histograma para una imagen según el vocabulario obtenido en el paso anterior. Dado un conjunto de vectores SIFT de una imagen, se calcula el número de vectores que se encuentran dentro de cada grupo, considerando que un vector pertenece al grupo cuyo centro de masa es el más cercano. Entonces, cada clase (o barra) del histograma reporta la cantidad de vectores en cada grupo. De esta manera el histograma provee una representación compacta y de tamaño constante de una imagen.

Más adelante, en la sección 4.3.3, se presentan los rasgos visuales más importantes caracterizados por los descriptores BoF para el corpus de imágenes de este trabajo. Si bien este enfoque es simple y no tiene en cuenta la información geométrica del objeto de interés, se ha demostrado que tiene un excelente rendimiento para diversas tareas de clasificación visual [76]. Para conocer el algoritmo en profundidad, puede consultar el documento de Csurka et al. (2004) [44] y el Capítulo 14.4 de Szeliski (2010) [144].

En este trabajo se ha utilizado la implementación del algoritmo SIFT provista en el módulo Non-Free de la librería $OpenCV^5$. Todos los parámetros del algoritmo se han dejado en sus valores por defecto. Si el lector se encuentra interesado en conocer estos valores puede consultar la documentación de la librería⁶ [79]. Para el caso del algoritmo

⁵http://opencv.org

⁶http://docs.opencv.org/3.1.0/d5/d3c/classcv_1_1xfeatures2d_1_1SIFT.html

BoF, también se usó la implementación provista en OpenCV. Para la creación del vocabulario se utilizó k-means como algoritmo de agrupamiento [18], mientras que la inicialización de centros se realiza según el algoritmo de Arthur y Vassilvitskii (2007) [8]. Existen diferentes parámetros involucrados en este proceso. El número máximo de iteraciones para k-means fue establecido en 100. El criterio de terminación por exactitud deseada fue fijado en 1×10^{-3} . Tan pronto como cada uno de los centros de grupo se mueve menos que este valor en alguna iteración, k-means se detiene. Durante el cómputo de los histogramas se utiliza el algoritmo de Muja y Lowe (2009) [110] para determinar de manera eficiente el centro más cercano para un punto de interés dado. Todos los parámetros de este algoritmo se han dejado en sus valores por defecto. Si el lector se encuentra interesado en conocer más detalles de la implementación puede consultar la documentación de la librería⁷, así como también la bibliografía disponible [79].

4.2.3. Algoritmo de clasificación

En este punto, cada parche del corpus cuenta con un descriptor BoF. La tercer etapa del enfoque consiste en entrenar un clasificador para que aprenda el concepto de yema y no-yema a partir de este conjunto de descriptores BoF etiquetados, llamado conjunto de entrenamiento. Más tarde, el clasificador es usado para decidir la clase más probable para un conjunto de descriptores de patches aún no clasificados. En este trabajo se utiliza el algoritmo máquinas de vector soporte (MVS) [153] para aprender el clasificador binario. Primero se presenta una breve descripción de MVS que resume la intuición general del algoritmo, y luego se dan los detalles para el entrenamiento del mismo.

Máquinas de vector soporte (MVS)

MVS fue explicado en los apartados 2.2.3 y 3.2.7. Como antes, para una revisión completa del mismo puede dirigirse a [19, 66, 28, 152]. En este trabajo se ha utilizado la implementación de MVS para clasificación binaria provista en la librería $LIBSVM^8$ [34]. Se utilizó una función kernel de tipo radial~(RBF) para el entrenamiento del clasificador. Los parámetros γ (propio del kernel) y C (penalización por cada punto mal clasificado) fueron establecidos en tiempo de entrenamiento sobre un conjunto de validación, tal como se explica en el siguiente apartado. La comprensión de los parámetros que definen al algoritmo MVS requiere de un profundo entendimiento

⁷http://docs.opencv.org/3.1.0/d4/d72/classcv_1_1BOWKMeansTrainer.html

⁸https://www.csie.ntu.edu.tw/~cjlin/libsvm/

teórico sobre el algoritmo y el problema, por lo que se invita al lector a seguir la referencias de Vapnik (2013) [152] o Bishop (2006) [18] si está interesado en conocer mayores detalles.

Entrenamiento del clasificador

Para el entrenamiento del clasificador MVS se utiliza un corpus compuesto por descriptores BoF de parches etiquetados (c.f. 4.2.1), con un total de 367 parches yema y más de 25 000 parches no-yema. Es importante notar que este conjunto de entrenamiento se encuentra altamente desbalanceado respecto a la cantidad de elementos de cada clase, con una relación de desequilibrio de 1 a 68 de la clase yema frente a la clase no-yema. Un conjunto desbalanceado es un problema para MVS durante su entrenamiento, ya que afecta negativamente su capacidad de generalización cuando los ejemplos negativos superan fuertemente en número a los ejemplos positivos [2]. Un enfoque muy simple utilizado frecuentemente para resolver el desbalance de clases consiste en preprocesar los datos haciendo sub-muestreo de la clase mayoritaria, sobremuestreo de la clase minoritaria, o una combinación de ambos, con el fin de crear un conjunto de datos balanceado. El sobre-muestreo implica aumentar el número de casos de una clase copiando elementos de forma aleatoria, mientras que el sub-muestreo consiste en disminuir el número de casos de una clase mediante la eliminación aleatoria de elementos (más detalles se pueden encontrar en el trabajo de Akbani et al. (2004) [2]). En este trabajo el balanceo del conjunto de entrenamiento se realizó mediante sobre-muestreo de la clase yema y sub-muestreo de la clase no-yema, por medio del siguiente procedimiento: (i) se definió una tasa de balanceo $R = \{1, 2, 4, 8, 16\}$; (ii) se definió el valor $parches_{yema} = 367$, i.e. la cantidad de elemento de clase yema en el conjunto de entrenamiento original; (iii) se realizó sub-muestreo de la clase no-yema de tamaño $R*parches_{yema}$; y (iv) se realizó sobremuestreo de tamaño $R*parches_{yema}$ de la clase yema. De esta manera el conjunto de entrenamiento se mantuvo balanceado pero se aumentó su tamaño en un factor R. El valor de R=1 se corresponde al caso especial donde sólo se realiza submuestreo de la clase no-yema. Una vez balanceado el conjunto de entrenamiento, se procedió a entrenar el clasificador MVS, el cual requiere establecer el valor de los parámetros C y γ . Los mismos se eligieron mediante una afinación de parámetros por validación cruzada de 5 iteraciones en tiempo de entrenamiento [18], para los rangos de valores $\gamma = \{2^{-14}, 2^{-13}, ..., 2^{-7}\}$ y $C = \{2^5, 2^6, ..., 2^{14}\}$. Estos rangos fueron preseleccionados por experimentación preliminar que no se detalla aquí. Finalmente, se tomó la mejor combinación de valores de C y γ como aquella que minimiza el error de validación promedio sobre las 5 iteraciones, donde el error de validación se definió como 1 menos medida-F de la clase yema, siendo la medida-F una medida de evaluación típica en problemas de clasificación [123].

4.3. Resultados experimentales

En esta sección se presentan los detalles de la experimentación realizada y las medidas de rendimiento empleadas para evaluar la calidad del algoritmo propuesto. Esta calidad es evaluada mediante la medición del error de generalización en un conjunto de parches etiquetados retenidos que no se utilizan durante el entrenamiento, denominado en adelante conjunto de prueba. Este conjunto consta de 133 parches yema y 133 parches no-yema, que corresponden a elementos del corpus original etiquetado que no se utilizan para el entrenamiento, es decir, que son conjuntos disjuntos, cuya unión produce el corpus original de parches etiquetados. Para calcular el error de generalización, se clasifica cada descriptor y su salida se compara con su valor de etiqueta. Cualquier discrepancia se cuenta como un error de generalización en las medidas de rendimiento utilizadas típicamente en problemas de clasificación binaria [123], que ya fueron introducidas en la sección 3.2.4: exactitud (ecuación 3.1), precisión (ecuación 3.2), sensibilidad (ecuación 3.3) y medida-F (ecuación 3.4). Intuitivamente, en este capítulo, exactitud se utiliza para evaluar la eficacia general del algoritmo y corresponde a la proporción de parches yemas y no-yemas que han sido clasificados correctamente; sensibilidad representa la proporción de parches yema clasificados correctamente; precisión especifica la proporción de parches clasificados como yema y que son realmente yema; y la medida-F es la media armónica de precisión y sensibilidad (para una explicación más detallada remitirse a la sección 3.2.4).

Como se explica en la sección 4.2.2, BoF requiere que el usuario especifique el tamaño del vocabulario visual, denotado por S. En el primer experimento, se muestran los resultados de rendimiento del clasificador para diferentes tamaños de vocabulario, con $S = \{12, 25, 50, 100, 200, 350, 600\}$, sobre un promedio de diez conjuntos de entrenamiento diferentes equilibrados con R = 1. El proceso de sub-muestreo para equilibrar el conjunto de entrenamiento descarta un gran número de elementos noyema (e.g. 67 de 68 cuando R = 1). Esto podría influir fuertemente en la calidad del clasificador aprendido, beneficiándose o perjudicándolo por la elección fortuita de elementos más o menos representativos. Para mitigar el impacto del sub-muestreo, se entrena una serie de clasificadores para conjuntos de entrenamiento del mismo tamaño (i.e. igual R), cada uno con exactamente los mismos elementos de la clase yema, pero con diferentes muestras de la clase no-yema seleccionadas al azar.

Los resultados para las cuatro medidas de rendimiento se presentan en la figura 4.3 y en la tabla 4.1, que resumen la media y la desviación estándar de las medidas

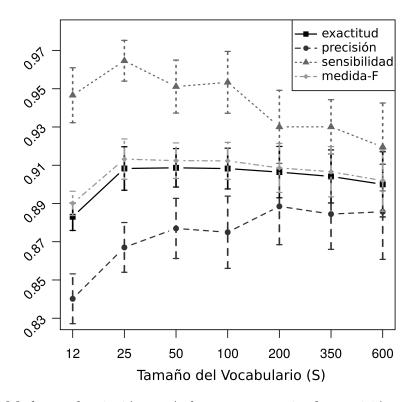


Figura 4.3: Media y desviación estándar para exactitud, precisión, sensibilidad y medida-F con tamaños de vocabulario $S = \{12, 25, 50, 100, 200, 350, 600\}$. Las barras verticales representan los valores de desviación estándar. Estos valores son obtenidos sobre 10 clasificadores entrenados para diferentes conjuntos de entrenamiento balanceados con R = 1.

\overline{S}	Exactitud		Precisión		Sensibilidad		Medida-F	
12	0,883	(0,007)	0,84	(0,013)	0,947	(0,014)	0,89	$\overline{(0,006)}$
25	0,908	(0,011)	$0,\!867$	(0,013)	0,965	(0,011)	0,913	(0,011)
50	0,909	(0,01)	0,877	(0,016)	0,951	(0,014)	0,912	(0,009)
100	0,908	(0,011)	0,875	(0,019)	0,953	(0.016)	0,912	(0,01)
200	0,906	(0,013)	$0,\!888$	(0,02)	0,93	(0,019)	0,909	(0,013)
350	0,904	(0,014)	0,884	(0.018)	0,93	(0.014)	0,907	(0,013)
600	0,9	(0,017)	0,886	(0,025)	0,92	(0,023)	0,902	(0,016)

Tabla 4.1: Media y desviación estándar (en paréntesis y cursiva) para exactitud, precisión, sensibilidad y medida-F con tamaños de vocabulario $S = \{12, 25, 50, 100, 200, 350, 600\}$. Los números en negrita corresponden al valor de media más alto para cada medida. Estos valores son obtenidos sobre 10 clasificadores entrenados para diferentes conjuntos de entrenamiento balanceados con R = 1.

de rendimiento obtenidas para los 10 clasificadores entrenados para cada S, sobre el mismo conjunto de prueba. En la figura, la desviación estándar está representada por las barras verticales, mientras que en la tabla se ha resaltado en cursiva y paréntesis. Para el vocabulario con S=25, el enfoque propuesto logra una media y desviación

estándar para exactitud de 0,908 y (0.011), con una medida-F de 0,913 y (0.011), sensibilidad de 0,965 y (0.011) y precisión de 0,867 y (0.013). El análisis de todos los tamaños de vocabulario, demuestra que S=25 consigue la segunda peor precisión mientras que logra los mejores valores de sensibilidad y medida-F. El caso S=12 registró los peores valores de exactitud, precisión y medida-F. Para los casos $S=\{200,350,600\}$, alcanzan resultados similares entre sí, pero en comparación con S=25, solamente para precisión logran resultados ligeramente superiores. Por último, para los casos $S=\{50,100\}$, sus resultados difieren ligeramente en comparación con S=25: mientras que los primeros obtienen resultados de precisión superior en aproximadamente 0,01, su sensibilidad está por debajo en aproximadamente 0,012. Sin embargo, en términos de significación estadística, no hay pruebas suficientes para asegurar que un tamaño del vocabulario es mejor que otro, ya que la $media\pm desviación$ estándar de la medida-F para diferentes valores de S se superponen, como puede verse en la figura 4.3.

Los resultados presentados anteriormente se calculan sobre la base de diferentes conjuntos de entrenamiento balanceados con R=1. Sin embargo, también se evaluó el rendimiento del clasificador para conjuntos de entrenamiento equilibrados con $R=\{2,4,8,16\}$. Para esto, se realizaron 10 repeticiones completas de entrenamiento y prueba para cada combinación de valores de S y R, siempre sobre el mismo conjunto de prueba, reportando la media y desviación estándar de medida-F. Estos resultados se muestran en la figura 4.4, donde las barras representan el valor medio y los segmentos verticales muestran la desviación estándar. Por cada S la figura muestra un grupo de barras que corresponde a cada R a fin de facilitar la comparación de los resultados. Como muestra la figura, para todos los valores de R siempre que $S \neq 12$ y $S \neq 600$ no existen diferencias importantes de medida-F. Gráficamente se puede verificar que todos los segmentos de $media\pm desviación$ estándar presentan un cierto grado de solapamiento. Esto demuestra que R>1 no es útil para mejorar el rendimiento del clasificador, a pesar de que el sub-muestreo reduce de manera drástica la cantidad de elementos de clase no-yema, con 67 de 68 elementos desechados.

4.3.1. Evaluación de clasificador sobre los parches no-yema

En este apartado se presenta un análisis más minucioso sobre los parches de clase no-yema, ya que estos pueden ser catalogados en diferentes sub-categorías fácilmente distinguibles en la escena. Además, estas sub-categorías pueden presentar diferentes niveles de dificultad durante la clasificación para ser distinguidos de los parches yema. Por lo tanto, se ha diseñado un experimento para evaluar el desempeño del clasificador contra estas sub-categorías. Para esto, se agruparon manualmente los parches no-yema

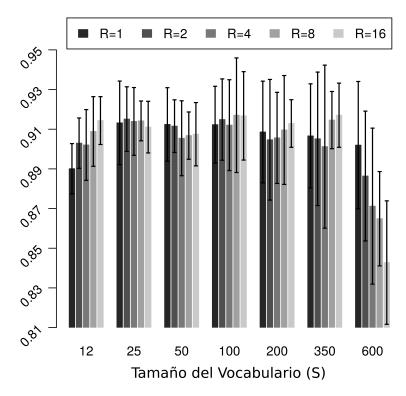


Figura 4.4: F-measure para diferentes conjuntos de entrenamiento obtenidos con $R = \{1, 2, 4, 8, 16\}$ y tamaño de vocabulario $S = \{12, 25, 50, 100, 200, 350, 600\}$. Los valores mostrados corresponden a la media y desviación estándar alcanzados por 10 clasificadores entrenados con diferentes conjuntos de entrenamiento para cada combinación de S y R.

en las siguientes 10 subcategorías según su contenido (con ejemplos que se muestran en la figura 4.5): (a) fuera de foco: parches totalmente borrosos; (b) borde de rama: muestran claramente el borde de una rama; (c) interno rama: contienen solo píxeles de rama; (d) alambre: contienen una parte de un alambre metálico, típico en viñedos; (e) zarcillo: muestran zarcillos; (f) tronco con corteza: muestran parcial o totalmente el tronco principal de la planta, normalmente cubierto con la corteza; (g) hoja seca: muestra parcial o totalmente hojas secas, típico durante la temporada de invierno; (h) racimo seco: muestra total o parcialmente racimos de uvas secos todavía colgando en la planta; (i) cercanía de la yema: parches de las zonas cercanas a la yema, presentando diferentes irregularidades de textura y otras sub-categorías o combinaciones de ellas; y (j) nudo: muestra un típico nudo de madera (i.e. área ligeramente ensanchada en el brote, donde se unen las hojas) o parte de él.

Para evaluar la calidad del clasificador para este nuevo conjunto de pruebas, se ha utilizado los 10 clasificadores entrenados durante los experimentos previos para todos los valores de S con R=1, cuyo corpus de entrenamiento contiene estas diez subcategorías con su representatividad natural en ambientes naturales. Para cuantificar

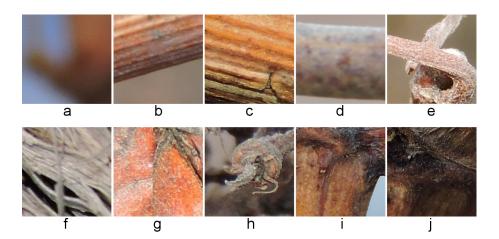


Figura 4.5: Ejemplos de subcategorías de los parches no-yema extraídos del corpus de prueba: (a) fuera de foco; (b) borde de rama; (c) interno rama; (d) alambre; (e) zarcillo; (f) tronco con corteza; (g) hoja seca; (h) racimo seco; (i) cercanía de la yema; y (j) nudo.

Subcategoría	12	25	50	100	200	350	600
Fuera de foco	0,999	1	0,999	1	1	1	0,917
Borde de rama	0,96	0,973	0,981	0,977	0,972	0,982	0,967
Interno rama	0,945	0,935	0,948	0,956	0,963	0,965	0,953
Alambre	0,917	0,945	0,935	0,935	0,965	0,961	0,941
Zarcillo	0,954	0,967	0,964	0,957	0,961	0,96	0,954
Tronco con corteza	0,888	0,892	0,91	0,904	0,895	0,908	0,887
Hoja seca	0,793	0,793	0,806	0,804	0,834	0,829	0,815
Racimo seco	0,674	0,716	0,779	0,786	0,801	0,798	0,759
Cercanía de la yema	0,634	0,582	0,655	0,695	0,734	0,713	0,723
Nudo	0,57	0,61	0,659	0,685	0,683	0,668	0,661

Tabla 4.2: Proporción de parches no-yema clasificados correctamente (i.e. sensibilidad de clase no-yema) sobre cada una de las diez sub-categorías, para diferentes tamaños de vocabulario. Los valores mostrados corresponden a la media obtenida sobre 10 clasificadores entrenados para diferentes conjuntos de entrenamiento balanceados con R=1.

los errores de cada subcategoría, se reporta la proporción de parches en cada una que fueron clasificados correctamente, i.e. el número vn/(vn+fp) (siendo los fp los clasificados erróneamente como yema), que corresponde formalmente a la sensibilidad de la clase no-yema. La tabla 4.2 muestra estas cantidades (sensibilidad de la clase no-yema) para las diez subcategorías, y para diferentes tamaños de vocabulario, dispuestas en orden decreciente según su valor de sensibilidad de la clase no-yema para la columna S=350, la que en mayormente presenta los mejores resultados. Las categorías más difíciles para cualquier vocabulario fueron: nudos, con una sensibilidad de no-yema entre 0,58 y 0,68, i.e. entre un 42 % y un 32 % de los parches de nudo se clasificaron

incorrectamente como yemas; cercanía de la yema, con sensibilidad de no-yema entre 0.57 y 0.73, i.e. entre un 43 % y un 27 % de los parches en las cercanías de la yema fueron clasificadas incorrectamente como yema; seguido de racimo seco, hoja seca y tronco con corteza, todos con una sensibilidad de no-yema entre 0.798 y 0.909 para S=350 (considerado el mejor caso) y entre 0.674 y 0.888 para S=12 (considerado el peor caso). Todas las demás categorías muestran valores de sensibilidad de no-yema superiores a 0.95 para la mayoría de los valores de S.

4.3.2. Evaluación del clasificador para parches típicamente extraídos de un proceso de escaneo-ventana

En los experimentos anteriores, el clasificador propuesto fue probado sobre un corpus que contiene parches de yema muy especiales, poco realistas, que circunscriben las yemas. En el siguiente experimento se evalúa el clasificador sobre un corpus de prueba más realista, que contiene parches generados típicamente por un algoritmo escaneo-ventana, el cual produce, con muy alta probabilidad, parches que están lejos de circunscribir una yema. Para generar estos parches yema-realista, se implementó un programa que toma cada parche yema en el conjunto de prueba original, y añade o elimina porciones de píxeles en las fronteras de parche. Para lograr esto, el programa utiliza la información de la máscara y geometría del parche yema original, y modifica aleatoriamente la escala y/o el desplazamiento. Para cada parche yema en el conjunto de prueba, el programa produce varios nuevos parches yema-realista con diferentes configuraciones de escala y desplazamiento (véase la figura 4.6). Para cada parche yema-realista, el número de píxeles de cada clase se puede obtener usando la máscara del parche yema original. Utilizando la información de la cantidad de píxeles yema que se mantuvieron, y de los píxeles no-yema añadidos o descartados, se pueden agrupar los nuevos parches yema-realista mediante dos cantidades: píxeles-yema-conservados que mide el porcentaje de píxeles yema que se mantienen en el nuevo parche, y píxelesyema-relativos que mide la proporción de píxeles yema en todo el parche. Dada la forma rectangular del parche y la forma irregular de la yema, algunas combinaciones de estas proporciones fueron imposibles de obtener, como por ejemplo 100 % de píxeles-yema-conservados y 100 % de píxeles-yema-relativos, ya que un parche de estas características debería contener todos los píxeles de una yema y solo esos píxeles. Además, ciertas combinaciones fueron, aunque no imposible, más difíciles de obtener que otras, y como resultado sólo unos pocos ejemplos fueron generados por el script.

A continuación se presentan los resultados del error de generalización alcanzado por el clasificador para este corpus de parche más complejo y realista. El espacio definido



Figura 4.6: Izquierda: un parche con la yema circunscrita. Derecha: cuatro parches yema-realista obtenidos a partir del parche de la izquierda. La primera de ellas tiene claramente $100\,\%$ de píxeles-yema-conservados, pero un valor pequeño de píxeles-yema-relativos. La segunda presenta un valor inferior para píxeles-yema-conservados ya que algunos no se muestran, y un valor mayor para píxeles-yema-relativos, ya que se acerca a la yema. La tercera va en la misma dirección, con un valor aún más bajo para píxeles-yema-conservados y mayor valor para píxeles-yema-relativos. El último es un caso con valor bajo para píxeles-yema-conservados, pero un valor de $100\,\%$ para píxeles-yema-relativos, ya que sólo existen píxeles yema en el parche.

por los atributos píxeles-yema-conservados y píxeles-yema-relativos fue discretizado en una grilla regular de 10×10 . Para cada celda de la grilla se generaron 4 parches yema-realista de cada parche yema original, resultando en $133 \times 4 = 532$ parches yemarealista por celda. Dado que todos los parches yema-realista se etiquetaron como yema, sólo tiene sentido evaluar la sensibilidad de clase yema, i.e. la proporción de parches yema que fueron clasificados correctamente como yema, en virtud de las diferentes perturbaciones del parche. Para evaluar la calidad del clasificador sobre este nuevo conjunto de prueba, se utilizaron los 10 clasificadores entrenados con S=25 y R=1durante los experimentos anteriores. Esta elección de S se justifica por el hecho que para los primeros experimentos con R=1, el caso de S=25 entregó las mejores medidas de rendimiento. La figura 4.7 muestra estos resultados resumidos en un mapa de calor en el espacio discretizado por la grilla 10×10 , donde cada celda de la grilla muestra el valor medio de sensibilidad alcanzado por el clasificador para el subconjunto de parches que caen en el rango definido por la celda. En la escala de colores, el gris oscuro es el valor más bajo y el blanco el más alto, i.e. el mejor. Como ya se explicó, algunas combinaciones fueron difíciles o incluso imposibles de alcanzar, alcanzando la cantidad de 532 parches yema-realista en solo 84 de las 100 celdas del mapa de calor. Los restantes celdas, i.e. aquellas para las que no se logró generar una cantidad suficiente de parches yema-realista, se desecharon y fueron coloreadas en negro en el mapa de calor. Como se puede leer en la figura 4.7, las celdas en el rango (20; 80] % de píxeles-yema-relativos y (60; 100) % de píxeles-yema-conservados tiene una sensibilidad medio mayor o igual a 0,89, lo que significa que el clasificador distingue a 9 de cada 10 parches que tienen al menos un 60 % de los píxeles de la yema, y estos píxeles representan entre un 20 % y un 80 % del total del píxeles del parche. Por último,

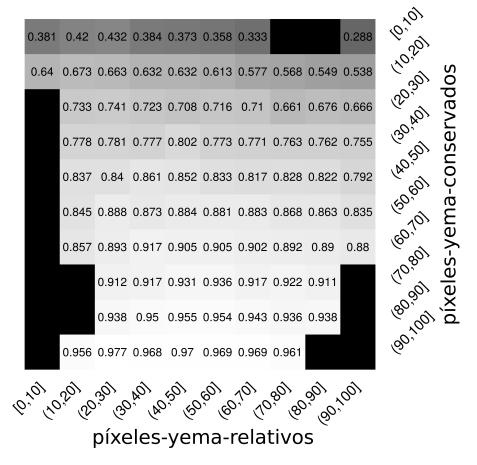


Figura 4.7: Mapa de calor para la sensibilidad de la clase yema en el espacio definido por $p\'{ixeles-yema-conservados}$, i.e proporción de píxeles de la yema original que se han conservado, y $p\'{ixeles-yema-relativos}$, i.e. proporción de píxeles yema en el parche. El espacio está discretizado en una grilla regular de 10×10 . Los valores de cada celda corresponden a la media de la sensibilidad obtenida sobre el corpus de parches yema-realista, para los 10 clasificadores entrenados con S=25 y R=1 durante los experimentos anteriores. Cada celda del mapa se calcula sobre 532 parches, obtenidos mediante cuatro variaciones para cada parche del corpus original de yemas circunscritas.

si el análisis se limita a la dimensión del atributo píxeles-yema-relativos en el rango (20; 80] %, se observa que a medida que los valores de rango de este atributo disminuyen (lo que implica que el tamaño del parche aumenta en relación con el tamaño de la yema), la sensibilidad muestra una leve tendencia a aumentar, alcanzando su valor máximo de 0,977 para los parches con píxeles-yema-conservados en el rango (90; 100] % (fila inferior) y (20; 30] % de píxeles-yema-relativos.

4.3.3. Análisis para los histogramas producidos por BoF

En este apartado se discuten las características más importantes de los histogramas generados por el algoritmo BoF, poniendo de relieve las rasgos visuales más relevantes de los parches que pertenecen a la clase yema y no-yema. Para simplificar la complejidad del análisis se presentan los resultados para el vocabulario de tamaño 12. Por otra parte, para el análisis se han filtrado todos los parches que obtuvieron una clasificación débil, considerando sólo aquellos parches del corpus que se clasificaron en sus respectivas clases con una probabilidad superior a 0,85, con este umbral elegido subjetivamente por los autores. Este filtrado resultó en 49 histogramas seleccionados para la clase yema, y 114 para la clase no-yema. Por último, todos los histogramas de una clase se promediaron para obtener dos histogramas promedio, uno por clase, que se muestran en la figura 4.8. A partir de esta figura se puede llegar a dos conclusiones principales: para la clase yema todas las palabras visuales (contenedores grises oscuros) tienen aproximadamente la misma expresividad (i.e. su valor), con frecuencias relativas que van entre 0,05 y 0,10, siendo las palabras 3, 9, 10, 11 y 12 las que muestran los mayores valores; mientras que el histograma para la clase no-yema (contenedores de color gris claro) muestra una mayor variabilidad entre sus palabras, con valores que oscilan entre aproximadamente 0,03 y 0,20, con las palabras 1, 3 y 6 las de mayor expresividad. También es evidente a partir de la figura que, para ambas clases, cada palabra tiene algo de expresividad no despreciable en sus histogramas. Este hecho puede explicarse por dos factores: (i) como se menciona en la sección 4.2.1, es inevitable la presencia de píxeles no-yema en los parches yema (principalmente en las esquinas), contribuyendo potencialmente algunos puntos de interés no-yema en los histogramas

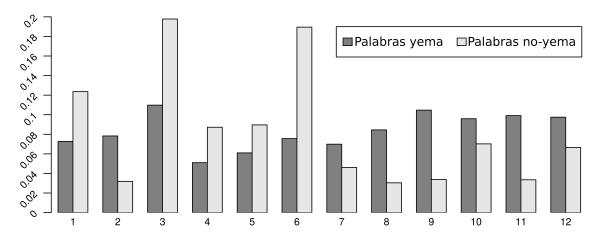


Figura 4.8: Promedio entre los histogramas de parches yema (gris oscuro), y no-yema (gris claro), sobre aquellos parches clasificadas en su clase con una probabilidad mayor a 0,85, para el caso de vocabulario con 12 palabras. Para cada palabra, los contenedores de cada clase se muestran lado a lado para facilitar la comparación de los histogramas.

de yema; y (ii) teniendo en cuenta las características biológicas y fisiológicas de las yemas y las otras partes de la vid (como ser brotes, nudos, hojas, racimos secos, y más), se puede esperar que los parches yema y no-yema compartan algunos rasgos visuales. Si bien este análisis está lejos de ser suficiente para justificar completamente los resultados de 0,965 para sensibilidad y 0,867 de precisión obtenidos por el clasificador MVS, al menos proporciona una aproximación para justificar empíricamente el desempeño del clasificador sobre algunos parches particulares (i.e. aquellos que tienen probabilidad de clase mayor a 0,85).

Al realizar un análisis más profundo, es posible preguntarse si las características visuales de las palabras pueden explicar las diferencias entre los histogramas promedios de cada clase, en particular, el hecho de que las palabras 2, 4, 5, 7, 8, 9, 10, 11 y 12 se expresan menos que las palabras 1, 3 y 6 en la clase no-yema. Al hacer esto, se busca confirmar que, efectivamente, ciertos rasgos visuales se expresan más en ciertas palabras que en otras. A continuación se presenta un análisis de la distribución espacial de los puntos de interés (posición y tamaño) sobre cada una de las 12 palabras del histograma, para un solo caso de una imagen de yema seleccionada del corpus de prueba, tal como se muestra en la figura 4.9. Si bien es claro que este análisis está lejos de ser exhaustivo, se presenta fundamentalmente como un ejemplo ilustrativo. Para realizar el análisis se examinó la yema presentada en la imagen 4.9a, a fin de reconocer rasgos visuales típicos de las yemas resumidos por la máscara de la figura 4.9b: (i) el fondo se muestra en blanco, y aunque no forma parte de una yema, generalmente se presenta alrededor de la yema sobre las esquinas del parche; (ii) la corteza de la cáscara, típica de las yemas se muestra en gris claro, con la etiqueta A; (iii) la pelusa de la cáscara, también típico de las yemas, se muestra en gris medio con la etiqueta B; (iv) las nervaduras típicas de las yemas, que se muestran con líneas negras continuas; y (v) la corteza oscura mostrada en color gris oscuro con etiqueta C, típico de las yemas, pero también típica de la corteza del tronco de la vid.

Como se muestra en el siguiente análisis, los rasgos visuales (ii), (iii) y (iv), parecen ser más típicos de las yemas y prácticamente no se dan en las palabras visuales 1, 3 y 6, en concordancia con la figura 4.8, donde se observa que estas palabras se expresan más en parches de tipo no-yema. Por otro lado, los rasgos visuales (v) se dan en todas las palabras, incluyendo las palabras 1, 3 y 6. Esto demuestra, al menos para esta imagen de ejemplo, cómo los patrones de expresividad mostrados en el histograma promedio de la figura 4.8 pueden ser explicados por los patrones de expresividad según los rasgos visuales típicos de las yemas, y las que uno podría esperar ver tanto en las yemas como en otras partes de la planta de vid. Este argumento se basa en la figura 4.9c, la cual presenta una imagen para cada una de las 12 palabras del vocabulario, donde

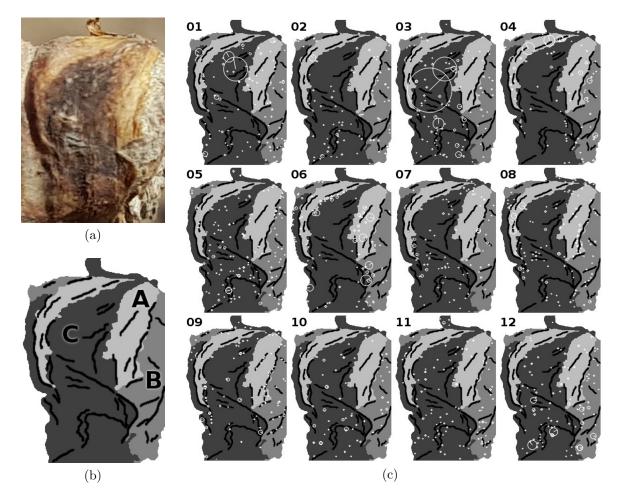


Figura 4.9: Distribución espacial (posición y tamaño) de los puntos de interés para un parche de yema seleccionado entre el conjunto de datos de prueba (a), agrupadas en 12 imágenes (c), una por cada palabra visual en el vocabulario. Cada punto de interés es mostrado sólo en la imagen correspondiente a su palabra más cercana. Cada una de las 12 imágenes muestra regiones de la yema con rasgos visuales típicos de las yemas, según fueron etiquetados en (b): (i) el fondo se muestra en blanco; (ii) la corteza de la cáscara se muestra en gris claro con etiqueta A; (iii) la pelusa de la cáscara se muestra en gris medio con etiqueta B; (iv) las nervaduras se muestran con líneas negras continuas; y (v) la corteza oscura se muestra en color gris oscuro con etiqueta C.

se muestra la partición de los rasgos visuales con los puntos de interés de la palabra superpuestos en su posición espacial y su correspondiente círculo que representa su tamaño.

Después de realizar una cuidadosa inspección de la figura 4.9c es posible encontrar varios patrones visuales. Primero se analiza la región A, la cual aparece en dos partes de la imagen, i.e. arriba izquierda y arriba derecha. Respecto a la parte superior izquierda, se puede observar que las palabras 4, 6 y 8 son las únicas que muestran alguna expresividad (i.e. puntos de interés) en la región, con algunos puntos lo suficientemente

grandes como para destacar su diámetro, donde la palabra 6 en su mayoría muestra puntos en los bordes de la región. Todas las demás palabras muestran muy poca expresividad en esta región, sin puntos en las palabras 10 y 11. En el caso de la región A superior derecha, se expresa en gran medida en las palabras 1, 4, 6, 8, 9 y 12, donde los puntos de las palabras 8 y 9 particularmente se dan sobre nervaduras. En general, la región A tiene muy poca expresividad en las palabras 2, 3, 5 y 10. Por otro lado, la región B también aparece en dos partes distintivas de la imagen: en la parte superior izquierda y a la derecha. La parte superior izquierda se expresa sobre todo en las palabras 2, 3, 6 y 9, con la mayoría de los puntos de interés situados en los bordes para las palabras 3 y 6, y prácticamente ninguna expresividad en las palabras 1, 7, 10, 11 y 12. Para el caso de las nervaduras (líneas negras continuas), el caso más característico es el de las palabras 8 y 9. Estas palabras presentan puntos de interés principalmente sobre nervaduras, tanto en la región B de la derecha, las dos regiones de A, como también en algunas nervaduras de la región C. Además, la nervadura que se muestra casi en ángulo recto sobre la parte baja de la región C, se expresa claramente en la palabra 12, y un poco menos en las palabras 2 y 3. Por último, se puede observar que las nervaduras, un rasgo visual considerado típico de la yema, se expresan muy poco en las palabras 1, 3 y 6, en concordancia con el histograma promedio que muestra más expresividad en estas palabras para los parches no-yema. Por último, se realiza el análisis para el rasgo visual de corteza oscura, etiquetada como C. Se distingue la parte media y media alta con la parte baja. En la parte media y media alta se puede ver que las palabras 1 y 3 muestran un gran punto de interés sobre una rugosidad que aparece en la corteza, que normalmente se observa también en las cortezas de troncos y nudos. También se puede ver que la palabra 6 no muestra prácticamente ninguna expresividad para este rasgo visual. Sobre la parte inferior de la región C se puede ver que la expresividad es mayor en las palabras 2, 3, 5, 7 y 12 y más baja en las palabras 8, 10 y 11, y prácticamente inexistente en todos los demás, incluyendo palabras 1 y 6. A partir de este análisis se puede concluir para las palabras 1, 3 y 6 son típicas de los parches no-yema, que la palabra 3 expresa en su mayoría características visuales de la corteza, que aparece también en los nudos y yemas, y la palabra 6 expresa en su mayoría fronteras. Además, se puede observar que los rasgos típicos de las yemas como pelusa, cáscara y nervaduras se expresan muy poco en las palabras 1, 3 y 6.

A fin de proporcionar un informe más detallado de estos resultados, se ha compartido la distribución espacial de los puntos de interés sobre cada palabra para todos los parches del corpus de imágenes de prueba (incluyendo parches yema y no-yema) en la url http://dharma.frm.utn.edu.ar/vise/bc/.

4.4. Discusión

En esta sección se discuten los alcances y limitaciones de los resultados obtenidos en la evaluación del algoritmo de clasificación propuesto, interpretando sus resultados en el contexto de los algoritmos de detección escaneo-ventana, e incluyendo posibles maneras de mejorarlo y eventuales trabajos futuros.

La especificación del tamaño del vocabulario es una decisión de diseño que se establece durante la ejecución del algoritmo. Los resultados presentados en este trabajo indican que tamaños de vocabulario entre 25 y 100 son adecuados para el problema de clasificación de parches que contienen o no yemas. Para S=25, el enfoque propuesto logró una sensibilidad de 0,965 y una precisión de 0,867 sobre un corpus de prueba con parches que circunscriben la yema. Puesto que estos parches de yema circunscritos son situaciones atípicas en los algoritmos escaneo-ventana, se creó un nuevo conjunto de prueba de parches yema más realistas, que relajan la condición de circunscripción. En este nuevo escenario, el clasificador fue robusto a parches que contienen al menos un 60% de los píxeles de la yema original (i.e. casi la mitad de la yema fue excluida en el parche), y la proporción de píxeles yema en el parche fue entre un 20 % y un 80 % (i.e. entre el 20 % y el 80 % de los píxeles en el parche fueron píxeles no-yema), logrando una sensibilidad por encima de 0,89 en todos los casos, con precisión sin cambios de 0,867. En particular, el mejor valor de sensibilidad fue 0,977 para los parches que contienen entre un 90 % y un 100 % de píxeles yemas y estos píxeles representar entre un 20 % y un 30 % del parche, i.e. parches de tres a cinco veces más grande que la yema. Este comportamiento puede explicarse en parte por el análisis presentado en el apartado 4.3.1, sobre la dificultad en la clasificación de las diez subcategorías de la clase no-yema. El clasificador comete los errores más grandes para los parches de tipo nudo y cercanía de la yema, en ambos casos parches que se encuentran en las proximidades de la yema (ya que los nudos siempre se encuentran muy cerca de las yemas de la vid). Al aumentar el tamaño del parche en relación con la yema, se incluye información visual de categorías no-yema en los parches yema, y el clasificador se hace más robusto. Tal comportamiento es coherente con el hecho de que los parches yema del conjunto de entrenamiento contienen en promedio un 25 % de píxeles no-yema, y con varios trabajo de la literatura, donde se muestra que incluir información espacial del contexto del objeto en algoritmos de clasificación y detección puede aumentar su robustez [51, 36, 165, 30, 67]. Sin embargo, en los algoritmos de detección escaneoventana el uso de parches de tamaño superior a una yema puede provocar errores por desplazamiento en la ubicación real de la yema.

Por otra parte, un análisis más detallado sobre las características de la clase noyema, muestra que hay partes del entorno de la yema que presentan diferentes grados

de dificultad para el clasificador. En particular, los parches que contienen nudos o regiones cercanas a la yema fueron parches donde se vio afectada negativamente la calidad clasificador. En el caso de las zonas cercanas a la yema, este comportamiento puede ser explicado por el hecho de que los parches yema utilizados en la etapa de entrenamiento son regiones rectangulares que contienen pequeñas porciones de píxeles no-yema (25 % en promedio). Por esto, se espera que las características SIFT de bajo nivel calculadas sobre estas regiones para un parche yema, se expresen en el descriptor BoF del parche, confundiendo el clasificador. Los parches con nudos, presentan características visuales muy similares a las características visuales que se producen en la yema. Una forma de mejorar la robustez del clasificador en estas situaciones es utilizar otros tipos de características de bajo nivel, como ser la forma [167]. Sin embargo, las yemas se desarrollan siempre junto a los nudos de la vid, por lo que los nudos pueden aumentar la eficacia en la clasificación y detección de yemas, como se sugiere en la discusión anterior sobre la información espacial del contexto del objeto. Existen otras categorías, como ser racimo seco, hoja seca y tronco con corteza, que muestran resultados de calidad media, con una sensibilidad de no-yema que va desde 0.67 a 0.91 aproximadamente, para el todos los valores de S. A fin de evaluar el impacto de estos errores de clasificación, se comparó la representatividad de parches para las subcategorías de no-yema con la representatividad de parches para las yemas, en el entorno natural fotografiado. Estos valores se calcularon como la proporción de parches de estas sub-categorías con respecto al número total de parches. Se obtuvo una representatividad de de 1,35 % para las yemas; 0,35 % para racimos secos, 0,4 % para hojas secas, y 0,54 % para tronco con corteza. Se puede observar que las tres sub-categorías juntas representan 1,29%, comparable al de las yemas. Esto implica que entre un 9 % y un 33 % de estos parches se clasifican incorrectamente como yema. Todas las sub-categorías restantes presentan una sensibilidad de no-yema por encima de 0,95 para la mayoría de los valores de S. Sin embargo el caso de parches fuera de foco merece un análisis más detallado. Como se muestra en la tabla 4.2 para S=25, la sensibilidad de no-yema de esta categoría es 1, por lo que es claro que el clasificador no tiene problemas para clasificarlos. Estos parches son imágenes borrosas de diferentes elementos de la escena (cielo, tierra, otras plantas), y se deben al hecho de que las imágenes son tomadas enfocando a las yemas en primer plano y con una profundidad focal pequeña (ver figura 4.1). En términos del clasificador, la separación entre un "fondo fuera de foco" y un "primer plano enfocado" es de alguna manera similar al uso de un fondo de color homogéneo, tal como se propone en el trabajo de [167] y [68], en el sentido de que el algoritmo SIFT no produce características de bajo nivel en regiones muy homogéneos o sin textura. En este caso, obtener un fondo fuera

de enfoque no es esencial, pero es una situación que se produce en la mayoría de las imágenes capturadas, debido a la distancia de enfoque a las yemas y la profundidad de campo: cuanto más cerca se captura la yema, menor será la profundidad de campo y más borroso el fondo. En este trabajo, con yemas de al menos 100 píxeles de diámetro, la profundidad de campo fue suficiente para desenfocar el fondo completamente. La principal motivación para el uso de imágenes de estas características no fue difuminar el fondo, sino obtener las yemas en una escala visible para que puedan ser reconocidas por el algoritmo de clasificación que aquí se presenta. En trabajos futuros, se espera empezar a trabajar en un algoritmo de clasificación y detección multi-escala que permite la búsqueda de yemas a partir de imágenes en donde la yema no es claramente visible, i.e. cuando la yema ocupa muy pocos píxeles en la imagen o incluso menos de un pixel.

Desde el punto de vista de un algoritmo de detección escaneo-ventana, la detección de una yema en particular es el resultado de la superposición de parches que fueron clasificados como yema en una región de la imagen. Si la sensibilidad de clasificación disminuye, la cantidad de parches yema mal clasificados aumenta, con el riesgo de que ninguno de los parches que incluyen píxeles de yema se encuentren, lo que resulta en que la yema no sea detectada. Esto demuestra que la sensibilidad de clasificación es una cota inferior de la sensibilidad de detección, ya que en presencia de parches yema mal clasificados todavía hay lugar a una detección exitosa, siempre que algunos parches yema vecinos se clasifiquen correctamente. Por ejemplo, la sensibilidad de detección es al menos de 0,89 para los parches que contienen al menos 60 % de los píxeles yema originales y la proporción de estos píxeles en el parche es mayor a 20 %, como se muestra en las celdas correspondientes al mapa de calor de la figura 4.7. Esto también muestra que la sensibilidad de detección podría mejorarse aún más mediante un algoritmo de detección escaneo-ventana que presenta flexibilidad de movimiento dentro de las celdas del mapa de calor, variando el tamaño y el paso del desplazamiento de la ventana de escaneo. Al elegir valores de píxeles-yema-conservados (afectado principalmente por el desplazamiento) y valores de píxeles-yema-relativos (afectado principalmente por el tamaño), un algoritmo de detección podría proponer un paso y un tamaño de ventana que garantice que en algún momento de la exploración la ventana satisfaga dos condiciones: el valor de píxeles-yema-relativos esté entre un $20\,\%$ y 30 %; y el valor de píxeles-yema-conservados se encuentre entre un 90 % y un 100 %. De esta manera, se puede garantizar una cota mínima para la sensibilidad de detección de yemas de 0,977 (como se muestra en la celda correspondiente en el mapa de calor). Sin embargo, en la práctica esto puede no ser de mucha utilidad, ya que valores de píxeles-yema-relativos dentro de un $20\,\%$ y $30\,\%$ producirían una localización más

bien imprecisa de la yema. Sin embargo, estos resultados pueden ser un buen punto de partida para el algoritmo de detección, que luego podría moverse a la derecha del mapa de calor mediante la reducción del tamaño de la ventana (siempre que sea mayor que una yema), mientras que la exploración garantiza que toda la yema cae dentro de alguna de las ventanas. Este procedimiento, aunque bastante hipotético, muestra que la sensibilidad de detección presenta un límite inferior en el rango de 0,89 a 0,977.

Similar a la sensibilidad, una baja precisión significa que algunos parches no-yema se clasifican como yemas, lo que resultaría en la detección de yemas donde no las hay. El clasificador presentado logra una precisión de 0,86, i.e. un 14 % de parches no-yema en el corpus de prueba ha dado lugar a falsos positivos, es decir, fueron incorrectamente clasificados como parches yema. En la práctica, el número de parches no-yema puede supera en gran medida a los de clase yema, por lo que este resultado puede ser grave dependiendo de la aplicación. Este análisis, sin embargo, ignora la evaluación de las subcategorías de la clase no-yema, que se muestran en la tabla 4.2. El complemento de los valores de la tabla es la proporción de falsos positivos en cada sub-categoría. Dejando de lado las subcategorías poco representadas (i.e. hoja seca, racimo seco, y tronco con corteza) y aquellas cercanas a la yema (nudo y cercanía de la yema), se demuestra que todas las demás subcategorías tienen una proporción de falsos positivos menor a 8 % para S=12 (el peor caso), y menor a 4 % para S=350 (el mejor caso), valores mucho más bajos que el 14 % original. Haciendo el mismo análisis ahora para las subcategorías poco representadas, se verifica que la proporción de falsos positivos es de 9 % a 33 %, pero todos estos parches juntos tienen una representatividad poco menor a la de la yema (1,29 % frente a 1,35 %, respectivamente). Finalmente, los parches de las sub-categorías nudo y cercanía de la yema, que en principio presentan resultados muy bajos en la proporción de falsos positivos (43 % para nudo, con S=12), en realidad podrían ayudar a un algoritmo de detección a detectar las cercanías de la yema, como se sugiere en la discusión anterior sobre la información espacial del contexto del objeto. En resumen, en la detección es posible esperar una tasa de falsos positivos de 4% a 8%. Una mejora adicional, aunque más bien especulativa, sugiere que los falsos positivos en el tronco y otras subcategorías que aparecen lejos de las yemas son escasos, y se producen más bien en lugares dispersos. En cambio, en las proximidades de una yema se esperaría una gran superposición de parches verdaderos positivos. Esto sugiere un procedimiento de agregación de parches para la detección que descarta parches aislados clasificados como yema.

Por último, el algoritmo de clasificación de imágenes de yema es robusto cuando el entrenamiento se lleva a cabo para conjuntos de entrenamiento balanceados de acuerdo a la clase yema. Se evaluaron varios escenarios para conjuntos de entrenamiento balanceados por submuestreo de la clase no-yema y sobre-muestreo de la clase yema. Los resultados reportados para diferentes frecuencias de muestreo R>1 no presentaron tendencias de mejora. Si la capacidad de cómputo durante la etapa de entrenamiento es un problema, la elección de R=1 es la más conveniente, debido a que el cálculo de las características SIFT y los descriptores BoF, y el entrenamiento del clasificador MVS, crece proporcionalmente con el tamaño del conjunto de entrenamiento. Por otra parte, el tiempo de CPU incurrido para clasificar un nuevo parche fue de 200 milisegundos aproximadamente (sobre un procesador Intel Core i7-4770 de 3.40GHz). Si bien esta cantidad puede parecer elevada, en este trabajo no se tomó en cuenta una implementación eficiente de los algoritmos, por lo que las características SIFT se calcularon para cada parche individual. En la práctica, durante la ejecución de un algoritmo de detección escaneo-ventana, se podría reducir el tiempo de clasificación precalculando las características SIFT para toda la imagen, y luego extraer las características que pertenecen a cada parche, ahorrando importantes cantidades de cómputo.

4.5. Conclusiones

En este capítulo se ha presentado un algoritmo de clasificación de imágenes de yemas de vid a partir de imágenes tomadas viñedos en su ambiente natural. Este algoritmo implica el uso de características invariantes en escala transformada para el cálculo de características de bajo nivel, bolsa de características para la construcción de un descriptor de imagen, y máquinas de vector soporte para el entrenamiento de un clasificador. Los requisitos del algoritmo de clasificación están especificados para ser utilizado en imágenes producidas normalmente por algoritmos de detección escaneoventana. Se justifica la importancia de la detección de la yema a través de sus posibles aplicaciones, como ser sistemas autónomos de poda, fenotipado de plantas de vid, y reconstrucción 3D de la estructura de la vid. Además, se argumentó que los trabajos del estado del arte para la detección de yemas presentan varias limitaciones: (i) utilización de fondo artificial; (ii) iluminación controlada en interiores; (iii) necesidad de interacción con el usuario; (iv) la detección de yemas en etapa muy avanzada de desarrollo. En contraste, el método presentado en este trabajo resulta en una contribución al área de aplicación, ya que se espera que el enfoque de clasificación propuesto, embebido en un algoritmo de detección escaneo-ventana, sea robusto para escenarios más realistas, es decir, al aire libre, en condiciones de campo naturales, en invierno, sin fondos artificiales, y con los requisitos mínimos de equipamiento.

El enfoque propuesto logra una sensibilidad de clasificación superior a 0,89 en

parches que contienen al menos un 60 % de los píxeles de la yema original, donde la proporción de píxeles yema en el parche es mayor a 20 %, y la yema tiene al menos 100 píxeles de diámetro. En particular, el mejor valor de sensibilidad de clasificación fue de 0,977, para parches que contienen entre 90 % y un 100 % de los píxeles yema, y estos píxeles representan entre 20 % y un 30 % del parche, i.e. parches de tres a cinco veces más grande que la yema. Un algoritmo de detección escaneo-ventana podría proponer un esquema para elegir el desplazamiento y el tamaño de la ventana que garantice que en algún momento de la exploración, la ventana satisface parches con valores de píxeles-yema-conservados de al menos 60 %, y valores píxeles-yema-relativos de mayor a 20 %. Tal esquema podría comenzar con un desplazamiento y tamaño de ventana apropiado, y luego manipular estos parámetros, siempre que la exploración garantice que toda la yema cae dentro de alguna ventana. Este procedimiento muestra que 0,89 es una cota inferior y 0,977 es una cota superior, para la sensibilidad de un algoritmo de detección.

En cuanto a los resultados de precisión de clasificación, el clasificador alcanzó un valor de 0,86, i.e. 14% de los parches no-yema en el corpus dieron lugar a falsos positivos, erróneamente clasificados como parches yema. Otra conclusión importante de este trabajo es que para la detección, es posible reducir la tasa de falsos positivos entre un 4% y un 8%, ya que una gran parte de estos errores provienen de parches en las proximidades de la yema (i.e. nudos y cercanías de la yema) y parches de categorías poco representadas (i.e. hojas secas, racimos secos, y tronco con corteza), y no debería producir errores en la detección. Finalmente, en este trabajo se destacan los aspectos de diseño más relevantes para la implementación de un algoritmo de detección de tipo escaneo-ventana para una aplicación real: (i) tamaño y desplazamiento de la ventana; (ii) fallas en la detección por clasificación binaria; (iii) estimación de la posición de la yema en la imagen; y (iv) complejidad computacional de la solución propuesta.

Capítulo 5

Hacia un Algoritmo de Búsqueda Visual Activa para Yemas de Vid

En este capítulo se presentan diversos enfoques para la codificación del conocimiento de sentido común en el desarrollo de un algoritmo de búsqueda visual activa para yemas de una planta de vid, iniciando la búsqueda a partir de vistas capturadas a una distancia tal que las yemas poseen resolución menor a 20×20 píxeles y finalizando cuando se alcanzan vistas a una distancia tal donde las yemas tienen resoluciones entre 80×80 y 150×150 píxeles. La búsqueda avanza permitiendo realizar acercamientos de manera telescópica hacia las zonas donde se espera que encuentren yemas, evitando explorar el resto de escena. Concretamente, se presenta un estudio para diferentes enfoques que permiten planificar qué zonas de la escena explorar sobre vistas capturadas a diferentes distancias de una planta, procurando descartar lo antes posible aquellas zonas donde no se espera que existan yemas, bajo la intuición de que cuanto mayor sea la distancia en la que se descarta una zona sin yema, más eficiente será la búsqueda. El objetivo principal de estos enfoques es mantener la efectividad y la eficiencia de la búsqueda de yemas lo más elevada posible. Una efectividad de búsqueda alta implica minimizar el número de yemas que se pierden por errores en las zonas descartadas. Por otro lado, una eficiencia de búsqueda alta requiere no visitar inútilmente aquellas zonas que efectivamente no contienen yema. Los enfoques son contrastados con los resultados teóricos del enfoque por fuerza bruta, i.e. explorar todas las vistas de manera exhaustiva en la distancia más cercana al plano de las yemas. De esta manera, este enfoque tiene la efectividad más alta posible, dado que la búsqueda no descarta ninguna yema, pero una eficiencia de búsqueda muy baja ya que requiere explorar exhaustivamente toda la escena de manera ineficiente.

5.1. Introducción

Tal como se describe en el capítulo 1, el objetivo general de la investigación en la que se enmarca esta tesis consiste en el desarrollo de algoritmos de visión computacional para la medición y detección autónoma de variables vitícolas. Particularmente, en este capítulo se aborda el problema de búsqueda visual activa (BVA) de yemas de vid a partir de imágenes tomadas en condiciones naturales de campo en temporada de invierno, en el contexto específico del problema de detección autónoma de yemas de una planta de vid. Este problema requiere agentes autónomos inteligentes que tomen decisiones acertadas sobre cómo explorar el complejo entorno vitícola, esperando que el equilibrio entre la efectividad de la detección de yemas y la eficiencia del proceso de búsqueda sea adecuado para el problema. Estos agentes deben contar con: (i) la capacidad de sensar el entorno a través de imágenes o vistas; (ii) la inteligencia para navegar eficientemente el espacio; y (iii) la habilidad para detectar las yemas de una planta. Es importante destacar que los sistemas de BVA se ocupan principalmente de los primeros dos aspectos, mientras que el proceso de detección podría realizarse de manera independiente. Por ejemplo, un agente para la búsqueda de yemas podría comenzar el proceso en vistas donde las yemas no pueden detectarse directamente debido a que ocupan muy pocos píxeles en la imagen (e incluso menos de un píxel). Sin embargo, el algoritmo de BVA puede realizar suposiciones de los lugares del espacio en donde es probable que se encuentre una yema, y a partir de allí conseguir vistas más adecuadas para el proceso de detección. Con esto, no tiene sentido intentar la detección de yemas en todas las vistas que explora el sistema de BVA.

En la práctica, sería posible construir un agente autónomo para detección de yemas equipando un vehículo aéreo no tripulado (VANT) con una cámara RGB y los algoritmos necesarios que aporten la autonomía e inteligencia requerida. Sin embargo, construir un agente de estas características es una tarea compleja que implica el desarrollo de sistemas de navegación, estabilización, ubicación espacial, ahorro de energía, manejo de colisiones y más [151]. Por lo tanto, en este trabajo los resultados se verifican en un ambiente controlado que se abstrae de todos los inconvenientes operacionales y técnicos que se derivan de la puesta en marcha de un VANT en campo. Los detalles de la construcción del ambiente de prueba se posterga hasta la sección 5.2.1. Es importante notar que un algoritmo de BVA no depende de todos estos sistemas para ser evaluado, dado que su objetivo principal es mantener la efectividad y la eficiencia de la búsqueda de yemas lo más elevada posible. Esto implica que se debe maximizar el número de yemas encontradas a la vez que se minimiza el número de vistas capturadas durante todo el proceso. Respecto a la eficiencia de estos algoritmos, se presenta una breve discusión que justifica tomar la cantidad de vistas procesadas como medida de

calidad. En la práctica, realizar el proceso de búsqueda tiene un costo determinado por diferentes factores, entre los que se pueden nombrar: distancia total recorrida, tiempo de cómputo requerido, tiempo de vuelo en movimiento, tiempo de vuelo estabilizado, energía consumida, y más. Sin embargo, es razonable asumir que una menor cantidad de vistas implica un ahorro directo para el tiempo de cómputo requerido y el tiempo de vuelo estabilizado, ya que estos costos se pagan cada vez que se realiza la toma de una nueva vista. Respecto a la distancia total recorrida y el tiempo de vuelo en movimiento (consideradas proporcionales), si bien es posible que una menor cantidad de vistas reduzca sus costos, podría suceder que búsquedas con diferente número de vistas necesiten recorrer la misma distancia (por ejemplo, una búsqueda en línea recta directa al objetivo puede realizarse en dos pasos o en diez). Por otro lado, el orden en el que se pueden visitar exactamente las mismas vistas influye en la distancia total recorrida. Para esto, es evidente que se requiere un componente dentro del sistema de BVA que permita determinar el ordenamiento óptimo para estas vistas. El enfoque presentado aquí no tienen en cuenta ordenamiento, por lo que no tiene sentido reportar distancia recorrida.

Concretamente, en este capítulo se presentan las bases para un algoritmo de búsqueda visual activa (BVA) sobre los requerimientos particulares de la detección autónoma de yemas, como así también diversos enfoques para construir el conocimiento de sentido común requerido durante la planificación de vistas a explorar, tal como se explica en el capítulo 1. Esta búsqueda de yemas se realiza en escenas de plantas de vid cuyo espacio no se puede acomodar en una única vista (o imagen), a menos que el agente se aleje para que el campo de visión de la cámara pueda encuadrar toda la planta. Por otro lado, al alejarse de la planta, las yemas empiezan a tener cada vez menos resolución, perdiendo inevitablemente información visual fundamental para una detección autónoma efectiva. Por lo tanto, para mantener la información visual propia de las yemas es necesario acercarse y tomar por partes toda la planta con imágenes suficientes a una distancia específica, lo que genera una gran cantidad de vistas a procesar, tal como se detalla más adelante en esta sección. Esta disputa entre la distancia que afecta a la resolución de las yemas (con impacto en la efectividad de la detección) y la cantidad de vistas a procesar (con impacto en la eficiencia de la búsqueda) se presenta como un problema de fuerzas opuestas cuando se intenta una solución trivial de detección por fuerza bruta, i.e. capturar todas las vistas a una distancia fija barriendo exhaustivamente la escena completa. Sin embargo, el esquema de BVA propuesto permite encontrar un equilibrio más adecuado entre ambos requerimientos utilizando el conocimiento del sentido común para el problema de búsqueda de yemas. Este conocimiento permite elaborar conjeturas sobre si una vista a una distancia dada contiene o

no yemas, independientemente si la información visual de la yema (i.e. su resolución) es suficiente para la detección o no. De esta manera el algoritmo de planificación de vistas puede tomar la decisión de descartar zonas de la escena en las subsiguientes exploraciones del espacio. Intuitivamente, se espera que cuanto mayor sea la distancia de la vista en la que se descarta una zona sin yema, más eficiente sea la búsqueda.

Para destacar aún más la importancia del algoritmo de BVA en el problema de detección de yemas, a continuación se presentan de forma más detallada el funcionamiento de un hipotético algoritmo de detección de yemas por fuerza bruta. Desde esta perspectiva, sería posible construir un agente autónomo para detección de yemas equipando un VANT con una cámara y la capacidad de cómputo necesaria para ejecutar el algoritmo de clasificación de yemas tal como se propone en el capítulo 4. Dado que el agente no posee información a priori sobre la ubicación de las yemas en la escena, necesita realizar un barrido completo capturando todas la imágenes necesarias para cubrir exhaustivamente el espacio que ocupa una planta. En este esquema es esperable que se mantengan los mismos resultados de precisión y sensibilidad para detección de yemas que se presentaron en el capítulo 4. Sin embargo, este algoritmo de clasificación de yemas tienen un requerimiento particular: las imágenes capturadas deben estar a una distancia tal que las yemas tengan como mínimo una resolución de 100×100 píxeles. Dado que las yemas tienen un diámetro promedio de 4 milímetros, una imagen obtenida mediante una cámara con resolución de 5312×2988 debe ser tomada aproximadamente a unos 12 centímetros de la yema para cumplir con esta exigencia. En contraste, una planta de vid típica con sistema de conducción de tipo espaldero, tal como se muestra en la figura 5.1a, tiene aproximadamente 200 centímetros de ancho por 180 centímetros de alto, y suele desarrollar hasta 150 yemas. Este hecho pone de manifiesto dos problemas que emergen durante la detección autónoma de yemas: (i) para cubrir el espacio completo que ocupa una planta a una distancia muy cercana es necesario tomar una gran cantidad de imágenes; y (ii) existen grandes zonas de la escena que no contienen yemas y que igualmente serán fotografiadas. Para ilustrar estos problemas de manera sencilla, es posible asumir que la estructura de una planta se encuentra exactamente sobre el plano determinado por los alambres de la hilera del viñedo, asumiendo también que todas sus yemas se ubican sobre ese mismo plano. Bajo el requisito de capturar yemas con resolución de 100×100 píxeles, una imagen de 5312 × 2988 capturada a 12 centímetros del plano de las yemas lograría cubrir un área del plano aproximada de 21 × 12 centímetros. Asumiendo además que no hay superposición entre las imágenes que se capturan sobre este plano, se requeriría aproximadamente un total de 150 tomas para abarcar una planta completa a la distancia de 12 centímetros. El esquema presentado en la figura 5.1b ilustra los problemas (i) y

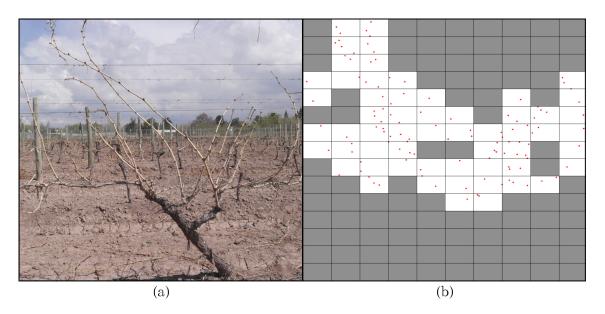


Figura 5.1: (a) Planta de vid típica con sistema de conducción de tipo *espaldero*. (b) Posición y tamaño aproximado de las yemas de la planta (puntos rojos), junto a la grilla de imágenes (o vistas) requeridas para abarcar la totalidad de la planta. Las zonas sombreadas de la grilla indican imágenes que no contienen yemas.

(ii) antes mencionados. En esta figura los puntos en rojo indican la ubicación y tamaño aproximado de las yemas de la planta, junto a una grilla que ejemplifica la región del plano que sería capturada por cada una de las 150 imágenes tomadas para abarcar la totalidad de la planta. Las zonas sombreadas de la grilla indican imágenes que no contienen yemas. Esto implica que un enfoque de fuerza bruta sobre este escenario hipotético, requiere capturar 150 imágenes donde solo 53 contienen efectivamente yemas. Es importante notar, que este problema tiene un impacto aún peor cuando las imágenes son capturadas con un cámara de menor resolución, lo que produciría una grilla mucho más densa, esperando que la cantidad de imágenes sin yema crezca mucho más rápido que la cantidad de imágenes con yemas. Frente a estos problemas es evidente que el enfoque por fuerza bruta presenta serias limitaciones que lo hacen muy ineficiente, aunque su potencial efectividad para detectar todas las yemas de una planta se mantiene muy elevada.

Con la intención de reducir la cantidad de vistas a procesar con el enfoque de fuerza bruta, se podría pensar en aplicar el mismo proceso de barrido exhaustivo a la escena, pero desde una distancia más alejada de la planta. Como ya se observó anteriormente, esta estrategia tiene un impacto negativo en la sensibilidad del algoritmo de clasificación de yemas propuesto en el capítulo 4, dado que la resolución de las yemas se reduce demasiado. Específicamente, el algoritmo de clasificación de yemas del capítulo 4 se ve afectado de la siguiente forma: mayor distancia y menor resolución tienen un

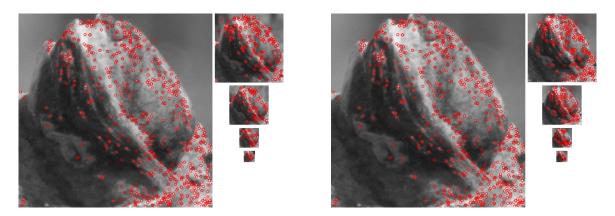


Figura 5.2: Izquierda: puntos claves detectados en parches obtenidos a partir de imágenes capturadas a diferentes distancias de una misma yema. Derecha: puntos claves detectados en parches obtenidos a partir de un cambio en la resolución de la imagen original. Para ambos casos, la imagen de mayor resolución es la misma. En el caso de la derecha, la imagen original fue escalada a las mismas resoluciones de las imágenes tomadas en diferentes distancias: 171×171 , 98×98 , 49×49 y 28×28 .

impacto equivalente sobre el algoritmo SIFT durante el cómputo de puntos de interés y descriptores de una yema (c.f. 4.2.2), al resultar en la misma tendencia de pérdida potencial de ciertos puntos de interés para parches de yema, afectando finalmente el rendimiento del clasificador en el esquema SIFT-BoF-SVM. Este comportamiento se ilustra en la figura 5.2. En la figura de la izquierda se presenta un parche de yema capturado a una distancia tal que su resolución es de 483 × 483, acompañada de cuatro parches más de la misma yema, pero capturados a distancias tal que sus resoluciones se redujeron a 171×171 , 98×98 , 49×49 y 28×28 . En la figura de la derecha, se presenta el mismo parche capturado a la distancia más cercana, acompañado por cuatro parches obtenidos al escalar el parche original a las resoluciones de las imágenes tomadas a diferentes distancias. El proceso de escalado se realizó a través de la reducción directa de la resolución, sin ningún proceso de interpolación entre sus píxeles. Como se puede observar en la figura, la cantidad de puntos de interés detectados en ambos casos disminuye a mayor distancia y menor resolución. Además, es notorio como los puntos de interés se mantienen aproximadamente en las mismas zonas de la vema fotografiada. De esta manera se verifica empíricamente que distancia y resolución tienen un impacto equivalente sobre el algoritmo SIFT, eliminar sistemáticamente información visual relevante de las yemas a medida que se aleja en distancia o se reduce su resolución.

Para ilustrar el problema de pérdida de sensibilidad del clasificador de yemas a medida que se disminuye la cantidad de información visual por alejamiento, se generó un nuevo corpus de parches con yemas capturadas a una distancia aproximada de 12 centímetros, resultando en parches de yema con resoluciones entre 80×80 y 150×150 .

Precisión vs Sensibilidad

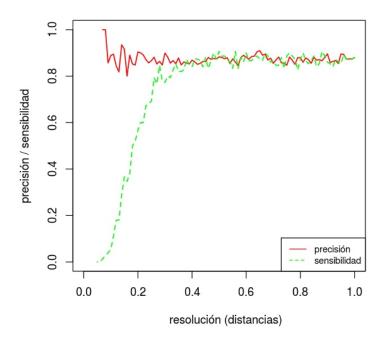


Figura 5.3: Resultados de precisión y sensibilidad alcanzados por el clasificador de yemas para diferentes distancias, según reducciones de la resolución original.

A partir de este nuevo corpus con resoluciones más homogéneas que las del capítulo 4, se separó un conjunto de parches para entrenamiento y otro conjunto para prueba, ambos conjuntos balanceados en la cantidad de elementos de cada clase. Luego se procedió a entrenar nuevamente el algoritmo de clasificación de yemas (en la sección 5.2.3 se dan más detalles sobre el corpus y el entrenamiento del clasificador). A partir de allí se generaron diferentes conjuntos de prueba a partir del conjunto de prueba original, reduciendo sucesivamente la resolución de todos sus parches en un 1 %, hasta llegar a un conjunto de prueba con parches a una resolución del 5% de su tamaño original. El efecto esperado de este cambio de resolución es eliminar sistemáticamente información visual relevante en los parches de yema, de manera similar a lo que ocurre cuando una yema es fotografiada desde distancias cada vez más lejanas. En la figura 5.3 se muestran los resultados de precisión y sensibilidad para la clase yema alcanzados por el clasificador de yemas para diferentes distancias, entendiendo que la distancia aumenta a medida que se reduce la resolución. En la figura se observa que la sensibilidad del clasificador se mantiene relativamente constante aproximadamente hasta una resolución del 50% de la original. A partir de allí a medida que se reduce la resolución la sensibilidad disminuye, con una caída muy pronunciada para resoluciones menores al 30 %. Para la resolución de 5 % la sensibilidad alcanzada es de 0. Por otro lado,

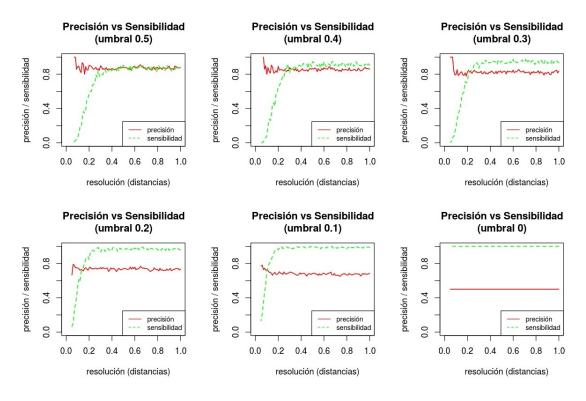


Figura 5.4: Resultados de precisión y sensibilidad alcanzados por el clasificador de yemas para diferentes distancias y diferentes umbrales de clasificación.

la precisión siempre se mantiene por arriba de 0,8, lo cual no muestra evidencia de que la misma se vea afectada por la eliminación sistemática de información visual de la vema. Finalmente, en un último intento por mejorar los resultados de sensibilidad para el clasificador de yemas a distancias muy alejadas, se presentan los resultados de precisión y sensibilidad para diferentes umbrales de clasificación sobre el mismo rango de distancias, esperando que el clasificador de yemas con un umbral menor al utilizado por defecto (i.e. 0,5) compense la pérdida de sensibilidad para distancias mayores. Estos resultados se pueden consultar en la figura 5.4. Las seis gráficas de la figura están ordenadas de mayor a menor umbral de clasificación. Vale la pena notar que la gráfica para umbral de 0,5 es la misma que la gráfica de la figura 5.3, pero se incluye aquí nuevamente a fin de facilitarle al lector el análisis comparativo de todas ellas. Comparando las seis gráficas, si bien se observa que la curva de sensibilidad en general comienza a elevarse cuando se baja el umbral, también es cierto que sin importar el umbral empleado, la sensibilidad cae para las distancias más lejanas (salvo para el caso de umbral 0 que clasifica todo como yema). Respecto a la precisión, se observa que a medida que baja el umbral esta se mantiene relativamente constante pero toda la curva baja. El problema con una precisión baja, es que el algoritmo clasifica muchos parches no-yema como yema. Dado que el conjunto de prueba se encuentra

balanceado, la peor precisión esperada es de 0,5 que ocurre en el caso de umbral 0 cuando todo es clasificado como yema . Vemos entonces que variando el umbral de clasificación no es posible obtener sensibilidad y precisión alta para distancias alejadas, forzando a operar a cortas distancias, con la consecuente sobrecarga en el número de vistas a tomar. Con esto, la dificultad fundamental que emerge al intentar realizar la detección de yemas por fuerza bruta con el enfoque del capítulo 4, es la gran cantidad de imágenes que se deben capturar y procesar a una distancia cercana a las yemas. Además, este enfoque no puede hacer nada para evitar procesar zonas de la escena que no contienen yema, teniendo en cuenta que las yemas no se encuentran distribuidas de manera uniforme en la escena y que ocupan un área ínfima de la misma.

Por lo tanto, el objetivo del algoritmo de BVA propuesto en este capítulo es reducir la cantidad de vistas requeridas por el enfoque de fuerza bruta al valor más cercano posible de la eficiencia máxima ideal. Asumiendo que para la distancia más cercana a la planta existe una yema por vista, la eficiencia máxima ideal corresponde a procesar una cantidad de vistas exactamente igual a la cantidad de vemas presentes en la escena. Sin embargo, es posible esperar que este valor sea aún menor, debido a que en ciertas ocasiones puede existir naturalmente más de una yema por vista. Si bien no es posible alcanzar este valor de eficiencia ideal, en este capítulo se presentan diferentes enfoques que logran distintos niveles de equilibrio entre estos extremos, explorando la posibilidad de encontrar las yemas de una planta desde una distancia alejada, permitiendo realizar desplazamientos y acercamientos de forma telescópica hacia las zonas donde se espera que se encuentren las yemas, evitando explorar el resto de la escena. De esta manera la navegación de las vistas está restringida sólo a movimientos lineales respecto al plano que forma la hilera del viñedo, i.e. la cámara que captura las imágenes solo posee tres grados de libertad: movimientos horizontales, verticales y acercamientos. Es importante notar que bajo este esquema de navegación no es posible manejar la pérdida de yemas por oclusión con otros elementos de la escena, siendo los casos más comunes yemas en la parte opuesta de la rama y yemas tapadas por otras ramas, entre otros.

Concretamente, en este capítulo se propone un enfoque de BVA para encontrar las yemas de una planta comenzando en vistas capturadas a una distancia lejana tal que las yemas tienen resoluciones menores a 20×20 píxeles y la cantidad de vistas requeridas para cubrir la escena a esa distancia es mucho menor que la requerida en la distancia más cercana. La planificación de vistas se realiza mediante un procesamiento de la imagen que permite determinar aquellas zonas de la vista actual que requieren exploraciones más cercanas y si es posible descartar otras zonas de la escena en exploraciones futuras. Este procesamiento se realiza a través de filtros que permi-

ten codificar el conocimiento de sentido común del problema: las yemas se desarrollan sobre las ramas; las ramas pertenecen a la clase de material madera de la vid; ramas y madera pueden ser detectadas en vistas donde las yemas son tan pequeñas que no sería posible detectarlas por sí solas (c.f. figura 1.1). Por lo tanto, se proponen un conjunto de filtros distintos inspirados en los resultados de los capítulos 3 y 4, pero ahora con una función específica para filtrar vistas dentro del proceso de planificación, independiente de su uso para segmentación de madera y clasificación de parches de yema que se les dio en sus respectivos capítulos. Además se incorpora otro algoritmo para detección de ramas basado en un algoritmo de detección de líneas, que resulta muy robusto y se destaca por su sencillez. Por último, estos filtros se combinan para mejorar los resultados que cada uno obtiene por separado.

5.2. Elementos de trabajo y metodología

En esta sección se dan detalles de la construcción del corpus de imágenes sobre las que se realizó la validación de los enfoques propuestos. Además se presentan las tecnologías empleadas para construir el conocimiento de sentido común como elemento principal del algoritmo de BVA de yemas de vid propuesto.

5.2.1. Construcción del escenario de evaluación

Como se explicó en la sección 5.1, la validez del enfoque de BVA propuesto en este capítulo se verificó en un ambiente controlado, a fin de abstraer los inconvenientes operacionales y técnicos que se derivan de la puesta en marcha de un VANT en la práctica. En este escenario controlado el objetivo es evaluar la tarea de búsqueda de yemas de una planta de vid a través de vistas tomadas a distancias diferentes, utilizando una estructura auxiliar en campo que permite posicionar de forma precisa y sencilla la cámara en el espacio. Vale la pena aclarar que una planta de vid con sistema de conducción de tipo espaldero (c.f. figura 5.1) se desarrolla a lo largo de varios alambres a diferentes alturas, en donde su ramas (o sarmientos) se sostienen durante toda la temporada. Una vez llegado el invierno, las ramas de la planta quedan ubicadas aproximadamente sobre el plano de la hilera que definen sus alambres. De esta manera, se puede esperar que todas las yemas de una planta se encuentren muy cerca del plano de la hilera sobre el espacio que ocupa la planta. Por lo tanto, para construir este escenario de evaluación fue necesario capturar un conjunto de imágenes perpendicular al plano de la hilera, aproximadamente a unos 12 centímetros de dicho plano, sobre todo el espacio que ocupa la planta. Dados estos requerimientos,



Figura 5.5: Estructura construida para capturar las imágenes en campo dados los requerimientos del escenario de evaluación para el algoritmo de BVA.

el proceso para capturar las imágenes en campo resultó muy exigente para realizarse manualmente. En vez, fue necesario construir una estructura donde se montó un dispositivo celular con una aplicación desarrollada para este problema en particular de captura de imágenes. Tal como se muestra en las imágenes de la figura 5.5, esta estructura consta de dos soportes verticales de 2 metros de altura anclados en la tierra paralelos al plano de la hilera. Sobre estos soportes se sostiene un perfil horizontal de 3 metros de ancho, cuya altura al suelo puede ser ajustada entre 15 centímetros y 2 metros. Dentro del perfil se dispone un carro deslizante que sujeta un mecanismo de agarre para el dispositivo celular. Este mecanismo fue construido a partir de piezas extraídas del artefacto comúnmente conocido como "palo o bastón selfie". Para asegurar que la estructura no se mueva durante una sesión de fotografías, el extremo de cada soporte vertical queda anclado al piso por tres cables tensores que afirman la estructura, como así también un cable adicional entre ambos extremos de los soportes. El dispositivo celular que se utilizó durante las sesiones de fotografía es un Samsung Galaxy S6 con sistema operativo Android 6.0 Marshmallow, cuyo modelo de cámara es SM-G920I con una resolución de 5312×2988 píxeles. La aplicación para capturar las imágenes se desarrollo específicamente para esta tarea a partir de la aplicación

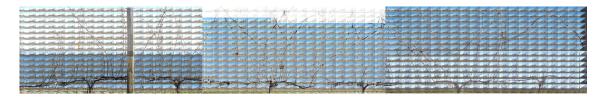


Figura 5.6: Montaje a partir de las 900 imágenes capturadas en campo a través de la estructura presentada en la figura 5.5.

de código abierto Open Camera¹. Sobre las funciones básicas de este aplicación se agregaron otras funciones requeridas para la captura de imágenes en campo, como ser superposición horizontal y vertical de las imágenes capturadas en la posición anterior y superior. Esta superposición se estableció (de manera sobredimensionada) en un 40 %, a fin evitar con toda certeza excluir zonas de la escena por pequeñas variaciones en la dirección de la cámara, ya que el mecanismo de desplazamiento introducía pequeñas perturbaciones en los ángulos de dirección de la misma. Utilizando esta estructura, se fotografiaron un total de 6,5 plantas de vid dispuestas a lo largo de unos 10,5 metros de hilera, requiriendo un total de 900 imágenes, tomadas en una grilla con 15 posiciones verticales y 60 posiciones horizontales. La figura 5.6 muestra un montaje de las 900 imágenes capturadas. Tal como se puede observar en la figura, existen zonas de la escena que presentan diferentes condiciones de luminosidad, situación que se mantuvo intencionalmente para no alterar las condiciones naturales de campo. Esta variabilidad en luminosidad se debe a que el proceso para capturar las 900 imágenes requirió tres campañas separadas, a lo largo de días y horarios diferentes.

Concretamente, el escenario de evaluación fue construido a partir de 108 imágenes con resoluciones de 5312×2988 píxeles, distribuidas en una grilla de 9 columnas por 12 filas, de manera tal que abarcan el espacio para una única planta. Específicamente, la planta utilizada es la segunda de derecha a izquierda de la figura 5.6. En la figura 5.7 se presenta el montaje de estas 108 imágenes. El resto de las 892 imágenes fue utilizada para propósitos de entrenamiento de los algoritmos utilizados en este capítulo. Dado que el problema requiere un escenario que permita realizar acercamientos telescópicos hacia las zonas donde se encuentran las yemas, fue necesario manipular las resoluciones de las imágenes para producir vistas a diferentes distancias. Además, es importante resaltar que todas las vistas deben tener exactamente la misma resolución, sin importar su distancia a la planta, ya que en la práctica se espera que sean capturadas con una única cámara de resolución fija. En primera instancia, cada una de las 108 imágenes fueron escaladas a una resolución de 664×374 píxeles, donde cada una es considerada una vista para la distancia más alejada con la que trabaja el algoritmo de 8VA.

¹Disponible en http://opencamera.sourceforge.net/

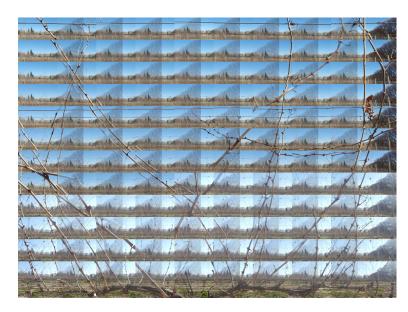


Figura 5.7: Montaje a partir de las 108 imágenes capturadas en campo a través de la estructura presentada en la figura 5.5, para la segunda planta de derecha a izquierda de la figura 5.6. Para realizar el montaje se superpusieron las imágenes un 40 % en dirección vertical y horizontal, transparencia en zonas donde se solapan las imágenes.

Si bien esta resolución puede parecer arbitraria, su justificación se encuentra en la necesidad de generar vistas a distancias más cercanas durante el procedimiento de acercamiento telescópico propuesto en este escenario de evaluación: partiendo de la vista más lejana, el acercamiento telescópico consiste en dividir la vista actual en 4 partes iguales, donde cada una de esas partes se convierte en las nuevas vistas a explorar en el siguiente nivel de distancia. De esta manera, establecer la resolución de las vistas a 664×374 píxeles permite generar 4 niveles de vistas con diferentes distancias de la siguiente manera: (i) la distancia más alejada, a partir de ahora nivel 3, corresponde a las 108 imágenes escaladas a 664×374 píxeles, tal como se explicó anteriormente; (ii) el siguiente acercamiento, llamado nivel 2, corresponde a 432 vistas obtenidas de particionar cada una de las 108 imágenes en cuatro, y escalar cada imagen a 664×374 píxeles; (iii) el nivel 1 contiene 1728 vistas producidas al dividir cada una de las 108 imágenes en 16 partes iguales y escalarlas; y (iv) el nivel 0 corresponde a las 6912 vistas generadas al particionar cada una de las 108 imágenes en 64 partes iguales, esta vez, sin necesidad de escalarlas, ya que al dividir el alto y el ancho de la imagen en 8 da exactamente la resolución de 664×374 píxeles. Este acercamiento telescópico se ilustra en la figura 5.8, donde las cuatro imágenes presentadas de izquierda a derecha corresponden respectivamente a una vista de nivel 3, cuatro vistas de nivel 2, 16 vistas de nivel 1 y 64 vistas de nivel 0. Finalmente, la resolución de las yemas disponibles en cada nivel se van reduciendo en un factor 2 de nivel a nivel. Por lo tanto, las vistas de nivel 0 contienen yemas con resoluciones aproximadas entre 80×80 y 150×150

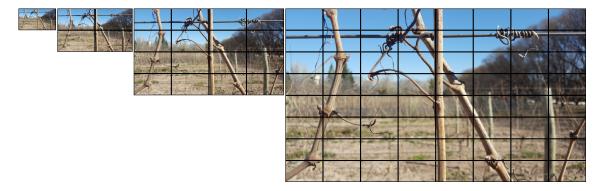


Figura 5.8: Ejemplo de acercamiento telescópico para una imagen del escenario de evaluación. Las cuatro imágenes presentadas de izquierda a derecha corresponden a: una vista de nivel 3; cuatro de nivel 2; 16 de nivel 1; y 64 de nivel 0.

píxeles; las vistas de nivel 1 yemas con resoluciones entre 40×40 y 75×75 píxeles; las de nivel 2 yemas entre 20×20 y 35×35 píxeles; y las vistas de nivel 3 yemas con resoluciones entre 10×10 y 18×18 píxeles.

5.2.2. Creación de un nuevo corpus de parches

Además de la construcción del escenario de evaluación, también fue necesario crear un nuevo corpus de parches con el objetivo de ser utilizado durante el entrenamiento del clasificador de yemas del capítulo 4, pero para la función específica de ser utilizado como filtro durante el proceso de planificación de vistas del algoritmo de BVA propuesto, tal como se explica en la sección 5.2.3. Este nuevo corpus se creó a partir de las 892 imágenes que quedaron fuera del escenario de evaluación, para lo cual se procesaron todas las imágenes y se extrajeron manualmente todos los parches de yema presentes en la escena y los parches no-yema fueron muestreados al azar, tal como se explica en la sección 4.2.1. En este nuevo corpus los parches de yema cuentan con resoluciones entre 80×80 y 150×150 . Una diferencia importante con el corpus del capítulo 4, es que los parches de yema de este corpus tienen resoluciones más homogéneas, lo que se interpreta como mucha menos variabilidad en la cantidad de información visual disponible para las yemas. Sin embargo, la calidad de la información visual entre ambos corpus es diferente. Por un lado, para el nuevo corpus las posiciones de la cámara están limitadas por las posiciones permitidas según la estructura, lo que implica que todas la imágenes son capturadas de forma perpendicular al plano de la hilera y aproximadamente a 12 centímetros del mismo. Para el corpus anterior las imágenes fueron tomadas a diferentes distancias y desde puntos de vista determinados según el criterio del operario en campo, con lo que la calidad del enfoque y definición de los parches de yema es superior en el corpus anterior.

Esta diferencia en la calidad de parches de yema afecta directamente al algoritmo SIFT de la siguiente manera: yemas mal enfocadas o con poca definición producen la pérdida de texturas que contienen información visual relevante para el proceso de clasificación. Si bien el detector de puntos de interés del algoritmo SIFT es invariante a escala y rotaciones, suele descartar puntos que se dan en zonas de una imagen con texturas suaves [115, 168]. Con esto, independientemente de la distancia a la que se capturó una imagen, si una yema no está bien enfocada y su definición es baja, inevitablemente se van a suavizar sus texturas relevantes. Por lo tanto, se puede establecer una relación entre la calidad de los parches de yema y la cantidad de puntos de interés que fueron detectados, asumiendo que luego de cierta cantidad, la calidad es suficientemente buena. Teniendo en cuenta esto, se presenta una forma de mejorar el rendimiento del esquema de clasificación de yema a partir de un corpus con parches de diferentes calidades: durante el entrenamiento del clasificador se filtran todos aquellos parches de yema que contienen menos puntos de interés que cierto umbral. En este caso, el umbral se estableció a 15 puntos de interés luego de realizar un análisis empírico para determinar el valor que mejora el rendimiento del clasificador, dejando fuera del entrenamiento sólo aquellos parches de la clase yema que no alcanzaban esta cantidad. Vale la pena resaltar que el filtro puntos de interés sólo se aplica a parches de la clase yema, exclusivamente durante la etapa de entrenamiento. Concretamente, el clasificador entrenado con parches de yema filtrados por cantidad de puntos de interés logra una exactitud de 0,857, con precisión y sensibilidad para la clase yema de 0,878 y 0,847 respectivamente, mientras que el clasificador entrenado sin filtro logra una exactitud de 0,795, con precisión y sensibilidad para yema de 0,772 y 0,843 respectivamente.

5.2.3. Planificación de vistas y construcción del conocimiento de sentido común

Como se comentó anteriormente, un aspecto muy importante de los algoritmos de detección en problemas de BVA es que codifican el conocimiento de sentido común relevante para la tarea de encontrar el objeto buscado como parte del proceso de planificación de vistas. Para el caso de la búsqueda de yemas de vid presentado en este capítulo el conocimiento relevante consiste en que las yemas se encuentran en lugares particulares de la vid y con una alta correlación espacial con otros elementos de la planta (c.f. figura 1.1): la yemas se desarrollan en las ramas y las ramas pertenecen al material madera. Esta información es importante para limitar el espacio de búsqueda y mejorar la eficiencia de la detección y exploración en cualquier entorno natural de

viñedo desconocido, función que cumple el planificador de vistas. Además, el conocimiento de estas relaciones (madera-rama-yema), permite hacer suposiciones de las potenciales ubicaciones en la vista actual de las instancias de una yema, información útil cuando las yemas son tan pequeñas que no sería posible detectarlas por si solas.

Respecto a la planificación de vistas, el enfoque utilizado en este capítulo es bastante elemental y se encuentra acoplado al escenario de evaluación descrito en la sección 5.2.1. Básicamente, el planificador evalúa la vista actual a través de los filtros que codifican el sentido común, y luego toma la decisión hacia qué zonas de la escena se debe realizar un acercamiento telescópico y qué partes pueden ser obviadas en sucesivas búsquedas, teniendo en cuenta que cada una de las vistas de un nivel abarcan la misma escena que cuatro vistas del siguiente nivel. Con esto, el planificar divide la vista actual en cuatro cuadrantes iguales en correspondencia directa a las cuatro vistas del siguiente nivel. Si el filtro indica que dentro de ese cuadrante es factible que existe una yema, entonces su vista correspondiente en el siguiente nivel es tenida en cuenta para continuar la búsqueda. En caso contrario, si el filtro indica que no es factible que exista una yema, descarta la vista del siguiente nivel. En la figura 5.9 se ilustra el funcionamiento del planificador. En la fila superior, de izquierda a derecha, primero se muestra la vista actual a procesar. En segundo lugar se presenta como una máscara binaria la información que entregaría un filtro de yemas perfecto (en este caso corresponde a una segmentación manual de las yemas). La última imagen presenta un esquema que resume la división de la vista en sus cuadrantes y su posterior análisis respecto a la información del filtro. El cuadrante coloreado en gris indica que no es factible que exista una yema en esa zona de la escena, por lo que el planificador toma

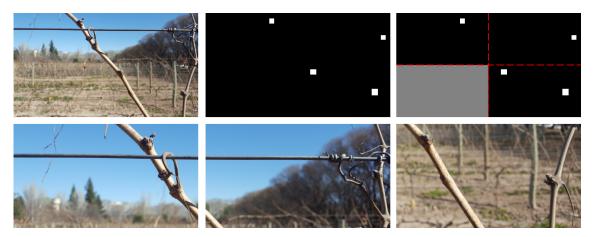


Figura 5.9: Arriba, de izquierda a derecha: vista actual a procesar; segmentación manual de las yemas; esquema para la división de la vista en sus cuadrantes y su análisis respecto a la información del filtro. Abajo, de izquierda a derecha: vistas que sobrevivieron al proceso de planificación y requieren ser exploradas en el próximo nivel.

la decisión de descartar la vista correspondiente al cuadrante del siguiente nivel. En la fila inferior se muestran las vistas que sobrevivieron al proceso de planificación y requieren ser exploradas iterativamente con el mismo procedimiento en el siguiente nivel. En este ejemplo se puede ver que la ganancia en procesamiento de vistas aumenta cuando las zonas de la escena que no contienen yemas son descartadas en los niveles más alejados.

A continuación se presentan los filtros utilizados para codificar este conocimiento en el algoritmo de BVA. Estos filtros están inspirados en las herramientas presentadas en los capítulos 3 y 4, mientras que se incorpora un nuevo algoritmo para detección de ramas que se utiliza en la construcción del conocimiento del sentido común.

Filtro de yema

Inspirado en la resultados presentados en la sección 5.1 sobre la precisión y sensibilidad alcanzados por el clasificador de yemas del capítulo 4 para diferentes distancias y umbrales de clasificación, este filtro de yema consiste en realizar un proceso de escaneo-ventana sobre una vista y correr el clasificador de yemas, pero esta vez con un umbral de clasificación adecuado para que la efectividad de la búsqueda de yemas en distancias muy alejadas no se vea perjudicada. De aquí en adelante, se hace referencia al umbral de clasificación como el parámetro μ . Dado que el algoritmo de clasificación de yemas ahora cumple una función de filtro de vistas dentro del escenario de validación, fue necesario realizar el proceso de entrenamiento de un nuevo modelo de clasificación a partir del nuevo corpus de parches presentado en la sección 5.2.2. El entrenamiento del clasificador se realizó de la misma manera en la que se presenta en el capítulo 4. El conjunto de entrenamiento fue balanceado según el valor de R=1, i.e. se mantuvo la cantidad de elementos de la clase yema fijo y se realizó un submuestreo de la clase no-yema con la misma cantidad de elementos de yema. El tamaños de diccionario utilizado para la construcción de los escritores BoF fue S=25. Finalmente, los parámetros C y γ del algoritmo MVS fueron elegidos mediante una validación cruzada de 5 iteraciones para los rangos de valores $\gamma = \{10^{-4}, 10^{-3}, ..., 10^3, 10^4\}$ y $C = \{10^{-4}, 10^{-3}, ..., 10^{3}, 10^{4}\}$. Por último el proceso de escaneo-ventana que se aplica para extraer los parches de cada vista requiere la definición de cuatro valores: tamaño de la ventana, i.e. alto y ancho; y desplazamiento de la ventana, tanto horizontal como vertical. En este trabajo se toman ventanas de igual alto y ancho, asumiendo que es la mejor relación para la tarea de clasificación de yemas dadas sus características morfológicas. Para cada nivel del escenario de evaluación se define el tamaño de la ventana en píxeles según la mayor resolución esperada de una yema en ese nivel. Respecto al desplazamiento horizontal y vertical de cada nivel, se establecen ambos a la

mitad del valor del alto y ancho de la ventana respectivamente. Por lo tanto, solo es necesario establecer el valor de un solo parámetro, referido como ω_0 , para definir el proceso de escaneo-ventana en todos los niveles del escenario de evaluación. Para el caso del nivel 0, ω_0 se establece a 150 (i.e. máxima resolución esperada de una yema para la distancia más cercana), por lo que el tamaño de la ventana es de 150 × 150 píxeles, con un desplazamiento horizontal y vertical de 75 píxeles. En el nivel 1 se define $\omega_1 = \frac{1}{2}\omega_0$; para el nivel 2 se define $\omega_2 = \frac{1}{4}\omega_0$; y por último, en el nivel 3 se define $\omega_3 = \frac{1}{8}\omega_0$. Para ilustrar el funcionamiento de este algoritmo, a continuación se presenta un breve ejemplo para umbrales de clasificación diferentes.

Como se puede observar en la figura 5.4, para $\mu = 0.5$ la sensibilidad del clasificador es alrededor de 0,85 cuando se mantiene la resolución original de las yemas, y comienza a caer cuando los parches son escalados en un 50 % o menos. Dado este análisis, es posible esperar que el clasificador de yemas sea efectivo para descartar vistas en los niveles 0 y 1 del escenario de evaluación. Por otro lado, para los niveles 2 y 3, que corresponden a valores de resolución de 0,25 y 0,125 en la figura 5.4, la efectividad de este filtro se ve perjudicada. Como antes, disminuyendo μ es posible recuperar valores altos de sensibilidad. De esta manera podría pensarse en establecer un régimen de umbrales que depende del nivel al que pertenece la vista que se está procesando. Sin embargo, a medida que μ disminuye, la precisión también comienza a bajar afectando la eficiencia del filtro para descartar vistas que no contienen yemas. Esta situación se ve reflejada en la figura 5.10, donde se muestra una vista (primer imagen de izquierda a derecha), seguida de las máscaras binarias generadas a partir de un proceso de escaneo-ventana y clasificación de yemas con $\mu = 0.5$ y $\mu = 0.2$ (segunda y tercer imagen respectivamente). Se puede observar que para la máscara de umbral 0,5 el filtro pierde tres de las cuatro yemas que se encuentran en la escena². Por otro lado, para la máscara de umbral 0,2, todas las yemas son detectadas, pero se agregan incorrectamente muchas zonas donde no existen yemas. En ambas máscaras las yemas han sido destacadas manualmente con rectángulos de color rojo. Luego de este análisis se hace evidente que no es posible recuperar los resultados de precisión y sensibilidad en distancias alejadas, y reducir el valor del umbral de clasificación tampoco es una solución: el balance entre precisión y sensibilidad nunca se acomoda a los requerimientos de eficiencia y efectividad de la BVA. Frente a esta situación, surge la necesidad de conservar la efectividad del clasificador de yemas con variación de umbral para mantener aquellas vistas que contienen yemas, a la vez que se aumenta su eficiencia para descartar vistas sin yemas por otros medios. Por lo tanto, a continuación

 $^{^2}$ Notar que existen dos yemas extras sobre los nudos de ambas ramas, pero están ocluidas para la misma rama y no se tienen en cuenta

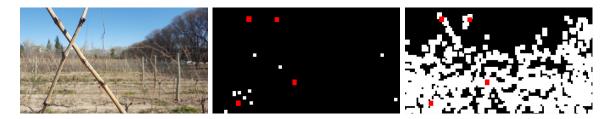


Figura 5.10: Proceso para el filtrado por yemas. Izquierda, la vista a procesar. Centro, máscara binaria generada a partir del clasificador de yemas $\mu=0,5$. Derecha, máscara binaria generada a para $\mu=0,2$. La ubicación de las yemas en las máscaras se resaltan (manualmente) en rojo. Mientras que en el primer caso se pierden tres de las cuatro yemas, en el segundo todas las yemas son detectadas a costa de agregar zonas donde no existen yemas.

se propone una forma de mejorar los resultados del filtro de yemas con variación de umbrales combinándolo con otros algoritmos que se desprenden del conocimiento de sentido común del problema. Para esto primero se procesan las vistas con un algoritmo para la detección de ramas y se combinan sus resultados con la clasificación de yemas, lo que permite elevar la eficiencia de la búsqueda. Luego se mejoran aún más estos resultados aplicando un nuevo filtro que implica segmentar todo aquello que corresponde a madera y combinar sus resultados con la detección de ramas y clasificación de yemas.

Filtro de rama-yema

Este filtro, denominado filtro de rama-yema, surge bajo la consigna de mejorar la eficiencia de la búsqueda utilizando información del proceso de clasificación de yemas sobre zonas de la vista donde se espera detectar ramas, tal como lo sugiere el sentido común. Para esta tarea primero fue necesario desarrollar un algoritmo para la detección de ramas, el cual ha demostrado ser robusto en el escenario de evaluación propuesto. A continuación se dan los detalles del algoritmo de detección de ramas y luego se explica cómo se combina con el filtro de yema.

Básicamente, el algoritmo de detección de ramas realiza el procesamiento de la imagen en tres etapas: (i) se realiza una detección de bordes sobre la vista, asumiendo que los bordes de las ramas tienen suficiente contraste del resto de la escena; (ii) sobre la detección de bordes se realiza una detección de líneas rectas, bajo el hecho de que las ramas son rectas al menos por tramos; y (iii) a partir de cada línea recta se establece una región donde se espera que efectivamente exista una rama. Para la detección de bordes del punto (i) se utiliza el algoritmo de Canny, propuesto por John F. Canny en 1986 [29]. La implementación empleada es la provista por la librería

OpenCV, dejando todos su parámetros por defecto³. Por otro lado, la detección de líneas rectas a partir de los bordes detectados se realiza con el algoritmo propuesto por von Gioi et al. (2012) [155], cuya implementación también se encuentra disponible en la librería OpenCV. Nuevamente, todos los parámetros de este algoritmo se dejaron en sus valores por defecto⁴. Respecto al punto (iii) del algoritmo, para establecer la región de rama es necesario definir el parámetro δ , que corresponde al ancho esperado de una rama para la distancia de la vista que se está explorando. Si bien el parámetro δ debe ser establecido externamente, se define sólo para la distancia más cercana y luego se escala para cada nivel del escenario de evaluación. Concretamente, en este trabajo el valor de δ_i para cada nivel i fue establecido a $\frac{1}{2}\omega_i$. Para generar una región a partir de una línea recta detectada, en primer lugar se estira la línea desde sus dos extremos en una magnitud igual a 2δ , para luego ser ensanchada hacia cada lado de la línea en una magnitud igual δ . El procedimiento de estiramiento se justifica por el hecho de que las ramas son rectas por tramos, presentando leves curvas en las zonas donde se desarrollan sus nudos, lo que produce una potencial discontinuidad en la detección de líneas rectas. Además, sobre los nudos de la rama se desarrollan justamente las yemas, por lo que una manera sencilla de asegurar la detección sobre estas zonas particulares de la rama es estirar las líneas rectas detectadas. Por otro lado, dado que las líneas rectas se detectan sobre los contornos de la rama, un valor de δ igual a su ancho asegura que el procedimiento de ensanchamiento genere una región que cubra todo el interior de la rama, aunque también produce que la región se extienda más allá de sus bordes. Si bien una región que supera los límites de la rama puede parecer un problema, ofrece garantías para esperar que las yemas de las ramas también sean cubiertas por la región, ya que, dependiendo de la toma de la imagen, la yema podría ubicarse sobre el contorno de la rama, justo sobre el límite de sus bordes. En la figura 5.11 se muestra un ejemplo del funcionamiento del algoritmo de detección de ramas en sus tres etapas. La primer imagen de la figura (de izquierda a derecha y de arriba a abajo) muestra la vista a procesar, seguida de la detección de bordes, la detección de líneas y finalmente la generación de regiones a partir de ellas. En esta figura se puede observar la ventaja de realizar la detección de líneas rectas, ya que esta etapa permite eliminar algunos bordes espurios sobre zonas que no corresponde a las ramas. Incluso durante esta etapa se eliminan bordes que se dan sobre las yemas por no ser líneas rectas, lo que se compensa estirando las líneas en sus extremos en el último paso, tal como se ilustra en la imagen de abajo a la derecha de la figura. El resultado final del proceso entrega una máscara binaria que puede ser utilizada para tomar la decisión si

³http://docs.opency.org/3.1.0/da/d5c/tutorial_canny_detector.html

⁴http://docs.opencv.org/3.1.0/db/d73/classcv_1_1LineSegmentDetector.html



Figura 5.11: Proceso para la detección de ramas. De izquierda a derecha y de arriba a abajo: vista original; detección de bordes; detección de líneas rectas; y creación de regiones de rama.

es necesario continuar la búsqueda telescópica de yemas por alguna de las vistas del siguiente nivel.

Finalmente, para construir el filtro de rama-yema, se aplica el siguiente procedimiento: para cada vista a procesar se crean dos máscaras binarias, una a partir del algoritmo de detección de ramas y otra partir del filtro de yema del apartado anterior para un valor de μ determinado, y se genera una nueva máscara que consiste en la multiplicación píxel a píxel de ambas máscaras, lo que equivale a una conjunción lógica entre ellas. En la figura 5.12 se muestra el resultado producido por este filtro combinado para la misma vista de la figura 5.11. En la imagen derecha de la figura se presenta la máscara binaria final producida por la multiplicación de las otras dos



Figura 5.12: Proceso para el filtrado por clasificación de yemas y detección de ramas. Izquierda, máscara de regiones producida por el algoritmo de detección de ramas para la vista presentada en la figura 5.11. Centro, máscara binaria generada a partir del clasificador de yemas $\mu=0,2$ para la misma vista. Derecha, máscara binaria a partir de la multiplicación de las dos máscaras anteriores.

máscaras, i.e. la máscara para la detección de ramas (imagen izquierda) y la máscara para la clasificación de yemas (imagen central) con $\mu=0,2$. En la máscara final se puede observar cómo han desaparecido algunas zonas de la imagen que efectivamente no contienen yemas, esperando que este comportamiento en general mejore la eficiencia del filtro de yema propuesto en el apartado anterior.

Filtro de rama-madera-yema

En un intento por mejorar aún más el proceso de filtrado mediante el algoritmo de detección de ramas combinado con el filtro de yemas, se presenta un nuevo filtro, llamado filtro de rama-madera-yema, que hace uso de otro hecho que emerge del sentido común: las ramas forman parte del tipo de material madera. Por lo tanto se propone modificar el algoritmo de detección de ramas para eliminar líneas rectas que se dan sobre elementos de la escena que no son madera, como ser la aparición de líneas espurias en elementos del fondo o sobre los alambres de la hilera. El nuevo algoritmo, denominado algoritmo de detección de rama-madera, consiste en agregar un paso extra durante la etapa (iii) del mismo: antes de alargar y ensanchar una línea recta para establecer su región, se verifica que la línea tenga al menos un píxel de intersección con la máscara obtenida al segmentar la madera del resto de los elementos, caso contrario se descarta la línea como espuria y no se genera su correspondiente región. Para obtener la máscara de madera se utilizó el algoritmo propuesto en el capítulo 3. Se entrenaron dos modelos de mezcla de gaussianas, uno para cada clase de interés, i.e. madera y no-madera, de la misma manera como se explica en la sección 3.2.6. Para el entrenamiento se tomó una muestra de aproximadamente 3 000 000 de píxeles para madera y otros 3 000 000 para no-madera, extraídos de las 892 imágenes que quedaron fuera del escenario de evaluación. La cantidad de componentes C de cada modelos fue C=2 para madera y C=4 para no-madera. El resto de los parámetros del algoritmo de segmentación son los mismos a los que se proponen en la sección3.2.6. El funcionamiento de este nuevo algoritmo se ilustra en la figura 5.13. La primer imagen de la figura (de izquierda a derecha y de arriba a abajo) muestra la vista a ser procesada, seguido de la máscara producida por la detección de regiones según el algoritmo del apartado anterior, donde se observa que se generan una gran cantidad de regiones derivadas de líneas espurias. La tercer imagen muestra la máscara de madera producida por el algoritmo de segmentación. Finalmente, la última imagen muestra la nueva máscara producida por el algoritmo de detección rama-madera presentado en este apartado, donde se puede observar que se han eliminado varias zonas de la imagen que no corresponden a madera, como ser las zonas sobre el alambre de la hilera y otras zonas en el fondo de la imagen.



Figura 5.13: Proceso para la detección de ramas con eliminación de líneas según la máscara de segmentación de madera. De izquierda a derecha y de arriba a abajo: vista original; máscara para la detección de ramas del apartado anterior; máscara producida por la segmentación de madera; y nueva máscara según la detección de ramas con segmentación de madera explicado en este apartado.

De las misma manera que para el filtro de rama-yema, esta nueva detección de ramas puede ser combinada con el filtro de yema para generar una nueva máscara construida como una multiplicación píxel a píxel de las máscaras producidas por el algoritmo de detección rama-madera y el filtro de yema, esperando que esta combinación logre mejores resultados que cuando se usan por separado. Los resultados de este filtro se presentan en la figura 5.14. En la figura se observa el potencial que tiene realizar esta combinación de filtros. En ninguno de los dos filtros anteriores hubiese sido posible descartar alguna vista del siguiente nivel debido a que sus máscaras tiene zonas activas distribuidas en toda la imagen, mientras que combinando ambas ahora es posible descartar la vista del siguiente nivel que corresponde al cuadrante inferior



Figura 5.14: Filtro de rama-madera-yema. Izquierda, máscara para la detección de rama-madera para la vista presentada en la figura 5.13. Centro, máscara producida por el clasificador de yemas con $\mu = 0,3$. Derecha, máscara de rama-madera-yema producto de la multiplicación píxel a píxel de las dos máscara anteriores.

izquierdo de la vista actual.

Con la introducción de este filtro finaliza la presentación de los diferentes enfoques utilizados en la construcción del conocimiento de sentido común para el algoritmo de BVA evaluado en el escenario de evaluación propuesto. A continuación se presentan los resultados experimentales logrados para cada uno de estos enfoques.

5.3. Resultados experimentales

En esta sección se presentan los detalles de la experimentación realizada y las medidas de rendimiento empleadas para evaluar la calidad de los filtros propuestos dentro del proceso de planificación de vistas del algoritmo de BVA. Particularmente, la calidad es evaluada a través de la efectividad y de la eficiencia que presenta el planificador durante la búsqueda de las yemas en el escenario de evaluación propuesto en la sección 5.2.1. Por lo tanto, la efectividad de la búsqueda es un valor entre 0 y 1 que se define como el número de yemas que están contenidas en las vistas visitadas de nivel 0 respecto a las 338 yemas presentes en las imágenes que conforman la escena total. Por vista visitada se refiere a las vistas de nivel 0 que no fueron descartadas durante el proceso de planificación. Por otro lado, la eficiencia de la búsqueda es un número entero que se define como la cantidad total de vistas procesadas por el planificador en todos los niveles, incluyendo las vistas visitadas del nivel 0. Esta medidas son contrastadas con la efectividad y eficiencia esperada del enfoque de fuerza bruta en el nivel 0. Como ya se destacó anteriormente, este enfoque tiene una efectividad de 1, aunque acompañado de una eficiencia aproximada de 0,049 (asumiendo el caso de una yema por vista), ya que necesita procesar 6912 vistas en el nivel 0 para encontrar las 338 yemas presentes en el escenario de evaluación, es decir que como mínimo el 95%de las vistas del nivel 0 no contienen yemas.

Como se explica en el apartado 5.2.3, el filtro de yema requiere que se especifique el umbral de clasificación para el algoritmo de clasificación de yemas, denotado por μ , parámetro que también es requerido en el resto de los filtros, donde también se utiliza el filtro de yema. Además, un mismo valor de μ puede impactar de forma muy distante cuando se aplica el filtro en diferentes niveles, por lo que también es necesario establecer un valor μ_i para cada nivel i. Sin embargo, si en el nivel i se establece un valor determinado para μ_i , en el nivel (i-1) (i.e. el siguiente nivel más cercano) el valor de μ_{i-1} debe ser mayor o igual que μ_i . Esto se debe a que la reducción del valor del umbral de clasificación busca mantener la sensibilidad de la clase yema elevada para las vistas del siguiente nivel más alejado. Por ejemplo, para las vistas de nivel 3 (el más alejado) puede tener sentido utilizar un $\mu_3 = 0.1$, pero a medida que se realizan

acercamientos avanzando a los siguientes niveles, el valor de μ para cada uno debería ser como mínimo de 0,1. Por otro lado, en este trabajo se fijó el valor máximo que puede tomar μ a 0,5, lo que corresponde al valor de umbral que suele ser utilizado en la práctica. Además, valores mayores que 0,5 producen que la sensibilidad del algoritmo de clasificación de yemas disminuya, impactando negativamente en la efectividad de la búsqueda para encontrar yemas. Durante la experimentación se presentan los resultados de las medidas de rendimiento para diferentes valores de μ_i , tal que $\mu_i = K_j$, con $K = \{0; 0,1; 0,2; 0,3; 0,4; 0,5\}$, bajo la restricción de que $\mu_{i-1} \geq \mu_i$. Por ejemplo, la asignación de umbrales ($\mu_3 = 0; \mu_2 = 0,1; \mu_1 = 0,2; \mu_0 = 0,3$) está permitida, mientras que la asignación ($\mu_3 = 0; \mu_2 = 0,1; \mu_1 = 0,2; \mu_0 = 0$) no lo está. En este escenario, la cantidad total de combinaciones de umbrales es de 126, que resulta de todas las posibles combinaciones de μ_i para los cuatro niveles propuestos bajo la restricción impuesta. Por lo tanto, para cada una de los tres filtros propuestos se cuentan con 126 casos que entregan diferentes valores de eficiencia y efectividad.

Los resultados obtenidos por cada filtro sobre el escenario de validación propuesto se presentan en la figura 5.15. En la figura se resume el valor de eficiencia y efectividad obtenido por cada uno de los tres filtros con sus 126 combinaciones de umbrales, junto a los resultados del caso por fuerza bruta para el nivel 0 presentado en la sección 5.1. Para el caso de fuerza bruta se considera una efectividad igual a 1 con una eficiencia de 6912 vistas visitadas y se presenta en la gráfica con un círculo negro sin relleno. En la figura también se presentan dos casos más: (i) el caso particular de un filtro que no descarta ninguna vista durante la planificación, logrando una efectividad de 1, pero la peor eficiencia posible dentro del escenario de evaluación, que corresponde a la suma de todas las vistas disponibles en cada uno de los cuatro niveles; y (ii) el peor caso posible donde un filtro descarta absolutamente todas las vistas visitadas, logrando la mayor eficiencia posible, i.e. 108 vistas visitadas, con una efectividad igual a 0. El caso (i) se indica con un círculo relleno de color negro bajo la etiqueta de filtro nulo, mientras que el caso (ii) se indica con un cuadrado relleno de color gris bajo la etiqueta de filtro total. Tanto el caso de fuerza bruta como los casos de filtro nulo y filtro total se utilizan como valores de línea base que se toman como referencia para la evaluación del resto de los enfoques.

Comenzando por el filtro de yema (denotado con triángulo rojo) se puede observar que existe un conjunto particular de casos que mantienen una efectividad muy elevada, pero se encuentran agrupados muy cerca del caso de filtro nulo, necesitando procesar mucha más cantidad de vistas que las 6912 del caso de fuerza bruta. Este grupo corresponde a 45 casos para diferentes combinaciones de umbrales. El resto de los 81 casos logran valores de efectividad entre 0,994 y 0,846, pero producen una planificación

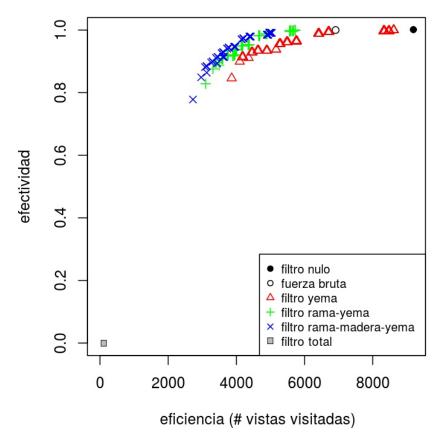


Figura 5.15: Resultados de eficiencia y efectividad obtenido por cada uno de los tres filtros con sus 126 combinaciones de umbrales, junto a los resultados del caso por fuerza bruta y los casos especiales de filtro nulo y filtro total.

de vistas que resulta ser más eficiente que el caso de fuerza bruta, con valores de eficiencia que van desde 6722 hasta 3864 vistas visitadas. Por otro lado, la totalidad de los 126 casos reportados para el filtro de rama-yema (denotados con el símbolo "+" verde) presenta mejor eficiencia que el caso de fuerza bruta, con valores entre 5722 y 3100 vistas visitadas, y una efectividad entre 1 y 0,828. Comparándolo con el filtro de yema, en la figura se puede observar que el filtro rama-yema desplaza todos los puntos hacia la izquierda de la gráfica con una pequeña tendencia a disminuir el valor de efectividad. En general esto implica que el filtro de rama-yema logra mejores resultados de eficiencia para aproximadamente lo mismo resultados de efectividad. Finalmente, el análisis para los 126 casos del filtro de rama-yema-madera (simbolizados con "x" azul) muestra una nueva mejora en eficiencia respecto al resto de los filtros. En la figura se puede observar que el filtro rama-madera-yema desplaza todos los puntos del filtro rama-yema hacia la izquierda, con una tendencia a disminuir el valor de efectividad. Nuevamente, este comportamiento implica que este filtro logra mejores resultados de eficiencia para aproximadamente lo mismo resultados de efectividad. Particularmente, la efectividad de este filtro varía entre 0,991 y 0,778, mientras que los valores de eficiencia van desde 5033 hasta 2728.

A fin de poder realizar una comparativa directa entre los filtros propuestos en este trabajo, se propone construir una curva para cada filtro de manera que se pueda calcular el área bajo la curva (ABC) de cada una, entendiendo que una curva es mejor cuando su valor de ABC es más alto. Para construir las curvas fue necesario agregar dos puntos extras a los 126 puntos disponibles de cada filtro: el primero corresponde al punto determinado por el filtro total, considerado el punto inicial de las curvas; y el segundo corresponde al punto del filtro nulo, considerado el punto final de las curvas. Estas curvas se presentan en la figura 5.16, junto a los valores alcanzados de ABC. El valor de ABC para la curva del filtro yema es de 0,735, mientras que para la curva del filtro rama-yema es de 0,783. Finalmente, el mejor ABC lo alcanza la curva del filtro rama-madera-yema con un valor de 0,795. Además, en la gráfica se puede observar que en general la curva de yema se mantiene por debajo de las otras dos curvas, mientras que la curva de región-yema en general se encuentra por debajo de la curva región-madera-yema, salvo en la parte más elevada de ambas curvas, donde se invierte la situación. Esta situación es acorde al análisis basado en valores de ABC.

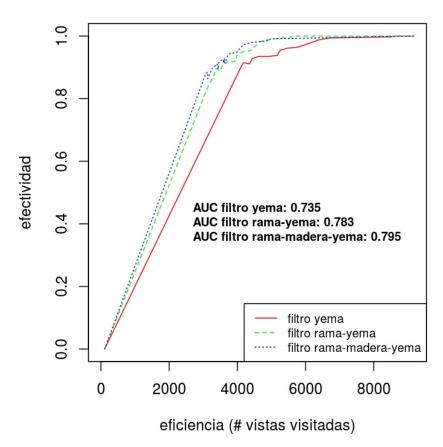


Figura 5.16: Curvas correspondiente a cada filtro, junto al valor calculado del *área* bajo la curva (ABC).

Por último, en la tabla 5.1 se presentan los resultados de efectividad y eficiencia obtenidos para un subconjunto de las 126 asignaciones de umbrales a niveles disponibles para cada filtro. Por cuestiones de espacio y conveniencia, en la tabla se hace referencia a los filtros de yema, rama-yema y rama-madera-yema con las abreviaciones Y, RY y RMY respectivamente.

Umbrales				Efecti	Efectividad			 Eficiencia		
μ_3	μ_2	μ_1	μ_0	Y	RY	RMY	Y	RY	RMY	
0	0	0	0	1	1	0,991	8615	5722	5033	
0	0	0,1	0,1	0,997	0,997	0,991	8487	5673	4991	
0	0	0,2	0,2	0,997	0,997	0,985	8347	5597	4931	
0	0	0,3	0,3	0,994	0,988	0,979	6722	4900	4423	
0	0	0,4	0,4	0,964	0,953	0,947	5776	4382	3971	
0	0	0,5	0,5	0,935	0,92	0,914	4909	3957	3640	
0	0,1	0,1	0,1	0,997	0,997	0,991	8470	5659	4975	
0	0,1	0,2	0,2	0,997	0,997	0,985	8330	5583	4915	
0	0,1	0,3	0,3	0,994	0,988	0,979	6707	4886	4408	
0	0,1	0,4	0,4	0,964	0,953	0,947	5761	4368	3956	
0	0,1	0,5	0,5	0,935	0,92	0,914	4894	3943	3625	
0	0,2	0,2	0,2	0,997	0,997	0,985	8322	5568	4900	
0	0,2	0,3	0,3	0,994	0,988	0,979	6699	4872	4394	
0	0,2	0,4	0,4	0,964	0,953	0,947	5753	4354	3942	
0	0,2	0,5	0,5	0,935	0,92	0,914	4886	3929	3611	
0	0,3	0,3	0,3	0,988	0,982	0,973	6428	4684	$\boldsymbol{4209}$	
0	0,3	0,4	0,4	0,962	0,95	0,944	5502	4177	3782	
0	0,3	0,5	0,5	0,935	0,92	0,914	4647	3755	3454	
0	0,4	0,4	0,4	0,956	0,938	0,929	5291	4006	3619	
0	0,4	0,5	0,5	0,929	0,908	0,899	4470	3597	3307	
0	0,5	0,5	0,5	0,914	0,896	0,885	4195	3409	3129	
0,1	0,1	0,1	0,1	$\boldsymbol{0,997}$	$\boldsymbol{0,997}$	0,991	8470	5658	4972	
0,1	0,1	0,2	0,2	$\boldsymbol{0,997}$	$\boldsymbol{0,997}$	0,985	8330	5582	4912	
0,1	0,1	0,3	0,3	0,994	0,988	0,979	6707	4885	4405	
0,1	0,1	0,4	0,4	0,964	0,953	0,947	5761	4367	3953	
0,1	0,1	0,5	0,5	0,935	0,92	0,914	4894	3942	3622	
0,1	0,2	0,2	0,2	0,997	$\boldsymbol{0,997}$	0,985	8322	5567	4897	
0,1	0,2	0,3	0,3	0,994	0,988	0,979	6699	4871	4391	
0,1	0,2	0,4	0,4	0,964	0,953	0,947	5753	4353	3939	

0,1	0,2	0,5	0,5	0,935	0,92	0,914	4886	3928	3608
0,1	0,3	0,3	0,3	0,988	0,982	0,973	6428	4683	4206
0,1	0,3	0,4	0,4	0,962	0,95	0,944	5502	4176	3779
0,1	0,3	0,5	0,5	0,935	0,92	0,914	4647	3754	3451
0,1	0,4	0,4	0,4	$0,\!956$	0,938	0,929	5291	4005	3616
0,1	0,4	0,5	0,5	0,929	0,908	0,899	4470	3596	3304
0,1	0,5	0,5	0,5	0,914	0,896	0,885	4195	3408	3126
0,3	0,3	0,3	0,3	0,988	0,982	0,97	6401	4657	4156
0,3	0,3	0,4	0,4	0,962	0,95	0,941	5475	4150	3730
0,3	0,3	0,5	0,5	0,935	0,92	0,911	4620	3728	3402
0,3	0,4	0,4	0,4	$0,\!956$	0,938	0,926	5264	3979	3571
0,3	0,4	0,5	0,5	0,929	0,908	0,896	4443	3570	3259
0,3	0,5	0,5	0,5	0,914	0,896	0,882	4169	3383	3082
0,4	0,4	0,4	0,4	0,938	0,917	0,893	5167	3896	3437
0,4	0,4	0,5	0,5	0,911	0,888	0,864	4360	3495	3132
0,4	0,5	0,5	0,5	0,899	0,876	0,849	4101	3314	2968
0,5	0,5	0,5	0,5	$0,\!846$	0,828	0,778	3864	3100	2728

Tabla 5.1: Resultados de efectividad y eficiencia obtenidos para un subconjunto de las 126 asignaciones de umbrales a niveles disponibles para cada filtro. Y, RY y RMY hacen referencia a los filtros de yema, rama-yema y rama-madera-yema respectivamente. En negrita se destacan los mejores resultados de cada asignación, para cada medida, sobre los tres filtros presentados.

Como se puede observar en la figura 5.15, para cada filtro existe una gran cantidad de puntos que se superponen entre sí. Por lo tanto, para reducir esta información redundante, y en consecuencia el número de casos presentados en la tabla, se realizó el siguiente procedimiento: primero se verificó para cada filtro por separado cuales asignaciones tenían exactamente los mismo valores de efectividad y eficiencia, y, salvo el primer caso de ese grupo, el resto fueron marcados como asignaciones duplicadas dentro de los resultados de ese filtro; luego se descartaron de la tabla todas aquellas asignaciones que para los tres filtros fueron marcadas como duplicadas. Este proceso redujo la cantidad de casos de 126 a 46. Por ejemplo, como se muestra en la tabla, a partir de la asignación ($\mu_3 = 0; \mu_2 = 0; \mu_1 = 0; \mu_0 = 0$) fue posible descartar los cinco casos que resultan al establecer el valor de μ_0 en 0,1, 0,2, 0,3, 0,4 y 0,5 ya que eran casos con valores duplicados en cada filtro. En la tabla, los mejores resultados de cada medida para cada asignación se destacan en negrita. Examinando estos valores, se verifica que en todos los casos la efectividad más alta la logra el filtro de yema, mientras que la eficiencia mayor corresponde al filtro de rama-madera-yema. Por otro

lado, el filtro de rama-madera logra igualar en algunos pocos casos la efectividad del filtro de yema, pero baja notablemente la eficiencia respecto al anterior. En la tabla también se puede observar los comportamientos descritos a partir del análisis de las gráficas, como ser el desplazamiento de los puntos hacia zonas de mayor eficiencia a medida que se pasa del filtro de yema, al de rama-yema y al de rama-madera-yema.

5.4. Discusión

En esta sección se discuten los alcances y limitaciones de los diferentes filtros propuestos para realizar el proceso de planificación de vistas, interpretando sus resultados en el contexto del problema de BVA de yemas para la detección de yemas de una planta de vid, presentando además algunas vías de mejora.

En una primera etapa, en este capítulo se exploró la posibilidad de aplicar el algoritmo de clasificación de yemas del capítulo 4 en un enfoque de detección de yemas por fuerza bruta a diferentes distancias. Si bien para la distancia más cercana se puede esperar que se mantengan los mismos resultados de precisión y sensibilidad presentados en el capítulo 4, el costo de realizar la operación resulta ser muy elevado. Por otro lado, al alejarse de la planta, el algoritmo de clasificación comienza a perder sensibilidad, la cual puede ser recuperada a costas de resignar precisión. Dada la imposibilidad que presenta este enfoque para establecer un equilibrio conveniente para la tarea de detección de yemas, surge la necesidad de desarrollar un algoritmo de BVA para encontrar las yemas en una planta de vid desde distancias más lejanas, donde el conocimiento del sentido común juega un rol clave en el proceso de planificación de vistas de la BVA. Este capítulo se centra principalmente en crear las herramientas necesarias para la construcción del conocimiento del sentido dentro del algoritmo de BVA. Esta construcción se realizó en el contexto del problema general de medición de variables vitícolas, como ser segmentación de madera, clasificación de yemas y detección de ramas. Cada una de estas herramientas tiene sus aplicaciones específicas, como ser medición de diámetro de tronco, conteo de yemas, fenotipado de plantas, longitud de brote, poda autónoma, y más. Sin embargo, en este capítulo estos algoritmos son utilizados como filtros que codifican el conocimiento del sentido común durante la planificación de vistas, de manera que permiten dirigir la búsqueda de yemas de manera activa.

El primer filtro presentado, el filtro de yema, corresponde al enfoque más simple respecto al conocimiento del problema que codifica durante la planificación: sugiere explorar una vista si existe al menos un parche de la vista con probabilidad de yema mayor a μ , donde el valor de μ puede ser alterado según la distancia de la vista a

la planta. Luego, siguiendo el sentido común, se presenta el filtro de rama-yema que introduce un nuevo algoritmo de detección de ramas y lo combina con los resultados del filtro anterior, el cual enriquece el conocimiento codificado: las yemas se desarrollan en las ramas y las ramas pueden ser detectadas desde distancias más lejanas que las que requieren las yemas. Por último, el filtro de rama-madera-yema avanza sobre una construcción del conocimiento del sentido común más elaborada que las anteriores: las ramas pertenecen al material madera, por lo tanto, al procesar una vista, todo aquello que ha sido detectado como rama también debe ser segmentado como madera. Como se demuestra en los resultados de la sección 5.3, esta construcción del sentido común responde de la manera esperada, mejorando los resultados de efectividad y eficiencia de la búsqueda en la medida que se codifica un conocimiento más elaborado y completo del problema.

Los resultados obtenido por los filtros propuestas en este capítulo en general logran un equilibrio más adecuado que el enfoque de fuerza bruta para la tarea de detección autónoma de yemas. Mientras que el enfoque de fuerza bruta tiene una efectividad igual a 1, presenta una eficiencia muy baja, ya que requiere visitar un total de 6912 vistas del nivel 0 para poder encontrar las 338 yemas dispuestas en la escena. Asumiendo que existe una yema por vista, este esquema visita inútilmente 6574 vistas, lo que representa un poco más del 95 % de las vistas totales. Por otro lado, en términos de eficiencia, el mejor resultado corresponde al filtro de rama-madera-yema, con un total de 2728 vistas visitadas en diferentes niveles de distancia, por lo que, asumiendo nuevamente que cada yema se encuentra exactamente en una vista, tiene una sobrecarga de 2390 vistas visitadas. Esto implica una ganancia en eficiencia de casi 64 % respecto a la sobrecarga del enfoque de fuerza bruta. Es importante destacar además la eficiencia en comparación con el caso de filtro nulo, i.e. aquel que no descarta ninguna vista durante la planificación. Este filtro también logra una efectividad de 1, pero la peor eficiencia posible, con un total de 9180 vistas visitadas en todos los niveles. En este caso, la ganancia en eficiencia es de un 70 %. En contraposición, para la mejor eficiencia posible, la efectividad del filtro rama-madera-yema es de 0,778, perdiendo un total de 75 yemas durante el proceso de planificación. Por otro lado, el filtro rama-yema se destaca al lograr una efectividad de 1 para el caso ($\mu_3 = 0; \mu_2 = 0; \mu_1 = 0; \mu_0 = 0$), lo que implica que el filtro de yema no está activado. En este caso la eficiencia es de 5722 vistas visitadas, lo que representa un 82 % de las vistas requeridas por el enfoque de fuerza bruta, manteniendo su efectividad. Por último, si se fija la efectividad esperada en aproximadamente 0,9, aparecen tres casos de interés: para una efectividad de 0,889 el filtro yema requiere visitar un total de 4101 vistas (40% y 55% de ganancia respecto a fuerza bruta y filtro nulo, respectivamente); el filtro rama-yema,

con una efectividad de 0,896, tiene una eficiencia de 3408 vistas visitadas (51% y 62% de ganancia respecto a fuerza bruta y filtro nulo, respectivamente); y finalmente el filtro rama-madera-yema logra una eficiencia de 3304 para una efectividad de 0,899 (52% y 64% de ganancia respecto a fuerza bruta y filtro nulo, respectivamente). En este análisis, el filtro rama-madera-yema se vuelve a destacar al obtener los mejores resultados de eficiencia para una valor determinado de efectividad.

Por otro parte, un análisis más fino sobre los resultados reportados en las figuras 5.15 y 5.16 de la sección 5.3, sugiere que sería posible esperar que los valores de eficiencia y efectividad del filtro yema para una asignación específica de μ_i sean mejorados mediante el filtro rama-yema con otra asignación para μ_i . Tal como muestra la figura 5.16, la curva del filtro de rama-yema se encuentra desplazada hacia la izquierda de la gráfica, lo que implica que para un mismo nivel de efectividad es posible mejorar la eficiencia con el filtro de rama-yema. De la misma manera, es esperable que se mantenga la misma situación para los resultados del filtro de rama-yema frente al filtro de rama-madera-yema. Esta intuición se puede verificar también en la tabla 5.1. La efectividad de ciertas asignaciones del filtro de yema puede ser igualada con asignaciones diferentes de los otros dos filtros, a la vez que mejoran el valor de eficiencia. Por ejemplo, el valor de efectividad y eficiencia del filtro de yema para la asignación ($\mu_3 = 0$; $\mu_2 = 0$; $\mu_1 = 0.4$; $\mu_0 = 0.4$) es de 0.964 y 5776. Sin embargo, el filtro ramamadera-yema logra una efectividad equivalente de 0.97 pero con una eficiencia de 4156 para la asignación ($\mu_3 = 0.3$; $\mu_2 = 0.3$; $\mu_1 = 0.3$; $\mu_0 = 0.3$).

Es posible evaluar el impacto de los resultados obtenidos para la búsqueda visual activa sobre el problema específico de detección de yemas combinándolos con los resultados de precisión y sensibilidad de algoritmo de clasificación de yemas tal como se propone en el capítulo 4. Tal como se ha planteado el problema en este capítulo, el proceso de detección de yemas se activa una vez alcanzado el nivel 0 exclusivamente sobre las vistas que sobrevivieron al proceso de planificación. Por lo tanto, es evidente que la cantidad de yemas detectadas en toda la escena estará supeditado por la efectividad de la búsqueda. Por ejemplo, el filtro de rama-madera-yema logra la efectividad más baja con un valor de 0,778, pero presenta la eficiencia más elevada con un total de 2728 vistas visitadas. Esto implica que la cantidad de yemas que han sobrevivido al proceso de búsqueda es de 263. Como se reporta en el capítulo 4, la sensibilidad esperada del clasificador de yemas durante una detección de tipo escaneo-ventana es de 0,89. Con este valor se puede esperar que de las 263 yemas que sobrevivieron a la búsqueda, solo sean detectadas 234, i.e. un 89%. Teniendo en cuenta que el escenario de evaluación contiene un total de 338 yemas, solo habrán sido detectadas un $69\,\%$ de las yemas existentes. Por otro lado, dada la precisión de 0,867 lograda por el clasificador del capítulo 4, se espera que un 13,3 % de los parches generados durante el proceso de escaneo-ventana para realizar la detección, sean incorrectamente clasificados como parches yema. Un supuesto proceso de escaneo-ventana para las 2728 vistas que sobrevivieron a la búsqueda podría arrojar un total de 122000 parches a clasificar, lo que implica una cantidad esperada de 17000 falsos positivos. En la sección 4.4 se discuten algunas maneras de mejorar estos resultados, argumentando que ciertas mejoras podrían elevar la sensibilidad del clasificador hasta a un 0,977, mientras que la precisión podría llegar hasta 0,96. Sin embargo, en la práctica incluso una precisión de 0,99 puede resultar una dificultad, produciendo 1200 falsos positivos para el ejemplo presentado. Sin duda este problema se presenta como uno de los aspectos más desafiantes e interesantes a resolver en el futuro. Las herramientas para construir el conocimiento del sentido común en este capítulo podrían utilizarse durante el proceso de detección, ayudando a mejorar aún más la precisión del clasificador de la misma forma en que ayudan a mejorar la eficiencia de la búsqueda, i.e. descartando falsos positivos mediante la aplicación de los filtros rama-yema o rama-madera-yema una vez realizada la clasificación de los parches.

A continuación se analizan algunos aspectos del procedimiento realizado para la adquisición de las imágenes requeridas en la experimentación de este capítulo, el cual presenta algunas particularidades que vale la pena discutir. Tal como se describe en la sección 5.2.1, fue necesario construir una estructura que limita el movimiento de la cámara a dos grados de libertad a fin de capturar las imágenes de forma perpendicular al plano de la hilera, aproximadamente a unos 12 centímetros de dicho plano, sobre todo el espacio que ocupa la planta. De esta manera, en este procedimiento se imponen fuertes restricciones que no existen durante la creación del corpus del capítulo 4. En el capítulo anterior la imágenes fueron adquiridas por operarios humanos sin ninguna restricción de movimiento, lo que permitió resoluciones de parches de yemas entre $100 \times 100 \text{ y}$ $1600 \times 1600 \text{ pixeles}$, con una orientación adecuada de la cámara frente a la yema, y en general una mejor calidad de enfoque y definición. Es posible determinar el impacto que acarrea esta diferencia durante la captura de las imágenes a partir de los resultados presentados en la sección 5.1 para el algoritmo de clasificación de yemas del capítulo 4 entrenado sobre el nuevo corpus de este capítulo. Tal como se reporta en la sección, el clasificador de este capítulo logra sensibilidad y precisión de 0,847 y 0,878 respectivamente. Comparado con los resultados del capítulo anterior, la sensibilidad del clasificador de este capítulo es inferior en 0,043, aunque la precisión logra una pequeña mejora de 0,018. Si bien el deterioro respecto a la sensibilidad no es muy elevado, esta diferencia en la calidad de las imágenes durante la generación de ambos corpus altera efectivamente los resultados de sensibilidad y precisión del

clasificador. Por lo tanto, en trabajos futuros sería esperable mejorar los resultados de búsqueda y detección quitando las restricciones, lo que permitiría mayor libertad en los movimientos de la cámara. Sin duda, este problema requiere un algoritmo de planificación de vistas que tome decisiones mucho más complejas durante la estimación de la próxima vista a explorar.

Por otro lado, existen algunos aspectos del proceso de planificación de vistas que se discuten a continuación. Una de las limitaciones del enfoque presentado en este capítulo es que no tiene en cuenta ordenamiento de vistas. En la práctica, esta capacidad puede ser necesaria si se intenta minimizar la distancia total recorrida o el tiempo de vuelo en movimiento del VANT. Para esto, es evidente que se requiere un componente específico que permita determinar el ordenamiento óptimo para un conjunto de vistas a explorar. Por ejemplo, se puede considerar el caso en el que hay un conjunto de vistas a explorar que se agrupan a lo largo de dos ramas de la escena. Una forma eficiente de recorrerlas es desplazándose linealmente sobre las vistas de una rama y luego sobre la otra, en vez de hacer un escaneo sobre una grilla sobre impuesta y sin ningún ordenamiento de vistas. Esta situación cae dentro de los problemas de optimización NP-duro (puede ser considerado un caso del problema del viajante). Por lo tanto, puede resultar complejo de resolver en la práctica pero sin dudas se presenta como un desafío interesante para ser tenido en cuenta en trabajos futuros. Por otro lado, en este trabajo el planificador de vista no toma ninguna decisión sobre la condición de finalización de la búsqueda, momento a partir de cual debería entrar en acción el algoritmo de detección de yemas. Por el contrario, el algoritmo finaliza cuando no existen más vistas disponibles, i.e. cuando alcanzó el nivel 0, límite que está determinado por el escenario de evaluación propuesto. En la práctica, la condición debería estar dada por la distancia actual de cámara a los elementos en frente de la escena. Una alternativa es equipar al agente con sensores de proximidad. Esta opción, además de elevar el costo de la herramienta y demandar mayor cantidad de recursos de cómputo, es contraria a la finalidad de este trabajo, en donde se espera que toda la tarea sea realizada exclusivamente a través de imágenes obtenidas con una sola cámara. Por lo tanto, este problema se presenta como uno de los aspectos más interesantes para el desarrollo de agentes autónomos monoculares de bajo costo. Al respecto, en la literatura se presentan variados enfoques para estimar profundidades a partir de imágenes capturadas con una única cámara. Concretamente, en este problema sería viable obtener información de profundidad a partir de secuencias de imágenes tomadas desde diferentes perspectivas de la escena [47, 132] o incluso utilizando diferentes valores de amplitud focal [166, 70].

Una de las limitaciones más importantes del trabajo presentado en este capítulo

pasa por las simplificaciones introducidas para la construcción del escenario de evaluación. En primer instancia, realizar el proceso de BVA sobre las 108 vistas del nivel 3 es equivalente a ejecutar el proceso sobre cada una de las vistas individuales, avanzando de forma telescópica hacia las vistas de nivel 0. Con esto, el proceso de planificación no puede tener en cuenta información relevante de las vistas vecinas, que podría modificar la respuesta del filtro para la vista actual. Este escenario presenta otro obstáculo para el desarrollo de un planificador más elaborado, ya que no permite establecer de manera obvia un mecanismo para calcular otras medidas de calidad interesante, como se distancia total recorrida y tiempo de vuelo en movimiento. Si bien el enfoque presentado aquí no tiene en cuenta estas medidas, en trabajos futuros sería importante poder estimar sus valores. Otro problema que sufre este escenario surge del proceso de división de vistas para realizar los acercamientos telescópicos, en donde las vistas son particionadas de manera tal que no existe superposición en las vistas del siguiente nivel. En este proceso, podría perderse información relevante para los filtro, por ejemplo quedando una yema repartida en cuatro cuadrantes. Este problema pone de manifiesto otra limitación importante: la subdivisión se realiza particionando la vista en cuadrantes iguales. Una mejora obvia para este enfoque consiste en posibilitar que los acercamientos telescópicos se realicen sobre áreas de la vista sin restricciones, permitiendo que el algoritmo de planificación decida, por ejemplo, en cuántas zonas dividir la vista, o hacia donde dirigir la atención de la búsqueda según la densidad de píxeles activos de la máscara. En la práctica, resolver este problema no es sencillo ya que requiere elegir las nuevas vistas a explorar entre un continuo de posibilidades. En trabajos futuros se hace evidente la necesidad de mejorar este escenario de evaluación, a fin de contar con herramientas más robustas para validar el funcionamiento de los enfoque de planificación propuestos aquí. Sin embargo, la intención principal de este escenario fue argumentar que la idea general tiene tiene potencial, y que mejoras posteriores no pueden hacer más que mejorar los resultados presentados.

Respecto a la utilidad de los sistemas de BVA para otros problemas de la viticultura, es posible vislumbrar algunas aplicaciones concretas: (i) detección de racimos con presencia de oclusión para estimar diferentes medidas de interés, como ser cantidad de racimos, tamaño de racimo, peso de racimo, y más; (ii) estimación precisa del área foliar expuesta de una planta de vid, i.e. hojas que se encuentran en la capa exterior de la planta; (iii) detección de racimos en periodo de floración con presencia de oclusión y conteo de flores del racimo para la estimación temprana de la producción; (v) longitud de brote, peso de poda, distancia de entrenudos, y más, con aplicación en la estimación del crecimiento de la planta para la temporada próxima; entre otros. Finalmente, existen algunas vías generales para continuar mejorando los resultados de

este capítulo. Por un lado, es posible incorporar nuevo conocimiento de sentido común a través de la algoritmos que permitan obtener mayor cantidad de información visual de la escena, como ser extracción de características visuales globales, segmentación de texturas, detección de zonas fuera de foco, codificación de relaciones espaciales entre clases de objetos de la escena, y más.

5.5. Conclusiones

En este capítulo se han presentado diversos enfoque para codificar el conocimiento de sentido común durante la planificación de vistas en un algoritmo de BVA para el problema de detección de yemas de una planta de vid en condiciones naturales de campo. La construcción de este conocimiento se realiza a través de herramientas que además presentan potenciales aplicaciones en la tarea de automatización de medición de variables vitícolas, como ser los algoritmos de los capítulos 3 y 4, y un nuevo algoritmo de detección de ramas desarrollado en este capítulo. Concretamente, se presenta un estudio para diferentes enfoques que permiten planificar qué zonas de la escena explorar a sobre vistas capturadas a diferentes distancias de una planta, procurando descartar lo antes posible aquellas zonas donde no se espera que existen yemas, bajo la intuición de que cuanto mayor sea la distancia en la que se descarta una zona sin yema, más eficiente será la búsqueda. La construcción del sentido común propuesta en este capítulo, se valida sobre un escenario controlado, diseñado para poder evaluar la eficiencia y la efectividad de la búsqueda de yemas. Esta construcción del conocimiento de sentido común mejora los resultados de efectividad y eficiencia de la búsqueda.

En términos de eficiencia, el mejor resultado corresponde a una planificación que requiere un total de 2728 vistas visitadas, lo que implica una ganancia en eficiencia de casi 64 % respecto al enfoque de fuerza bruta. En contraposición, para esta ganancia en eficiencia, la efectividad de la búsqueda cae a 0,778, perdiendo un total de 75 yemas durante el proceso de planificación. Por otro lado, el filtro rama-yema logra una efectividad de 1 con eficiencia de 5722 vistas visitadas, lo que representa un 82 % de las vistas requeridas por el enfoque de fuerza bruta, manteniendo su efectividad. Si se fija efectividad esperada aproximadamente a 0,9, aparecen una serie de casos que dan diferentes valores de efectividad: para una efectividad de 0,889 el filtro yema requiere visitar un total de 4101 vistas; el filtro rama-yema, con una efectividad de 0,896, tiene una eficiencia de 3408 vistas visitadas; y finalmente el filtro rama-madera-yema logra una eficiencia de 3304 para una efectividad de 0,899.

Por último, se discuten los alcances y limitaciones de los diferentes filtros propues-

tos para realizar el proceso de planificación de vistas, interpretando sus resultados en el contexto del problema de BVA de yemas para la detección de yemas de una planta de vid, presentando además algunas vías de mejora y otras posibles aplicación de las herramientas desarrolladas aquí.

Capítulo 6

Conclusiones Generales y Trabajos Futuros

El aporte principal de esta tesis consiste en el estudio, implementación y desarrollo de algoritmos de segmentación, clasificación y detección de objetos en imágenes que establecen las bases para la construcción preliminar de un sistema de búsqueda visual activa (BVA) para yemas de vid que cumple con los siguientes requerimientos: (i) las imágenes son tomadas mediante una cámara RGB en condiciones naturales de campo, durante la temporada de invierno, cuando las plantas han perdido sus hojas y frutos, a fin de evitar la oclusión de las yemas; (ii) el proceso de búsqueda comienza en vistas donde las yemas tienen una resolución menor a 20×20 píxeles, y continúa realizando acercamientos hacia las zonas donde es probable que se encuentren yemas, hasta que la distancia entre la cámara y la yema sea suficiente para ser detectada de manera precisa y poder estimar de forma autónoma sus características geométricas y visuales; (iii) la elección de la siguiente vista a explorar se determina por un proceso de planificación de vistas sobre la base del conocimiento de sentido común del problema codificado en el algoritmo; (iv) la navegación de las vistas está restringida sólo a movimientos lineales respecto al plano que forma la hilera del viñedo, i.e. la cámara que captura las imágenes solo posee 3 grados de libertad (movimientos horizontales, verticales y acercamientos).

Este aporte se justifica en la necesidad de mejorar los instrumentos actuales para la medición de variables agrícolas, esperando que a través del desarrollo de herramientas de medición autónoma se pueda aumentar la resolución espacial en la adquisición de datos y reducir el costo de la tecnología de medición, con un impacto directo en la mejora de los modelos de predicción de variables que intentan estimar la productividad y calidad del cultivo. Con el propósito de verificar que la escasez de datos afecta negativamente la calidad de las estimaciones en los modelos de interpolación espacial,

en el capítulo 2 se presentó un estudio para técnicas de interpolación estadísticas y de aprendizaje de máquinas sobre un conjunto de variables vitícolas medidas en diversos viñedos de la provincia de Mendoza, cuyos datos fueron suministradas por la empresa Xoomcode S.A., especializada en servicios de geolocalización. La comparación de cada técnica se realiza sobre la base de la calidad de los modelos de interpolación obtenidos y las limitaciones en el modelado de los patrones espaciales que sufren frente a la falta de datos. Entre los métodos de interpolación evaluados se encuentran ponderación de distancia inversa, kriging, máquinas de vector soporte y máquinas de vector soporte local (MVSL). Sobre un total de 25 conjuntos de datos, MVSL se destacó como el de mejor calidad en un 68 % de los casos según los valores reportados para la media del error absoluto. Además, los resultados experimentales mostraron que MVSL también reduce la desviación estándar del error absoluto y la raíz del error cuadrático medio, situación deseable en problemas de interpolación, ya que valores bajos para estas medidas indican estimaciones más precisas para los datos del conjunto de prueba, lo que implica un mejor estimador. Por último, el análisis basado en curvas de aprendizaje sobre los 25 conjuntos de datos disponibles indicó que al menos 7 de estos modelos podrían mejorar directamente su calidad de predicción aumentando la cantidad de datos. Por lo tanto, se hace evidente la necesidad de desarrollar nuevas herramientas de medición autónoma de variables que permitan mejorar la calidad de los modelos de interpolación con un impacto real en la estimación de la calidad y productividad del cultivo.

Para atender esta necesidad, este trabajo se orientó al desarrollo de algoritmos que buscan fundar las bases para un sistema de búsqueda visual activa de yemas de vid en condiciones naturales de campo durante la temporada de invierno. El foco principal de este desarrollo se encauza hacia la creación de herramientas que permitieron integrar en el proceso de planificación de vistas el conocimiento del sentido común para el problema de búsqueda autónoma de yemas de vid, a saber: las yemas se desarrollan en las ramas (o sarmientos); las ramas pertenecen a la clase de material madera de la vid; el resto de los elementos de la escena pueden ser omitidos durante la búsqueda. En esta tesis, el conocimiento de sentido común se construyó a partir del desarrollo de algoritmos de segmentación de madera, detección de ramas y detección de yemas de vid. Sin embargo estas herramientas por sí solas tienen potenciales aplicaciones para la medición autónoma de otras variables vitícolas. Por lo tanto, en los capítulos 3 y 4 se presentaron dos de los algoritmos requeridos para construir el conocimiento de sentido común, i.e. segmentación de madera y clasificación de imágenes para detección de yemas, discutiendo sus resultados independientemente en el contexto de sus aplicaciones para la medición autónoma de variables. Por otro lado, los detalles del

algoritmo de detección de ramas se postergaron hasta el capítulo 5, donde además se ensamblan todas las herramientas para resolver el problema específico de búsqueda visual activa de yemas de vid.

En el capítulo 3 se presentaron varios enfoques de segmentación de madera sobre imágenes de viñedos basados en los algoritmos de mezcla de gaussianas (MG) y máquinas de vector soporte (MVS). Además, inspirado en trabajos de reconocimiento de texturas para segmentación de piel, se introdujo un enfoque sencillo que mejora los resultados de los algoritmos MG y MVS mediante un proceso de reclasificación basado en información espacial local, llamados MG-RA y MVS-RA. Comparando MG y MVS se puede observar que en general MVS obtiene mejores resultados para la clase madera, con exactitud de 0,852, precisión de 0,718, sensibilidad de 0,806, lo que produce un valor de medida-F de 0,751, superior al valor de medida-F de MG por un margen pequeño de 0,024. El análisis sobre MG-RA y MVS-RA revela mejoras considerables sobre las técnicas anteriores para madera. Particularmente, MG-RA logra una exactitud de 0,898, precisión de 0,96 y sensibilidad de 0,681 lo que provoca un valor de medida-F de 0.791. Además, los algoritmos introducidos en este capítulo son contrastados sobre un caso de uso potencial de estas tecnologías: medición del diámetro de tronco en vides a partir de imágenes. Estos resultados incentivan el estudio y la investigación de otras posibles aplicaciones en viticultura, como ser: área foliar, índice verde, conteo de yemas, conteo de flores, conteo de racimos por planta, conteo de racimos por brote, entre otros. Además, estos resultados también se discutieron en el contexto de los algoritmos de detección resaltando su utilidad para codificar el conocimiento de sentido común para el problema específico de detección de yemas de vid en imágenes naturales de viñedos.

En el capítulo 5 se presenta un algoritmo de clasificación de imágenes de yemas de vid a partir de imágenes tomadas en viñedos en su ambiente natural. Los requisitos del algoritmo de clasificación están especificados para ser utilizado en imágenes producidas normalmente por algoritmos de detección escaneo-ventana. El enfoque propuesto logró una sensibilidad de clasificación superior a 0,89 en parches que contienen al menos un 60 % de los píxeles de la yema original, donde la proporción de píxeles yema en el parche es mayor a 20 %, y la yema tiene al menos 100 píxeles de diámetro. En particular, el mejor valor de sensibilidad de clasificación fue de 0,977, para parches que contienen entre 90 % y un 100 % de los píxeles yema, y estos píxeles representan entre 20 % y un 30 % del parche, i.e. parches de tres a cinco veces más grande que la yema. En cuanto a los resultados de precisión de clasificación, el clasificador alcanzó un valor de 0,86, i.e. 14 % de los parches no-yema en el corpus dieron lugar a falsos positivos. Otra conclusión importante de este trabajo es que para la detección, es posible reducir la

tasa de falsos positivos entre un 4 % y un 8 %, ya que una gran parte de estos errores provienen de parches en las proximidades de la yema (i.e. nodos y cercanías de la yema) y parches de categorías poco representadas (i.e. hojas secas, racimos secos, y tronco con corteza), y no debería producir errores en la detección.

En el capítulo 5 se introduce un algoritmo de BVA para el problema de detección de yemas de una planta de vid en condiciones naturales de campo. Uno de los aspectos más importantes de este algoritmo consiste en la codificación del conocimiento de sentido común, utilizado durante la planificación de vistas. La construcción de este conocimiento se realizó a través de las herramientas introducidas en los capítulos 3 y 4, junto a un nuevo algoritmo de detección de ramas desarrollado en el capítulo 5. La construcción del sentido común propuesta en este capítulo, se valida sobre un escenario controlado, diseñado para poder evaluar la eficiencia y la efectividad de un vehículo aéreo no tripulado (VANT) para la búsqueda de yemas. Esta construcción del conocimiento de sentido común mejora los resultados de efectividad y eficiencia de la búsqueda. En términos de eficiencia, el mejor resultado corresponde a una planificación que requiere un total de 2728 vistas visitadas, lo que implica una ganancia en eficiencia de casi 64 % respecto al enfoque de fuerza bruta, acompañada de una efectividad de la búsqueda de 0,778, perdiendo un total de 75 yemas durante el proceso de planificación. Por otro lado, el filtro rama-yema logra una efectividad de 1 con eficiencia de 5722 vistas visitadas, lo que representa un 82 % de las vistas requeridas por el enfoque de fuerza bruta, manteniendo su efectividad. Por último, si se fija efectividad esperada aproximadamente a 0,9, aparecen tres casos de interés: para una efectividad de 0,889 el filtro yema requiere visitar un total de 4101 vistas (40 % y 55 % de ganancia respecto a fuerza bruta y filtro nulo, respectivamente); el filtro rama-yema, con una efectividad de 0.896, tiene una eficiencia de 3408 vistas visitadas (51 % y 62 % de ganancia respecto a fuerza bruta y filtro nulo, respectivamente); y finalmente el filtro rama-madera-yema logra una eficiencia de 3304 para una efectividad de 0.899 (52% y 64% de ganancia respecto a fuerza bruta y filtro nulo, respectivamente).

A continuación se presentan las conclusiones generales para el aporte principal de la tesis en un contexto más amplio que permite vislumbrar cuales son las debilidades que sufre el enfoque de BVA propuesto, como así también posibles vías de mejora y nuevas oportunidades de investigación que definen los posibles trabajos futuros.

Respecto a las limitaciones del trabajo presentado en esta tesis, en primera instancia aparece la necesidad de mejorar el escenario de evaluación para el algoritmo de BVA, de manera que permita validar el funcionamiento de los enfoques de planificación de una forma más general, a la vez que admite enfoques de planificación de mayor complejidad. Específicamente, un escenario de evaluación mejorado consiste en

el desarrollo de un agente de búsqueda basado en un VANT que permita explorar el espacio sin restricciones. En principio, para llevar esta propuesta a la práctica se requiere el desarrollo de complejos sistemas, como ser sistemas de navegación, estabilización, ubicación geoespacial, administración de energía, manejo de colisiones y más. Sin embargo, este esquema admite simplificaciones que hacen viable su implementación como entorno de validación para los algoritmos de BVA en su etapa de investigación y desarrollo. Por ejemplo, en esta etapa se podrían omitir el desarrollo de sistemas para la administración de energía, ubicación geoespacial y manejo de colisiones. Con las oportunidades de este nuevo escenario, surgen nuevas posibilidades de desarrollo para los sistemas de BVA, como ser: planificación de vistas con información global obtenida a partir de vistas vecinas; cálculo del número óptimo de vistas a explorar en el siguiente paso; determinación del ordenamiento óptimo para un conjunto de vistas a explorar; integración de condiciones de finalización de la búsqueda; incorporación de mecanismos para la evaluación de otras medidas de calidad interesante, como se distancia total recorrida y tiempo de vuelo en movimiento; entre otros.

Por lo tanto, se puede concluir que aún se requiere una gran cantidad de trabajo y esfuerzo para lograr una aplicación práctica real del enfoque para BVA de yemas propuesto en esta tesis, aunque los avances reportados aquí han demostrado ser efectivos y robustos a ciertos aspectos de la práctica, como ser variabilidad en la calidad de las imágenes de los corpus y variabilidad respecto a los parámetros que definen la detección de yemas de tipo escaneo-ventana. Concretamente, para alcanzar una aplicabilidad práctica todavía es necesario: (i) probar el sistema de BVA en un VANT real; (ii) permitir un régimen de distancias a la planta menos restrictivo, incluso contínuo; (iii) reducir el altísimo número de falsos positivos que presenta el clasificador de yemas durante la detección; (iv) comenzar la búsqueda a distancias más alejadas de la planta, incorporando nuevo conocimiento de sentido común, como ser segmentación de la textura específica de ramas, estructura espacial del tronco, y más; (v) determinar distancias a partir de imágenes tomadas con una sola cámara; (vi) incorporar condiciones de terminación en el proceso de búsqueda; (vii) desarrollar sistemas de navegación, estabilización, ubicación geoespacial, administración de energía, manejo de colisiones y más.

Además de una implementación práctica del VANT, se considera extender este trabajo para otros problemas de la viticultura. En este sentido es posible vislumbrar algunas aplicaciones concretas que se presentan como opciones viables para desarrollar en trabajo futuros: (i) detección de racimos con presencia de oclusión para estimar diferentes medidas de interés, como ser cantidad de racimos, tamaño de racimo, peso de racimo, y más; (ii) estimación precisa del área foliar expuesta de una planta de

vid, i.e. hojas que se encuentran en la capa exterior de la planta; (iii) detección de racimos en periodo de floración con presencia de oclusión para el conteo de flores del racimo con impacto en la estimación temprana de la producción; (iv) longitud de brote, peso de poda, distancia de entrenudos, y más, con aplicación en la estimación del crecimiento de la planta para la temporada próxima; (v) fenotipado de yemas a partir de su detección y posterior clasificación de su estadio de desarrollo; (vi) reconstrucción 3D de la planta; entre otros.

A modo de conclusión final, este trabajo permite sentar las bases para el desarrollo de un sistema de búsqueda visual activa de yemas de vid y otras variables de interés vitícola, a través de la construcción del conocimiento del sentido común del problema durante el proceso de planificación de vistas.

Bibliografía

- [1] Abdiansah Abdiansah and Retantyo Wardoyo. Time complexity analysis of support vector machines (svm) in libsvm. *International Journal Computer and Application*, 2015.
- [2] Rehan Akbani, Stephen Kwek, and Nathalie Japkowicz. Applying support vector machines to imbalanced datasets. In *Machine learning: ECML 2004*, pages 39–50. Springer, 2004.
- [3] Gustavo Aliquó, Aníbal Catania, and Germán Aguado. La poda de la vid. *Instituto Nacional de Tecnología Agropecuaria. Estación Experimental Agropecuaria Mendoza*, http://www. inta. gov. ar, 2010.
- [4] Sharolyn Anderson. An evaluation of spatial interpolation methods on air temperature in phoenix, az. *De-partment of Geography, Arizona State University*, 2002.
- [5] Arturo Aquino, Borja Millan, Daniel Gaston, María-Paz Diago, and Javier Tardaguila. vitisflower®: development and testing of a novel android-smartphone application for assessing the number of grapevine flowers per inflorescence using artificial vision techniques. Sensors, 15(9):21204–21218, 2015.
- [6] Arturo Aquino, Borja Millan, Salvador Gutiérrez, and Javier Tardáguila. Grapevine flower estimation by applying artificial vision techniques on images with uncontrolled scene and multi-model analysis. *Computers and Electronics in Agriculture*, 119:92–104, 2015.
- [7] J Arnó Satorra, JA Martínez Casasnovas, M Ribes Dasi, and JR Rosell Polo. Review. precision viticulture. research topics, challenges and opportunities in site-specific vineyard management. Spanish Journal of Agricultural Research, 2009, vol. 7, núm. 4, p. 779-790, 2009.
- [8] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.
- [9] Alper Aydemir, Andrzej Pronobis, Moritz Göbelbecker, and Patric Jensfelt. Active visual object search in unknown environments using uncertain semantics. *IEEE Transactions on Robotics*, 29(4):986–1002, 2013.

- [10] Alper Aydemir, Kristoffer Sjöö, John Folkesson, Andrzej Pronobis, and Patric Jensfelt. Search in the real world: Active visual object search based on spatial relations. In Robotics and Automation (ICRA), 2011 IEEE International Conference on, pages 2818–2824. IEEE, 2011.
- [11] Marco A Azpurua and K Dos Ramos. A comparison of spatial interpolation methods for estimation of average electromagnetic field magnitude. *Progress In Electromagnetics Research M*, 14:135–145, 2010.
- [12] Terry Bates and Justin Morris. Mechanical cane pruning and crop adjustment decreases labor costs and maintains fruit quality in new york 'concord'grape production. *HortTechnology*, 19(2):247–253, 2009.
- [13] C Bauckhage and K Kersting. Data mining and pattern recognition in agriculture. KI-Künstliche Intelligenz, 27(4):313–324, 2013.
- [14] Avital Bechar and Clément Vigneault. Agricultural robots for field operations: Concepts and components. *Biosystems Engineering*, 149:94–111, 2016.
- [15] Ron Berenstein, Ohad Ben Shahar, Amir Shapiro, and Yael Edan. Grape clusters and foliage detection algorithms for autonomous selective vineyard sprayer. *Intelligent Service Robotics*, 3(4):233–243, 2010.
- [16] Bettina Berger, Bas de Regt, and Mark Tester. High-throughput phenotyping of plant shoots. *High-Throughput Phenotyping in Plants: Methods and Protocols*, pages 9–20, 2012.
- [17] G Billikopf, M Norton, et al. Pay method affects vineyard pruner performance. *California Agriculture*, 46(5):12–13, 1992.
- [18] Christopher M. Bishop. Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
- [19] CM Bishop. Pattern recognition and machine learning. Springer, 2009.
- [20] Simon Blackmore. Toward robotic agriculture. Resource Magazine, 22(2):27–27, 2015.
- [21] J Blasco, N Aleixos, and E Moltó. Machine vision system for automatic quality grading of fruit. *Biosystems engineering*, 85(4):415–423, 2003.
- [22] R Bongiovanni and J Lowenberg-DeBoer. Precision agriculture in argentina. Modern Agr.: The J. for Site-Specific Crop Management, 2002.
- [23] Antoine Bordes, Seyda Ertekin, Jason Weston, and Léon Bottou. Fast kernel classifiers with online and active learning. *Journal of Machine Learning Research*, 6(Sep):1579–1619, 2005.
- [24] L Bottou and V Vapnik. Local learning algorithms. *Neural computation*, 4(6):888–900, 1992.

- [25] RGV Bramley. Lessons from nearly 20 years of precision agriculture research, development, and adoption as a guide to its appropriate application. *Crop and Pasture Science*, 60(3):197–217, 2009.
- [26] F Bromberg and DS Pérez. Interpolación espacial mediante aprendizaje de máquinas en vinedos de la provincia de mendoza, argentina. In 13th Argentine Symposium on Artificial Intelligence, ASAI, volume 41, 2012.
- [27] J Bustos and José Ricardo. Inteligencia artificial en el sector agropecuario. Seminario de Investigación I. Universidad Nacional de Colombia. Colombia, 2005.
- [28] C Campbell. An introduction to kernel methods. Studies in Fuzziness and Soft Comptuing, 66:155–192, 2001.
- [29] John Canny. A computational approach to edge detection. *IEEE Transactions* on pattern analysis and machine intelligence, (6):679–698, 1986.
- [30] Peter Carbonetto, Nando de Freitas, and Kobus Barnard. A statistical model for general contextual object recognition. In *Computer Vision-ECCV 2004*, pages 350–362. Springer, 2004.
- [31] KG Cassman. Ecological intensification of cereal production systems: yield potential, soil quality, and precision agriculture. *Proceedings of the National Academy of Sciences*, 96(11):5952–5959, 1999.
- [32] M Castelan-Estrada, P Vivin, and JP Gaudillière. Allometric Relationships to Estimate Seasonal Above-ground Vegetative and Reproductive Biomass of Vitis vinifera L. *Annals of Botany*, 89(4):401–408, 2002.
- [33] DR Causton. Biometrical, structural and physiological relationships among tree parts. Attributes of trees as crop plants. Edited by MGR Cannel and JE Jackson. Titus Wilson & Son Ltd.. Cumbria, Great Britain, pages 137–159, 1985.
- [34] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology (TIST), 2(3):27, 2011.
- [35] Kang-tsung Chang. Introduction to geographic information systems. McGraw-Hill Higher Education Boston, 2006.
- [36] Qiang Chen, Zheng Song, Jian Dong, Zhongyang Huang, Yang Hua, and Shuicheng Yan. Contextualizing object detection and classification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(1):13–27, 2015.
- [37] YR Chen, K Chao, and MS Kim. Machine vision technology for agricultural applications. *Computers and electronics in Agriculture*, 36(2):173–191, 2002.
- [38] HD Cheng, XH Jiang, Y Sun, and J Wang. Color image segmentation: advances and prospects. *Pattern recognition*, 34(12):2259–2281, 2001.

- [39] Jean-Paul Chiles and Pierre Delfiner. Geostatistics: modeling spatial uncertainty, volume 497. John Wiley & Sons, 2009.
- [40] Marc Claesen, Frank De Smet, JA Suykens, and Bart De Moor. Fast prediction with svm models containing rbf kernels, arxiv preprint. arXiv preprint arXiv:1403.0736, 2014.
- [41] Sam Corbett-Davies, Tom Botterill, Richard Green, and Valerie Saxton. An expert system for automatically pruning vines. In *Proceedings of the 27th Conference on Image and Vision Computing New Zealand*, pages 55–60. ACM, 2012.
- [42] N Cressie. The origins of kriging. Mathematical Geology, 22(3):239–252, 1990.
- [43] N Cressie. Statistics for spatial data. Terra Nova, 4(5):613-617, 1992.
- [44] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision*, *ECCV*, volume 1, pages 1–2. Prague, 2004.
- [45] Jnaneshwar Das, Gareth Cross, Chao Qu, Anurag Makineni, Pratap Tokekar, Yash Mulgaonkar, and Vijay Kumar. Devices, systems, and methods for automated monitoring enabling precision agriculture. In 2015 IEEE International Conference on Automation Science and Engineering (CASE), pages 462–469. IEEE, 2015.
- [46] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM, 2006.
- [47] Andrew J Davison, Ian D Reid, Nicholas D Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE transactions on pattern analysis and machine intelligence*, 29(6), 2007.
- [48] Maria P Diago, Andres Sanz-Garcia, Borja Millan, Jose Blasco, and Javier Tardaguila. Assessment of flower number per inflorescence in grapevine by image analysis under field conditions. *Journal of the Science of Food and Agriculture*, 94(10):1981–1987, 2014.
- [49] Maria-Paz Diago, Christian Correa, Borja Millán, Pilar Barreiro, Constantino Valero, and Javier Tardaguila. Grapevine yield and leaf area estimation using supervised classification methodology on rgb images taken under field conditions. Sensors, 12(12):16988–17006, 2012.
- [50] S Dimitriadis and C Goumopoulos. Applying machine learning to extract new knowledge in precision agriculture applications. In *Informatics*, 2008. PCI'08. Panhellenic Conference on, pages 100–104. IEEE, 2008.
- [51] Santosh K Divvala, Derek Hoiem, James H Hays, Alexei A Efros, and Martial Hebert. An empirical study of context in object detection. In *Computer Vision and Pattern Recognition*, 2009. CVPR 2009. IEEE Conference on, pages 1271–1278. IEEE, 2009.

- [52] RO Duda, PE Hart, and DG Stork. *Pattern classification*. John Wiley & Sons, 2012.
- [53] T Fawcett. Roc graphs: Notes and practical considerations for researchers. *Machine Learning*, 31:1–38, 2004.
- [54] Björn Fröhlich, Erik Rodner, Michael Kemmler, and Joachim Denzler. Large-scale gaussian process multi-class classification for semantic segmentation and facade recognition. *Machine vision and applications*, 24(5):1043–1053, 2013.
- [55] Ming Gao and Tien-Fu Lu. Image processing and analysis for autonomous grapevine pruning. In *Mechatronics and Automation*, *Proceedings of the 2006 IEEE International Conference on*, pages 922–927. IEEE, 2006.
- [56] Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58, 1992.
- [57] E Gil, Arnó J, Llorens J, Sanz R, Llop J, Rosell-Polo JR, and Escolà MGA. Advanced technologies for the improvement of spray application techniques in spanish viticulture: An overview. *Sensors*, 14:691–708, 2014.
- [58] N Gilardi and S Bengio. Local machine learning models for spatial data analysis. Journal of Geographic Information and Decision Analysis, 4(1):11–28, 2000.
- [59] JN Goetz, A Brenning, H Petschko, and P Leopold. Evaluating machine learning and statistical prediction techniques for landslide susceptibility modeling. Computers & Geosciences, 81:1–11, 2015.
- [60] JFS Gomes and FR Leta. Applications of computer vision techniques in the agriculture and food industry: a review. *European Food Research and Technology*, 235(6):989–1000, 2012.
- [61] Pierre Goovaerts. Geostatistics for natural resources evaluation. Oxford University Press on Demand, 1997.
- [62] Stephen Gould, Jim Rodgers, David Cohen, Gal Elidan, and Daphne Koller. Multi-class segmentation with relative location prior. *International Journal of Computer Vision*, 80(3):300–316, 2008.
- [63] RM Haralick and LG Shapiro. Computer and robot vision, volume 1. Addison-Wesley, 1992.
- [64] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Citeseer, 1988.
- [65] Anja Hartmann, Tobias Czauderna, Roberto Hoffmann, Nils Stein, and Falk Schreiber. Htpheno: an image analysis pipeline for high-throughput plant phenotyping. *BMC bioinformatics*, 12(1):148, 2011.
- [66] TJ Hastie, RJ Tibshirani, and JH Friedman. The elements of statistical learning: data mining, inference, and prediction. Springer, 2011.

- [67] Geremy Heitz and Daphne Koller. Learning spatial context: Using stuff to find things. In *Computer Vision–ECCV 2008*, pages 30–43. Springer, 2008.
- [68] Katja Herzog, Ribana Roscher, M Wieland, A Kicherer, Thomas Läbe, Wolfgang Förstner, Heiner Kuhlmann, and Reinhard Töpfer. Initial steps for high-throughput phenotyping in vineyards. VITIS-Journal of Grapevine Research, 53(1):1, 2015.
- [69] L Hidalgo Fernandez-Cano and J Higalgo Togores. Tratado de viticultura. Mundi prensa, 2011.
- [70] Shinsaku Hiura and Takashi Matsuyama. Depth measurement by the multifocus camera. In Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on, pages 953–959. IEEE, 1998.
- [71] Li Hongqin, Wu Jianzhen, Zhang Liping, and Ning Jun. Real-time digital image exposure status detection and circuit implementation. *International Journal of Advanced Computer Science and Applications*, 6(5):19–24, 2015.
- [72] AB Iandolino, RW Pearcy, and LE Williams. Simulating three-dimensional grapevine canopies and modelling their light interception characteristics. *Australian Journal of Grape and Wine Research*, 19(3):388–400, 2013.
- [73] Katsushi Ikeuchi. Computer Vision: A Reference Guide. Springer Publishing Company, Incorporated, 2014.
- [74] Edward H Isaaks and R Mohan Srivastava. Applied geostatistics: Oxford university press. *New York*, page 561, 1989.
- [75] R Jain, R Kasturi, and BG Schunck. Machine vision, volume 5. McGraw-Hill New York, 1995.
- [76] Yu-Gang Jiang, Chong-Wah Ngo, and Jun Yang. Towards optimal bag-of-features for object categorization and semantic video retrieval. In *Proceedings* of the 6th ACM international conference on Image and video retrieval, pages 494–501. ACM, 2007.
- [77] AR Jimenez, R Ceres, JL Pons, et al. A survey of computer vision methods for locating fruit on trees. *Transactions of the ASAE-American Society of Agricultural Engineers*, 43(6):1911–1920, 2000.
- [78] J Jutila, K Kannas, and A Visala. Tree measurement in forest by 2d laser scanning. In *International Symposium on Computational Intelligence in Robotics* and Automation, CIRA 2007, pages 491–496, Jacksonville, Florida, U.SA, 2007. IEEE.
- [79] A. Kaehler and G. Bradski. Learning OpenCV 3: Computer Vision in C++ with the OpenCV Library. O'Reilly UK Limited, 2016.

- [80] J Kan, W Li, and R Sun. Automatic measurement of trunk and branch diameter of standing trees based on computer vision. In 3rd IEEE Conference on Industrial Electronics and Applications, ICIEA 2008, pages 995–998, Singapore, Singapore, 2008. IEEE.
- [81] M Kanevski and S Canu. Spatial data mapping with support vector regression. *IDIAP Research Report*, *RR-00-09*, 2000.
- [82] Y Karimi, SO Prasher, A Madani, and S Kim. Application of support vector machine technology for the estimation of crop biophysical parameters using aerial hyperspectral observations. *Canadian Biosystems Engineering*, 50(7):13–20, 2008.
- [83] Markus Keller. The science of grapevines: anatomy and physiology. Academic Press, 2015.
- [84] Raj Khosla. Precision agriculture: challenges and opportunities in a flat world. In 19th World Congress of Soil Science, Soil Solutions for a Changing World, Brisbane, pages 1–6, 2010.
- [85] Anna Kicherer, Katja Herzog, Michael Pflanz, Markus Wieland, Philipp Rüger, Steffen Kecke, Heiner Kuhlmann, and Reinhard Töpfer. An automated field phenotyping pipeline for application in grapevine research. *Sensors*, 15(3):4823–4836, 2015.
- [86] NR Kitchen. Emerging technologies for real-time and integrated agriculture decisions. Computers and Electronics in Agriculture, 61(1):1–3, 2008.
- [87] Maria Klodt, Katja Herzog, Reinhard Töpfer, and Daniel Cremers. Field phenotyping of grapevine growth using dense stereo reconstruction. BMC bioinformatics, 16(1):1, 2015.
- [88] Min Hyuc Ko, Beom-Sahng Ryuh, Kyoung Chul Kim, Abhijit Suprem, and Nitaigour P Mahalik. Autonomous greenhouse mobile robot driving strategies from system integration perspective: Review and application. 2015.
- [89] WS Lee and DC Slaughter. Recognition of partially occluded plant leaves using a modified watershed algorithm. *Transactions of the ASAE*, 47(4):1269, 2004.
- [90] Pierre Legendre and Loic FJ Legendre. *Numerical ecology*, volume 24. Elsevier, 2012.
- [91] Jin Li and Andrew D Heap. Spatial interpolation methods applied in the environmental sciences: A review. *Environmental Modelling & Software*, 53:173–189, 2014.
- [92] Jin Li, Andrew D Heap, Anna Potter, and James J Daniell. Application of machine learning methods to spatial interpolation of environmental variables. Environmental Modelling & Software, 26(12):1647–1659, 2011.

- [93] Suk Hwan Lim, Jonathan Yen, and Peng Wu. Detection of out-of-focus digital photographs. Technical report, Hewlett-Packard Laboratories Technical Report, 2005.
- [94] Raphael Linker, Oded Cohen, and Amos Naor. Determination of the number of green apples in rgb images recorded in orchards. *Computers and Electronics in Agriculture*, 81:45–57, 2012.
- [95] Y-C Liu, H Gao, and X Zhang. Development and Application of Mobile Traceability Data Construction for Agriculture. In 8th Asian Conference for Information Technology in Agriculture and World Conference on Computer in Agriculture, Taipei City, Taiwan, 2012.
- [96] DH Lorenz, KW Eichhorn, H Bleiholder, R Klose, U Meier, and E Weber. Growth stages of the grapevine: phenological growth stages of the grapevine (vitis vinifera l. ssp. vinifera)—codes and descriptions according to the extended bbch scale†. Australian Journal of Grape and Wine Research, 1(2):100–103, 1995.
- [97] Gaëtan Louarn, Jérémie Lecoeur, and Eric Lebon. A three-dimensional statistical reconstruction model of grapevine (vitis vinifera) simulating canopy structure variability within and between cultivar/training system pairs. *Annals of Botany*, 101(8):1167–1184, 2008.
- [98] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.
- [99] L Lucchese and SK Mitra. Color image segmentation: A state-of-the-art survey. Proceedings of the Indian National Science Academy (INSA-A). Delhi, Indian: Natl Sci Acad, 67(2):207–221, 2001.
- [100] JW MacAdam. Structure and Function of Plants. John Wiley & Sons, July 2013.
- [101] Roger Martin-Clouaire, Jean-Pierre Rellier, Nakié Paré, Marc Voltz, and Anne Biarnès. Modelling management practices in viticulture while considering resource limitations: The dhiving model. *PloS one*, 11(3):e0151952, 2016.
- [102] Alessandro Matese and SF Di Gennaro. Technology in precision viticulture: A state of the art review. *Int. J. Wine Res*, 7:69–81, 2015.
- [103] G Matheron. Principles of geostatistics. *Economic geology*, 58(8):1246–1266, 1963.
- [104] A McBratney, B Whelan, T Ancev, and J Bouma. Future directions of precision agriculture. *Precision Agriculture*, 6(1):7–23, 2005.
- [105] CL McCarthy, NH Hancock, and SR Raine. Applied machine vision of plants: a review with implications for field deployment in automated farming operations. *Intelligent Service Robotics*, 3(4):209–217, 2010.

- [106] GJ McLachlan and T Krishnan. *The EM algorithm and extensions*, volume 274. Wiley New York, 1997.
- [107] Liquan Mei and Shujuan Zhang. A new weighted support vector machine for regression and its parameters optimization. In *International Conference on Intelligent Computing*, pages 597–604. Springer, 2008.
- [108] RJ Melchiori, S Albarenque, and AC Kemerer. Uso, adopción y limitaciones de la agricultura de precisión en Argentina. Curso Internacional de Agricultura de Precisión. 12. 17 al 19 de julio de 2013. Manfredi, Córdoba. AR., 2013.
- [109] Hossein Mousazadeh. A technical review on navigation systems of agricultural autonomous off-road vehicles. *Journal of Terramechanics*, 50(3):211–232, 2013.
- [110] Marius Muja and David G Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. VISAPP (1), 2:331–340, 2009.
- [111] A Méndez, J Vélez, D Villarroel, and F Scaramuzza. Evolución de la agricultura de precisión en Argentina en los últimos 15 años. Curso Internacional de Agricultura de Precisión. 13. 24 y 25 de septiembre 2014. Manfredi, Córdoba. AR., 2014.
- [112] KJ Niklas. *Plant Allometry: The Scaling of Form and Process*. University of Chicago Press, October 1994.
- [113] KJ Niklas. Size-dependent allometry of tree height, diameter and trunk-taper. Annals of botany, 75(3):217–227, 1995.
- [114] Stephen Nuske, Supreeth Achar, Terry Bates, Srinivasa Narasimhan, and Sanjiv Singh. Yield estimation in vineyards by visual grape detection. In *Intelligent Robots and Systems (IROS)*, 2011 IEEE/RSJ International Conference on, pages 2352–2358. IEEE, 2011.
- [115] Stephen O'Hara and Bruce A Draper. Introduction to the bag of features paradigm for image classification and retrieval. arXiv preprint arXiv:1101.3354, 2011.
- [116] K Omasa, F Hosoi, TM Uenishi, Y Shimizu, and Y Akiyama. Three-dimensional modeling of an urban park and trees by combined airborne and portable onground scanning LIDAR remote sensing. *Environmental Modeling & Assessment*, 13(4):473–481, 2008.
- [117] HN Patel, RK Jain, and MV Joshi. Automatic segmentation and yield measurement of fruit using shape analysis. *International Journal of Computer Applications*, 45(7):19–24, 2012.
- [118] Diego Sebastián Pérez, Facundo Bromberg, and Francisco Gonzalez Antivilo. Computer vision approach for low cost, high precision measurement of grapevine trunk diameter in outdoor conditions. arXiv preprint arXiv:1406.4845, 2015.

- [119] Diego Sebastián Pérez, Facundo Bromberg, and Carlos Ariel Diaz. Image classification for detection of winter grapevine buds in natural conditions using scale-invariant features transform, bag of features and support vector machines. Computers and electronics in Agriculture, In press.
- [120] DS Pérez and Facundo Bromberg. Segmentación de imágenes en viñedos para la medición autónoma de variables vitícolas. In XVIII Congreso Argentino de Ciencias de la Computación, 2012.
- [121] Claudia Perlich. Learning curves in machine learning. In *Encyclopedia of Machine Learning*, pages 577–580. Springer, 2011.
- [122] C Popa. Adoption of artificial intelligence in agriculture. Bulletin of the University of Agricultural Sciences & Veterinary Medicine Cluj-Napoca. Agriculture, 68(1), 2011.
- [123] David Martin Ward Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *International Journal of Machine Learning Technology*, 2(1):37–63, 2011.
- [124] Xingye Qiao and Lingsong Zhang. Distance-weighted support vector machine. Statistics and Its Interface, 8(3):331–345, 2015.
- [125] Andrew Rabinovich, Andrea Vedaldi, Carolina Galleguillos, Eric Wiewiora, and Serge Belongie. Objects in context. In 2007 IEEE 11th International Conference on Computer Vision, pages 1–8. IEEE, 2007.
- [126] M Reichardt and C Jürgens. Adoption and future perspective of precision farming in germany: results of several surveys among different agricultural target groups. *Precision Agriculture*, 10(1):73–94, 2009.
- [127] Manuel JCS Reis, Raul Morais, Emanuel Peres, Carlos Pereira, Olga Contente, S Soares, António Valente, José Baptista, Paulo Jorge SG Ferreira, and J Bulas Cruz. Automatic detection of bunches of grapes in natural environment from color images. *Journal of Applied Logic*, 10(4):285–290, 2012.
- [128] Sam Roweis. Em algorithms for pca and spca. Advances in neural information processing systems, pages 626–632, 1998.
- [129] Pravakar Roy, Nikolaos Stefas, Cheng Peng, Haluk Bayram, Pratap Tokekar, and Volkan Isler. Robotic surveying of apple orchards. Technical report, Department of Computer Science and Engineering, University of Minnesota, 2015.
- [130] G Ruß and A Brenning. Data mining in precision agriculture: management of spatial information. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 350–359. Springer, 2010.

- [131] Sindhuja Sankaran, Lav R Khot, Carlos Zúñiga Espinoza, Sanaz Jarolmasjed, Vidyasagar R Sathuvalli, George J Vandemark, Phillip N Miklas, Arron H Carter, Michael O Pumphrey, N Richard Knowles, et al. Low-altitude, high-resolution aerial imaging systems for row and field crop phenotyping: A review. European Journal of Agronomy, 70:112–123, 2015.
- [132] Ashutosh Saxena, Sung H Chung, and Andrew Y Ng. Learning depth from single monocular images. In *NIPS*, volume 18, pages 1–8, 2005.
- [133] B Schölkopf, AJ Smola, R.C Williamson, and P.L Bartlett. New support vector algorithms. *Neural computation*, 12(5):1207–1245, 2000.
- [134] Linda Shapiro and George C Stockman. Computer vision. *ed: Prentice Hall*, 2001.
- [135] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 1968 23rd ACM national conference*, pages 517–524. ACM, 1968.
- [136] Risto Sievänen, Christophe Godin, Theodore M DeJong, and Eero Nikinmaa. Functional–structural plant models: a growing paradigm for plant studies. *Annals of botany*, 114(4):599–603, 2014.
- [137] Kristoffer Sjöö, Alper Aydemir, and Patric Jensfelt. Topological spatial relations for active visual search. *Robotics and Autonomous Systems*, 60(9):1093–1107, 2012.
- [138] DC Slaughter, DK Giles, and D Downey. Autonomous robotic weed control systems: A review. *Computers and electronics in agriculture*, 61(1):63–78, 2008.
- [139] Richard E Smart. Principles of grapevine canopy microclimate manipulation with implications for yield and quality. a review. *American Journal of Enology and Viticulture*, 36(3):230–239, 1985.
- [140] Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. Statistics and computing, 14(3):199–222, 2004.
- [141] M Sokolova, N Japkowicz, and S Szpakowicz. Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. AI 2006: Advances in Artificial Intelligence, pages 1015–1021, 2006.
- [142] A Srinivasan. Handbook of Precision Agriculture: Principles and Applications. Crop science. Taylor & Francis, 2006.
- [143] JV Stafford. Implementing precision agriculture in the 21st century. *Journal of Agricultural Engineering Research*, 76(3):267–275, 2000.
- [144] R Szeliski. Computer vision: algorithms and applications. Springer Science & Business Media, 2010.

- [145] Xiaodong Tang, Manhua Liu, Hui Zhao, and Wei Tao. Leaf extraction from complicated background. In *Image and Signal Processing*, 2009. CISP'09. 2nd International Congress on, pages 1–5. IEEE, 2009.
- [146] J Tardaguila, MP Diago, J Blasco, B Millán, S Cubero, OL García-Navarrete, and N Aleixos. Automatic estimation of the size and weight of grapevine berries by image analysis. In *International Conference of Agricultural Engineering*. Valencia Spain, pages 8–12, 2012.
- [147] J Tardaguila, MP Diago, B Millan, J Blasco, S Cubero, and N Aleixos. Applications of computer vision techniques in viticulture to assess canopy features, cluster morphology and berry size. In *I International Workshop on Vineyard Mechanization and Grape and Wine Quality 978*, pages 77–84, 2012.
- [148] J Taylor and B Whelan. A general introduction to precision agriculture. Grains Research and Development Corporation, www. usyd. edu. au/su/agric/acpa, 2005.
- [149] ND Tillett, T Hague, and JA Marchant. A robotic system for plant-scale husbandry. *Journal of Agricultural Engineering Research*, 69(2):169–178, 1998.
- [150] Y Uno. Application of machine learning methods and airborne hyperspectral remote sensing for crop yield estimation. PhD thesis, McGill University, 2003.
- [151] Kimon P Valavanis, P Oh, and Les A Piegl. *Unmanned aircraft systems*. Springer, 2013.
- [152] Vladimir Vapnik. The nature of statistical learning theory. Springer Science & Business Media, 2013.
- [153] Vladimir Naumovich Vapnik and Vlamimir Vapnik. Statistical learning theory, volume 1. Wiley New York, 1998.
- [154] A Vibhute and SK Bodhe. Applications of image processing in agriculture: a survey. *International Journal of Computer Applications*, 52(2), 2012.
- [155] Rafael Grompone von Gioi, Jérémie Jakubowicz, Jean-Michel Morel, and Gregory Randall. Lsd: a line segment detector. *Image Processing On Line*, 2:35–55, 2012.
- [156] Achim Walter, Frank Liebisch, and Andreas Hund. Plant phenotyping: from bean weighing to image analysis. *Plant methods*, 11(1):14, 2015.
- [157] Jin-Feng Wang, A Stein, Bin-Bo Gao, and Yong Ge. A review of spatial sampling. *Spatial Statistics*, 2:1–14, 2012.
- [158] Le Wang, Jianru Xue, Nanning Zheng, and Gang Hua. Automatic salient object extraction with contextual cue. In 2011 International Conference on Computer Vision, pages 105–112. IEEE, 2011.

- [159] Qi Wang, Stephen Nuske, Marcel Bergerman, and Sanjiv Singh. Automated crop yield estimation for apple orchards. In *Experimental Robotics*, pages 745–758. Springer, 2013.
- [160] SC Wang. Artificial neural network. In *Interdisciplinary Computing in Java Programming*, pages 81–100. Springer, 2003.
- [161] Xiao-Feng Wang, De-Shuang Huang, Ji-Xiang Du, Huan Xu, and Laurent Heutte. Classification of plant leaf images with complicated background. *Applied mathematics and computation*, 205(2):916–926, 2008.
- [162] Xiaoyu Wang, Tony X Han, and Shuicheng Yan. An hog-lbp human detector with partial occlusion handling. In *Computer Vision*, 2009 IEEE 12th International Conference on, pages 32–39. IEEE, 2009.
- [163] Richard Webster and Margaret A Oliver. Geostatistics for environmental scientists (statistics in practice). 2001.
- [164] JL Whalley and S Shanmuganathan. Applications of image processing in viticulture: A review. In 20th MSSANZ-International Congress on Modelling and Simulation, pages 531–538. The Modelling and Simulation Society of Australia and New Zealand Inc.(MODSIM), 2013.
- [165] Lior Wolf and Stanley Bileschi. A critical view of context. *International Journal of Computer Vision*, 69(2):251–261, 2006.
- [166] Yalin Xiong and Steven A Shafer. Depth from focusing and defocusing. In Computer Vision and Pattern Recognition, 1993. Proceedings CVPR'93., 1993 IEEE Computer Society Conference on, pages 68–73. IEEE, 1993.
- [167] Sheng Xu, Yi Xun, Tingmeng Jia, and Qinghua Yang. Detection method for the buds on winter vines based on computer vision. In *Computational Intelligence* and Design (ISCID), 2014 Seventh International Symposium on, volume 2, pages 44–48. IEEE, 2014.
- [168] Jun Yang, Yu-Gang Jiang, Alexander G Hauptmann, and Chong-Wah Ngo. Evaluating bag-of-visual-words representations in scene classification. In *Proceedings* of the international workshop on Workshop on multimedia information retrieval, pages 197–206. ACM, 2007.
- [169] MH Yang and N Ahuja. Gaussian mixture model for human skin color and its application in image and video databases. In *Proc. SPIE: Storage and Retrieval for Image and Video Databases VII*, volume 3656, pages 458–466. Citeseer, 1999.
- [170] Xulei Yang, Qing Song, and Yue Wang. A weighted support vector machine for data classification. *International Journal of Pattern Recognition and Artificial Intelligence*, 21(05):961–976, 2007.
- [171] Yiming Ye and John K Tsotsos. Sensor planning in 3d object search: its formulation and complexity. In *The 4th International Symposium on Artificial Intelligence and Mathematics, Florida, USA*, 1996.

- [172] F Zhang and LS Fu. Application of computer vision technology in agricultural field. In *Applied Mechanics and Materials*, volume 462, pages 72–76. Trans Tech Publ, 2014.
- [173] N Zhang, M Wang, and N Wang. Precision agriculture—a worldwide overview. Computers and electronics in agriculture, 36(2):113–132, 2002.
- [174] YJ Zhang. A summary of recent progresses for segmentation evaluation. Advances in image and video segmentation, page 423, 2006.