



UNIVERSIDAD AUTÓNOMA DE YUCATÁN

FACULTAD DE MATEMÁTICAS

DETECCIÓN DEL ÁRBOL DE DZIDZILCHÉ USANDO  
IMÁGENES AÉREAS MULTIESPECTRALES Y REDES  
NEURONALES.

T E S I S

EN OPCIÓN AL GRADO DE:

**Maestro en Ciencias de la Computación**

PRESENTA:

**I.C. Juan Jose Negron Granados**

ASESORES DE TESIS:

Dr. Ricardo Legarda Saénz

Dr. Víctor Uc Cetina

Mérida, Yucatán, México

Octubre, 2019



A mis padres,  
a mis hermanos,  
a Dayry, mi novia,  
a mis amigos.

# Agradecimientos

A mi familia, especialmente a mis padres, Beatriz Granados y Juan Negron, por su apoyo y amor incondicional durante todo este tiempo.

A Dayry Chan, por ser la luz que me ilumina en los momentos más oscuros; por tu amor y tu asombrosa habilidad de hacerme sentir orgulloso de lo que hago. A ti, mi amor, infinitas gracias.

A mis profesores, por enriquecer mis conocimientos académicos y ser ejemplo de los valores de la universidad; principalmente al Dr. Ricardo Legarda y al Dr. Víctor Uc, quienes asesoraron la presente investigación. Gracias por su rigor académico, tolerancia y confianza. Un agradecimiento especial al Dr. Carlos Brito, por su amistad brindada y apoyarme siempre durante mi vida universitaria.

Al Dr. Jorge Argáez, coordinador de la Unidad de Posgrado e Investigación, por el apoyo para la divulgación del conocimiento científico producido durante la elaboración de esta investigación.

Al Consejo Nacional de Ciencia y Tecnología, por el financiamiento a través de su programa de becas.

A todos, mil gracias. Sin ustedes esto no hubiera sido posible.

# Resumen

La apicultura es una de las actividades agropecuarias más importantes para México, al año se producen 57,000 toneladas de miel por año, siendo el sexto productor a nivel mundial y el tercer mayor exportador del mundo [15]. En el estado de Yucatán, se estima que se exportan anualmente 10,000 toneladas de miel, pues dicho estado cuenta con alrededor de 6,000 apicultores sumando cerca de 240,000 colonias distribuidas en todo Yucatán [25]. Sin embargo, esta actividad pecuaria enfrenta diversas problemáticas que afectan la producción y rentabilidad de los productos apícolas, los problemas que destacan son: la africanización de las abejas, la presencia del ácaro varroa, pérdida de competitividad, descapitalización y una deficiente legislación de la actividad [39]; de tal manera que la situación presenta contaminación por el uso de plaguicidas, incrementos en costos de producción, reducción de áreas para apicultura y deficiente distribución de los apiarios en la región [25].

La flor del árbol de dzidzilché es el principal recurso néctar-polinívoro para la producción de miel en el área de estudio para esta investigación [6], por esta razón, se hace la propuesta de detectar el árbol de dzidzilché mediante imágenes aéreas multiespectrales y redes neuronales convolucionales. Las imágenes multiespectrales se tomaron con el sensor espectral *Parrot Sequoia* desde un vehículo aéreo no tripulado, cuyo vuelo se realizó en las instalaciones del Campus de Ciencias Biológicas y Agropecuarias de la Universidad Autónoma de Yucatán, a las afueras de la ciudad de Mérida. La caracterización es un método que permite diseñar estrategias para la conservación del ecosistema [35] por lo tanto, la detección del árbol de dzidzilché como técnica de caracterización facilitará el desarrollo de métodos para conservar y explotar el potencial de producción de miel que aún se encuentra vigente en el estado de Yucatán, permitiendo así, el repoblamiento de apiarios y una adecuada distribución de las colonias de abejas, entre otros beneficios [25].

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Objetivos . . . . .	2
1.1.1. Objetivo general . . . . .	2
1.1.2. Objetivos particulares . . . . .	2
1.2. Planteamiento del problema . . . . .	3
1.3. Revisión de la literatura . . . . .	3
1.4. Publicaciones . . . . .	9
1.5. Estructura de la tesis . . . . .	10
<b>2. Marco teórico</b>	<b>11</b>
2.1. Árbol de dzidzilché . . . . .	11
2.2. Área de estudio . . . . .	11
2.3. Espectro electromagnético . . . . .	12
2.4. Sensor multiespectral . . . . .	14
2.4.1. Modelo de cámara y calibración . . . . .	15
2.4.2. Distorsión ojo de pescado . . . . .	17
2.4.3. Alineación de las imágenes . . . . .	18
2.4.4. Homografía . . . . .	19
2.5. Aprendizaje Máquina . . . . .	20
2.6. Red neuronal . . . . .	21
2.6.1. Red neuronal artificial . . . . .	21
2.6.2. Red neuronal convolucional . . . . .	22
2.6.3. Red convolucional <i>LeNet</i> . . . . .	25
2.7. Validación cruzada . . . . .	26
2.8. Evaluación del modelo . . . . .	27
2.8.1. Matriz de confusión . . . . .	27
2.8.2. Curva ROC y AUC . . . . .	28

<b>3. Metodología</b>	<b>31</b>
3.1. Descripción general . . . . .	31
3.2. Etapa de captura . . . . .	33
3.3. Etapa de corrección y alineación . . . . .	33
3.4. Etapa de etiquetado . . . . .	34
3.5. Etapa de entrenamiento . . . . .	36
3.6. Etapa de evaluación . . . . .	39
<b>4. Resultados</b>	<b>40</b>
4.1. Calibración y alineación . . . . .	40
4.2. Red neuronal . . . . .	42
4.3. Validación cruzada . . . . .	43
4.4. Validación cruzada de $k$ iteraciones . . . . .	45
<b>5. Conclusiones</b>	<b>52</b>
5.1. Trabajos futuros . . . . .	53

# Índice de tablas

1.1. Especies de plantas más importantes para la producción de miel en Yucatán. . .	2
1.2. Estadios simulados en los experimentos [53]. . . . .	5
1.3. Comparación de los resultados con otros modelos basados en SVM. . . . .	8
2.1. Ejemplo de resultados de diferentes clasificadores y sus respectivas etiquetas. .	29
3.1. Relación de parámetros que requieren ser entrenados en cada capa de la arquitectura. . . . .	37
3.2. Distribución del conjunto de entrenamiento para diferentes valores de $k$ . . . .	38
4.1. Descripción de las etiquetas en los metadatos. . . . .	41
4.2. Vectores de traslación de acuerdo a la documentación [49]. . . . .	41
4.3. Homografías promedio para la alineación entre bandas. . . . .	42
4.4. Resultados de los modelos con menor costo. . . . .	46
4.5. Tabla de contingencia promedio con los valores absolutos. . . . .	46
4.6. Desviación estándar de los valores en la tabla de contingencia. . . . .	46
4.7. Valores estadísticos inferidos a partir de la tabla de contingencia. . . . .	47

# Índice de figuras

1.1. Arquitectura general de la red convolucional usada en [40]. . . . .	8
2.1. Flores del árbol de dzidzilché. . . . .	12
2.2. Facultad de Veterinaria de la Universidad Autónoma de Yucatán. . . . .	13
2.3. Ubicación del CCBA se puede notar que alrededor se conserva la flora nativa del estado de Yucatán. . . . .	13
2.4. El espectro electromagnético [58]. . . . .	14
2.5. Cámara Sequoia . . . . .	15
2.6. Representación de los parámetros externos de una cámara. . . . .	16
2.7. Geometría de la perspectiva en una cámara sin distorsión. . . . .	17
2.8. Representación de la distorsión ojo pescado de un objeto cuadrado. . . . .	18
2.9. Ejemplo de la técnica de homografía (fuente:[47]). . . . .	19
2.10. Base de datos etiquetada para ser usados en métodos de aprendizaje supervisado.	20
2.11. Estructura de una neurona artificial. . . . .	21
2.12. Ejemplo de perceptrón multicapa completamente conectado con un vector de entrada de tamaño $k$ , una capa oculta de $n$ neuronas y $m$ salidas. . . . .	22
2.13. Estructura general de una Red Neuronal Convolucional [52]. . . . .	23
2.14. El operador de convolución. . . . .	23
2.15. Función tangente hiperbólica. . . . .	24
2.16. Ejemplo de submuestreo por promedio ( <i>Average pooling</i> ). . . . .	25
2.17. Arquitectura de la red neuronal convolucional <i>LeNet-1</i> . . . . .	25
2.18. Diagrama de flujo del método de validación cruzada. . . . .	26
2.19. Diagrama de flujo del método validación cruzada de $k$ iteraciones. . . . .	27
2.20. Composición de la matriz de confusión. . . . .	28
2.21. Ejemplo de curvas ROC para distintos clasificadores, con su respectiva área bajo la curva (AUC). . . . .	30
3.1. Gráfico general de la metodología propuesta. . . . .	32

3.2.	Zona del CCBA donde se tomaron las imágenes multiespectrales. . . . .	33
3.3.	Ejemplo para el proceso de etiquetado. . . . .	35
3.4.	<i>Data augmentation</i> . Cada elemento del conjunto de datos etiquetados se divide en 16 partes iguales. . . . .	35
3.5.	División del conjunto de datos. . . . .	36
3.6.	Red neuronal convolucional propuesta. . . . .	38
4.1.	Resultado de realizar la corrección de ojo de pescado descrito en la subsección 2.4.2. El resultado final del proceso es la imagen de la derecha, sin embargo la figura central se incluyó para mostrar cómo se deforma el margen de la imagen. . . . .	41
4.2.	Posición de los puntos internos del tablero en cada imagen espectral. . . . .	42
4.3.	Posición de los puntos internos del tablero después de realizar la alineación usando las homografías promedio correspondientes. . . . .	43
4.4.	Función de pérdida en la validación con diferentes tasas de aprendizaje. . . . .	44
4.5.	Función de pérdida en la validación con diferentes tasas de aprendizaje suavizada con un filtro Gaussiano con $\sigma = 2$ . . . . .	44
4.6.	Función de pérdida promedio por cada validación cruzada de $k$ iteraciones. . . . .	45
4.7.	Curvas ROC generadas con sus respectivas áreas bajo la curva. . . . .	47
4.8.	Resultados del clasificador usando una imagen espectral. La Subfigura a) tiene combinación color infrarrojo. En la Subfigura b) el área blanca indica presencia de dzidzilché. . . . .	48
4.9.	Resultados del clasificador usando una imagen espectral de dzidzilché en toda el área. La Subfigura a) tiene combinación color infrarrojo. El área blanca en la Subfigura b) corresponde al área con presencia de dzidzilché. . . . .	49
4.10.	Resultados del clasificador usando una imagen espectral de plantas diferentes a dzidzilché. La Subfigura a) tiene combinación color infrarrojo. El área blanca en la Subfigura b) corresponde al área con presencia de dzidzilché. . . . .	49
4.11.	Resultados del clasificador usando una imagen espectral de plantas diferentes a dzidzilché. La Subfigura a) tiene combinación color infrarrojo. El área blanca en la Subfigura b) corresponde al área con presencia de dzidzilché. . . . .	50
4.12.	Resultados del clasificador usando una imagen espectral con presencia de dzidzilché, suelo y sombra. La Subfigura a) tiene combinación color infrarrojo. La Subfigura b) indica la salida del clasificador. La Subfigura c) presenta el resultado de realizar apertura y cerradura a la salida b). . . . .	50

4.13. Resultados del clasificador usando una imagen espectral de dzidzilché en toda el área. La Subfigura a) tiene combinación color infrarrojo. La Subfigura b) indica la salida del clasificador. La Subfigura c) presenta el resultado de realizar apertura y cerradura a la salida b). . . . .	50
4.14. Resultados del clasificador usando una imagen espectral de plantas diferentes a dzidzilché. La Subfigura a) tiene combinación color infrarrojo. La Subfigura b) indica la salida del clasificador. La Subfigura c) presenta el resultado de realizar apertura y cerradura a la salida b). . . . .	51
4.15. Resultados del clasificador usando una imagen espectral de plantas diferentes a dzidzilché y cemento. La Subfigura a) tiene combinación color infrarrojo. La Subfigura b) indica la salida del clasificador. La Subfigura c) presenta el resultado de realizar apertura y cerradura a la salida b). . . . .	51

# Capítulo 1

## Introducción

La península de Yucatán es, por tradición, una importante región productora de miel a nivel mundial, ya que 95 % de su producción se destina al mercado internacional [27], no obstante, la apicultura ha sufrido un grave deterioro en la región pues ha disminuido la cantidad de recursos néctar-polinívoros y por tanto, la alimentación de las abejas, con lo consiguiente una baja en la producción de productos mielíferos. Por lo que existe la necesidad de mejorar los sistemas de comercialización y de diversificación de la actividad, y actualizar las técnicas de producción y administración del proceso productivo por parte de los apicultores para obtener la calidad de miel requerida por el mercado [27].

Uno de los principales métodos para comprender los procesos ecológicos y conservar la fauna es mediante la caracterización del ambiente [35]; por ejemplo, el empleo de la caracterización ha permitido señalar los procesos que afectan los humedales del norte de Argentina [54] lo que permitirá el diseño de planes de manejo y conservación de sus comunidades biológicas; en Chile, se realizó la caracterización de la flora y vegetación de los humedales de la costa de Toltén, los cuales han sido poco estudiados, y estableció los tipos de humedales presentes en las cuencas hidrográficas de los ríos en la comuna de Toltén y su espectro biológico presente en ellas [28]. De igual forma, es posible aplicar la caracterización a favor de las actividades pecuarias, por ejemplo, para la colecta y conservación de especies vegetativas útiles para la ganadería [38]. Asimismo, la caracterización puede tener un impacto positivo en la apicultura; en Tabasco, México, se comprobó que dicho estado puede competir en el comercio de mieles monoflorales, que son altamente apreciadas en Europa, a pesar que Tabasco no destaca en volumen de producción de miel [16]. Por lo tanto una manera de mejorar la producción y calidad de la miel es caracterizando la vegetación que se encuentra alrededor de los apiarios. La dieta de las abejas es muy específica (Tabla 1.1), por lo que se espera que 5 km a la redonda de un apiario [48] se encuentre una abundante vegetación adecuada para evitar los problemas de producción de miel.

Tabla 1.1: Especies de plantas más importantes para la producción de miel en Yucatán.

Nombre común	Nombre científico
Ts'its'ilche' o dzidzilché	Gymnopodium floribundum
Taj	Viguiera dentata
Ja'abin	Piscidia piscipula
Box katsim	Acacia gaumeri
Sac katsim	Mimosa bahamensis
Tsalam	Lysiloma latisiliquum
X-tabentún	Turbina corymbosa
Kitimché	Caesalpinia gaumeri
Enredaderas	Varias especies de la familia Convolvulaceae

La clasificación de plantas en salidas de campo es un proceso complicado y nada eficiente, por lo que, es común utilizar imágenes satelitales para la evaluación y clasificación de la vegetación por su alto contenido de información. Sin embargo, la distancia de los sensores evita tener imágenes de buena resolución, pues en una imagen multiespectral satelital un pixel puede representar un área de  $30m^2$ . Obtener imágenes aéreas con un dron mejora resolución espacial.

Este trabajo se centra en la detección del árbol de dzidzilché, el cual tiene una gran importancia apícola, pues una de las mieles con más demanda en el mundo es la miel producida con el polen de la flor del dzidzilché [6].

## 1.1. Objetivos

### 1.1.1. Objetivo general

Detectar el árbol de dzidzilché utilizando imágenes multiespectrales y redes neuronales convolucionales.

### 1.1.2. Objetivos particulares

- Tomar fotos aéreas, en regiones donde se encuentren árboles de dzidzilché utilizando el sensor multiespectral *Parrot Sequoia*.
- Corregir la distorsión de cada cámara en el sensor multiespectral.
- Diseñar y entrenar una red neuronal convolucional para detectar el árbol de dzidzilché.

## 1.2. Planteamiento del problema

La apicultura es una actividad pecuaria muy importante para México; Yucatán es uno de las entidades federativas que más aporta en producción y exportación de miel. Sin embargo, la apicultura se ve amenazada por una deficiente organización de los apicultores, mala ubicación de los apiarios y reducción de la vegetación, lo que afecta a la flora polinífera, principal fuente de alimento de las diferentes especies de abejas. Este trabajo hace la propuesta de utilizar información espectral para identificar el árbol de dzidzilché, principal planta productora de polen; la metodología de la propuesta consiste en utilizar visión remota, procesamiento de imágenes y redes neuronales artificiales en imágenes multiespectrales tomadas desde un vehículo aéreo no tripulado con el sensor *Parrot Sequoia*.

En este trabajo se sientan las bases para realizar un *software* que ayude a los apicultores en la cuantificación de la flora polinífera que existan alrededor de sus apiarios y les sirva como herramienta para prevenirse ante hambrunas, optimización en la ubicación estratégica de colonias de abejas y mejorar una de las actividades pecuarias más importantes de México.

## 1.3. Revisión de la literatura

Entre los problemas del sistema actual de producción de miel se observa una reducción de las áreas para la apicultura, una deficiente distribución de apiarios y un bajo uso de tecnología, lo que ha conllevado al despoblamiento de apiarios y a la pérdida de recursos poliníferos; sin embargo, es posible mejorar la situación que enfrenta la apicultura, pues Yucatán aún cuenta con condiciones naturales adecuadas para su desarrollo [25]. De acuerdo con Leopold, para comprender y conservar los procesos ecológicos es necesario realizar la caracterización del ambiente [35], además, en [64] se menciona que es importante conocer la flora circundante a los apiarios para determinar el uso que las abejas le dan para la elaboración de los diferentes productos mielíferos que producen, de hecho el trabajo afirma que esta actividad es el primer requisito para asegurar el éxito de la apicultura. Por lo tanto, en este capítulo se hace una revisión de antecedentes en el trabajo de caracterización de coberturas terrestres con un enfoque vegetativo.

La forma tradicional para caracterizar la vegetación polinífera es mediante la observación de las abejas en las zonas cercanas a sus colmenas. En [5] se identifican las diferentes plantas poliníferas que se encuentran en la región centro-este del municipio de Pátzcuaro, Michoacán, cuyo método consistió en realizar visitas semanales a apiarios de referencia y recorrer el área

que abarca un radio de  $1km$  alrededor de las referencias. Posteriormente, se recolectaron especies vegetales poliníferas y se determinó el periodo de floración de cada especie y los problemas de uso de suelo y plagas que las afectan; lo cual logró identificar 93 especies de plantas poliníferas de 33 familias botánicas diferentes, además se determinó que la deforestación de este tipo de vegetación para el cultivo de aguacate y la presencia de plagas ha provocado un retraso o ausencia de la floración polinífera. De igual forma, la metodología usada en [44] consiste en recorrer el área de  $1km$  alrededor de una colmena, identificar las especies poliníferas y darles un seguimiento mensual, lo que ayudó a remarcar la importancia de los recursos vegetativos para la cosecha de la miel.

También es posible encontrar trabajos que utilicen herramientas de los sensores remotos y procesamiento de imágenes, pues ofrecen una alternativa cómoda y económica para la caracterización de coberturas terrestres; por ejemplo, en [45] se procesan las imágenes de los satélites LANSAT y MODIS para obtener información de los cultivos alrededor de apiarios, determinando que existe una relación entre el índice de vegetación de diferencia normalizada con la producción de miel de los apiarios analizados. En [59] se utiliza la información geográfica proporcionada por la cartografía digital de uso de suelo y vegetación del INEGI para identificar la vegetación presente en las áreas apícolas de su estudio; es importante mencionar que la cartografía digital mencionada incluye información del satélite Landsat ETM, la cual proporciona la ubicación y distribución de los tipos de vegetación y agricultura en la república mexicana [29]. El trabajo que se describe en [3] utiliza la información de imágenes temporales de LANSAT y Google Earth para hacer un análisis de la cobertura terrestre para investigar la relación del impacto de áreas cultivadas y la producción de miel. En el trabajo [41] se concluye como los Sistemas de Información Geográfica pueden ayudar a la toma de decisiones en la selección de la zona adecuada para la apicultura; sin embargo para lograr esto, la aplicación de los sensores remotos también debe enfocarse en la etapa de reproducción de la vegetación, es decir en su etapa de floración, como se realiza en el trabajo [53], donde se diseñó un modelo para simular las etapas de floración partiendo del análisis de las bandas espectrales rojo e infrarrojo cercano proporcionadas por un radiómetro con 5 bandas espectrales (azul, verde, rojo, infrarrojo cercano e infrarrojo medio, con espectros similares proporcionadas por el satélite Landsat), posteriormente se analizaron los espectros rojo e infrarrojo cercano para determinar la etapa de floración, los cuales están definidos por estadios numerados del 1-8 y corresponde a las características presentadas en la vegetación (Tabla 1.2); no obstante, los resultados presentan limitaciones entre la floración y los espectros analizados, pues las bandas roja e infrarroja tienen patrones similares y no es posible discriminar el término de la floración y el inicio del follaje, por lo que es necesario el desarrollo de nuevas técnicas para el análisis espectral.

Tabla 1.2: Estadios simulados en los experimentos [53].

Estadio	Descripción
1	Ramas y estratos bajos verdes, paisaje seco.
2	Aparecen las yemas florales, ramas parcialmente verdes en las puntas, estratos bajos verdes, paisaje semiseco.
3	Aparecen los botones florales, ramas verdes, estratos bajos verdes, paisaje parcialmente verde.
4	Los sépalos dejan ver los pétalos, ramas verdes, estratos bajos verdes, paisaje parcialmente verde con tonos tenues de prefloración.
5	Aparecen las primeras flores, ramas verdes, estratos bajos verdes, paisaje parcialmente verde con tonos moderados de floración.
6	Plena floración, ramas verdes, estratos bajos verdes, paisaje dominado por el color de la flor con tonos escasos de verde.
6.33	Caída parcial de flores, ramas verdes, estratos bajos verdes, paisaje codominado por el tono de la floración y tonos secos de la vegetación; tonos verdes dispersos y escasos.
6.66	Caída plena de flores, ramas verdes, estratos bajos verdes, paisaje semiseco, fondo con tonos de la floración.
7	Paisaje codominante tonos verdes y secos.
8	Paisaje verde.

Por otro lado, el uso de redes neuronales artificiales en la clasificación de datos de sensores remotos ha crecido en los últimos años [37], esto se puede ver reflejado en el trabajo [62], donde se usa la red neuronal *Hopfield*, un algoritmo de agrupamiento, utilizado en imágenes satelitales a color de la región Al Baha. La estructura de la red consiste de  $N \times M$  neuronas, diseñado para clasificar la imagen de  $N$  píxeles con  $P$  características a través de  $M$  clases, tal que la asignación de píxeles minimiza la siguiente función:

$$E = \frac{1}{2} \sum_{k=1}^N \sum_{l=1}^M R_{kl}^2 V_{kl}^2 + c(t) \sum_{k=1}^N \sum_{l=1}^M N_{kl} V_{kl} \quad (1.1)$$

donde  $R_{kl}$  es la distancia Mahalanobis entre el píxel  $k$  y el centro de la clase  $l$ , y  $V_{kl}$  es la salida de la neurona  $k$ .  $N_{kl}$  es un vector  $N \times M$  de ruido blanco, para evitar que la red quede atrapada en mínimos locales. El término  $c(t)$  es el parámetro que controla la magnitud del ruido, el cual es cero cuando la red alcanza la convergencia, la cual se logra resolviendo la siguiente ecuación:

$$\frac{\partial U_{kl}}{\partial t} = -\mu(t) \frac{\partial E}{\partial V_{kl}} \quad (1.2)$$

donde  $U_{kl}$  es la entrada de la  $k$ -ésima neurona y  $\mu(t)$  es una función escalar positiva de tiempo

definida por:

$$\mu(t) = t(T_s - t) \quad (1.3)$$

donde  $t$  es el número de iteración actual y  $T_s$  es el número previamente definido de iteraciones necesarias para alcanzar la convergencia de la red, empíricamente hallada en 120 iteraciones [61]. El algoritmo consiste en:

1. Inicializar la entrada de las neuronas con valores aleatorios.
2. Establecer la relación de cada píxel hacia una sola clase:

$$V_{km}(t+1) = 1, \text{ Si } U_{km} = \text{máx}[U_{km}(t), \forall l], \quad (1.4)$$

$$V_{km}(t+1) = 0, \text{ de otra forma, } 1 \leq k \leq N, 1 \leq l \leq M. \quad (1.5)$$

3. Computar los centros  $\bar{X}$  y la matriz de covarianza  $\Sigma_l$  para cada clase  $l$ :

$$\bar{X} = \frac{\sum_{k=1}^N X_k V_{kl}}{n_l}, \quad 1 \leq l \leq M. \quad (1.6)$$

$$\Sigma_l = \frac{[V_{kl}(X_k - \bar{X}_l)(X_k - \bar{X}_l)^T]}{n_l - 1}, \quad 1 \leq l \leq M. \quad (1.7)$$

donde  $n_l$  indica el número de píxeles en la clase  $l$ . La matriz de covarianza es normalizada dividiendo cada elemento por el determinante.

4. Se actualizan las entradas de cada neurona utilizando :

$$U_{kl}(t+1) = U_{kl}(t) + \frac{dU_{kl}}{dt}, \quad 1 \leq k \leq N, \quad 1 \leq l \leq M. \quad (1.8)$$

5. Si  $t \leq T_s$  repetir desde el paso 2, si no, terminar.

Los resultados hallados por [62] recomiendan utilizar 4 y 5 centros para obtener una buena segmentación de la imágenes utilizadas; además, es importante notar como la clasificación de la cobertura terrestre puede ser aplicada para la caracterización de la vegetación, por consiguiente, se mencionan trabajos enfocados en la clasificación de la cobertura terrestre utilizando redes neuronales.

En [11] se hace uso de las imágenes multiespectrales, proporcionadas por el satélite *Quick-Bird* y de redes neuronales artificiales para la detección de superficies con presencia de invernaderos. La metodología usada consiste en etiquetar las imágenes satelitales en 8 clases: invernaderos no activos, invernaderos activos, vegetación, agua superficial, asfalto, suelo natural, suelo sin vegetación y sombra. La entrada de la red neuronal consta de la imagen espectral de 4 canales. En este trabajo, se registró un error promedio entre 21.88 % y 22.17 % para las etapas de entrenamiento y pruebas, respectivamente. Sin embargo, los trabajos más recientes acerca de la clasificación general de la cobertura terrestre utiliza técnicas de aprendizaje profundo y se encuentran en las investigaciones de Chen y Lin, del Instituto de Tecnología Harbin, China. En el año 2013 Lin et al. hacen la propuesta de utilizar autoencoders para la clasificación de imágenes hiperespectrales [36], posteriormente, en 2014 Chen et al. reportaron sus resultados de aplicar los autoencoders [12] para el reconocimiento de diferentes regiones terrestres, logrando identificar con mucho éxito: suelo desnudo, prados, ladrillos, sombras, grava, agua, diferentes tipos de pantano, maleza y pinos. Un año después, Chen et al. realizaron el mismo experimento, cambiando el método autoencoder por una red de creencia profunda (DBN por sus siglas en inglés, Deep Belief Network) [13]. Los resultados fueron bastante buenos, la red pudo identificar las mismas regiones que detectó la técnica de *autoencoder*, sin embargo la arquitectura del autoencoder supera ligeramente a DBN, en promedio el *autoencoder* tuvo una efectividad del 96 % y DBN del 93 %.

También es posible encontrar trabajos que empleen técnicas de Redes Neuronales Convolucionales (CNN, por sus siglas en inglés), por ser métodos efectivos para el procesamiento de imágenes y la extracción de características. En [40] se utiliza CNN para la clasificación de datos hiperespectrales adquiridos por los sensores ROSIS y AVIRIS, en particular los conjuntos de datos llamados: *Indian Pines*, *Salinas*, *Pavia Centre* y *Pavia University*, estos datos se encuentran etiquetados; la arquitectura que proponen corresponde a la Figura 1.1, donde  $c$  indica la cantidad de los primeros  $n$  componentes espectrales utilizados, *R-PCA* (Análisis de componentes principales usando SVD aleatorizado, por sus siglas en inglés) dando la dimensión  $c_r$ , la primera etapa de la red consta de  $C_1$  capas convolucionales, definida por  $3 \times c_r$  con filtros de dimensión de  $3 \times 3$ , y la segunda capa está definida por  $C_2 = 3 \times C_1$  filtros de  $3 \times 3$ , donde  $C_2$ , representa el total de características extraídas por la red convolucional. La segunda parte de la arquitectura consiste en un perceptrón multicapa con una capa oculta de dimensión  $6 \times c_r$ , la entrada está dada por el vector  $C_2$  y la salida es el número de clases. Los conjuntos de datos fueron divididos en 3 secciones en una relación 80-10-10 % para entrenamiento, validación y pruebas; la Tabla 1.3 presenta los resultados obtenidos por la CNN y son comparados con otros métodos basados en Máquinas de Soporte Vectorial (mejor conocidos como SVM por sus siglas en inglés): SVM de

kernel lineal (linear-SVM), SVM de kernel radial (RBF-SVM) y sus variantes usando R-PCA.

Tabla 1.3: Comparación de los resultados con otros modelos basados en SVM.

Conjunto	Exactitud por método (%)				
	CNN	R-PCA RBF-SVM	R-PCA Linear - SVM	RBF-SVM	Linear- SVM
Pavia Centre	99.91	98.87	97.63	99.01	97.86
Pavia University	99.62	93.82	84.39	93.94	84.67
Salinas	99.53	93.73	90.52	93.97	90.68
Indian Pines	98.88	82.71	79.47	82.79	79.56

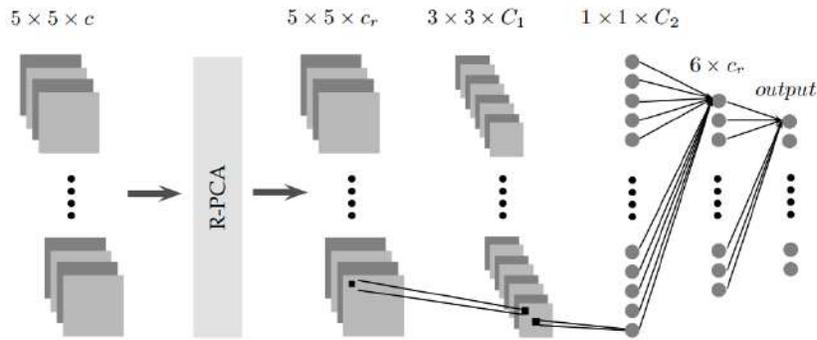


Figura 1.1: Arquitectura general de la red convolucional usada en [40].

Los trabajos anteriores describen la metodología para clasificar las coberturas utilizando métodos de aprendizaje profundo utilizando imágenes espectrales, las cuales resultaron ser altamente eficientes. No obstante, en la literatura también hay metodologías que utilizan imágenes multiespectrales; este tipo de imágenes constan de un set entre 4 y 20 imágenes, dependiendo del satélite o sensor que se utilice. Para el procesamiento de imágenes multiespectrales en vegetación se suelen usar 4 bandas del espectro de la luz: verde (530-570 nm), rojo (640-680 nm), borde rojo (730-740 nm) e infrarrojo cercano (770-810 nm). A continuación se mencionan algunos de los trabajos citados que utilizan las bandas espectrales mencionadas para la identificación de las diferentes coberturas; además cabe mencionar que se utiliza una red neuronal artificial como método de clasificación y sensores satelitales para la obtención de la información espectral. Entre las aplicaciones más importantes de este método se encuentra el análisis multitemporal [24], que consiste en la clasificación de las coberturas pero analizando las variaciones espectrales a través del tiempo para determinar los cambios en las características biofísicas de las regiones, determinar el avance de la restauración de zonas ecológicas [56] o el monitoreo de parques nacionales [65]. Del mismo modo, la técnica de combinar imágenes multiespectrales

con redes neuronales ha llevado al análisis de cultivos, por ejemplo el de caña de azúcar [63] o el análisis de las condiciones del suelo, como el nivel de salinidad en la tierra [43]. Debido a que varias coberturas terrestres pueden presentar respuestas espectrales similares, se han realizado variantes que complementan la información de las imágenes con información cartográfica [42] como modelos digitales de elevación (MDE), mapas temáticos de la región o datos de precipitación atmosférica [9], mejorando la efectividad aproximadamente en un 10 %.

En los trabajos presentados anteriormente se describen las ventajas de utilizar las imágenes multiespectrales y las redes neuronales para la clasificación de diferentes coberturas del suelo, sin embargo, una de las limitaciones de utilizar imágenes satelitales es el costo de la adquisición y procesamiento de las imágenes, por ejemplo, el sensor satelital *QuickBird* tiene dimensiones mayores a 27000 píxeles<sup>2</sup> con una resolución de 11 bits por cada una de las bandas que opera el sensor en el satélite. Otra limitante que es común observar es la relación del píxel con la dimensión que terrestre que representa; en el satélite *QuickBird* se tiene una resolución de 61cm por píxel, esto quiere decir que un píxel representa un área de 0.36m<sup>2</sup> [20]; esta resolución puede ser útil cuando se trata de clasificar superficies extensas, sin embargo para detectar plantas específicas requiere una resolución más precisa. Adicionalmente, en los trabajos presentados se identifica el proceso general para la clasificación de coberturas usando imágenes espectrales: adquisición de las imágenes, un preprocesamiento en la imagen, como el cálculo de los índices de vegetación o las correcciones y calibración de las imágenes satelitales, finalmente se hace el uso de técnicas de algoritmos de aprendizaje máquina para la clasificación.

## 1.4. Publicaciones

Los resultados preliminares que se fueron obteniendo durante la realización de este trabajo tuvieron las siguientes publicaciones:

- Juan Jose Negron-Granados, Ricardo Legarda-Sáenz y Victor Uc-Cetina. *Enfoque para la clasificación de vegetación polinifera usando imágenes multiespectrales y redes neuronales*. Research in Computing Science, 2018
- Juan Jose Negron-Granados. *Clasificación de hojas usando redes neuronales artificiales*. SOMI Congreso de Instrumentación, 2018. ISSN: 2395-8499.
- Juan Jose Negron-Granados. *Detección del árbol de dzidzilché usando descriptores de Haralick y redes neuronales artificiales*. Research in Computing Science, 2019. Publicación pendiente.

- Juan Jose Negron-Granados. *Calibración de la cámara Parrot Sequoia para el procesamiento de imágenes multiespectrales*. SOMI Congreso de Instrumentación, 2019. ISSN: 2395-8499. Publicación pendiente

## **1.5. Estructura de la tesis**

En este capítulo se ha descrito el planteamiento del problema, se ha realizado una revisión de la literatura y los objetivos de la presente tesis. En el capítulo 2 se presentan los fundamentos que se utilizan para llevar a cabo esta investigación. El capítulo 3 describe la metodología usada detallando sus diferentes etapas. En el capítulo 4 se exponen los resultados de las diferentes etapas descritas en la metodología. Finalmente, en el capítulo 5 se presentan las conclusiones y los trabajos futuros.

# Capítulo 2

## Marco teórico

### 2.1. Árbol de dzidzilché

En Yucatán existen unas 2400 especies vegetativas, de las cuales 600 son flora apícola y solo 30 se consideran de gran interés para la producción de miel [18]. Entre las plantas que forman parte de la vegetación apícola, destaca el dzidzilché, pues juega un papel muy importante en la industria apícola, ya que de la miel que se produce con el polen de sus flores es de las más cotizadas en el mundo [6].

El dzidzilché es un árbol que puede llegar a medir hasta 12 metros de altura, sus meses de floración van desde febrero hasta mayo [4]; es una especie nativa de Mesoamérica, que se distribuye en México, por los estados de Oaxaca, Tabasco, Yucatán, Campeche y Quintana Roo; y por los países de Belice, Guatemala y Honduras [6]. Esta especie juega un papel importante en la apicultura, pues la miel que hacen las abejas con el polen de sus flores es de las más cotizadas en el mundo. Yucatán se caracteriza por ser el principal productor de miel en México, pues el estado cuenta con más de 11,000 productores de miel [19], siendo el primer exportador de miel en el mundo, donde el 90% de la producción tiene como destino países latinoamericanos, Estados Unidos, Alemania y Japón.

### 2.2. Área de estudio

En el periodo de 2012 a 2016, Yucatán produjo 11 millones de toneladas de miel cada año [21], por lo que, hay mucha actividad apícola distribuida en todo el estado, incluso, se puede encontrar productores en los municipios cercanos a la ciudad de Mérida, capital de Yucatán, y esto



Figura 2.1: Flores del árbol de dzidzilché. (Fuente: [www.cicy.mx](http://www.cicy.mx))

conlleva a que es posible encontrar abundante flora polinífera alrededor de estas comunidades.

El Campus de Ciencias Biológicas y Agropecuarias (CCBA) de la Universidad Autónoma de Yucatán (Figura 2.2) destaca por realizar trabajos de investigación en apicultura, debido a esto, el campus cuenta apiarios y a sus alrededores se encuentran extensos terrenos de flora apícola (Figura 2.3), tales características hacen del CCBA el lugar ideal para realizar la toma de fotos para crear el conjunto de datos que utilizará el algoritmo de aprendizaje máquina.

### **2.3. Espectro electromagnético**

El espectro electromagnético (Figura 2.4) es el conjunto de todos los tipos de radiación ordenados de acuerdo a su longitud de onda. Para el estudio de los recursos naturales suele utilizarse la radiación solar como fuente de ondas electromagnéticas [23]. Se puede utilizar la reflectancia de la radiación solar en la superficie terrestre, pues la energía electromagnética reflejada está en función del tipo de cubierta presente sobre ella [14]. La reflectancia se recoge por el sensor y la información suele ser almacenada como un conjunto de imágenes.



Figura 2.2: Facultad de Veterinaria de la Universidad Autónoma de Yucatán. (Fuente: [www.ccba.uady.mx](http://www.ccba.uady.mx))



Figura 2.3: Ubicación del CCBA se puede notar que alrededor se conserva la flora nativa del estado de Yucatán. (Fuente: [maps.google.com](http://maps.google.com))

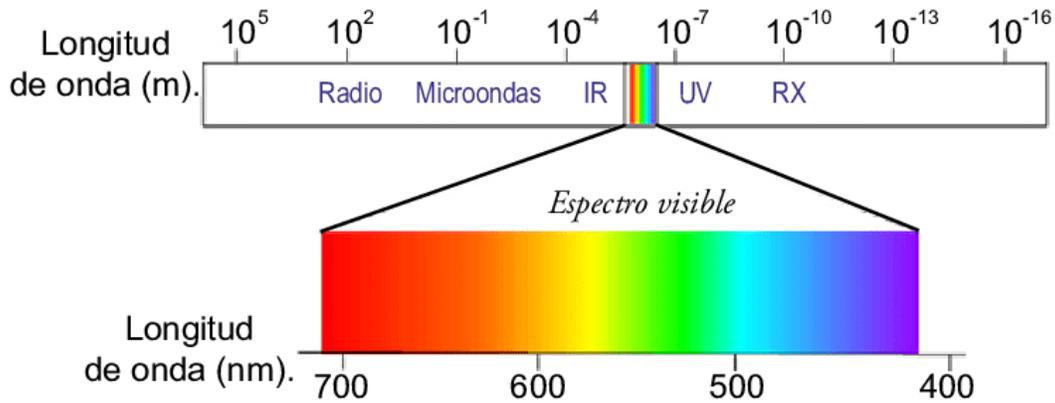


Figura 2.4: El espectro electromagnético [58].

## 2.4. Sensor multispectral

El sensor multispectral *Parrot Sequoia* está enfocado a la agricultura, pues usando sus dos módulos, el sensor es capaz de detectar la luz que las plantas absorben y reflejan; este sensor captura la información espectral de las bandas verde y roja, así como dos infrarrojas, invisibles para el ojo humano. Partiendo de la información espectral proporcionada por este sensor es posible saber el estado de salud vegetativo; además, este tipo información puede usarse para diferenciar tipos de plantas como propone esta investigación [50].

Este sensor multispectral está compuesto por dos módulos: el cuerpo (*body*), donde se encuentran las cámaras; y el sensor de luz (*sunshine sensor*). El cuerpo está integrado por cuatro cámaras espectrales de 1.2 Mpx, una cámara a color (RGB) de 16 Mpx, WiFi, una unidad de medición inercial (IMU, por sus siglas en inglés) y un magnetómetro, por otro lado, el sensor de luz cuenta con 4 sensores espectrales que tienen los mismos filtros que el cuerpo, sistema de posicionamiento global (GPS, por sus siglas en inglés), IMU y un magnetómetro (Figura 2.5).

La información de las diferentes frecuencias de la luz que se pueden obtener son las bandas: verde (530-570 nm), rojo (640-680 nm), borde rojo (730-740 nm) e infrarrojo cercano (770-810 nm); además de tomar fotos a color (Imágenes RGB). Las imágenes de las 4 bandas espectrales tienen formato RAW de 10 bits en un archivo TIFF. La imagen RGB se guarda en formato JPG, en total son 5 imágenes por cada captura que realiza la cámara.

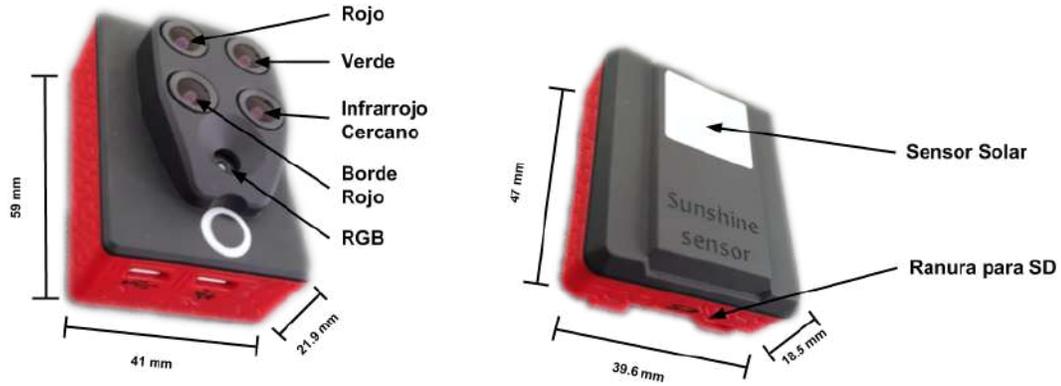


Figura 2.5: Dimensiones del sensor *Parrot Sequoia*. A la izquierda, el cuerpo del dispositivo. A la derecha, el sensor de luz solar.

### 2.4.1. Modelo de cámara y calibración

Las cámaras pueden ser caracterizadas por sus parámetros intrínsecos y extrínsecos, que describen la información interna y externa de la cámara, respectivamente. Los parámetros intrínsecos son aquellos que proporcionan información del funcionamiento de la cámara, los cuales son la distancia focal, el punto central y el modelo de calibración; por otro lado, los parámetros extrínsecos describen la posición y orientación de la cámara con respecto al mundo real [10].

$$\begin{aligned}
 R &= R_x(\omega)R_y(\phi)R_z(\kappa) & (2.1) \\
 &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\omega) & -\sin(\omega) \\ 0 & \sin(\omega) & \cos(\omega) \end{pmatrix} \begin{pmatrix} \cos(\phi) & 0 & \sin(\phi) \\ 0 & 1 & 0 \\ -\sin(\phi) & 0 & \cos(\phi) \end{pmatrix} \begin{pmatrix} \cos(\kappa) & -\sin(\kappa) & 0 \\ \sin(\kappa) & \cos(\kappa) & 1 \\ 0 & 0 & 1 \end{pmatrix}.
 \end{aligned}$$

Los parámetros externos de la cámara son diferentes para cada imagen, los cuales están dados por:

- $T = (T_x, T_y, T_z)$ , que representa la posición del centro de proyección de la cámara en el sistema de coordenadas del mundo real.
- $R$  que es la matriz de rotación que define la orientación de la cámara con los ángulos  $\omega, \phi, \kappa$ , los cuales definen la rotación en los ejes  $x, y$  y  $z$ , respectivamente (convención de PATB [7]). Donde  $R$  se construye a partir de la ecuación (2.1).

Si de la Figura 2.6 tomamos que  $\mathbf{X} = (X, Y, Z)$  es un punto 3D en el plano del mundo y el punto  $\mathbf{X}' = (X', Y', Z')$  se encuentra en el sistema de coordenadas de la cámara, entonces

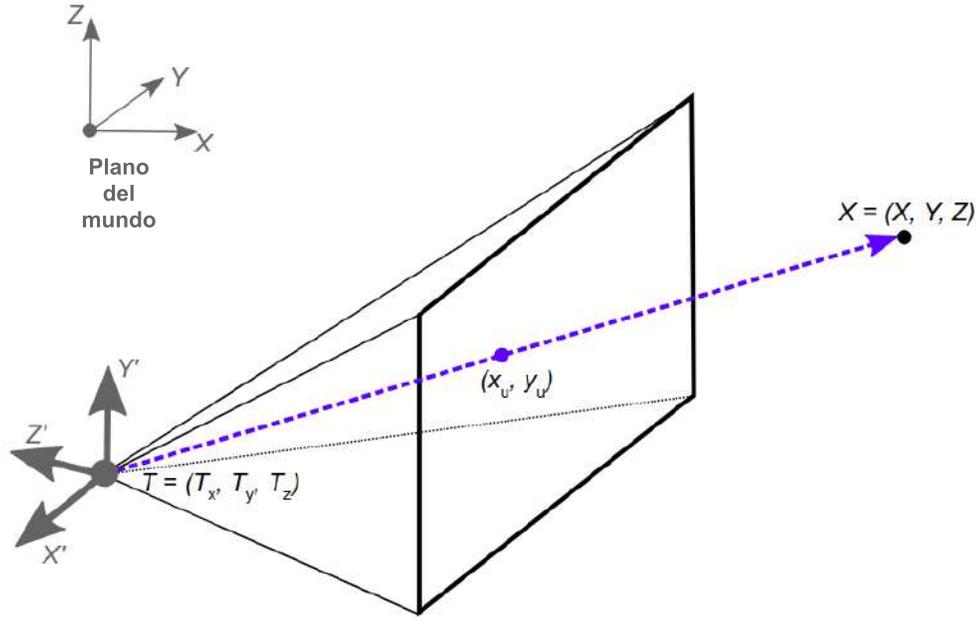


Figura 2.6: Representación de los parámetros externos de una cámara.

podemos relacionar ambos puntos por la relación dada en la ecuación (2.2).

$$\mathbf{X}' = R^T(\mathbf{X} - T). \quad (2.2)$$

$$\begin{pmatrix} x_u \\ y_u \\ 1 \end{pmatrix} = K \begin{pmatrix} \frac{X'}{Z'} \\ \frac{Y'}{Z'} \\ 1 \end{pmatrix} \quad (2.3)$$

$$\begin{pmatrix} x_u \\ y_u \\ 1 \end{pmatrix} = \begin{pmatrix} f & 0 & c_x \\ 0 & f & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \frac{X'}{Z'} \\ \frac{Y'}{Z'} \\ 1 \end{pmatrix}. \quad (2.4)$$

En un modelo sin distorsión el píxel con coordenadas  $(x_u, y_u)$  de la proyección de un punto 3D está dado por la ecuación (2.3). Donde K indica los parámetros internos de la cámara con el centro en  $(c_x, c_y)$  y distancia focal  $f$ , ambos parámetros están dados en píxeles (ecuación (2.4)). Sin embargo, el modelo de las cámaras en la *Parrot Sequoia* está definido por un tipo de distorsión llamado ojo de pescado.

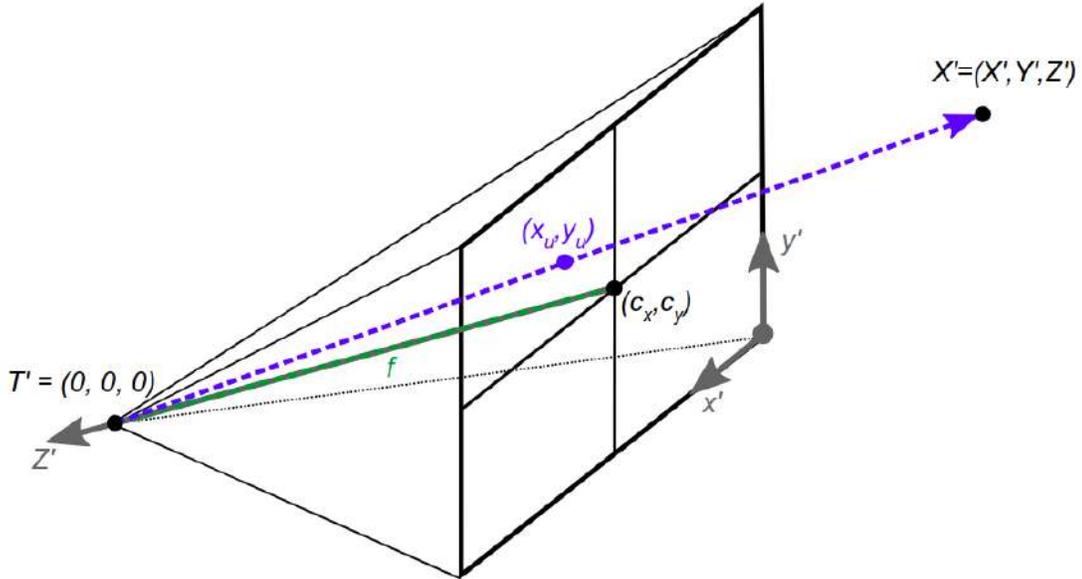


Figura 2.7: Geometría de la perspectiva en una cámara sin distorsión.

## 2.4.2. Distorsión ojo de pez

La distorsión de ojo de pez, mejor conocido por su nombre en inglés, *fish-eye distortion*, la cual se caracteriza por deformar los puntos del plano ideal de la imagen en una distribución circular, esto debido al modelo de la lente (Figura 2.8). Para corregir esta distorsión se requieren conocer los parámetros que definen la deformación afín (matriz afín) y los coeficientes del modelo de distorsión, a pesar de que en muchos casos estos parámetros son desconocidos y requieren técnicas de calibración para ser calculados, en este caso los parámetros son dados por el fabricante y se encuentran como metadatos en cada una de las imágenes espectrales proporcionadas por el sensor.

Primero, para corregir una imagen con distorsión de ojo de pez, se debe obtener el punto proyectado del plano del mundo real en el plano de la imagen (ecuación (2.2)), a continuación se debe calcular el valor de un parámetro  $\theta$  el cual se define en la ecuación (2.5), este valor se utiliza para calcular el coeficiente de distorsión  $\rho$  usando la ecuación (2.6); finalmente estos valores junto con la matriz afine (M) y el centro de la imagen es posible calcular la proyección del plano del mundo real en el plano de la imagen  $(x_d, y_d)$  sin la distorsión de ojo de pez (ecuación (2.7)) [55].

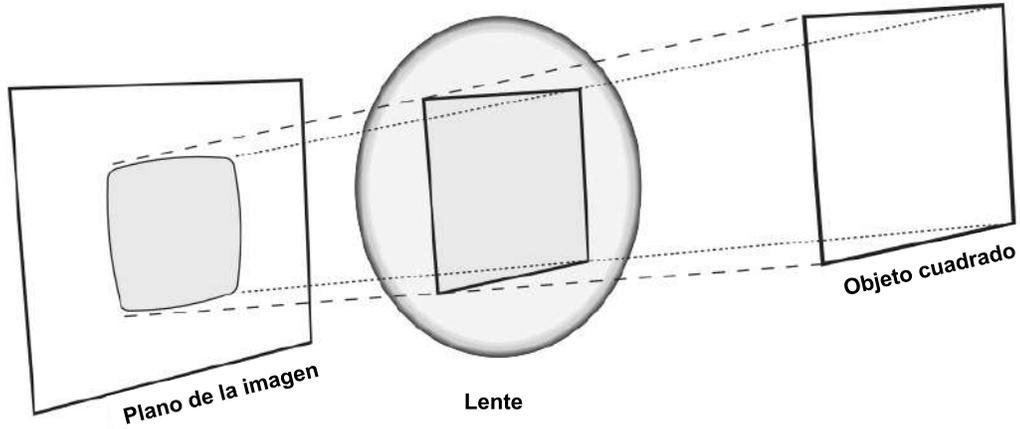


Figura 2.8: Representación de la distorsión ojo pescado de un objeto cuadrado (fuente:learningOpenCv).

$$\theta = \frac{2}{\pi} \arctan\left(\frac{\sqrt{X'^2 + Y'^2}}{Z'}\right), \quad \theta \in [0, 1]. \quad (2.5)$$

$$\rho = \theta + p_2\theta^2 + p_3\theta^3 + p_4\theta^4. \quad (2.6)$$

$$\begin{pmatrix} x_d \\ y_d \end{pmatrix} = M \begin{pmatrix} \frac{\rho X'}{\sqrt{X'^2 + Y'^2}} \\ \frac{\rho Y'}{\sqrt{X'^2 + Y'^2}} \end{pmatrix} + \begin{pmatrix} c_x \\ c_y \end{pmatrix}. \quad (2.7)$$

$$= \begin{pmatrix} C & D \\ E & F \end{pmatrix} \begin{pmatrix} \frac{\rho X'}{\sqrt{X'^2 + Y'^2}} \\ \frac{\rho Y'}{\sqrt{X'^2 + Y'^2}} \end{pmatrix} + \begin{pmatrix} c_x \\ c_y \end{pmatrix}. \quad (2.8)$$

Hasta este punto, se tienen las imágenes de cada cámara sin distorsión, si bien, hay que tomar en cuenta que dichas imágenes se encuentran desalineadas debido a la distribución de cada lente, en los metadatos de cada imagen se encuentran los parámetros para su alineación.

### 2.4.3. Alineación de las imágenes

Las imágenes de cada sensor espectral pueden ser relacionadas por constantes geométricas que se definen en los metadatos de cada imagen, esta información consiste en los ángulos de rotación  $\omega$ ,  $\phi$  y  $\kappa$ , cuyas unidades están en grados; los valores de traslación no se encuentran en los metadatos, debido a la versión de la cámara, sin embargo, el fabricante publicó una nota donde se aclara esta información [49].

Para empezar, se toma la imagen del sensor con espectro verde como banda de referencia, la cual se puede definir con la rotación ( $R_m$ ) y traslación ( $T_m$ ) respecto al plano de la tierra. Dada

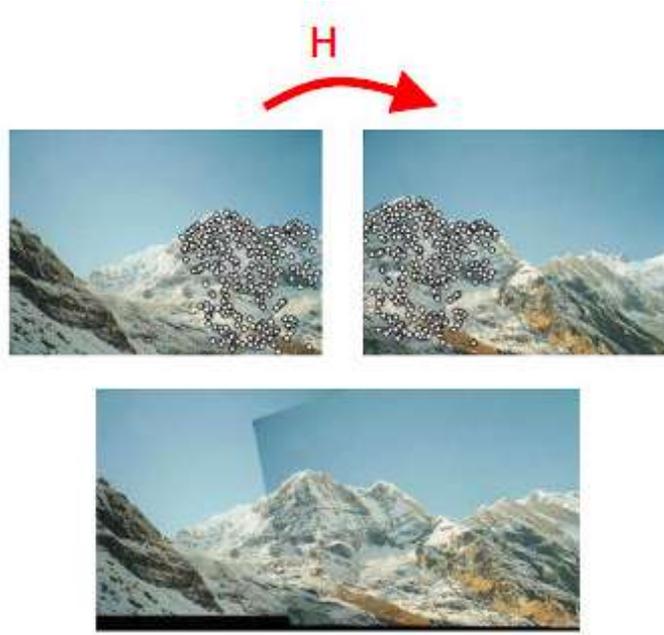


Figura 2.9: Ejemplo de la técnica de homografía (fuente:[47]).

esta referencia el resto de las imágenes pueden ser alineadas con la ecuación (2.9) dada por

$$X' = R_{rel}^T [R_m^T (X - T_m) - T_{rel}], \quad (2.9)$$

donde  $R_{rel}$  y  $T_{rel}$  es la rotación y traslación de cada imagen respecto a la referencia maestra, además,  $R_m$  suele definirse como matriz identidad de  $3 \times 3$ , y  $R_m$  como un vector 0 [55].

#### 2.4.4. Homografía

$$\begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} = H \begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x_2 \\ y_2 \\ 1 \end{pmatrix}. \quad (2.10)$$

En geometría, se define como homografía a la matriz  $H$  capaz de relacionar dos puntos  $(x_1, y_1)$  y  $(x_2, y_2)$  en el espacio (ecuación 2.10, esta transformación al ser aplicada en imágenes es capaz de alinear dos tomas diferentes en una misma escena (imagen 2.9); en la matriz de homografía el vector de traslación está dado por  $(h_{13}, h_{23})$ .

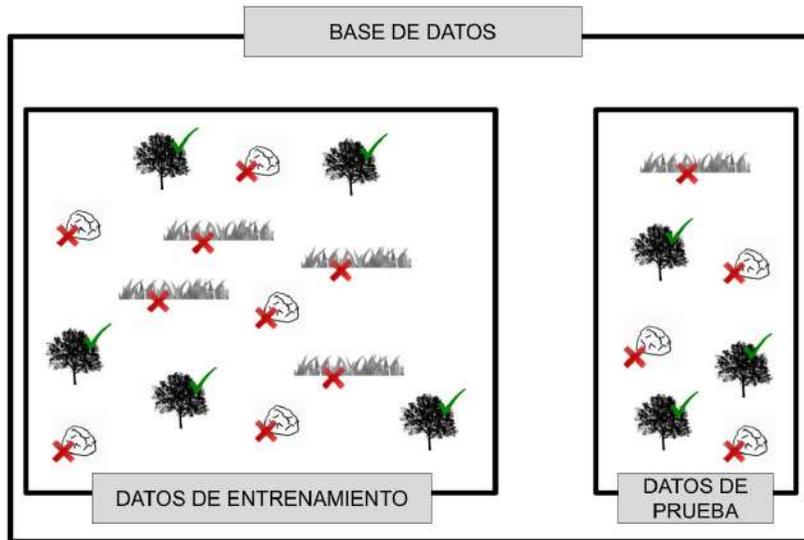


Figura 2.10: Base de datos etiquetada para ser usados en métodos de aprendizaje supervisado.

## 2.5. Aprendizaje Máquina

El aprendizaje máquina, mejor conocido en inglés como *Machine Learning* (ML) es la ciencia y arte de la programación computacional que permite a las máquinas aprender mediante datos de entrada [26]; Arthur Samuel, pionero de la inteligencia artificial, definió ML como el campo de estudio que da a las computadoras la habilidad de aprender sin ser programadas explícitamente (1959). Existen diferentes tipos de algoritmos de ML, por lo tanto es posible clasificarlos en diferentes categorías, si se clasifican de acuerdo si requieren o no supervisión humana estos pueden ser: supervisado, no supervisado, semisupervisado o aprendizaje por refuerzo [26].

En este trabajo se utiliza la categoría de aprendizaje supervisado, el cual se caracteriza por requerir datos de entrada, normalmente denotado por  $X$  que están etiquetadas por una salida  $Y$ ; los datos de entrada pueden ser descriptores, audios, imágenes o simplemente datos y se les conoce como variables independientes. Los datos salida o etiquetas son conocidos como variables dependientes y de acuerdo al objetivo del algoritmo utilizado puede ser un vector, imagen, sonido o palabras entre otras [26, 30]. El problema que se aborda en esta investigación es tipo de clasificación, pues a cada entrada le corresponde una etiqueta de salida, específicamente se trata de un clasificador binario pues la etiqueta de salida solo puede ser 1 o 0 que corresponde si en la imagen está presente el árbol de dzidzilché o no, respectivamente; esto con el fin de clasificar correctamente nuevos datos de entrada (Figura 2.10).

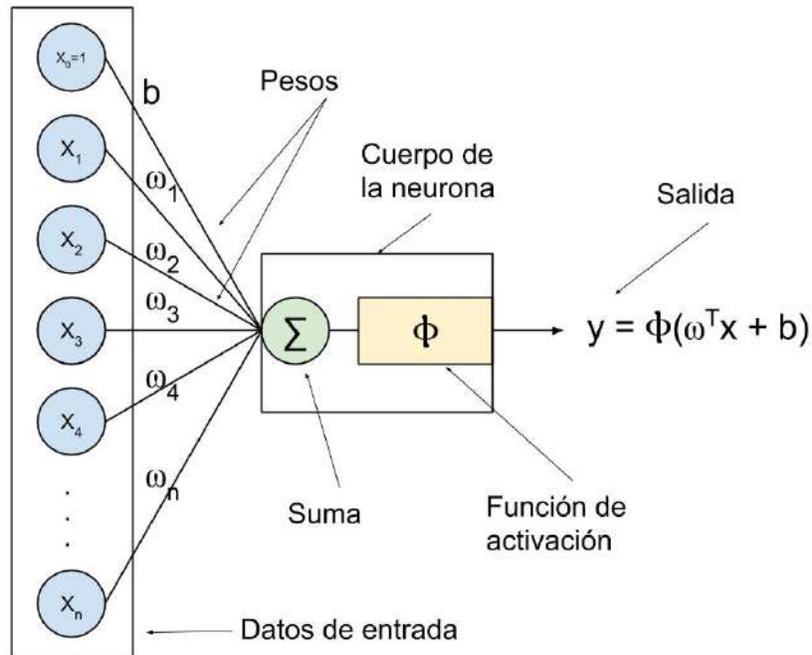


Figura 2.11: Estructura de una neurona artificial.

## 2.6. Red neuronal

### 2.6.1. Red neuronal artificial

La neurona artificial están inspiradas en las neuronas biológicas, por lo que comparten varias similitudes, al igual que en la biología, las neuronas artificiales tienen dendritas, ramificación de la neurona que reciben la información de sus neuronas vecinas y que pueden atenuar o amplificar los datos de entrada; luego, la información de cada dendrita pasa al cuerpo de la neurona, donde se suman los valores de cada dendrita; posteriormente el resultado de la suma pasa por una función de activación para tener la salida dentro de un rango deseado (Figura 2.11), normalmente entre 0 y 1, pero también puede ser entre 1 y  $-1$  o entre el rango  $(0, \infty)$  entre otras, lo cual depende de la función de activación usada [51], esta neurona artificial recibe el nombre de perceptrón [66].

El uso del perceptrón es muy efectivo para estimar modelos de aprendizaje supervisado, datos complejos no lineales pueden ser modelados con varios perceptrones interconectados, mejor conocido con el término de Red Neuronal Artificial (ANN por sus siglas en inglés *Artificial Neuronal Network*) [66]. Una de las arquitecturas de ANN a más utilizadas es el perceptrón multicapa completamente conectado (MLP por sus siglas en inglés *Multi Layer Perceptron*), dicha arquitectura está compuesta por una capa de entrada por donde se introducen los datos en la red,

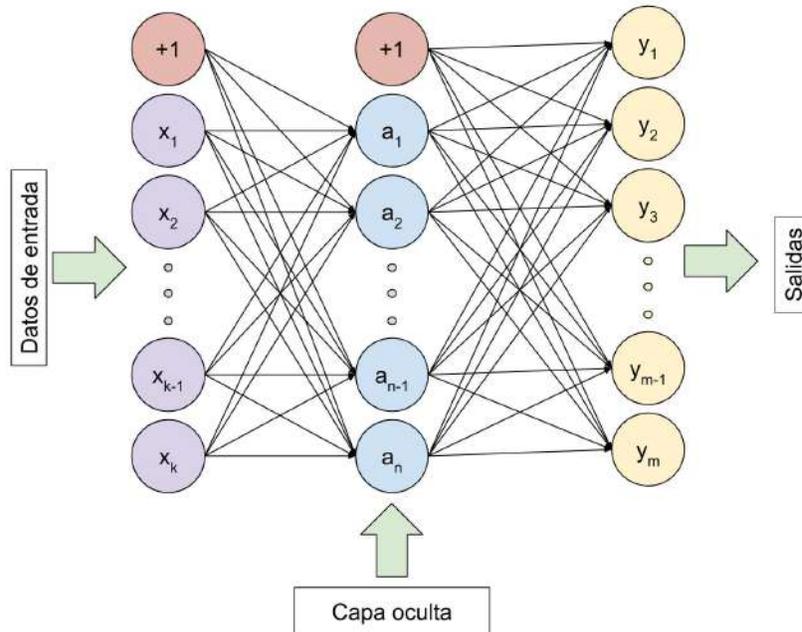


Figura 2.12: Ejemplo de perceptrón multicapa completamente conectado con un vector de entrada de tamaño  $k$ , una capa oculta de  $n$  neuronas y  $m$  salidas.

posteriormente se encuentran las capas ocultas compuesta por neuronas ordenadas cuyas entradas están conectadas a una capa anterior y cuyas salidas son la entrada de una siguiente capa, finalmente la capa de salida es donde se encuentran los valores finales de la ANN [17]; además se le denomina completamente conectada debido a que cada neurona de la capa de entrada y de las ocultas tienen sus salidas conectadas a cada neurona de la siguiente capa. Un MLP con una capa oculta (Figura 2.12) es conocido como aproximador universal, sin embargo es posible mejorar los resultados agregando más capas ocultas [31].

### 2.6.2. Red neuronal convolucional

Otra arquitectura utilizada es la Red Neuronal Convolutiva (CNN, por sus siglas en inglés *Convolutional Neural Network*), cuyo objetivo es extraer características usando el operador convolutivo. Este tipo de arquitectura es adecuada para la clasificación de imágenes y para el análisis del sonido [52]; existen muchas variantes de CNN, sin embargo la estructura general de esta red se compone por una capa de entrada, una capa de extracción de características (aprendizaje) y una capa de clasificación (Figura 2.13).

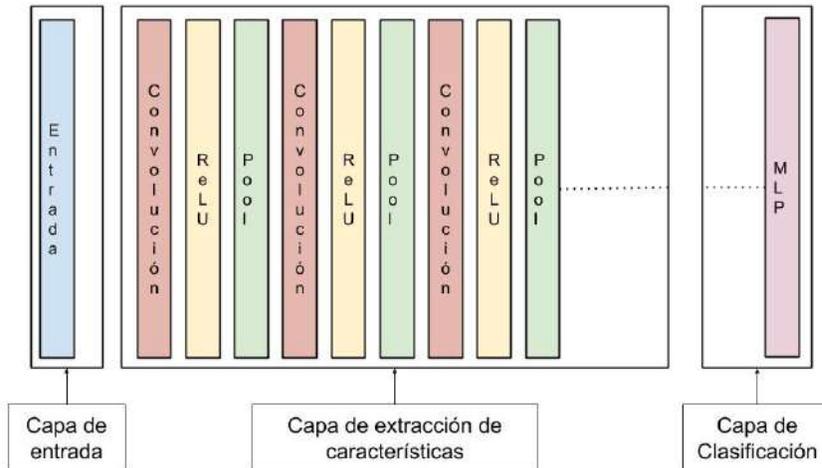


Figura 2.13: Estructura general de una Red Neuronal Convolutional [52].

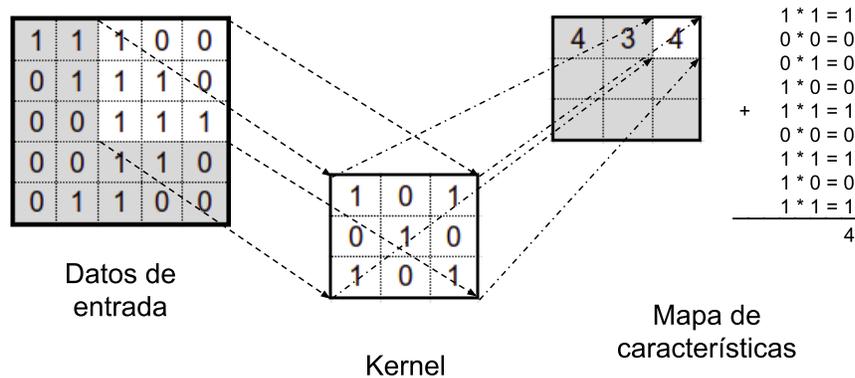


Figura 2.14: El operador de convolución.

### Operador de convolución

La convolución se define como una operación matemática que describe una regla sobre cómo combinar dos conjuntos de información, es de gran importancia para la física y las matemáticas pues define un puente entre el dominio del espacio o tiempo y de la frecuencia, en una CNN se transforman los datos de entrada en un mapa de características [52]. La operación se encuentra definida por

$$(I \otimes K)(p_x, p_y) = S(p_x, p_y) = \sum_i \sum_j I(p_x - i, p_y - j) * K(i, j), \quad (2.11)$$

donde  $I$  es un matriz,  $S$  es el kernel de tamaño  $i, j$  y  $S$  es la salida de la operación; la Figura 2.14 muestra un ejemplo de cómo se calcula.

$$\frac{N - F}{S} + 1, \quad (2.12)$$

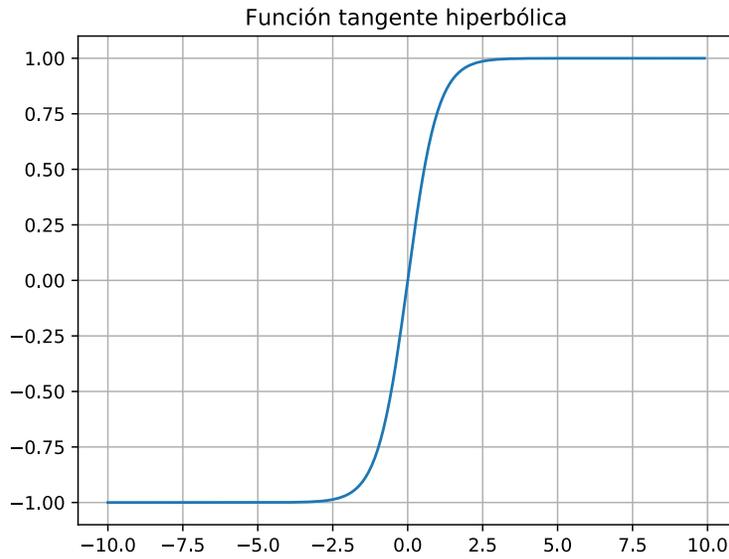


Figura 2.15: Función tangente hiperbólica.

donde  $N$  es la dimensión de la matriz,  $F$  la dimensión del kernel y  $S$  es el desplazamiento.

### Función de activación

Seguido de la capa de convolución es común aplicar una función de activación no lineal a la salida, de manera formal, si  $\omega_{i,d}$  son los pesos que conectan la entrada  $d$  con la capa oculta  $i$  entonces los valores de la capa  $h_i$  están dados por

$$h_i = f(\omega_i \cdot x), \quad (2.13)$$

donde  $f$  es la función de activación,  $\omega_i$  se refiere al vector de pesos en la capa oculta  $i$  y  $x$  corresponde al vector de entrada. La función de activación más utilizada es la tangente hiperbólica,  $\tanh$  [17].

### Submuestreo

La capa de submuestreo mejor conocido en inglés como *pooling*, es el procedimiento donde se realiza un resumen estadístico de los parámetros de entrada, los cuales, en este caso, corresponden a conjuntos de matrices bidimensionales. En este trabajo se hace uso del agrupamiento por promedio (*Average pooling*), el cual consiste en desplazar sobre una matriz bidimensional una región de tamaño  $L \times L$  y calcular el promedio de dicha región cada  $s$  pasos (Figura 2.16).

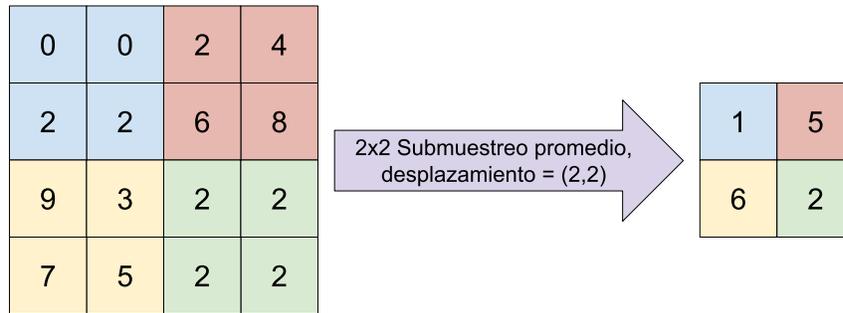


Figura 2.16: Ejemplo de submuestreo por promedio (*Average pooling*).

### 2.6.3. Red convolucional *LeNet*

En 1990 Le Cun *et al.* propusieron una red neuronal convolucional especializada para reconocer dígitos escritos a mano en imágenes bidimensionales [32]; esta red neuronal, conocida como *LeNet-1*, surge como propuesta para resolver el dilema entre redes neuronales pequeñas que no podían aprender del conjunto de datos, y de grandes redes neuronales que parecen estar sobreparametrizadas [34]. En la Figura 2.17 se presenta la arquitectura de red *LeNet-1*, donde se puede notar que esta red solo utiliza las características aprendidas para clasificar mediante una capa de convolución después de la segunda capa de muestreo. La tasa de error fue de 1.7% [34].

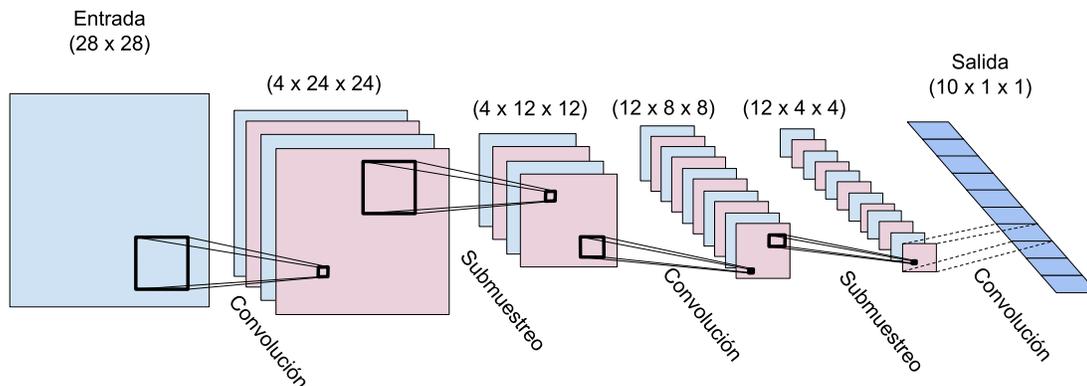


Figura 2.17: Arquitectura de la red neuronal convolucional *LeNet-1*.

Posteriormente surge la red convolucional *LeNet-4*, la cual se caracteriza para hacer más óptimo el entrenamiento de grandes conjuntos de datos, su tasa de error fue de 1.1% [34]. Finalmente, la red neuronal *LeNet-5*, tiene una arquitectura similar a la *LeNet-4*, sin embargo, se le agregaron más mapas de convoluciones, y se utilizó una red neuronal completamente conectada como clasificador; la tasa de error, en el problema de detectar números escritos a mano, fue de 0.9% [34]. La *LeNet-5* es la arquitectura usada en el presente trabajo y se encuentra ampliamente descrita en la sección 3.5.

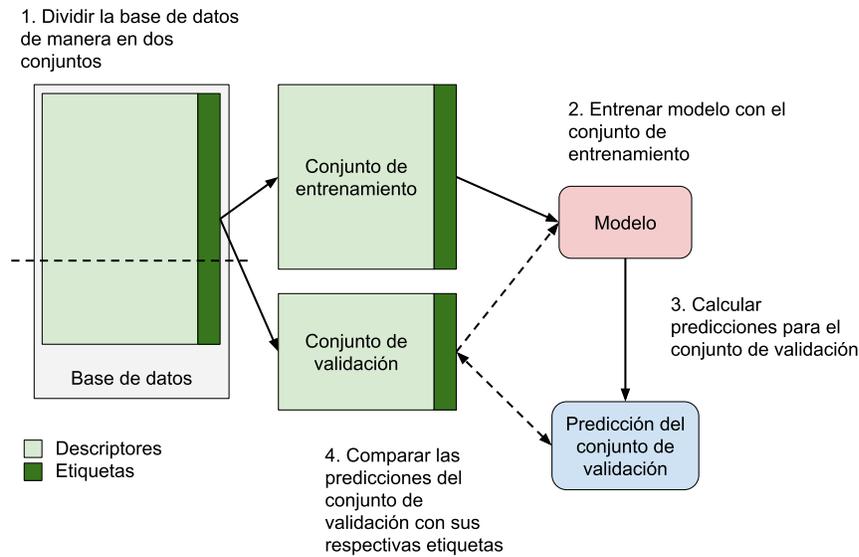


Figura 2.18: Diagrama de flujo del método de validación cruzada.

## 2.7. Validación cruzada

La validación cruzada es un método de evaluación que proporciona indicadores de cómo se desempeñará un modelo predictivo ante datos que no ha visto, el tipo más simple de validación cruzada consiste en dividir la base de datos en conjuntos de entrenamiento y prueba (Figura 2.18); el modelo se entrena con el conjunto de entrenamiento, luego se calculan los valores de salida para el conjunto de prueba y se compara con sus respectivas etiquetas, lo que permite evaluar el modelo ante nuevos datos [46], sin embargo, una mejor manera de evaluar la robustez del modelo ante futuros parámetros es la validación cruzada de  $k$  iteraciones, aunque computacionalmente es más complejo que el anterior [8].

En la validación cruzada de  $k$  iteraciones se parte la base de datos en  $k$  partes iguales, luego, se entrena el modelo con  $k - 1$  partes del conjunto de datos y se valida con el segmento restante (Figura 2.19), este proceso se repite  $k$  veces tomando en cada iteración un diferente segmento para la validación (segmento  $i$ ) [17]. Finalmente se calculan las predicciones para cada segmento  $i$  y se compara con sus etiquetas correspondientes [8]. Se puede utilizar el número de particiones  $k$  que se desee, sin olvidarse que mientras más particiones más tiempo computacional se requerirá.

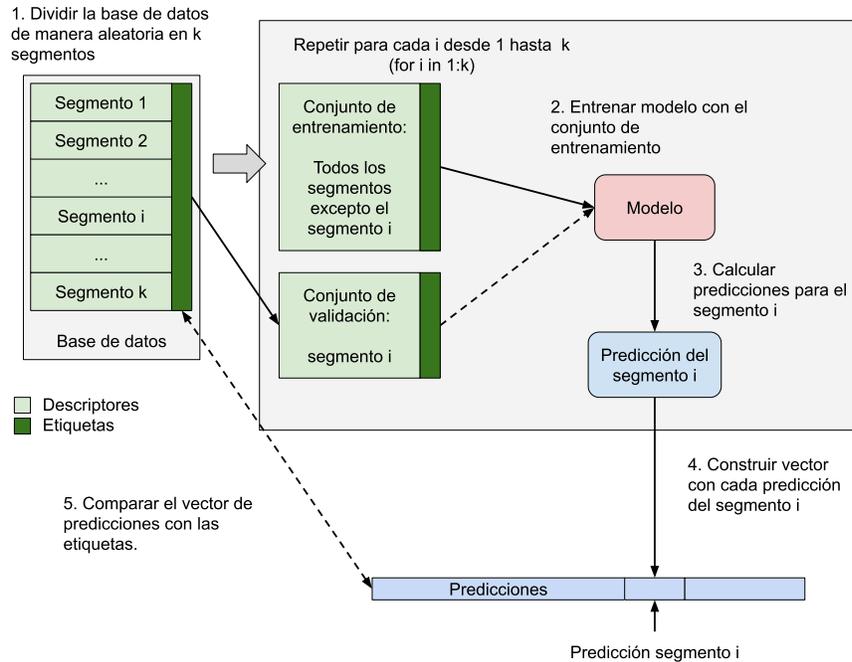


Figura 2.19: Diagrama de flujo del método validación cruzada de  $k$  iteraciones.

## 2.8. Evaluación del modelo

A continuación se describen los métodos estadísticos empleados para evaluar el rendimiento del modelo, utilizando el conjunto de datos para pruebas en los modelos entrenados.

### 2.8.1. Matriz de confusión

Dado un modelo clasificador binario y un elemento a clasificar (instancia), existen 4 salidas posibles. Si la instancia es positiva y es clasificada como positiva se categoriza como verdadero positivo (VP); si es clasificada como negativa es considerada como un falso negativo (FN). De igual manera, si la instancia evaluada es negativa y es clasificada como negativa se trata de un verdadero negativo (VN), si se clasifica como positiva se nombra como falso positivo (FP). Si evaluamos el conjunto de datos para prueba es posible construir una tabla que contabilice los 4 posibles casos anteriores, dicha tabla es conocida como matriz de confusión o tabla de contingencia [22]; la Figura 2.20 muestra su composición.

Partiendo de la matriz de confusión es posible calcular valores estadísticos que cuantifiquen el error de confusión del clasificador. El primer valor que suele ser calculado es la precisión, la cual determina el porcentaje de elementos correctamente clasificados y es calculado con la fórmula

		Etiqueta	
		Positivo	Negativo
Predicción	Positivo	Verdadero Positivo	Falso Positivo
	Negativo	Falso Negativo	Verdadero Negativo
Total		P	N

Figura 2.20: Composición de la matriz de confusión.

$$precisión = \frac{VP + VN}{P + N}, \quad (2.14)$$

donde  $P$  y  $N$  son el total de instancias con etiqueta positiva y negativa, respectivamente. Otros estadísticos relevantes que se pueden calcular a partir de la matriz de confusión son la sensibilidad y la especificidad [57]; la primera es el porcentaje de instancias positivas que fueron correctamente clasificadas y se determina con

$$sensibilidad = \frac{VP}{P} = \frac{VP}{VP + FN}. \quad (2.15)$$

Finalmente, la especificidad es la proporción de instancias negativas que fueron clasificadas como negativas, la fórmula para calcular la proporción está dada por

$$especificidad = \frac{VN}{N} = \frac{VN}{VN + FP}. \quad (2.16)$$

### 2.8.2. Curva ROC y AUC

La curva Característica Operativa del Recepto o curva ROC (por sus siglas en inglés *Receiver Operating Characteristics*) es una técnica para la visualización, organización y selección de clasificadores basados en su desempeño. La curva ROC es de 2 dimensiones, en el eje  $x$  indica  $1 - Especificidad$ , y el eje  $y$  la *Sensibilidad* y es creada variando el umbral del clasificador en-

Tabla 2.1: Ejemplo de resultados de diferentes clasificadores y sus respectivas etiquetas.

Etiqueta	C0	C1	C2	Etiqueta	C0	C1	C2
0.0	0.08	0.08	0.08	1.0	0.9	0.9	0.9
0.0	0.03	0.03	0.03	1.0	0.94	0.94	0.06
0.0	0.08	0.08	0.92	1.0	0.99	0.01	0.99
0.0	0.1	0.9	0.1	1.0	0.92	0.92	0.08
0.0	0.09	0.09	0.09	1.0	0.92	0.08	0.92
0.0	0.02	0.02	0.02	1.0	0.93	0.93	0.93
0.0	0.02	0.98	0.02	1.0	0.07	0.07	0.93
0.0	0.08	0.92	0.08	1.0	0.0	1.0	1.0
0.0	0.08	0.92	0.08	1.0	0.08	0.92	0.92
0.0	0.02	0.98	0.02	1.0	0.92	0.92	0.92

tre  $[0, 1]$  al momento de evaluar el conjunto de evaluación, de esta manera variamos las tasas de especificidad y sensibilidad. Sin embargo, para cuantificar y evaluar el desempeño del modelo evaluado mediante la curva ROC, es necesario calcular su área bajo la curva (AUC), la cual se encuentra entre un rango de  $[0, 1]$ , e indica la probabilidad de que el clasificador detecte como positiva una instancia positiva, por tanto, mientras más cercana el área sea a 1, mejor será el clasificador [22].

La Tabla 2.1 presenta un ejemplo de 3 diferentes clasificadores, nombrados C0, C1 y C2; para un conjunto de 20 datos etiquetados conformado por 10 elementos con etiqueta 1 y 10 con la etiqueta 0. Luego se calcula la especificidad y la sensibilidad para diferentes valores en el umbral del clasificador; esto construye la curva presentada en la Figura 2.21, y poder calcular el área bajo cada curva. En este ejemplo se observa que el clasificador 1 (C1) tiene una área igual a 0.47, pues en la tabla presenta más valores que pueden ser tomados como falsos positivos (5 elementos más cercanos a 1 que a 0); por otro lado, nótese que los clasificadores 0 y 2 (C0 y C2, respectivamente) tienen el mismo número elementos que pueden caer en la categoría de falsos positivos, sin embargo, C2 presenta mejor rendimiento pues C0 tiene más candidatos a falsos negativos, esto se puede verificar en las áreas bajo la curva de los clasificadores 0 y 2, siendo el área de C2 mayor que la de C0.

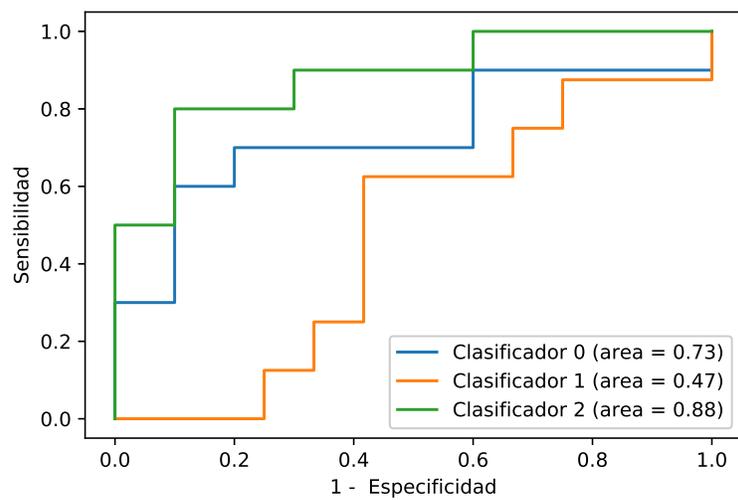


Figura 2.21: Ejemplo de curvas ROC para distintos clasificadores, con su respectiva área bajo la curva (AUC).

# Capítulo 3

## Metodología

En el presente capítulo se describe el procedimiento para detectar el árbol de dzidzilché utilizando las imágenes espectrales del dispositivo *Sequoia Parrot* y una arquitectura de red neuronal convolucional, este método incluye la captura de las imágenes espectrales, la corrección de la distorsión de ojo de pescado, la alineación de las imágenes y la creación de la base de datos, de igual forma se explica la arquitectura de la red neuronal planteada, el método del entrenamiento y cómo se evalúa el modelo.

### 3.1. Descripción general

La propuesta que se hace para detectar el árbol de dzidzilché consiste en la adquisición de imágenes aéreas multiespectrales utilizando el sensor espectral *Parrot Sequoia* de las zonas vegetativas del área de estudio donde haya presencia del árbol en cuestión, posteriormente se realiza la corrección del ojo de pescado y el alineamiento de cada conjunto de imágenes, en este punto las dimensiones de las imágenes son de (1200,900,4); luego, cada imagen es dividida en una cuadrícula de  $12 \times 9$  (donde cada cuadro es de 100 píxeles por lado) para su etiquetado manual.

El criterio para la asignación manual de la etiqueta consiste en que si el segmento de  $100 \times 100$  píxeles tiene en toda su cobertura el árbol de dzidzilché se etiqueta con el valor numérico 1, por otro lado, si toda la cobertura corresponde a otros objetos que no sean el árbol de dzidzilché se clasificará como 0; si algún cuadrante al ser analizado tiene presencia del árbol de dzidzilché y otros objetos, entonces ese cuadrante no se etiqueta por lo que no forma parte del conjunto de datos. Luego de que los datos han sido etiquetados se realiza un aumento de datos (*Data Augmentation*) dividiendo cada elemento de (100,100,4) en 16 partes de dimensión (25,25,4). Después de realizar la división, se parte el conjunto de datos en los conjuntos de entrenamiento

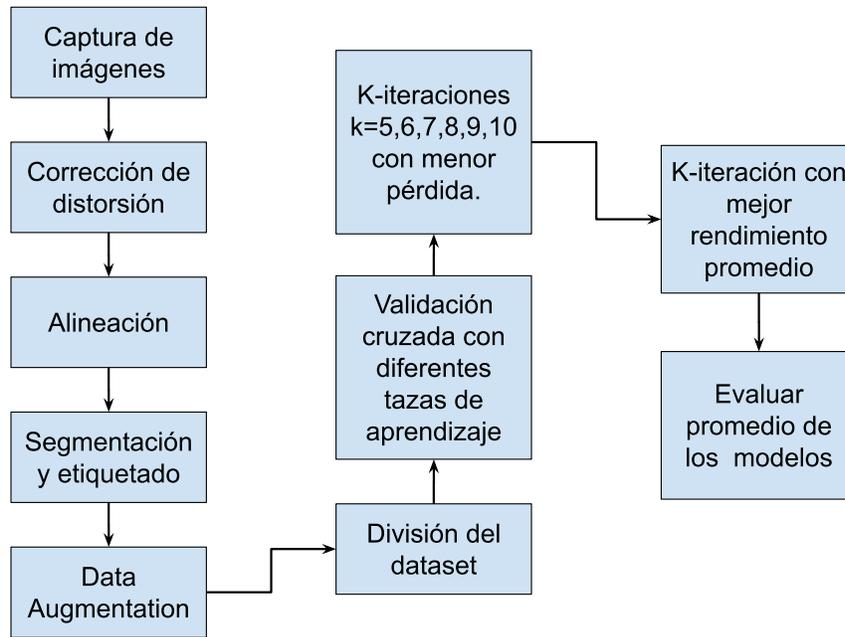


Figura 3.1: Gráfico general de la metodología propuesta.

y prueba que se usarán para entrenar y evaluar el modelo, respectivamente.

Teniendo los conjuntos de entrenamiento y prueba formados, se toma el conjunto de entrenamiento y se divide en 2 subconjuntos: datos de entrenamiento y validación en proporción 70% y 30%, respectivamente, que servirán para entrenar el modelo neuronal convolucional propuesto con diferentes tasas de aprendizaje, con el objetivo de encontrar la mejor tasa para entrenar el modelo usando validación cruzada de  $k$  iteraciones. Se comparan las funciones de pérdida y el número de épocas requerido para llegar al valor mínimo de la función de pérdida para determinar el número aproximado de épocas con el cual el modelo debe ser entrenado en la siguiente etapa del método.

Una vez definidos la tasa de aprendizaje y el número aproximado de épocas en el paso anterior, se procede a entrenar el modelo usando validación cruzada de  $k$  iteraciones para  $k = 5, 6, 7, 8, 9, 10$ . A continuación, por cada iteración  $k$  se les calcula el promedio y la desviación estándar del error, con la finalidad de evaluar la robustez del modelo. Por último, se escoge la iteración con mejor rendimiento y se selecciona el mejor modelo de la iteración para evaluar los datos de prueba y determinar el rendimiento ante nuevos datos; con las predicciones que realice el modelo y las etiquetas correspondientes de los datos se puede elaborar la matriz de confusión que cuantifique los datos clasificados correctamente (verdaderos positivos y verdaderos negativos) y los que no (falsos positivos y falsos negativos); y a partir de esos datos es



Figura 3.2: Zona del CCBA donde se tomaron las imágenes multiespectrales.

posible determinar la exactitud, la sensibilidad y la especificidad, de igual manera se determina la *Curva Característica Operativa del Receptor* y su área bajo la curva [22].

## 3.2. Etapa de captura

La captura de imágenes se realizó mediante un vehículo aéreo no tripulado (dron) equipado con la cámara multiespectral *Parrot Sequoia*, lo que permitió la toma de imágenes multiespectrales en los terrenos del CCBA, a las afueras de la ciudad de Mérida en las coordenadas geográficas  $20^{\circ}52'04.5''N, 89^{\circ}37'28.5''W$  (Figura 3.2), lugar en los que crece abundantes árboles de dzidzilché. La captura se realizó en marzo entre 9 y 11 a.m. a una altura de vuelo alrededor de los 5 metros.

## 3.3. Etapa de corrección y alineación

Luego de la etapa de recolección de datos, se realiza la corrección de la distorsión del ojo de pescado, como se mencionó en la sección 2.4.2, la información de los parámetros intrínsecos y extrínsecos se encuentran como metadatos en la imagen, por lo que, se desarrolla un algoritmo en *Python* que lea los parámetros intrínsecos y se elimina la distorsión en la imagen siguiendo las fórmulas proporcionadas por el fabricante (ecuaciones (2.5-2.8)). Posteriormente, se realiza un recorte de la imagen, pues el resultado de la corrección presenta zonas con información nula.

Para realizar la alineación se hace uso de un tablero de ajedrez de  $7 \times 9$  cuadros, del que

se tienen 400 imágenes multiespectrales (100 por cada sensor espectral); el algoritmo para la alineación consiste en calcular las homografías del tablero sin distorsión por cada conjunto de 4 imágenes de una misma toma, las homografías se calculan en relación con la banda espectral verde (verde - rojo, verde - borde rojo, verde - infrarrojo), teniendo 3 homografías distintas por cada uno de las 100 tomas que se realizaron; a continuación, se promedian las homografías de cada relación dando la transformación geométrica para alinear las 4 bandas con referencia a la banda verde.

### 3.4. Etapa de etiquetado

En este punto de la metodología las imágenes se encuentran alineadas cuya dimensión es de  $1200 \times 900$  píxeles compuesta por 4 canales  $(1200, 900, 4)$ . Para realizar el etiquetado se genera una cuadrícula de  $12 \times 9$ , como apoyo visual para la asignación de etiquetas, por lo tanto, cada cuadro tendrá una dimensión de  $100 \times 100$  píxeles. Las etiquetas que se usan para asignarle valor a cada cuadrante son los números 1 y 0 que corresponden a si existe o no presencia de dzidzilché, respectivamente, en el cuadrante que se analiza. Para que la sección pueda ser de clase 1 toda el área debe contar con el árbol de dzidzilché, por otro lado, si en todo el cuadrante no hay presencia del árbol en cuestión se le asigna la clase 0, sin embargo, en un tercer caso, si el cuadrante que se analiza tiene elementos mezclados de las clases 0 y 1, entonces dicha sección no se etiqueta (Figura 3.3).

Siguiendo los criterios de asignación, se etiquetaron 1,200 cuadrantes compuestos por 600 elementos en cada clase. A continuación, se realiza un aumento en el conjunto de datos (*data augmentation*) dividiendo cada elemento en 16 cuadrados iguales de  $25 \times 25$  píxeles (Figura 3.4), además, las clases asignadas no se ven afectadas por este proceso debido a nuestro criterio de etiquetado; finalmente, el conjunto de datos aumentó de 1,200 a 19,200 elementos.

Para definir los conjuntos que se usaron en las etapas de entrenamiento, validación y de prueba, se dividió el conjunto formado en la etapa de *data augmentation*; de manera aleatoria, un 30% de los datos (5,760 elementos) formaron el conjunto de pruebas, el que se usó para evaluar el rendimiento final del modelo; el 70% de los datos (13,440 elementos) restantes se usaron para el entrenamiento del modelo usando la técnica de validación cruzada de  $k$  iteraciones; sin embargo, el modelo se entrenó antes usando validación cruzada simple para determinar la tasa de aprendizaje adecuada y el número de épocas aproximado que requirió el entrenamiento con validación cruzada de  $k$  iteraciones, por esto, los 13,440 datos que se reservaron para entrenar se dividen en dos subconjuntos en relación 70 – 30% que sirvieron en la etapa de entrenamiento

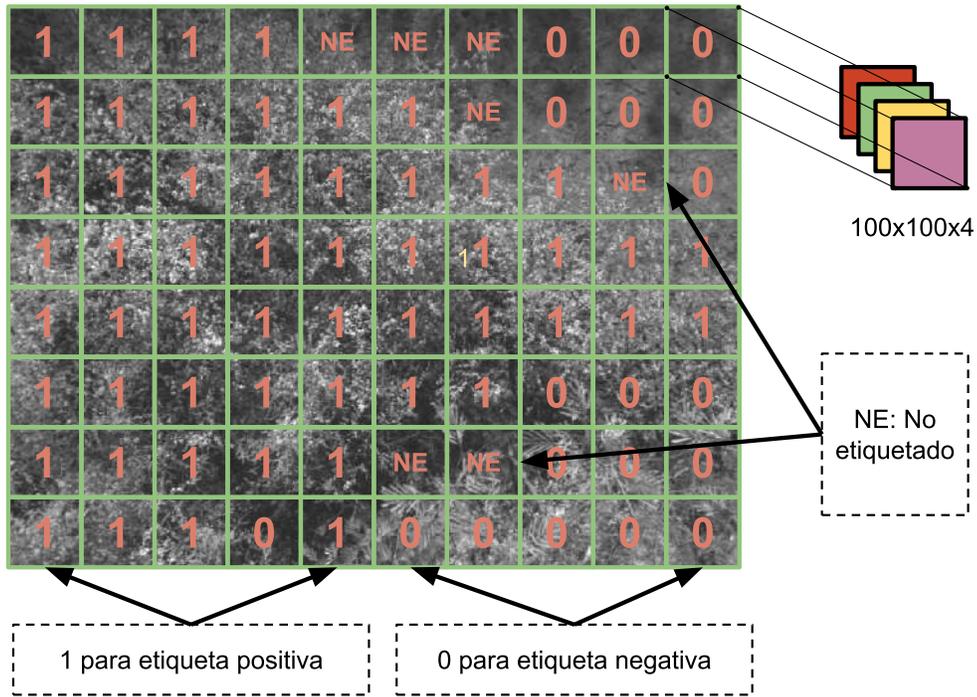


Figura 3.3: Ejemplo para el proceso de etiquetado.

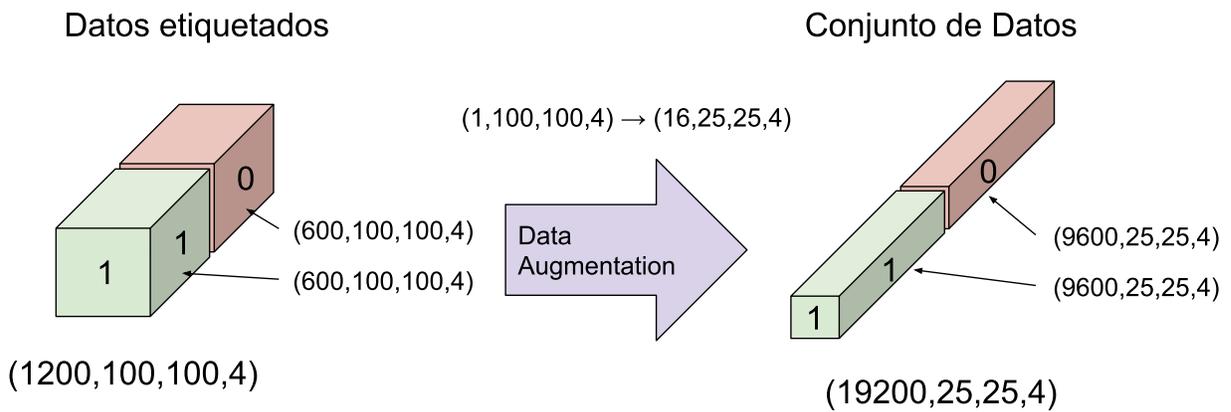


Figura 3.4: *Data augmentation*. Cada elemento del conjunto de datos etiquetados se divide en 16 partes iguales.

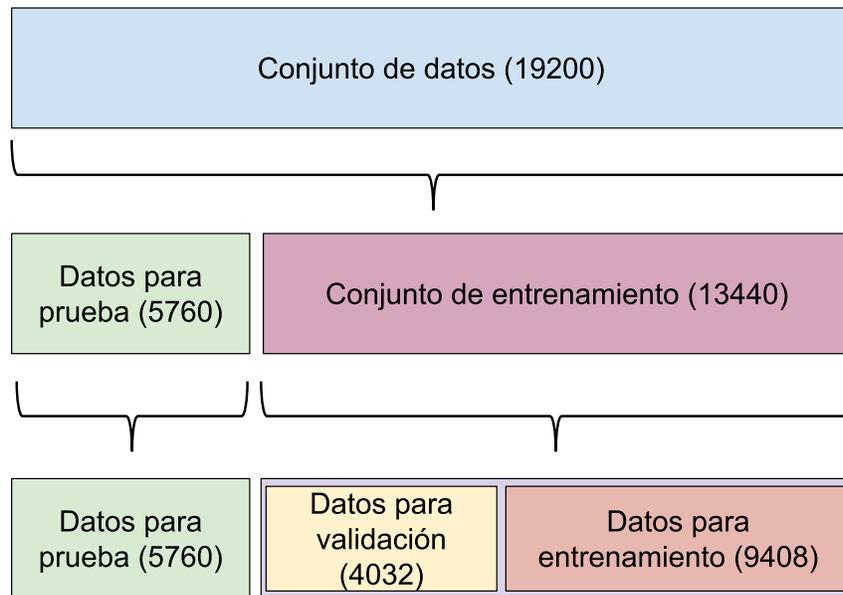


Figura 3.5: División del conjunto de datos.

y validación, respectivamente (Figura 3.5).

### 3.5. Etapa de entrenamiento

La arquitectura propuesta como clasificador consiste en una red neuronal convolucional típica, conocida como *LeNet* debido a ser una arquitectura bastante robusta [33]. Esta arquitectura se divide en dos partes, la primera consiste en el extractor de características y la segunda en el clasificador.

La entrada de nuestro modelo consiste de la imagen multispectral de dimensión  $(25, 25, 4)$  escalada a un rango entre  $[0, 1]$ ; a la cual se le aplica 6 filtros de dimensión  $(5, 5, 4)$ , el desplazamiento ( $s$ ) utilizado es de una unidad en los ejes  $x$  y  $y$  ( $s = (1, 1)$ ), por lo tanto, para que el desplazamiento pueda ser válido se requiere que la entrada tenga un *padding* de 2 unidades tanto en los márgenes laterales como en el superior e inferior ( $p = (2, 2)$ ). A continuación se realiza una capa de submuestreo promedio (*Average Pooling*) con una ventana de muestreo  $k = (2, 2)$  para mapa de características generados en la capa anterior; la ventana de muestreo tiene un desplazamiento  $s = (1, 1)$  sin necesidad de añadir *padding* ( $p = (0, 0)$ ). La salida de esta capa es de dimensión  $(24, 24, 6)$ . La tercera capa consiste en una capa de convolución con 16 filtros de dimensión  $(5, 5, 6)$ , un desplazamiento  $s = (1, 1)$  y un *padding* cero ( $p = (0, 0)$ ). La salida consiste en 16 mapas de características de dimensión  $(20, 20)$ . Luego, se realiza un submuestreo

Tabla 3.1: Relación de parámetros que requieren ser entrenados en cada capa de la arquitectura.

Capa	Dimensión de Salida	Parámetros
Conv1	(25,25,6)	606
Prom. Pool	(24,24,6)	0
Conv2	(20,20,16)	2,416
Prom. Pool	(10,10,16)	0
Conv3	(6,6,120)	48,120
Capa de entrada	4,320	0
Capa oculta	82	362,964
Capa de salida	2	170
Total de parámetros	N/A	414,276

promedio a la salida de la capa anterior con una ventana  $k = (2, 2)$ , un desplazamiento  $s = (1, 1)$  y sin *padding* ( $p = (0, 0)$ ). El extractor de características termina con una tercera capa de convolución la cual tiene un filtro de dimensión  $(5, 5, 16)$ , un desplazamiento de  $s = (1, 1)$  y un *padding*  $p = (0, 0)$ . La salida de esta capa tiene una dimensión de  $(6, 6, 120)$  que corresponde a 4,320 características.

La segunda parte de la arquitectura *LeNet* consiste en una red completamente conectada con una capa oculta, cuyo vector de entrada son los 4,320 parámetros que se obtienen del extractor de características; la capa oculta de la red consiste en 84 neuronas y la capa de salida con 2 neuronas, una para clasificación positiva y una para la negativa. La Figura 3.6 presenta de manera gráfica la arquitectura *LeNet-5* con la descripción mencionada anteriormente y la Tabla 3.1 presenta la cantidad de parámetros que la red va a entrenar y cómo se distribuyen en las capas previamente descritas. Dicha arquitectura se implementó en el lenguaje de programación *Python* usando la biblioteca de redes neuronales *Keras* [1], que se ejecuta sobre *TensorFlow*, la cual es una biblioteca que se encarga de las operaciones matemáticas en el proceso de aprendizaje del modelo [2].

Antes de realizar el entrenamiento usando validación cruzada de  $k$  iteraciones, se realiza una validación cruzada simple para determinar la tasa de aprendizaje y el número de épocas que se usarán en la siguiente fase. Como se mencionó anteriormente, se hace uso del conjunto de entrenamiento donde, de manera aleatoria, se usan 9,408 elementos para entrenar y 4,032 para validar el modelo. El entrenamiento con la validación cruzada simple se realizó utilizando diferentes tasas de aprendizajes  $([0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09])$ , por lo que se generan 10 modelos, los cuales se usan para comparar la función de pérdida en la etapa de validación y escoger así la tasa de aprendizaje que permita al modelo aprender a discriminar

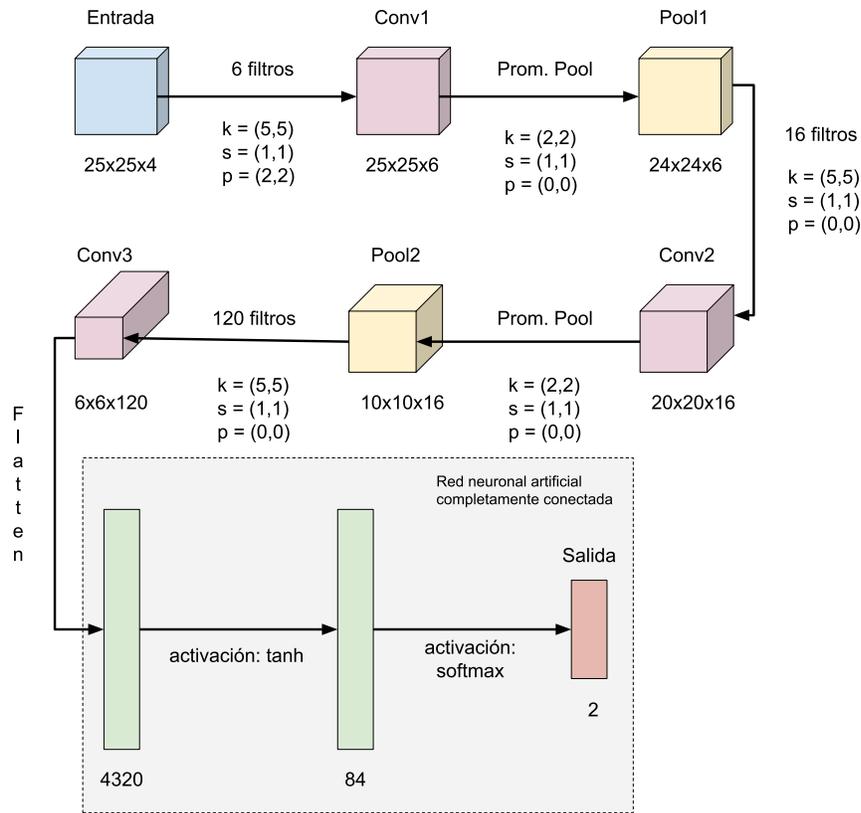


Figura 3.6: Red neuronal convolucional propuesta.

correctamente si existe la presencia del árbol de dzidzilché o no; de igual manera, al comparar las funciones de pérdida podemos observar cuantas épocas son requeridas para que la función de pérdida del modelo pueda alcanzar el valor más bajo. Una vez definidos la tasa de aprendizaje y el número de épocas se procedió a realizar el entrenamiento usando validación cruzada de  $k$  iteraciones con valores para  $k$  desde 5 hasta 10, esto con el propósito de evaluar el modelo con diferentes proporciones del conjunto de entrenamiento (Tabla 3.2).

Tabla 3.2: Distribución del conjunto de entrenamiento para diferentes valores de  $k$ .

$k$	Entrenamiento	Validación
5	10752	2688
6	11200	2240
7	11520	1920
8	11760	1680
9	11946	1493
10	12096	1344

### **3.6. Etapa de evaluación**

Para evaluar la robustez del modelo se comparan los resultados de la función de pérdida promedio de los modelos generados en cada validación cruzada con diferente valor de  $k$ . De igual forma se compara la función de exactitud promedio y se calcula su desviación estándar; con esta información podemos escoger qué validación cruzada ofrece un mejor rendimiento. Luego se seleccionan los  $k$  modelos de la mejor validación cruzada y se calcula la matriz de confusión promedio, partiendo de esta información podemos inferir los porcentajes de exactitud, la especificidad y sensibilidad. Además se pueden comparar las predicciones de los modelos con los valores reales y calcular la curva ROC del modelo y área bajo la curva, los cuales permiten evaluar la sensibilidad y especificidad del modelo a diferentes umbrales de discriminación.

# Capítulo 4

## Resultados

En este capítulo se muestran los resultados obtenidos al seguir la metodología, primeramente se presentan los resultados de la corrección y alineación de las imágenes multiespectrales, posteriormente los resultados de la validación cruzada simple y se analizan sus funciones de pérdida y los parámetros que se escogieron para realizar la validación cruzada de  $k$  iteraciones. Luego se comparan los resultados obtenidos con la validación cruzada de  $k$  iteraciones y se comparan los promedios de la función de pérdida y de exactitud de cada validación cruzada. A continuación se muestran las matriz de confusión promedio de los  $k$  mejores modelos y sus porcentajes de exactitud, sensibilidad y especificidad; más adelante se muestra la curva ROC obtenida y su área bajo la curva. Finalmente, en las discusiones se hace un breve análisis de los resultados obtenidos y las observaciones que se hicieron durante todas las etapas.

### 4.1. Calibración y alineación

Para la corrección de las imágenes se tomaron fotos de un tablero de ajedrez para tener puntos conocidos en el plano, primero se extrajeron los metadatos (Tabla 4.1) de cada imagen. Luego se realizó la corrección de ojo de pescado usando el método descrito en la subsección 2.4.2, en la Figura 4.1 se encuentra el resultado de realizar dicha corrección.

Al continuar con la alineación de las imágenes corregidas, tenemos un problema al usar los parámetros de traslación recomendados por la documentación (Tabla 4.2), pues al aplicar dichos valores, las imágenes salen de la escena respecto a la banda maestra, es por ello que se procedió a calcular los vectores de traslación usando homografías.

Para calcular la homografía que describa la relación entre la imagen maestra (verde) y el res-

Tabla 4.1: Descripción de las etiquetas en los metadatos.

Etiqueta	Descripción	Ejemplo
RigRelatives	Ángulos de rotación $\omega, \phi, \kappa$ en grados.	-0.2081, -0.0274, -0.5007
PrincipalPoint	Punto central $(c_x, c_y)$ de la imagen en milímetros.	2.264813, 1.938798
FisheyeAffineMatrix	Valores de la matriz afín $C, D, E, F$ .	1665.117565771, 0, 0, 1665.117565771
FisheyePolynomial	Valores $p_1, p_2, p_3, p_4$ para la corrección de ojo de pescado.	0, 1, 0.003918014, -0.136955888

Tabla 4.2: Vectores de traslación de acuerdo a la documentación [49].

Banda espectral	Traslación relativa ( $T_{rel}$ en $mm$ )
Verde	[0, 0, 0]
Rojo	[0, -15.0, 0]
Borde rojo	[-15.0, 0, 0]
Infrarrojo	[-15.0, -15.0, 0]

to de imágenes espectrales (rojo, borde rojo e infrarrojo), se usaron 100 fotos de tablero ajedrez, y por cada conjunto de 4 imágenes se halló la matriz de homografía respecto a la banda maestra, por lo que se tienen 100 homografías por cada pareja de bandas (verde-rojo, verde-borde rojo y verde-infrarrojo), luego las homografías se promediaron y se les calculó la desviación estándar, los resultados se presentan en la tabla 4.3.

La Figura 4.2 muestra la posición de los puntos internos del tablero en cada imagen espectral después de realizar la corrección de ojo de pescado no alineadas. En la Figura 4.3 se

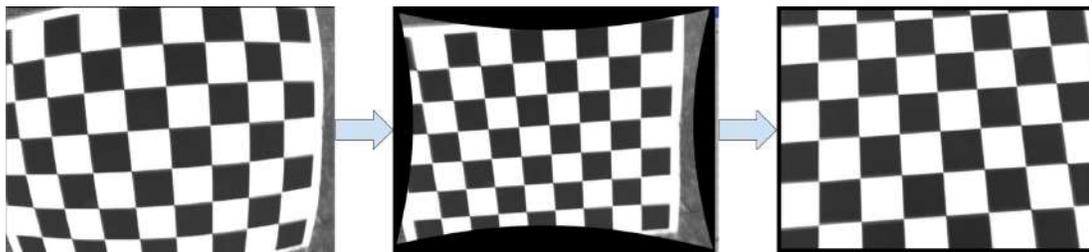


Figura 4.1: Resultado de realizar la corrección de ojo de pescado descrito en la subsección 2.4.2. El resultado final del proceso es la imagen de la derecha, sin embargo la figura central se incluyó para mostrar cómo se deforma el margen de la imagen.

Tabla 4.3: Homografías promedio para la alineación entre bandas.

Banda.	Homografía promedio.	Desviación estándar.
Roja	$\begin{pmatrix} 1.0 & 0.0 & 53.5 \\ 0.0 & 1.0 & -9.3 \\ 0.0 & 0.0 & 1.0 \end{pmatrix}$	$\begin{pmatrix} 0.0 & 0.0 & 0.27 \\ 0.0 & 0.0 & 0.74 \\ 0.0 & 0.0 & 0.0 \end{pmatrix}$
Borde Rojo	$\begin{pmatrix} 1.0 & 0.0 & -9.9 \\ 0.0 & 1.0 & 27.2 \\ 0.0 & 0.0 & 1.0 \end{pmatrix}$	$\begin{pmatrix} 0.0 & 0.0 & 0.89 \\ 0.0 & 0.0 & 0.33 \\ 0.0 & 0.0 & 0.0 \end{pmatrix}$
Infrarrojo	$\begin{pmatrix} 1.0 & 0.0 & 41.8 \\ 0.0 & 1.0 & 15.7 \\ 0.0 & 0.0 & 1.0 \end{pmatrix}$	$\begin{pmatrix} 0.0 & 0.0 & 0.70 \\ 0.0 & 0.0 & 0.56 \\ 0.0 & 0.0 & 0.0 \end{pmatrix}$

presenta la posición de los mismos puntos luego de realizar la alineación usando las matrices de homografías promedio correspondientes de cada banda espectral.

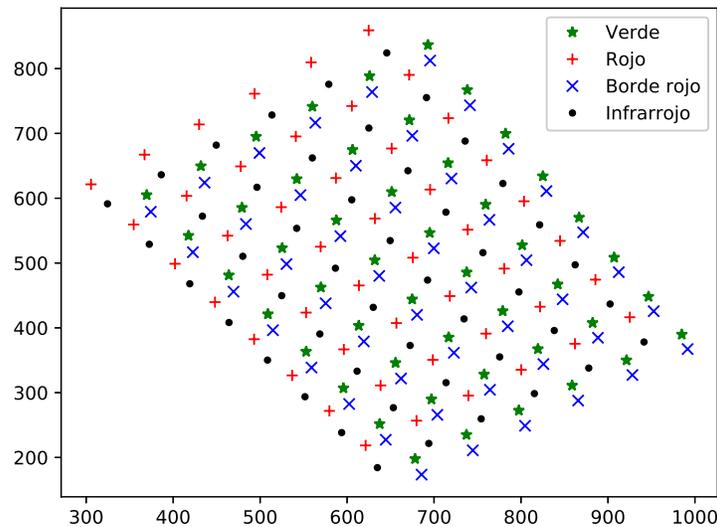


Figura 4.2: Posición de los puntos internos del tablero en cada imagen espectral.

## 4.2. Red neuronal

A continuación se presentan los resultados de los entrenamientos realizados en el modelo propuesto, dichos experimentos se realizaron en una computadora con *Ubuntu 16.04* como sistema operativo, un procesador *Intel (R) Core (TM) i7-6700 CPU @ 340 GHz* y una memoria RAM de 8 GB, además la computadora cuenta con una tarjeta gráfica *NVIDIA GM107* modelo

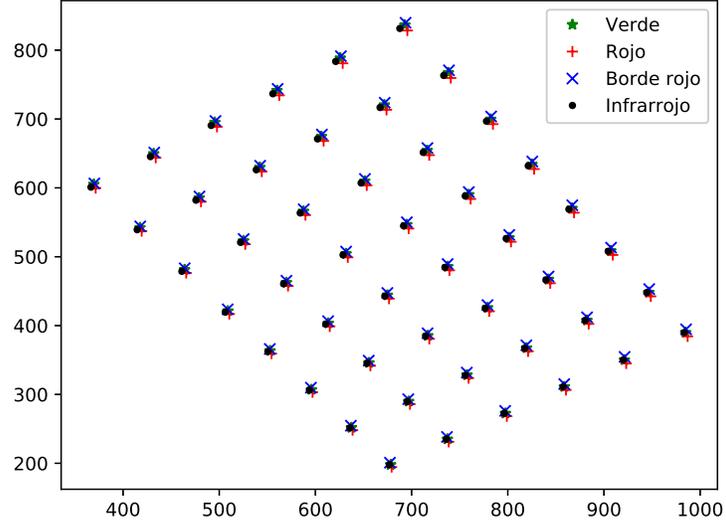


Figura 4.3: Posición de los puntos internos del tablero después de realizar la alineación usando las homografías promedio correspondientes.

*GeForce GTX 745.*

### 4.3. Validación cruzada

Una vez definida la red neuronal se procedió a seleccionar la mejor tasa de aprendizaje para el algoritmo. Para ello se entrenó el modelo con las siguientes tasas de aprendizaje: 0.01, 0.02, 0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09 durante 500 épocas; los resultados obtenidos en la validación se presentan en la Figura 4.4, donde se puede observar que los modelos alcanzan un punto mínimo antes de las 100 épocas, esto se puede apreciar mejor en la Figura 4.5 donde se ha suavizado la función de pérdida con un filtro Gaussiano con  $\sigma = 2$  y una ventana de 11 elementos ( $-10$  a  $10$ ). Pese a que el modelo tiene un mejor resultado cuando la tasa de aprendizaje es 0.01, se ha seleccionado 0.02 como la tasa de aprendizaje para el entrenamiento con validación cruzada de  $k$  iteraciones debido a que tiene un costo mínimo similar al de la tasa 0.01 pero en casi la mitad de épocas.

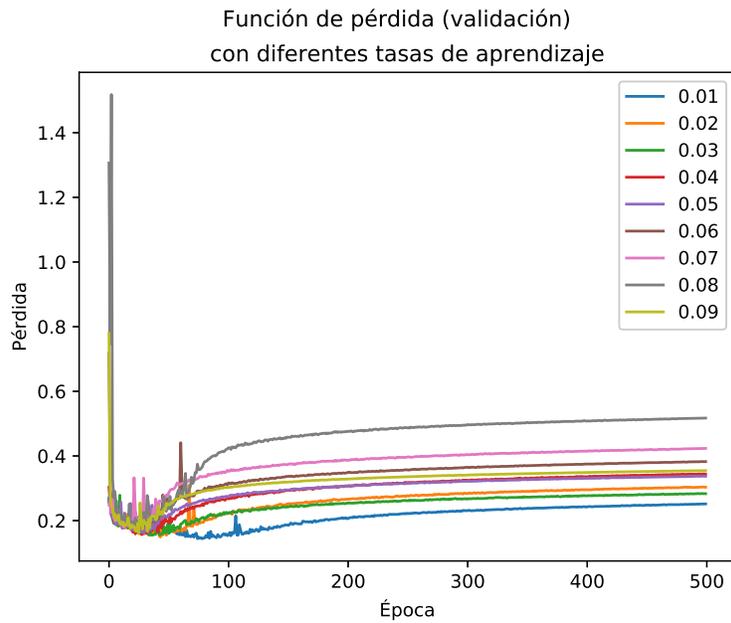


Figura 4.4: Función de pérdida en la validación con diferentes tasas de aprendizaje.

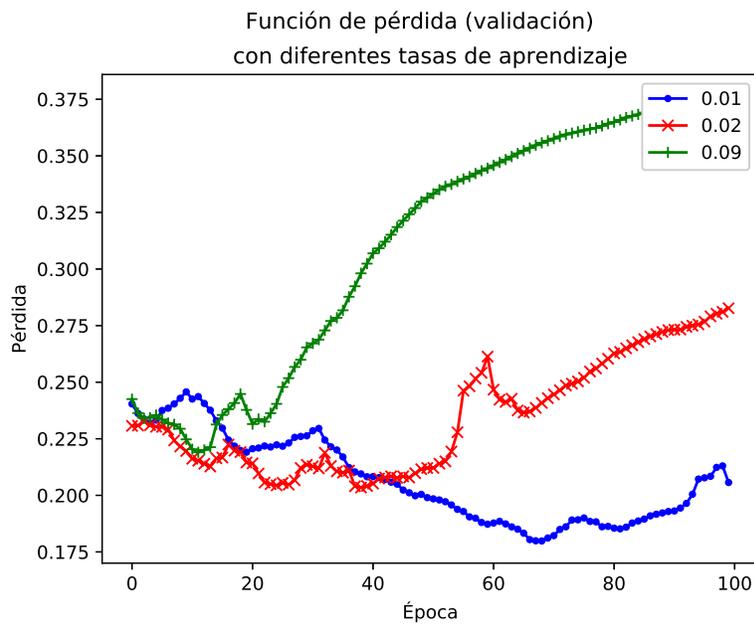


Figura 4.5: Función de pérdida en la validación con diferentes tasas de aprendizaje suavizada con un filtro Gaussiano con  $\sigma = 2$ .

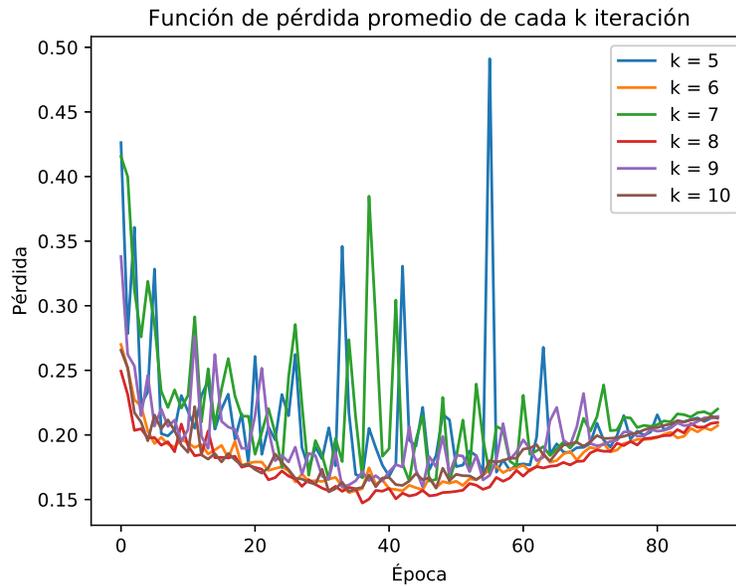


Figura 4.6: Función de pérdida promedio por cada validación cruzada de  $k$  iteraciones.

#### 4.4. Validación cruzada de $k$ iteraciones

De los resultados obtenidos en la validación cruzada simple, se tiene una tasa de aprendizaje de 0.02 ( $lr = 0.02$ ) y se determinó un entrenamiento de 100 épocas; definidos los parámetros el modelo se entrenó usando la validación cruzada con  $k$  iteraciones para  $k = 5, 6, 7, 8, 9, 10$ ; durante el entrenamiento se guardó el modelo con menor pérdida en la etapa de validación. Luego se promediaron las  $k$  funciones de pérdida de cada validación cruzada (Figura 4.6), posteriormente se localizó la época donde ocurre la menor pérdida y se obtuvo la exactitud promedio en ese punto, adicionalmente se calculó la desviación estándar de ambos parámetros. Los resultados de estos indicadores se presentan en la Tabla 4.4, donde se puede observar que los modelos tienen similitudes en su comportamiento, además, la similitud entre las desviaciones estándar del costo indican que la exactitud de predicción del modelo no depende del número de particiones ni del tamaño del conjunto de entrenamiento [60]; sin embargo, para evaluar el modelo se escogió la validación cruzada con  $k = 8$  por tener mayor exactitud.

A continuación se evaluó el conjunto de datos para pruebas con los 8 modelos de la validación cruzada, cuyas predicciones se compararon con las etiquetas reales para generar tablas de contingencia, las cuales se promediaron y registraron en la Tabla 4.5, donde el valor positivo corresponde a una etiqueta o predicción con el valor numérico 1, y negativo a 0. Adicionalmente se calculó la desviación estándar en cada celda para evaluar la variación de los valores en la tabla de contingencia promedio, estas variaciones son presentadas en la Tabla 4.6. Sin embargo,

Tabla 4.4: Resultados de los modelos con menor costo.

Validación cruzada de $k$ iteraciones	5	6	7	8	9	10
Época	36	34	46	36	33	31
Costo promedio	0.167	0.155	0.164	0.147	0.159	0.155
Desviación estándar del costo	0.012	0.01	0.013	0.013	0.02	0.009
Exactitud promedio	93.46 %	93.84 %	93.87 %	94.13 %	93.77 %	93.72 %
Desviación estándar de la exactitud	0.005	0.003	0.006	0.005	0.006	0.006

podemos inferir las variables estadísticas que describan de manera numérica el rendimiento del clasificador.

Tabla 4.5: Tabla de contingencia promedio con los valores absolutos.

		Valor real		
		Positivo	Negativo	Total
Predicción	Positivo	2642.75	237.25	2880
	Negativo	112.00	2768.00	2880
	Total	2754.75	3005.25	5760

Tabla 4.6: Desviación estándar de los valores en la tabla de contingencia.

		Valor real	
		Positivo	Negativo
Predicción	Positivo	24.76	24.76
	Negativo	26.86	26.86

En la Tabla 4.7 se presentan los resultados de calcular la sensibilidad, la especificidad y la precisión promedio de los modelos analizados. Es posible observar que, en promedio, la sensibilidad y especificidad es bastante buena, es decir que el clasificador pudo identificar correctamente en un 95.93 % segmentos de imagen con árbol de dzidzilché y un 92.10 % en segmentos

Tabla 4.7: Valores estadísticos inferidos a partir de la tabla de contingencia.

Estadístico	Valor	Porcentaje
Sensibilidad	0.9593	95.93 %
Especificad	0.9210	92.10 %
Precisión	0.9393	93.93 %

que no tienen presencia del árbol.

Luego, se procedió a calcular la Curva ROC y su área bajo la curva, la cual representa la relación entre la sensibilidad y especificidad a diferentes umbrales de discriminación. Debido a que se tienen dos clases, se generaron dos curvas, las cuales se presentan en la Figura 4.7; a pesar de que visualmente las curvas sugieren que el modelo es bastante bueno el parámetro que determina la capacidad discriminativa es el área bajo la curva (AUC), pues mientras más cercana sea el área a 1 ( $AUC = 1$ ) mejor será el clasificador [22]; por tanto, un área igual 0.98 corresponde a un clasificador muy bueno.

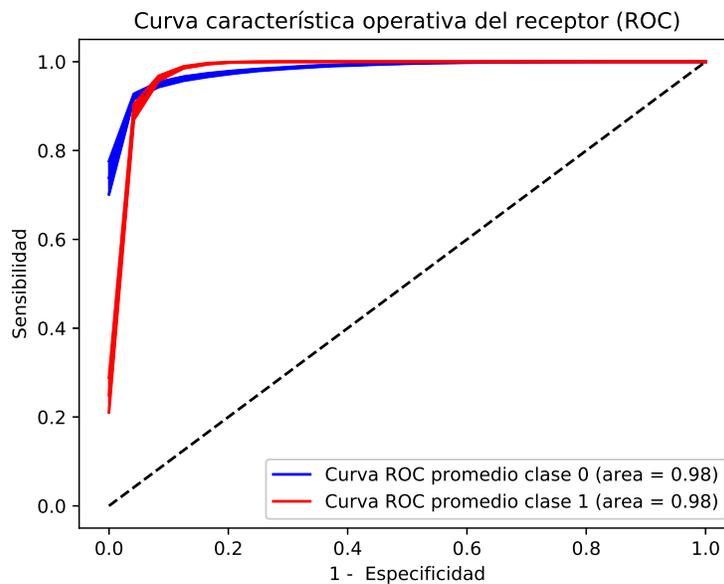


Figura 4.7: Curvas ROC generadas con sus respectivas áreas bajo la curva.

Finalmente, se determinaron las regiones con presencia de dzidzilché la imagen multispectral (de 4 bandas) se segmentando en regiones de  $25 \times 25 \times 4$ , los cuales se evaluaron en los modelos generados por la validación cruzada de 8 iteraciones ( $k = 8$ ), debido a su mejor desempeño; la salida del modelo determinó el color de la región analizada, si la salida del modelo

es 1 entonces la región tiene presencia de dzidzilché y es pintada de blanco, de otro modo, la región es pintada de negro. La Figura 4.8 presenta la salida del clasificador usando una imagen espectral con presencia de dzidzilché con textura de suelo y sombras, donde se puede apreciar que la gran área blanca corresponde al área de dzidzilché, sin embargo, existen falsos positivos donde corresponde a áreas de suelo y sombras; a pesar que también hay falsos negativos estos son pocos y pueden ser filtrados en una etapa de postprocesamiento (Figura 4.8a).

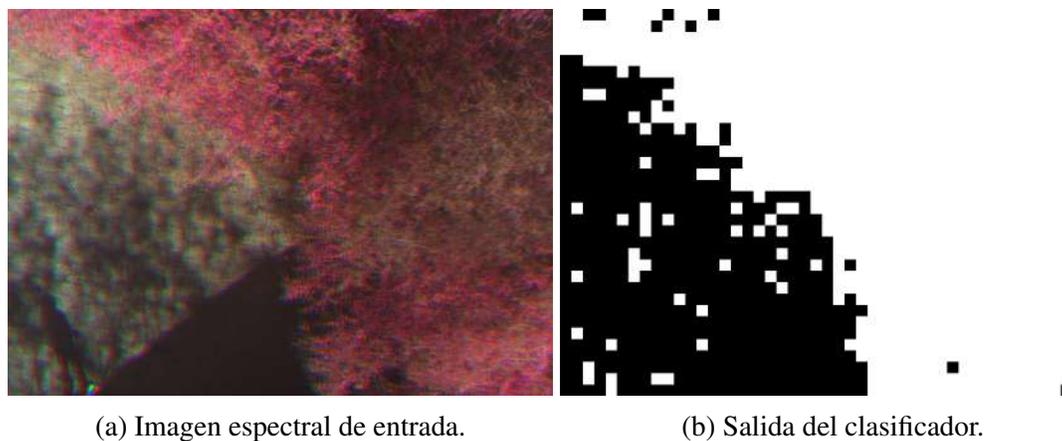


Figura 4.8: Resultados del clasificador usando una imagen espectral. La Subfigura a) tiene combinación color infrarrojo. En la Subfigura b) el área blanca indica presencia de dzidzilché.

Adicionalmente, se probó el clasificador con una imagen multiespectral donde toda el área tiene presencia de dzidzilché (Figura 4.9a); a pesar de que esta combinación espectral muestra diferentes tonalidades debido a las alturas de los árboles y la densidad de las flores en sus ramas la salida del clasificador (Figura 4.9b) demostró ser robusto ante las variaciones presentes en los árboles.

De igual forma, el clasificador fue probado con imágenes multiespectrales con presencia de otras plantas (Figura 4.10 y Figura 4.11). La Figura 4.10a consiste en una imagen multiespectral con presencia de árboles de limón y naranja, y a pesar que su salida (Figura 4.10b) presenta falsos positivos, esta no abarcan un área significativa de los árboles presentes y pueden ser eliminados en una etapa de postprocesamiento. Por otra parte, la Figura 4.11b tiene falsos positivos ocasionados por hierba que crece en el borde donde termina un camino de cemento y comienzan los árboles.

Como se ha mencionado, los falsos positivos y falsos negativos presentes en las máscaras resultantes pueden ser eliminados utilizando operaciones morfológicas. Se utilizó la apertura para eliminar falsos positivos y la cerradura para los falsos negativos. Las Figuras 4.12, 4.13, 4.14

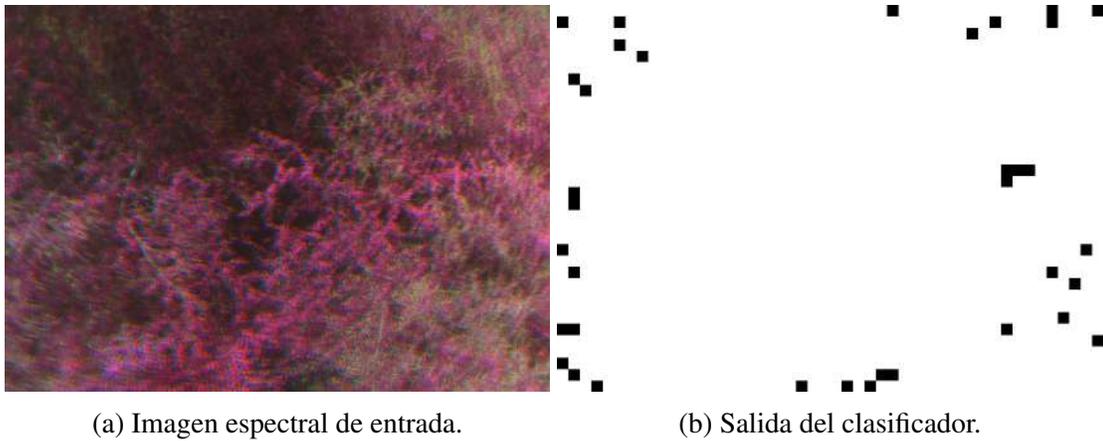


Figura 4.9: Resultados del clasificador usando una imagen espectral de dzidzilché en toda el área. La Subfigura a) tiene combinación color infrarrojo. El área blanca en la Subfigura b) corresponde al área con presencia de dzidzilché.

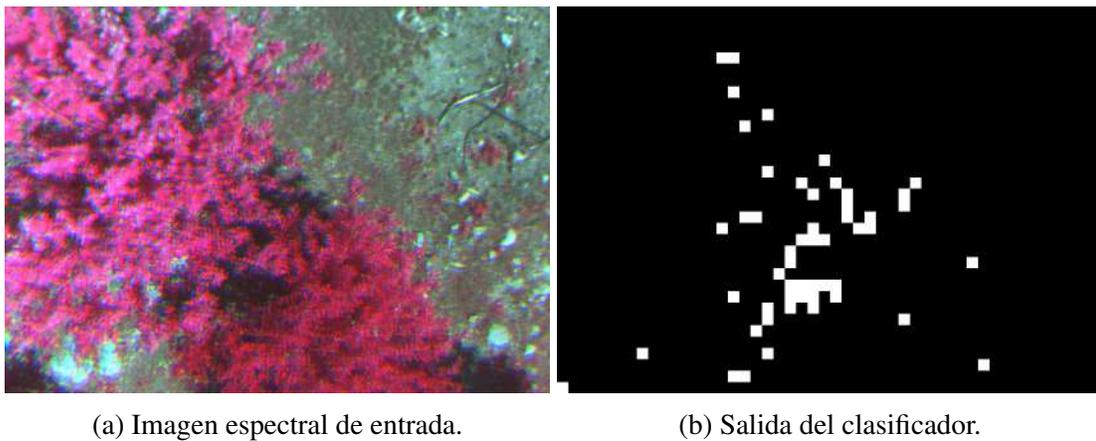
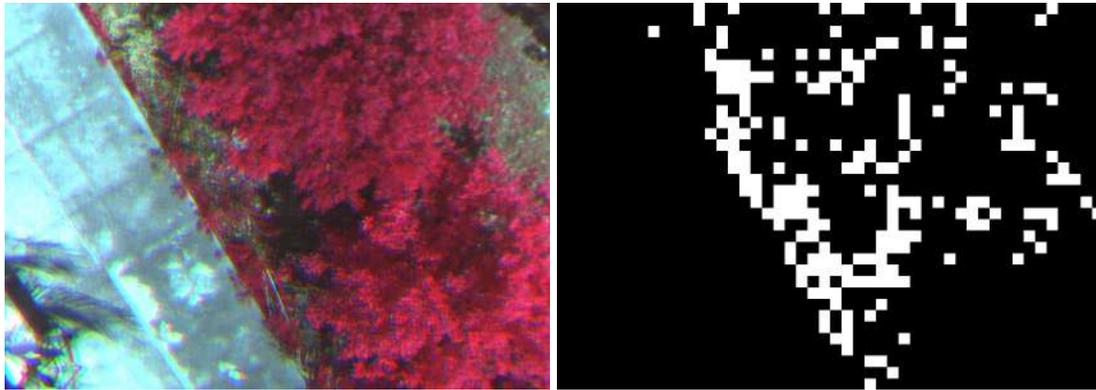


Figura 4.10: Resultados del clasificador usando una imagen espectral de plantas diferentes a dzidzilché. La Subfigura a) tiene combinación color infrarrojo. El área blanca en la Subfigura b) corresponde al área con presencia de dzidzilché.

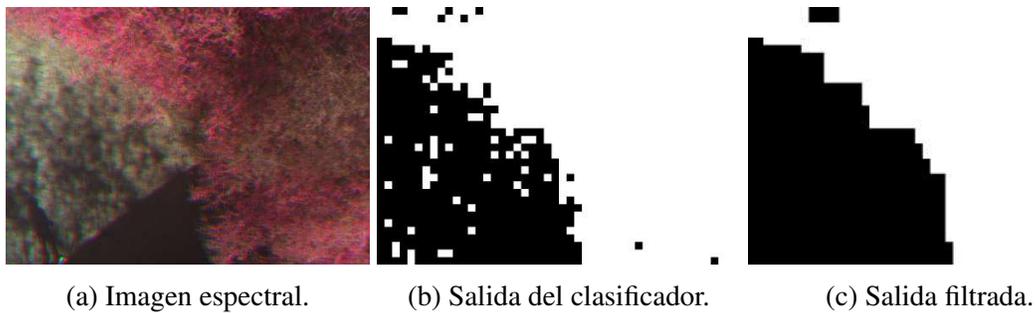
y 4.15 corresponden respectivamente a las Figuras 4.8, 4.9, 4.10 y 4.11, contrastando la salida del clasificador con el resultado de aplicar las operaciones morfológicas de apertura y cerradura para eliminar los falsos positivos y falsos negativos.



(a) Imagen espectral de entrada.

(b) Salida del clasificador.

Figura 4.11: Resultados del clasificador usando una imagen espectral de plantas diferentes a dzidzilché. La Subfigura a) tiene combinación color infrarrojo. El área blanca en la Subfigura b) corresponde al área con presencia de dzidzilché.

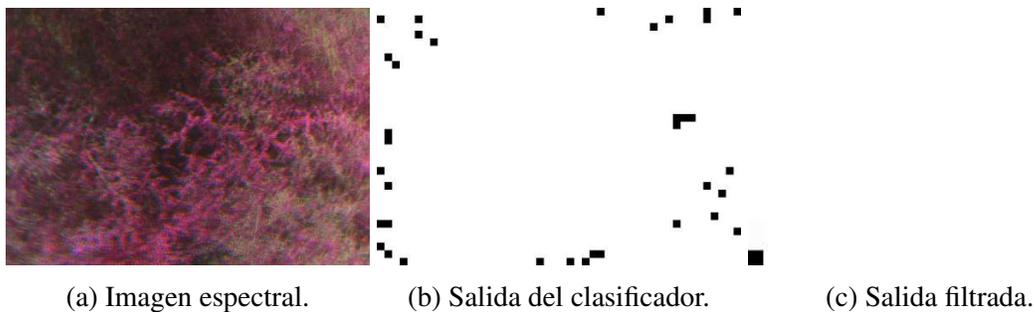


(a) Imagen espectral.

(b) Salida del clasificador.

(c) Salida filtrada.

Figura 4.12: Resultados del clasificador usando una imagen espectral con presencia de dzidzilché, suelo y sombra. La Subfigura a) tiene combinación color infrarrojo. La Subfigura b) indica la salida del clasificador. La Subfigura c) presenta el resultado de realizar apertura y cerradura a la salida b).



(a) Imagen espectral.

(b) Salida del clasificador.

(c) Salida filtrada.

Figura 4.13: Resultados del clasificador usando una imagen espectral de dzidzilché en toda el área. La Subfigura a) tiene combinación color infrarrojo. La Subfigura b) indica la salida del clasificador. La Subfigura c) presenta el resultado de realizar apertura y cerradura a la salida b).

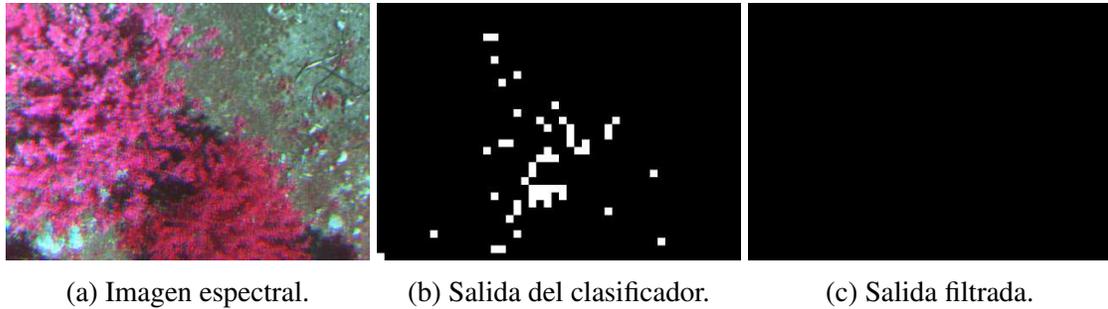


Figura 4.14: Resultados del clasificador usando una imagen espectral de plantas diferentes a dzidzilché. La Subfigura a) tiene combinación color infrarrojo. La Subfigura b) indica la salida del clasificador. La Subfigura c) presenta el resultado de realizar apertura y cerradura a la salida b).

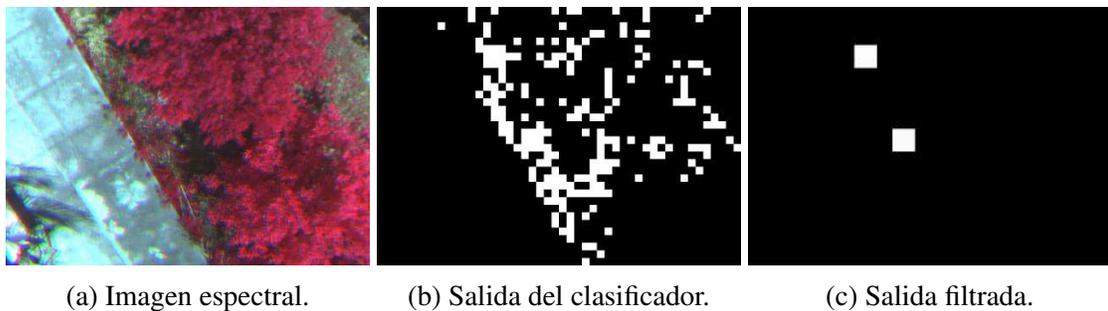


Figura 4.15: Resultados del clasificador usando una imagen espectral de plantas diferentes a dzidzilché y cemento. La Subfigura a) tiene combinación color infrarrojo. La Subfigura b) indica la salida del clasificador. La Subfigura c) presenta el resultado de realizar apertura y cerradura a la salida b).

# Capítulo 5

## Conclusiones

En el proceso de calibración y alineación se confirmó que el uso de los parámetros intrínsecos dados por el fabricante son adecuados para eliminar la distorsión del ojo de pescado, no obstante, para realizar la alineación se requirió de hacer la estimación de matrices de homografías. Durante la etapa para determinar los parámetros de tasa de aprendizaje y número de épocas se puede inferir que el valor mínimo de la función costo se encuentra entre 0.18 y 0.14, aproximadamente, por lo tanto, el rendimiento del modelo entrenado con tasas de aprendizaje más pequeñas a las utilizadas sólo tomaría más tiempo de entrenamiento y el aumento en el rendimiento no sería significativo. Las validaciones cruzadas de  $k$  iteraciones demostraron que el modelo es independiente a la variación de las imágenes contenidas en los diferentes conjuntos de entrenamiento y validación, además, el valor mínimo de la función de pérdida se obtiene entre 30 y 40 épocas de entrenamiento y presentan una precisión similar.

En las pruebas se obtuvo una buena clasificación en discriminar dzidzilché con suelo y sombras, sin embargo, las imágenes con presencia de otras plantas presentan un número considerable de falsos positivos, por lo que, es necesario ampliar el conjunto de datos con elementos vegetativos diferentes al dzidzilché, no obstante, es importante mencionar que el uso de operaciones morfológicas mejoraron la máscara de clasificación al eliminar los falsos positivos y los falsos negativos.

En general, la arquitectura de red neuronal convolucional utilizada (*LeNet-5*), es bastante buena para realizar la clasificación entre árboles de dzidzilché y otros elementos, sin embargo, fue necesario realizar los cálculos correspondientes de las dimensiones adecuadas para los filtros utilizados en las capas de esta red, debido a la dimensión de la entrada requerida; no obstante, la modificación de la red no presentó dificultades y los resultados que se obtuvieron fueron bastantes claros, pues no se presentaron situaciones donde las dos neuronas en la capa

de salida presenten el mismo estado al mismo tiempo. Por lo que, es factible modificar esta red para generar modelos que utilicen una dimensión mayor en las imágenes espectrales de entrada, así como aumentar el número de clases a identificar. Finalmente, para la arquitectura utilizada se recomienda aumentar el número de ejemplos etiquetados como negativos para robustecer el modelo y añadir imágenes de dzidzilché tomadas en diferentes épocas del año, para tener ejemplos positivos con diferente tipo de floración.

## 5.1. Trabajos futuros

En esta investigación se sientan las bases para la caracterización polinífera, lo cual es importante para resolver algunos de los problemas que enfrenta la apicultura en la actualidad, como la colocación estratégica de colmenas, desarrollar estrategias para combatir la deforestación, la prevención y manejo de enfermedades y plagas en la vegetación, evaluar la cantidad de recursos poliníferos disponibles para minimizar el uso de alimentos artificiales, lo que permitiría aumentar las áreas apropiadas para la apicultura y la producción de miel.

Entre los trabajos que pueden mejorar o ampliar la investigación realizada se encuentra mejorar el conjunto de datos agregando imágenes de dzidzilché durante las diferentes épocas del año; otro trabajo relevante es aumentar el número de clases de vegetación polinífera que la red pueda detectar; además con las imágenes que ofrece el sensor espectral, es posible determinar el estado de salud de las plantas poliníferas, así como mantener un monitoreo constante de los recursos poliníferos disponibles. Por otro lado, sería interesante emplear otras técnicas de aprendizaje máquina para resolver la problemática propuesta, por ejemplo, el uso de redes neuronales *autoencoder* como algoritmo no supervisado para determinar los diferentes objetos en las imágenes multispectrales.

# Bibliografía

- [1] Keras: The Python Deep Learning library. URL. <https://keras.io/>.
- [2] TensorFlow Core. URL. <https://www.tensorflow.org/overview>.
- [3] Hossam F Abou-Shaara. Using geographical information system (gis) and satellite remote sensing for understanding the impacts of land cover on apiculture over time. *Int J Rem Sens Appl*, 3(4):171–174, 2013.
- [4] B Alfaro and D Ortiz. Plantas melíferas: Melisopalinología. Parte III. Usos de la Biodiversidad. Uso de la flora y fauna silvestre. *Biodiversidad y Desarrollo Humano en Yucatán. CICY, PPD-FMAM, Conabio, SEDUMA. Mérida, Yuc., México*, pages 346–348, 2010.
- [5] Fernando Araujo-Mondragón and Rosario Redonda-Martínez. Flora melífera de la región centro-este del municipio de pátzcuaro, michoacán, méxico. *Acta Botanica Mexicana*, (126), 2019.
- [6] Ricardo Balam Narvaéz. Una planta común con varias funciones... aunque usted no lo crea! *Desde el herbario*, pages 30–31, 2009.
- [7] M Bäumker and FJ Heimes. New calibration and computing method for direct georeferencing of image and scanner data using the position and angular data of an hybrid inertial navigation system. In *OEEPE Workshop, Integrated Sensor Orientation*, pages 1–17, 2001.
- [8] Henrik Brink, Joseph Richards, and Mark Fetherolf. *Real-World Machine Learning*. Manning Publications Co., Greenwich, CT, USA, 1st edition, 2016.
- [9] Supaporn Bunrit, Ratiporn Chanklan, Supajittree Boonamnuay, Nittaya Kerdprasop, and Kittisak Kerdprasop. Neural network-based analysis of precipitation and remotely sensed data. In *Proceedings of the International MultiConference of Engineers and Computer Scientists*, volume 1, 2016.

- [10] Emmanuel Jimenez Camacho. Medición de distancias por medio de procesamiento de imágenes y triangulación, haciendo uso de cámaras de video. Tesis de maestría, Universidad de las Américas Puebla, 2009.
- [11] Fernando Carvajal Ramírez, Manuel ángel Aguilar Torres, Francisco Agüera Vega, and Fernando José Aguilar Torres. Clasificación de una imagen multispectral de satélite de alta resolución espacial mediante redes neuronales artificiales. URL. <http://cartesianos.com/geodoc/ingegraf2005/GIS1.pdf>.
- [12] Yushi Chen, Zhouhan Lin, Xing Zhao, Gang Wang, and Yanfeng Gu. Deep learning-based classification of hyperspectral data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(6):2094–2107, 2014.
- [13] Yushi Chen, Xing Zhao, and Xiuping Jia. Spectral–spatial classification of hyperspectral data based on deep belief network. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(6):2381–2392, 2015.
- [14] Emilio Chuvieco. *Fundamentos de teledetección espacial*. Ediciones Rialp, 1996.
- [15] Comisión Nacional para el Conocimiento y Uso de la Biodiversidad, CONABIO. Apicultura sostenible. URL. <https://www.biodiversidad.gob.mx/corredor/SPSB/apicultura.html>.
- [16] Claudia Ivette Córdova-Córdova, Elia Ramírez-Arriaga, Enrique Martínez-Hernández, and Juan Manuel Zaldívar-Cruz. Caracterización botánica de miel de abeja (*apis mellifera* l.) de cuatro regiones del estado de tabasco, México, mediante técnicas melisopalinológicas. *Universidad y ciencia*, 29(2):163–178, 2013.
- [17] Hal Daumé III. *A course in machine learning*. Self-Published, 2017. URL. <http://ciml.info/>.
- [18] Secretaría de Educación del Gobierno del Estado de Yucatán, editor. *La miel y las abejas*. Secretaría de Cultura, 2011.
- [19] Diario de Yucatán. Principal productor de miel. URL. <https://www.yucatan.com.mx/merida/principal-productor-de-miel>.
- [20] DigitalGlobe. *The QuickBird sensor*, 2011. URL. <https://www.spaceimagingme.com/downloads/sensors/datasheets/QuickBird-DS-QB-Web.pdf>. Manual técnico.

- [21] El financiero. México, uno de los mayores productores de miel a nivel mundial. URL. <https://www.elfinanciero.com.mx/economia/mexico-uno-de-los-mayores-productores-de-miel-a-nivel-mundial>, 2017.
- [22] Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.
- [23] Raúl Aguirre Gómez. *Los mares mexicanos a través de la percepción remota*, volume 1. Plaza y Valdes, 2001.
- [24] MT Gómez-Casero, F López-Granados, JM Peña-Barragán, M Jurado-Expósito, and L García-Torres. Caracterización espectral multitemporal de cultivos de regadío aplicando análisis discriminante y redes neuronales. *VI Congreso Ibérico de Ciencias Hortícolas*, 2019.
- [25] Carlos Manuel Echazarreta González. Apicultura y producción de miel. URL. <https://www.cicy.mx/documentos/CICY/Sitios/Biodiversidad/pdfs/Cap2/18%20Apicultura%20y%20produccion.pdf>.
- [26] Aurlien Gron. *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O’Reilly Media, Inc., 1st edition, 2017.
- [27] Francisco J Güemes Ricalde, Carlos Echazarreta González, Rogel Villanueva, Juan Manuel Pat Fernández, and Regino Gómez Alvarez. La apicultura en la península de yucatán. actividad de subsistencia en un entorno globalizado. *Revista Mexicana del Caribe*, 8(16), 2003.
- [28] Enrique Hauenstein, Marcos González, Fernando Peña-Cortés, and Andrés Muñoz-Pedrerros. Clasificación y caracterización de la flora y vegetación de los humedales de la costa de Toltén (IX Región, Chile). *Gayana. Botánica*, 59(2):87–100, 2002.
- [29] Geografía e Informática Instituto Nacional de Estadística. Conjunto de datos vectoriales de la carta de uso del suelo y vegetación: escala 1: 250 000. serie iii (continuo nacional), 2005.
- [30] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning: With Applications in R*. Springer Publishing Company, Incorporated, 2014.
- [31] LISA Lab. *Deep Learning Tutorial*. University of Montreal, Canada, 2015.

- [32] Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404, 1990.
- [33] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [34] Yann LeCun, LD Jackel, Leon Bottou, A Brunot, Corinna Cortes, JS Denker, Harris Drucker, I Guyon, UA Muller, Eduard Sackinger, et al. Comparison of learning algorithms for handwritten digit recognition. In *International conference on artificial neural networks*, volume 60, pages 53–60. Perth, Australia, 1995.
- [35] Aldo Leopold. *Game management*. Madison: University of Wisconsin Press, 1987.
- [36] Zhouhan Lin, Yushi Chen, Xing Zhao, and Gang Wang. Spectral-spatial classification of hyperspectral image using autoencoders. In *Information, Communications and Signal Processing (ICICS) 2013 9th International Conference on*, pages 1–5. IEEE, 2013.
- [37] Iván Lizarazo. Clasificación de la cobertura y del uso del suelo urbano usando imágenes de satélite y algoritmos supervisados de inteligencia artificial. *UD y la geomática*, (2):4–18, 2008.
- [38] Rey Machado, R Roche, Odalys Toral, and Eliel Gonzalez Garcia. Metodología para la colecta, conservación y caracterización de especies herbáceas, arbóreas y arbustivas útiles para la ganadería. *Pastos y Forrajes*, 22(3):181–204, 1999.
- [39] Miguel A Magaña Magaña and Carlos E Leyva Morales. Costos y rentabilidad del proceso de producción apícola en México. *Contaduría y administración*, 235:99–119, 2011.
- [40] Konstantinos Makantasis, Konstantinos Karantzalos, Anastasios Doulamis, and Nikolaos Doulamis. Deep supervised learning for hyperspectral data classification through convolutional neural networks. In *Geoscience and Remote Sensing Symposium (IGARSS), 2015 IEEE International*, pages 4959–4962. IEEE, 2015.
- [41] Nour Maris, Shattri Mansor, and H Shafri. Apicultural site zonation using GIS and multi-criteria decision analysis. *Pertanika J. Trop. Agric. Sci*, 31(2):147–162, 2008.
- [42] Jean-François Mas. Un método para combinar datos espectrales e información auxiliar en una red artificial neuronal. *Anais XII Simpósio Brasileiro de Sensoriamento Remoto*, pages 3543–3549, 2005.

- [43] Rolando Renee Badaracco Meza and Joel Rojas Acuña. Clasificación de la salinidad del suelo mediante imágenes de satélite y las redes neuronales artificiales. *ECIPeru: Revista del Encuentro Científico Internacional*, 10(1):4–8, 2013.
- [44] Bibiana Patricia Montoya-Bonilla, Aida Elena Baca-Gamboa, and Blana Lilia Bonilla. Flora melífera y su oferta de recursos en cinco veredas del municipio de piendamó, cauca. *Bioteología en el Sector Agropecuario y Agroindustrial*, 15(spe):20–28, 2017.
- [45] Carlos Alberto Muñoz. *Relación entre la superficie de distintas coberturas vegetales y el Índice de Vegetación Normalizado con el rendimiento melífero de apiarios del centro de la provincia de Buenos Aires*. Tesis doctoral, Universidad de Buenos Aires, 2007.
- [46] Carnegie Mellon University School of Computer Science. Cross validation. URL. <http://www.cs.cmu.edu/~schneide/tut5/node42.html>.
- [47] OpenCV. Basic concepts of the homography explained with code. URL. [https://docs.opencv.org/master/d9/dab/tutorial\\_homography.html](https://docs.opencv.org/master/d9/dab/tutorial_homography.html).
- [48] José María Aceña Palomar. El sistema de comunicación de las abejas. *Didáctica. Lengua y Literatura*, 2:19, 1990.
- [49] Parrot. Application note: How to register images from sequoia single-band cameras. URL. <https://forum.developer.parrot.com/uploads/default/original/2X/7/7b4ae9060b9482975495333c118f22a799fa39db.pdf>.
- [50] Parrot. Parrot Sequoia+. Capta lo invisible y cuida tus cultivos. URL. <https://www.parrot.com/soluciones-business/profesional/parrot-sequoia#parrot-sequoia-details>.
- [51] Santanu Pattanayak. *Pro Deep Learning with TensorFlow. A Mathematical Approach to Advanced Artificial Intelligence in Python*. Apress, 2017.
- [52] Josh Patterson and Adam Gibson. *Deep Learning: A Practitioner's Approach*. O'Reilly, 2017.
- [53] Fernando Paz Pellat and Marcos Casiano. Caracterización fenológica de bosques tropicales caducifolios usando información espectral: experimentos con componentes. *Terra Latinoamericana*, 32(4):259–271, 2014.
- [54] María G Perotti, María C Diéguez, and Fabián G Jara. Estado del conocimiento de humedales del norte patagónico (argentina): aspectos relevantes e importancia para la conservación de la biodiversidad regional. *Revista chilena de historia natural*, 78(4):723–737, 2005.

- [55] Pix4D. How are the internal and external camera parameters defined? URL. <https://support.pix4d.com/hc/en-us/articles/202559089>.
- [56] Cynthia Yajaira Ponce Suarez. Análisis multitemporal de la cobertura vegetal del valle interandino del chota e identificación de zonas de restauración ecológica. Tesis de licenciatura, Universidad Técnica del Norte, 2017.
- [57] David Powers. Evaluation: From precision, recall and fmeasure to roc, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, 2:37–63, 01 2007.
- [58] Elia Quirós Rosado. *Introducción a la Fotogrametría y Cartografía aplicadas a la Ingeniería Civil*. Universidad de Extremadura, 2015.
- [59] José Luis Reyes-Carrillo, José Luis Galarza-Mendoza, Rubí Muñoz-Soto, and Alejandro Moreno-Reséndez. Diagnóstico territorial y espacial de la apicultura en los sistemas agroecológicos de la comarca lagunera. *Revista mexicana de ciencias agrícolas*, 5(2):215–228, 2014.
- [60] Juan D Rodriguez, Aritz Perez, and Jose A Lozano. Sensitivity analysis of k-fold cross validation in prediction error estimation. *IEEE transactions on pattern analysis and machine intelligence*, 32(3):569–575, 2009.
- [61] Rachid Sammouda, Noboru Niki, and Hiromu Nishitani. A comparison of hopfield neural network and boltzmann machine in segmenting mr images of the brain. *IEEE Transactions on Nuclear Science*, 43(6):3361–3369, 1996.
- [62] Rachid Sammouda, Ameer Tourir, Yaser A Reyad, Nuru Adgaba, Ahmed Ai-Ghamdi, and Said Saad Hegazy. Adapting artificial hopfield neural network for agriculture satellite image segmentation. In *2013 International Conference on Computer Applications Technology (ICCAT)*, pages 1–7. IEEE, 2013.
- [63] Gerardo Schneider, Alejandro Javier Hadad, and Alejandra Kemerer. Implementación de un software para el análisis de imágenes aéreas multiespectrales de caña de azúcar. *Ventana Informática*, (28), 2013.
- [64] Luz Marina Silva Arias and Sebastián Restrepo. *Flora apícola: determinación de la oferta floral apícola como mecanismo para optimizar producción, diferenciar productos de la colmena y mejorar la competitividad*. Bogotá, Instituto Humboldt, 2012. URL. <http://repository.humboldt.org.co/bitstream/handle/20.500.11761/32562/469.pdf>.

- [65] Arnol Suarez, Andrés Jiménez, Mauricio Castro Franco, and Angel Cruz-Roa. Clasificación y mapeo automático de coberturas del suelo en imágenes satelitales utilizando redes neuronales convolucionales. *Orinoquia*, 21(1), 2017.
- [66] Akhil Wali. *Clojure for Machine Learning*. Packt Publishing, 2014.