INSTITUTO TECNOLÓGICO DE CIUDAD MADERO

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

DOCTORADO EN CIENCIAS DE LA INGENIERÍA

TESIS

DESARROLLO DE UNA METODOLOGÍA PARA EL RECONOCIMIENTO DE OBJETOS BAJO CONDICIONES DE OCLUSIÓN Y AMBIENTES NO CONTROLADOS

Que para obtener el grado de
Doctora en Ciencias de la Ingeniería

Presenta
M.C.C. Lucía Janeth Hernández González
D09070934
626424

Director de Tesis
Dr. Juan Javier González Barbosa
202134

Co-director de Tesis
Dr. Juan Frausto Solís

Cd. Madero, Tamaulipas Marzo 2023

Instituto Tecnológico de Ciudad Madero
Subdirección Académica
División de Estudios de Posgrado e Investigación

Ciudad Madero, Tamaulipas, 23/noviembre/2022

OFICIO No.: U.150/22
ASUNTO: AUTORIZACIÓN DE IMPRESIÓN DE TESIS

C. LUCIA JANETH HERNÁNDEZ GONZÁLEZ
No. DE CONTROL D09070934
P R E S E N T E

Me es grato comunicarle que después de la revisión realizada por el Jurado designado para su Examen de Grado de Doctorado en Ciencias de la Ingeniería, se acordó autorizar la impresión de su tesis titulada:

"DESARROLLO DE UNA METODOLOGÍA PARA EL RECONOCIMIENTO DE OBJETOS BAJO CONDICIONES DE OCLUSIÓN Y AMBIENTES NO CONTROLADOS"

El Jurado está integrado por los siguientes catedráticos:

PRESIDENTE:	DR.	JUAN JAVIER GONZÁLEZ BARBOSA
SECRETARIO:	DR.	JUAN FRAUSTO SOLÍS
PRIMER VOCAL:	DR.	NELSON RANGEL VALDEZ
SEGUNDO VOCAL:	DR.	ULISES PÁRAMO GARCÍA
TERCER VOCAL:	DR.	PEDRO MARTÍN GARCÍA VITE
DIRECTOR DE TESIS:	DR.	JUAN JAVIER GONZÁLEZ BARBOSA
CO-DIRECTOR:	DR.	JUAN FRAUSTO SOLÍS

Es muy satisfactorio para la División de Estudios de Posgrado e Investigación compartir con usted el logro de esta meta. Espero que continúe con éxito su desarrollo profesional y dedique su experiencia e inteligencia en beneficio de México.

A T E N T A M E N T E
Excelencia en Educación Tecnológica®
"Por mi patria y por mi bien"®

EDUCACIÓN | TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE CIUDAD MADERO
DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

MARCO ANTONIO CORONEL GARCÍA
JEFE DE LA DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

c.c.p.- Archivo
MACG 'LFCS'

Declaración

Declaro y prometo que este documento de tesis es producto de mi trabajo original y que no infringe los derechos de terceros, tales como derechos de publicación, derechos de autor, patentes y similares.

Además, declaro que en las citas textuales que he incluido (las cuales aparecen entre comillas) y en los resúmenes que he realizado de publicaciones ajenas, indico explícitamente los datos de los autores y publicaciones.

Además, en caso de infracción de los derechos de terceros derivados de este documento de tesis, acepto la responsabilidad de la infracción y relevo de ésta a mi director y codirectores de tesis, así como al Instituto Tecnológico de Ciudad Madero y sus autoridades.

Atentamente,

M.C.C. Lucía Janeth Hernández González

Dedicatoria

Dedico esta tesis a DIOS, a mi hermano Luis, a mis hermanas Miriam y Lupe, a mi sobrina Lucero Janeth, quienes me dieron vida, educación, amor, apoyo y consejos. A mi madre Martha Patricia González Valdes, mi heroína de capa larga, quien con su fortaleza, sabiduría y amor incondicional me ha guiado en cada paso de mi vida. Gracias por ser mi luz en momentos oscuros y por creer en mí siempre. Este logro es un triunfo de los dos. Eres mi roca y mi ejemplo para seguir. Estoy orgulloso de llamarte madre.

También quiero dedicar esta tesis a mi pareja José Carlos Soto Monterrubio por ser tan maravilloso en cada aspecto de nuestra relación, por ser tan bondadoso con todas las personas que te rodean, por ser un ejemplo que seguir para mí y para muchos otros. Gracias por estos 13 años de lucha y fortaleza, gracias por amarme y estar a mi lado en mi crecimiento profesional y personal, te amo.

Agradecimientos

Agradezco a mis compañeros de estudio y amigos, quienes sin su ayuda nunca hubiera podido encontrar el ánimo para iniciar esta tesis. A mis maestros el Dr. Juan Frausto y el Dr. Juan Paulo quienes nunca desistieron al enseñarme, aun sin importar mis incontables errores y mis altibajos en el estudio. A mi querida Alma Mater y a todas las personas que la conforman les agradezco de todo corazón. No podría haber llegado hasta aquí sin su apoyo.

Igualmente, quiero dar un enorme agradecimiento al Dr. Juan Javier González Barbosa quien considero mi mentor, ya que ha sido el profesor con el que más tiempo he colaborado iniciando desde mi carrera en la ingeniería en el año 2012 y culminando en este doctorado, muchas gracias por todo el aprendizaje humano y académico que me ha brindado a lo largo de estos 12 años, deseando de corazón todo lo bueno de este mundo y algún día volver a coincidir ya no como profesor y estudiante, si no como colegas.

A los sinodales quienes estudiaron mi tesis y la aprobaron. A todos ellos se los agradezco desde el fondo de mi alma.

Al Consejo Nacional de Ciencia y Tecnología por el apoyo brindado a través de la beca *Beca Nacional (Tradicional) 2018-2*, que alcanzó para cubrir mis alimentos y gastos adicionales.

Al Tecnológico Nacional de México/Instituto Tecnológico de Ciudad Madero.

Para todos ellos es esta dedicatoria de tesis, pues es a ellos a quienes se las debo mi propósito, mi valía, mi amor y mi aprecio por su apoyo incondicional. Muchas gracias a todos.

Desarrollo de una metodología para el reconocimiento de objetos bajo condiciones de oclusión y ambientes no controlados

Lucía Janeth Hernández González

Resumen

En este trabajo se propone una metodología llamada PSEV-BF para el reconocimiento de objetos bajo condiciones de oclusión y ambientes no controlados. La metodología PSEV-BF incluye dos nuevas fases en comparación con las metodologías tradicionales de visión por computadora: presegmentación y mejora de variables. La presegmentación se realiza utilizando la tercera versión de YOLO (You Only Look Once), una arquitectura de red neuronal convolucional (CNN) diseñada para la detección de objetos. La mejora de variables, se propone el algoritmo de recocido simulado (SA) como selector de las variables relevantes. Asi mismo, se incorpora la técnica de superpixel en la etapa de extracción de características con una ventana de 15×15 píxeles. Para probar la metodología PSEV-BF, se utilizó el repositorio Commons Object in Context (COCO) con imágenes que muestran a los objetos (pajaros) en entornos no controlados. Por último, se utiliza la métrica APIoU (Average Precision Intersection over Union) como referencia de evaluación para comparar nuestra metodología con configuraciones estándar. Los resultados muestran que la metodología PSEV-BF tiene mejor rendimiento en todas las pruebas.

Development of a methodology for object recognition under conditions of occlusion and uncontrolled environments.

Lucía Janeth Hernández González

Abstract

In this paper, a methodology called PSEV-BF is proposed for object recognition under occlusion conditions and uncontrolled environments. The PSEV-BF methodology includes two new phases compared to traditional computer vision methodologies: pre-segmentation and variable enhancement. Pre-segmentation is performed using the third version of YOLO (You Only Look Once), a convolutional neural network (CNN) architecture designed for object detection. For variable enhancement, the simulated annealing (SA) algorithm is proposed as a selector of the relevant variables. Likewise, the superpixel technique is incorporated in the feature extraction stage with a 15×15 pixel window. To test the PSEV-BF methodology, the Commons Object in Context (COCO) repository was used with images showing objects (birds) in uncontrolled environments. Finally, the APIoU (Average Precision Intersection over Union) metric is used as an evaluation benchmark to compare our methodology with standard configurations. The results show that the PSEV-BF methodology performs better in all tests.

Índice general

Índice de Tablas

Índice de Figuras

CAPÍTULO 1

Introducción

La visión artificial o por computadora, siendo una rama de la inteligencia artificial, pretende mediante el uso de sensores (cámaras) simular las actividades que el ojo humano realiza tales como el reconocimiento, seguimiento y reconstrucción de objetos. Estas actividades también son conocidas como tareas de visión por computadora. Algunos autores como Pajarez [5] mencionan que la visión por computadora "consiste en la deducción automática de la estructura y propiedades de un mundo tridimensional, posiblemente dinámico, a partir de una o varias imágenes bidimensionales de ese mundo".

El reconocimiento de objetos es una de las actividades más estudiadas en los últimos años, por ejemplo se ha realizado reconocimiento de objetos en plantas [6–8], en personas [9, 10]. Aunque, en estos trabajos el objeto de estudio y las técnicas son diferentes, la metodología aplicada es similar. Una metodología de visión por computadora generalmente esta compuesta por los siguientes procesos: recopilación de datos, pre-procesamiento, segmentación, extracción de características, selección de

variables relevantes, clasificación y evaluación. Una metodología de visión por computadora debe ser lo más robusta posible debido a diversos problemas que dificultan localizar, segmentar o determinar el objeto de estudio, tales problemas son los entornos no controlados y la oclusión de los objetos.

Desafortunadamente, al momento de inspeccionar un objeto en una imagen se sitúan diversos problemas que evitan determinar el tipo de objeto, los cuales son: un entorno no controlado u oclusión.

Un entorno no controlado es la falta de un protocolo que describa las condiciones de los parámetros utilizados para capturar imágenes. Dichos parámetros incluyen condiciones de iluminación, posición, orientación y distancia de la cámara al objeto, número de objetos por imagen y clasificación de objetos. Finalmente, las oclusiones ocurren cuando los objetos de interés están parcialmente ocluidos por objetos presentes en la imagen; los objetos ocluidos pueden o no ser del mismo tipo que el objeto de interés. El preprocesamiento mediante técnicas de filtrado ayuda a mejorar la calidad de la imagen y mitigar algunos de los problemas descritos. La ventaja de reducir los problemas de una imagen es que reduce la complejidad de extraer información significativa de la imagen. En los trabajos de [6, 11–14] se observan imágenes bajo un entorno controlado y sin presencia de oclusión.

La presencia de oclusión y ambiente no controlado es un problema principal en la segmentación [15]. Para esto, se tienen que catalogar los pixeles de la imagen para determinar los diferentes elementos en ella. Por lo anterior, la segmentación puede proporcionar segmentos etiquetados, ofreciendo una asistencia guiada al proceso de extracción de características sobre qué tipo de objetos se realizará la extracción.

Por otro lado, Machine Learning (ML) y Deep Learning (DL) son un conjunto de técnicas para el aprendizaje de patrones. El aprendizaje puede ser supervisado y no supervisado. El supervisado entrena con un conjunto de variables de entrada a los que se le conoce su respectiva salida para encontrar un modelo que lo genere, este tipo de aprendizaje se aplica a problemas de clasificación y regresión. Mientras que el aprendizaje no supervisado se enfoca en aprender la estructura de las variables de entrada para buscar similitudes entre ellas y poder aglomerarlas, dando lugar a problemas de agrupamiento, reducción de dimensionalidad, entre otros. Las variables de entrada son conocidas

como atributos o características que describen un patrón.

Para aplicar técnicas de aprendizaje supervisado y no supervisado en la segmentación es necesario extraer características que describan de manera única a los elementos de una imagen. Las técnicas de reducción de dimensionalidad, así como metaheurísticas, permiten seleccionar las características que mejor describan a los objetos. A menudo, Los atributos de color, textura y forma son usados.

Por lo anterior se implementa una metodología para la segmentación y reconocimiento de objetos del tipo aves bajo condiciones de oclusión y ambientes complejos incorporando dos nuevas fases a una metodología general de visión: pre-segmentación y selector de características relevantes. Pre-segmentación está compuesta por la arquitectura de YOLOV3 [16]. Mientras que Recocido Simulado [17] es la metaheurística elegida como selector de variables relevantes.

1.1. PLANTEAMIENTO DEL PROBLEMA

Los impedimentos para realizar un reconocimiento a partir de una imagen son la oclusión, artefactos ambientales y el ambiente complejo. La oclusión se refiere a los obstáculos que no permiten observar en su totalidad al objeto de estudio, dichos obstáculos pueden ser incluso de la misma naturaleza del objeto de estudio. Los artefactos ambientales se refieren a elementos que no permiten capturar los objetos con claridad, como el humo, niebla, lluvia, entre otros elementos. La adquisición de imágenes puede estar sujeta tanto a ambientes controlados como a no controlados.

En la actualidad, el reconocimiento de objetos es un problema ampliamente estudiado. Existen trabajos donde reconocen objetos bajo condiciones de oclusión [18] y ambientes complejos [19].

1.2. JUSTIFICACIÓN

El proceso de reconocimiento de algún objeto a partir de una imagen o secuencia de ellas es un problema de vanguardia aun sin resolver. Las aplicaciones de reconocimiento en personas, animales u objetos inanimados son infinitas. Algunos beneficios que se obtendrían si se lograra una metodología con alta eficiencia para reconocer objetos, serían:

- Reconocimiento de plantas o frutos enfermos en un cultivo de una manera rápida y efectiva, lo cual evitaría la propagación de la enfermedad.
- Reconocimiento de personas lesionadas en zonas inaccesibles, esto aumentaría las probabilidades de vida de la misma. Así mismo se podría reconocer a criminales en una zona aglomerada.
- Reconocimiento de armas ayudaría a disminuir potencialmente el peligro para las personas.
- Reconocimiento de coches en carreteras ayudaría a disminuir los accidentes viales.
- Reconocimiento de instrumentos médicos u órganos, ayudaría a mejorar la asistencia en operaciones.

1.3. HIPOTESIS DE LA TESIS

H_0 : La metodología propuesta para el reconocimiento de objetos del tipo ave con presencia de oclusión y ambiente no controlado no supera el 40 % de precisión.

H_1 :La metodología propuesta para el reconocimiento de objetos del tipo ave con presencia de oclusión y ambiente no controlado supera el 40 % de precisión.

1.4. OBJETIVOS DE LA TESIS

1.4.1. Objetivo general

Desarrollar una metodología de visión computacional para el reconocimiento de objetos del tipo ave con presencia de oclusión en ambientes no controlados.

1.4.2. Objetivos específicos

Los específicos de la tesis son los siguientes:

- Construir la base de datos de conocimiento para el proceso de reconocimiento de objetos.
- Implementar los filtros y transformaciones para mejorar el procesamiento de las imágenes.
- Implementar métodos de segmentación para seleccionar el área de interes.
- Establecer las características relevantes del objeto de interés para su posterior evaluación.
- Evaluar los modelos de clasificación con las características establecidas para obtener el mejor modelo.
- Probar la metodología bajo condiciones de oclusión en ambientes no controlados.

1.5. ORGANIZACIÓN DE LA TESIS

En esta propuesta de doctorado se abordan conceptos, técnicas y métodos básicos para el procesamiento de imágenes tales como filtrado, transformaciones, suavizado, entre otros. La estructura de la tesis

esta compuesta por 6 capítulos. El capítulo 2 esta compuesto por los conceptos básicos entorno a este trabajo de tesis. El capítulo 3 se localiza el estado del arte relacionado con el reconocimiento de objetos y algoritmos de selección de características. La metodología propuesta en este trabajo de tesis esta descrita en el capítulo 4. El análisis y resultados obtenidos bajo las diferentes experimentaciones realizadas se muestran en el capítulo 5. Finalmente, en el capítulo 6 se muestran las conclusiones, contribuciones y trabajos futuros.

CAPÍTULO 2

Marco teórico

El siguiente capítulo está compuesto por teoría a cerca de la representación y las técnicas de visión por computadora aplicada a una imagen. Primeramente, en la Sección 2.1 se abordarán los temas relacionados con la composición y extracción de información a una imagen digital, tales como: la representación de una imagen digital, pre-procesamiento y extracción de características.

Como segundo, la Sección 2.2 menciona algunas técnicas para disminuir la dimensionalidad de las características o variables seleccionadas, en concreto: Análisis de Componentes principales (Principal Components Analysis - PCA) y metaheurísticas como Recocido Simulado (Simulated Annealing - SA) y Criterio de Aceptación (Threshold Accepting - TA).

Finalmente, la Sección 2.3 describe la arquitectura de Aprendizaje Profundo (DL) You Only Look Ones (YOLO) V3 utilizada para el proceso de pre-segmentación. Así mismo, se describen aspectos básicos de composición de las arquitecturas de DL.

2.1. VISIÓN POR COMPUTADORA

Visión por computadora es la ciencia que tiene como principal objetivo entender el mundo real mediante imágenes en 2D aplicando técnicas de procesamiento de imágenes, Machine Learning y Deep Learning, entre otros. En otras palabras, visión por computadora intenta desarrollar métodos para replicar el sistema visual del ser humano.

Sin embargo, la utilización de imágenes para entrenar modelos de reconocimiento es una tarea difícil. En una imagen existen diferentes factores a redimir con el fin de destacar aquellos elementos u objetos que sean de interés; tales factores pueden ser la variación de la luz, la posición de los objetos, lo borrosa o difusa que se encuentre una imagen, elementos inesperados al momento de la captura, entre otros. Una manera de suprimir estos percances es aplicando técnicas de procesamiento de imágenes o transformaciones.

2.1.1. Representación de una imagen digital

Una imagen puede ser representada como una matriz $I(M,N)$ donde M,N denota su tamaño (resolución), como se muestra a continuación:

$$I = \begin{pmatrix} I(1,1) & I(1,2) & I(1,3) & \cdots & I(1,M) \\ I(2,1) & I(2,2) & I(2,3) & \cdots & I(2,M) \\ I(3,1) & I(3,2) & I(3,3) & \cdots & I(3,M) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ I(N,1) & I(N,2) & I(N,3) & \cdots & I(N,M) \end{pmatrix} \tag{2.1}$$

Así mismo, una imagen puede ser representada como una función $f(x,y)$ donde x,y denotan las locaciones espaciales conocidas como píxeles [20]. Los píxeles son los puntos de color en una imagen,

dicho color se representa como la combinación de diferentes tonos. Un tono es representado por una matriz y sus tonalidades se representan en un rango de valores del 0 al 255, siendo así que una imagen a color este compuesta por tres matrices, conocidas como canales de color.

El número de canales y rango de los valores dependerá del modelo de color elegido. Por ejemplo, la Figura 2.1-a) es una matriz en escala de grises ; mientras que en la Figura 2.1-b) se muestra una imagen donde cada color es representado por una matriz. En ambos modelos el rango de color es del 0 al 9.

$$I(x,y) = \begin{pmatrix} 2 & 5 & 2 & 2 \\ 8 & 0 & 9 & 1 \\ 2 & 4 & 2 & 4 \\ 6 & 3 & 4 & 4 \end{pmatrix}$$

a)

$$\mathrm{R}(x,y) = \begin{pmatrix} 3 & 5 & 1 & 2 \\ 9 & 5 & 5 & 1 \\ 2 & 3 & 2 & 4 \\ 6 & 8 & 8 & 0 \end{pmatrix} \quad G(x,y) = \begin{pmatrix} 2 & 5 & 2 & 2 \\ 8 & 7 & 5 & 1 \\ 2 & 6 & 2 & 5 \\ 6 & 3 & 3 & 4 \end{pmatrix} \quad B(x,y) = \begin{pmatrix} 2 & 5 & 2 & 2 \\ 6 & 2 & 8 & 1 \\ 4 & 5 & 5 & 6 \\ 6 & 4 & 8 & 4 \end{pmatrix}$$

b)

Figura 2.1: *Representación de una imagen digital: a) Una matriz con niveles de intensidad en gris; b) tres matrices con niveles de intensidad en color rojo (R), verde (G) y azul (B).*

A partir de esta representación matricial en las imágenes se puede cuantificar características como la resolución espacial y los niveles de intensidad. En la Figura 2.2 se puede apreciar una imagen $I(x,y)$ con las propiedades de resolución y magnitud. La imagen de la izquierda tiene un tamaño o resolución de 480 pixeles en la coordenadas de las x y 480 pixeles en la coordenadas de las y. Por otro lado, la imagen de la derecha tiene una magnitud de 2 niveles de color, en este caso son el blanco y negro.

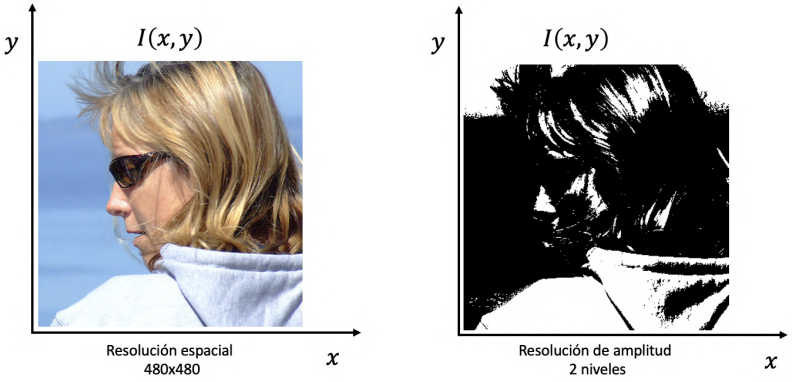

Figura 2.2: *Resolución y magnitud de una imagen* $I(x,y)$

La resolución hace referencia al número de pixeles que contiene una imagen dando lugar al largo y ancho de la misma. Los niveles de intensidad o magnitud es el número de niveles de intensidad que maneja una imagen, la cual puede presentarse en escala de grises o a color, siendo RGB y CMYK ejemplos de modelos de color. El rango de la magnitud puede variar dependiendo del modelo de color elegido.

2.1.2. Pre-procesamiento

Las técnicas de pre-procesamiento son métodos de filtrado para suavizamiento y realzado de las imágenes, dichos métodos pueden presentase bajo dos tipos de dominio en los valores en los pixeles de una imagen: el dominio espacial o el dominio de la frecuencia. El dominio espacial se destaca por realizar operaciones directamente a los píxeles de una imagen. El dominio de la frecuencia trata una imagen como una función a transformar mediante técnicas como la transformada de Fourier. En este trabajo de tesis, los métodos utilizados son la ecualización del histograma y la media gaussiana que son filtros del dominio espacial y de la frecuencia, respectivamente.

Ecualización del histograma

Las operaciones de realce o aumento de contraste se emplean para cambiar los valores de nitidez en las imágenes, a causa de la pobre distinción de todos los elementos por falta de iluminación uniforme en la escena [21]. Las técnicas que se aplican para el realce a menudo están relacionadas con las propiedades del histograma obtenido de la imagen. Dada una imagen $I(x,y)$ cuyos píxeles se encuentren definidos en escala de grises con valores entre 0 y 255, se llama histograma de la imagen al gráfico que se obtiene de la representación de la ocurrencia de cada uno de los valores, Figura 2.3.

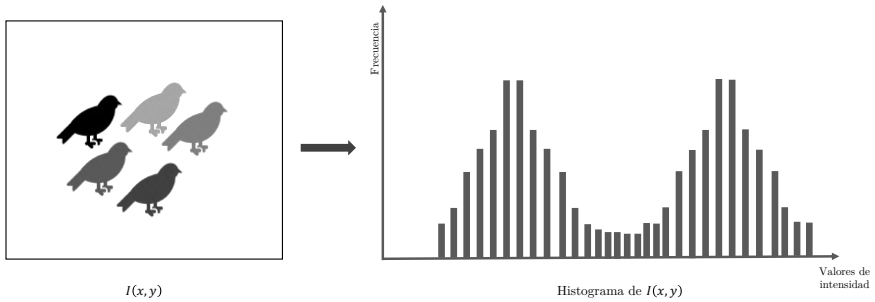

Figura 2.3: *Histograma de una imagen $I(x,y)$ en escala de grises.*

Los histogramas muestran la cantidad de píxeles de una imagen en función de su nivel de intensidad, Ecuación 2.2.

$$P(g) = \frac{N(g)}{M} \tag{2.2}$$

Donde $N(g)$ representa el número de píxeles en un nivel de intensidad g y M es el número de píxeles en la imagen $I(x,y)$ y $P(g)$ es la probabilidad de ocurrencia de una determinado nivel g [21]. Empleando las propiedades de los histogramas se pueden realizar expansiones o reducciones en la ocurrencia del nivel de gris, tal es el caso de la ecualización del histograma.

La ecualización del histograma de una imagen transforma los valores de intensidad a una distribución uniforme, es decir, que cada nivel de gris tenga la misma probabilidad de aparecer en la imagen, lo cual permite visualmente obtener una imagen con una mejor apreciación a los detalles de los elementos presentes en ella.

La ecualización del histograma consiste en, dada una imagen $I(x,y)$ con una resolución de $m \times n$, con n_k pixeles por cada r_k niveles en la escala de grises, se realiza una transformación sobre el nivel de intensidad de la imagen de la siguiente manera:

$$s_k = T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) = \frac{L-1}{MN}\sum_{j=0}^{k} n_j \tag{2.3}$$

donde L es el valor de intensidad máximo en la escala de grises. A menudo, mejorar la calidad de la imagen no es suficiente para lograr apreciar los detalles de manera completa, para lo cual se recurre a utilizar filtros que transforman a una imagen $I(x,y)$ mediante operaciones de convolución.

Media Gaussiana

Una manera de aumentar la calidad en una imagen es aplicando filtros de suavizamiento. Esta técnica se realiza mediante operaciones punto a punto entre un conjunto de pixeles (vecindad, ventana o máscara) y un núcleo (kernel) de convolución.

Una operación de vecindad transforma un valor de píxel $p(x,y)$ de una imagen $I(x,y)$ tomando en cuenta los valores de sus vecinos. En la Figura 2.4-a), se aprecia un pixel central $p(x,y)$ con un valor de 40 y sus 8 pixeles adyacentes en una imagen $I(x,y)$ con una resolución de 5×5 pixeles. La selección de un nuevo pixel central $p(x,y)$ se realiza mediante un desplazamiento de izquierda a derecha y de arriba a abajo, como se muestra en la Figura 2.4 en los incisos del $a) - d)$.

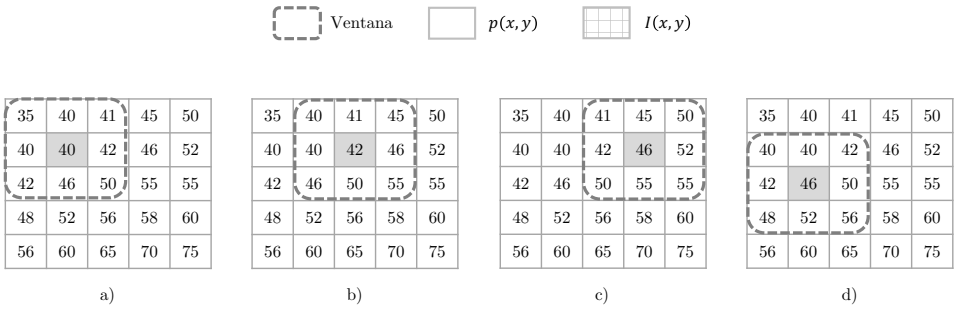

Figura 2.4: *Vecindad con un tamaño de 8 con respecto a un píxel $p(x,y)$ en una imagen $I(x,y)$*

El número de ventanas depende de la resolución de la imagen, el tamaño de la ventana, el número de pixeles a desplazar de izquierda a derecha y de arriba a abajo. Una imagen puede tener multiples ventanas $V = (2N+1) \times (2N+1)$ con $2N+1 < m,n$. Cada una de estas ventanas será sometida a un proceso de convolución.

El núcleo o kernel de convolución es una matriz de coeficientes o una función que permiten realizar una transformación a los valores definidos en una ventana para modificar un pixel central $p(x,y)$.

La media Gaussiana es una función que permite suavizar los niveles de intensidad de una imagen, logrando eliminar datos. Matemáticamente, aplicar un suavizado gaussiano a una imagen es lo mismo que convolucionar la imagen con una función gaussiana. La formulación matemática de la función gaussiana para dos dimensiones se presenta en la Ecuación 2.4.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{2.4}$$

donde x es la distancia desde el origen en el eje horizontal, y es la distancia desde el origen en el eje vertical y σ es la desviación estándar de la distribución gaussiana. Los resultados de aplicar este tipo de técnicas permiten a una imagen eliminar niveles de intensidad fuera del rango distribuido de toda la imagen, un ejemplo se observa en la Figura 2.5.

Figura 2.5: *Suavizado de una imagen para eliminar ruido tipo sal y pimienta.*

El kernel modificara un determinado pixel mediante un proceso llamado convolución. En este proceso la imagen sufre de modificaciones en los niveles de intensidad. Dada un matriz $I_{m,n}$ y un núcleo de convolución $h_{(2N+1)\times(2N+1)}$ donde $2N+1 < m,n$ se obtiene una nueva matriz $D = A \times h$ que se define mediante la Ecuación 2.5:

$$D_{ij} = \frac{1}{h} \sum_{r=1}^{2N+1} \sum_{s=1}^{2N+1} a_{i-N+r-1,i-N+r-1} hr,s \tag{2.5}$$

El filtro gaussiano permite suavizar una imagen con ruido *sal y pimienta*. Este tipo de ruido, es uno de los mas comunes al momento de realizar la captura de la imagen ya sea por las condiciones atmosféricas, defectos del dispositivo de captura o distorsión al momento de transferirlo a un equipo de computo.

2.1.3. Extracción de características

Para poder realizar el reconocimiento de un objeto se requiere entrenar al sistema de visión con parámetros que caractericen de manera única al objeto, con el propósito de obtener un porcentaje aceptable de eficacia al momento de realizar el proceso de clasificación.

Las características de un objeto vienen representadas mediante un vector del cual se constituye por un conjunto de variables que definen la esencia del objeto. Estas variables son extraídas mediante una serie de filtros y transformaciones.

Algunos canales de modelos de color, los contornos y formas, así como propiedades de textura y los niveles de intensidad, que se adquieren de una imagen enfocado a un objeto en particular, son solo algunos parámetros que podrían constituir al vector de características. Los parámetros referentes a color y textura son a menudo los que se emplean para caracterizar un objeto.

Modelos de espacio de color

Como se ha mencionado anteriormente, el color es un poderoso descriptor el cual a sido ampliamente utilizado como atributos para definir las características del objeto de estudio. La extracción de información, mediante los valores de los niveles de intensidad, se puede realizar extrayendo valores de medida de tendencia central tales como media, desviación estándar, rango, entre otras [6].

La variedad de modelos de color permite tener una perspectiva diferente sobre una imagen. A menudo no todos los canales de un modelo de color representan información relevante. Por lo anterior, la conversión de un modelo a otro es una técnica que a menudo se emplea para obtener una mejor caracterización del color.

En este trabajo se describe el proceso de conversión de color RGB a los modelos de color CMYK, LAB, XYZ y HSI. Los cuales han sido seleccionados para la caracterización del color en base al estado del arte.

En la Ecuación 2.6 se muestra la conversión del espacio de color de RGB a CMY. El modelo CMY se compone de los colores primarios cian, magenta y amarillo.

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2.6}$$

Otro modelo de color que representa la luminancia y la información de color en I y Q es el modelo YIQ. El canal I transmite en el rango naranja - azul y Q en el rango de purpura - verde. La conversión de RGB a YIQ y su inversa se muestran en las Ecuaciones 2.7 y 2.8.

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0,299 & 0.587 & 0.114 \\ 0.596 & -0.274 & -0.322 \\ 0.211 & -0.523 & 0.312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2.7}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.000 & 0.956 & 0.621 \\ 1.000 & -0.272 & -0.647 \\ 1.000 & -1.106 & -1.703 \end{bmatrix} \begin{bmatrix} Y \\ I \\ Q \end{bmatrix} \tag{2.8}$$

La selección de una región en una imagen permite agregar información al vector de características basadas en propiedades métricas tales como área, perímetro y centro de gravedad (Ecuaciones 2.9 y 2.10). Donde el área de una región es el número de píxeles que la conforman.

Perímetro a partir del código de cadena:

$$P = \sum_i \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \tag{2.9}$$

Centro de gravedad por valor de intensidad:

$$\bar{x} = \frac{\sum_i f(x_i, y_i) x_i}{\sum_i f(x_i, y_i)}, \bar{y} = \frac{\sum_i f(x_i, y_i) y_i}{\sum_i f(x_i, y_i)} \tag{2.10}$$

El espacio de color HSI esta basado en percibir información del color de manera similar a la percepción humana. El canal H caracteriza el color en términos de tono (Hue), el canal S recolecta información de la saturación del color y finalmente el canal I almacena el brillo (intensity). Las transformaciones mas comunes en la conversión de RGB a HSI son las ecuaciones 2.12, 2.11 y 2.13.

$$I = \frac{R+G+B}{3} \tag{2.11}$$

$$H = \frac{\alpha - arctan\left(\frac{\sqrt{3}(R-I)}{G-B}\right)}{2\pi} \tag{2.12}$$

con $\alpha = \frac{\pi}{2}$ si $G > B$, $\alpha = \frac{3\pi}{2}$ si $G < B$. En el caso de tener un color con igual componente verde y azul (G=B), el tono será igual a la unidad (H=1).

$$S = \sqrt{R^2 + G^2 + B^2 - 2RG - RB - BG} \tag{2.13}$$

Textura

Los patrones de repetición tales como texturas, son otra fuente de información suficiente como para conformar el vector de características. Los pixeles de textura son llamados *Texel*, compuestos de información invariable sin importar las diferentes posiciones, las deformaciones u orientaciones de la región [21,22]. En ocaciones los *Texel* contienen dos niveles de patrón: alto y bajo, Figura 2.6.

La técnica más empleada para adquirir información de una textura es la estadística de los niveles de gris de la imagen [21]. Haralick (1979) y Ballard y Brown (1982) propusieron un conjunto de descriptores. Estos descriptores son los utilizados para la extracción de características mediante textura. Algunos descriptores se muestran a continuación.

Figura 2.6: *Imagen de textura con 2 texel: nivel bajo patrón de cosido y nivel alto patrón de cuadrados y rectángulos.*

$$max(C_{ij}) \tag{2.14}$$

$$\sum_i \sum_j |C_{ij}|^2 \tag{2.15}$$

$$\sum_i \sum_j |i-j|^k C_{ij} \tag{2.16}$$

$$\sum_i \sum_j \frac{1}{(i-j)^k} C_{ij} \tag{2.17}$$

$$-\sum_i \sum_j C_{ij} log c_{ij} \tag{2.18}$$

Los valores que se sustituyen en los descriptores se obtienen de la *matriz de correlación del nivel de gris* ,$\boldsymbol{C}$. Esta matriz se construye por medio de un operador de posición P (Ecuación 2.19) que define un desplazamiento y un ángulo en la matriz. P determina una distancia entre dos o mas pixeles, con el propósito de cuantificar la correlación en el nivel de gris entre ellos.

$$P = (d,a), \text{ Donde: } d = (...,-2,-1,0,1,2,..) \text{ y } a = (0,\frac{\pi}{4},\frac{\pi}{2},...) \tag{2.19}$$

Para construir la matriz C, es necesario definir el orden del nivel de gris, es decir, establecer el número de píxeles a concurrir. La matriz tendrá una dimensión de acuerdo al número de niveles de gris que contenga. Un ejemplo de la construcción de C se muestra en la Figura 2.7, donde el operador P esta definido como $P = (0,0)$ [23].

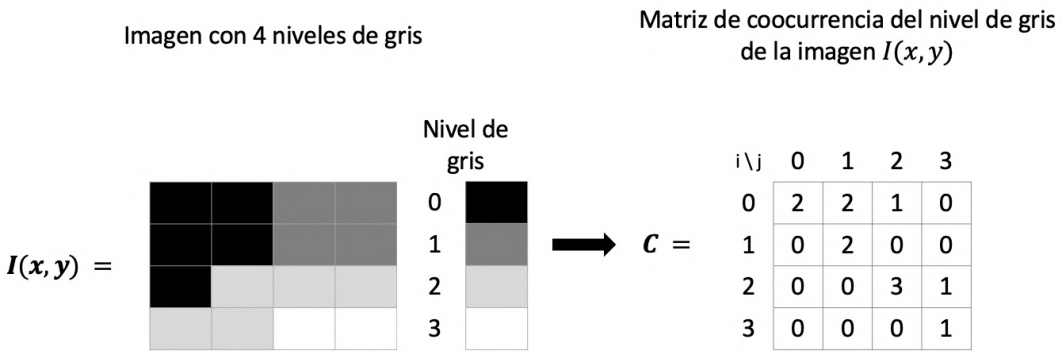

Figura 2.7: *Matriz de ocurrencia C dada una imagen* $I(x,y)$.

La interpretación de la matriz $C(0,0)$ de la Figura 2.7 es: para un píxel de referencia con nivel de intensidad 0 dos veces tiene de vecino a la derecha un pixel con un nivel de gris igual a 0.

2.2. ALGORITMOS Y TÉCNICAS PARA LA REDUCCIÓN DE DIMENSIONALIDAD

En **machine learning**, la reducción de dimensionalidad es una técnica aplicada en la fase de preprocesamiento para modelos tanto de aprendizaje supervisado y no supervisado. EL tratamiento de las variables en técnicas de reducción de dimensionalidad puede ser aplicada mediante dos formas:

- Variables compuestas: simplifican el espacio muestral de las variables mediante la combinación lineal normalizada de las mismas, por ejemplo: componentes principales, variables canónicas, etc [24].
- Selección o eliminación de variables (características): reducción de la dimensionalidad omitiendo o eliminando algunas variables de acuerdo a un criterio establecido, tal técnica es vista en algoritmos de optimización combinatoria, por ejemplo: Recocido Simulado (SA).

Incorporar técnicas de reducción de demensionalidad permite mejorar el rendimiento y la complejidad computacional, Figura 2.8. Sin embargo, se corre el riesgo de perder información significativa. Las técnicas empleadas en este trabajo de tesis son PCA, SA y TA, las cuales se describen en las Secciones 2.2.1, 2.2.2 y 2.2.3.

2.2.1. Análisis de Componentes Principales

La técnica es debida a Hotelling (1933), aunque sus orígenes se encuentran en los ajustes ortogonales por mínimos cuadrados introducidos por K. Pearson (1901).

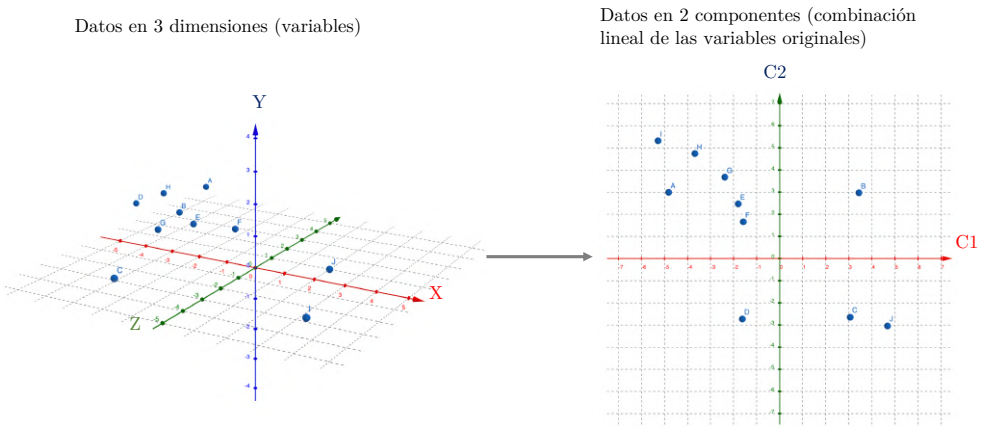

Figura 2.8: *Ejemplo de reducción de la dimensionalidad de 3D a 2D.*

El objetivo es identificar las combinaciones lineales que mejor representen las variables $X_1, X_2, \ldots, X_p$. Sean $Z_1, Z_2, \ldots, Z_h$ las combinaciones lineales de las p variables originales tal que $h < p$, es decir:

$$Z_h = \sum_{j=1}^{p} \phi_{jh} X_j \tag{2.20}$$

Donde la componente principal:

$$Z_1 = \phi_{11} X_1 + \phi_{21} X_2 + \phi_{31} X_3 + ... + \phi_{p1} X_p \tag{2.21}$$

Los términos ϕ reciben el nombre de *loadings* y pueden interpretarse como el peso o importancia que tiene cada variable en cada componente. La información relevante a obtener mediante PCA son :

La varianza de los datos centralizados:

$$\sum_{j=0}^{p} Var(X_j) = \sum_{j=1}^{p} \frac{1}{n} \sum_{i=0}^{n} x_{ij}^2 \tag{2.22}$$

La varianza explicada por cada componente h.

$$\frac{1}{n} \sum_{i=0}^{n} z_{ih}^2 = \frac{1}{n} \sum_{i=0}^{n} \left(\sum_{j=0}^{p} \phi x_{ij} \right)^2 \tag{2.23}$$

En la Figura 2.9 se observa un paso a paso para la obtención de variables relevantes. Cabe mencionar que la técnica PCA arroja una matriz de componentes donde cada uno es una combinación lineal de todas las variables originales, es decir, esta técnica no es una algoritmo de selección de variables, ya que como se mensionado, utiliza todas las variables originales para generar combinaciones lineales para ordenarlas de mayor a menor de acuerdo a el porcentaje de varianza explicada de cada componente Z_h.

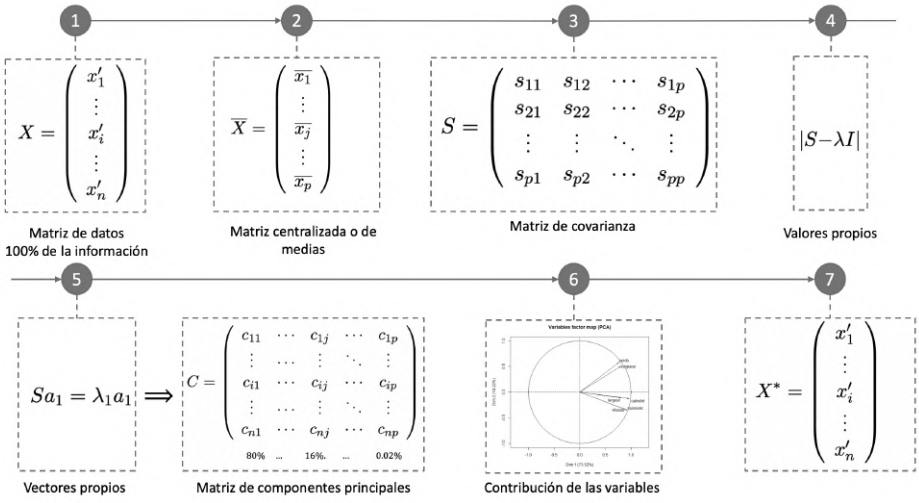

Figura 2.9: *Metodología para obtención del vector óptimo de variables X^* utilizando PCA.*

Sin embargo, seleccionando las variables con los mas altos valores de ϕ de la componente con mayor varianza explicada Z_1 esta técnica se puede emplear como selector de característica.

2.2.2. Algoritmo *Simulated Annealing*

El algoritmo *Simulated Annealing* (SA) fue propuesto por Kirpatrick en 1983 [17]. SA representa el proceso termodinámico donde el metal es calentado y enfriado para aumentar su ductilidad. El Algoritmo 1 muestra el SA clásico, el cual tiene dos ciclos. El ciclo externo, linea 3, permite ajustar la temperatura T_k mediante el parámetro α y la longitud de metropolis L_{max} con el parámetro β. El ciclo interno, también conocido como metrópolis, de la linea 4 realiza una búsqueda de una nueva solución hasta que el equilibrio estocástico (equilibrio térmico) es alcanzado en cada temperatura. Este algoritmo permite aceptar malas soluciones mediante el criterio de aceptación de Boltzmann presente en la línea 9.

Algorithm 1 Simulated Annealing

```
procedure SIMULATEDANNEALING(T_i, T_f, β, α, L_max)
    S_i ← initialsolution()
    while T_k > T_f do
        while L < L_max do
            S_j ← generteneighbor(S_i)
            Δ ← E(S_j) − E(S_i)
            if Δ_new ≤ 0 then
                S_i ← S_j
            else if RANDOM(0,1) > e^{−Δ/T_k} then
                S_i ← S_j
            end if
        end while
        T_{k+1} ← αT_k
        L_max ← βL_max
    end while
return S_i
end procedure=0
```

Muchos de los trabajos que emplean SA como optimización combinatoria mantienen una estructura similar al Algoritmo 1. A continuación se describen los aspectos de SA:

Temperatura inicial y final. La temperatura inicial del algoritmo debe de permitir la aceptación de todas las posibles transiciones y también el libre movimiento en el espacio de soluciones. Cuando la temperatura es muy alta se acepta casi cualquier solución aunque esta sea peor que la actual. Cuando la temperatura es muy pequeña sólo acepta soluciones que sean mejor que la actual. Si la temperatura inicial es muy alta se puede desperdiciar mucho tiempo en los primeros ciclos. Si es muy baja la probabilidad de quedar atrapado en un óptimo local es muy alta.

En cambio para la temperatura final, si esta es muy alta la probabilidad de quedar atrapado en un óptimo local es muy alta. Si es muy baja el proceso de búsqueda será muy exhaustivo y consumirá demasiado tiempo. Por lo tanto, una buena elección de la temperatura inicial y final tiene una gran importancia para alcanzar un buen desempeño del SA.

En [25] se muestra cómo calcular los valores de la temperatura inicial (T_i) y final (T_f), donde se realiza

lo siguiente. Sea $P_A(S_j)$ la probabilidad de aceptar una solución (S_j) generada de una solución actual (S_i), y $P_R(S_j)$ la probabilidad de rechazarla. La probabilidad de rechazar S_j se puede establecer en términos de $P_A(S_j)$:

$$P_R(S_j) = 1 - P_A(S_j) \tag{2.24}$$

aceptar o rechazar S_j únicamente depende del costo del deterioro que provocaría el cambio de la solución actual, esto es:

$$P_A(S_j) = g(Z(S_i) - Z(S_j)) = g(\Delta Z_{ij}) \tag{2.25}$$

donde $Z(S_i)$ y $Z(S_j)$ son los costos asociados a S_i y S_j respectivamente, y $g(\Delta Z_{ij})$ es la probabilidad de aceptar la diferencia del costo $\Delta Z_{ij} = Z(S_i) - Z(S_j)$

El esquema de vecindad para S_i se define como cualquier solución S_j donde únicamente varía un elemento en comparación con S_i. Por lo tanto el deterioro máximo y mínimo pueden ser definidas como:

$$\Delta Z_{V_{\text{máx}}} = \text{máx}\{Z(S_j) - Z(S_i)\} \quad \forall S_j \in V_{s_i}, \forall S_i \in S \tag{2.26}$$

$$\Delta Z_{V_{\text{mín}}} = \text{mín}\{Z(S_j) - Z(S_i)\} \quad \forall S_j \in V_{s_i}, \forall S_i \in S \tag{2.27}$$

donde $\Delta Z_{V_{\text{máx}}}$ y $\Delta Z_{V_{\text{mín}}}$ son los deterioros máximos y mínimos, respectivamente.

La probabilidad de aceptar cualquier solución al inicio debe ser casi uno:

$$P_A(S_j) = P_A(\Delta Z) \approx 1 \tag{2.28}$$

Como $\Delta Z_{V_{\text{máx}}}$ proporciona el máximo deterioro que se puede producir. Para asegurar que $\Delta Z_{V_{\text{máx}}}$ sea aceptado en la temperatura T_i es estableciendo la probabilidad de aceptación con $P_A(\Delta Z_{V_{\text{máx}}}) \approx 1$. El valor de T_i se puede obtener mediante la siguiente ecuación:

$$e^{\frac{-\Delta Z_{V_{\text{máx}}}}{T_i}} = P_A(\Delta Z_{V_{\text{máx}}}) \tag{2.29}$$

donde se puede obtener T_i con:

$$T_i = \frac{-\Delta Z_{V_{\text{máx}}}}{\ln(P_A(\Delta Z_{V_{\text{máx}}}))} \tag{2.30}$$

donde $P_A(\Delta Z_{V_{\text{máx}}})$ es el grado de aceptación que se le quiere dar, debe estar en el rango $[0,1)$. Si se le asigna un valor de 0.99 se garantiza que al inicio del proceso aceptará al peor miembro de la vecindad con una confiabilidad del 99 % [25]. Cuando SA se encuentra cerca del punto de congelamiento, la probabilidad de aceptar la degradación mínima posible del vecindario deber ser aproximadamente cero. Cerca del punto de congelamiento se debe cumplir:

$$P_A(S_j) = P_A(\Delta Z) \approx 0 \tag{2.31}$$

Como $\Delta Z_{V_{\text{mín}}}$ es el deterioro mínimo que puede ser producido, entonces se puede calcular T_f de la siguiente forma:

$$e^{\frac{-\Delta Z_{V_{\text{mín}}}}{T_f}} = P_A(\Delta Z_{V_{\text{mín}}}) \tag{2.32}$$

despejando T_f se obtiene:

$$T_f = \frac{-\Delta Z_{V_{\text{mín}}}}{\ln(P_A(\Delta Z_{V_{\text{mín}}}))} \tag{2.33}$$

donde $P_A(\Delta Z_{V_{\text{mín}}})$ se refiere al grado de rechazo. Si el deterioro es cero la temperatura final es cero. A esta probabilidad con un valor de 0.05 tiene una confiabilidad de 95 % [25].

Esquema de enfriamiento. El esquema de enfriamiento permite la convergencia al óptimo global del problema. Si la temperatura es disminuida lentamente aumentará la precisión de la solución, pero aumentará el tiempo de ejecución.

La velocidad de enfriamiento permite la convergencia a la solución óptima. La reducción geométrica es la más utilizada ya que es fácil e intuitiva de acelerar. Las principales funciones de enfriamiento son las siguientes.

Tabla 2.1: *Principales funciones de enfriamiento.*

Nombre	Función de enfriamiento
Geométrica	$T_{k+1} = \alpha T_k$
Exponencial	$T_{k+1} = e^{\alpha T_k}$
Logarítmica	$T_{k+1} = \frac{T_k}{\ln(\alpha)}$

Longitud de la cadena de Markov. La longitud de la cadena de Markov es el número de iteraciones que se realizan en el ciclo de metrópolis y este busca el equilibro de Markov o estocástico a cada temperatura c_k. Existen dos enfoques para determinar la longitud de la cadena de Markov. El enfoque estático, el cual ejecuta una misma cantidad de veces la metrópolis en todas las temperatura c_k. El enfoque dinámico que ejecuta la metrópolis dependiendo de ciertos parámetros.

En el enfoque estático se requiere una gran cantidad de tiempo para sintonziar los parámetros. Dentro del enfoque dinámico están los adaptativos y los de umbral. Los de umbral dependen del número de soluciones aceptadas o rechazadas en cada ciclo de temperatura. Los adaptativos se basan en la media y la desviación estándar. Esto puede provocar que en temperaturas cercanas a T_f el ciclo termine muy rápido debido a la baja probabilidad de aceptar soluciones.

Para establecer la longitud máxima de la cadena de Markov se utiliza la siguiente ecuación.

$$L_{\text{máx}} = -\ln(P_r(S_i))|V_{s_i}| \tag{2.34}$$

donde $|V_{s_i}|$ representa el tamaño de la vecindad y $P_r(S_i)$ es la probabilidad de rechazar una solución S_i.

Criterio de aceptación. SA tiene dos criterios de aceptación. El primero aceptan soluciones únicamente cuando son mejores que al actual. Mientras tanto, el segundo criterio acepta soluciones malas, dichas soluciones son aceptadas mediante la distribución de Boltzmann. Al aceptar una mala solución permite explorar nuevos espacios de soluciones y así escapar de óptimos locales aumentando la probabilidad de encontrar el óptimo global, Figura 2.10.

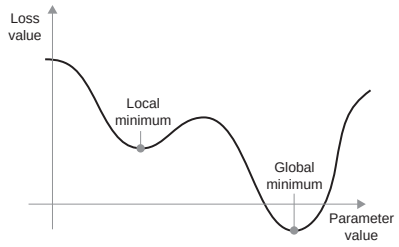

Figura 2.10: *Representación de óptimo local y global.*

Si la nueva solución es mejor que la actual entonces se actualiza la solución actual con la nueva. El criterio de aceptación más utilizado es la probabilidad de Botlzmann, el cual está dado por:

$$P_A(S_i) = \frac{-\Delta Z(S_i)}{T_k} \tag{2.35}$$

Conforme la temperatura va disminuyendo la aceptación de malas soluciones se vuelve más rigurosa. Al principio, como la temperatura inicial T_i es alta, es más alta la probabilidad de aceptar malas soluciones. Sin embargo, conforme va decrementando la temperatura T_k el criterio de aceptación bajo la distribución de Boltzmann se va haciendo más estricto.

2.2.3. Algoritmo *Threshold Accepting*

El algoritmo *Threshold Accepting* (TA) propuesto por [26] es formalmente muy similar a SA, la diferencia radica en el criterio de aceptación, Algoritmo 2. SA acepta una solución en base a dos criterios de aceptación: si la nueva solución S_j es mejor que la actual S_i o si la nueva solución cumple el criterio de distribución de Boltzmann $e^{\frac{-\Delta}{T_k}}$. Sin embargo, TA simplifica la aceptación de una solución descartando el criterio de distribución de Boltzmann.

Algorithm 2 Threshold Accepting

```
0: procedure THRESHOLDACCEPTING(T_i, T_f, β, α, L_max)
0:     S_i ← initialsolution()
0:     while T_k > T_f do
0:         while L < L_max do
0:             S_j ← genertneighbor(S_i)
0:             Δ ← E(S_j) − E(S_i)
0:             if Δ < T_k then
0:                 S_i ← S_j
0:             end if
0:         end while
0:         T_{k+1} ← αT_k
0:         L_max ← βL_max
0:     end while
   return S_i
0: end procedure=0
```

El criterio de aceptación de TA incorpora un parámetro T conocido como umbral, el cual permite un grado de tolerancia en la solución actual S_i en comparativa con la nueva solución S_j. El parámetro T se toma de un porcentaje de la temperatura T_k actual. Así mismo, el criterio de aceptación puede ser expresado como en el Algoritmo 3.

Algorithm 3 Threshold Accepting

```
0: procedure THRESHOLDACCEPTING(T_i, T_f, β, α, L_max)
0:     S_i ← initialsolution()
0:     while T_k > T_f do
0:         while L < L_max do
0:             S_j ← genertneighbor(S_i)
0:             Δ ← E(S_j) − E(S_i)
0:             if E(S_j) < E(S_i) then
0:                 S_i ← S_j
0:             else if Δ < T_k then
0:                 S_i ← S_j
0:             end if
0:         end while
0:         T_{k+1} ← αT_k
0:         L_max ← βL_max
0:     end while
   return S_i
0: end procedure=0
```

En la línea 9 del Algoritmo 3 se aprecia que la aceptación de la diferencia de la calidad de las soluciones Δ esta condicionada por el valor de temperatura T_k. Este valor se modifica mediante el esquema de enfriamiento geométrico en la línea 13, por lo cual se vuelve mas riguroso conforme se acerca al equilibrio dinámico.

2.3. APRENDIZAJE PROFUNDO

La inteligencia artificial, nacida en la década de los 50s, es un campo que abarca el aprendizaje automático y el aprendizaje profundo, entre otros [27]. El aprendizaje automático (Machine Learning, ML) es un paradigma de programación que permite transformar datos de entrada en salidas con significado es decir, encuentra representaciones apropiadas para dichos datos.

ML contempla 3 diferentes modalidades de aprendizaje: aprendizaje supervisado, no supervisado y por refuerzo. El aprendizaje supervizado se realiza mediante el entrenamiento de un modelo con un conjunto de datos previamente identificados, este aprendizaje se emplea para problemas de clasificación o regresión utilizando algoritmos como Máquinas de Soporte Vectorial (SVM), Árboles de decisión, Regresión logística, entre otros.

El aprendizaje no supervizado se realiza mediante técnicas de agrupamiento de datos, debido a que los datos de entrada no se encuentran identificados. Por lo tanto solo se puede describir la estructura de los datos. Los algoritmos empleados en este aprendizaje son algoritmos de clustering (k-means, DBSCAN), análisis de componentes principales, entre otros. Finalmente, el aprendizaje por refuerzo se basa en un proceso de recompensas en base a las respuestas de acuerdo a las acciones tomadas.

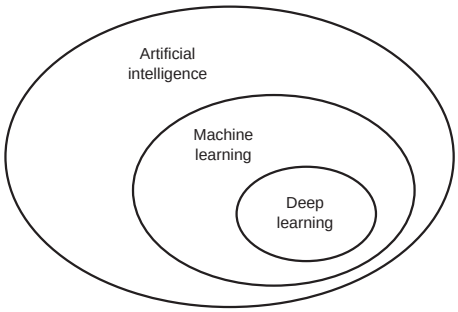

Figura 2.11: *Deep Learning en Machine Learning e IA.*

El aprendizaje profundo (Deep Learning, DL) es un subcampo de ML ,como se observa en la Figura 2.11, donde el aprendizaje se realiza sucesivamente en capas, es decir, la información aprendida en la primera capa es enviada como información de entrada a la siguiente capa. En las primeras capas aprenden aspectos sencillos por lo que aumentando del número de capas se volverán más complejos. DL lleva a cabo el aprendizaje utilizando Redes Neuronales.

2.3.1. Perceptrón

Perceptrón es la unidad básica de una Red Neuronal (NN), el cual simula el comportamiento de una neurona. Al igual que una neurona, un perceptrón esta compuesto por señales de entrada, pesos sinápticos, un núcleo de procesamiento y una salida. En la Figura 2.12 se tiene un conjunto de variables x_1, x_2 que representan las señales o vector de entrada. Los pesos sinápticos estan dados por w_1, w_2 asignan un valor a las variables de acuerdo a su importancia. El núcleo de procesamiento o perceptrón esta compuesto por una función de regresión lineal, el cual genera una salida binaria.

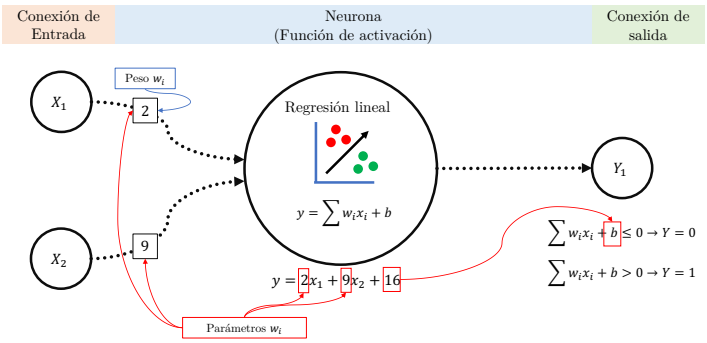

Figura 2.12: *Representación de una neurona o perceptrón.*

Sin embargo, este esquema tiene dos limitantes. La primera radica en su naturaleza de clasificación el cual es de manera binaria. La solución de esto es incorporar más perceptrones para realizar procesamientos multi-clase y poder generar mas de una salida, a esto se le conoce como una Red Neuronal (NN).

2.3.2. Red Neuronal

Una red neuronal esta compuesta por tres capas: capa de entrada, capa oculta y capa de salida. En la capa de entrada se representan todas la variables o señales de entrada al modelo. La capa oculta es un compendio de capas con neuronas, cada capa puede estar compuesta por una o varias neuronas. La capa de salida evalúa los resultados obtenidos de las capas anteriores mediante una función de coste. La salida puede ser de naturaleza bi-clase o multi-clase. En la Figura 2.13, se observan dos arquitecturas de NN.

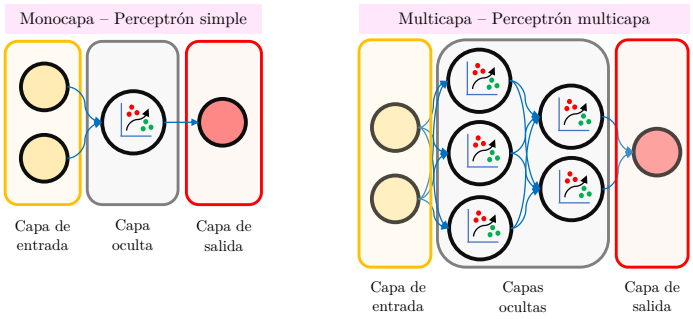

Figura 2.13: *Ejemplo de arquitecturas de redes neuronales.*

Sin embargo, cada neurona en cada capa esta compuesta por una función de regresión lineal que por su naturaleza se podría representar de igual manera como una única neurona, Figura 2.14, lo que da lugar a la segunda limitante.

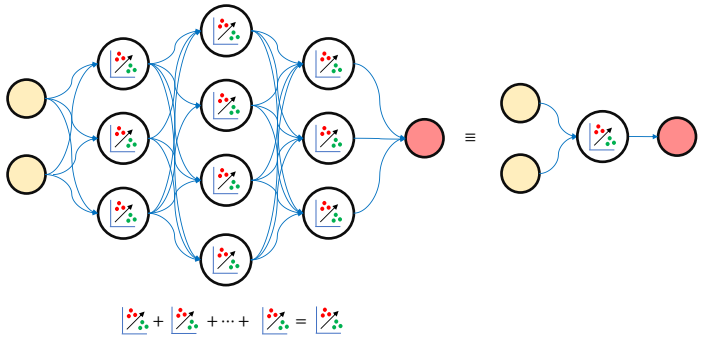

Figura 2.14: *Comparativa de una Red Neuronal y un perceptrón con funciones de activación lineales.*

La solución a esta limitante es incorporar en cada neurona una función que permita transformar la salida de la regresión lineal en una no líneal, dichas funciones son conocidas como funciones de activación. Estas funciones permiten obtener resultados diferentes a 0 y 1.

2.3.3. Función de activación

La función sigmoide, Ecuación 2.36, transforma los valores de x en una escala de (0, 1), donde los valores altos tienen de manera asintótica a 1 y los valores muy bajos tienden de manera asintótica a 0.

$$f(x) = \frac{1}{1 - e^{-x}} \tag{2.36}$$

La función Tangente hiperbólica, Ecuación 2.37, es muy similar a la sigmoide, sin embargo esta transforma los valores de x en una escala de $(-1, 1)$.

$$f(x) = \frac{2}{1 - e^{-2x}} - 1 \tag{2.37}$$

La función ReLU (Unidad Lineal Rectificada), Ecuación 2.38, a diferencia de la sigmoidal y la tangente hiperbólica que transforman todos los valores de x, esta anula los valores negativos y mantiene los positivos sin ningún tipo de transformación.

$$f(x) = max\{0, x\} = \begin{cases} 0 & \text{si } x < 0 \\ x & \text{si } x \geq 0 \end{cases} \tag{2.38}$$

La función de activación Softmax, Ecuación 2.39, transforma los datos x en función de probabilidades es decir, la suma de todas las salidas debe dar 1. Normalmente se utiliza en la capa de salida.

$$f(x) = max\{0, x\} = \begin{cases} 0 & \text{si } x < 0 \\ x & \text{si } x \geq 0 \end{cases} \tag{2.39}$$

2.3.4. Función de costo

La función de costo en la capa de salida permite evaluar el error de aprendizaje obtenido mediante el valor estimado y el real, con el fin de optimizar los parámetros de la NN. Las funciones de costo que pueden emplearse son:

$$\text{RMSE} = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}} \tag{2.40}$$

$$\text{MAE} = \sum_{i=1}^{n} \frac{|\hat{y}_i - y_i|}{n} \tag{2.41}$$

$$\text{MASE} = \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{\frac{n}{n-1} \sum_{i=2}^{n} |\hat{y}_n - y_{n-1}|} \tag{2.42}$$

Donde $\hat{y}$ es el valor predicho, $\hat{y}$ el valor real, n el total de resultados obtenidos por NN. Por otro lado, para poder retroalimentar a la NN es necesario analizar la gradiente de la función con el propósito de minimizar el error de la función de coste.

2.3.5. Descenso de gradiente y propagación hacia atrás

El aprendizaje de un modelo NN se realiza por medio de la aplicación de un algoritmo de optimización en la capa de salida el cual permita encontrar el mínimo valor donde la función converge, Figura 2.15. Dicha capa es el resultado del conjunto de funciones aplicados en las capas anteriores por lo que se conoce como composición de funciones, Ecuación 2.43.

$$y = C(a(Z^L)) \tag{2.43}$$

donde $C(x)$ es la función de costo, $a(x)$ es la función de activación, Z^L es la suma ponderada en la capa L. El descenso de gradiente es un algoritmo de optimización que mediante un proceso iterativo permite encontrar el valor mínimo donde converge una función, Ecuación 2.44.

$$\nabla f = \begin{bmatrix} \frac{\partial C}{\partial w_1} \\ \frac{\partial C}{\partial w_2} \\ \frac{\partial C}{\partial b} \end{bmatrix} \tag{2.44}$$

Para esto, es necesario encontrar la derivada parcial de la función con respecto a cada variable $\frac{\partial f}{\partial x}$ debido a que indica el valor y el sentido en que se encuentra el mínimo más próximo, puede ser tanto local como global. Obtenido el gradiente es restado a los parámetros para descender. Sin embargo, la velocidad o caminata de aprendizaje esta definido por α con valores entre 0 y 1; un valor $\alpha \approx 0$ el descenso será muy lento y corre el peligro de nunca alcanzar el mínimo; un valor $\alpha \approx 1$ el descenso es muy rápido y podría saltar el mínimo.

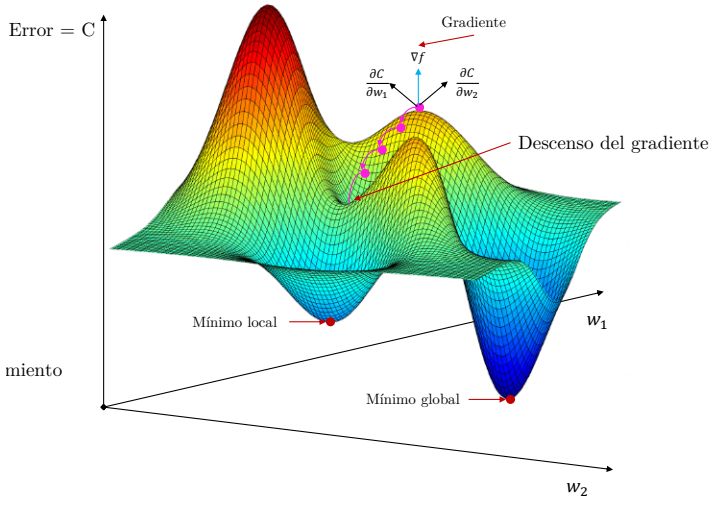

Figura 2.15: *Caminata a un mínimo local o global mediante el Descenso de gradiente.*

2.3.6. U-NET

U-Net es una red neuronal de convolución para el pronostico de máscaras binarias aplicada para la segmentación de imágenes [28]. La arquitectura de la red se ilustra en la Figura 2.16, la cual esta compuesta por dos fases de convolución:

- Downsampling: tambien llamado Encoder, permite reducir el mapa de características obtenida de un filtro de convolución, aplicando maxpooling o un filtro de convolución con un tamaño de kernel menor al anterior.
- Upsampling: tambien llamado Decoder, permite aumentar el mapa de características aplicando unpooling o usando una convolución transpuesta.

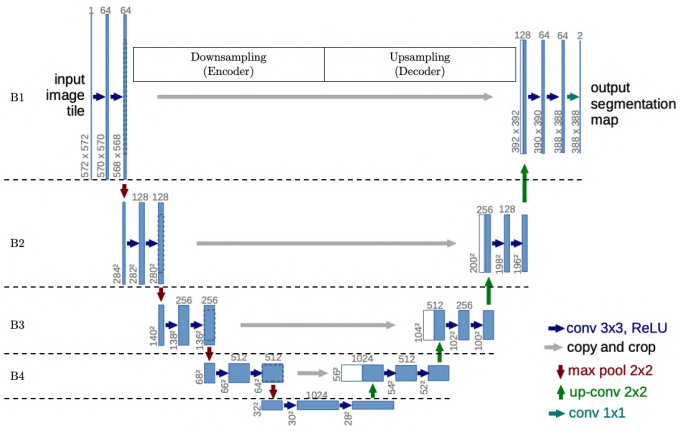

Figura 2.16: *Arquitectura de U-Net.*

Las fases de la arquitectura están compuestas por cuatro bloques de convolución. Los primeros cuatro bloques corresponden a la fase de contracción o Encoder. Esta primera ruta esta compuesta por cuatro bloques de convolución de 3x3, cada una seguida de una función de activacion unidad lineal rectificada (ReLU) y una operación de maxpooling de 2x2 para reducir la resolución. En cada paso de reducción de resolución, se duplica el número de canales de funciones. En la Figura 2.17 se puede observar una representación matricial de los primeros tres bloques de la fase de downsampling.

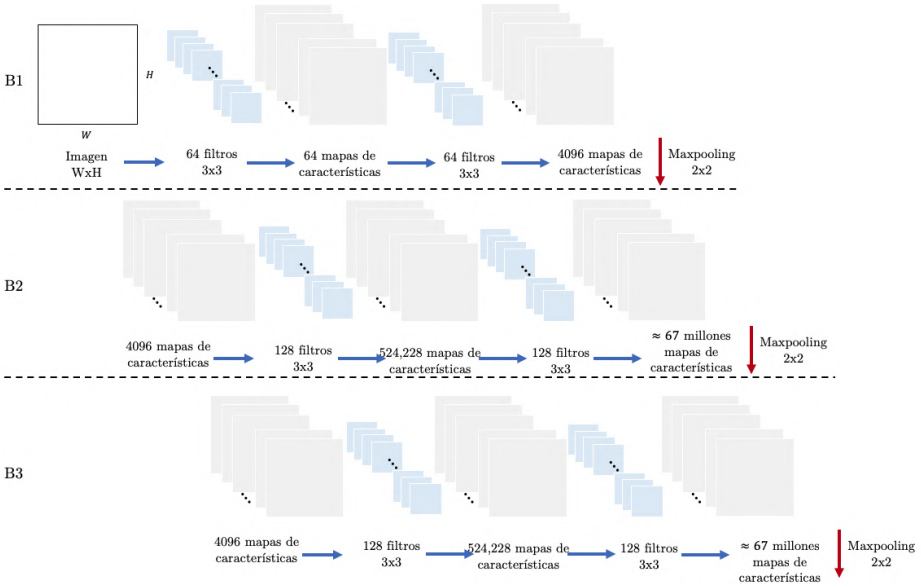

Figura 2.17: *Tres primeros bloques de convolución de la Arquitectura de U-Net.*

Los bloques correspondientes a la fase de Upsampling, ruta expansiva, consiste en un muestreo superior del mapa de características seguido de una convolución 2x2 que divide a la mitad el número de canales de características, una concatenación con el mapa de características recortado correspondientemente de la ruta de downsampling y dos 3x3 convoluciones, cada una con función de activación ReLU. El recorte es necesario debido a la pérdida de píxeles de borde en cada convolución. En la capa final, se usa una convolución 1x1 para mapear cada vector de características de 64 componentes al número deseado de clases. En total, la red tiene 23 capas convolucionales.

2.3.7. YOLO: You only look once

YOLO es una red neuronal de convolución para la localización de objetos mediante coordenadas aplicada para la detección de objetos. El modelo predice cuadros delimitadores (bounding box) utilizando grupos de dimensiones como cuadros de anclaje. La red predice 4 coordenadas para cada bounding box, t_x, t_y, t_w, t_h. Si la celda está desplazada desde la esquina superior izquierda de la imagen por (c_x, c_y) y el bounding box anterior tiene ancho y alto p_w, p_h, entonces las predicciones corresponden a:

$$\begin{aligned} b_x &= \sigma(t_x) + c_x \\ b_y &= \sigma(t_y) + c_y \\ b_w &= p_w e^{t_w} \\ b_h &= p_h e^{t_h} \end{aligned} \tag{2.45}$$

YOLOv3 predice un valor objetivo para cada bounding box mediante regresión logística, será 1 si el bounding box anterior se superpone a un objeto real más que cualquier otro bounding box anterior. Si el bounding box anterior no es el mejor pero se superpone a un objeto real ignoramos la predicción. YOLOV3 solo asigna un bounding box antes para cada objeto.

La representación del vector de solución de un bounding box se observa en la Ecuación 2.47. Sin embargo, se predicen B bounding boxes B y C probabilidades de la clase. La predicción del bounding box tiene seis componentes p, x, y, w, h, confianza. Las coordenadas (x, y) representan el centro del cuadro, en relación con la ubicación de la celda de la cuadrícula. Estas coordenadas se normalizan para caer entre 0 y 1. Las dimensiones del cuadro (w, h) también se normalizan a [0, 1], en relación con el tamaño de la imagen.

$$[x, y, w, h, confidence] \tag{2.46}$$

$$y = (p, x, y, w, h, confidence) \tag{2.47}$$

YOLOv3 predice cajas en 3 escalas diferentes: 13x13, 26x26, 52x52. Al usar el conjunto de imágenes de COCO que tiene 80 clases se puede formar un tensor con las siguientes dimensiones $NN[3(4+1+80)]$ para las 4 valores del *bounding box*, 1 predicción de objetividad y 80 predicciones de clase. Por lo que la predicción contiene una clasificación múltiple.

CAPÍTULO 3

Estado del arte

En este capitulo se señalan los trabajos relacionados con problemas de reconocimiento de objetos relacionados con casos de oclusión y ambientes complejos en la Sección 3.1. Del mismo modo, se revisan trabajos vinculados con problemas de segmentación asociados a procesos de algoritmos de selección de características, algoritmos de agrupamiento y clasificadores.

3.1. RECONOCIMIENTO DE OBJETOS EN IMÁGENES

Los artículos están relacionados con el estudio realizado en base a reconocimiento de patrones considerando factores como oclusión, factores ambientales o ambientes complejos.

El trabajo de [29], realiza un proceso de segmentación en donde trabaja con hojas de tomate tomadas

en un invernadero. Las condiciones de adquisición de las imágenes fueron bajo un ambiente controlado (control de la iluminación, posición y distancia de entre las plantas de tomate y las cámaras), es decir no se trabajo sobre un ambiente real o complejo. El vector de características estuvo compuesto por propiedades adquiridas mediante el modelo de color RGB.

Un trabajo sobre reconocimiento de plantas es el de [14], donde el vector de características que evalúa en los clasificadores de SVM y KNN esta conformado por la firma espectral de la hoja, patrones de las venas, propiedades del histogram y geométricas. La evaluación esta comprendida por la tasa de falso positivo, falso negativo y tasa de aceptación genuina. El conjunto de datos es un conjunto propio llamado $VISLeaf$. Para lo cual esta bajo condiciones ideales de adquisición y no contempla oclusión.

Otro trabajo donde se realiza un reconocimiento de plantas es el trabajo de [30]. En este trabajo se utiliza un conjunto de datos conocido como *iris*. Emplea los clasificadores de Naive Bayes, K vecinos próximos (KNN), SVM. El conjunto de datos proporciona atributos tales como largo y ancho del sépalo y pétalo. Para validar la clasificación de los algoritmos mencionados se utilizó la precisión de los clasificadores.

El trabajo de [6] se centra en el reconocimiento de plantas de plátano enfermas. Utiliza los descriptores de textura y métricas de los modelos de color para componer al vector de características. Realiza el entrenamiento y prueba con los clasificadores de SVM, Árbol de decisión, SVM- Lineal, SVM-Cuadrático, SVM-Cúbico y KNN. Los índices de evaluación son la sensibilidad, la precisión y la tasa de falso positivo. Las condiciones de las imágenes se realizaron bajo ambiente complejo pero sin oclusión.

Un trabajo que trata la oclusión en el problema de reconocimiento de objetos es [18], el cual trabaja con tres conjunto de datos públicos llamados *Swedish*, *Flavia* y *Leafsnap*. Estos conjuntos de datos están bajo un ambiente controlado y condiciones ideales. Realizaron experimentaciones sobre 0 %, 25 % y 50 % de oclusión. El algoritmo propuesto representa los contornos de las hojas con curvas y extrae puntos de características de las curvas. Un procedimiento de concordancia entre las características con conocimiento a priori es empleado para determinar la pertenencia de una hoja o no.

A continuación en la Tabla 3.1 se muestra un resumen de los trabajos citados en el estado del arte. La primera columna indica la referencia del trabajo, la segunda el tipo de imágenes utilizadas, la tercera y cuarta los tipos de características y clasificadores. Finalmente la quinta y sexta columna confirman las condiciones de oclusión o ambientes complejos (AC) presentes en las imágenes.

Tabla 3.1: *Reconocimiento de objetos.*

Referencia	Objeto	Caract.	Clasificador	Oclusión	AC
[29]	Hojas	Color			
[14]	Plantas	firma espectral	SVM		
		Patron en venas	KNN		
		Histograma			
		Geométricas			
[30]	Flores	Geometricas	SVM		
			NB		
			KNN		
[6]	Peste	Color	SVM-L		
		Textura	SVM-Q		
			SVM-Q		
			SVM-C		
			Árbol dec.		
[18]	Hojas	Contorno			
Trabajo	Aves	Color	SVM-L		
		Textura			

En resumen, se puede observar que hasta el momento no se ha encontrado trabajo que involucre los aspectos de oclusión y ambientes no controlados en problemas de reconocimiento de objetos.

3.2. ALGORITMOS DE SELECCIÓN DE CARACTERÍSTICAS EN PROBLEMAS DE SEGMENTACIÓN

La siguiente sección se presentan trabajos recientes sobre métodos de selección de características óptima en la fase de segmentación de imágenes. Tales métodos son tradicionales como PCA hasta enfoques modernos como el aprendizaje automático y los métodos heurísticos.

El trabajo de [31] usaron un algoritmo de selección de características basado en programación genética. Buscan segmentar y clasificar una serie de imágenes de caballos y aviones como objetos de interés a partir de dos conjuntos de imágenes llamadas *Weizmann horse* y *Pascal VOC2012*. Se proponen tres algoritmos de selección de características PGP-FS, NSGP-FS y SPGT-FS, que se compararon con SFS y SBS. Estos algoritmos de selección se validaron utilizando métricas de *Accuracy*, *F1*, *precision* y *Recall*. Los resultados muestran una selección entre 23 y 50 características. Estas características se enviaron a los clasificadores Árbol de decisión, Naive Bayes y Perceptrón multicapa de la herramienta *Weka*. Se extrajeron un total de 53 características relacionadas con el filtro de Gabor, el color y los valores estadísticos basados en una escala de grises.

Asimismo, el trabajo de [32] busca segmentar y clasificar imágenes relacionadas con lesiones cutáneas. Este trabajo utiliza la técnica PCA para seleccionar un conjunto de características mediante la puntuación y el método de entropía de *Boltzman*. Las características que consideran son color, textura y forma; dando un total de 3849 características. El uso de PCA y Boltzman redujo el número de características a 449. La validación de la selección de características esta sujeta a las métricas *DICE*, *Jaccard Index*, *Jaccard distance* y *Seg Diameter*. Las 449 características seleccionadas se clasificaron utilizando los siguientes modelos de aprendizaje automático: *SVM*, *Decision Tree*, *Bagged Trees*, *Subspace discriminant analysis*, *Weightd-KNN*, *Fine-KNN*, *Subspace-KNN*, *Linear discriminant analysis*, *Quadric discriminant analysis*, *Cubic-SVM* y *Quadric -SVM*. Los clasificadores se validaron utilizando las métricas de sensibilidad, especificidad, precisión y *F-score*.

El trabajo de [33] propone una metodología para la detección de enfermedades cítricas utilizando segmentación ponderada optimizada y selección de características. El Pre-procesamiento se compone del filtro *Top-had* para eliminar elementos de ruido, el filtro *Gaussian* para suavizar la imagen y eliminar variaciones de alta intensidad. La segmentación utilizan las técnicas de segmentación con asignación de peso y *Mapa de relevancia* que permite retener los elementos de la imagen con alto contraste. Las características extraídas están relacionadas con el color, la textura y las propiedades geométricas; dando un total de 270 características. PCA se usa para dar un *score* de acuerdo con la varianza explicada de los componentes; la entropía y *Skewness* se calculan para cada componente para seleccionar un vector de 100 entidades con los porcentajes más altos. Las características se entrenaron con los clasificadores

K-Nearest Weighted, Ensemble Boosted Tree, *Decision Tree* y *Linear Discriminant Analysis* con un *10-fold*. La validación de la metodología esta sujeta por las métricas de tasa falsa positiva, tasa falsa negativa, tasa verdadera positiva, tasa falsa negativa, valor de predicción positiva, tasa de descubrimiento falso, área bajo la curva y precisión. Demuestra tener resultados competitivos con el estado del arte.

La selección de características aplicadas a la segmentación de imágenes para problemas de visión se ha utilizado en diferentes campos, como la medicina y la agricultura. El trabajo de [34] busca detectar el glaucoma ubicado alrededor del disco óptico multiparamétrico. Para ello, en la fase de pre-procesamiento se aplicó un filtro bilateral que permite la eliminación del ruido, un *clipping* que permite activar un criterio de umbral para retener objetos con alta intensidad para descartar el ruido de fondo no deseado, y finalmente la normalización del canal R para retener información sobre los patógenos buscados. En la fase de segmentación, se aplican características estadísticas, texton-map y fractales; luego fue sometido a un proceso de selección utilizando el método de *redundancia mínima* ($M_{I(A,B)}$), estas características están entrenadas usando los clasificadores SVM, Random Forest, AdaBoostM1 y RusBoost. La validación del modelo esta sujeta a las métricas de sensibilidad, especificidad, coeficiente de similitud DICE, precisión y superposición de área basada en la matriz de confusión que muestra resultados competitivos con el estado del arte.

A continuación en la Tabla 3.2 se muestra un resumen de los trabajos citados en el estado del arte. La primera columna indica la referencia del trabajo, la segunda el tipo de imágenes utilizadas, la tercera y cuarta indica el número de características y clasificadores, respectivamente. Finalmente la quinta columna indica el algoritmo de selección de características (ASC) utilizado.

Tabla 3.2: *Segmentación de imágenes con algoritmos de selección de características.*

Referencia	Objeto	#Caract.	Clasificador	ASC
[31]	Caballos Aviones	23-50	Decision Tree Naive Bayes NN	Genetico
[32]	lesiones cutaneas	449	SVM Desicion Tree Bagged Tree SDA W-KNN F-KNN Sub-KNN LDA QDA C-SVM Q-SVM	PCA
[33]	Enfermedad cítricos	270	W-KNN EBT DT LDA	PCA
[34]	Glaucoma		SVM RF AdaBoostM1 RusBoost	Redundancia Min.
Propuesta	Ave	20-35	SVM-L	SA TA

De la Tabla 3.2 se puede observar que hasta el momento se han encontrado pocos trabajos aplicando algoritmos de optimización combinatoria para la selección de características óptima.

CAPÍTULO 4

Metodología propuesta

El siguiente capitulo describe la metodología propuesta para el reconocimiento de objetos bajo condiciones de oclusión y ambientes complejos, la cual se observa en la Figura 4.1. Esta se compone de dos fases llamadas entrenamiento y reconocimiento las cuales están sujetas a un proceso de validación.

Se puede observar que la segmentación esta compuesta por 5 procesos. Así mismo que el proceso de selección de características se utiliza en la segmentación como eLos procesos de cada fase se describen en las siguientes secciones.

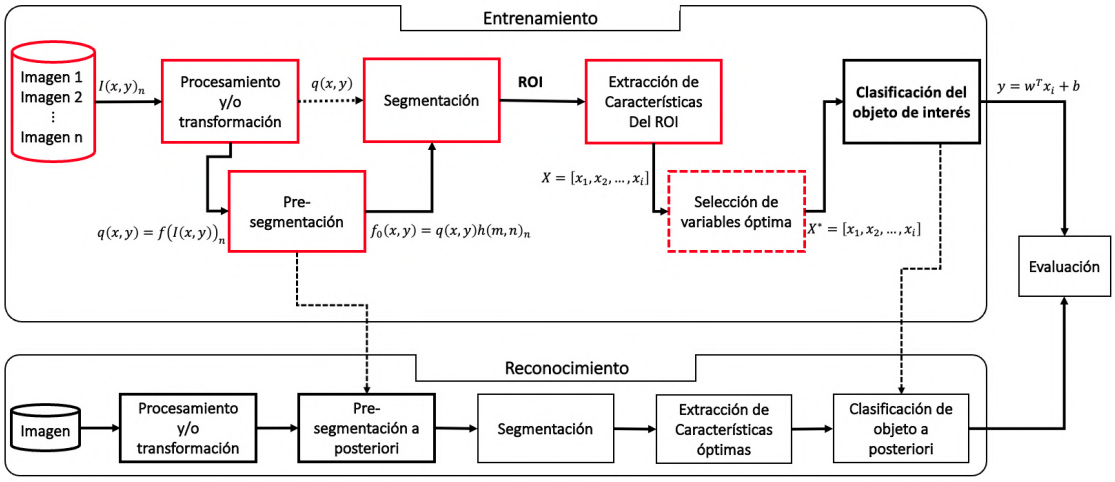

Figura 4.1: *Metodología propuesta*

4.1. BASE DE DATOS: COCO

La primera fase de la metodología esta compuesta por la recolección o adquisición de imágenes tales como el conjunto COCO (Common Contexts in Objects). COCO contiene 91 categorías de las cuales 82 tienen mas de 5000 instancias etiquetadas. En total tiene 2,500,000 de instancias etiquetadas manualmente en 328,000 imágenes 2D. La cantidad de información etiquetada podría ayudar a que los modelos aprendan más detalladamente los objetos y permitir el reconocimiento. En la Figura 4.2 se observan algunos ejemplos de imágenes del COCO. La principal característica distintiva de entre otros conjuntos de imágenes, es que mantiene un etiquetado individual de cada uno de los objetos de una imagen independientemente si pertenecen a una misma categoría.

Figura 4.2: *Ejemplos de imágenes del conjunto COCO*

Las imágenes seleccionadas para este trabajo son las que tienen una categoría de ave (*bird*). Las

condiciones y tamaño del conjunto de imágenes sometidas en la fase de entrenamiento están detallas en el Capítulo 5 Análisis y Resultados.

4.2. PRE-PROCESAMIENTO DE IMÁGENES

La fase de preprocesamiento tiene como proposito mejorar la calidad de la imagen utilizando técnicas de realzado y suavizamiento de los niveles de intensidad de un determinado modelo de color. La ecualizacion del histograma es la técnica utilizada para realzar los niveles de intensidad de la imagen, la cual esta descrita en la sección 2.1.2 Preprocesamiento.

En la Figura 4.3 se observa un ejemplo del filtro ecualización del histograma. La imagen de la derecha es la imagen original, la cual se observa con pixeles con bajo contraste. La imagen de la izquierda muestra el resultado de realzar la imagen al aplicar el filtro.

Figura 4.3: *Imagen de la base de datos COCO con aplicación del filtro Ecualizacion del Histograma.*

En la Figura 4.4, muestra el resultado de suavizar la imagen con el filtro de la media gaussiana. La media gaussiana permite corregir puntos atípicos en los niveles de intensidad provocados por la aplicación de filtros de realzado.

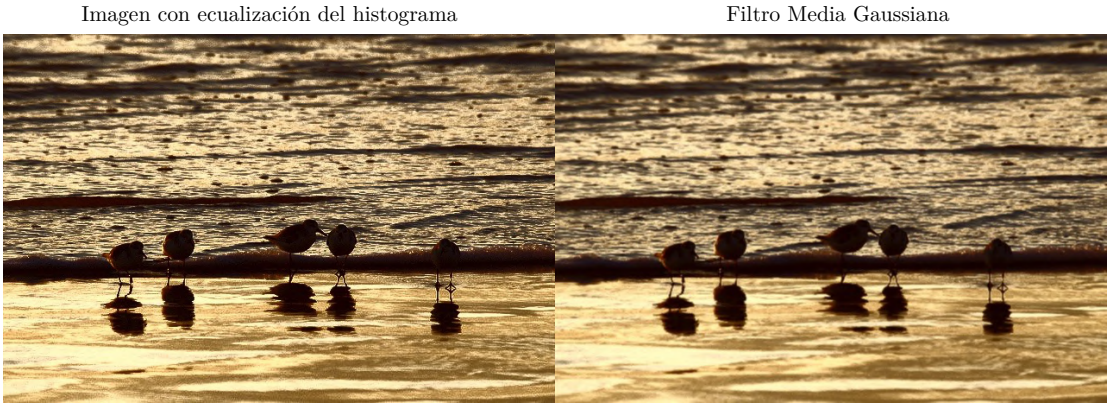

Figura 4.4: *Filtro gaussiano aplicado a una imagen con filtro de realzado.*

4.3. PRESEGMENTACIÓN DE IMÁGENES

Segmentar una imagen permite obtener los diferentes objetos de una escena. Para poder segmentar o extraer una zona de interés la metodología de este trabajo se propone utilizar una Red Convolucional para la detección de la región de interés del objeto (ROI).

La presegmentación tiene como objetivo identificar las coordenadas donde se encuentra el objeto de interés en una imagen con un cuadro delimitador. Hay varios trabajos en la literatura que utilizan cuadros delimitadores para determinar la posición de los objetos de interés, siendo YOLO [35] uno de los métodos más utilizados. YOLO [17] es una red neuronal convolucional para la localización de objetos, es muy rápida para aplicaciones en tiempo real y tuvo varias versiones.

La arquitectura YOLOV3 [36] se puede describir mediante dos procesos principales: un extractor de características llamado Darknet-53 y un método convolucional llamado *SetConvolutional*. La arquitectura de YOLOV3 se puede observar en la Figura 4.5.

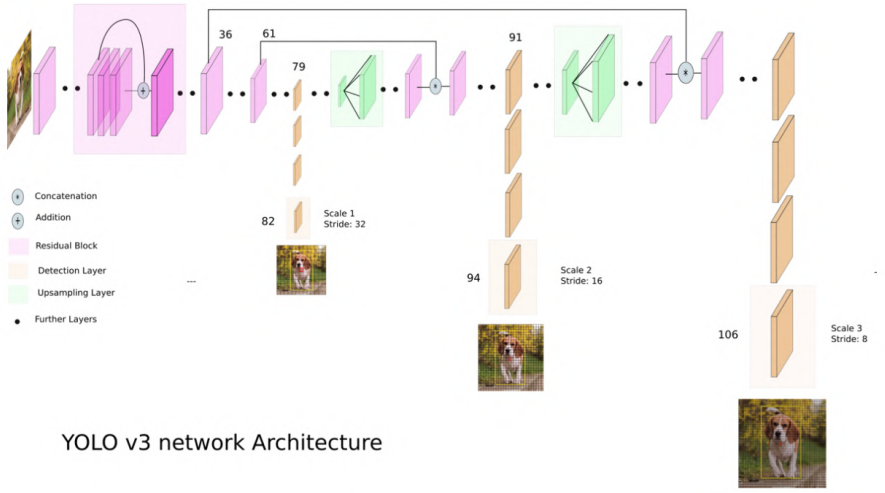

Figura 4.5: *Esquema general de YOLOV3*

Darknet-53 es una CNN de 53 capas de profundidad y es utilizada como extractor de características. Los bloques color rosa de la Figura 4.5 muestran la arquitectura del extractor de características, el cual está compuesto por cinco bloques de capas de convolución. En los bloque de color amarillo, que son capas de convolución se encuentra la información más importante obtenida de esta CNN, la cual se utiliza para extraer tres detecciones en diferentes escalas, estos bloques utilizan el método de transformacion de $SetConvolutional$. La información obtenida en los bloques amarillos de convolución en Dartknet-53 se utiliza para hacer predicciones con diferentes escalas.

YOLOV3 predice un valor objetivo para cada cuadro delimitador mediante regresión logística. La predicción del cuadro delimitador consta de 5 componentes, podemos verlo en la Ecuación 4.1.

$$y = (x_1, y_1, x_2, y_2, confianza) \tag{4.1}$$

donde las coordenadas (x_1, y_1, x_2, y_2) representan el centro del cuadro con respecto a la ubicación de la celda de la cuadrícula. Estas coordenadas se normalizan entre 0 y 1. El valor de confianza indica la probabilidad de que el cuadro contenga un objeto y la precisión del cuadro delimitador. La fase

de pre-segmentación es una etapa crítica, para efectos de comparación debe incluirse en la etapa de resultados.

4.4. SEGMENTACIÓN

La segmentación tiene como objetivo refinar o ajustar la región delimitada por las coordenadas de la pre-segmentación para definir la región de ave y no ave. Proponemos delinear regiones para aves y no aves en base a las coordenadas obtenidas por Pre-segmentación, Figura 4.6-a). El ajuste de las coordenadas de presegmentación se define mediante dos configuraciones:

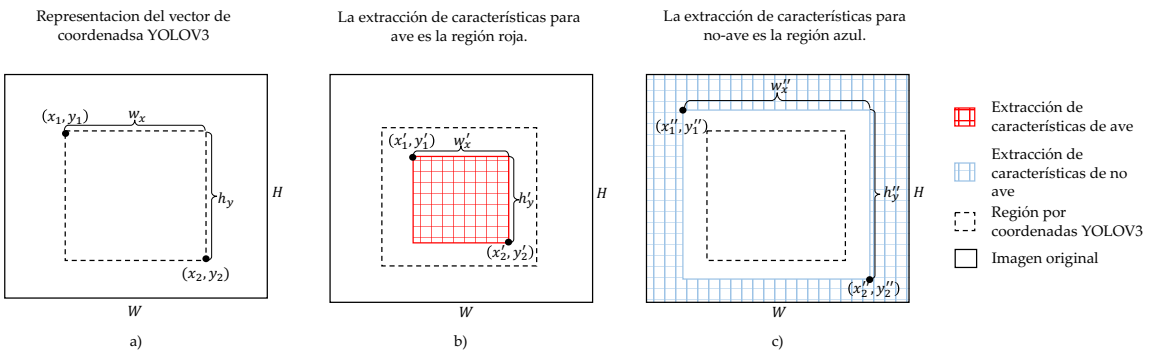

Figura 4.6: *Definición del área de segmentación.*

Configuración 1: Las coordenadas de presegmentación se reducen en un 50 %. Los píxeles dentro del rango de la Configuración 1 se clasifican como aves, Figura 4.6-b). Las coordenadas de la Configuración 1 se describen a continuación:

Dado un vector de coordenadas con valores $[x_1, y_1, x_2, y_2]$, se determina el ancho de la región $w_x = x_2 - x_1$ y la altura de la región $h_y = y_2 - y_1$, y la región del ave se define en las Ecuaciones 4.2:

$$\begin{aligned} x_1^{'} &= x_1 + \frac{w_x}{4} \quad & x_2^{'} &= x_2 - \frac{w_x}{4} \\ y_1^{'} &= y_1 + \frac{h_y}{4} \quad & y_2^{'} &= y_2 - \frac{y_x}{4} \\ w_x^{'} &= \frac{w_x}{2} \quad & h_y^{'} &= \frac{h_y}{2} \end{aligned} \tag{4.2}$$

donde x_1, x_2 son las coordenadas de la imagen de origen $(0,0)$ en el eje horizontal; y_1, y_2 son las coordenadas de la imagen de origen $(0,0)$ en el eje vertical; $x_1^{'}, x_2^{'}$ son las nuevas coordenadas de la imagen de origen $(0,0)$ en el eje horizontal; $y_1^{'}, y_2^{'}$ son las nuevas coordenadas de la imagen de origen $(0,0)$ en el eje vertical.

Configuración 2: Las coordenadas de presegmentación se incrementan en un 20 %. Los píxeles fuera de la Configuración 2 se clasifican como no aves, Figura 4.6-c). Las coordenadas de la Configuración 1 se describen a continuación:

Dado un vector de coordenadas con valores $[x_1, x_2, y_1, y_2]$, se determina el ancho de la región $w_x = x_2 - x_1$, y la altura de la región $h_y = y_2 - y_1$, la región no ave se define en las Ecuaciones 4.3:

$$\begin{aligned} x_1^{'} &= x_1 - \frac{w_x}{8} \quad & x_2^{'} &= x_2 + \frac{w_x}{8} \\ y_1^{'} &= y_1 - \frac{h_y}{8} \quad & y_2^{'} &= y_2 + \frac{y_x}{8} \\ w_x^{''} &= w_x + \frac{w_x}{4} \quad & h_y^{'} &= h_y + \frac{h_y}{4} \end{aligned} \tag{4.3}$$

Los píxeles entre las regiones de aves y no aves no se consideran en la fase de extracción de características. La etiqueta de un vector de características se asigna de acuerdo con la región en la que se encuentra.

4.5. EXTRACCIÓN DE CARACTERÍSTICAS DEL ROI

Dado un conjunto de imágenes suavizadas y realzadas se procede a la extracción de características de Color y Textura. Las características de Color hacen referencia al comportamiento estadístico de cada uno de los canales de los modelos de color. Los modelos de color fueron elegidos de acuerdo al estado de arte y estos son: HSI, CMYK, LAB y XYZ. La varianza y desviación estándar son las características a extraer para cada canal.

Sin embargo, el procedimiento tiene incorporado una técnica de *pooling* la cual divide la imagen en ventanas del tamaño especificado. Para este trabajo se utilizo un *pooling* de 3×3 pixeles, Figura 4.7. Dada una imagen a color $I(x,y)$ se divide la imagen en ventanas de tamaño de 3×3 pixeles, a esta ventana se le extraen las correspondientes características de color y textura. El conjunto de ventanas obtenidas dan lugar a la imagen original.

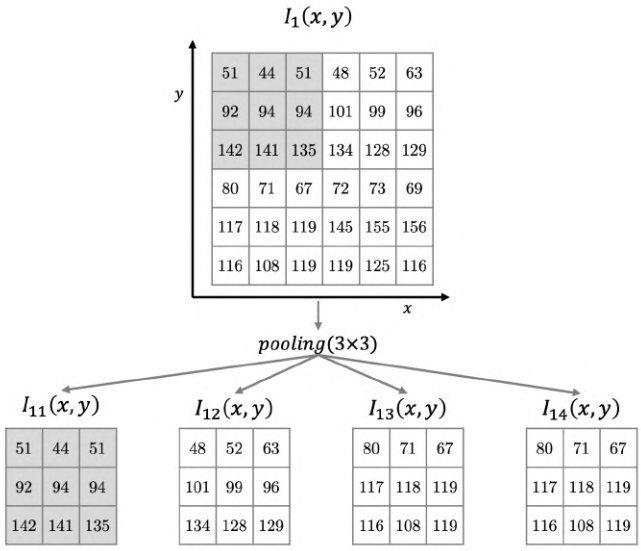

Figura 4.7: *Division de la representación de una imagen en ventanas de* 3×3 *pixeles*

Cada ventana fue sometida a la extracción de las características correspondientes a color, dando lugar a un total de 26 características a fines al color. El proceso se puede apreciar en la Figura 4.8.

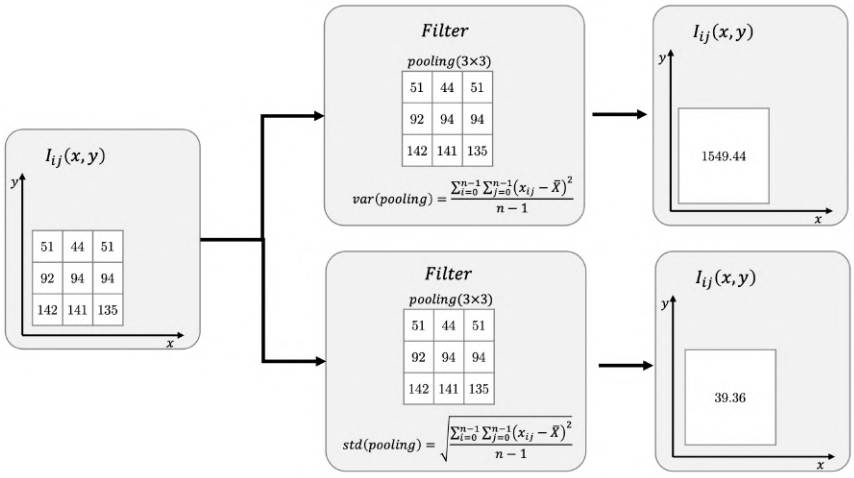

Figura 4.8: *Extracción de características de varianza y desviación estándar correspondientes a color en una ventanas $I_{ij}(x,y)$ de 3×3 pixeles*

Las características de textura de Haralick [1] son descriptores de textura comunes en el análisis de imágenes basados en que la textura y el tono están relacionados. Las características se determinan utilizando una matriz de correlación de los niveles de intensidad de una imagen, GLCM. El número de niveles de gris en la imagen determina el tamaño del GLCM. La Figura X muestra un ejemplo de cómo se determina el GLCM [37].

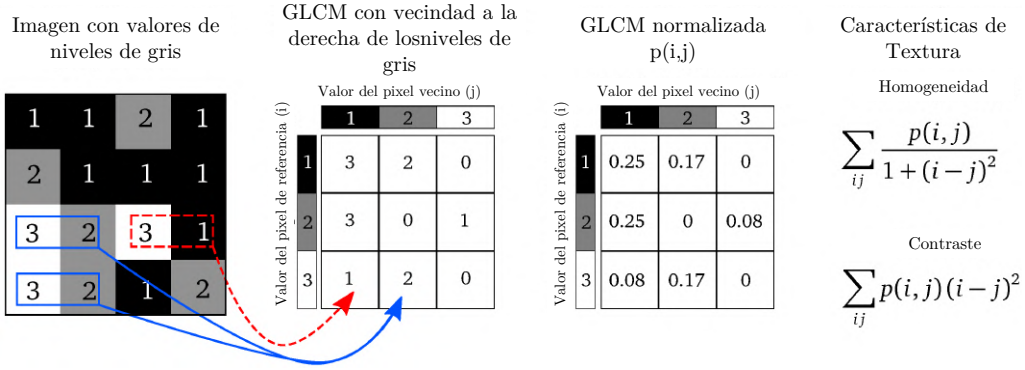

Figura 4.9: *Procedimiento para aplicar características de Textura en GLCM .*

Las características de Textura extraídas mediante el procedimiento descrito en el Marco Teórico se muestran en la Tabla 4.1. Estas 17 características se extrajeron de las ventanas de cada imagen. Mediante

este procedimiento se obtuvieron 43 características pertenecientes a color y textura en cada una de las ventanas de una imagen. Hasta este punto, se puede deducir que las observaciones representan información a nivel de ventanas.

Tabla 4.1: *Características de Textura Haralick utilizadas en este trabajo de tesis .*

Num.	Nombre Característica	Ecuación
1	Autocorrelation [38]	$\sum_{i=1}^{N}\sum_{j=1}^{N}(i\cdot j)p(i,j)$
2	Cluster prominence [1]	$\sum_{i=1}^{N}\sum_{j=1}^{N}(i+j-2\mu)^3 p(i,j)$
3	Cluster shadow [1]	$\sum_{i=1}^{N}\sum_{j=1}^{N}(i+j-2\mu)^4 p(i,j)$
4	Constrast [1]	$\sum_{i=1}^{N}\sum_{j=1}^{N}(i-j)^2 p(i,j)$
5	Correlation [1]	$\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{(i\cdot j)p(i,j)-\mu_x\mu_y}{\sigma_x\sigma_y}$
6	Difference entropy [1]	$-\sum_{k=0}^{N-1} p_{x-y}(k)logp(k)$
7	Difference variance [1]	$\sum_{k=0}^{N-1}(k-\mu_{x-y})^2 p_{x-y}(k)$
8	Dissimilarity [1]	$\sum_{i=1}^{N}\sum_{j=1}^{N}\lvert i-j\rvert\cdot p(i,j)$
9	Energy [1]	$\sum_{i=1}^{N}\sum_{j=1}^{N}p(i,j)^2$
10	Entropy [1]	$\sum_{i=1}^{N}\sum_{j=1}^{N}p(i,j)logp(i,j)$
11	Homogeneity [38]	$\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{p(i,j)}{1+(i+j)^2}$
12	Inverse difference [39]	$\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{p(i,j)}{1+\lvert i-j\rvert}$
13	Maximum probability [1]	$maxp(i,j)$
14	Sum Average μ_{x+y} [1]	$\sum_{k=2}^{2N} kp_{x+y}(k)$
15	Sum entropy [1]	$-\sum_{k=2}^{2N} p_{x+y}(k)logp_{x+y}(k)$
16	Sum square [1]	$\sum_{i=1}^{N}\sum_{j=1}^{N}(i-\mu)^2 p(i,j)$
17	Sum variance [1]	$\sum_{k=2}^{2N}(k-\mu_{x+y})^2 p_{x+y}(k)$

En base a lo anterior, cada observación es transformada para caracterizar a una imagen. Para esto se aplicaron medidas de tendencia central como la media aritmética, con el fin de promediar las características de todas las ventanas pertenecientes a una imagen.

4.6. SELECCIÓN DE VARIABLES ÓPTIMA

Las características extraídas en una imagen a menudo no representan la información de los píxeles de la mejor manera. Por esto, es necesario aplicar técnicas de reducción de dimensionalidad con el objetivo de determinar un conjunto de variables que permita representar eficiente y eficazmente la información de los píxeles en comparación con el conjunto original de variables.

Por lo anterior, se propone utilizar la metaheurística SA para la selección de un subconjunto de variables a partir del conjunto original, Algoritmo 6. Las aportaciones realizadas están ubicadas en la solución inicial, el método de perturbación, la función objetivo y en los criterios de convergencia.

- Solución inicial: utilizando la técnica de PCA, se evaluó la relevancia de las variables de Color y Textura. A partir de la componente con mayor porcentaje de varianza explicada, las variables que obtuvieron un porcentaje igual o mayor al 70 % de contribución fueron seleccionadas como solución inicial. Linea 2 del Algoritmo 6.
- Perturbación de una solución: esta compuesta por una ruleta. Dicha función aporta pesos de manera individual a las variables que hayan sido parte de una buena solución. Los pesos asignados proveen mayor probabilidad de ser seleccionados.
- Función objetivo: el modelo de Random forest evalúa el desempeño del nuevo conjunto de soluciones, linea 4 del Algoritmo 6.
- Criterios de convergencia: Algoritmo 6 contiene 3. El primer criterio de convergencia radica en alcanzar el equilibrio térmico en el ciclo de metropolis, linea 15. El segundo criterio de

convergencia esta dado al llegar al 95 % de la temperatura final, linea 33. Finalmente el tercer criterio es alcanzando una solución igual o menor al parámetro de épsilon ε linea 12.

Algorithm 4 Recocido Simulado propuesto

0: **function** SIMULATEDANNEALING($T_i, T_f, \beta, \alpha, L_{max}$)
0: $X_{old} \leftarrow solution()$
0: $X_{best} \leftarrow X_i$
0: $E_{old} \leftarrow objFunction()$
0: $E_{best} \leftarrow E_{old}$
0: **if** $E_{best} \neq 0$ **then**
0: **while** $T_i > T_f$ And $\neg converge$ **do**
0: **while** $L < L_{max}$ And $\neg converge$ **do**
0: $X_{new} \leftarrow perturbation_{roullete}(X_{old})$
0: $E_{new} \leftarrow objFunction(X_{new})$
0: $\Delta E \leftarrow E_{new} - E_{old}$
0: **if** $E_{new} = \varepsilon$ **then**
0: $converge$
0: **end if**
0: **if** $converge(metropoly)$ **then**
0: $converge$
0: **end if**
0: **if** $\Delta E \leq 0$ **then**
0: $X_{old} \leftarrow X_{new}$
0: $E_{old} \leftarrow E_{new}$
0: **if** $E_{old} < E_{best}$ **then**
0: $X_{best} \leftarrow X_{old}$
0: $E_{best} \leftarrow E_{old}$
0: **end if**
0: **else if** RANDOM$(0,1) > e^{-\Delta E/T_i}$ **then**
0: $X_{old} \leftarrow X_{new}$
0: $E_{old} \leftarrow E_{new}$
0: **end if**
0: $T_i \leftarrow \alpha T_i$
0: $L_{max} \leftarrow \beta L_{max}$
0:
0: **end while**
0: **if** $T_i \geq 0.95 T_f$ **then**
0: **if** $converge(Temp)$ **then**
0: $converge$
0: **end if**
0: **end if**
0:
0: **end whilereturn** X_{best}, E_{best}
0: **end if**
0: **end function**=0

El algoritmo TA en el Algoritmo 5, al igual que SA, contiene 3 criterios de convergencia y perturba una solución empleando una función de ruleta. El primer criterio de convergencia radica en alcanzar el equilibrio térmico en el ciclo de metropolis, linea 15. El segundo criterio de convergencia esta dado al llegar al 95 % de la temperatura final, linea 30. Finalmente el tercer criterio es alcanzando una solución igual o menor al parámetro de épsilon ε linea 12.

Algorithm 5 Aceptación de umbral propuesto

```
0: function THRESHOLDACCEPTING(T_i, T_f, β, α, L_max)
0:     X_old ← solution()
0:     X_best ← X_i
0:     E_old ← objFunction()
0:     E_best ← E_old
0:     if E_best ≠ 0 then
0:         while T_i > T_f And ¬converge do
0:             while L < L_max And ¬converge do
0:                 X_new ← perturbation_roullete(X_old)
0:                 E_new ← objFunction(X_new)
0:                 ΔE ← E_new − E_old
0:                 if E_new = ε then
0:                     converge
0:                 end if
0:                 if converge(metropoly) then
0:                     converge
0:                 end if
0:                 if ΔE ≤ T_i then
0:                     X_old ← X_new
0:                     E_old ← E_new
0:                     if E_old < E_best then
0:                         X_best ← X_old
0:                         E_best ← E_old
0:                     end if
0:                 end if
0:                 T_i ← αT_i
0:                 L_max ← βL_max
0:
0:             end while
0:             if T_i ≥ 0.95T_f then
0:                 if converge(Temp) then
0:                     converge
0:                 end if
0:             end if
0:
0:         end whilereturn X_best, E_best
0:     end if
0: end function=0
```

4.7. CLASIFICACIÓN DEL OBJETO DE INTERÉS

Se utilizó un Random Forest como procedimiento de clasificación. Las observaciones utilizadas se componen de dos tamaños de vectores. El primero consta de 43 funciones y una etiqueta, y el segundo de 14 funciones y una etiqueta. Este último se obtiene a través de la fase de selección de las variables relevantes. Random Forest se sometió a un proceso de ajuste que incluía una búsqueda aleatoria de hiperparámetros. Los parámetros del mejor modelo encontrado usando este esquema de ajuste están en la configuración Experimental.

4.8. EVALUACIÓN

La evaluación del desemepeño esta basado en los criterios de evaluación utilizados en las competencias de la base de datos COCO [4]. La métrica utilizada para evaluar el rendimiento del modelo es Intersección de precisión promedio sobre unión (APIoU), que se muestra en la Ecuación 4.4.

$$APIoU = \sum_{i=1}^{m} \frac{TP_i}{FP_i + TP_i} \tag{4.4}$$

donde m es el número de imágenes, TP son los verdaderos positivos y FP son los falsos positivos para la imagen i. La métrica $APIoU$ estuvo sujeta bajo dos umbrales. El primer umbral $APIoU$ es de 0,05 a 0,95 el cual muestra el promedio de las ventanas de 15×15 píxeles correctamente clasificadas; mientras que el segundo umbral $APIoU$ es de 0,75 a 0,95 llamado $APIoU^{75}$.

CAPÍTULO 5

Análisis y resultados

En este capítulo se concentra el análisis de los resultado obtenidos de la serie de experimentaciones realizadas en este trabajo de tesis, las cuales son:

- Experimentación 1: Instancia con 129 características y ventana de 3×3 píxeles
- Experimentación 2: Instancia con 54 características y ventana de 3×3 píxeles
- Experimentación 3: Instancia con 47 características y ventana de 15×15 píxeles

Así mismo se describen las condiciones experimentales, material y equipo utilizado para realizarlas. El algoritmo SA es utilizado como selector de variables relevantes. El proceso de sintonización de SA esta sometido a una ejecución de 30 veces. Los valores obtenidos de cada corrida son utilizados para obtener los valores delta máximos y mínimos. Las ecuaciones 5.1 y 5.2 son utilizados para obtener los parámetros delta máximo ΔZ_{max} y delta mínimos ΔZ_{min} .

$$\sigma_{\Delta E_{min}} = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{\Delta E_{min}})^2} \qquad ;\sigma_{\Delta E_{max}} = \sqrt{\frac{1}{N-1}\sum_{j=1}^{N}(x_j - \overline{\Delta E_{max}})^2}$$

$$\Delta Z_{min} = \overline{\Delta E_{min}} + 3\sigma_{\Delta E_{min}} \tag{5.1}$$

$$\Delta Z_{max} = \overline{\Delta E_{max}} + 3\sigma_{\Delta E_{max}} \tag{5.2}$$

donde N es el número de corridas, x_i es el valor i-esimo del conjunto de los mínimos, x_j es el valor i-esimo del conjunto de los máximos, $\overline{\Delta E_{max}}$ es el promedio de los valores mínimos, $\overline{\Delta E_{max}}$ es el promedio de los valores máximos.

5.1. EXPERIMENTACIÓN 1

5.1.1. Condiciones experimentales

El conjunto de imágenes fueron seleccionadas del conjunto COCO [4] siendo un total de 434 imágenes, donde 217 imágenes tienen objetos del tipo de ave y 217 imagenes tienen otro tipo de objetos, las cuales son: autobús, carro, bicicleta y mochila. La extracción de características se realizo bajo una vecindad o ventana con dimensiones de 3x3 píxeles. En cada ventana se extrajeron 2 variables de Color y 17 de Textura. Las variables de color son la varianza y la desviación estándar.

La extracción de características fueron sobre la estación de trabajo del LANTI con procesador Intel Xeon 8cores y memoria de 32 GB. El lenguaje programación utilizado fue python 3.7.1 junto con las

librerías Opencv 3.4.2 y Scikit-learn 0.21. Los algoritmos de SA y TA son empleados como selector de variables relevantes y ejecutados en el clsuter LANTI.

Así mismo, K-means es utilizado como algoritmo de agrupamiento. El número óptimo de grupos fue obtenido aplicando las métricas Elbow, GAP y Silhoutte. El rango de grupos definido es $k = 3 - 10$. El clasificador es SVM con un kernel lineal con un costo $C = 2.05$.

5.1.2. Sintonización

En la Tabla 5.1 se observan los parámetros obtenidos tras 30 corridas del algoritmo básico de SA. Los valores obtenidos son utilizados para configurar los párametros de los algoritmos propuestos de SA y TA.

Tabla 5.1: *Parámetros de sintonización para una instancia de 129 características y una ventana de* 3×3.

Total de Corridas	5	10	15	20	25	30
ΔZ_{min}	0.230415	0.230415	0.230415	0.230415	0.230415	**0.230415**
ΔZ_{max}	57.33161	56.91602	58.98118	58.827	60.37675	**60.8626**
T_i, T_0	5704.447	5663.096	5868.578	5853.237	6007.436	**6055.778**
T_f, T_n	0.050034	0.050034	0.050034	0.050034	0.050034	**0.050034**
n	227.0093	226.8675	227.5623	227.5113	228.0183	**228.1745**
L_{max}	594.067	594.067	594.067	594.067	594.067	**594.067**
β	1.028535	1.028553	1.028465	1.028471	1.028407	**1.028387**

5.1.3. Resultados

En la Tabla 5.2 se observan el conjunto de características seleccionadas por las estrategias SA y TA propuestas. La primera columna enumera cada una de las características. Las columnas del 2 a 5 enlistan las características obtenidas por los algoritmo SA y TA con un una configuración de un $\varepsilon = 0$ y $\varepsilon = 5$. Todas las configuraciones tiene definido una solución inicial S_i por PCA. El rango de características obtenidas es de 21-34. Las características con un $\varepsilon = 0$ corresponden a una solución S_i con un $0\,\%$ de error.

Tabla 5.2: *Características seleccionadas por las estrategias propuestas de SA y TA para una instancia de 129 características y una ventana de* 3×3.

Num	SAE0SiPCA129F_21f	SAE5SiPCA129F_22f	TAE0SiPCA129F_35f	TAE5SiPCA129F_28f
1	std_S_2	std_H_2	std_H_2	std_S_1
2	std_S_3	var_I_1	std_S_3	std_S_2
3	var_H_1	std_K_3	std_I_2	std_S_3
4	var_H_2	var_C_1	std_I_3	std_I_1
5	var_S_2	var_K_3	var_H_2	var_S_2
6	std_C_2	std_A_3	var_S_3	var_S_3
7	var_C_1	std_B_lab_3	std_M_1	std_C_1
8	var_C_2	var_L_3	std_K_1	std_C_2
9	var_K_1	var_A_1	var_Y_xyz_3	std_Y_xyz_2
10	var_K_3	var_A_2	std_B_lab_1	std_Y_xyz_3
11	std_L_3	std_Z_3	std_B_lab_2	var_Y_xyz_3
12	var_L_2	var_Y_xyz_1.1	std_B_lab_3	var_K_3
13	std_Y_xyz.1_3	var_Y_xyz_3.1	std_X_1	std_L_3
14	var_X_2	correlation_2	std_X_3	var_L_2
15	entropy_1	correlation_3	std_Z_2	std_X_2
16	correlation_1	dissimilarity_1	var_X_1	std_X_3
17	energy_2	homogeneity_3	var_Y_xyz_1.1	std_Y_xyz.1_3
18	homogeneity_1	difference_entropy_3	var_Y_xyz_2.1	std_Z_3
19	cluster_prominente_2	maximun_probability_1	var_Y_xyz_3.1	var_X_2
20	sum_of_square_1	sum_of_square_1	var_Z_3	var_X_3
21	sum_variance_2	sum_of_square_2	contrast_3	var_Y_xyz_2.1
22		sum_average_2	correlation_2	var_Y_xyz_3.1
23			energy_1	entropy_2
24			energy_2	contrast_2
25			ASM_1	dissimilarity_2
26			ASM_3	difference_entropy_1
27			cluster_prominente_1	difference_entropy_3
28			difference_variance_1	difference_variance_2
29			inverse_difference_1	
30			maximun_probability_1	
31			sum_of_square_2	
32			sum_of_square_3	
33			sum_average_2	
34			sum_average_3	
35			sum_variance_1	

Los resultados en la Tabla 5.2 muestran una seleccion de variables del tipo Color mas alta en comparación de las del tipo Textura. Cada uno de los conjuntos seleccionados por las estrategias fueron agrupados con el algoritmo de k-means y entrenados con SVM lineal con el fin de observar el grado de aprendizaje que obtuvieron al someterlas a casos de prueba.

5.1.4. Predicción

Los resultados de predicción una región 3×3 píxeles bajo las diferentes estrategias se muestran en la Figura 5.1. La primera fila de imagenes son el resultados de la estrategia SA empleando una perturbación con ruleta. Mientras que en la segunda fila de imagenes se emplea una perturbación básica. Ambos métodos tienen el $10\,\%$ de perturbación en una solución S_i. La tercera y cuarta fila tienen la misma descripción anterior pero tienen un $\varepsilon = 5$. La primera columna tiene como referencia la imagen original.

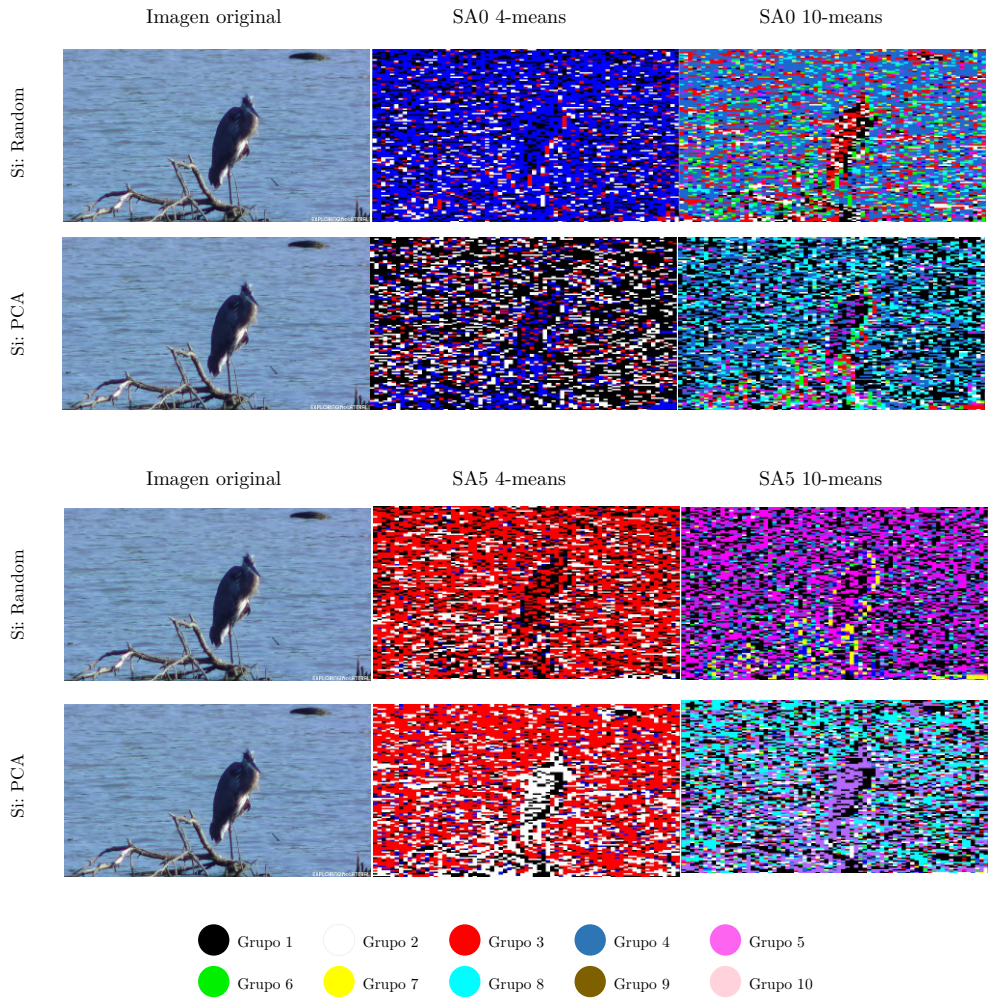

Figura 5.1: *Caso de prueba al predecir segmentos con las características de la estrategia SA con $\varepsilon = 0$ y $\varepsilon = 5$.*

Cada grupo o tipo de píxeles esta agrupado por color, con esto es posible observar la ubicación de las ventanas con el grupo al que fue predicho.

En la Figura 5.2 se muestra un formato similar al descrito en la Figura 5.1. Sin embargo, mediante inspección visual se observa que los resultados de TA son de menor calidad de predicción comparado con SA.

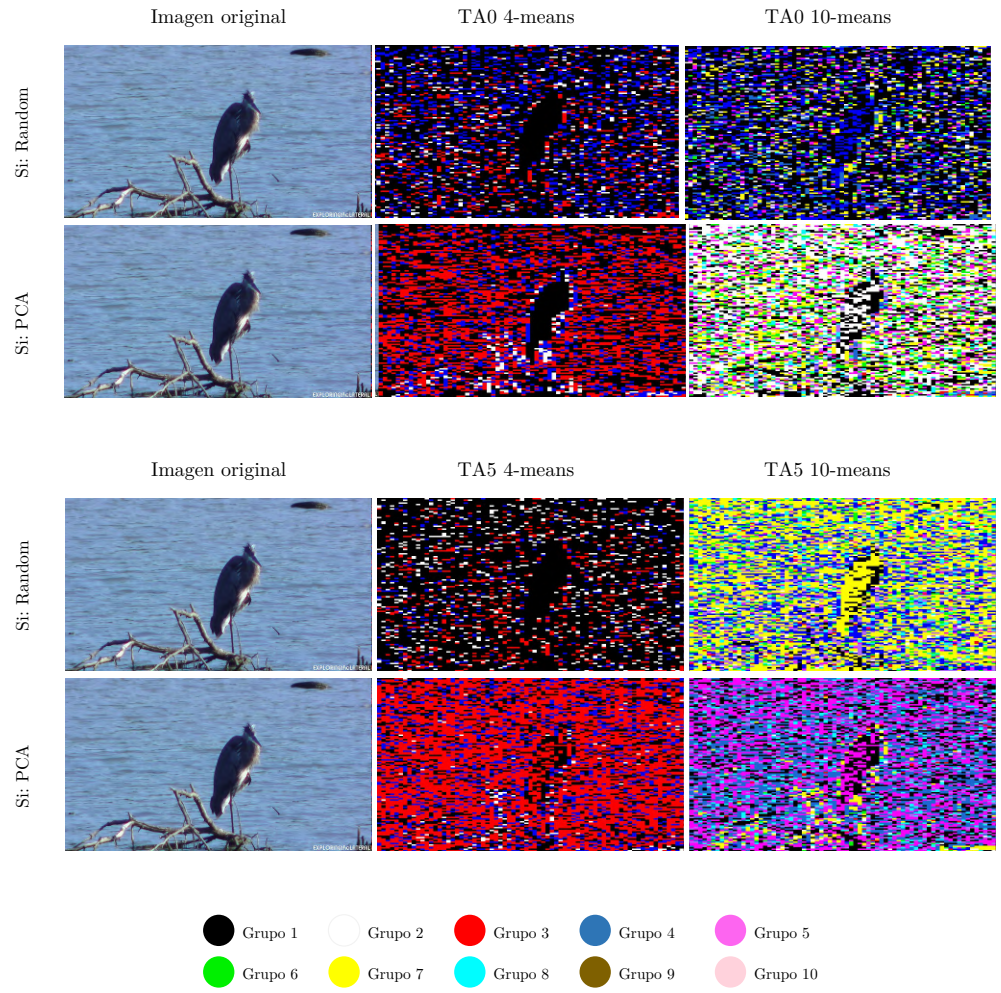

Figura 5.2: *Caso de prueba al predecir segmentos con las características de la estrategia TA con* $\varepsilon = 0$ *y* $\varepsilon = 5$.

5.2. EXPERIMENTACIÓN 2

5.2.1. Condiciones experimentales

Los resultados presentados son utilizando una instancia de 43 características correspondientes a Color y textura. La extracción de características se realizo utilizando un equipo de computo con un sistema operativo de macOS y un procesador M1. A continuación se presenta la configuración para detección del ROI, la extracción de características y selección de variables relevantes:

- Detección del ROI: YOLOV3 bajo el lenguaje de programación C++ utilizando la red convolucional Darknet53 [36].
- Extracción de características: python 3.7.1 junto con las librerías Opencv 3.4.2 y Scikit-learn 0.21.
- Selección de variables óptimas: R 4.0.4 junto con las librerías MASS, e1071, hash, dplyr.

El conjunto de imágenes fueron seleccionadas del conjunto COCO [4] siendo un total de 30 imágenes conteniendo una única AVE. El número de instancias obtenidas de las 30 imágenes es de 805,033 vectores de características, siendo 24,151(3 %) vectores correspondientes a AVES y 780,882 (97 %) a NOAVES. Para esta experimentación se utilizan 4,000 vectores de características por clase (AVES y NOAVES). La extracción de características se realizo bajo una vecindad o ventana con dimensiones de 3x3 píxeles.

5.2.2. Sintonización

En la Tabla 5.3 se observan los parámetros obtenidos tras realizar 30 corridas del algoritmo básico de SA. Los valores obtenidos son utilizados para configurar los párametros del algoritmos propuesto de

SA.

Tabla 5.3: *Parámetros de sintonización obtenidos tras 30 corridas para una instancia con 43 características y una ventana de* 3×3.

Corridas	5	10	15	20	25	30
ΔZ_{min}	0.05	0.05	0.05	0.05	0.05	0.05
ΔZ_{max}	12.4623476	16.7608372	15.6776212	15.7418525	15.4465434	**15.7819646**
T_i, T_0	1239.99314	1667.68926	1559.91018	1566.30114	1536.91813	**1570.29226**
T_f, T_n	0.01085736	0.01085736	0.01085736	0.01085736	0.01085736	**0.01085736**
n	227.042797	232.820027	231.517503	231.597214	231.22801	**231.646828**
L_{max}	198.022318	198.022318	198.022318	198.022318	198.022318	**198.022318**
β	1.02356582	1.02297439	1.02310513	1.02309708	1.02313439	**1.02309208**

5.2.3. Resultados

En la Figura 5.3 se observan el conjunto de características seleccionadas por la estrategia SA propuesta. En promedio se seleccionan 25 características para la estrategia sin una ruleta en la perturbación dela solución y 19 características aplicando la ruleta para la perturbación de las soluciones. Las características con un $\varepsilon = 0$ corresponden a una solución S_i con un 0 % de error. También se pude observar que en todas ellas existen características de color y textura.

Strategy	SA without Roullete				SA with Roullete			
Si	SiRandom		SiPCA		SiRandom		SiPCA	
Epsilón	0	5	0	5	0	5	0	5
1	std_S	std_S	std_I	std_S	std_S	std_H	var_H	var_S
2	var_S	std_I	var_H	std_I	std_I	var_S	var_I	var_I
3	std_C	var_H	std_C	var_H	var_H	var_I	std_C	std_M
4	std_M	var_I	std_M	std_C	var_S	std_C	std_M	std_Y_xyz
5	std_Y_xyz	std_C	std_Y_xyz	std_M	std_C	std_M	std_Y_xyz	std_K
6	var_C	std_M	var_C	std_K	std_M	std_Y_xyz	var_M	var_M
7	var_M	std_K	var_M	var_C	std_K	std_K	std_L	var_L
8	var_Y_xyz	var_C	var_Y_xyz	var_M	var_C	var_C	std_A	var_B_lab
9	var_K	var_K	std_L	var_K	var_M	var_M	std_B_lab	std_X
10	std_A	std_L	var_B_lab	std_L	var_K	std_L	var_A	std_Z
11	var_L	std_B_lab	std_Y_xyz.1	std_A	std_Y_xyz.1	std_A	var_B_lab	var_X
12	contrast	var_B_lab	var_X	std_B_lab	std_Z	std_B_lab	var_X	cluster_prominente
13	energy	std_Y_xyz.1	var_Y_xyz.1	var_L	var_Z	var_L	entropy	sum_of_square
14	dissimilarity	var_X	var_Z	var_B_lab	entropy	var_A	energy	difference_entropy
15	cluster_shade	var_Y_xyz.1	contrast	std_Y_xyz.1	contrast	std_Y_xyz.1	difference_entropy	sum_average
16	sum_of_square	var_Z	correlation	std_Z	homogeneity	var_Y_xyz.1	difference_variance	
17	sum_variance	entropy	energy	var_X	cluster_prominente	homogeneity	maximun_probability	
18	sum_entropy	contrast	cluster_prominente	var_Y_xyz.1	cluster_shade	cluster_prominente		
19		correlation	cluster_shade	var_Z	difference_variance	cluster_shade		
20		energy	inverse_difference	contrast	sum_average	sum_of_square		
21		dissimilarity		correlation		difference_entropy		
22		cluster_prominente	difference_entropy	cluster_shade		difference_variance		
23		inverse_difference	maximun_probability	inverse_difference		maximun_probability		
24		sum_of_square	sum_average	sum_of_square		sum_average		
25		difference_entropy	sum_variance	difference_entropy		sum_variance		
26		maximun_probability	sum_entropy	maximun_probability				
27		sum_average		sum_average				
28		sum_variance						
29		sum_entropy						

Figura 5.3: *Características seleccionadas por la estrategia SA propuesta para una instancia de 43 características y una ventana de* 3×3.

5.3. EXPERIMENTACIÓN 3

5.3.1. Condiciones experimentales

Para los resultados presentados se utilizo una instancia de 43 características correspondientes a Color y textura. La extracción de características fue sobre la estación de trabajo del LANTI con procesador Intel Xeon 8cores y memoria de 32 GB. El lenguaje programación utilizado fue python 3.7.1 junto con las librerías Opencv 3.4.2 y Scikit-learn 0.21.

El conjunto de imágenes fue obtenida de la base de datos COCO [4] siendo un total de 260 imágenes con categoría ave. Las imágenes fueron clasificadas en grandes y medianas. Para el entrenamiento fueron usadas 190 imágenes y para prueba 70 imágenes, siendo en esta última 35 imágenes por categoría grande y mediana. La extracción de características se realizo bajo una vecindad o ventana con dimensiones de 15x15 píxeles.

La selección de características fue realizada por el algoritmo propuesto de SA, con una solución inicial proporcionada por PCA, aplicación de una ruleta en el proceso de perturbación y utilizando *Random forest* como función objetivo.

5.3.2. Sintonización

En la Tabla 5.4 se observan los parámetros obtenidos del algoritmo de sintonización de SA. Los parámetros fueron calculados por medio de las corridas acumuladas en bloques de 5. Finalmente se utilizaron los valores obtenidos tras realizar 30 corridas.

Tabla 5.4: *Parámetros de sintonización obtenidos tras 30 corridas para una instancia de 43 características y una ventana de* 15×15.

Corridas	5	10	15	20	25	30
ΔZ_{min}	0.05	0.05	0.05	0.05	0.05	0.05
ΔZ_{max}	12.4623476	16.7608372	15.6776212	15.7418525	15.4465434	**15.7819646**
T_i, T_0	1239.99314	1667.68926	1559.91018	1566.30114	1536.91813	**1570.29226**
T_f, T_n	0.01085736	0.01085736	0.01085736	0.01085736	0.01085736	**0.01085736**
n	227.042797	232.820027	231.517503	231.597214	231.22801	**231.646828**
L_{max}	198.022318	198.022318	198.022318	198.022318	198.022318	**198.022318**
β	1.02356582	1.02297439	1.02310513	1.02309708	1.02313439	**1.02309208**

5.3.3. Resultados

El algoritmo de SA proporcionó 14 variables relevantes, los cuales se presentan en la Tabla 5.5. La primera columna representa el número de variables. La segunda columna el tipo de variable a la que pertenece. La tercera columna, en caso de ser la variable del tipo COLOR, indica el canal de modelo de color seleccionado. Finalmente, la cuarta columna indica el nombre de la variable en caso de ser Textura; en caso de ser de Color, indica el tipo de medida de tendencia central.

Tabla 5.5: *Variables relevantes seleccionadas por SA para una instancia de 43 características y una ventana de* 15×15.

Núm	Tipo	Canal	Variable
1	Color	H	Desviación estandar
2		S	Desviación estandar
3		I	Varianza
4		M	Desviación estandar
5		C	Varianza
6		K	Varianza
7		L	Desviación estandar
8		A	Desviación estandar
9		A	Varianza
10		Y_{XYZ}	Varianza
11		Z	Varianza
12	Textura		Correlation
13		-	Difference entropy
14			Difference variance

Las variables de Color representan el 80 % del conjunto, siendo las características de Textura el 30 % restante. Las características relevantes fueron seleccionadas para el entrenamiento del clasificador *Random Forest* con el fin de observar el grado de aprendizaje que obtuvieron en comparación del conjunto original.

Los resultados obtenidos, para dos grupos de tamaño de ave Grande y Mediano, se dan en la Tabla 4. El desempeño de la metodología propuesta PSEV-BF es comparado con diferentes configuraciones: M1, M2, M3. Donde M1, aplica el proceso tradicional de Preprocesamiento, Clasificación, Evaluación y una técnica de súper píxeles. M2, incluye el mismo proceso de M1 pero no el uso de superpíxeles aunque se implementa un método de selección de variables. M3 solo se implementa una fase de presegmentaciones con YOLOV3.Finalmente, nuestro PSEV-BF propuesto incluye todas las configuraciones propuestas en este documento. Es importante aclarar que todos los métodos utilizan la presegmentación con YOLOV3

con fines comparativos.

En la Tabla 4, la primera columna indica el modelo utilizado: Random Forest Tuned. La primera columna indica el tamaño de las aves utilizadas: Grande o Mediano. En la segunda columna, indicamos las diferentes metodologías con las denominaciones M1, M2, M3 y Nuestra Propuesta. La tercera y cuarta columnas indican con si utilizamos la técnica de superpíxeles y en caso contrario en las fases de Presegmentación, Segmentación o Mejora de Características. Finalmente, las dos últimas columnas son los resultados de Precisión Promedio de Intersección sobre Unión con dos umbrales: 0.5 a 0.95 y 0.75 a 0.95.

Tabla 5.6: *Comparativa del desempeño de las diferentes metodologías.*

Tamaño objeto	Método	Super pixeles	Pre-segmentación	Segmentación	Mejora Características	Métrica APIoU	Métrica APIoU^{75}
Grande	PSEV-BF	✓	✓	✓	✓	**0.5485**	**0.8614**
	M1	✓	✓	✗	✗	0.5256	0.8235
	M2	✗	✓	✓	✓	0.5481	0.8347
	M3	✗	✓	✗	✗	0.4133	0.8459
Mediano	PSEV-BF	✓	✓	✓	✓	**0.3613**	**0.8097**
	M1	✓	✓	✗	✗	0.3264	0
	M2	✗	✓	✓	✓	0.3325	**0.8097**
	M3	✗	✓	✗	✗	0.3483	**0.8097**

Para objetos grandes, PSEV-BF y M2 muestran valores de alrededor del 54 % de precisión, mientras que M2 no tiene la fase de superpíxel. Primero, M1 y M2 utilizan al menos dos de los métodos propuestos en la metodología. Mientras que M3 no utiliza las fases propuestas, resultando en un 41 % de precisión, que es el valor más bajo entre las metodologías comparadas. En segundo lugar, M2 no utiliza el método de superpíxel, lo que conduce a un valor de precisión muy similar en comparación con PSEV-BF, mientras que M1 tiene una diferencia del 2 % en comparación con M3. Podemos decir que el uso de los procesos propuestos para objetos grandes mejora la precisión de la metodología.

Para objetos de tamaño mediano, la metodología de PSEV-BF muestra valores en torno al 36 % de precisión. Primero, M1 muestra una precisión del 32 %, que es el valor más bajo entre las metodologías

comparadas. Esto significa que los efectos son tan grandes cuando no se utilizan la segmentación y las variables mejoradas. PSEV-BF y M1 difieren en un 4 %, la diferencia se debe al uso de un método de superpíxel. Encontramos que el uso de los procedimientos propuestos en objetos de tamaño mediano mejora la precisión de la metodología.

5.3.4. Predicción

Los resultados de la predicción de una región bajo las diferentes estrategias se muestran a continuación. La Figura 5.4 muestra algunos ejemplos exitosos. La Figura 5.4-a) muestra las imágenes en su estado original; La Figura 5.4-b) muestra las imágenes segmentadas por COCO; y la Figura 5.4-c) muestra la adaptación resultante de la fase de segmentación. Finalmente, la Figura 5.4-d) muestra algunos de los casos obtenidos utilizando la metodología PSEV-BF; se observa que los píxeles correspondientes a no pájaros forman parte del fondo. Asimismo, cerca del 86 % de los píxeles correspondientes a aves fueron correctamente clasificados.

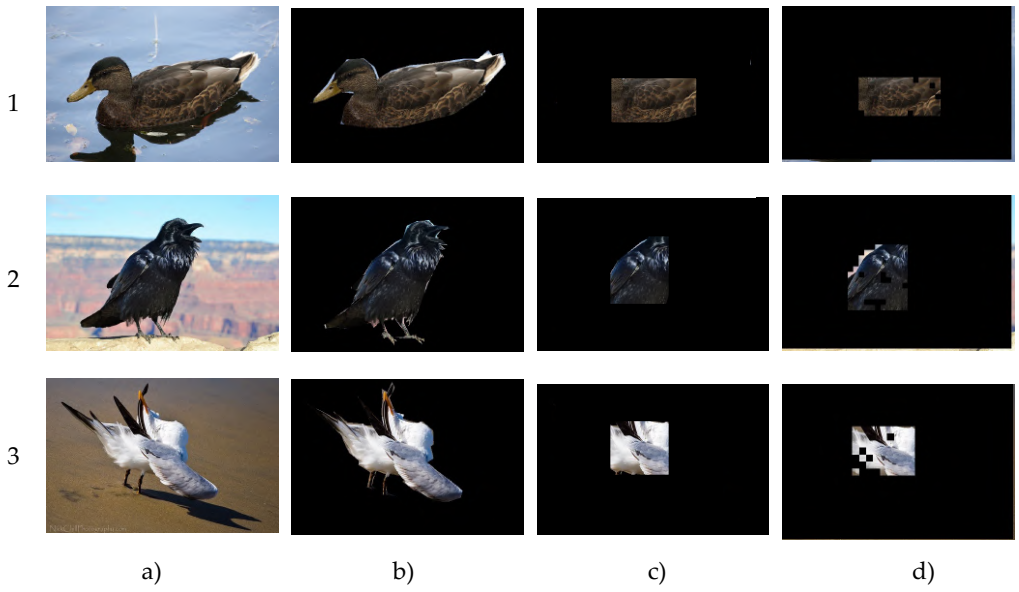

Figura 5.4: *Resultado de la metodología con un APIoU* $< 80\,\%$.

Finalmente, la Figura 5.5-d) muestra algunos ejemplos con un porcentaje de IoU menor al 30 % obtenido

por la metodología PSEV-BF; se puede observar que los píxeles correspondientes a aves no fueron correctamente clasificados.

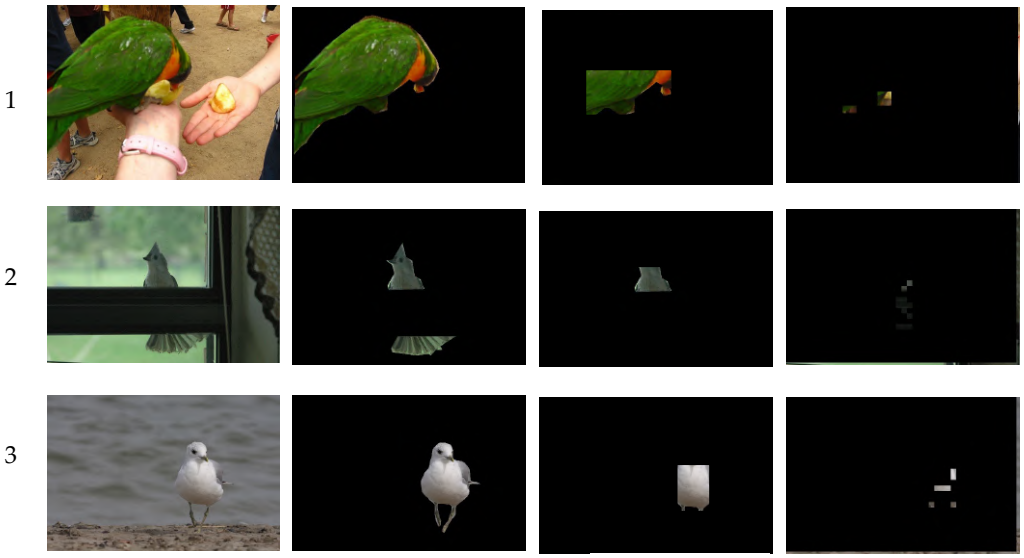

Figura 5.5: *Resultado de la metodología con un APIoU* $< 30\,\%$.

CAPÍTULO 6

Conclusiones y trabajos futuros

Todos los objetivos que están planteados en la sección 1.4.2 de este proyecto de investigación fueron alcanzados satisfactoriamente. En este capítulo se describen las contribuciones principales, la producción científica generada y las líneas de trabajo futuro.

6.1. CONTRIBUCIONES PRINCIPALES

En esta tesis se propone una nueva metodología de visión computacional para el reconocimiento de objetos del tipo ave con presencia de oclusión y ambientes no controlados, la cual es llamada PSEV-BF (Pre-Segmentation and Enhanced Variables for Bird Features).

PSEV-BF contribuye con tres nuevos procesos en comparación de una metodología tradicional de

reconocimiento de objetos, tales procesos son: pre-segmentación, extracción de características mediante la técnica de maxpooling y mejoramiento de la fase de selección de características. Los nuevos procesos pueden ser integrados por otras metodologías los cuales pueden presentar cuantiosas mejorías.

6.2. PRODUCCIÓN CIENTÍFICA

Durante el desarrollo de esta tesis doctoral se realizaron las siguientes publicaciones en revistas:

Artículos:

- Frausto-Solís Juan, **Hernández-González Lucía J**, González-Barbosa Juan J, Sánchez-Hernández Juan Paulo, Román-Rangel Edgar, "*Convolutional neural Network–Component transformation (CNN–CT) for confirmed COVID-19 cases*," Mathematical and Computational Applications., vol 26, Abril 2021. [40]. Apéndice A.

  https://doi.org/10.3390/mca26020029.

- **Hernández-González Lucía J**, Frausto-Solís Juan, González-Barbosa Juan J, Sánchez-Hernández Juan Paulo, Hernández-Rabadán Deny Lizbeth, Román-Rangel Edgar, "*PSEV-BF methodology for detection and classification of birds in uncontrolled environments*," Axioms, vol 12, febrero 2023. JCR Q2. Factor de impacto: 1.824 (2021) [41].

  https://doi.org/10.3390/axioms12020197.

Ponencias:

- Frausto-Solís Juan, **Hernández-González Lucía J**, González-Barbosa Juan J, Sánchez-Hernández Juan Paulo, Román-Rangel Edgar.(Noviembre 18-19, 2020). "*Forecast for confirmed cases using*

CNN, ARIMA, and Exponential Smoothing "[Sesion de conferencia]. 8to International Workshop on Numerical and Evolutionary Optimization. Xalapa, Veracruz, Mexico.

https://neo.cinvestav.mx/NEO2020/index.php/neo

- Martínez-Neri, Gerardo de Jesús, Frausto-Solís Juan, González-Barbosa Juan Javier, **Hernández-González Lucía J**. (Noviembre 08-10, 2022) "*Intelligent Forecasting Methods for COVID-19, MLP-SVRES*"[Sesion de conferencia]. 10th International Workshop on Numerical and Evolutionary Optimization. Xalapa, Veracruz, Mexico. https://neo.cinvestav.mx/NEO2022/

6.3. TRABAJOS FUTUROS

Para trabajos futuros, proponemos utilizar técnicas similares para la segmentación de imágenes supervisadas. También implementar otra metaheurística para mejorar la detección de variables de características, como la aceptación de umbrales (TA). También planeamos mejorar nuestros resultados de reconocimiento utilizando diferentes estrategias de clasificación como Support Vector Machine y Multilayer Perceptron.

Bibliografía

[1] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-3, pp. 610–621, 1973.

[2] L.-K. Soh and C. Tsatsoulis, "Texture analysis of sar sea ice imagery using gray level co-occurrence matrices," *IEEE Transactions on geoscience and remote sensing*, vol. 37, pp. 780–795, 1999.

[3] D. A. Clausi, "An analysis of co-occurrence texture statistics as a function of grey level quantization," *Canadian Journal of remote sensing*, vol. 28, pp. 45–62, 2002.

[4] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," 2014, pp. 740–755.

[5] E. Alegre, G. Pajares, and A. De La Escalera, *Conceptos y métodos en visión por computador*, E. Alegre Gutiérrez, G. Pajares Martinsanz, and A. De La Escalera, Eds., 2016.

[6] M. A. Khan, T. Akram, M. Sharif, M. Awais, K. Javed, H. Ali, and T. Saba, "Ccdf: Automatic system for segmentation and recognition of fruit crops diseases based on correlation coefficient and deep cnn features," *Computers and Electronics in Agriculture*, 2018.

[7] W. S. Lee and D. C. Slaughter, "Recognition of partially occluded plant leaves using a modified watershed algorithm," *Transactions of the ASAE*, vol. 47, no. 1999, pp. 1269–1280, 2004. [Online]. Available: http://www.abe.ufl.edu/wlee/Publications/TransASAE-Vol47-No4-p1269-1280.pdf

[8] M. A. Ebrahimi, M. H. Khoshtaghaza, S. Minaei, and B. Jamshidi, "Vision-based pest detection based on SVM classification method," *Computers and Electronics in Agriculture*, 2017.

[9] J. Sivic, C. L. Zitnick, and R. Szeliski, "Finding people in repeated shots of the same scene," *Procedings of the British Machine Vision Conference 2006*, pp. 93.1–93.10, 2006. [Online]. Available: http://www.bmva.org/bmvc/2006/papers/246.html

[10] C. Su, S. Zhang, J. Xing, W. Gao, and Q. Tian, "Multi-type attributes driven multi-camera person re-identification," *Pattern Recognition*, 2018.

[11] I. J. Agric, B. Eng, Z. Chuanlei, Z. Shanwen, Y. Jucheng, S. Yancui, and C. Jia, "Apple leaf disease identification using genetic algorithm and correlation based feature selection method," 2017. [Online]. Available: https://www.ijabe.org

[12] B. K. Tripathy and A. Agrawal, "Efficiency analysis of hybrid fuzzy c-means clustering algorithms and their application to compute the severity of disease in plant leaves," pp. 2581–6640, 2019. [Online]. Available: http://purkh.com/index.php/tocomp

[13] X. Liu, F. Xu, Y. Sun, H. Zhang, and Z. Chen, "Convolutional recurrent neural networks for observation-centered plant identification," 2018. [Online]. Available: https://doi.org/10.1155/2018/9373210

[14] P. Salve, P. Yannawar, and M. Sardesai, "Multimodal plant recognition through hybrid feature fusion technique using imaging and non-imaging hyper-spectral data," *Journal of King Saud University - Computer and Information Sciences*, 2018.

[15] P. G. Vázquez, " Estrategias para identificar oclusiones y planificación monocular para una mejora de la percepción visual de la escena," Ph.D. dissertation, Universidad de Alicante, 2008. [Online]. Available: https://rua.ua.es/dspace/bitstream/10045/29216/1/tesis{_}pablo{_}gil{_}vazquez.pdf

[16] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 2018, cite arxiv:1804.02767Comment: Tech Report. [Online]. Available: http://arxiv.org/abs/1804.02767

[17] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," vol. 220, p. 4598, 1983.

[18] A. Chaudhury and J. L. Barron, "Partially Occluded Leaf Recognition via Subgraph Matching and Energy Optimization," 2017.

[19] D. L. Hernández-Rabadán, "Metodología integrativa para segmentación de vegetación en imá genes de color bajo ambientes no controlados," Ph.D. dissertation, Instituto Tecnológico y Estudios Superiores de Monterrey, 2015.

[20] E. Alegre, G. Pajares, and A. De La Escalera, *Conceptos y métodos en visión por computador*, E. Alegre Gutiérrez, G. Pajares Martinsanz, and A. De La Escalera, Eds. España: CEA, 2016.

[21] G. Pajares, G. Martinsanz, and J. de la Cruz García, *Visión por computador. Imágenes Digitales y Aplicaciones. 2a Edición*. RA-MA S.A. Editorial y Publicaciones, 2008. [Online]. Available: https://books.google.com.mx/books?id=EQqsPgAACAAJ

[22] A. C. Pinto Leal, "Segmentación De Imágenes Por Textura," Ph.D. dissertation, Universidad de concepción, 2006. [Online]. Available: http://repositorio.udec.cl/bitstream/handle/11594/873/TESIS{_}SEGMENTACION{_}DE{_}IMAGENES{_}POR{_}TEXTURA.Image.Marked.pdf?sequence=1{&}isAllowed=y

[23] M. Presutti, "La Matriz de co-ocurrencia en la clasificación multiespectral: tutorial para la enseñanza de medidas texturales en cursos de grado universitario," p. 10, 2004. [Online]. Available: http://www3.inpe.br/unidades/cep/atividadescep/jornada/programa/t-9{_}trab{_}27.pdf

[24] C. M. Cuadras, *Analysis Multivariante*, 2004.

[25] H. Sanvicente-Sánchez and J. Frausto-Solís, "A method to establish the cooling scheme in simulated annealing like algorithms," in *Computational Science and Its Applications–ICCSA 2004*. Springer, 2004, pp. 755–763.

[26] G. Dueck and T. Scheuer, "Threshold Accepting: A General Purpose Optimization Algorithm Appearing Superior to Simulated Annealing," vol. 90, pp. 161–175, 1990.

[27] F. Chollet, *Deep Learning with Python*, 1st ed. USA: Manning Publications Co., 2017.

[28] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.

[29] R. Xiang, "Image segmentation for whole tomato plant recognition at night," *Computers and Electronics in Agriculture*, 2018.

[30] L. D. S. Pacifico, V. Macario, and J. F. L. Oliveira, *Plant Classification Using Artificial Neural Networks*, 2018.

[31] Y. Liang, M. Zhang, and W. N. Browne, "Image feature selection using genetic programming for figure-ground segmentation," *Engineering Applications of Artificial Intelligence*, vol. 62, pp. 96–108, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.engappai.2017.03.009

[32] M. Nasir, M. A. Khan, M. Sharif, I. U. Lali, T. Saba, and T. Iqbal, "An improved strategy for skin lesion detection and classification using uniform segmentation and feature selection based approach," *Microscopy Research and Technique*, vol. 81, pp. 528–543, 2018.

[33] M. Sharif, M. A. Khan, Z. Iqbal, M. F. Azam, M. I. U. Lali, and M. Y. Javed, "Detection and classification of citrus diseases in agriculture based on optimized weighted segmentation and feature selection," *Computers and Electronics in Agriculture*, vol. 150, pp. 220–234, 2018. [Online]. Available: https://doi.org/10.1016/j.compag.2018.04.023

[34] Z. U. Rehman, S. S. Naqvi, T. M. Khan, M. Arsalan, M. A. Khan, and M. Khalil, "Multi-parametric optic disc segmentation using superpixel based feature classification," *Expert Systems with Applications*, vol. 120, pp. 461–473, 4 2019. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S095741741830770X

[35] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[36] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[37] P. Brynolfsson, D. Nilsson, T. Torheim, T. Asklund, C. T. Karlsson, J. Trygg, T. Nyholm, and A. Garpebring, "Haralick texture features from apparent diffusion coefficient (adc) mri images depend on imaging and pre-processing parameters," *Scientific reports*, vol. 7, no. 1, pp. 1–11, 2017.

[38] L.-K. Soh and C. Tsatsoulis, "Texture analysis of sar sea ice imagery using gray level co-occurrence matrices," *IEEE Transactions on geoscience and remote sensing*, vol. 37, no. 2, pp. 780–795, 1999.

[39] D. A. Clausi, "An analysis of co-occurrence texture statistics as a function of grey level quantization," *Canadian Journal of remote sensing*, vol. 28, no. 1, pp. 45–62, 2002.

[40] J. Frausto-Solís, L. J. Hernández-González, J. J. González-Barbosa, J. P. Sánchez-Hernández, and E. Román-Rangel, "Convolutional neural network–component transformation (cnn–ct) for confirmed covid-19 cases," *Mathematical and Computational Applications*, vol. 26, no. 2, p. 29, 2021.

[41] L. J. Hernández-González, J. Frausto-Solís, J. J. González-Barbosa, J. P. Sánchez-Hernández, D. L. Hernández-Rabadán, and E. Román-Rangel, "Psev-bf methodology for object recognition of birds in uncontrolled environments," *Axioms*, vol. 12, no. 2, 2023. [Online]. Available: https://www.mdpi.com/2075-1680/12/2/197

[42] K. K. Sahu, A. K. Mishra, and A. Lal, "Coronavirus disease-2019: An update on third coronavirus outbreak of 21st century," *QJM: An International Journal of Medicine*, vol. 113, no. 5, pp. 384–386, may 2020. [Online]. Available: https://academic.oup.com/qjmed/article/113/5/384/5775511

[43] J. F. Solis, J. E. O. Vazquez, and J. J. G. Barbosa, "The Hybrid Forecasting Method SVR-ESAR for Covid-19 Background," vol. 12, no. April, pp. 42–48, 2021.

[44] W. O. Kermack and A. G. McKendrick, "A contribution to the mathematical theory of epidemics," *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character*, vol. 115, no. 772, pp. 700–721, aug 1927. [Online]. Available: https://royalsocietypublishing.org/doi/10.1098/rspa.1927.0118

[45] G. Hyndman, Rob J and Athanasopoulos, "Forecasting: principles and practice," 2018. [Online]. Available: https://otexts.com/fpp2/seasonal-arima.html

[46] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, sep 1995. [Online]. Available: http://link.springer.com/10.1007/BF00994018

[47] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT press Massachusetts, USA, 2016, vol. 1, no. 2. [Online]. Available: https://d1wqtxts1xzle7.cloudfront.net/62266271/Deep{_}Learning20200303-80130-1s42zvt.pdf?1583287496={&}response-content-disposition=inline{%}3B+filename{%}3DDeep{_}Learning.pdf{&}Expires=1613256371{&}Signature=PsBCgNOukIJXPL-cL9ZC1sqZjqmFc9MWmqmgJftHnHY0qht7wpOAKla76Eo

[48] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," *The handbook of brain theory and neural networks*, vol. 3361, no. 10, pp. 255–258, 1995.

[49] SRK, "Novel CoronaVirus 2019 dataset," 2020.

[50] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 8, no. 9, pp. 1735–1780, 1997.

[51] M. J. Roberts, *Signals and Systems: Analysis Using Transform Methods and MATLAB*. McGraw-Hill, 3rd. Ed., 2018.

[52] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in Neural Information Processing Systems*, 2015.

[53] S. Bai, J. Z. Kolter, and V. Koltun, "An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling," *arXiv:1803.01271*, 2018.

[54] M. J. Keeling, E. M. Hill, E. E. Gorsich, B. Penman, G. Guyver-Fletcher, A. Holmes, T. Leng, H. McKimm, M. Tamborrino, L. Dyson, and M. J. Tildesley, "Predictions of COVID-19 dynamics in the UK: Short-term forecasting and analysis of potential exit strategies," *PLOS Computational Biology*, vol. 17, no. 1, p. e1008619, jan 2021. [Online]. Available: https://dx.plos.org/10.1371/journal.pcbi.1008619

[55] E. E. Ramirez-Torres, A. R. Selva Castañeda, Y. Rodríguez-Aldana, S. Sánchez Domínguez, L. E. Valdés García, A. Palú-Orozco, E. Oliveros-Domínguez, L. Zamora-Matamoros, R. Labrada-Claro, M. Cobas-Batista, D. Sedal-Yanes, O. Soler-Nariño, P. A. Valdés-Sosa, J. I. Montijano, and L. E. Bergues Cabrales, "Mathematical modeling and forecasting of COVID-19: experience in Santiago de Cuba province," *Revista Mexicana de Física*, vol. 67, no. 1 Jan-Feb, p. 123, jan 2021. [Online]. Available: https://rmf.smf.mx/ojs/rmf/article/view/5369

[56] M. A. Capistran, A. Capella, and J. A. Christen, "Forecasting hospital demand in metropolitan areas during the current COVID-19 pandemic and estimates of lockdown-induced 2nd waves," *PLOS ONE*, vol. 16, no. 1, p. e0245669, jan 2021. [Online]. Available: https://dx.plos.org/10.1371/journal.pone.0245669

[57] L.-P. Chen, Q. Zhang, G. Y. Yi, and W. He, "Model-based forecasting for Canadian COVID-19 data," *PLOS ONE*, vol. 16, no. 1, p. e0244536, jan 2021. [Online]. Available: https://dx.plos.org/10.1371/journal.pone.0244536

[58] M. Ala'raj, M. Majdalawieh, and N. Nizamuddin, "Modeling and forecasting of COVID-19 using a hybrid dynamic model based on SEIRD with ARIMA corrections," *Infectious Disease Modelling*, vol. 6, pp. 98–111, 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2468042720301032

[59] S. Deb and M. Majumdar, "A time series method to analyze incidence pattern and estimate reproduction number of COVID-19," *arXiv*, pp. 1–14, 2020.

[60] S. M. Parvez, S. S. A. Rakin, M. Asadut Zaman, I. Ahmed, R. A. Alif, Ania-Nin-Ania, and R. M. Rahman, "A Comparison Between Adaptive Neuro-fuzzy Inference System and Autoregressive Integrated Moving Average in Predicting COVID-19 Confirmed Cases in Bangladesh," 2021, pp. 741–754. [Online]. Available: http://link.springer.com/10.1007/978-981-15-8354-4{_}73

[61] F. Petropoulos and S. Makridakis, "Forecasting the novel coronavirus COVID-19," *PLoS ONE*, vol. 15, no. 3, pp. 1–8, 2020. [Online]. Available: http://dx.doi.org/10.1371/journal.pone.0231236

[62] Z. Hussain and M. Dutta Borah, "Forecasting Probable Spread Estimation of COVID-19 Using Exponential Smoothing Technique and Basic Reproduction Number in Indian Context," 2021, pp. 183–196. [Online]. Available: http://link.springer.com/10.1007/978-981-15-9735-0{_}10

[63] V. K. R. Chimmula and L. Zhang, "Time series forecasting of COVID-19 transmission in Canada using LSTM networks," *Chaos, Solitons & Fractals*, vol. 135, p. 109864, jun 2020. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0960077920302642

[64] R. Chandraa, A. Jainb, and D. S. Chauhanc, "Deep learning via LSTM models for COVID-19 infection forecasting in India," *arXiv*, 2021.

[65] A. Zeroual, F. Harrou, A. Dairi, and Y. Sun, "Deep learning methods for forecasting COVID-19 time-Series data: A Comparative study," *Chaos, Solitons and Fractals*, vol. 140, 2020.

[66] T. Saba, I. Abunadi, M. N. Shahzad, and A. R. Khan, "Machine learning techniques to detect and forecast the daily total COVID-19 infected and deaths cases under different lockdown types," *Microscopy Research and Technique*, p. jemt.23702, feb 2021. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/jemt.23702

[67] D. Parbat and M. Chakraborty, "A python based support vector regression model for prediction of COVID19 cases in India," *Chaos, Solitons and Fractals*, vol. 138, pp. 3–7, 2020.

[68] C. Katris, "A time series-based statistical approach for outbreak spread forecasting: Application of COVID-19 in Greece," *Expert Systems with Applications*, vol. 166, p. 114077, mar 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0957417420308368

[69] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR*, 2015.

[70] S. Makridakis, J. S. Armstrong, R. Carbone, and R. Fildes, "An editorial statement," *Journal of Forecasting*, vol. 1, no. 1, pp. 1–2, jan 1982. [Online]. Available: http://doi.wiley.com/10.1002/for.3980010102

[71] J. B. Legler and T. D. Robertson, "National and regional econometric models," in *Studies in applied regional science*. Springer US, 1976, pp. 16–31. [Online]. Available: https://doi.org/10.1007/978-1-4613-4360-8_2

[72] S. Raschka and V. Mirjalili, "Python machine learning: Machine learning and deep learning with python," *Scikit-Learn, and TensorFlow. Second edition ed*, 2017.

[73] S. Islam, S. I. A. Khan, M. M. Abedin, K. M. Habibullah, and A. K. Das, "Bird species classification from an image using vgg-16 network," 2019, pp. 38–42. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/3348445.3348480

[74] A. Malhi and R. X. Gao, "Pca-based feature selection scheme for machine defect classification," *IEEE Transactions on Instrumentation and Measurement*, vol. 53, pp. 1517–1525, 2004.

[75] Y. Lu, I. Cohen, X. S. Zhou, and Q. Tian, "Feature selection using principal feature analysis," 2007, pp. 301–304.

[76] F. Song, Z. Guo, and D. Mei, "Feature selection using principal component analysis," vol. 1, 2010, pp. 27–30.

[77] M. Uçar, "Classification performance-based feature selection algorithm for machine learning: P-score," *IRBM*, vol. 41, pp. 229–239, 8 2020.

[78] S. Gu, R. Cheng, and Y. Jin, "Feature selection for high-dimensional classification using a competitive swarm optimizer," *Soft Computing*, vol. 22, pp. 811–822, 2018.

[79] R. Adhao and V. Pachghare, "Feature selection using principal component analysis and genetic algorithm," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 23, pp. 595–602, 2 2020.

[80] F. N. Koutanaei, H. Sajedi, and M. Khanbabaei, "A hybrid data mining model of feature selection algorithms and ensemble learning classifiers for credit scoring," *Journal of Retailing and Consumer Services*, vol. 27, pp. 11–23, 2015.

[81] R. Kavitha and E. Kannan, "An efficient framework for heart disease classification using feature extraction and feature selection technique in data mining," 2016, pp. 1–5.

[82] A. K. Gárate-Escamila, A. H. E. Hassani, and E. Andrès, "Classification models for heart disease prediction using feature selection and pca," *Informatics in Medicine Unlocked*, vol. 19, p. 100330, 2020.

[83] E. O. Abiodun, A. Alabdulatif, O. I. Abiodun, M. Alawida, A. Alabdulatif, and R. S. Alkhawaldeh, "A systematic review of emerging feature selection optimization methods for optimal text classification: the present state and prospective opportunities," *Neural Computing and Applications*, vol. 33, pp. 15 091–15 118, 11 2021.

[84] R.-C. Chen, C. Dewi, S.-W. Huang, and R. E. Caraka, "Selecting critical features for data classification based on machine learning methods," *Journal of Big Data*, vol. 7, p. 52, 12 2020.

[85] S. Rath, S. Kumar, V. S. K. Guntupalli, S. M. Sourabh, and S. Riyaz, "Analysis of deep learning methods for detection of bird species." IEEE, 2 2022, pp. 234–239.

[86] S.-J. Hong, Y. Han, S.-Y. Kim, A.-Y. Lee, and G. Kim, "Application of deep-learning methods to bird detection using unmanned aerial vehicle imagery," *Sensors*, vol. 19, p. 1651, 4 2019.

[87] W. Xiang, Z. Song, G. Zhang, and X. Wu, "Birds detection in natural scenes based on improved faster rcnn," *Applied Sciences*, vol. 12, p. 6094, 6 2022.

[88] F. Mashuk, A. Sattar, N. Sultana *et al.*, "Machine learning approach for bird detection," 2021, pp. 818–822.

[89] H. G. Akçay, B. Kabasakal, D. Aksu, N. Demir, M. Öz, and A. Erdoğan, "Automated bird counting with deep learning for regional bird distribution mapping," *Animals*, vol. 10, p. 1207, 7 2020.

[90] A. E. Öztürk and E. Erçelebi, "Real uav-bird image classification using cnn with a synthetic dataset," *Applied Sciences*, vol. 11, p. 3863, 4 2021.

[91] H. Wang, Y. Xu, Y. Yu, Y. Lin, and J. Ran, "An efficient model for a vast number of bird species identification based on acoustic features," *Animals*, vol. 12, p. 2434, 9 2022.

[92] Y.-Q. Ou, C.-H. Lin, T.-C. Huang, and M.-F. Tsai, "Machine learning-based object recognition technology for bird identification system," 2020, pp. 1–2.

[93] A. Kumar and S. D. Das, "Bird species classification using transfer learning with multistage training," 2018, pp. 28–38.

[94] R. C. Gonzalez and R. E. W. 3rd, "Edition," *Digital Image Processing. Upper Saddle River, USA: Prentice Hall*, 2008.

[95] D. J. Ketcham, "Real-time image enhancement techniques," vol. 74, 1976, pp. 120–125.

[96] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2016, pp. 779–788.

[97] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[98] X. Zhang, Y. Gao, H. Wang, and Q. Wang, "Improve yolov3 using dilated spatial pyramid module for multi-scale object detection," *International Journal of Advanced Robotic Systems*, vol. 17, p. 1729881420936062, 2020.

[99] P. Brynolfsson, D. Nilsson, T. Torheim, T. Asklund, C. T. Karlsson, J. Trygg, T. Nyholm, and A. Garpebring, "Haralick texture features from apparent diffusion coefficient (adc) mri images depend on imaging and pre-processing parameters," *Scientific reports*, vol. 7, pp. 1–11, 2017.

[100] H. Sanvicente-Sánchez and J. Frausto-Solís, "A method to establish the cooling scheme in simulated annealing like algorithms," 2004, pp. 755–763.

[101] L. Boltzmann, "The second law of thermodynamics." Springer, 1974, pp. 13–32.

[102] R. Genuer and J.-M. Poggi, "Random forests." Springer, Cham, 2020, pp. 33–55.

[103] S. R. Sain and V. N. Vapnik, "The nature of statistical learning theory," *Technometrics*, vol. 38, p. 409, 11 1996.

ANEXO A

Red Neuronal Convolucional-Transformación de Componentes (CNN-CT) para casos confirmados de COVID-19

Los coronavirus son una gran familia de virus caracterizados por tener picos en forma de corona en su superficie. Hoy en día, son siete tipos identificados de coronavirus que pueden transmitirse entre humanos. Los coronavirus más peligrosos conocidos hasta los últimos años son MERS-CoV y SARS-CoV, y habían provocado enfermedades graves, como MERS y SARS, en 2003 y 2012, respectivamente [42]. Sin embargo, a finales de 2019, en Wuhan, China, surgió el nuevo brote epidemiológico de COVID-19; causó el nuevo coronavirus llamado SARS-CoV2.

La importancia de los modelos y algoritmos matemáticos para analizar esta enfermedad ha crecido porque permiten encontrar patrones, hacer predicciones y comprender fluctuaciones. Los modelos epidemiológicos se pueden clasificar en dos grupos [43]:

- Modelos dinámicos. Son modelos antiguos que generalmente dividen a la población en varios subconjuntos conocidos como compartimentos, por ejemplo en el modelo *Susceptible, Infectious, Recovered* o SIR. El modelo SIR fue propuesto en 1902 por Sir Roland Ross, luego ampliado por Kermack y McKendrick en 1927 [44].
- Modelos de predicción mediante series de tiempo. Aquí encontramos métodos clásicos como ARIMA y Exponential Smoothing (ES) [45]. Además, los métodos de aprendizaje automático como Support Vector Machines [46] y Deep Learning [47] están en este grupo.

Este trabajo presenta un nuevo método del segundo grupo, basado en Convolutional Neural Network (CNN) [48] y una propuesta de Component Transformation (CT) que denominamos CNN-CT, cuya formulación matemática se presenta. El método CNN-CT se aplica para pronosticar el número de casos confirmados de COVID-19 para Estados Unidos (EE. UU.), México, Brasil y Colombia [49]. El CT cambia las observaciones diarias en datos semanales y viceversa. El pronóstico realizado por nuestro método híbrido CNN-CT se ajusta aún más con los métodos ARIMA o ES. Comparamos el método híbrido propuesto con los métodos individuales. Nuestros resultados muestran que el método combinado logra consistentemente resultados competitivos en términos de la métrica MAPE, a diferencia de cualquiera de sus elementos: CNN, ARIMA o ES, cuyo desempeño como métodos individuales varía en gran medida para los diferentes países. Además, el método CNN-CT propuesto también supera a la memoria a corto plazo a largo plazo (LSTM) [50], que se encuentra entre los métodos más utilizados para tratar con series de tiempo.

Tanto CNN como LSTM corresponden a métodos de Deep Learning, el primero equipado con filtros convolucionales mientras que el segundo con operaciones recurrentes, pero en ambos casos con parámetros que se aprenden a través de métodos similares al descenso de gradientes en escenarios donde los datos se utilizan para el entrenamiento a medida que se vuelven. disponible. Por el contrario,

ARIMA y ES son métodos de regresión tradicionales que consideran un conjunto completo de datos de entrenamiento a la vez, por lo que tienen el potencial de aproximarse mejor a dicho conjunto de entrenamiento, pero pierden la capacidad de ajustarse a los datos recientemente disponibles como CNN y LSTM. El método propuesto de CNN-CT aprovecha tanto el potencial de incorporar datos recientemente disponibles como la fortaleza de mirar un conjunto completo de datos, lo que da como resultado un método de pronóstico enriquecido.

Elegimos usar CNN dado que la literatura sobre procesamiento de señales establece que los filtros convolucionales son más estables que las operaciones recurrentes como LSTM [51]. Además, el rendimiento superior de los CNN sobre los métodos tradicionales, como ARIMA, ha sido confirmado por trabajos previos centrados en la clasificación de texto [52] y el modelado de secuencias [53], donde las convoluciones obtuvieron un mayor rendimiento con respecto a otros métodos.

El resto de este capitulo está organizado de la siguiente manera. En la sección B.2, discutimos trabajos relacionados con el pronóstico de casos confirmados de COVID-19. En la sección **??**, mostramos el método de pronóstico propuesto para los casos confirmados diarios de COVID-19, destacando la aplicación de los métodos Deep Learning, ARIMA y ES. En la sección **??**, presentamos detalles sobre los datos y las herramientas utilizadas para validar nuestro método. Finalmente, las secciones **??** y **??** presentan resultados y conclusiones de este trabajo.

A.1. TRABAJOS RELACIONADOS

COVID-19 es una enfermedad con una alta tasa de propagación, lo que ha generado un interés en estimar y pronosticar el número de casos de personas infectadas. Recientemente, se han presentado varios trabajos con modelos epidemiológicos tradicionales o Modelos Dinámicos. El modelo *Susceptible, Exposed, Infectious, Recovered* (SEIR) [54] se utilizó para pronosticar los casos confirmados en el Reino Unido, y los modelos SIR y SEIR se aplicaron para pronosticar los casos infectados y recuperados acumulados en Santiago de Cuba [55]. El modelo *Susceptible, Expuesto, Infeccioso, Recuperado,*

Muerto (SEIRD) [56] se utilizó para pronosticar casos confirmados y de muerte en México.

En Chen [57], se realizó un trabajo comparativo para predecir 11 días de casos confirmados en algunas regiones de Canadá y Estados Unidos. Usan modelos SIR, Neural Network y ARIMA.

Se utilizaron ARIMA y ES como métodos de ajuste para mejorar los resultados obtenidos para otros modelos como los obtenidos para los modelos SIR, Redes Neuronales y Algoritmos de Regresión de Vector de Soporte [43, 58]. Sin embargo, en la mayoría de los casos, el número de días pronosticado suele ser demasiado corto. Por ejemplo, in [59] se utilizó ARMA para pronosticar casos confirmados durante tres días en provincias chinas, países asiáticos y algunos países occidentales (Alemania, Estados Unidos, Italia y España). Parvez et al., Compararon un Sistema de Inferencia Neuro-difuso Adaptativo versus ARIMA para predecir diez días de casos confirmados por COVID-19 en Bangladesh [60]. Además, Petropoulus et al. [61], utilizó el método ES conocido como Holt-Winter para pronosticar diez días de casos confirmados de Covid-19 acumulados a nivel mundial. Hussain y col. [62], usó un ES para estimar doce días de casos confirmados y el parámetro R_0 conocido como el número de reproducción básico.

Los métodos ARIMA y Deep Learning se han utilizado solos para pronosticar casos de COVID-19. Chimmula [63] utilizó LSTM para predecir casos diarios, obteniendo con este método un error del ocho por ciento utilizando MAPE. En Chandraa [64], LSTM, BiLSTM y EDLSTM se utilizaron para pronosticar la propagación de infecciones por COVID-19 entre estados seleccionados de la India. El trabajo presentado por Zeroul et al. [65], utilizó el aprendizaje profundo para predecir 10 días de personas infectadas, obteniendo un error MAPE entre 1.28 % y 59 %. Saba et al. [66], compararon los modelos de regresión polinomial, Holt-Winter, ARIMA y SARIMA, para predecir los casos confirmados y de muerte. Parbat et al. [67], propusieron utilizar un modelo SVR-Radial para pronosticar el total de muertes y las muertes diarias recuperadas, acumuladas confirmadas y confirmadas en la India; este método obtuvo alrededor del trece por ciento de error MAPE para todo el país.

Además, los métodos de pronóstico clásicos se han combinado con técnicas de aprendizaje automático [43, 58, 68]. Katris [68] utilizó modelos ARIMA, ES, Neural Network y MARS, donde los métodos

combinados funcionaron mejor que los métodos individuales.

En general, los métodos ARIMA y ES se utilizan para pronosticar casos con períodos de corto plazo, mientras que los modelos de aprendizaje automático y aprendizaje profundo pueden predecir casos durante períodos más prolongados. Sin embargo, el último caso no siempre obtiene buenos resultados cuando se utilizan como métodos individuales.

A.1.1. Método CNN-CT

Mostramos el método CNN-CT propuesto en la Figura 1.1, donde se utiliza una CNN como método de pronóstico primario para los casos confirmados diarios de COVID-19, y se complementa con ARIMA o ES, que se utilizan como métodos de ajuste contra errores diarios.

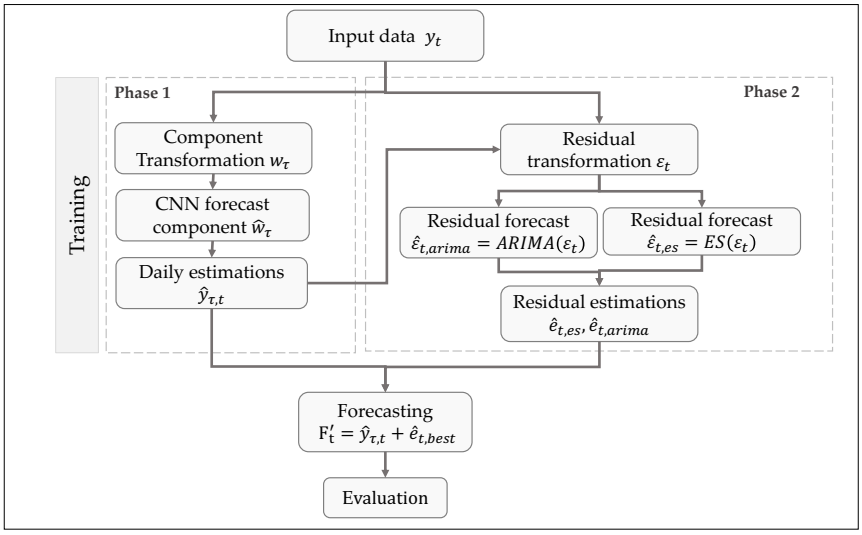

Figura 1.1: *Método CNN-CT propuesto: Entrenamiento con dos fases, la primera fase corresponde al método de pronóstico utilizando valores de componentes. La segunda fase utilizó valores residuales con método de pronóstico residual.*

En primer lugar, la etapa de entrenamiento de nuestro método se compone de dos fases, cada una de las cuales está formada por tres subprocesos internos más un subproceso de integración global, como se muestra en la Figura ref fig: CNN-CT.

En el primer subproceso de la fase 1, comenzamos transformando los valores diarios y_t en componentes semanales w_τ, donde t es un índice de día y τ es un índice de componente. Estos w_τ componentes representan estimaciones de previsión semanal promedio. En el segundo subproceso, se utiliza una CNN para pronosticar el componente $\hat{w}_\tau$. Finalmente, en el tercer subproceso, convertimos la estimación del componente w_τ nuevamente en estimaciones diarias $\hat{y}_{t,\tau}$.

En la fase 2, se entrenan los métodos de ajuste. Primero, obtenemos el ε_t residual de la diferencia entre la predicción diaria y su correspondiente valor de verdad fundamental, es decir, $\hat{y}_{\tau,t} - y_t$. Escalamos estos valores residuales para que estén en el rango $[1, 10]$, como lo requieren los métodos de Holt-Winter.

En el segundo subproceso de la fase 2, usamos los residuos ε_t para entrenar un modelo autorregresivo usando ARIMA o ES, que se usa para pronosticar valores residuales $\hat{\varepsilon}_t$ (concretamente, $\hat{\varepsilon}_{t,es}$ y $\hat{\varepsilon}_{t,arima}$ para ES y ARIMA, respectivamente).

Posteriormente, en el tercer subproceso de la fase 2, los pronósticos residuales $e_{t,es}$ o $e_{t,arima}$ se obtienen a partir de los valores de pronóstico residual calculados previamente. Finalmente, este pronóstico residual $e_{t,X}$ se suma a la estimación diaria $\hat{y_{\tau,t}}$ obtenida de la CNN, lo que da como resultado el valor de predicción final F_t'.

A.1.2. Transformación de los datos

Los modelos de predicción reflejan un mayor error a medida que aumenta el número de períodos de predicción. Seleccionamos pronosticar más casos transformando registros diarios en componentes semanales con el módulo CT, que mapea los casos diarios y_t en componentes w_τ que representan un promedio ponderado de los casos diarios obtenidos en una semana. Los valores w_τ se calculan con la Ecuación (A.1).

$$w_\tau = \frac{\sum_{t=7\tau-6}^{7\tau} y_t}{7}, \tag{A.1}$$

donde w_τ es el promedio semanal de la semana τ, y $w_1, w_2, \ldots, w_\tau$ es un conjunto de observaciones transformadas a componentes. Por ejemplo $w_2 = \frac{y_8+y_9+\ldots+y_{14}}{7}$.

A.1.3. Pronostico de componentes por CNN

Usamos una CNN como método de pronóstico de componentes. Las etapas de entrenamiento y validación se componen de valores w_τ. La arquitectura CNN contiene una capa de entrada con 50 neuronas de convolución, una capa de agrupación máxima de tamaño igual a 2. Una capa MLP completa de 50 neuronas y una capa de salida con una sola neurona. Las capas de convolución utilizan la función de activación ReLU. Los parámetros de configuración de entrenamiento son los siguientes: Adam optimizer [69], error absoluto medio como función de pérdida, 100 épocas y tamaño de lote igual a 10. La configuración anterior se usa para pronosticar componentes semanales $\hat{w}_\tau$.

A.1.4. Estimaciones diarias

La transformación inversa o las estimaciones diarias implican convertir los componentes semanales w_τ nuevamente en valores diarios. Para ello, es necesario calcular los subcomponentes de un componente, que definimos como se muestra en la Tabla A.1.4.

Week w_τ						
subcomponent $\delta_{\tau,1}$				subcomponent $\delta_{\tau,2}$		
Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday

La segmentación de la semana en dos subcomponentes proporciona información sobre el comportamiento social de los países por separado al principio y al final de la semana. La distribución de los casos diarios con respecto a sus subcomponentes se puede obtener mediante la Ecuación (A.2) y la Ecuación (A.3).

$$\delta_{\tau,1} = \frac{\sum_{t=1,\tau}^{4,\tau} y_{t,\tau}}{4}, \tag{A.2}$$

$$\delta_{\tau,2} = \frac{\sum_{t=5,\tau}^{7,\tau} y_{t,\tau}}{3}, \tag{A.3}$$

donde $\delta_{\tau,1}, \delta_{\tau,2}$ son subcomponentes ADS-1 (Lunes a Jueves) y ADS-2 (Viernes a Domingo) para la componente τ. Determinamos la razón diaria X que representa la proporción de los valores diarios originales para el subcomponente 1 y 2 para el componente τ (Ecuación (A.4)). La razón diaria $\mu_{\tau,t}$ nos permite determinar los casos de normalización de días de la semana x_t (Ecuación (A.5)) de la fase de entrenamiento. En otras palabras, $x_1 = mondeys_{avg}, \ldots, x_7 = sundays_{avg}$ son casos promedio confirmados de cada día de la semana a lo largo de la serie de tiempo.

$$\mu_{\tau,t} = \begin{cases} \frac{y_{\tau,t}}{\delta_{\tau,1}}, & \text{if } 1 \leq t \leq 4, \\ \frac{y_{\tau,t}}{\delta_{\tau,2}} & \text{if } 5 \leq t \leq 7, \end{cases} \tag{A.4}$$

$$x_t = \frac{\sum_{i=1}^{\tau} \mu_i}{\tau}. \tag{A.5}$$

La ponderación de los casos diarios obtenidos con la razón $\mu_{\tau,t}$ permite obtener una estimación estadística de la relevancia de las personas infectadas en el primer y segundo subcomponente τ, j de cada componente τ durante todo el período de formación. La transformación inversa determina los casos diarios predichos a partir de los componentes usando la Ecuación (A.6) y la Ecuación (A.7).

$$\hat{\delta}_{\tau,i} = \hat{w}_\tau \frac{w_\tau}{\delta_{\tau,i}}, \tag{A.6}$$

$$\hat{y}_{\tau,t} = \begin{cases} x_t \hat{\delta}_{\tau,1}, & \text{if } 1 \leq t \leq 4, \\ x_t \hat{\delta}_{\tau,2} & \text{if } 5 \leq t \leq 7, \end{cases} \tag{A.7}$$

donde $\hat{y}_{\tau,t}$ representa los valores del caso de pronóstico del componente τ en el momento t; $\hat{\delta}_{\tau,i}$ es el pronóstico del número promedio de subcomponentes infectados i en el componente tau. Los datos para el aprendizaje de los métodos de ajuste se obtienen a partir de los valores de predicción diaria de la fase de validación de los componentes $y_{\tau,t}$.

A.1.5. Transformación de residuales

Un valor residual esta dado por la diferencia del valor real y el valor predicho, como se muestra en la Ecuación (A.8).

$$e_t = y_t - \hat{y}_t = y_t - y_{t-1}, \tag{A.8}$$

donde y_t es el valor real en el tiempo t, $\hat{y}_t$ es el valor pronosticado en el tiempo t. Usando la Ecuación (A.8), los residuales e_t son extradidos mediante la diferencia de y_t y $y_{\tau,t}$, como se muestra en la Ecuación (A.9).

$$e_t = y_{\tau,t} - y_t, \tag{A.9}$$

donde $y_{\tau,t}$ es el valor pronosticado en el tiempo t del componente τ. Los métodos de ARIMA y ES usan valores positivos; debido a esto, los residuales fueron normalizados como se muestra en la Ecuación (A.10).

$$\varepsilon_t = |y_{\tau,t} - y_t|, \tag{A.10}$$

donde $|.|$ representa la normalización de e_t en el rango de valores $[1, 10]$.

A.1.6. Pronostico de residuales

Los métodos de ARIMA y ES son utilizados como métodos de ajuste de pronósticos. El conjunto de entrenamiento y validación estan compuestos por los valores de ε_t.

La configuración del modelo ARIMa es la siguiente: start_p=0, d=0, start_q=0, max_p=5, max_q=5, max_d=5, start_Q=0, max_P=5, max_D=5, max_Q=5, m=4, seasonal=True, error_action='warn', trace=True, suppress_warnings=True, stepwise=True, random_state=20, n_fits=50, information_criterion = 'aic', and alpha=0.05.

Así mismo, Holt-Winter (HW) es el método de ES implementado. Las variantes de HW que se usaron son: additive, multiplicative, additive damped, multiplicative damped. Estas variantes fueron entrenadas con residuales normalizadas ε_t.

A.1.7. Estimación de residuales

Usamos las transformaciones residuales ε_t para entrenar ARIMA y ES, del cual obtuvimos cuatro métodos híbridos CNN-ARIMA, CNN-ES, LSTM-ARIMA, and LSTM-ES. El pronostico $\varepsilon_{t,es}$ y $\varepsilon_{t,arima}$ de estos métodos híbridos son transformados en residuales $e_{t,es}, e_{t,arima}$ los cuales están en un dominio no normalizado.

A.1.8. Pronostico

Finalmente, evaluamos los valores pronosticados de la fase de validacion F_t', los cuales están compuestos de pronósticos diarios $y_{\tau,t}$ de CNN, como se muestra en la Ecuación (A.11).

$$F_t' = y_{\tau,t} + e_{t,best}. \tag{A.11}$$

A.1.9. Configuración de experimentos

A continuación se describe el conjunto de datos, el preprocesamiento aplicado, el criterio de separación de datos en el entrenamiento, la validación y las pruebas utilizado en este trabajo. Finalmente, se describen las métricas de evaluación.

A.1.10. Conjunto de datos

La base de datos COVID-19 utilizada en este trabajo es la series de tiempo del Nuevo Coronavirus 2019 [49], cuyos registros informan la cantidad de personas infectadas, recuperadas y fallecidas en cada país del mundo. Las series de tiempo de Estados Unidos, México, Brazil y Colombia fueron usados a partir del 22 de enero de 2020.

El periodo de entrenamiento es del 2 de marzo de 2020 hasta el 28 de junio de 2020 (17 semanas); del 29 de junio de 2020 al 19 de julio de 2020 para la validación (3 semanas); y del 20 de julio de 2020 al 9 de agosto de 2020 para prueba (3 semanas). La figura 1.2 muestra un esquema para esta división de datos.

Con este split, el entrenamiento del método CNN-CT para EE. UU., Se realizó con 17 componentes semanales w_τ, como se explica en el apartado A.1.2. En el caso de México, Brasil y Colombia, usamos solo 15 componentes semanales ya que los datos correspondientes a las primeras dos semanas fueron descartados por falta de información significativa, es decir, los valores de la semana uno y dos son considerablemente bajos con respecto al resto de la serie. Notamos que el procesamiento de estas dos primeras semanas dan como resultado una subestimación de los valores del pronóstico.

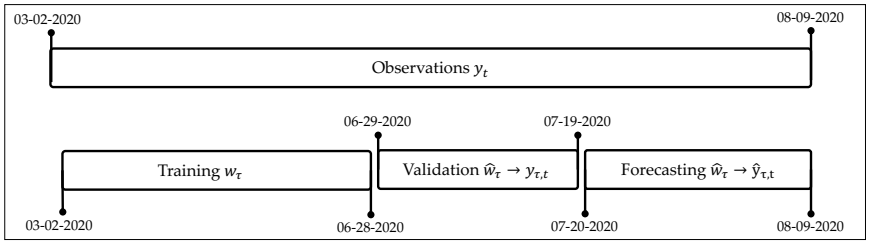

Figura 1.2: *Division de las observaciones correspondientes al conjunto de entrenamiento y validación para los métodos de pronostico diarios.*

Aunque el entrenamiento se realiza usando componentes semanales w_τ, el pronóstico para las etapas de validación y prueba ocurre en valores diarios $y_{\tau,t}$, como se explica en la sección A.1.4.

Los pronósticos residuales permiten ajustar el pronóstico diario con ARIMA y ES. Además, se estaba entrenando con los residuales de la validación diaria del pronóstico, es decir, los pronósticos w_τ obtenidos en la fase de validación se transforman en estimaciones diarias $y_{\tau,t}$ para ser utilizados en el entrenamiento y fase de validación de los métodos de ajuste. La figura 1.3 muestra un esquema para esta división de datos para los métodos de ajuste.

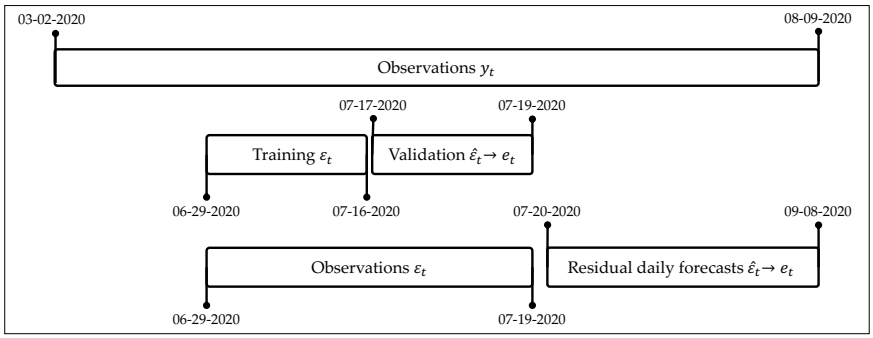

Figura 1.3: *Division de las observaciones correspondientes al conjunto de entrenamiento y validación para los métodos de pronóstico de ajuste.*

Dado que el problema que abordamos corresponde a un escenario de autoregresión, la estructura real de los datos es tal que, cada variable de salida y_t depende de un vector de valores pasados $\mathbf{x} = [y_{t-1}, y_{t-2}, \ldots, y_{t-T}]$. Para este trabajo, usamos hasta tres valores en el pasado, $t-3$, $t-2$ y $t-1$.

A.1.11. Métricas

El método CNN-CT híbrido propuesto y sus métodos de composición individuales son evaluados por MAPE [70], ya que ha sido ampliamente utilizado en los trabajos discutidos en la Sección B.2. El MAPE calcula el porcentaje de precisión en el valor predicho con respecto a la verdad del terreno. Cuanto más cerca de cero, más preciso es. Otra métrica común es RMSPE [71] que también se usa en parte de este documento.

$$MAPE = \frac{100}{n} \sum_{t=1}^{n} \frac{|y_t - \hat{y}_t|}{y_t}, \tag{A.12}$$

$$RMSPE = \sqrt{\frac{\sum_{t=1}^{n} (y_t - \hat{y}_t)^2}{n}} * 100, \tag{A.13}$$

donde, y_t es el valor real, $\hat{y}_t$ es el valor predicho, y n indica el número total de muestras.

A.1.12. Herramientas

Este trabajo se realizo en una computadora con sistema operativo iOS, 8 GB y un procesador Intel Core i5 Dual-Core de 2.3 GHz. Usamos Python 3.7.1 y el modelo de CNN se creó con las bibliotecas de Tensorflow y Keras [72].

A.1.13. Resultados

Esta sección muestra los resultados del método CNN-CT propuesto para los casos de pronóstico diario de COVID-19 en los EE. UU., México, Brasil y Colombia. Primero, comparamos el desempeño de usar CNN y LSTM como los principales métodos de pronóstico con ARIMA y ES (Holt-Winter, HW)

como métodos de ajuste. Luego, presentamos la comparación del modelo CNN-CT versus los modelos individuales CNN, LSTM, ARIMA y Holt-Winters para cada país.

Podemos ver en la Figura 1.4 la comparación de los modelos de pronóstico de mejor desempeño para los países de Estados Unidos, México, Brasil y Colombia. En EE. UU., Figura 1.4 - (a), los pronósticos de LSTM-ARIMA logran mantener los patrones de tendencia y estacionalidad con respecto a la verdad del terreno. Sin embargo, el pronóstico de CNN-HW está muy por debajo de los datos reales. Podemos ver en la Tabla 1.1 que LSTM-ARIMA logra el MAPE más bajo para los EE. UU.

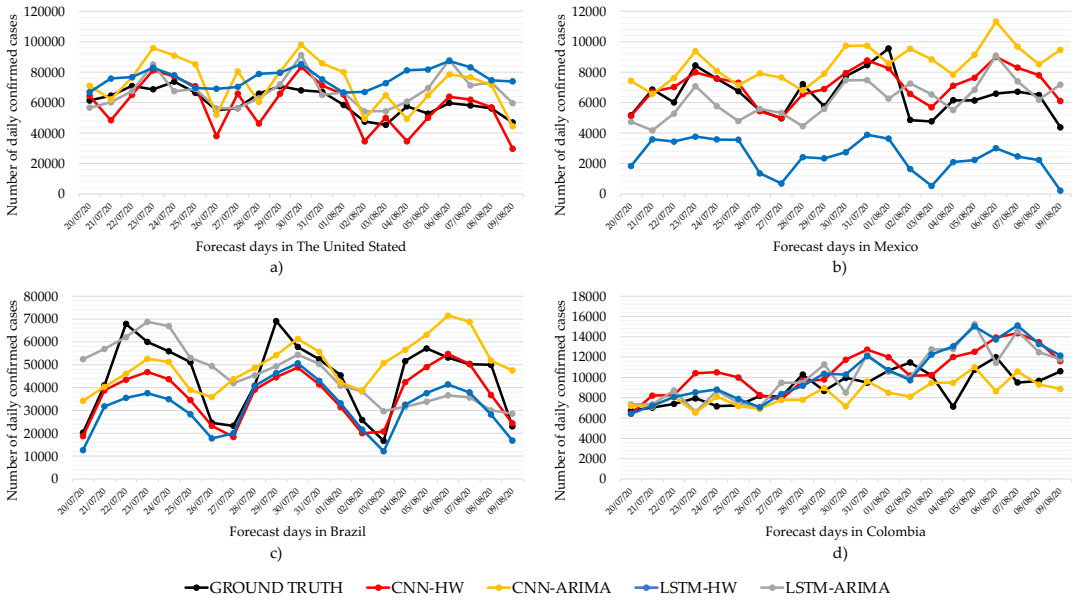

Figura 1.4: *Daily forecast with CNN-CT method using CNN and LSTM as main forecast methods.*

Así mismo, la Figura 1.4 - (b) muestra el comportamiento de los pronósticos para los casos diarios de COVID-19 en México. Podemos ver que los cuatro modelos pueden mantener patrones de tendencia y estacionalidad con respecto a los valores reales. Sin embargo, LSTM-ARIMA presenta una alta tasa de error debido a la diferencia con respecto a estos. Por otro lado, los pronosticos de CNN-HW estan muy cercas de los datos reales, lo que permite obtener un mejor desempeño con respecto a los otros métodos. El MAPE promedio y su desviación estándar se muestran en la Tabla 1.1, donde podemos ver que CNN-HW logra el mejor desempeño promedio entre los cuatro modelos.

De manera similar, la Figura 1.4 - (c) muestra el pronóstico comparativo de Brasil para todos los modelos. Podemos ver que LSTM-ARIMA logra mantener patrones de estacionalidad con respecto a los valores reales. En el caso de CNN-HW, sigue los patrones de tendencia y estacionalidad con respecto a lestos. El MAPE promedio y su desviación estándar se muestran en la Tabla 1.1. Así mismo, CNN-HW tiene el mejor desempeño.

Podemos ver en la Figura 1.4 - (d), que LSTM-ARIMA logra mantener patrones de estacionalidad con respecto a la verdad básica para Colombia. En el caso de CNN-HW, sigue los patrones de tendencia y estacionalidad relacionados con la verdad básica. Según la Tabla 1.1 CNN-ARIMA muestra el mejor rendimiento de MAPE, ya que su curva es la más cercana a los datos de prueba.

En general, nuestros experimentos muestran que el suavizado con ARIMA o ES ayuda a obtener un MAPE más bajo en el caso de CNN. Este no es el caso de LSTM. La tabla 1.1 muestra un resumen de los valores de pronóstico diario de MAPE de CNN-CT y LSTM-CT para EE. UU., México, Brasil y Colombia. En el caso de EE. UU., El método con mejor rendimiento es LSTM-ARIMA, con un $MAPE \approx 14\,\%$. En el caso de México y Brasil, CNN-HW es mejor con MAPE $14.18\,\%$ y $29.3\,\%$. Es posible ver que LSTM-ARIMA y CNN-HW obtienen mejores resultados en diferentes países. En Colombia, CNN-ARIMA obtiene los mejores MAPE.

Hicimos un promedio del MAPE de todos los países para cada método en la Tabla 1.1. Observamos que los métodos CNN-CT tienen un mejor rendimiento que los de LSTM-CT. Además, para cada país, determinamos la desviación estándar. Notamos que CNN-CT tiene la desviación estándar más baja, lo que indica que su desempeño es consistente en todos los países.

Tabla 1.1: *Desempeño de CNN-CT. Los mejores valores de MAPE estan marcados en negro.*

Country	CNN-HW		CNN-ARIMA		LSTM-HW		LSTM-ARIMA	
	MAPE	RMSPE	MAPE	RMSPE	MAPE	RMSPE	MAPE	RMSPE
United State	15.53	19.35	22.64	26.57	38.57	43.64	**13.35**	**16.70**
Mexico	**14.19**	**18.78**	36.82	47.37	71.66	73.32	25.73	31.53
Brazil	**29.30**	**31.27**	39.69	62.58	62.63	70.75	44.26	54.59
Colombia	21.76	28.46	**13.39**	**16.84**	24.56	32.48	20.00	26.07
Average	**20.19**	**24.47**	28.14	38.34	49.36	55.05	25.84	32.22
Standar Desv	**5.98**	**5.50**	10.68	17.82	18.74	17.46	11.51	13.96

Finalmente, en la Figura 1.5 mostramos una comparación del MAPE para el modelo CNN-HW versus los modelos CNN, LSTM, ARIMA y Holt-Winters individuales para cada país.

Si bien ARIMA obtuvo un buen desempeño para EE.UU. (11.18) y México (16.31), primer y tercer lugar, respectivamente, brinda un MAPE alto para Brasil (50.99) y Colombia (29.75), con el último y el penúltimo lugar, respectivamente. De manera similar, la CNN pura es un buen método para México (14.04) y Colombia (14.96), pero no tan bueno para Estados Unidos (42.75) y Brasil (38.19).

En contraste, CNN-CT (CNN-HW) es consistentemente competitivo en todos los casos, obteniendo el segundo lugar para EE. UU. (15.53), México (14.18, tan bueno como la CNN con mejor desempeño solo) y Colombia (21.75), y el primero. para Brasil (29.30).

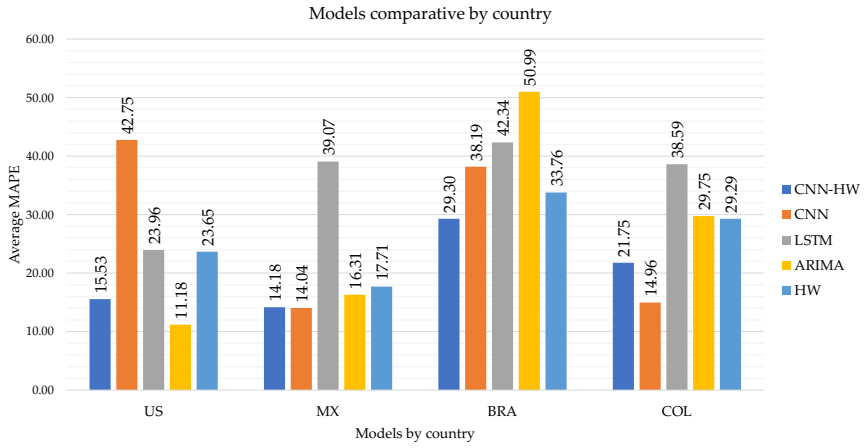

Figura 1.5: *Pronósticos diarios de CNN-CT (using HW) contra los métodos individuales CNN, LSTM, ARIMA and HW.*

Tabla 1.2: *Desempeño de CNN-CT vs métodos individuales. Los mejores valores de MAPE están marcados en negro.*

	métrica de MAPE				
País	CNN-HW	CNN	LSTM	ARIMA	HW
Estados Unidos	15.53	42.75	23.96	**11.18**	23.65
México	14.18	**14.04**	39.07	16.31	17.71
Brazil	**29.30**	38.19	42.34	50.99	33.76
Colombia	**21.75**	14.96	38.59	29.75	29.29
Promedio de MAPE	**20.19**	27.49	35.99	27.06	26.10
Desviación Std.	**5.98**	13.09	7.09	15.39	6.03

Mostramos la comparación de CNN-HW versus los cuatro métodos individuales en Table 1.2. Podemos ver que CNN-HW supera todos estos métodos individuales para Brasil y Colombia. Para el caso de México, CNN-HW está por debajo del método de mejor desempeño (CNN) solo por 0.14 puntos de MAPE. Además, CNN-HW logra resultados competitivos para EE. UU.

A.1.14. Conclusiones

Esta trabajo investiga el problema de pronosticar casos diarios confirmados de COVID-19 en México, Brasil, Colombia y EE. UU. Dada la cantidad limitada de datos disponibles en el momento de realizar nuestros experimentos, se hicieron evidentes varias limitaciones de los métodos de predicción. Estas limitaciones fueron aún más notorias debido a la presencia de ruido en los datos diarios, que muy bien podría ser una consecuencia de las restricciones al flujo de datos impuestas por la crisis sanitaria relacionada con COVID-19 a nivel mundial.

En particular, la mayoría de los métodos de predicción disminuyen su precisión a medida que aumentan los períodos de predicción. Para mitigar este problema, propusimos una transformación de componentes que convierte los valores diarios en componentes semanales para una predicción correcta en esos casos.

Presentamos un método de pronóstico híbrido denominado Transformación de componentes con una red neuronal de convolución (CNN-CT), que utiliza CNN y LSTM como principal método de predicción; y ES y ARIMA como métodos de ajuste para la corrección diaria de errores. Como resultado, hay dos variantes del método propuesto: CNN-CT con Holt-Winters y LSTM-CT con ARIMA.

Comparamos el rendimiento de predicción de los métodos individuales que componen el CNN-CT propuesto utilizando la métrica MAPE. Notamos que CNN y LSTM son muy buenos con la tendencia de aprendizaje y la estacionalidad de la serie temporal; sin embargo, los pronósticos de LSTM tienden a generar una tendencia creciente y decreciente, lo que hace que el error aumente. Nuestros experimentos muestran que el suavizado con ARIMA o ES ayuda a obtener un MAPE más bajo en el caso de CNN. Este no es el caso de LSTM.

Como trabajos futuros, proponemos aplicar esta metodología a otros métodos de pronóstico populares como SVR, Red neuronal recurrente, etc. Para medir la calidad del rendimiento en más países y aplicar una potente limpieza de datos como etapa de preprocesamiento. Además, podría ser interesante utilizar diferentes métodos de ajuste. Finalmente, probar si la metodología propuesta es completamente general o determinar qué estrategia aplicar en diferentes escenarios de pronóstico.

ANEXO B

Metodología PSEV-BF para reconocimiento de objetos del tipo aves en ambientes no controlados.

B.1. INTRODUCCIÓN

Los científicos medioambientales suelen utilizar a las aves para comprender los ecosistemas porque son sensibles a los cambios medioambientales [73]. Por ello, hay muchas zonas protegidas en todo el mundo dedicadas a la conservación de especies de aves. Sin embargo, identificar y clasificar aves utilizando visión artificial convencional es una tarea difícil. Se trata de un problema especialmente complicado en el caso de imágenes con oclusión en entornos no controlados.

La visión por ordenador es un campo de la inteligencia artificial que intenta extraer información

significativa analizando y procesando patrones de imágenes. Además, este campo, tiene varias ramas: clasificación, localización de objetos, detección de objetos, reconocimiento de objetos y segmentación.

El reconocimiento de objetos es una tarea que identifica un objeto presente en imágenes o vídeos. Es una de las aplicaciones más importantes del aprendizaje automático y del aprendizaje profundo. El propósito de este campo es reconocer el contenido de una imagen utilizando técnicas de aprendizaje automático o arquitectura de aprendizaje profundo.

La figura 2.1a muestra la metodología clásica de visión computacional para el reconocimiento de objetos en visión por computador. La otra alternativa es utilizar una arquitectura de aprendizaje profundo (Figura 2.1b). Como vemos, esta última es menos interpretable porque equivale a una caja negra donde se ocultan los principales procesos de extracción y selección de características. Este trabajo propone una metodología que combina elementos de las dos metodologías

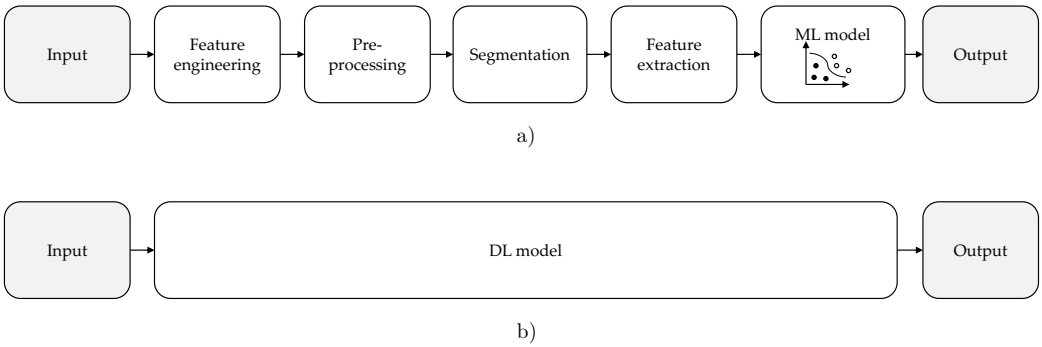

Figura 2.1: *(a) Traditional computer vision methodology with ML. (b) Deep learning computer methodology.*

El reconocimiento de objetos se considera uno de los problemas críticos porque existen varios retos al tratar con imágenes, como:

A. Oclusión, es decir, obstrucción parcial del objeto por elementos similares o no al objeto, Figura 2.2-a).

B. Los artefactos ambientales, como la lluvia y la niebla, pueden afectar a la calidad de la imagen,

Figura 2.2-b).

C. Los entornos no controlados se deben a la falta de un protocolo para la captura de imágenes, como el fondo contrastado del objeto, la altura, la distancia a la cámara y la corrección de la luz, Figura 2.2-c)

a) b) c)

Figura 2.2: *Ejemplos de los retos con imágenes: (a) El pájaro aparece ocluido. (b) La aparición de neblina en la imagen suaviza el color del fondo. (c) La imagen aparece con desenfoque respecto al ángulo de la cámara [4].*

Por lo tanto, para reconocer un objeto, la metodología considera tareas como la detección y la segmentación de objetos. La segmentación es el problema principal y en el que nos centramos en este trabajo.

El objetivo de la segmentación es identificar los píxeles que pertenecen al objeto objetivo o Región de Interés (ROI). Sin embargo, determinar el número óptimo de regiones por imagen requiere mucho tiempo y es costoso desde el punto de vista computacional. Los métodos de segmentación basados en la clasificación píxel a píxel pueden dividirse a grandes rasgos en dos familias: segmentación semántica y segmentación por instancias. El primer tipo separa todos los píxeles que pertenecen a la misma clase de objeto. La segunda identifica cada uno de los objetos presentes en la imagen como un individuo.

Tradicionalmente, la selección de variables o características se realiza mediante variables compuestas, como la técnica de Análisis de Componentes Principales (PCA) [74–76] y otros métodos de Clasificación [77] . Las variables compuestas son métodos que simplifican el espacio muestral de las variables mediante la normalización de combinaciones lineales de las mismas. Sin embargo, en los últimos años

se han publicado métodos para mejorar la selección de características incorporando métodos de optimización combinatoria [78–83] y selecciones de modelos para machine learning [84]. Por esta razón, en este trabajo, proponemos incluir un método mejorado para la selección de características utilizando el algoritmo Simulated Annealing (SA), una metaheurística para la optimización combinatoria, que se utiliza para mejorar el conjunto de características seleccionadas con la técnica PCA.

Presentamos una nueva metodología para el reconocimiento de objetos denominada PSEV-BF (Pre-Segmentación y Variables Mejoradas para Características de Aves) que utiliza la información de pre-segmentación antes de la segmentación, para refinar el área delimitada. Esta metodología tiene las fases de Pre-procesamiento, Pre-segmentación, Segmentación, extracción de características ROI, realce de variables relevantes, y Clasificación.

El resto del trabajo está organizado de la siguiente manera. Sección 2, presenta trabajos relacionados con una comparación cualitativa del reconocimiento de objetos. Sección 3, presenta la formulación y descripción de todas las fases de la metodología propuesta. La Sección 4, define los datos, las métricas de rendimiento y las herramientas utilizadas en este trabajo. En la Sección 5, presentamos los algoritmos propuestos y su método de ajuste y mostramos la aplicación de la metodología al conjunto de datos presentado en el trabajo. Por último, comparamos los resultados con la metodología clásica. La Sección 6 presenta nuestras conclusiones.

B.2. TRABAJOS RELACIONADOS

En esta sección, se discuten varios trabajos relacionados con los problemas de las fases de visión por computador. Por ejemplo, en el trabajo de [31], se propone un algoritmo de selección de características basado en programación genética (GP). En [31], se implementó la segmentación y clasificación de imágenes de caballos y aviones utilizando los algoritmos Parsimony GP features Selection (PGP-FS), Nondominated Sorting GP Feature Selection (NSGP-FS), y Strength Pareto GP Feature Selection (SPGT-FS). Estas características se sometieron a los clasificadores Decision Tree, Naive Bayes y

Multilayer Perceptron de la herramienta Weka. Se extrajeron un total de 52 características en términos de filtro de Gabor, color y valores estadísticos basados en una escala de grises. Se utilizaron las métricas accuracy, F1, precision y recall. El método de selección muestra que, por término medio, se seleccionan 15 características de las 52 originales.

Existen trabajos relacionados con la segmentación y clasificación de imágenes de lesiones cutáneas. Por ejemplo, en [32] los autores utilizan la técnica PCA y el método de entropía de Boltzmann para seleccionar un conjunto de características. La selección de características se realiza considerando la puntuación (varianza explicada) de cada componente PCA. Las características consideradas fueron color, textura y forma, dando como resultado un total de 3849 características. Utilizando PCA y Boltzmann, el número de características se redujo a 449. La selección de características se validó mediante las métricas DICE, índice de Jaccard, distancia de Jaccard y diámetro Seg. Las características seleccionadas se clasificaron mediante los siguientes modelos de aprendizaje automático: Máquina de vectores soporte (SVM), Árboles de decisión (DT), Árboles en bolsa (BT), Análisis discriminante subespacial (SDA), Vecino más cercano ponderado-K (W-KNN), Vecino más cercano fino-K (F-KNN), Vecino más próximo K subespacial (S-KNN), análisis discriminante lineal (LDA), análisis discriminante cuádruple (QDA), máquina de vectores de soporte cúbica (C-SVM) y máquina de vectores de soporte cuádruple (Q-SVM). Los clasificadores se validaron utilizando las métricas de sensibilidad, especificidad, precisión y puntuación F.

En 2018, Sharif y colaboradores propusieron una metodología para la detección de enfermedades de los cítricos utilizando segmentación ponderada optimizada y selección de características [33]. La fase de procesamiento consiste en un filtro Top-hat para eliminar elementos de ruido y un filtro gaussiano para suavizar la imagen y eliminar las fluctuaciones de alta intensidad. En la fase de segmentación, utilizaron una combinación de técnicas de segmentación con asignación de pesos y mapa de relevancia, que permite retener los elementos de la imagen con alto contraste. Las características extraídas están relacionadas con el color, la textura y las características geométricas, lo que da un total de 270 características. Se utiliza PCA para obtener una puntuación correspondiente a la varianza explicada de los componentes. Se calculan la entropía y la asimetría de cada componente para seleccionar un vector de 100 características con los porcentajes más altos. Estas características se obtuvieron entrenando un

clasificador K-Nearest Weighted (KNN), Ensemble Boosted Trees (EBT), DT y LDA y se evaluaron con 10 veces. La validación de la metodología se realizó utilizando las métricas Tasa de falsos positivos, Tasa de falsos negativos, Tasa de verdaderos positivos, Tasa de falsos negativos, Valor predictivo positivo, Tasa de detección de falsos, Área bajo la curva y Precisión. Los autores demostraron que sus resultados están a la altura del estado actual de la técnica.

Rehman et al en 2018, aplicaron una selección de características para la segmentación de imágenes para detectar glaucoma en la región del disco óptico utilizando varios parámetros [34]. Además, en el Preprocesado, se aplicó un filtro bilateral que permite la eliminación de ruido, un recorte que permite activar un criterio de umbral para mantener los objetos con alta intensidad y descartar el ruido de fondo no deseado, y por último, la normalización del canal rojo (R) de la imagen para obtener información sobre los excitadores buscados. En la fase de segmentación se utilizaron características estadísticas, de texto en el mapa y fractales. A continuación se realizó un proceso de selección según el método de mínima redundancia $(M_{I(A,B)})$. Estas características se entrenaron utilizando clasificadores SVM, Random Forest (RF), AdaBoostM1 y Rus Boost. El modelo se validó utilizando las métricas de sensibilidad, especificidad, coeficiente de similitud DICE, precisión y superposición de áreas basadas en la matriz de confusión, con resultados que compiten con el estado del arte.

Más recientemente, los métodos basados en aprendizaje profundo se utilizan para la detección de aves [85–89], clasificar imágenes de aves [90], y el reconocimiento de aves [91]. Por ejemplo, [88] utilizó redes neuronales convolucionales (CNN) y You Only Look Once V3 (YOLOV3) para la detección de aves a partir de imágenes. En este trabajo, los autores proponen una CNN con una arquitectura similar a la red Darknet-53. El modelo fue validado usando la métrica Accuracy y comparado con arquitecturas similares como la red neuronal convolucional basada en regiones (R-CNN), VGG-16+SVM, y YOLO. Asimismo, Q. Ou et al, en un trabajo anterior de 2020, utilizaron la arquitectura You Only Look Once (YOLO) para identificar aves. Otros trabajos proponen métodos híbridos para mejorar la detección e identificación de aves [92]. Kumar y Das en 2018, propusieron una R-CNN [93], que se utilizó para obtener máscaras binarias de la ROI y se entrenó con instancias de la base de datos Commons Object in Context (COCO).

La Tabla 2.1 muestra un resumen de los aspectos más importantes de los trabajos relacionados en comparación con nuestra propuesta. La primera columna contiene el nombre del método utilizado y su referencia. La segunda columna determina si las aves son el objeto de interés. Las columnas tercera y cuarta indican si las imágenes utilizadas están ocluidas y si se encuentran en entornos no controlados. Las columnas quinta a séptima indican si la metodología del trabajo fue sometida a Preprocesamiento, Presegmentación y Segmentación. La octava columna indica el número de características seleccionadas. La novena columna, denominada Çaracterísticas mejoradas", indica si se utilizaron o no métodos específicos para mejorar la variable seleccionada por PCA. Por último, la última columna indica si se utilizaron técnicas de clasificación. La Tabla 1 muestra los temas considerados en las diferentes metodologías y relacionados con este trabajo. Observamos que la presegmentación y las características mejoradas no se utilizan habitualmente.

Tabla 2.1: *Comparativa entre las principales procesos de las metodologías en trabajos relacionados.*

Método	**Ave**	**Oclusión**	**UE**	**Pre-P**	**Pre-S**	**Seg.**	**C.S.**	**C.M.**	**C.ML**
Genetic Programming [31]	55	55	51	55	55	51	51	51	51
Classical-PCA/SVM [32]	55	55	55	51	55	51	51	55	51
Classical-PCA/SVM [33]	55	55	55	51	55	51	51	51	51
Classical-Minimun Redundancy/SVM [34]	55	51	55	51	55	51	51	51	51
Deep CNN-53 [88]	51	51	51	51	55	55	55	55	51
Deep CNN-19 [92]	51	51	51	55	55	55	55	55	51
CNN-Transfer Learning [93]	51	51	51	55	55	51	51	55	51
CNN-16 [73]	51	55	51	51	55	55	51	55	51
PSEV-BF (proposal)	51	51	51	51	51	51	51	51	51

UE: Ambientes no controlados, Pre-P: Pre-Procesamiento, Pre-S: Pre-Segmentación, Seg: Segmentación; C.S.: Características seleccionadas; C.M.: Carqacterísticas mejoradas; C.ML: Clasificadores Machine Learning.

B.2.1. Metodología propuesta

La metodología PSEV-BF propuesta (Figura 2.3) consta de siete fases para el entrenamiento y seis para las pruebas: Pre-procesamiento, Pre-segmentación, Segmentación, Extracción de características ROI, Selección óptima de variables, Clasificación y Evaluación. En esta sección se describen detalladamente todas las fases de este trabajo.

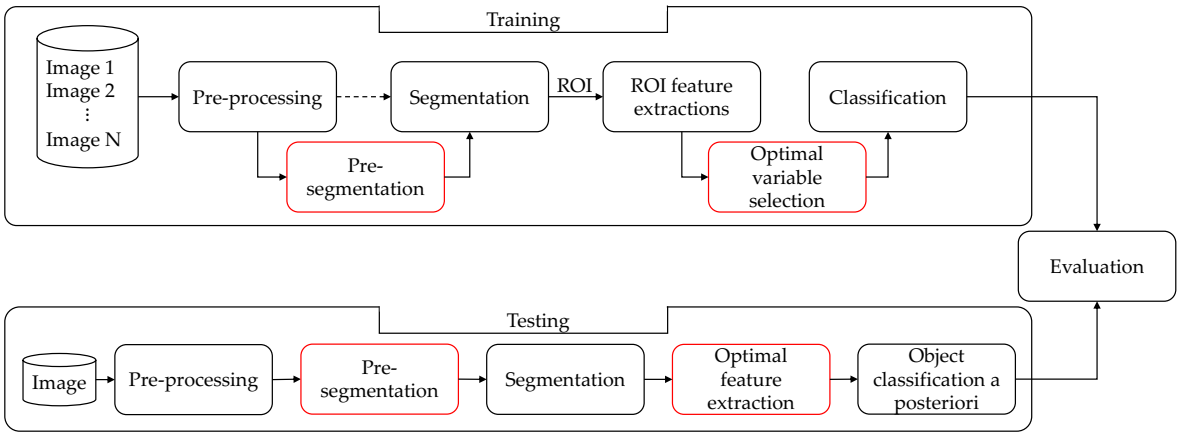

Figura 2.3: *Metodología PSEV-BF propuesta con preprocesamiento, presegmentación, segmentación, extracción de características de ROI, selección óptima de variables, clasificación y evaluación.*

B.2.2. Métricas

$$MAPE = \frac{100}{n}\sum_{t=1}^{n}\frac{|y_t - \hat{y}_t|}{y_t}, \tag{B.1}$$

$$RMSPE = \sqrt{\frac{\sum_{t=1}^{n}(y_t - \hat{y}_t)^2}{n}} * 100, \tag{B.2}$$

donde, y_t es el valor real, $\hat{y}_t$ es el valor predicho, y n indica el nÃºmero total de muestras.

B.2.3. Pre-procesamiento

El preprocesado de imágenes se utiliza para mejorar su calidad visual donde se podrían eliminar varios problemas como efectos de brillo, problemas de iluminación y desenfoque debido a un contraste pobre [33], [94]. Una imagen con bajo contraste afecta a la precisión de la segmentación y, por tanto, al resto de fases. En este trabajo se aplica una técnica de mejora del contraste basada en la función

de suavizado gaussiano y la ecualización de histogramas. En primer lugar, se aumenta el contraste de la imagen añadiendo un filtro de ecualización de histograma. A continuación, se aplica el filtro de suavizado gaussiano. El procedimiento de mejora se describe en los siguientes pasos:

Paso 1. La ecualización del histograma de una imagen es una transformación cuyo objetivo es obtener una distribución uniforme para cada nivel de intensidad de una imagen. En pocas palabras, ajusta las intensidades de la imagen para mejorar el contraste (ecuaciones 1 y 2). Un histograma de imagen se forma tabulando el número de veces que cada intensidad se produce en toda la imagen [95].

$$p_r(r_k) = \frac{n_k}{MN}; \tag{B.3}$$

$$T(r_k) = (L-1)\sum_{j=0}^{k} p_r(r_j) | k = 1,2,3,...,L-1, \tag{B.4}$$

donde p_r es la función de densidad de probabilidad de f; n_k denota el número de píxeles que tienen intensidad k; MN es el número total de píxeles de la imagen; L es el número de niveles de intensidad de píxeles de la imagen. La aplicación de esta operación transforma el histograma en un histograma con una forma perfectamente uniforme en todos los niveles de gris. Durante la transformación, todos los píxeles de un nivel de gris se convierten en otro nivel de gris y el histograma se distribuye por toda el área disponible, separando al máximo las ocupaciones de los distintos niveles.

Paso 2. Aplicación de la función de suavizado gaussiano. Sea $I(x,y,z)$ la imagen original en un RGB, y $G(x,y)$ una función gaussiana definida como:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{B.5}$$

donde x es la distancia desde el origen del eje horizontal, y es la distancia desde el origen en el eje vertical, y σ es la desviación típica de la distribución gaussiana.

B.2.4. Pre-segmentación

La presegmentación es una etapa en la que se pueden aplicar distintas técnicas para aproximar las coordenadas en las que se encuentra aproximadamente el objeto de interés dentro de una imagen. Existen varios trabajos en la literatura que utilizan bounding boxes para determinar la posición de los objetos de interés, siendo YOLO uno de los métodos más utilizados. YOLO [96] es una red neuronal convolucional para localización de objetos, muy rápida para aplicaciones en tiempo real, y que cuenta con varias versiones. La arquitectura de YOLOV3 [97] se compone de dos procesos principales: un extractor de características llamado Darknet-53 y un método Convolucional de la propia detección, la Figura 2.4 muestra un diagrama de bloques YOLOV3.

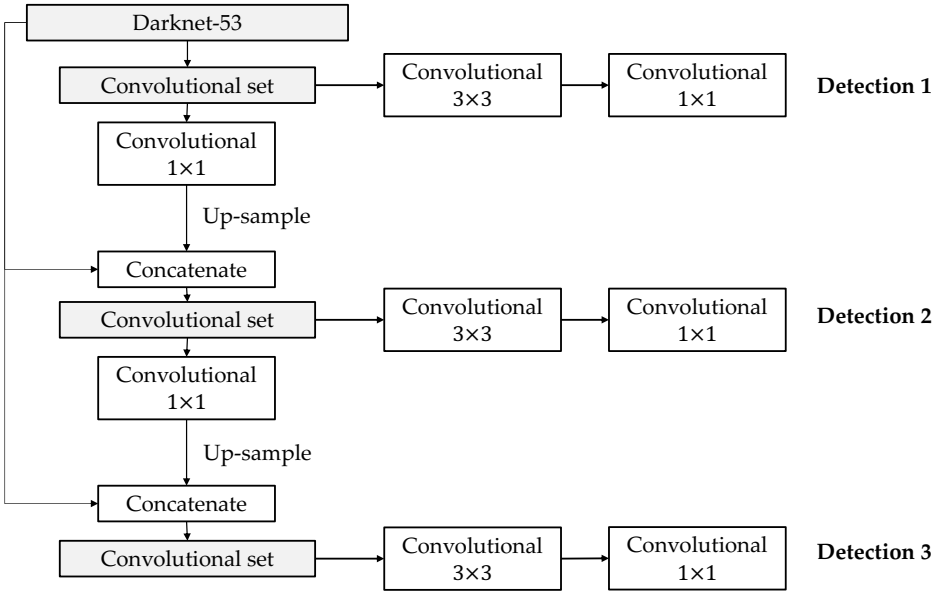

Figura 2.4: *Architecture YOLOV3.*

La figura 2.5, presenta la Darknet-53; que es una CNN con 53 capas de profundidad organizadas en

cinco bloques de capas de convolución donde cada capa es un extractor de características. El último bloque de capas de convolución contiene la información más importante obtenida de esta CNN, que se utiliza para extraer tres detecciones de diferentes escalas.

	Type	Filters	Size	Output	
	Convolutional	32	3 × 3	256 × 256	
	Convolutional	64	3 × 3 / 2	128 × 128	
1×	Convolutional	32	1 × 1		
	Convolutional	64	3 × 3		
	Residual			128 × 128	
	Convolutional	128	3 × 3 / 2	64 × 64	
2×	Convolutional	64	1 × 1		
	Convolutional	128	3 × 3		
	Residual			64 × 64	
	Convolutional	256	3 × 3 / 2	32 × 32	
8×	Convolutional	128	1 × 1		
	Convolutional	256	3 × 3		→ Detection 1
	Residual			32 × 32	
	Convolutional	512	3 × 3 / 2	16 × 16	
8×	Convolutional	256	1 × 1		
	Convolutional	512	3 × 3		→ Detection 2
	Residual			16 × 16	
	Convolutional	1024	3 × 3 / 2	8 × 8	
4×	Convolutional	512	1 × 1		
	Convolutional	1024	3 × 3		→ Detection 3
	Residual			8 × 8	
	Avgpool		Global		
	Connected		1000		
	Softmax				

Figura 2.5: *Architectura Darknet53.*

Un conjunto convolucional es un proceso para cambiar la dimensionalidad de las salidas de Darknet-53, que provienen de los tres últimos bloques. La figura 2.6 muestra un flujo de conjunto convolucional, que consiste en una secuencia de dos filtros convolucionales: 1x1 y 3x3. Un filtro convolucional 1x1 permite obtener un mapa de características con una única dimensión AnchoxAltox1. Normalmente, este filtro se aplica antes de un filtro de expansión: convolución 3x3 o convolución 5x5.

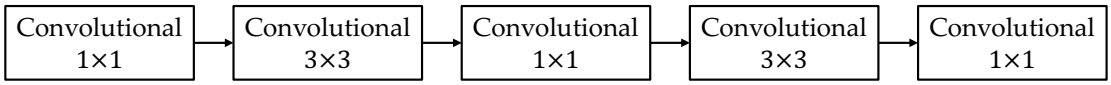

Figura 2.6: *Convolutional Set.*

YOLOV3 [98] predice un valor objetivo para cada cuadro delimitador mediante regresión logística. La

predicción del cuadro delimitador consta de 5 componentes, como vemos en la ecuación B.6.

$$y = (x_1, y_1, x_2, y_2, confidence) \tag{B.6}$$

donde las coordenadas (x_1, y_1, x_2, y_2) representan el centro de la caja relativo a la ubicación de la celda de la cuadrícula. Estas coordenadas están normalizadas entre 0 y 1. El valor de confianza indica la probabilidad de que la caja contenga un objeto y la precisión de la caja delimitadora. La fase de presegmentación es una etapa crítica para obtener resultados de segmentación precisos. A efectos de comparación, debe incluirse en la fase de resultados.

B.2.5. Segmentación

La fase de segmentación pretende refinar o ajustar la región delimitada por las coordenadas de la presegmentación para seleccionar una región de aves y no aves. La Figura 2.7-a muestra un ejemplo de fase de segmentación propuesta para delimitar regiones de aves y no aves a partir de las coordenadas obtenidas por la presegmentación. El ajuste de las coordenadas de presegmentación se define mediante dos configuraciones:

Configuración 1: Las coordenadas de presegmentación se reducen en un 50 %. Los píxeles dentro del rango de la Configuración 1 se clasifican como aves (Figura 2.7-b). Las coordenadas de la Configuración 1 se describen a continuación:

Dado un vector de coordenadas, Ecuación 4, con valores $[x_1, y_1, x_2, y_2]$, se determina la anchura de la región $w_x = x_2 - x_1$ y la altura de la región $h_y = y_2 - y_1$, y la región del ave se define en las ecuaciones 5, 6 y 7:

$$x_1^{'} = x_1 + \frac{w_x}{4}; x_2^{'} = x_2 - \frac{w_x}{4} \tag{B.7}$$

$$y_1^{'} = y_1 + \frac{h_y}{4}; y_2^{'} = y_2 - \frac{y_x}{4} \tag{B.8}$$

$$w_x^{'} = \frac{w_x}{2}; h_y^{'} = \frac{h_y}{2} \tag{B.9}$$

donde (x_1, x_2) y (y_1, y_2) son las coordenadas desde el origen $(0,0)$ en los ejes horizontal y vertical, respectivamente; $(x_1^{'}, x_2^{'})$ y $(y_1^{'}, y_2^{'})$ son las nuevas coordenadas en los ejes horizontal y vertical.

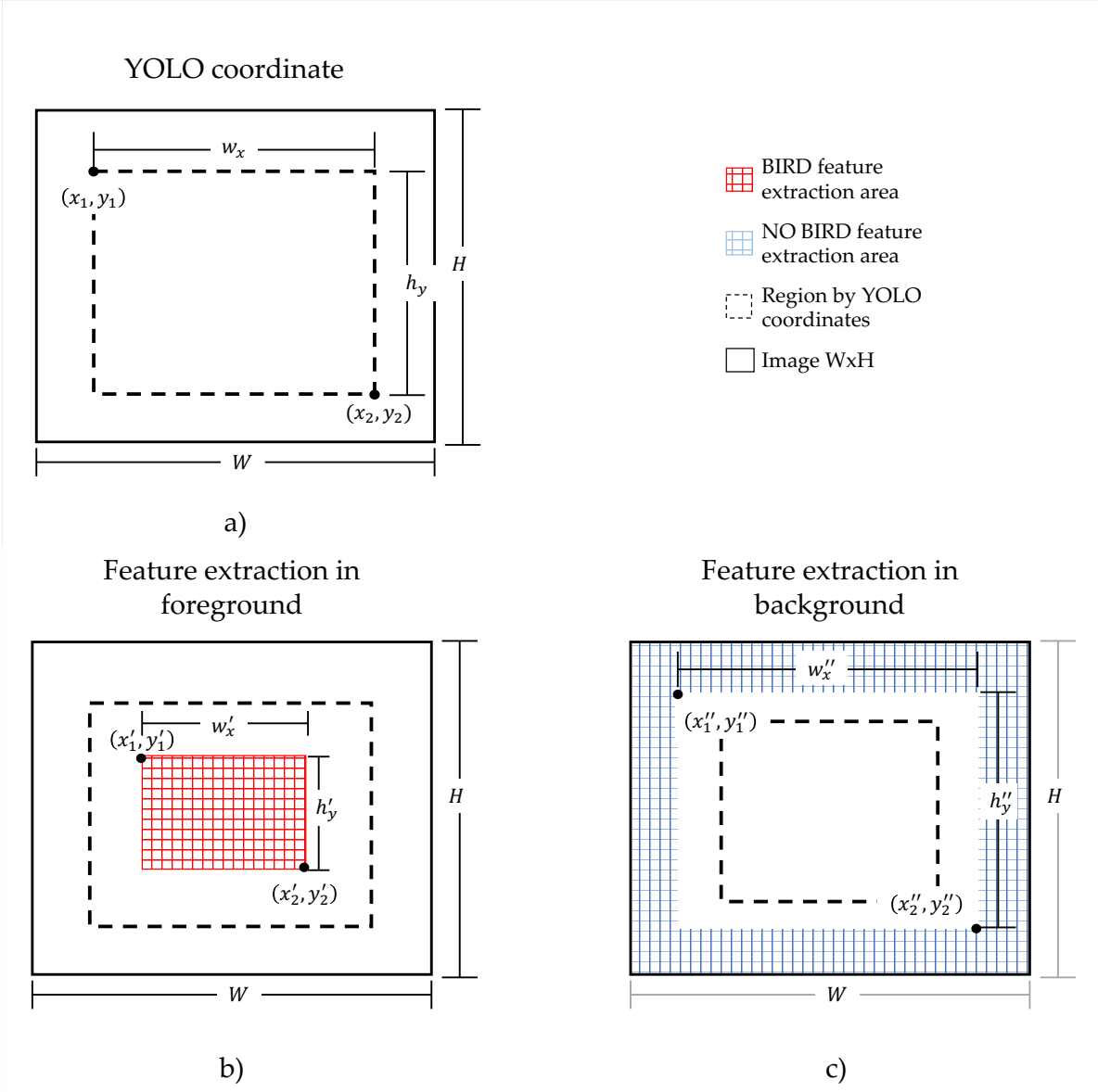

Figura 2.7: *The region selected as bird and non-bird: (a) YOLOV3 coordinates; (b) the provisional region as a bird; (c) the provisional region as non-bird.*

Configuración 2: las coordenadas de presegmentación aumentan un 20%. Los píxeles fuera de la Configuración 2 se clasifican como no-pájaros, Figura 7-c. A continuación se describen las coordenadas de la Configuración 1: Dado un vector de coordenadas, Ecuación 4, con valores $[x_1, x_2, y_1, y_2]$, se determina la anchura de la región $w_x = x_2 - x_1$, y la altura de la región $h_y = y_2 - y_1$, y la región del no-pájaro se define en las ecuaciones 8 a 10 con las coordenadas (x''_1, x''_2) y (y''_1, y''_2):

$$x_1^{''} = x_1 + \frac{w_x}{4}; x_2^{''} = x_2 - \frac{w_x}{4} \tag{B.10}$$

$$y_1^{''} = y_1 + \frac{h_y}{4}; y_2^{''} = y_2 - \frac{h_x}{4} \tag{B.11}$$

$$w_x^{''} = w_x + \frac{w_x}{2}; h_y^{''} = h_y + \frac{h_y}{2} \tag{B.12}$$

donde (x_1, x_2), (y_1, y_2), $(x_1^{'}, x_2^{'})$, y $(y_1^{'}, y_2^{'})$ se definieron previamente para las ecuaciones 5 a 7. Las regiones definidas en las ecuaciones 5 a 10 son los píxeles que se activan para la extracción de características. Los píxeles entre las regiones de aves y no aves no se consideran en la fase de extracción de características. La etiqueta de un vector de características se asigna en función de la región en la que se encuentra.

B.2.6. Extracción de características

Las características de color se extraen de regiones de 15x15 píxeles, denominadas superpíxel, un conjunto de imágenes suavizadas y mejoradas. Las características de color se refieren al comportamiento estadístico de las regiones en cada canal de los modelos de color. Los modelos de color se seleccionaron de acuerdo con el estado actual de la técnica y son HSI, CMYK, LAB y XYZ. La varianza y la desviación estándar son las características extraídas para cada canal.

Las características de textura de Haralick [1] son descriptores de textura comunes en el análisis de imágenes basados en el concepto de que la textura y el tono están relacionados. Las características se determinan utilizando una matriz de correlación de los niveles de intensidad de una imagen, la matriz de co-ocurrencia de niveles de gris (GLCM). El número de niveles de gris de la imagen determina el

tamaño de la GLCM. La figura 8 muestra cómo se determina la GLCM [99].

La GLCM comienza con la transformación de una imagen original en RGB a una imagen en escala de grises, representada en la Figura 2.8a,b. En el segundo paso, se crea una matriz de ocurrencia M(i, j), es decir, los valores en las posiciones en (i, j) representan el número de veces que el valor de intensidad de nivel de gris i es vecino del valor de intensidad de nivel de gris j, como mostramos en la Figura 2.8-c. Tras obtener la matriz de ocurrencia, M(i, j), los valores (i, j) se normalizan, como se muestra en la Figura 2.8-d. Finalmente, la matriz resultante p(i, j) es adecuada para la aplicación de las características de textura Haralick (Figura 8e). La Tabla 2 muestra la notación de las variables que intervienen en el cálculo de las características de textura Haralick. La primera columna representa el número de la variable; la segunda columna muestra la notación de las variables; y la tercera columna describe el significado de la notación.

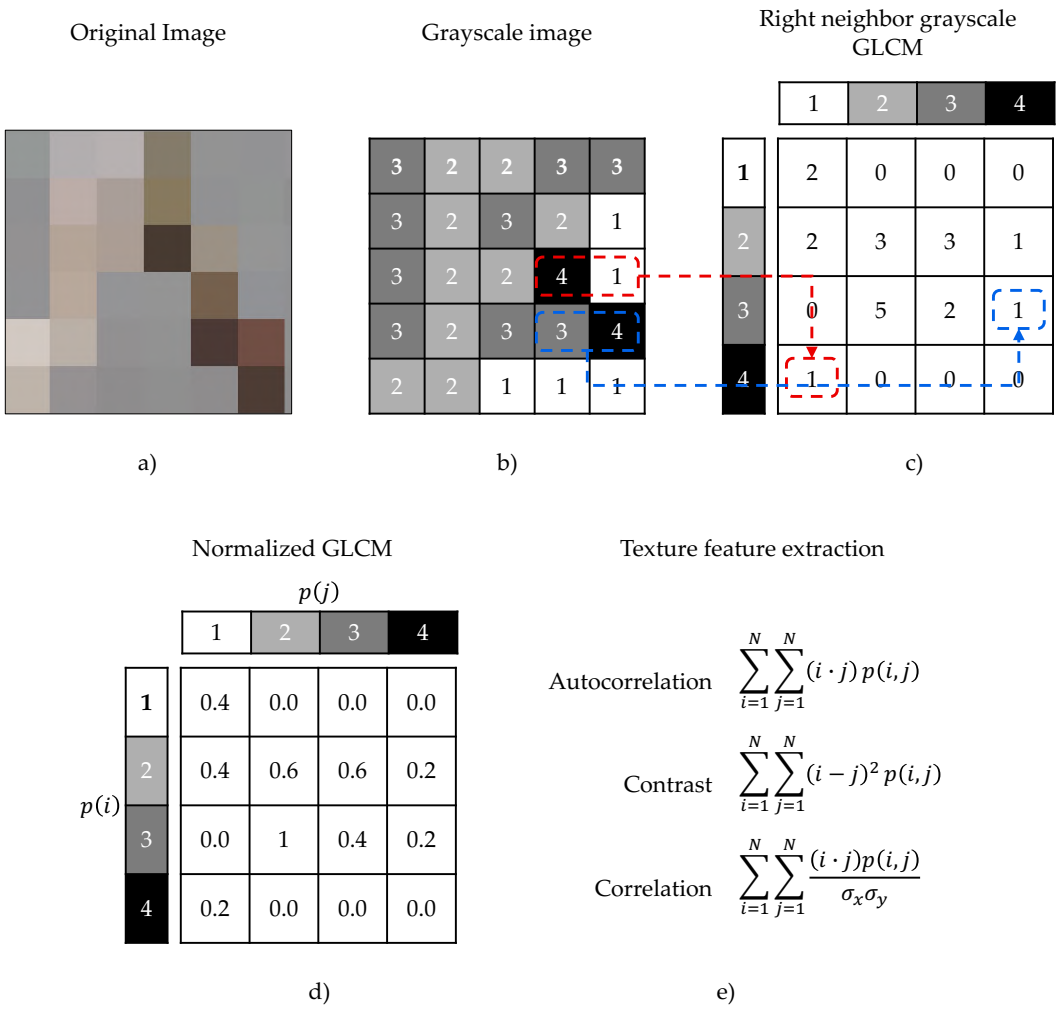

Figura 2.8: *Procedimiento GLCM para determinar la matriz de co-ocurrencia de niveles de intensidad de gris. (a) Imagen RGB, ((b) niveles de intensidad de gris de la imagen RGB, (c) matriz de co-ocurrencia GLCM de los niveles de intensidad de gris, (d) matriz GLCM normalizada entre 0 y 1, () ecuaciones de textura extraídas de la matriz GLCM normalizada.*

Tabla 2.2: *Notación del cálculo de las características de textura Haralick. Fuente: 33.*

Num.	Notation	Meaning	Description
1	$p(i,j)$	Values i,j in the normalized GLCM	
2	N	Number of gray levels	
3	$p_x(i)$	$\sum_{j=1}^{N} p(i,j)$	
4	$p_y(j)$	$\sum_{i=1}^{N} p(i,j)$	
5	μ_x	$\sum_{i=1}^{N} i \cdot p_x(i)$	
6	μ_y	$\sum_{j=1}^{N} j \cdot p_y(j)$	
7	σ_x^2	$\sum_{i=1}^{N} (i-\mu_x)^2 \cdot p_x(i)$	
8	σ_y^2	$\sum_{j=1}^{N} (j-\mu_y)^2 \cdot p_y(j)$	
9	$p_{x+y}(k)$	$\sum_{i=1}^{N}\sum_{j=1}^{N} p(i,j)^2 \Big\vert i+j=k$	
10	$p_{x-y}(k)$	$\sum_{i=1}^{N}\sum_{j=1}^{N} p(i,j)^2 \Big\vert \lvert i-j\rvert=k$	
11	HX	$-\sum_{i=1}^{N} p_x(i) \cdot log p_x(i)$	Used for determining 12 and 13 equations in this work.
12	HY	$-\sum_{i=1}^{N} p_y(i) \cdot log p_y(i)$	
13	HXY	$-\sum_{i=1}^{N} p(i,j) \cdot log p(i,j)$	
14	$HXY1$	$-\sum_{i=1}^{N}\sum_{j=1}^{N} p(i,j) \cdot log\lvert p_x(i) \cdot p_y(j)\rvert$	
15	$HXY2$	$-\sum_{i=1}^{N}\sum_{j=1}^{N} p_x(i) \cdot p_y(j) \cdot log\lvert p_x(i) \cdot p_y(j)\rvert$	

En la tabla 3 se enumeran todas las características de textura utilizadas en este trabajo. La primera columna representa el número de características; la segunda columna es una característica de nombre; en la tercera columna, hemos dado una ecuación para la característica de nombre.

Tabla 2.3: *Características de textura Haralick usadas en estra trabajo. [1–3].*

Num.	Feature Name	Equation
1	Autocorrelation [34]	$\sum_{i=1}^{N}\sum_{j=1}^{N}(i\cdot j)p(i,j)$
2	Cluster prominence [32]	$\sum_{i=1}^{N}\sum_{j=1}^{N}(i+j-2\mu)^3 p(i,j)$
3	Cluster shadow [32]	$\sum_{i=1}^{N}\sum_{j=1}^{N}(i+j-2\mu)^4 p(i,j)$
4	Contrast [32]	$\sum_{i=1}^{N}\sum_{j=1}^{N}(i-j)^2 p(i,j)$
5	Correlation [32]	$\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{(i\cdot j)p(i,j)}{\sigma_x\sigma_y}$
6	Difference entropy [32]	$-\sum_{k=0}^{N-1}p_{x-y}(k)logp(k)$
7	Difference variance [32]	$\sum_{k=0}^{N-1}(k-\mu_{x-y})^2 p_{x-y}(k)$
8	Dissimilarity [32]	$\sum_{i=1}^{N}\sum_{j=1}^{N}\lvert i-j\rvert\cdot p(i,j)$
9	Energy [32]	$\sum_{i=1}^{N}\sum_{j=1}^{N}p(i,j)^2$
10	Entropy [32]	$\sum_{i=1}^{N}\sum_{j=1}^{N}p(i,j)logp(i,j)$
11	Homogeneity [34]	$\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{p(i,j)}{1+(i+j)^2}$
12	Information measure of correlation 1 [32]	$\frac{HXY-HXY1}{\max(HX,HY)}$
13	Information measure of correlation 2 [32]	$\sum_{i=1}^{N}\sum_{j=1}^{N}\sqrt{1-exp[-2(HXY2-HXY)]}$
14	Inverse difference [35]	$\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{p(i,j)}{1+\lvert i-j\rvert}$
15	Maximum probability [32]	$maxp(i,j)$
16	Sum average μ_{x+y} [32]	$\sum_{k=2}^{2N}kp_{x+y}(k)$
17	Sum entropy [32]	$-\sum_{k=2}^{2N}p_{x+y}(k)logp_{x+y}(k)$
18	Sum square [32]	$\sum_{i=1}^{N}\sum_{j=1}^{N}(i-\mu)^2 p(i,j)$
19	Sum variance [32]	$\sum_{k=2}^{2N}(k-\mu_{x+y})^2 p_{x+y}(k)$

B.2.7. Variable feature selector

El algoritmo SA fue propuesto por Kirkpatrick en 1983 [17]. SA representa el proceso termodinámico de calentamiento y enfriamiento de un metal para aumentar su ductilidad y es un método de optimización para encontrar soluciones casi óptimas a problemas combinatorios no deterministas de dureza polinómica en tiempo (NP-hardness) [100].

Según trabajos relacionados, la técnica PCA se utiliza a menudo como selector de variables relevantes. Esta técnica consiste en describir los datos en términos de nuevas variables denominadas componentes. Los componentes se ordenan según su varianza explicada, que representa el porcentaje de retención de la información original. Sin embargo, cada componente está compuesto por una combinación lineal de todas las variables originales. Por lo tanto, puede decirse que PCA es un reductor de dimensionalidad y no un método de selección de variables.

Para resolver el problema, se desarrolló un algoritmo híbrido basado en la técnica del Recocido Simulado (SA) y el Análisis de Componentes Principales (PCA), que se denomina SA-PCA. SA-PCA tiene como solución un vector binario con una longitud de 43, que es el número de descriptores que Color y Textura presentan en este trabajo. La solución inicial la establece PCA a partir de la contribución porcentual de las variables componentes con mayor porcentaje de varianza explicada. La figura 9 muestra un ejemplo de la represen-tación de la solución inicial para este trabajo. Los valores 0 y 1 indican si una variable ha sido seleccionada.

S_i	0	0	1	1	0	0	1	1	1	⋯	1	1	0
	1	2	3	4	5	6	7	8	9	⋯	$n-2$	$n-1$	n

Figura 2.9: *Representación de la solucion inicial S_i en SA-PCA.*

La definición de los parámetros del SA-PCA se sometió a un proceso de ajuste [100], que se analiza con más detalle en la sección 4.4. El Algoritmo 1 muestra el algoritmo SA-PCA propuesto, que se basa en el Recocido Simulado Kirkpatric [17]. En primer lugar, las líneas 2-5 definen la solución inicial S_i y

la función objetivo E_new asociada a esta solución. Se define como la mejor solución encontrada hasta el momento. La línea 6 verifica que la mejor solución encontrada hasta el momento ha alcanzado el valor mínimo.

SA-PCA se define con dos ciclos principales (líneas 7 y 8). Aquí se comprueba tradicionalmente si la temperatura inicial T_i ha alcanzado la temperatura final T_f y si el ciclo Metropolis ha alcanzado su longitud máxima L_max o si existe un estado de convergencia. La temperatura T_i se ajustará mediante el parámetro α, línea 29. El ciclo interno realiza una búsqueda de una nueva solución X_new hasta que un equilibrio estocástico L_{max} se alcanza en cada temperatura baja por el parámetro β. L_max se ajusta por el parámetro β en la línea 30. Este algoritmo permite la aceptación de malas soluciones por el criterio de aceptación de Boltzmann en la línea 25.

Algorithm 6 Recocido Simulado propuesto

0: **function** SIMULATEDANNEALING($T_i, T_f, \beta, \alpha, L_{max}$)
0: $X_{old} \leftarrow solution()$
0: $X_{best} \leftarrow X_i$
0: $E_{old} \leftarrow objFunction()$
0: $E_{best} \leftarrow E_{old}$
0: **if** $E_{best} \neq 0$ **then**
0: **while** $T_i > T_f$ And $\neg converge$ **do**
0: **while** $L < L_{max}$ And $\neg converge$ **do**
0: $X_{new} \leftarrow perturbation_{roullete}(X_{old})$
0: $E_{new} \leftarrow objFunction(X_{new})$
0: $\Delta E \leftarrow E_{new} - E_{old}$
0: **if** $E_{new} = \varepsilon$ **then**
0: $converge$
0: **end if**
0: **if** $converge(metropoly)$ **then**
0: $converge$
0: **end if**
0: **if** $\Delta E \leq 0$ **then**
0: $X_{old} \leftarrow X_{new}$
0: $E_{old} \leftarrow E_{new}$
0: **if** $E_{old} < E_{best}$ **then**
0: $X_{best} \leftarrow X_{old}$
0: $E_{best} \leftarrow E_{old}$
0: **end if**
0: **else if** RANDOM$(0,1) > e^{-\Delta E/T_i}$ **then**
0: $X_{old} \leftarrow X_{new}$
0: $E_{old} \leftarrow E_{new}$
0: **end if**
0: $T_i \leftarrow \alpha T_i$
0: $L_{max} \leftarrow \beta L_{max}$
0:
0: **end while**
0: **if** $T_i \geq 0.95 T_f$ **then**
0: **if** $converge(Temp)$ **then**
0: $converge$
0: **end if**
0: **end if**
0:
0: **end whilereturn** X_{best}, E_{best}
0: **end if**
0: **end function**=0

El algoritmo SA-PCA tiene una fase de perturbación denominada $perturbation_{roullete}$, que incluye un método de ruleta con el propósito de aumentar la probabilidad de selección en aquellas variables que han formado parte de buenas soluciones en los ciclos anteriores (línea 9). El criterio de aceptación de una solución viene dado por el cambio en el valor de la función objetivo entre la solución real, E_{old}, y la nueva solución, E_{new}, es decir, $\Delta E = E_{new} - E_{old}$, donde se acepta si $\Delta E \leq 0$, como se ve en las líneas 11, 18-24. En caso contrario, se aplica la distribución de Boltzmann-Gibbs [101], que es un mecanismo de decisión o probabilidad, para determinar si la mala solución se acepta aleatoriamente.

La convergencia del algoritmo se define en dos casos de estados estables: Alcanzar el valor mínimo de la función objetivo y estancamiento. El estancamiento se define por r repeticiones sucesivas del valor de la función objetivo de la nueva solución E_{new}, el parámetro $r = 5$. El criterio de convergencia en las líneas 33-35 significa que existe convergencia si la temperatura inicial T_i está dentro del 5 % de la temperatura final T_f.

B.2.8. Clasificación

Un Bosque Aleatorio (RF) es un algoritmo utilizado habitualmente para la clasificación que se compone de varios clasificadores de árboles de decisión, que utiliza el rendimiento medio del conjunto de clasificadores para mejorar la precisión de la predicción, con el objetivo de optimizar el conjunto. Dado que los árboles individuales son perturbados aleatoriamente, el bosque se beneficia de una exploración más amplia del espacio de todos los posibles predictores del árbol, lo que, en la práctica, se traduce en un mejor rendimiento predictivo [102]. Los aspectos más importantes a considerar en un RF son el número de árboles de decisión en el bosque M, la función para medir la calidad de la predicción y la profundidad máxima de los árboles de decisión. Un árbol de decisión con M hojas divide el espacio de características en M regiones $R_m, 1 \leq m \leq M$ [103]. Para cada árbol, la función de predicción $f(x)$ se define como:

$$f(x) = \sum_{m=1}^{M} c_m \prod (x, R_m) \tag{B.13}$$

$$\prod(x, R_m) = \begin{cases} 1, & \text{if } x \in R_m \\ 0, & \text{otherwise} \end{cases} \tag{B.14}$$

donde M es el número de regiones en el espacio de características, R_m es una región adecuada a m; c_m es una constante adecuada a m en la ecuación 12. Los hiperparámetros del clasificador RF se tratan con más detalle en la sección 4.3.

B.3. CONFIGURACIÓN EXPERIMENTAL

B.3.1. Datos

Para este trabajo se utilizaron 263 imágenes de aves. El conjunto de imágenes se dividió en 193 para el entrenamiento y 70 para las pruebas. El conjunto de imágenes se clasificó por objetos de aves grandes y medianas. La clasificación se basa en los criterios de evaluación de las competencias de la base de datos COCO [4], en la que se define el área del objeto en píxeles:

- objects medianos: (32x32, 96x96) pixeles
- objectos grandes: desde 96x96 pixeles

B.3.2. Métricas

La métrica utilizada para evaluar el rendimiento del modelo es la Intersección de Precisión Media sobre Unión (APIoU), que se muestra en la Ecuación B.15.

$$APIoU = \sum_{(} i = 1)^{m} \frac{TP_i}{FP_i + TP_i} \tag{B.15}$$

donde m es el número de imágenes, TP son los verdaderos positivos, y FP son los falsos positivos para la imagen i. El primer umbral $APIoU$ va de $0,05$ a $0,95$; mientras que el segundo $APIoU$ va de $0,75$ a $0,95$, denominado $APIoU^{75}$.

B.3.3. Configuración del clasificador

La selección del clasificador se realizó para comparar el rendimiento de clasificación de RandomForest y Multi-Perceptron utilizando la herramienta WEKA. Los clasificadores se seleccionaron basándose en los trabajos relacionados. Las observaciones utilizadas constan de dos tamaños de vector. El primero consta de 43 características y una etiqueta, y el segundo consta de 14 características y una etiqueta. Este último se obtiene mediante la fase de selección de las variables relevantes.

La tabla 2.4 muestra los resultados obtenidos de la clasificación correcta e incorrecta de las observaciones. El conjunto de observaciones se dividió en tres conjuntos diferentes: conjunto de entrenamiento utilizado, validación cruzada y división 70 %-30 %. El conjunto de entrenamiento utilizado construye el clasificador a partir de todas las observaciones y vuelve a aplicar todas esas observaciones al clasificador. La validación cruzada divide los datos en 10 conjuntos (normalmente) de igual tamaño, cada conjunto se divide en entrenamiento y prueba. Construye un clasificador utilizando los datos de entrenamiento de cada conjunto, que se aplica a los datos de prueba de cada conjunto para obtener un rendimiento medio. Dividir 70 %-30 % es dividir los datos en entrenamiento y prueba, construir el clasificador con los datos de entrenamiento y medir el rendimiento con los datos de prueba. La tabla 2.4 muestra que Random Forest obtuvo un mejor rendimiento de clasificación en los tres conjuntos con división de datos.

Tabla 2.4: *Resultados de los clasificadores mediante el software WEKA.*

Clasificadores	**Particion de datos**	**Total instancias**	**Instancias Correctamente Clasificadas**	**Instancias Incorrectamente Clasificadas**
Random Forest	Use training set	16988	16978(99.94 %)	10(0.05 %)
	Cross-validation	16988	12892(75 %)	4096(24.11 %)
	Split 70 %-30 %	5096(30 %)	3853 (75.6 %)	1243 (24.4 %)
MLP	Use training set	16988	11963 (70.4 %)	5025 (29.5 %)
	Cross-validation	16988	11649 (68.5 %)	5339 (31.4 %)
	Split 70 %-30 %	5096(30 %)	3494 (68.5 %)	1602 (31.4 %)

RF es uno de los métodos de clasificación y regresión del aprendizaje automático. La Tabla 5 muestra los parámetros del clasificador Random Forest. El método RF se ejecutó utilizando la biblioteca Sklearn. La sintonización se sometió al método de búsqueda de cuadrícula aleatoria. La primera columna enumera los parámetros obtenidos a los que se asignó una configuración distinta de los valores por defecto. La segunda columna muestra los valores asignados a los parámetros. La tercera columna describe brevemente cada uno de los parámetros.

Tabla 2.5: *Hyperparametros de RF.*

Parámetros	**Valor**	**Descripción**
n_estimators	1400	El número de arboles en el bosque
max_depth	80	La profundidad máxima de los arboles.
max_feature	auto	El número de característica a considerar cuando buscamos la mejor particion: $max\_features = sqrt(n\_features)$

B.3.4. Algoritmo de mejora SA ajustado

El algoritmo SA se utiliza como método de optimización aleatoria para encontrar un subconjunto de características (variables) que funcione mejor que las 43 características originales.

Los parámetros de este modelo son los siguientes Temperatura inicial $T_i = 1570,29$, temperatura final $T_f = 0,01$, longitud de la metrópolis $L_{max} = 198$, $\alpha = 0,95$, $\beta = 1,02$, tasa de perturbación $= 0,10$.

B.3.5. Características de Color y Textura

En este trabajo se extrajeron un total de 43 características, 26 correspondientes al color y 17 a la textura. Las características de color son dos medidas de tendencia central: desviación estándar y varianza. Los modelos de color son HSI, CMYK, LAB y XYZ extrayéndose las características de color para cada canal. Las características de textura se obtienen generando la matriz GLCM.

La tabla 2.6 enumera las características seleccionadas por las técnicas PCA y SA, siendo esta última el algoritmo de mejora propuesto. La primera columna muestra el método, la segunda y la tercera las características de color y textura seleccionadas por cada método, y la cuarta las características totales de cada método.

Tabla 2.6: *Caractrerísticas seleccionadas por PCA y SA.*

Método	**Caract. de color**	**Caract. de Textura**	**Total de Caract.**
PCA	std_S, var_S, std_Y_cmyk, std_K, var_K, std_L, var_L, var_A, var_B_lab, std_X, std_Z, var_X, var_Y_xyz, var_Z	-	14
Algoritmo de mejora de SA	std_H , std_S, var_I, std_M, var_C, var_K, std_L, std_A, var_A, var_Y_xyz, var_z	Correlation, diferen-ce_entropy, diferen-ce_variance	14

B.4. RESULTADOS

Probamos nuestra metodología PSEV-BF con un conjunto de datos de la base COCO. Evaluamos nuestra metodología propuesta con las métricas medias de segmentación semántica de APIoU para objetos medianos y grandes. En esta sección, mostramos los resultados obtenidos en la fase de sintonización y selección de variables relevantes para SA, así como el rendimiento del proceso de Presegmentación y

Clasificación del modelo.

En el estudio del mejor modelo, se implementaron y probaron dos fases distintas en visión por computador. La primera fase, Pre-segmentación, fue una arquitectura CNN presentada en la Sección 3.1.2., consistente en localizar las regiones donde se encuentra la ROI; para ello se utilizó YOLOV3. Las coordenadas proporcionadas por YOLOV3 permiten determinar la región donde se podría localizar el objeto de interés, lo cual se realiza durante la fase de segmentación.

Utilizamos el algoritmo Simulated Annealing (SA) como selector de variables relevantes para mejorar la fase de entrenamiento en el proceso de clasificación. SA-PCA utilizó un clasificador Random Forest. SA configura la solución inicial utilizando las variables obtenidas por PCA. También utiliza un método de perturbación mediante una ruleta. En la Tabla 2.7, observamos las soluciones de SA-PCA propuestas. La primera columna indica el número de ejecuciones, la segunda columna muestra el número de características seleccionadas en cada ejecución, y la tercera columna es la función objetivo asociada al número de variables seleccionadas. La elección de una solución, es decir, un conjunto de variables, obtenida por el SA-PCA viene definida por el tamaño de la solución y la función objetivo, siendo esta última la más importante. Así, la Tabla 7 muestra que utilizando 14 variables se obtiene una función objetivo con una menor tasa de error. Por lo tanto, estas características se utilizan como variables relevantes.

Tabla 2.7: *Resultados del algoritmo SA-PCA con RF.*

Número	Número de variables seleccionadas	Funcion Objetivo
1	16	28.02
2	18	28.14
3	**14**	**25.12**
4	11	27.85
5	13	29.30
6	12	39.03
7	14	32.06
8	10	32.97
9	10	36.97
10	17	32.36
11	16	30.89
12	18	27.96
13	18	28.38
14	17	28.39
15	11	34.38
Minimal	10	25.12
Maximal	18	39.03
Standard Deviations	3.02	3.83

Los rendimientos de clasificación para dos grupos de tamaños de aves se presentan en la Tabla 2.4. Observamos el rendimiento comparativo de la metodología propuesta con diferentes configuraciones: Metodología 1 (M1), Metodología 2 (M2), y Metodología 3 (M3). La configuración M1 aplica el proceso tradicional de Preprocesamiento, Clasificación, Evaluación y una técnica de superpíxel. M2 implica el mismo proceso tradicional pero no utiliza superpíxeles, aunque se implementa un método de selección de variables. M3 sólo implementa una fase de presegmentación con YOLOV3. Y por último, nuestra propuesta PSEV-BF incluye todas las configuraciones propuestas en este trabajo. Es importante aclarar que todas las metodologías utilizan la Presegmentación con YOLOV3 a efectos comparativos.

En la Tabla 2.8, la primera columna indica el tamaño de las aves utilizadas: Grandes o Medianas. La segunda columna enumera las diferentes metodologías etiquetadas como M1, M2, M3 y Nuestro PSEV-BF. La tercera y cuarta columnas indican con 51 si la técnica de superpíxeles se utiliza en la fase de presegmentación, segmentación o características mejoradas, y 55 en caso contrario. Finalmente, las dos últimas columnas son los resultados de APIoU con dos umbrales: 0,5 a 0,95 y 0,75 a 0,95.

Tabla 2.8: *Desempeño de la metodología PSEV-BF vs configuraciones alternativas.*

Tamaño objeto	**Método**	**SuperP.**	**Pre-S.**	**Seg.**	**Caract. Mejorada**	*APIoU*	$APIoU^{75}$
Large	PSEV-BF	51	51	51	51	0.5585	0.8614
	M1	51	51	55	55	0.5256	0.8235
	M2	55	51	51	51	0.5481	0.8347
	M3	55	51	55	55	0.4133	0.8459
Medium	PSEV-BF	51	51	51	51	0.3613	0.8097
	M1	51	51	55	55	0.3264	-
	M2	55	51	51	51	0.3325	0.8097
	M3	55	51	55	55	0.3483	0.8097

Super P.: super pixels; Pre-S: pre-segmentation; Seg: segmentation.; Caract.: Característica

La tabla 2.8 muestra que la metodología PSEV-BF propuesta para objetos grandes tiene valores en torno al 50 % con la métrica APIoU y al 80 % con la métrica APIoU75. Por otro lado, los objetos de tamaño medio tienen valores en torno al 36 % con la métrica APIoU y al 80 % con la métrica APIoU75 para las metodologías M2 y M1. Sin embargo, M1 para objetos de tamaño medio no obtuvo imágenes con un valor superior al umbral del 75 % de la métrica APIoU75 y no se calcularon. El tiempo medio de procesamiento obtenido por PSEV-BF en objetos grandes y medianos fue de 78,03 y 3,07 segundos, respectivamente. Asimismo, el tiempo de procesamiento de la metodología M1 para objetos grandes fue de 90,09 segundos, y para objetos medianos fue de 9,01. En el caso de las metodologías M2 y M3, no se incluye el superpixel, no se informa del tiempo de procesamiento porque el tiempo excede el tiempo máximo permitido para cada imagen.

En la Figura 2.10, presentamos el rendimiento de las diferentes configuraciones del método propuesto basado en la métrica APIoU con un umbral que comienza en 0,5 para objetos de tamaño grande y mediano y alcanza valores en torno al 50 %. En la Figura 2.10-a, mostramos que nuestra propuesta alcanza un rendimiento del 54 % de precisión para objetos grandes, con una diferencia de alrededor del 12 % con la metodología M3, que alcanza la precisión más baja. Las metodologías M2 y M3 obtuvieron valores de APIoU muy próximos a los de la metodología PSEV-BF. En ellas intervienen al menos dos de los procesos propuestos: Superpíxel y Presegmentación.

En la Figura 2.10-b, observamos que el PSEV-BF alcanza un rendimiento del 36 % de precisión para objetos de tamaño medio, con una diferencia de alrededor del 4 % respecto a la metodología M1, que alcanza la precisión más baja. La metodología M3 muestra valores cercanos a los del método propuesto, y éstos intervienen en la Presegmentación.

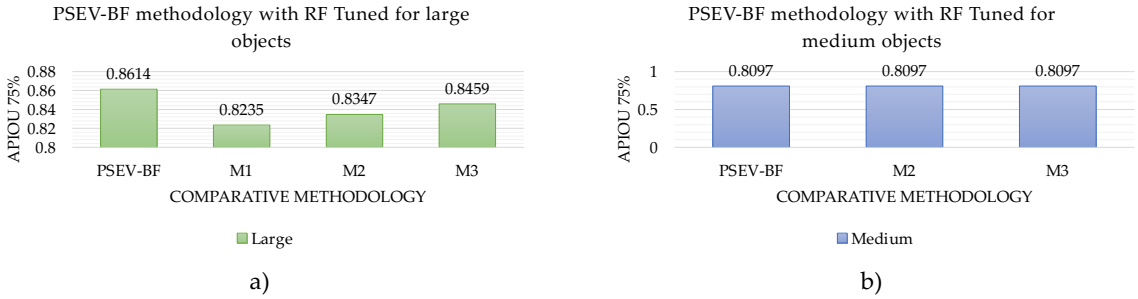

Figura 2.10: *Resultados de la metodología comparada con diferentes configuraciones basadas en la métrica APIoU para (a) objetos grandes y (b) objetos medianos..*

En la Figura 2.11, podemos observar el rendimiento de las diferentes configuraciones del método propuesto basado en la métrica APIoU con un umbral que comienza en 0,75 para objetos de tamaño grande y mediano y alcanza valores en torno al 80 %. La figura 2.11-a, muestra que nuestra propuesta alcanza un rendimiento del 86 % de precisión para objetos grandes, con una diferencia de alrededor del 4 % con la metodología M1, que alcanza la precisión más baja. La metodología M3 muestra valores muy próximos a los de la metodología PSEV-BF, que intervienen en al menos dos de los procesos propuestos: Superpíxel y Presegmentación.

En la Figura 2.11, presentamos el rendimiento de los métodos para objetos de tamaño medio y grande considerando aquellas imágenes que alcanzan un umbral igual o superior al 75 % ($APIoU^{75}$). En la Figura 2.11-b, observamos que la metodología PSEV-BF, M2 y M3 alcanza una precisión del 80 % para objetos de tamaño medio. Por el contrario, la metodología M1 no consigue obtener una precisión por encima de un umbral del 75 %.

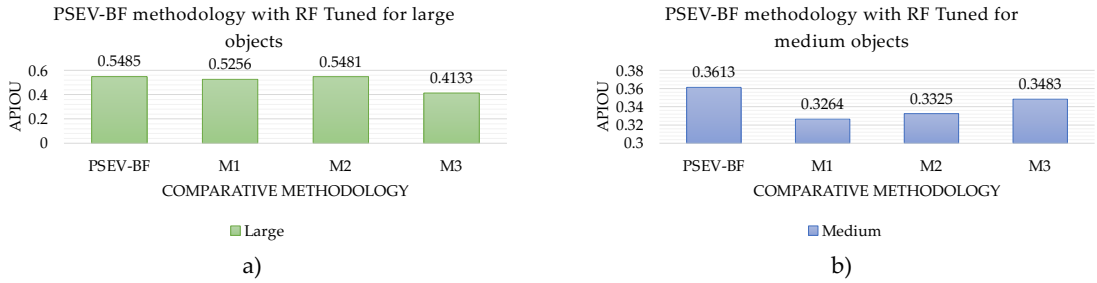

Figura 2.11: *Resultados de la metodología comparada con diferentes configuraciones basadas en la métrica $APIoU^{75}$ para (a) objetos grandes y (b) objetos medianos.*

La figura 2.12 muestra algunos ejemplos de objetos de gran tamaño procesados. La Figura 2.12-a muestra las imágenes segmentadas por COCO; la Figura 2.12-b muestra la adaptación resultante de la fase de segmentación. Por último, la Figura 2.12-c muestra algunos de los casos obtenidos mediante la metodología PSEV-BF. Observamos que los píxeles correspondientes a los no-pájaros (bloques negros) forman parte del fondo. Asimismo, alrededor del 86 % de los píxeles correspondientes a aves se clasificaron correctamente.

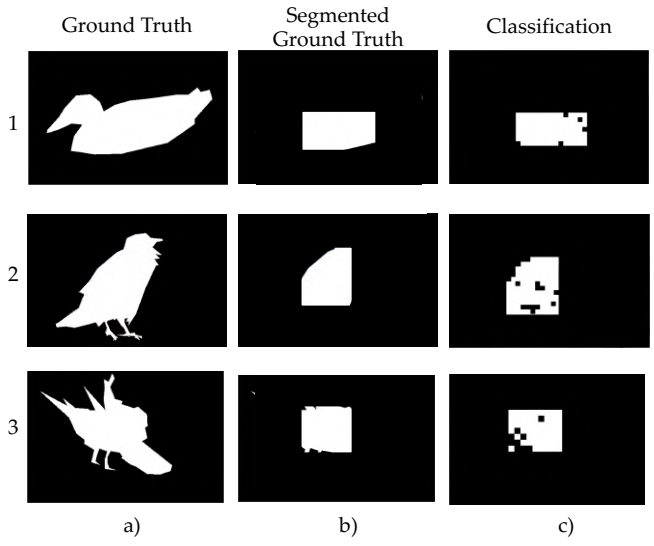

Figura 2.12: *Resultados comparativos de la metodología para tres imágenes de gran tamaño. (a) imagen segmentada original, (b) imagen segmentada mediante la segmentación propuesta, (c) clasificación de ventanas de 15por15 píxeles.*

La figura 2.13 muestra algunos ejemplos de objetos de Gran Tamaño procesados con oclusión. La figura 2.13-a muestra que los píxeles blancos corresponden a aves y los negros a no aves (o píxeles mal clasificados). Asimismo, alrededor del 86 % de los píxeles correspondientes a aves se clasificaron correctamente.

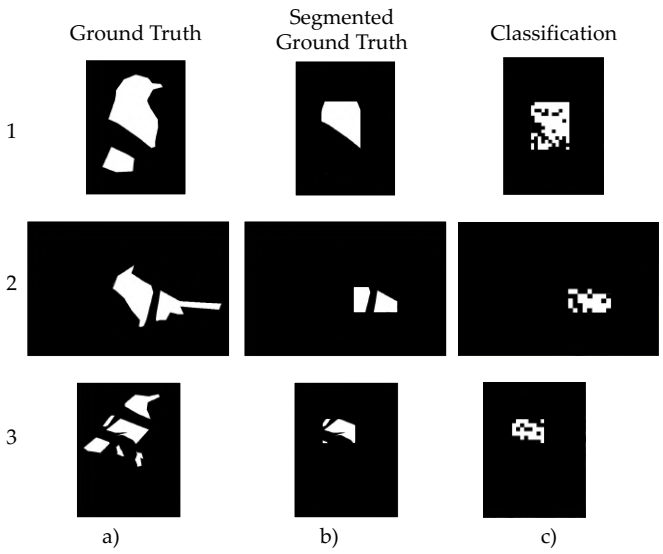

Figura 2.13: *Resultados comparativos de la metodología para tres imágenes grandes con oclusión. (a) imagen segmentada original, (b) imagen segmentada mediante la segmentación propuesta, (c) clasificación de ventanas de 15por*15 *píxeles.*

La figura 2.14 muestra algunos ejemplos de objetos de Tamaño Medio procesados por la metodología PSEV-BF. Observamos en la última columna, los píxeles correspondientes a aves (píxeles blancos) que fueron correctamente clasificados en la primera fila

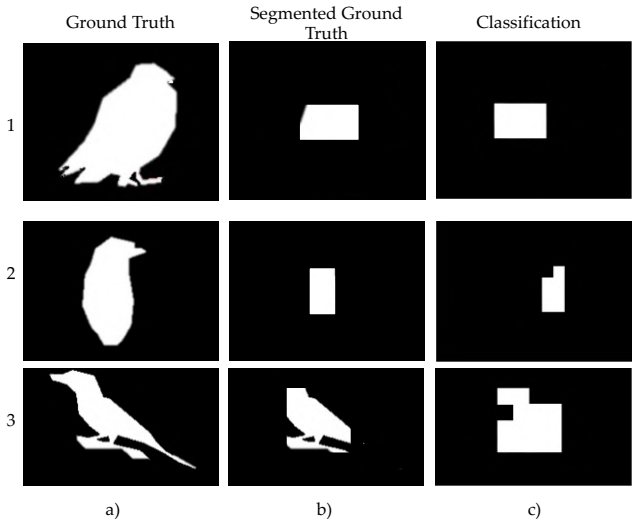

Figura 2.14: *Resultados de la metodología comparativa para tres imágenes medias. (a) imagen segmentada original, (b) imagen segmentada mediante la segmentación propuesta, (c) clasificación por ventanas de* 15×15 *píxeles. Nota: las imágenes se amplificaron para una mejor ilustración.*

Por último, la Figura 2.15 muestra algunos ejemplos de objetos de Tamaño Medio procesados con oclusión por la metodología PSEV-BF. Observamos en la última columna los píxeles correspondientes a pájaros fueron correctamente clasificados.

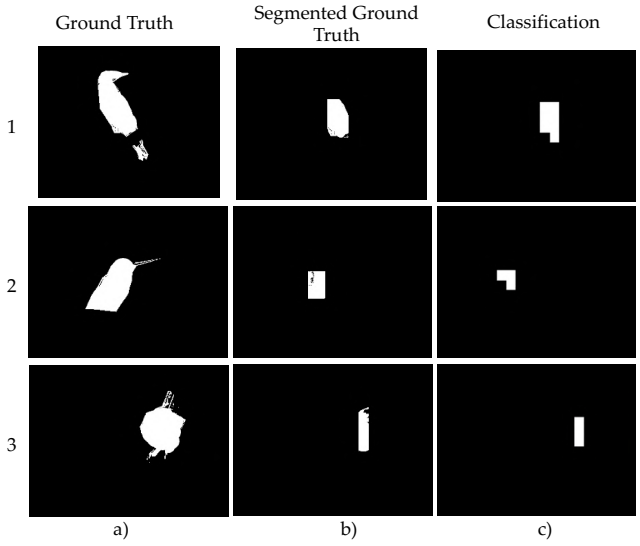

Figura 2.15: *Resultados metodológicos comparativos para tres imágenes medias con oclusión. (a) imagen segmentada original, (b) imagen segmentada mediante la segmentación propuesta, (c) clasificación por ventanas de* 15×15 *píxeles. Nota: las imágenes se amplificaron para una mejor ilustración.*

B.5. CONCLUSIONES

En este trabajo presentamos una metodología de detección y clasificación de aves denominada PSEV-BF (Pre-Segmentación y Variables Mejoradas para Características de Aves) que utiliza la Pre-segmentación y un algoritmo Simulated Annealing con Análisis de Componentes Principales denominado SA-PCA propuesto para mejorar las variables. PSEV- BF incorpora una nueva metodología en comparación con los métodos modernos. Además, puede aplicarse a imágenes con oclusiones y entornos no controlados.

La metodología de PSEV- BF consta de las fases de Preprocesamiento, Presegmentación, Segmentación, Extracción de Características, Selección de Variables relevantes y Clasificación. El preprocesamiento incluye ecualización de histogramas y filtrado gaussiano para la mejora y suavizado de la imagen. Para la presegmentación se utilizó una técnica de detección de CNN, YOLOV3, que proporciona un vector de coordenadas. Las coordenadas delimitan una región que tiene una alta probabilidad de pertenecer a un ave.

La segmentación refina las coordenadas obtenidas de la presegmentación redefiniendo la región dada. La región interior de las coordenadas se reduce en un 50 % y se cataloga como píxeles de primer plano. La región exterior de las coordenadas se incrementa en un 20 % y se cataloga como píxeles de fondo. Se utilizó una técnica de superpíxeles en la extracción de características para obtener un vector de 43 características con color y textura. La técnica de superpixel cubre un área de 15x15 píxeles.

Comparamos nuestra metodología con la tradicional. La metodología se probó con imágenes de categorías de aves de la base de datos COCO. Las imágenes se clasificaron según el tamaño del objeto deseado: Grande y Mediano. Se utilizaron un total de 193 imágenes para el entrenamiento y la validación del clasificador y 70 imágenes para las pruebas. Las imágenes de prueba se dividieron en grupos Grandes y Medianas, lo que corresponde a 35 imágenes por grupo. Se utilizaron un total de 16.988 vectores de características como muestras para el entrenamiento y la validación del clasificador Random Forest.

PSEV-BF se comparó con las metodologías M1, M2 y M3. Estas metodologías difieren en su configuración de la metodología propuesta. Para objetos grandes, PSEV-BF y M2 muestran valores con una precisión aproximada del 54 % con la métrica APIoU, mientras que M2 no dispone de la fase superpixel. En primer lugar, M1 y M2 utilizan al menos dos de los métodos propuestos en la metodología. Mientras que M3 no utiliza las fases propuestas, lo que da como resultado un 41 % de precisión de la métrica APIoU, que es el valor más bajo entre las metodologías comparadas. En segundo lugar, M2 no utiliza el método superpixel, lo que conduce a un valor de precisión muy similar en comparación con PSEV-BF, mientras que M1 tiene una diferencia del 2 % en comparación con M2. Podemos decir que el uso de los procesos propuestos para objetos grandes mejora la precisión de la metodología.

Para objetos clasificados como de tamaño Medio, la metodología de PSEV-BF muestra valores con una precisión aproximada del 36 % de la métrica APIoU. En primer lugar, M1 muestra una precisión del 32 %, que es el valor más bajo entre las metodologías comparadas. Esto significa que los efectos son tan grandes cuando no se utilizan la Segmentación y las Variables Mejoradas. PSEV-BF y M1 difieren en un 4 %, y la diferencia se debe al uso de un método superpixel. Encontramos que en PSEV-BF en objetos de tamaño medio se mejora la precisión de la predicción.

Este trabajo presenta una metodología con presegmentación, método de selección de variables y extracción de características empleando superpíxeles. Una vez puesta a punto la metodología puede ser utilizada para resolver problemas de identificación de objetos, por ejemplo, clasificar por tipo de pájaro u otros objetos más rápidamente que los métodos tradicionales.

Como trabajo futuro, proponemos utilizar técnicas similares para la segmentación supervisada de imágenes. PSEV-BF no se diseñó para reconocer especies de aves. Tenemos previsto incorporar otras estrategias de presegmentación, variables de características mejoradas y clasificación para reconocer distintas especies de aves.