

ESCUELA POLITÉCNICA NACIONAL

**FACULTAD DE DE INGENIERÍA ELÉCTRICA Y
ELECTRÓNICA**

APLICACIONES DE APRENDIZAJE AUTOMÁTICO

**CLASIFICACIÓN DE IMÁGENES DE PARÁSITOS DE REPTILES
MEDIANTE TÉCNICAS DE VISIÓN POR COMPUTADOR Y
APRENDIZAJE AUTOMÁTICO.**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
TELECOMUNICACIONES**

GUILLERMO ALEXANDER YANASCUAL GARCÍA

guillermo.yanascual@epn.edu.ec

DIRECTOR(A): Ing. VIVIANA CRISTINA PARRAGA VILLAMAR MsC.

viviana.parragav@epn.edu.ec

DMQ, abril 2023

CERTIFICACIONES

Yo, Guillermo Alexander Yanascual García, declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.



GUILLERMO ALEXANDER YANASCUAL GARCÍA

Certifico que el presente trabajo de integración curricular fue desarrollado por Guillermo Alexander Yanascual García, bajo mi supervisión.



**Ing. VIVIANA CRISTINA PARRAGA VILLAMAR MSc.
DIRECTOR(A) DE PROYECTO**

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

GUILLERMO ALEXANDER YANASCUAL GARCÍA
ING. VIVIANA CRISTINA PÁRRAGA VILLAMAR MSc.

DEDICATORIA

Soy de aquí y soy de allá,
Desde otra perspectiva, dirán
que no pertenezco a ningún lugar.

Y en esta realidad
como un grano de arena en medio de la eternidad
persigo mi leyenda personal

Si se me arrebatase mi nombre actual,
mi historia no cambiaría al final.
Al momento de mis alas desplegar
recordare que soy de aquí, y volaré sin dudar.

AGRADECIMIENTO

A mi padre celestial, que me permite enfrentar la realidad con bondad. A mis progenitores que me dieron los valores que forman mi identidad. A mis hermanas, que me recuerdan el futuro que vendrá y son imagen del ayer que quedará. A mis docentes, desde la primaria hasta la actualidad, pues ellos me forjaron y depositaron su confianza. A mis amigos de la vida, que me han acompañado en mi camino y han aportado a mi personalidad. A mi persona, por cada día incierto que deseo contemplar.

ÍNDICE DE CONTENIDO

CERTIFICACIONES	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO	IV
ÍNDICE DE CONTENIDO	VI
RESUMEN	VII
ABSTRACT	VIII
1 INTRODUCCIÓN	1
1.1 OBJETIVO GENERAL	2
1.2 OBJETIVOS ESPECÍFICOS	2
1.3 ALCANCE	3
1.4 MARCO TEÓRICO	4
1.4.1 AGENTES PARASITARIOS EN REPTILES BAJO CAUTIVERIO	4
1.4.2 MACHINE LEARNING	6
1.4.3 VISIÓN COMPUTACIONAL	10
1.4.4 ENTRENAMIENTO DE MODELO MEDIANTE APRENDIZAJE SUPERVISADO	16
1.4.5 MÉTRICAS PARA LA EVALUACIÓN DEL RENDIMIENTO DE UN CLASIFICADOR	19
1.4.6 ENTORNO DE SIMULACIÓN MATLAB	21
1.4.7 TRABAJOS RELACIONADOS	23
2 METODOLOGÍA	25
2.1 CONJUNTO DE DATOS	26
2.2 PREPROCESAMIENTO DE LAS IMÁGENES	27
2.2.1 PROCESO	28
2.3 EXTRACCIÓN DE CARACTERÍSTICAS	29
2.4 ENTRENAMIENTO Y VALIDACIÓN DE LOS DATOS	31
2.4.1 SELECCIÓN DEL CLASIFICADOR Y APLICACIÓN DE K-FOLD	31
2.4.2 MÉTRICAS DE EVALUACIÓN	32
3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES	34
3.1 RESULTADOS OBTENIDOS EN LA SEGMENTACIÓN Y APLICACIÓN DEL VOCABULARIO DE BoVW	34

3.2	RESULTADOS EN EL ENTRENAMIENTO DEL MODELO	36
3.3	CONCLUSIONES	41
3.4	RECOMENDACIONES	42
4	REFERENCIAS BIBLIOGRÁFICAS	43
4.1	Bibliografía	43
5	ANEXOS	I

RESUMEN

Ecuador, un país rico en flora y fauna, está comprometido con la protección de la vida en la región y la promoción del desarrollo biológico sostenible. El grupo de los reptiles, con una gran variedad de especies, son contempladas y analizadas en ambientes de cautiverio. De las diversas tareas en centros de laboratorio que se enfocan en la vida silvestre, como los reptiles, los expertos en el campo identifican los parásitos a partir de la recolección de muestras en base a la experiencia. De esta manera no se aplica ninguna automatización que agilice este proceso mecanizado. En respuesta a este problema, el presente trabajo propone un método de visión computacional para la clasificación de agentes parasitarios presentes en las muestras de seis reptiles en cautiverio para el personal de laboratorio, siendo de gran utilidad en el proceso de clasificación de parásitos en muestras, permitiéndoles estimar efectivamente las poblaciones de parásitos.

En el primer capítulo se presentaron los conceptos fundamentales respecto a las técnicas de visión computacional tradicional para el procesamiento, extracción de características y la clasificación de imágenes. En el segundo capítulo se abarca la metodología para clasificar los parásitos de muestras de heces de reptiles en cautiverio en el software Matlab mediante el extractor de Bolsa de palabras visuales (Bag of Visual Words, BoVW) y la técnica de clasificación de Máquinas de Vectores de Soporte (Support Vector Machine, SVM).

En el tercer capítulo se presentaron los resultados obtenidos para el clasificador propuesto y las conclusiones a los que estos llevaron. Se alcanzó un Área Bajo la Curva ROC (Area Under the ROC Curve, AUC) de 96 % y una exactitud de 84.4 %, lo que demuestra la factibilidad de usar visión computacional para este trabajo.

KEYWORDS: Modelo de clasificación, BoVW, predicción, agentes parasitarios.

ABSTRACT

Ecuador, a country rich in flora and fauna, is committed to protecting life in the region and promoting sustainable biological development. The group of reptiles, with a great variety of species, are contemplated and analyzed in captive environments. Of the various tasks in laboratory centers that focus on wildlife, such as reptiles, experts in the field identify parasites from sample collection based on experience. In this way, no automation is applied to speed up this machining process. In response to this problem, the present work proposes a computational vision method for the classification of parasitic agents present in the samples of six reptiles in captivity for laboratory personnel, being very useful in the process of classification of parasites in samples, allowing them to effectively estimate parasite populations.

The first chapter presented the fundamental concepts regarding traditional computer vision techniques for processing, feature extraction and image classification. The second chapter explains the methodology to classify parasites from feces samples of reptiles in captivity in Matlab software using Bag of Visual Words (BoVW) and the classification technique of Support Vector Machines (SVM).

In the third chapter presented the results obtained for the proposed classifier and the conclusions to which they led. An Area Under the ROC Curve (AUC) of 96 % and an accuracy of 84.4 % were achieved, demonstrating the feasibility of using computer vision for this work.

KEYWORDS: Classification model, BoVW, prediction, parasitic agents.

1 INTRODUCCIÓN

Ecuador forma parte de la región Neotropical, caracterizada por una diversidad de flora y fauna en el ecosistema del planeta. El país sudamericano se caracteriza por albergar la mayor cantidad de especímenes de reptiles con relación a unidad de área. A partir de la base de datos The Reptile Database [1], se describe una lista de reptiles de Ecuador de 498 grupos registrados. Esta distribución corresponde a las especies continentales y de las islas Galápagos [1]. La gran mayoría de la diversidad actual de reptiles en el país, que corresponde aproximadamente el 4.3% de la diversidad mundial, ha sido descubierta en años recientes, y probablemente el número de especies de reptiles en el Ecuador aumente considerablemente durante los próximos años [1].

Al ser un país mega diverso, incursiona en la responsabilidad de garantizar la preservación de los animales, defensa de su bienestar y protección que ha sido abordado por instituciones enfocadas al cuidado y organización de la fauna silvestre. Entre los hechos recopilados del trabajo de las instituciones en [2] se encuentran las actividades en Biogeografía y Zoología frente a la amenaza global de la extinción de reptiles, la aceptación del servicio ecológico por parte de los animales silvestres para controles de sanidad, y preservación de alimento o políticas que aúnan el establecimiento de centros de manejo de fauna silvestre para favorecer la supervivencia. Sin embargo, existen diversos planteamientos que demandan la intervención de los organismos legales, para enfrentar ámbitos como las afecciones del sistema gastrointestinal de diferentes animales silvestres [3].

Respecto al último punto, las infestaciones de parásitos en los grupos de animales silvestres en el Ecuador implican un grave problema para el ciclo de vida, en especial si están en situación de cautiverio. En este contexto, las aves y los reptiles de diferentes especies son más propensos a presentar síntomas visibles [2] como lesiones o inmunosupresión luego de infectarse con agentes parasitarios. A partir de un estudio de investigación [2] en dos centros de conservación de fauna silvestre del país, al realizar la identificación de parásitos intestinales en muestras recolectadas de reptiles, aves y mamíferos en cautiverio, se obtuvo una mayor carga parasitaria en las muestras de los reptiles. Es de resaltar que el parásito

se denomina como uno de los organismos vivos infecciosos más comunes y de fácil propagación en los centros de manejo y cuidado de fauna [3]. Los efectos de su propagación en el grupo más vulnerable consisten en lesiones o limitación de la inmunidad, desencadenando el aumento de enfermedades secundarias hasta el punto de elevar la tasa de mortalidad [2]. Así, la demanda de garantizar una correcta vigilancia de parasitismo en especies en cautiverio consiste en prevenir posibles fuentes de transmisión de enfermedades zoonóticas (de animal a humano).

La aplicación de métodos de clasificación manual de identificación parásitos (métodos coprológicos) [3], demanda gran carga de trabajo y tiempo frente a la posibilidad de implementar herramientas de clasificación automática de agentes parasitarios. El método manual se refiere a la utilización de un ser humano como el instrumento para interpretar la información que posee las muestras. Al lograr un proceso autómatas, se mejora el post proceso de diagnósticos y protocolos de desparasitación adecuados con el fin de prevenir enfermedades en los reptiles.

En el presente trabajo se formula, utilizando el software bajo licencia de Matlab, un enfoque para clasificar los parásitos en áreas de parasitología durante los procesos coprológicos de una forma confiable, segura y práctica, a través de métodos de visión computacional y algoritmos de clasificación estadísticos aplicados a las imágenes digitales. El proceso consiste en la identificación de los parásitos gastrointestinales en muestras de heces de reptiles a partir de imágenes obtenidas de un microscopio. Se procesa y analiza las muestras mediante técnicas de visión computacional y aprendizaje automático supervisado para clasificar especies de parásitos o la ausencia del organismo.

1.1 OBJETIVO GENERAL

Clasificar mediante técnicas de visión por computador y aprendizaje automático, parásitos de reptiles en cautiverio a partir de un conjunto de datos de imágenes de heces capturadas con microscopio.

1.2 OBJETIVOS ESPECÍFICOS

- Estudiar los aspectos más relevantes de la parasitología en reptiles.
- Comprender el funcionamiento de la técnica de Bag of Visual Words BoVW o bolsa

de palabras visuales.

- ❑ Comprender los conceptos fundamentales de aprendizaje supervisado.
- ❑ Aplicar la técnica de bolsa de palabras visuales a partir de una muestra de imágenes de parásitos obtenidas por microscopio para identificar el tipo de parásito.
- ❑ Determinar el rendimiento del clasificador mediante métricas de clasificación.

1.3 ALCANCE

El presente trabajo implementa en el entorno de simulación Matlab un programa de clasificación de parásitos. El software mencionado trabaja bajo licenciamiento, utilizando la herramienta de trabajo Computer Vision Toolbox para el manejo de la técnica de Bolsa de palabras visuales BoVW y la herramienta de Aprendizaje de Clasificación (Classification Learner) para la etapa de entrenamiento y evaluación.

En primer lugar, se realiza el proceso de preparación de la base de datos provista por la especialista en Veterinaria Alejandra Núñez, que realizó un proceso de extracción de imágenes de parásitos a partir del microscopio en el trabajo de investigación “Identificación de parásitos con diferentes métodos coprológicos en muestras de reptiles en el vivarium de Quito” [3]. Estas imágenes fueron clasificadas en seis categorías. Además, se incorpora un bloque de imágenes con la ausencia de algún agente parasitario.

A continuación, se presenta un bloque para la extracción de características de las imágenes de agentes parasitarios. Para este fin se emplea el modelo BoVW (Bolsa de Palabras Visuales) [4].

Posteriormente, se utiliza el vocabulario obtenido de BoVW para entrenar un clasificador que permita obtener las etiquetas de determinado parásito o ausencia de este.

Para finalizar, mediante la aplicación de K-fold, se evalúa el parámetro de precisión del sistema propuesto.

En síntesis, el sistema propuesto (ver Figura 1.1) maneja un bloque de extracción de características mediante la Bolsa de Palabras Visuales (BoVW) en conjunto con el clasificador Máquina de vectores de Soporte (SVM). El entrenamiento emplea como entradas las características obtenidas de BoVW y como salidas las etiquetas de los parásitos (i.e. BLAS, OPHY, OXI, RHYTI, STRONG, TAE, NP).

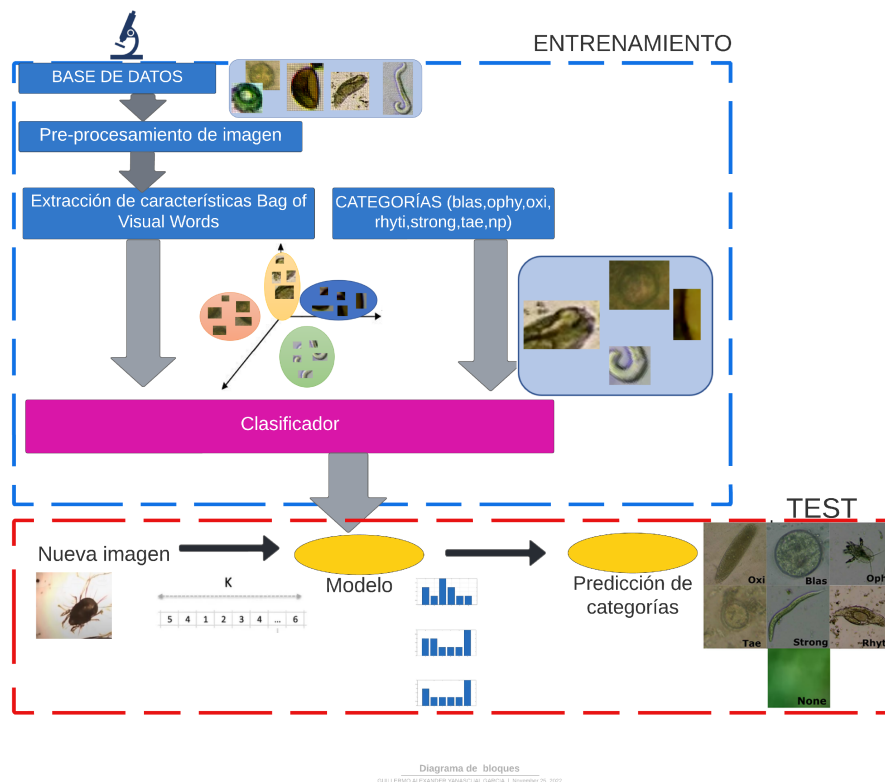


Figura 1.1: Diagrama de bloques de clasificación para parásitos de heces de reptiles utilizando visión computacional.

1.4 MARCO TEÓRICO

En este apartado se distribuye la información relevante en cuatro bloques, incluyendo al final los trabajos relacionados. Se revisan los aspectos más importantes de la parasitología. Luego continúa con el concepto del Aprendizaje Supervisado, las técnicas de Visión Computacional en el aprendizaje automático, clasificación con la máquina de Vectores de Soporte (SVM) y las métricas para evaluar el rendimiento del clasificador.

1.4.1 AGENTES PARASITARIOS EN REPTILES BAJO CAUTIVERIO

La parasitología en reptiles ha sido uno de los campos con menor investigación científica [2]. Los parásitos son uno de los patógenos comunes y transmisibles en las zonas de cuidado y manejo de vida silvestre, causando daño e inmunodeficiencia en los reptiles, aumentando la mortalidad y predisponiendo a enfermedades secundarias [3]. Existe una serie de enfermedades y efectos generados por la presencia de estos huéspedes en los reptiles. Desde el desarrollo de hepatitis en serpientes, alteración en el comportamiento y la fisiología de

las lagartijas, indicios de enteritis crónica con edema y sangrado de la mucosa intestinal, o debilidad por pérdida de sangre acompañado neumonía y septicemia por presencia de ácaros [3].

La necesidad de implementar medidas preventivas y de control frente a la propagación de estos agentes en las especies cautivas es indispensable. Esto se debe a que los parásitos pueden ser responsables de la transmisión de enfermedades de animales a humanos (zoonosis) [2]. Los animales al estar fuera de su ecosistema natural evidencian alteraciones en el estado inmunitario, sumado al estrés y condiciones del lugar en cautiverio, proliferan una mayor susceptibilidad a diversas enfermedades infecciosas. Los reptiles son huéspedes de una gran diversidad de parásitos internos, en su mayoría de tipo intestinal [3].

Para el seguimiento sanitario de los animales silvestres y cautivos, se implementa un análisis de muestras fecales [2]. Se utiliza un método manual de clasificación de parásitos que incluía el análisis de imágenes microscópicas de heces realizadas por expertos humanos. Sin embargo, este trabajo requiere de un esfuerzo y tiempo considerable debido a que el espacio donde se preservan los reptiles puede presentar una carga parasitaria en gran proporción.

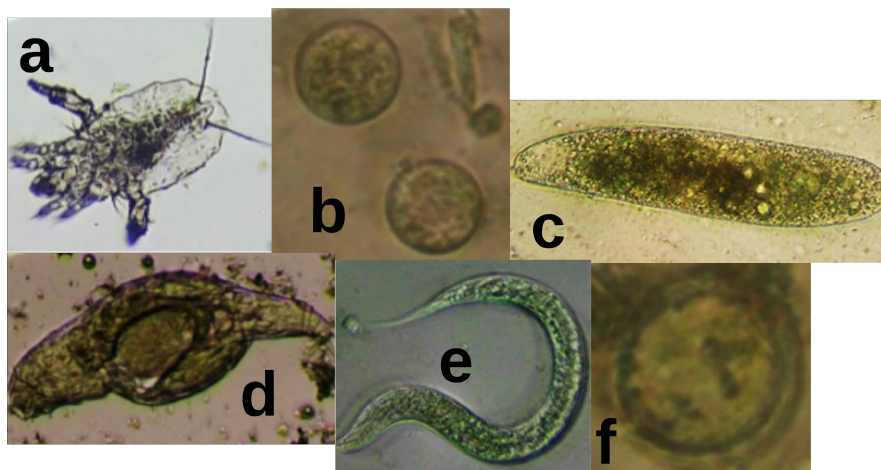


Figura 1.2: Ejemplos de cada parásito. a) *Ophionyssus natrici* (ácaro), b) *Blastocystis* sp (protozoario), c) huevos de Oxiurido (nematodo), d) *Rhytidoides similis* (trematodo), e) larvas de *Strongyloides* (nematodo) y f) *Taenia* spp (cestodo). Adaptado de [3].

Se presenta en la Figura 1.2 algunos de los grupos de parásitos intestinales con mayor presencia según diagnósticos realizados en reptiles. Los nombres de parásitos [3] *Ophionyssus natrici* (ácaro), *Blastocystis* sp (protozoo), huevos de Oxiurido (nematodo), *Rhytidoides similis* (trematodo), larvas de *Strongyloides* (nematodo) y *Taenia* spp (cestodo), corresponde

a las etiquetas para las categorías de estudio del trabajo presente.

1.4.1.1 Toma y análisis de muestras

Se puede realizar mediante el método directo e indirecto. El primer método es el hisopado y la colocación de la muestra en un frasco o bolsa, mientras que el método indirecto recolecta heces frescas del mismo nido, recinto o terrario de cada especie. En ambos casos, las muestras se colocan en un recipiente hermético y a temperatura de refrigeración hasta el análisis de laboratorio [3].

Las muestras de heces pueden ser procesadas mediante tres exámenes de heces. El manejo de pruebas de heces es empleado en mayor proporción para diagnosticar infecciones parasitarias. El primer examen consiste en el "método directo"[2] que identifica muestras de heces frescas en busca de formas parasitarias bajo el microscopio. Luego, la "técnica de flotación"[3], consiste en separar las formas parásitas, que se acumulan en la superficie debido a su menor densidad que la solución en la que están sumergidas. La tercera llamada "sedimentación"[3], utiliza un medio de menor densidad a la del parásito y la gravedad una sedimentación natural.

De esta manera se puede identificar los parásitos variando la calidad de los resultados de acuerdo con la técnica a utilizar. Existen modelos de microscopio táctil digital para obtener imágenes y vídeos de parásitos. No obstante, aún son utilizados los microscopios ópticos. Finalmente, los resultados visuales son registrados por un veterinario especialista empleando las etiquetas de acuerdo con el parásito presente en ellos [2].

1.4.2 MACHINE LEARNING

A partir del gran volumen de datos generados en el mundo como imágenes digitales, documentos, procesamiento de videos en tiempo real, el área de aprendizaje automático permite evolucionar de los sistemas tradicionales de análisis de datos basados en patrones y reglas manuales, a manejo de volúmenes de información con sistemas automatizados desarrollando sus propias reglas de programación con la capacidad de adaptación frente a las nuevas exigencias del mundo global.

El aprendizaje automático (Machine Learning, ML), que se basa en la inteligencia artificial, se enfoca en brindar un aprendizaje mediante una lógica deductiva sin reglas previas de programación a las máquinas. De esta forma se emigra de un sistema de programación

tradicional que prescindía de un algoritmo de instrucciones de forma manual a un modelo de aprendizaje autodidáctico con toma de decisiones a partir de ejemplos, mejorando el aprendizaje del modelo a partir del conocimiento adquirido sobre la información que está procesando [5].

Actualmente, el área de aprendizaje automático se encuentra presente a través de aplicaciones como reconocimiento de imágenes, filtros en redes sociales, algoritmo de spam en bandejas de correo, así como sistemas de traducción de texto y voz. Con el desarrollo de las redes 5G, se perfila el manejo del aprendizaje automático en conducción autónoma, redes ópticas dinámicas, data centers autónomos, etc.

Se menciona en [5], al aprendizaje automático como el proceso de entrenar un bloque de software, denominado modelo, utilizando datos para realizar predicciones útiles. El aprendizaje del modelo se representa mediante una relación matemática entre los elementos de datos que utiliza un sistema de ML para hacer predicciones.

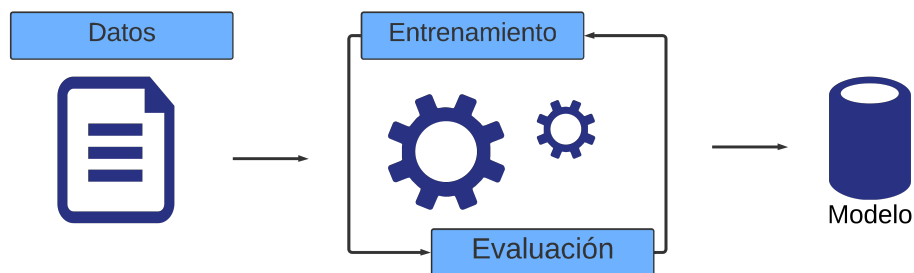


Figura 1.3: Etapas para el proceso de aprendizaje automático. Adaptado de [6].

1.4.2.1 Elementos para el desarrollo del aprendizaje automático

El aprendizaje automático involucra secuencialmente los elementos (ver Figura 1.3) de datos, modelo, entrenamiento (train) y evaluación (test) [5].

Datos

Se caracteriza por su tamaño y diversidad. El tamaño hace mención del número de ejemplos. En cambio, la diversidad indica el rango que cubren esos ejemplos. Se considera un conjunto de datos adecuado si es de gran tamaño y muy diversos.

Modelo

Es una representación matemática de un sistema que define una relación matemática desde un patrón de atributo de propiedad de entrada particular hasta un valor o resultado de etiqueta de salida particular. El fin del modelo es ser entrenado para aprender patrones y

relaciones, y luego se utilizan para hacer predicciones sobre nuevos datos.

Los modelos pueden variar en su complejidad y en la forma en que representan los datos y las relaciones entre ellos. Algunos modelos, como los árboles de decisión y los modelos lineales, son relativamente simples y fáciles de comprender, mientras que otros, como las redes neuronales, pueden ser mucho más complejos y difíciles de interpretar.

Entrenamiento

Para que un modelo pueda hacer predicciones, debe previamente ser entrenado. Para entrenar un modelo, se entrega un conjunto de datos con ejemplos etiquetados (en el caso de aprendizaje no supervisado no se conoce las etiquetas y se identifican patrones que guardan los datos para reagruparlos). De esta forma el modelo se centra en determinar el mejor aprendizaje para predecir las etiquetas o grupos a partir de las características. El modelo encuentra los parámetros que mejor se ajusten a los datos de entrenamiento, minimizando una función de pérdida que mide la discrepancia entre las predicciones del modelo y los valores reales de las etiquetas de los datos de entrenamiento. Una vez que se han encontrado los parámetros óptimos, el modelo se puede utilizar para hacer predicciones sobre nuevos datos.

Evaluación

Se evalúa un modelo entrenado determinando el nivel de aprendizaje. Si el modelo entrenado utiliza etiquetas, para la etapa de evaluación se ingresan los atributos del conjunto de datos y posteriormente se compara las predicciones del modelo con los valores esperados. Si no se manejaba un conjunto de etiquetas y se agrupaban los datos de entrada obteniendo determinadas categorías, se revisa si los datos de evaluación guardan relación en la redistribución de grupos. En este bloque destaca la importancia de discriminar entre múltiples opciones de modelos. De acuerdo con el grado de predicción del modelo, puede darse la realización de nuevas etapas de entrenamientos y evaluaciones hasta consolidar el modelo en una aplicación real. De un conjunto de datos, se considera un desglose estándar de 70-80% de datos en el bloque de entrenamiento mientras que el 20-30% restante se denomina conjunto de prueba [7]. Si se asigna un porcentaje diferente para el segmento de entrenamiento puede repercutir en el fenómeno de underfitting & overfitting.

1.4.2.2 Tipos de ML

Los tipos de ML se catalogan en tres grupos distintos respecto a su forma de aprendizaje para realizar predicciones [5, 6]:

- ❑ Aprendizaje supervisado.
- ❑ Aprendizaje sin supervisión.
- ❑ Aprendizaje reforzado.

Aprendizaje Supervisado

Modelo que realiza predicciones a partir de la relación de un conjunto de datos y resultados correctos denominados etiquetas (targets). Desarrolla conexiones entre los elementos de los datos que producen las respuestas correctas. Se denomina "supervisado" debido a que se proporciona al sistema de ML, datos con los resultados correctos conocidos.

Se presentan dos casos de uso para el aprendizaje supervisado que corresponde a la regresión y la clasificación.

- ❑ **Regresión:** Centra su proceso de aprendizaje en predecir un valor numérico [5]. Para ejemplo se menciona de un modelo meteorológico que realiza predicciones del nivel de lluvia, en pulgadas o milímetros.
- ❑ **Clasificación:** Enfoca la predicción en la probabilidad de que el resultado se representa en determinada categoría. Se diferencia del modelo de regresión (resultado numérico), respecto a determinar si algo pertenece o no a una categoría específica [5]. Como ejemplo se menciona a los modelos de clasificación de correo de un usuario, determinando si un correo electrónico es spam. Los modelos de clasificación se dividen en dos grupos: clasificación binaria y clasificación multiclase.

Aprendizaje no supervisado

Destacan por realizar predicciones a partir de datos que no contienen respuestas. En este caso el modelo centra sus esfuerzos en identificar patrones significativos entre los datos para luego agruparlos [5]. Un ejemplo de uso común de este aprendizaje consiste en la técnica "clustering". La tarea del modelo ahonda en encontrar puntos de datos que delimiten agrupaciones naturales.

Aprendizaje reforzado

Maneja predicciones a partir de un método de recompensas-penalizaciones como efecto de las acciones realizadas dentro de un entorno. En este caso se habla de una política que promueve la mejor estrategia para obtener la mayor cantidad de recompensas [6].

1.4.3 VISIÓN COMPUTACIONAL

La visión por computadora es una de las áreas más populares de la aplicación de tecnología de aprendizaje automático. Se enfoca en el desarrollo de sistemas que puedan procesar y comprender imágenes y vídeos, permitiendo a las computadoras observar de forma similar las características, colores, contornos, entre otros aspectos, a la percepción visual del humano. La Visión Computacional trata mediante un conjunto de procesos de emular la capacidad de visión humana en los ordenadores, permitiendo la interpretación de las imágenes adquiridas, reconocimiento de los diversos objetos en el ambiente y localización en el espacio [8].

Al profundizar sobre la visión computacional, es menester revisar tres aspectos importantes:

- ❑ Se aborda a la visión como un proceso computacional.
- ❑ La descripción como color, textura, forma, fondo depende del observador.
- ❑ Es necesario eliminar la información no útil (reducción de información o ruido de fondo).

El objetivo de la visión por computadora consiste en seleccionar características de las imágenes para la descripción e interpretación por computadora. En la Figura 1.4 se visualiza el enfoque de enseñanza a las computadoras para interpretar y comprender las imágenes del mundo real. Se digitaliza la imagen para una representación numérica de los píxeles ordenados en matrices y posteriormente aplicar diversas técnicas de procesamiento.

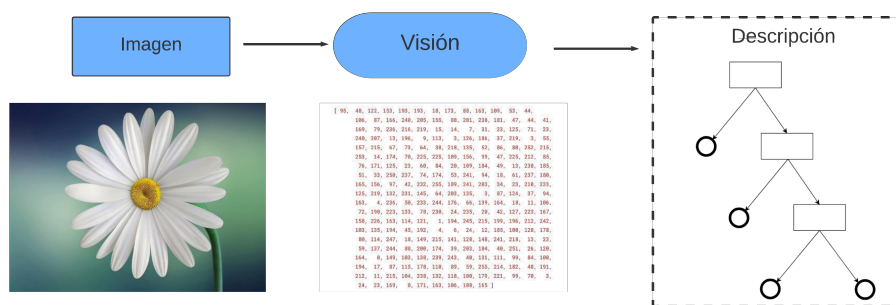


Figura 1.4: Esquema general de visión por computadora. La imagen de entrada es procesada extrayendo determinadas características. Adaptado de [8].

De esta forma se presenta en la Figura 1.5 un modelo de aprendizaje automático que aborda el bloque de visión computacional. Inicia con el preprocesamiento de la imagen, para

posterior realizar la extracción de características y segmentarlas bajo ciertos parámetros comunes, mediante un agrupamiento con la técnica k-means, representando a un vocabulario de vectores para cada imagen que corresponde a la técnica de BoVW. Luego se puede utilizar cualquier algoritmo de aprendizaje automático para clasificar o etiquetar las imágenes en función de sus vectores de palabras.

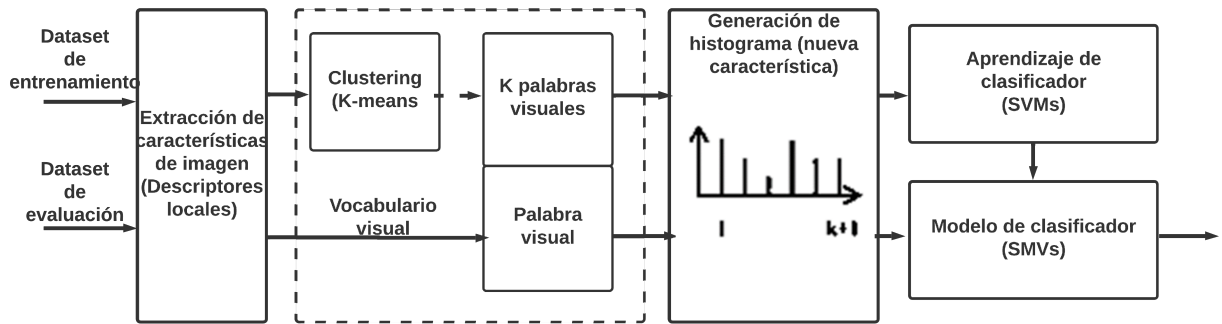


Figura 1.5: Etapas del modelo de clasificador de imágenes mediante BoVW. Adaptado de [8].

1.4.3.1 Preprocesamiento de imágenes

Un bloque generalmente aplicado dentro de la visión computacional. El propósito del preprocesamiento de imágenes es mejorar la calidad de la imagen para su posterior uso o interpretación. El enfoque principal de este bloque consiste en presentar una imagen de entrada seleccionando o suprimiendo ciertas características. Se entrega a la salida una imagen que permite realizar tareas como la detección de bordes, la segmentación de objetos y la extracción de características [8].

Existen diversas técnicas y mecanismos que permiten centrar la imagen al objeto de estudio y limitan características o entornos de distorsión. En la Figura 1.6 se visualiza un ejemplo de aplicación de algunas técnicas aplicadas en [9]. A nivel básico, se puede obtener una ventana de la imagen donde se limite al objeto de estudio. En este caso se determinan las coordenadas del elemento para obtener un nuevo trazo. A mayor nivel, se utilizan operaciones morfológicas que permiten cambiar ciertas regiones de la imagen, ofreciendo ventajas como opacar áreas donde resaltan propiedades o eliminar objetos no deseados o propiedades que no aportan a las imágenes resultantes. Otras consideraciones abarcan mejoras del contraste de imagen, conversión de imagen de color a escala de grises, binarización de imágenes, el funcionamiento morfológico basado en la erosión, funcionamiento del gradiente morfológico, etc.

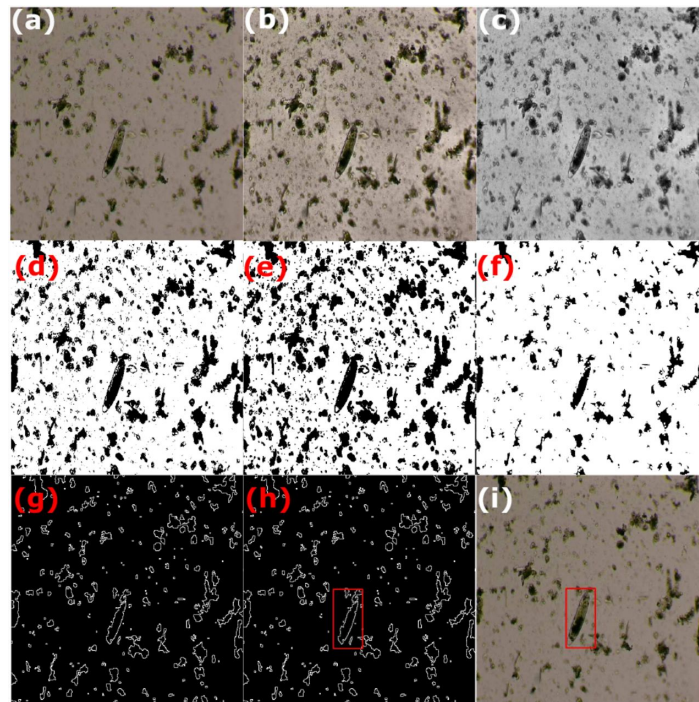


Figura 1.6: Ejemplo de aplicación de diversas técnicas procesamiento de imagen [9].

Contraste

La nitidez es el contraste entre diferentes colores y es aplicada a la imagen mediante una máscara de nitidez. La rápida conversión de negro a blanco corresponde a una gran nitidez mientras que la conversión gradual de negro a gris y de gris a blanco se cataloga de borrosa. Entonces si se realizan mejores transiciones a través de la nitidez, se obtiene mejor contraste a lo largo de los bordes donde se encuentran los diferentes colores.

La aplicación de la máscara de reducción de nitidez maneja el principio de sustraer una versión borrosa (no nítida) de la imagen.

Binarización a través del método de Otsu

Permite identificar el umbral de conversión adecuado al minimizar la variación intraclasses entre dos clases de píxeles asumidas. Se debe definir un valor umbral para segmentar gráficos tal que permita separar los objetos de estudio de una imagen del resto llamado ruido. De esta forma se puede diferenciar un objeto del fondo de la imagen [10].

El método de Otsu maneja la varianza para la dispersión de los niveles de gris. El valor de umbral se obtiene tal que la dispersión sea lo más pequeña posible en cierto segmento mientras que permite diferenciar con segmentos diferentes.

Redimensión y selección de área del parásito (eliminación del ruido)

Para el manejo de herramientas de procesamiento de imágenes, es de gran ventaja mantener una uniformidad dimensional. De esta manera se puede utilizar herramientas como obtención de una nueva ventana de la imagen que sea de interés como la supresión del ruido. Para la selección de la nueva ventana, se aplica de forma manual o mediante coordenadas tal que señale la posición del objeto de estudio.

1.4.3.2 Segmentación con K-means

Catalogado como una clasificación no supervisada, se busca agrupar a las imágenes de acuerdo con ciertos rasgos o patrones existentes. De esta forma, los miembros de cierto grupo son similares entre sí aproximando sus características a la par de una discriminación notoria respecto a otro grupo. [11].

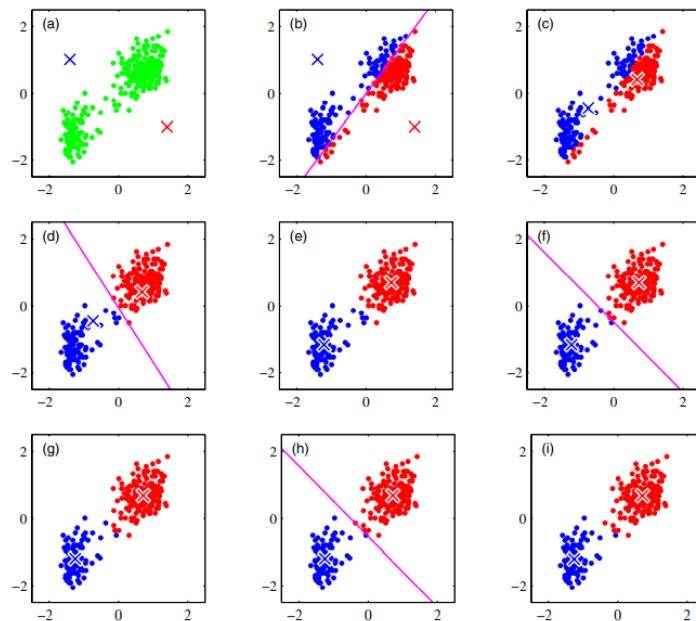


Figura 1.7: Ejemplo de la segmentación con k-means. [11].

En la Figura 1.7 se visualiza las etapas de funcionamiento de la técnica. En síntesis, se inicializan centroides al azar y se comienzan a formar grupos alrededor de los centroides. En cada uno de esos grupos, los elementos convergen al centroide, siendo la proximidad el rasgo esencial de esta técnica, recalculando los centroides y repitiendo este proceso.

En relación con la técnica de BoVW, el conjunto de datos corresponde a las características extraídas (digitalización) de las imágenes. El número de grupos, k , se selecciona de antemano y generalmente se realiza un análisis previo para determinar el valor óptimo de k . Después de aplicar k-means, cada grupo representa una palabra visual en el diccionario.

Cada característica de la imagen se asigna al grupo más cercano (o palabra visual más cercana) para su posterior uso en la representación de la imagen.

1.4.3.3 Método de Bolsa de Palabras Visuales

El modelo BoVW construye un vocabulario visual de palabras al combinar descriptores de imágenes locales extraídos y luego representando cada imagen como un histograma de palabras visuales.

Se maneja un vector disperso de ocurrencias de los descriptores de características de imágenes locales. Esta representación también se puede describir como un histograma de palabras visuales. El modelo BoVW se puede dividir en dos etapas principales [4]. Consiste en la detección y representación de características, seguido de un proceso de agrupamiento y formación del vocabulario [4].

Hay algunas ventajas y desventajas al utilizar la técnica de bolsa de palabras visuales (BoVW) en lugar de redes de aprendizaje profundo para resolver problemas de visión computacional. Algunas de las ventajas de BoVW incluyen:

- Es rápido y eficiente en términos de uso de recursos y ejecución.
- No requiere una gran base de datos de entrenamiento para generar resultados adecuados.
- Es robusto a los cambios de iluminación y a otras variaciones en las imágenes.

Sin embargo, también hay algunas desventajas a tener en cuenta al utilizar BoVW:

- Los resultados suelen ser menos precisos que los obtenidos con el aprendizaje profundo en tareas específicas.
- Depende de la selección manual de un diccionario de palabras y de otras opciones de parámetros.
- Presenta limitaciones para manejar imágenes complejas o que contienen muchos detalles.

En general, BoVW es una técnica útil y robusta que puede ser adecuada para muchas aplicaciones de visión computacional.

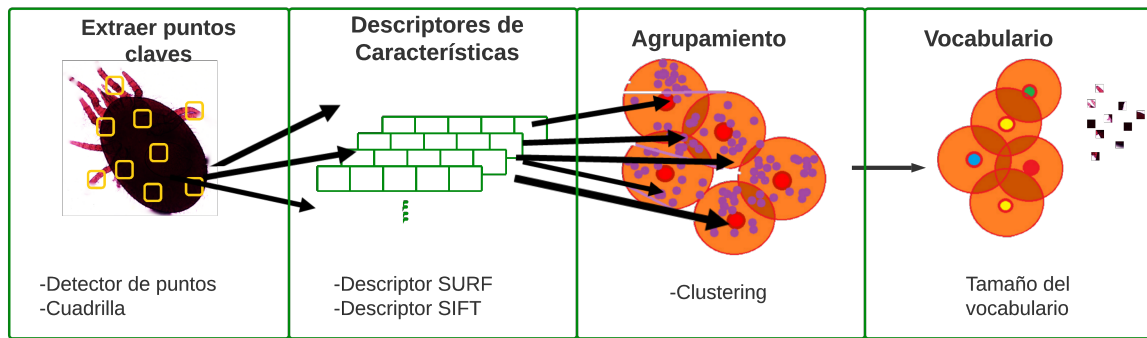


Figura 1.8: Representación del proceso de flujo de trabajo del algoritmo Bolsa de Palabras Visuales.

Se presenta a continuación una secuencia de pasos para la obtención del vocabulario de palabras visuales a partir de la base de datos (ver Figura 1.8).

Extracción de puntos clave

Consiste en la extracción de puntos clave o de interés. Estos puntos corresponden a lugares representativos de la imagen, como esquinas o colores [4]. La definición de los puntos se determina comúnmente mediante dos formas:

- ❑ Manejo de cuadrillas: Se manejan cuadrillas uniformes, donde las intersecciones de sus líneas determinan las ubicaciones de los puntos de interés [4].
- ❑ Detector de puntos: Se usan algoritmos de detección de puntos de características. Se presenta dos métodos principales denominados detección de Características Invariantes a Escala y Orientación (Scale Invariant Feature Transform, SIFT [12]) o detección de puntos de Características Robustas Aceleradas (Speeded Up Robust Features, SURF [13]).

Formación de descriptores de características

Al detectar un punto de interés (del paso anterior) se inspecciona sus alrededores y se extrae un descriptor de características que identifica los aspectos más importantes de un punto clave [4]. Se describe a continuación los descriptores mencionados:

- ❑ SIFT: Es de los primeros métodos para extracción de características. Corresponde a 128 dimensiones (características) donde las características extraídas son invariantes a traslaciones, rotaciones y escala. Esto le permite ser robusto. Consta de cuatro etapas principales: detectar extremos de espacio de escala, localizar puntos clave, asignar la descripción y orientación de puntos clave [12].

- ❑ SURF: Utiliza una técnica de detección de bordes para encontrar los puntos de características y luego una función de aproximación para encontrar los vectores de características para discriminar entre varios objetos. Maneja 64 dimensiones. Se caracteriza por ser invariante a la escala pero no a la rotación [13]. El proceso de extracción de características es similar a SIFT, no obstante, cambia el modo de localización de los keypoints debido a la función de aproximación.

De los resultados revisados en [13], se determinó que SIFT puede detectar mayor cantidad de características en comparación con SURF, sacrificando a cambio la velocidad. Por otro lado, SURF es rápido y tiene un buen rendimiento (aproximadamente 10 veces más rápido que SIFT). Esto permite que SURF sea más confiable que SIFT.

Agrupamiento de los descriptores

En el proceso de clustering se busca agrupar descriptores de características que guarden rasgos comunes. De esta forma se aplica el método de agrupamiento k-means para identificar el número de n clústers [4].

Construcción del vocabulario

Al formar los grupos (clústeres), se establecen centroides en cada uno de los conjuntos. A los centroides obtenidos, se los denomina palabras visuales (visual words) del vocabulario. El valor de " n " agrupamientos implica el tamaño del vocabulario [4].

Representación del histograma de características

Posteriormente, la frecuencia de cada palabra visual se registra en un histograma que representa el vector de características final de la imagen [4].

1.4.4 ENTRENAMIENTO DE MODELO MEDIANTE APRENDIZAJE SUPERVISADO

A continuación, se busca entrenar un determinado modelo para predecir la etiqueta de clase de la nueva imagen. Los histogramas de palabras visuales obtenidos se utilizan como los datos de entrada para el bloque de entrenamiento.

Luego de extraer las características de las imágenes mediante BoVW, se pueden aplicar diversos tipos de modelos de aprendizaje automático enfocados a tareas de clasificación y regresión. Algunos modelos de aprendizaje automático comunes que se pueden utilizar con características extraídas mediante BoVW incluyen:

- ❑ Máquinas de soporte vectorial (SVMs): son un tipo de modelo de clasificación que funciona buscando un hiperplano de separación óptimo entre dos clases en un espacio de características de alta dimensión.
- ❑ Redes neuronales: corresponde a modelos de aprendizaje automático basados en el funcionamiento del cerebro humano. Pueden ser utilizadas tanto para tareas de clasificación como de regresión.
- ❑ Árboles de decisión: son un tipo de modelo de clasificación que funciona dividiendo recursivamente el espacio de características en regiones más pequeñas basándose en diferentes criterios.
- ❑ Regresión logística: un modelo de clasificación que utiliza una función logística para estimar la probabilidad de que una muestra pertenezca a una clase.

En general, cualquier modelo de aprendizaje automático que utilice como entrada características extraídas mediante BoVW, enfoca el performance según la tarea específica a realizar y del conjunto de datos disponible.

1.4.4.1 Máquina de Vectores de Soporte (SVM)

El modelo de clasificación SVM se describe como un tipo de clasificador binario bajo un algoritmo supervisado. Consiste de un clasificador discriminativo que utiliza un hiperplano de separación entre los grupos, donde el hiperplano es una línea N-dimensional [14]. Es utilizado para bloques de entrenamiento etiquetados (aprendizaje supervisado) de clasificación binaria. Actualmente puede ser aplicado en temas de agrupamiento multiclase y regresión [14].

Una forma común de utilizar SVMs para la clasificación multiclase es la estrategia uno contra todos (One-vs-All). En esta estrategia, se entrenan tantos clasificadores binarios como clases exista en el conjunto de datos. Cada clasificador binario se entrena para distinguir entre una clase y el resto de las clases. Luego, cuando se quiere realizar una predicción para una nueva instancia, se utilizan todos los clasificadores binarios para hacer predicciones y se selecciona la clase con la mayor puntuación.

Otra forma de utilizar SVMs para la clasificación multiclase es mediante el uso de la estrategia uno contra uno (One-vs-One). En esta estrategia, se entrenan clasificadores binarios para distinguir entre todas las posibles parejas de clases. Por ejemplo, en un conjunto de

datos con tres clases, se entrenaron tres clasificadores binarios para distinguir entre las clases 1 y 2, entre las clases 1 y 3 y entre las clases 2 y 3. Luego, cuando se quiere realizar una predicción para una nueva instancia, se utilizan todos los clasificadores.

En el contexto de un clasificador SVM, el kernel es una función que se utiliza para transformar los datos de entrada en un espacio de características de mayor dimensionalidad. La idea es que, después de aplicar el kernel, los datos quedan representados en espacios fácilmente separables linealmente.

Hay diversos tipos de kernels a utilizar con SVM, cada uno con sus propias propiedades y aplicaciones específicas. Algunos ejemplos comunes son el kernel lineal, el kernel polinómico y el kernel radial.

1.4.4.2 Validación Cruzada K-fold

Consiste en dividir el conjunto de datos de entrenamiento en K subconjuntos o "pliegues". Luego, se entrena el modelo K veces, utilizando en cada iteración un subconjunto diferente como el bloque de prueba y los $K-1$ subconjuntos restantes como el bloque de entrenamiento. Al finalizar, los resultados obtenidos en cada iteración permiten obtener una medida del rendimiento del modelo.

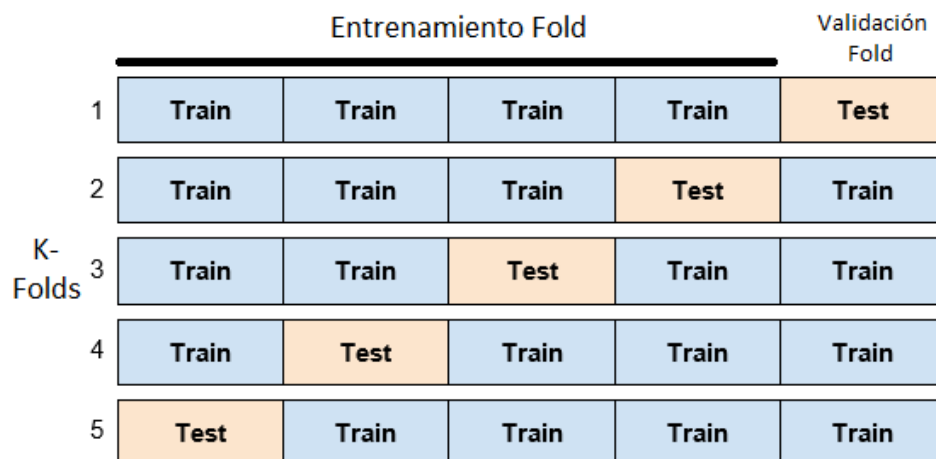


Figura 1.9: Partición de datos con k-fold cross-validation. Adaptado de [15].

En primera instancia se realiza una división única de entrenamiento/prueba, es decir, se separa la base de datos en un conjunto de entrenamiento y un conjunto de pruebas fijos. No obstante, se debe buscar una medida de rendimiento del modelo en general, por lo que la segmentación realizada refleja una medida de qué tan bien predice un conjunto de prueba específico. Entonces la técnica de validación cruzada permite dividir los datos en

los grupos de entrenamiento y de evaluación de diversas formas [15].

La técnica de *K*-fold es útil porque permite utilizar todos los datos disponibles para el entrenamiento y la evaluación del modelo, lo que puede resultar en un rendimiento mejorado (ver Figura 1.9). Además, al entrenar el modelo varias veces con subconjuntos diferentes de datos, se pueden obtener estimaciones más precisas del rendimiento del modelo en conjuntos de datos "desconocidos".

1.4.5 MÉTRICAS PARA LA EVALUACIÓN DEL RENDIMIENTO DE UN CLASIFICADOR

Entrenar un modelo consiste aprender (determinar) valores correctos frente a las ponderaciones y las ordenadas al origen de los ejemplos etiquetados. El fin de manejar métricas permite identificar los algoritmos y las representaciones de datos que funcionan por encima de una línea base de rendimiento y mejor que el promedio. De esta forma escoger determinada métrica permite evaluar la habilidad de las predicciones, en base a los requisitos del problema y el dominio. Probablemente la precisión (accuracy) es el más utilizado para modelos de clasificación [16]. Las métricas de evaluación del rendimiento de un modelo de aprendizaje automático son esenciales para evaluar la calidad y el rendimiento del modelo. Permiten determinar la precisión de los resultados del modelo respecto de los resultados esperados (modelo ideal) y presentan una forma de comparar diferentes modelos entre sí.

1.4.5.1 Precisión

La precisión (accuracy en inglés) mide la proporción de predicciones correctas del modelo en relación con el total de predicciones realizadas. [16]. Por ejemplo, si se manejan 100 puntos de datos de los cuales predice 70 de ellos correctamente y 30 incorrectamente, se maneja una precisión del 70 %. Se define como el número de puntos de datos pronosticados correctamente en relación con el número total de puntos de datos. La precisión es confiable cuando las clases manejan tamaños de conjunto similares, pero el grado de confianza puede disminuir si existen clases desequilibradas.

1.4.5.2 Matriz de confusión

Si bien es necesario conocer el número de datos acertados correctamente, existen otras medidas como predicción de datos positivos correctamente y predicción de datos negativos correctamente. Estas métricas se ubican en la llamada Matriz de Error (ver Figura 1.10) o

Tabla de Confusión tal que resume las diversas predicciones. La matriz de confusión es una tabla que presenta cuatro valores [16]:

- ❑ Puntos de datos pronosticados como positivos y son positivos verídicamente.
- ❑ Puntos de datos pronosticados como positivos, pero son negativos.
- ❑ Puntos de datos pronosticados como negativos que en realidad son positivos.
- ❑ Puntos de datos pronosticados como negativos y son negativos verídicamente.

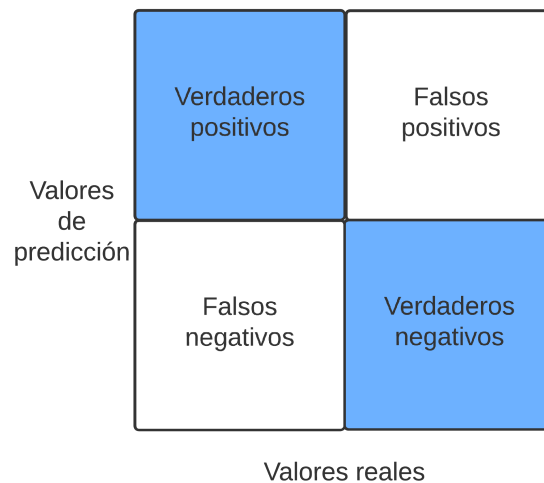


Figura 1.10: Ejemplo de una matriz de confusión de 2 clases.

1.4.5.3 Curva ROC/AUC

La Curva Característica Operativa del Receptor (Receiver Operating Characteristic, ROC) es una representación gráfica que presenta todos los modelos posibles y su rendimiento. Relaciona el valor de casos positivos que el modelo predice correctamente (sensibilidad) Respecto a la tasa de falsos positivos (1-especificidad) [17].

Esta curva mide la capacidad del modelo para distinguir entre dos clases. Al abarcar un problema multiclase, se emplea el AUC micro promedio para analizar los resultados. En la Figura 1.11 se visualiza un ejemplo de curvas ROC para un clasificador multiclase. Un modelo con una AUC de 1.0 es considerado un modelo perfecto, mientras que un modelo con una AUC de 0.5 es considerado un modelo aleatorio sin ninguna habilidad para distinguir entre las dos clases.

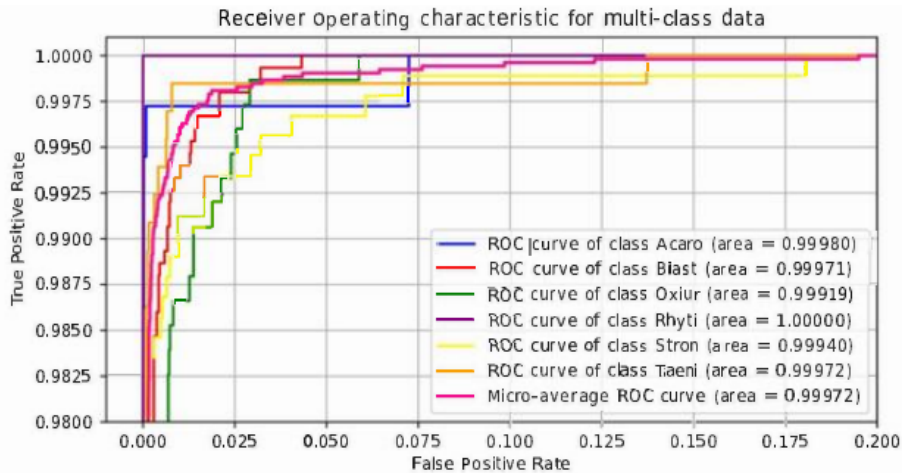


Figura 1.11: Ejemplo de una curva ROC/AUC.

1.4.5.4 t-SNE

Es una técnica de visualización de datos para graficar conjuntos de datos de varias dimensiones como datos de imágenes o de texto, en un espacio de baja dimensión. La técnica se basa en la idea de preservar la similitud entre los puntos de datos en el espacio de baja dimensión, mientras que se maximiza la separación entre los puntos que son disímiles [18].

Para transformar los datos, t-SNE primero calcula una medida de similitud entre cada par de puntos de datos en el espacio de alta dimensión. Esta medida de similitud se representa como una distribución de probabilidad en la que cada punto de datos se asigna a una probabilidad con la que es similar a los demás puntos de datos.

Luego, t-SNE transforma los datos en el espacio de baja dimensión de manera que se preserven las similitudes entre los puntos de datos. Para hacer esto, t-SNE optimiza una función de costo que mide la divergencia entre las distribuciones de probabilidad de los datos en los espacios de alta y baja dimensión. La optimización se realiza mediante un proceso iterativo en el que se actualiza la posición de los puntos de datos en el espacio de baja dimensión de manera que se minimice la divergencia entre las distribuciones de probabilidad [18].

1.4.6 ENTORNO DE SIMULACIÓN MATLAB

MATLAB es un software de computadora de alto rendimiento diseñado para realizar cálculos científicos y técnicos. Es ampliamente utilizado en el análisis de datos, el desarrollo de algoritmos y la creación de prototipos y sistemas. MATLAB es una plataforma de computación

técnica que ofrece un entorno de desarrollo integrado (IDE) y un lenguaje de programación de propósito general. La plataforma incluye una amplia gama de herramientas de visualización y análisis de datos, así como un conjunto completo de bibliotecas y funciones para tareas técnicas específicas [19].

1.4.6.1 Herramientas de aprendizaje automático mediante Matlab

MATLAB tiene una gran cantidad de funciones y herramientas para el aprendizaje automático [19]. Se destaca a continuación las actividades que se puede realizar con MATLAB en el área del aprendizaje automático:

- Preprocesar y limpiar los datos de aprendizaje
- Diseñar y entrenar modelos de aprendizaje automático usando técnicas como el aprendizaje supervisado, el aprendizaje no supervisado y el aprendizaje por refuerzo
- Evaluar el rendimiento de los modelos entrenados y elegir el mejor modelo
- Ajustar y afinar los modelos entrenados para mejorar el rendimiento
- Utilizar los modelos entrenados para hacer predicciones en datos nuevos

Para el desarrollo del trabajo, se empleó el entorno de programación y las herramientas que MATLAB ofrece para realizar una clasificación interactiva como el manejo del paquete Computer Vision Toolbox y la aplicación Classification Learner.

Computer Vision Toolbox

MATLAB se caracteriza por la presencia de soluciones específicas a problemas de aplicación denominadas cajas de herramientas (Toolbox) [19]. Utilizar estas herramientas como funciones de propósito específico de MATLAB, permiten al programa resolver problemas de mayor grado de complejidad.

De las diversas herramientas, las funciones de "Visión computacional" permiten diseñar y evaluar sistemas de procesamiento de vídeos y visión artificial. Destaca la detección, seguimiento y extracción de objetos y características. Por otra parte, los algoritmos de Deep Learning y Machine Learning permiten entrenar detectores de objetos personalizados.

Aplicación Classification Learner

Permite clasificar los datos usando varios algoritmos y comparar los resultados en el mismo

entorno [19]. El proceso consiste en seleccionar características, luego especificar esquemas de validación, posterior se entrena los modelos y finalmente se evalúa resultados.

Los modelos de clasificación disponibles en la aplicación corresponden a árboles de decisión, análisis discriminante, máquinas de vectores de soporte (SVM), regresión logística, vecinos más cercanos y clasificación de conjuntos. Esta aplicación maneja un aprendizaje automático supervisado. El modelo creado se puede exportar para predecir nuevos datos.

Para la evaluación de los resultados se manejan métodos de validación que permiten examinar la precisión predictiva de los modelos ajustados. Esta herramienta permite discriminar el mejor modelo en función del rendimiento estimado del modelo en los nuevos datos. Resalta la técnica de la validación cruzada, donde se debe elegir número de pliegues para particionar el conjunto de datos usando el control deslizante y brindar protección contra el ajuste excesivo (overfitting).

1.4.7 TRABAJOS RELACIONADOS

A continuación, se abarcan diversos estudios desarrollados que guardan similitud con el enfoque del presente trabajo.

En [20] consiste en la identificación de la mosca blanca adulta (*Bemisia tabaci*) y trips (*Frankliniella occidentalis*) en invernaderos. Se desarrolló una base de datos mediante un bloque de adquisición de imágenes en trampas adhesivas. Para la etapa de segmentación destaca el algoritmo OTSU y algoritmos de procesamiento de imágenes digitales. Para el entrenamiento del modelo se utilizó la clasificación mediante una red neuronal de avance. En [21], se abordó el planteamiento anterior obteniendo varias características morfológicas respecto al tamaño y el color de los especímenes para clasificarlos. En [22] consiste en un modelo de detección y etiquetación de infecciones a células atribuidos al parásito de la malaria. En este caso se manejan operadores morfológicos como bloque de segmentación [23], y posterior se clasifica a partir de una red neuronal donde ingresa conjuntos de características de textura y forma que alimentan.

Por otra parte, en [24], empleando el aprendizaje de transferencia, se utiliza bloques de lot y clasificadores de insectos en base a una red CNN (R-CNN) con técnicas de región Faster y ResNet 50, se construye un equipo de monitoreo remoto de trampas de insectos en tiempo real. De esta forma se obtiene un identificador automático de insectos de 94 % de precisión. Además, en [25] se despliega un clasificador de organismos protozoos y metazoos. El

modelo compara las características por análisis discriminante, los árboles de decisión y las redes neuronales. El rendimiento obtenido guarda relación entre la técnica discriminante y las redes neuronales, sin embargo, existe menor eficiencia respecto al árbol de decisión.

Finalmente, en [9] y [26], se despliega un modelo de clasificación de parásitos reptilianos utilizando redes neuronales convolucionales a partir de imágenes microscópicas de heces. Se preprocesa la base de datos mediante técnicas de binarización y segmentación [20, 22, 25]. Y para la etapa de entrenamiento se maneja un modelo de red neuronal convolucional CNN con un esquema de aprendizaje de transferencia.

Las contribuciones del trabajo se enfocan en clasificar parásitos de reptiles sin manejar recursos computacionales de gran nivel para simular redes neuronales como se realizó en [26]. Se destaca el manejo de extracción de características mediante BoVW para aprovechar sistemas de clasificación de menor nivel de complejidad como en las redes neuronales. Se utiliza el conjunto de datos públicos expuesto en [9] que contenga imágenes microscópicas de heces recopiladas y anotadas por expertos para desarrollar el contraste de técnicas, así como la complejidad de recursos computacionales.

2 METODOLOGÍA

En el presente capítulo se muestra la implementación del clasificador de parásitos basados en visión computacional en Matlab. En la sección 2.1 se utiliza una base de datos proporcionada. Luego en la sección 2.2 se realiza el preprocesado de las imágenes. De igual manera se incorpora las coordenadas donde se limita al parásito en una imagen. Se utilizan las herramientas y funciones provistas del paquete Computer Vision Toolbox de Matlab. A continuación, en la sección 2.3 se plantean los pasos de extracción de características de imágenes mediante BoVW. En la sección 2.4 se aborda la técnica SVM y validación cruzada (K-Fold cross-validation) para realizar la clasificación de los agentes parasitarios.

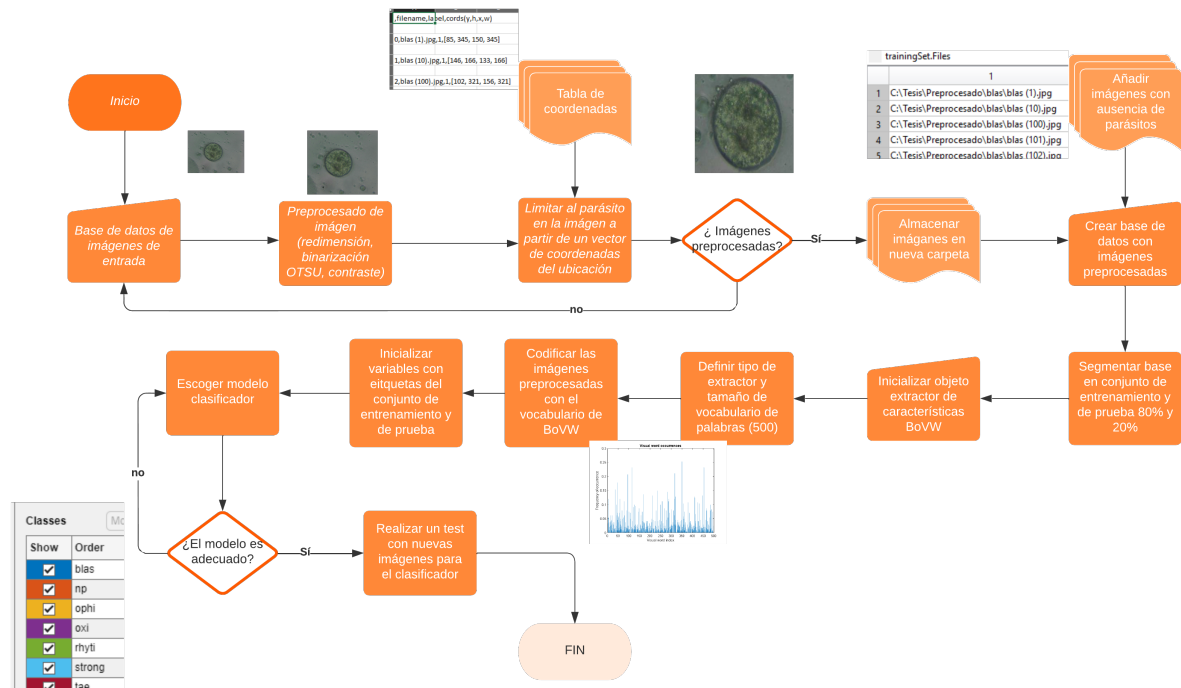


Figura 2.1: Diagrama de flujo del enfoque propuesto.

En la Figura 2.1 se visualiza el diagrama de bloques del enfoque propuesto. Se trabaja sobre una base de imágenes de determinado parásito a estimar identificar y clasificar. En primer lugar, se realiza el preprocesamiento de la imagen para ofrecer una mejor percepción de las características. Posterior se extrae características al conjunto de imágenes para localizar

los puntos de interés y asociarlo a determinada especie. Luego se codifican las imágenes utilizando el vocabulario de la bolsa de palabras visuales y posteriormente se utiliza el algoritmo Support Vector Machines (SVM) para la clasificación. A continuación, se realiza la optimización del modelo variando parámetros y finalmente realizar una predicción con un bloque de datos para la evaluación.

2.1 CONJUNTO DE DATOS

En el trabajo actual se utiliza la base de datos revisada en [9], de dominio público. En el trabajo citado, se presenta la recolección de visualizaciones de organismos parasitarios (ver Figura 2.2) en muestras de heces de reptiles etiquetadas por un veterinario especialista según el parásito presente en ellas.

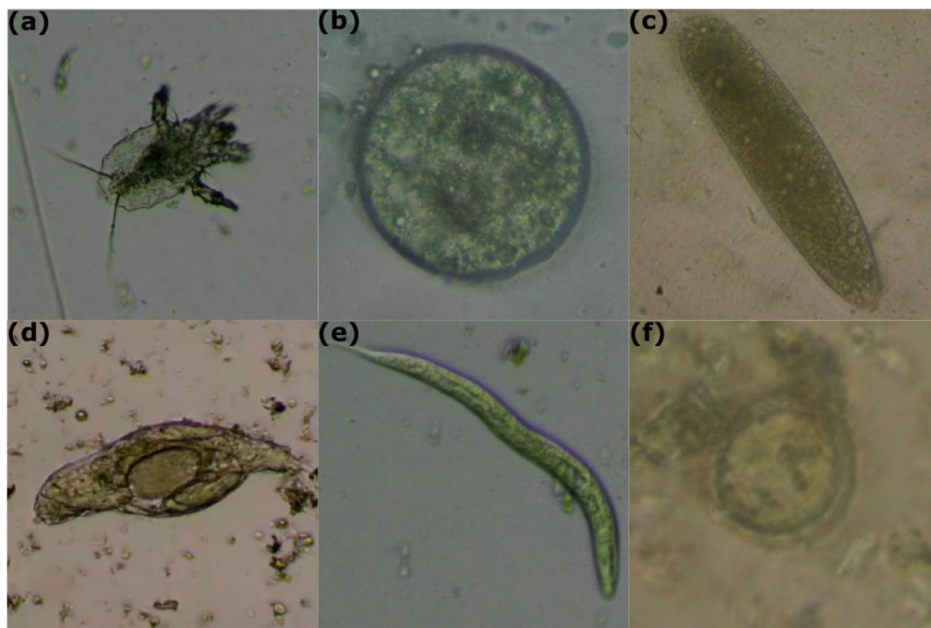


Figura 2.2: Ejemplo de parásito de la base de datos a utilizar. a) Ácaro (*Ophionyssus natricis*); b) *Blastocystis* sp; c) Huevo de oxiurdo; d) *Rhytidoides similis*; e) *Strongyloides*; f) *Taenia*

La base de datos está conformada de un total de 5360 imágenes las cuales fueron redistribuidas, donde corresponde 3616 imágenes a las categorías de los seis conjuntos de parásitos. El otro bloque de imágenes de 1744 referencia a la ausencia de algún parásito y aporta como una séptima etiqueta para clasificar imágenes que no contengan parásitos (e.g. ruido de fondo). La distribución de las imágenes de los 6 tipos de parásitos se presenta en la Tabla 2.1:

Tabla 2.1: Arreglo del conjunto de datos de agentes parásitos.

Parásito	Imágenes
"blas" Blastocystis sp	950
"ophi" Acaro (Ophionyssus natricis)	245
"oxi" Oxiurdo huevo	345
"rhyti" Rhytidoides similis	1072
"strong" Strongyloides	648
"tae" Taenia	356
TOTAL	3616

2.2 PREPROCESAMIENTO DE LAS IMÁGENES

En este apartado se revisa a detalle la preparación de las imágenes para su uso en el entrenamiento y la evaluación de un modelo de clasificación mediante tareas de redimensionamiento, eliminación de ruido, recorte de imágenes y distinción de características relevantes de las imágenes.

Para este bloque se van a generar nuevas imágenes en la carpeta del computador. Debido a que se trabaja con una gran cantidad de elementos, se hace uso de un almacén de datos en Matlab [27] para administrar de forma individual en la memoria a medida que el programa las necesita (en lotes en lugar de todas al mismo tiempo). Con este objetivo, las imágenes fueron procesadas mediante los siguientes pasos :

1. Las imágenes del conjunto de datos están separadas en subcarpetas para cada clase. Se genera un almacén de datos de imágenes a partir de las rutas.
2. Se define las extensiones de archivo de tipo jpg y png para la validación de las imágenes. Además se incluye los nombres de las subcarpetas como nombres de etiquetas.
3. A continuación se importa una tabla coordenadas de la base de datos que permite aislar la región de interés donde se encuentra el parásito, provisto de [9]. Este punto permite realizar el bloque de segmentación, el cual abarca un conjunto de técnicas para identificar al parásito que forma parte de futuros trabajos de investigación.
4. De la tabla obtenida, se utiliza las columnas que manejan las coordenadas espaciales, especificados como un vector numérico de 4 elementos. Se debe colocar el orden de las columnas como xmin, ymin, width y height para definir el rectángulo de recorte en Matlab.
5. Se debe garantizar que el rectángulo a utilizar corresponda con la imagen respectiva,

por lo que se usa el nombre de la imagen para verificar con cada elemento de la columna filename.

Luego de las técnicas aplicadas, se realiza la etapa de segmentación con cada imagen.

2.2.1 PROCESO

A continuación se preparan las imágenes para alcanzar un buen rendimiento en la etapa de entrenamiento de la red, debido a la presencia de ruido en las imágenes de la base de datos (imágenes se tomaron de heces de animales con nivel de impureza). En la Figura 2.3 se puede visualizar las siguientes estrategias de segmentación:

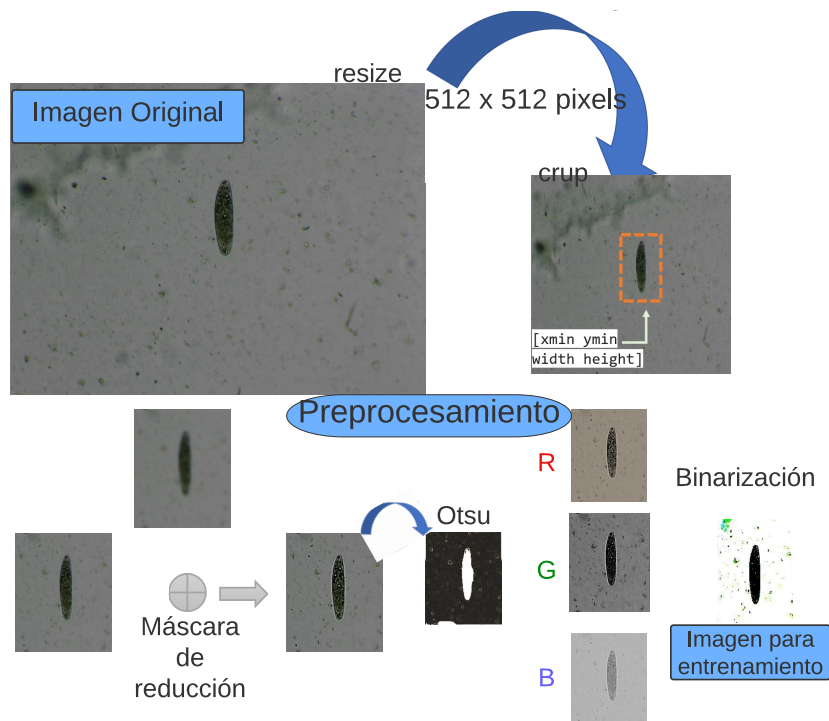


Figura 2.3: Supresión de ruido de fondo ajustando parámetros y características de las imágenes. Adaptado de [8].

1. Se redimensiona las imágenes a 512 x 512 píxeles mediante el uso del comando `imresize`.
2. Se recorta el parásito para limitar el objeto de interés suprimiendo el ruido de fondo. Se utiliza la función `imcrop` indicando la imagen y el rectángulo de coordenadas para determinado elemento [27].
3. Se utiliza la función `imsharpen` definiendo como parámetros un filtro paso bajo gaussiano con desviación estándar 2 y el valor de la unidad para resaltar la nitidez de

regiones estrechas cercanas a las fronteras [27].

4. Binarización a través del método de Otsu. Se aplica un umbral global a partir de la imagen de escala de grises y el método OTSU (varianza interclase de píxeles blancos y negros). Luego se guardan los datos en uint8.

Una vez preparado y procesado adecuadamente las imágenes, se pueden utilizar para la extracción de características (selección y representación de los aspectos más importantes de una imagen). Se añade el bloque de no parásitos (no parasites, np) para generar un clasificador de siete categorías, las cuales están en la base de datos imds2.

2.3 EXTRACCIÓN DE CARACTERÍSTICAS

Para la obtención de características, se utilizó el modelo de Bolsa de Palabras Visuales (Bag of Visual Words, BoVW) como se visualiza en la Figura 2.4, que se explica a continuación:

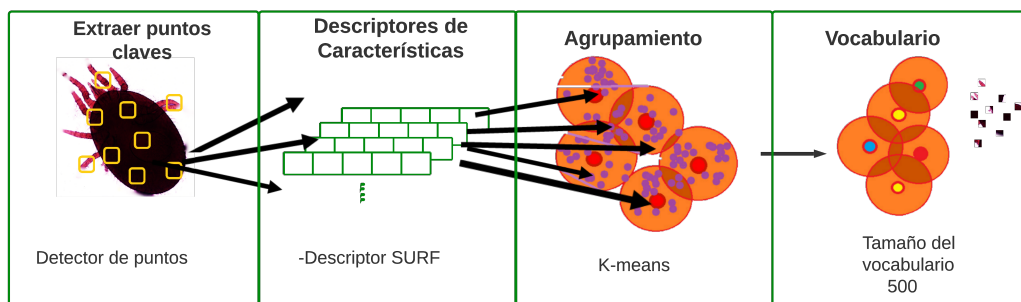


Figura 2.4: Representación del proceso de flujo de trabajo del algoritmo Bolsa de Palabras Visuales con los datos del presente trabajo: Se maneja la técnica detección de puntos en conjunto con el descriptor SURF, luego se realiza el agrupamiento de los centroides definiendo un total de 500 palabras en el vocabulario.

1. Se inicializa un nuevo bloque de datos de almacenamiento con las imágenes procesadas y con la carpeta de no parásitos (np). A continuación se divide en conjuntos de entrenamiento y validación en proporción de 4 a 1 (80 % y 20 %).
2. Se inicializa un objeto de bolsa de palabras visuales "bagOfFeatures"[28]. Se utiliza el objeto ingresando un grupo reducido de imágenes del conjunto de entrenamiento para identificar los valores de vocabulario con el algoritmo de K-means y fracción de características.
3. Para la extracción de los puntos clave o también llamados puntos de interés, se va a utilizar la técnica de detección de puntos de Características Robustas Aceleradas

(Speeded Up Robust Features, SURF) principalmente porque esta técnica es más eficiente respecto al algoritmo Transformada de Características Invariantes a la Escala (Scale Invariant Feature Transform, SIFT).

4. Luego, para cada punto de interés detectado en el paso anterior, el algoritmo analiza su vecindad y extrae un descriptor de características. Específicamente, se empleó el descriptor SURF de 64 dimensiones por defecto en Matlab.
5. Posteriormente, sobre los descriptores de características, se utiliza el algoritmo k-means como método de agrupamiento que permite trabajar con k clústeres. Con el proceso de agrupamiento (clustering) es posible agrupar los descriptores que poseen características similares.
6. Después de obtener los grupos (clústeres), se determina los centroides de cada clúster. Como resultado, las palabras visuales del vocabulario son los centroides de los clústeres que corresponde a k elementos.
7. De esta manera se utiliza de nuevo el objeto, pero inicializando el conjunto de entrenamiento completo, parámetros de tamaño de vocabulario de $k = 500$ y la fracción de características más fuertes a usar de cada etiqueta en la entrada del conjunto de 0.8, que son los valores obtenidos de los pasos anteriores.
8. Al generarse el vocabulario, se codifica los conjuntos de entrenamiento y prueba con el vocabulario [28]. De esta forma se tiene la representación de cada imagen mediante el vocabulario obtenido. Finalmente se extrae las etiquetas de los conjuntos.

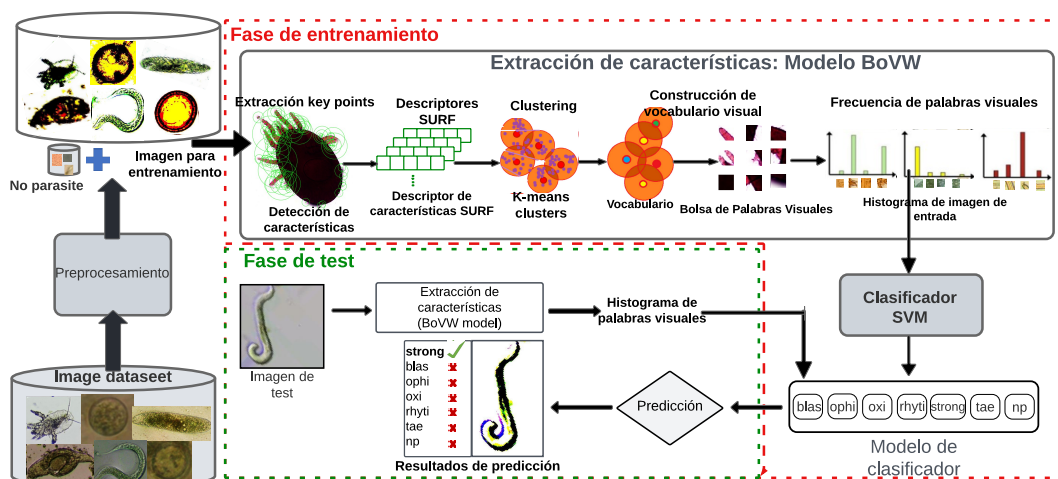


Figura 2.5: Clasificación de Imágenes con Bolsa de Palabras Visuales.

De esta manera, se preparan los datos para el bloque de entrenamiento y validación (ver Figura 2.5).

2.4 ENTRENAMIENTO Y VALIDACIÓN DE LOS DATOS

En esta parte se efectúa tanto la clasificación como el proceso de validación cruzada a partir de un bloque de pasos (ver Figura 2.6). Se abre la aplicación de clasificación de aprendizaje ingresando en la ventana de comandos “classificationLearner”. A continuación en una nueva sesión se vincula los conjuntos de entrenamiento y evaluación con las etiquetas en la aplicación. Se activa la técnica de validación cruzada en grupos de $K=5$. De esta forma la aplicación protege contra el sobreajuste. Finalmente se utiliza el conjunto de test para evaluar el rendimiento del modelo en un conjunto de datos nuevos.

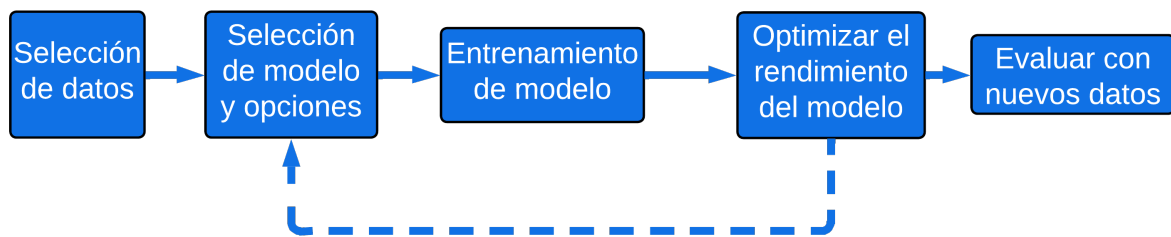


Figura 2.6: Flujo de trabajo de la aplicación de aprendizaje de clasificación.

2.4.1 SELECCIÓN DEL CLASIFICADOR Y APLICACIÓN DE K-FOLD

En primer lugar se entrena automáticamente una selección de diferentes modelos de clasificación en los datos con el fin de explorar modelos prometedores de forma interactiva. En la herramienta “Modelos”, se selecciona All Quick-To-Train para entrenar con todos los ajustes preestablecidos del modelo disponibles para el conjunto de datos que se ajustan rápidamente.

Aparece una selección de tipos de modelo en el panel Modelos. Cuando los modelos terminan el entrenamiento, el mejor puntaje porcentual de Precisión (Validación) se destaca en un recuadro.

A continuación se procede a optimizar el modelo que mejor se adapte al conjunto de prueba. Se emplea el modelo Máquinas de Vectores de Soporte (SVM) que es un clasificador de algoritmo supervisado utilizado en la discriminación de clases. Se pueden utilizar varios kernels como el kernel lineal o kernel cúbico, [14], cambiar la restricción de cuadro de

una SVM, variar la escala de kernel, definir el método multiclase o estandarizar los datos. De una forma guiada se utiliza la optimización de hiperparámetros dentro de la aplicación para automatizar la selección de valores de hiperparámetros mediante la optimización bayesiana. Para un determinado modelo, la aplicación establece valores de hiperparámetros aleatorios y luego se van ajustando al minimizar el error de clasificación del modelo, obteniendo el modelo con los valores de parámetros que mejor se ajustan. Se limita el tiempo de optimización para garantizar un método seguro de finalización de la actividad.

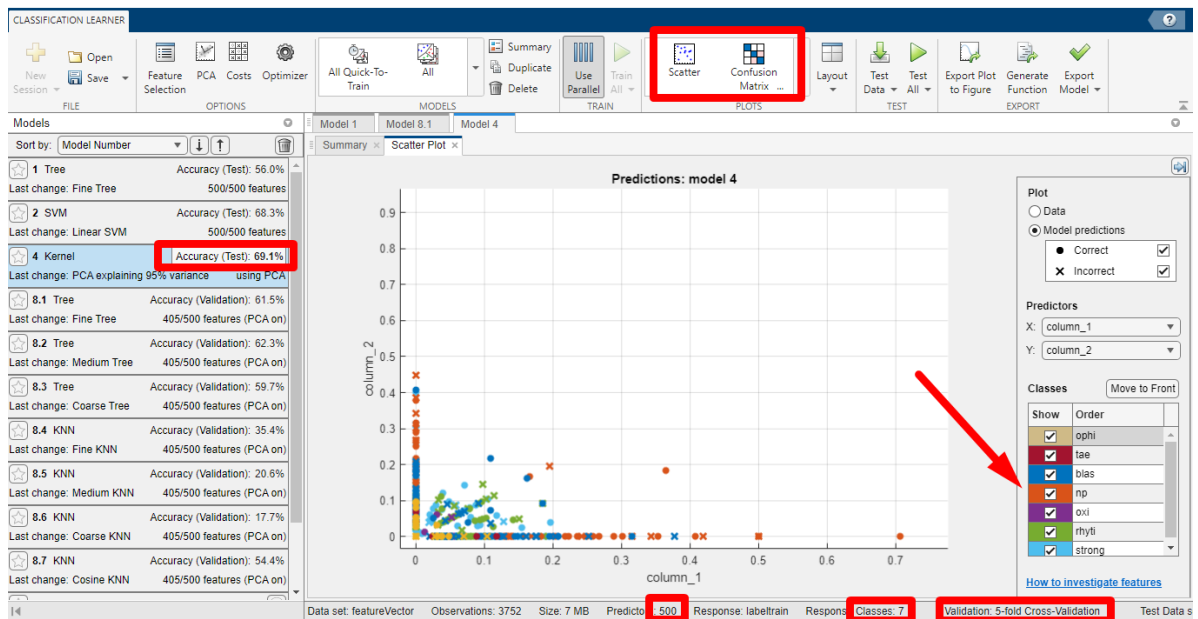


Figura 2.7: Sesión de modelos para entrenamiento de los diversos modelos. Se visualiza el bloque de resultados y el modelo de mejor performance.

Luego del gráfico de error de clasificación mínimo se selecciona la iteración que corresponde a los hiperparámetros optimizados. De esta forma se obtiene el clasificador SVM con la función de kernel cúbico que es un tipo de kernel polinómico de grado 3, con método multiclase One vs One, nivel de restricción de cuadro de 0.18187 y sin estandarización de datos.

2.4.2 MÉTRICAS DE EVALUACIÓN

Debido a que el ajuste de hiperparámetros a menudo conduce a modelos sobre ajustados, se verifica el rendimiento del modelo SVM optimizable en un conjunto de prueba (imágenes que no fueron utilizadas en el entrenamiento) y posterior se compara con el rendimiento del mejor modelo SVM preestablecido. Se utilizan los datos que se reservó para realizar pruebas (evaluación). De esta forma se evidencia la precisión del rendimiento del modelo

en datos nuevos.

En la sección Test de la ventana principal, seleccionando en la flecha de Test All, especificamos el modelo seleccionado. La aplicación calcula el rendimiento del conjunto de prueba del modelo entrenado en el resto de los datos, es decir, en los datos de validación. Se puede revisar la curva ROC y la matriz de confusión para el conjunto de entrenamiento, así como del conjunto de prueba. La visualización de los resultados para el conjunto es similar como el que se visualiza en la Figura 2.7.

Por ende, se realiza una comparativa visual del rendimiento del conjunto de prueba del modelo escogido. Al manejar un problema multiclase, se utiliza la curva AUC micropromedio que presenta la tasa de verdaderos positivos (TPR) respecto a la tasa de falsos positivos (FPR) sobre diferentes umbrales de puntajes de clasificación, calculados por el clasificador seleccionado. De esta manera, si se genera una tasa de falsos positivos de 0,4 corresponde a asignar incorrectamente el 40% de las observaciones de la clase negativa a la clase positiva. En cambio, la asignación correcta del 90% de las observaciones de clase positiva corresponde a una tasa positiva verdadera de 0,9.

Finalmente, se utiliza la validación cruzada de cinco veces y la precisión general de todas las ejecuciones para resumir los valores en la matriz de confusión. De esta manera se identifica valores donde el clasificador se desempeñó mal examinando las celdas fuera de la diagonal que muestran números altos y son rojas (los puntos de datos están mal clasificados). Se puede visualizar los valores en porcentajes seleccionando la opción de Tasas positivas verdaderas.

3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

A continuación se abordan los resultados de la utilización de los modelos de entrenamiento y la optimización del clasificador SVM cúbico en la base de datos de las imágenes de parásitos en heces de reptiles, empleando k-fold cross validation y métricas tales como, la precisión, y matriz de confusión entre otras para evaluar el rendimiento del clasificador.

3.1 RESULTADOS OBTENIDOS EN LA SEGMENTACIÓN Y APLICACIÓN DEL VOCABULARIO DE BOVW

Los cambios aplicados en la etapa de procesamiento de las imágenes permite mejorar la precisión del clasificador como la segmentación de imágenes y algunas técnicas adicionales de procesamiento de imágenes. La técnica OTSU es un algoritmo de segmentación de imágenes que busca optimizar la separación entre dos grupos de píxeles en una imagen, lo que puede ser útil para eliminar el ruido de fondo y enfocar la atención en los objetos de interés. Mejorar la nitidez de las imágenes y recortar las imágenes para centrar el enfoque en el parásito también pueden ayudar a mejorar la precisión del algoritmo. Es importante tener en cuenta que la precisión del algoritmo depende en mayor proporción de la calidad y cantidad de los datos de entrenamiento y de la elección adecuada de los parámetros.

La Figura 3.1 muestra la aplicación del bloque de preprocesamiento para un ejemplar de la clase oxi, en contraste con imágenes de la categoría de no parásitos. Al limitar el organismo, se mejora en la precisión de detectar un parásito o determinar su ausencia. Por otro lado, la técnica de BoVW destaca por utilizar determinadas características como color, forma, o textura. Por lo que al usar la técnica de escala de grises, se disminuye el nivel de precisión. De esta manera es importante seleccionar las técnicas que permitan utilizar de mejor manera al modelo. Identificar nuevas técnicas y enfoques para mejorar los resultados forma parte de futuros casos de estudio.

Respecto a los resultados utilizando k-means, debido a que se trabaja con un vocabulario visual, cada palabra corresponde a ciertos rasgos de las imágenes, los cuales son agrupa-

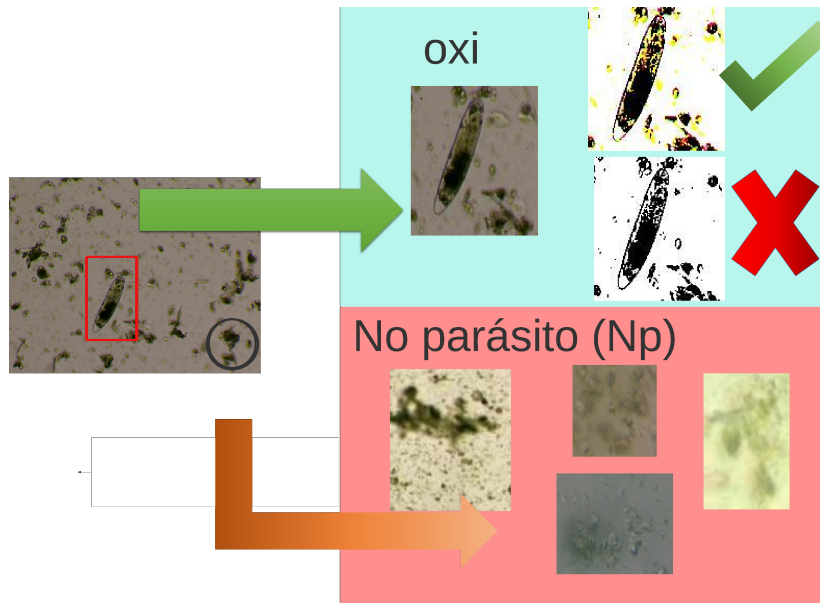


Figura 3.1: Ejemplo de preprocesamiento con técnicas de redimensión, contraste, umbral con OTSU y contraste a escala de grises.

dos de acuerdo a su semejanza. De esta manera en la Figura 3.2 se visualiza 2 ejemplares de cada clase donde se identifica los rasgos significativos, así como su emparejamiento y posterior visualización de las repeticiones de los patrones mediante barras. Para el caso de la clase ophi, las extremidades consisten en un rasgo utilizado para la identificación del parásito. Mientras que para la clase strong, destaca la superficie que presenta a la mitad de su estructura.

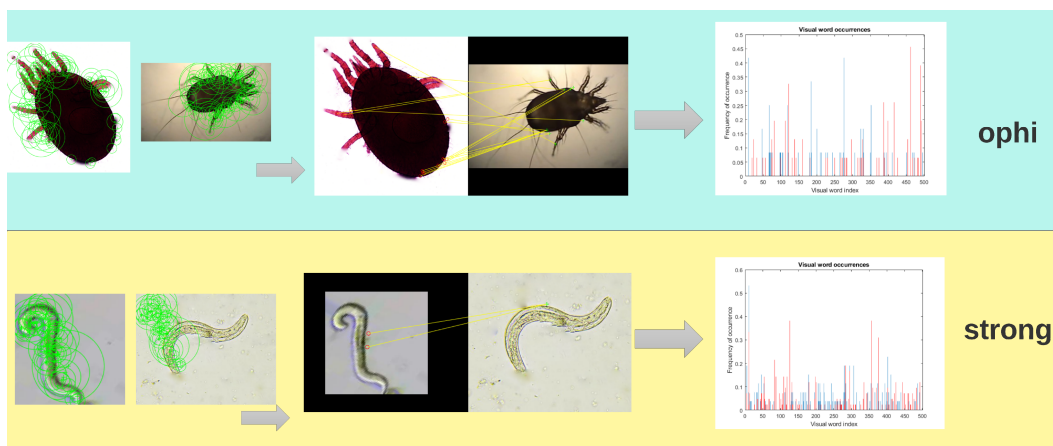


Figura 3.2: Visualización de rasgos significativos y conteo de palabras mediante histograma para muestras de la clase ophi y strong.

De esta forma se cataloga determinada imagen de acuerdo con ciertos rasgos, en lugar de procesar la imagen en su totalidad. Para una base de datos amplia, implica la ejecución del algoritmo sin gran demanda de recursos computacionales. Note en la Figura 3.2, como

el histograma para la categoría ophi guarda relación mientras existe diferenciación con la categoría strong.

Respecto a la aplicación de la técnica de BoVW, en la Figura 3.3 se presentan los resultados de la técnica SURF para identificar los rasgos de cada categoría y su vocabulario representado en histograma.

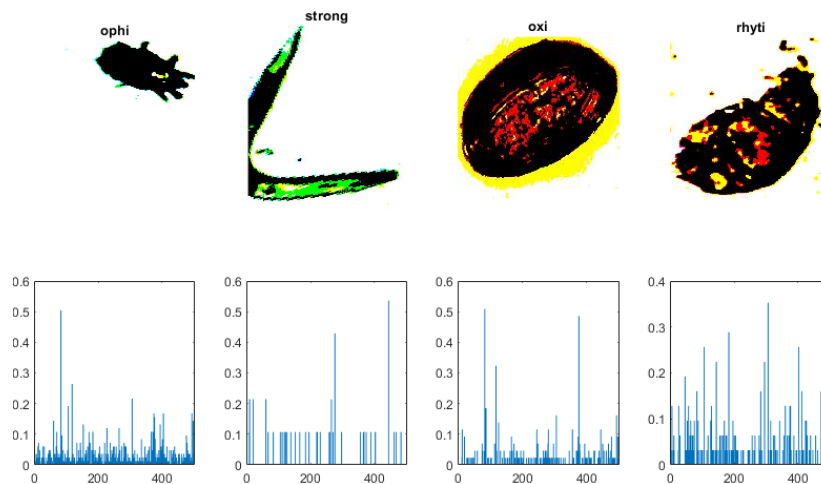


Figura 3.3: Resultados del vocabulario de BoVW mediante histograma.

De esta manera se asigna cada característica de la imagen al clúster más cercano (la palabra visual más cercana). Posteriormente se contabiliza el número de veces que se asignó las características de la imagen a cada palabra visual.

3.2 RESULTADOS EN EL ENTRENAMIENTO DEL MODELO

En el presente trabajo, el espacio de características es de 500 dimensiones para cada imagen que fue codificada con el vocabulario de la técnica BoVW que corresponde a un vector de $k = 500$. Entonces, se incorporan los vectores de características de alta dimensión a dos dimensiones, los cuales sí se pueden visualizar fácilmente en un plano 2D guardando las similitudes entre los puntos. La Figura 3.4 muestra un ajuste de baja dimensión debido a que se maneja datos en un espacio de características de alta dimensión. Se reduce la dimensión de los datos de 500 dimensiones originales a 100 usando PCA, y luego a dos usando el algoritmo t-SNE de algoritmo exacto que optimiza la divergencia de distribuciones entre el espacio original y el espacio incrustado. Los efectos de la configuración del algoritmo son difíciles de predecir. A veces pueden mejorar la agrupación, pero en su mayor parte la configuración predeterminada es utilizada. Entonces para el caso se considera la métrica de distancia euclidiana estandarizada para tratar de obtener una mejor separación de los

conjuntos.

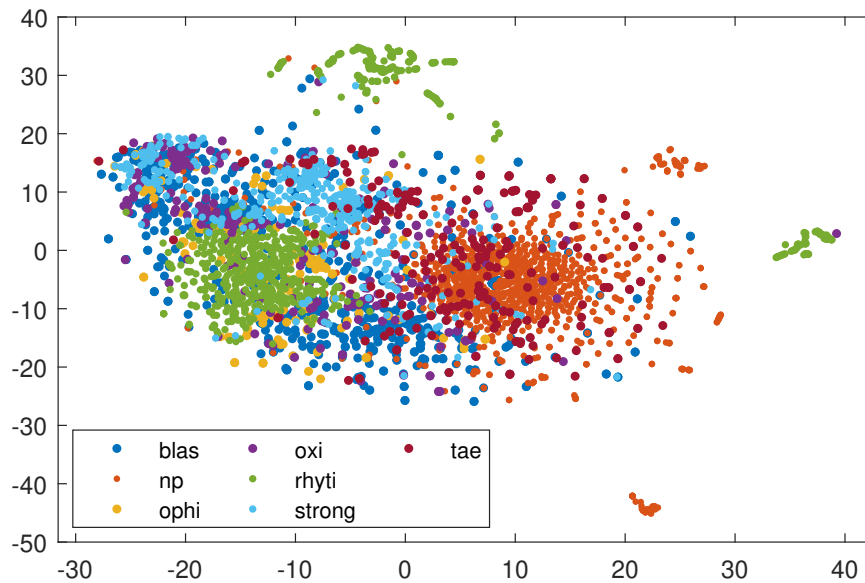


Figura 3.4: Plano de visualización para las 7 clases mapeando las características de alta dimensión en características de baja dimensión.

Es importante recalcar cierta superposición entre las clases a razón de las similitudes en la morfología y el color. Note como la categoría de no parásito "np" se encuentra alejada de las demás categorías, siendo la categoría de "tae" la que se superpone. Al revisar el conjunto de datos de una clase, se visualiza al organismo en diferentes posiciones, lo cual incide en la agrupación de los resultados. Es así que existe superposición de todas las clases.

De acuerdo a los resultados de la Figura 3.4, se observa que las muestras no se pueden separar linealmente, lo que implica la elección de un clasificador no lineal. Para el trabajo presente se maneja SVM del núcleo polinomial (de grado cúbico).

Por otra parte, a pesar de que se realizaron pruebas con diversos modelos de clasificación, se obtuvo los mejores resultados con SVM y función de núcleo cúbico, alcanzando un mejor rendimiento. La Tabla 3.1 presenta los valores de precisión de los diversos modelos con el conjunto de datos de entrenamiento. Se resalta el valor de mejor precisión que corresponde a los modelos de SVM.

Si se procede a variar el número de validación cruzada, se tiene para fold = 5 el mejor valor de precisión frente a fold = 4 y fold = 6, aunque la diferencia entre ellos no sea muy notoria.

Si se varía la distribución asignada para el conjunto de entrenamiento y evaluación, se puede observar que para la distribución 70 % a 30 %, 75 % a 25 % y 80 % a 20 %, se obtiene

Tabla 3.1: Valor de precisión con diversos modelos de clasificación.

Modelo	Segmentación (train) %	K-fold	PCA %	Accuracy %
Fine Tree	70	5	-	56.5
Fine Tree	70	5	95	62
Fine Tree	70	6	-	56.3
Fine Tree	70	4	-	56.3
Fine Tree	75	5	-	56.2
Linear SVM	70	5	-	67.9
Linear SVM	70	5	95	67.5
Linear SVM	80	5	-	68.7
Cubic SVM	80	5	-	70.3
Conjunto Discriminante subespacial	80	5	-	67.3
Red neuronal amplia	80	5	-	65.7
Conjunto de árboles embolsados	80	5	-	65.4
Kernel de regresión logística	80	5	-	65.4

el mejor resultado para la última distribución (80 a 20).

La Figura 3.5, en cambio presenta los resultados de la optimización Bayesiana de hiper parámetros empleando el clasificador SVM y variando los diversos parámetros como núcleo, caja de restricción o estandarización. Se indica el resultado de los parámetros que ofrecen el menor error de costos que ocurre en la onceava iteración. Cabe señalar que el valor de BoxConstraint cumple el rol de constante de penalización (llamada también como regularización) que evita el sobreajuste (overfitting). Para reducir el tiempo de simulación y utilizar gran carga computacional se habilita el manejo de PCA utilizando para una compresión del 95 %, 412 de los 500 predictores.

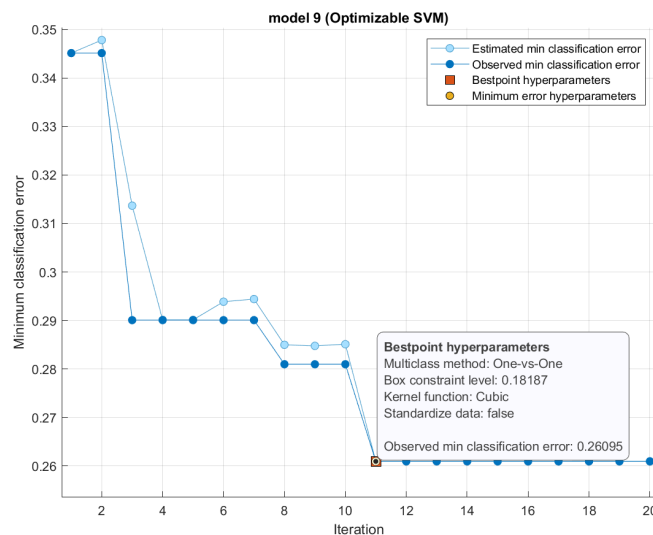


Figura 3.5: Resultados de la optimización de hiperparámetros para el modelo de SVM.

Ahora bien, en la Figura 3.6 se pueden observar una representación de la tasa de verdaderos positivos (TPR) frente a la tasa de falsos positivos (FPR) mediante las curvas ROC de

cada clase empleando el clasificador SVM utilizado en este trabajo.

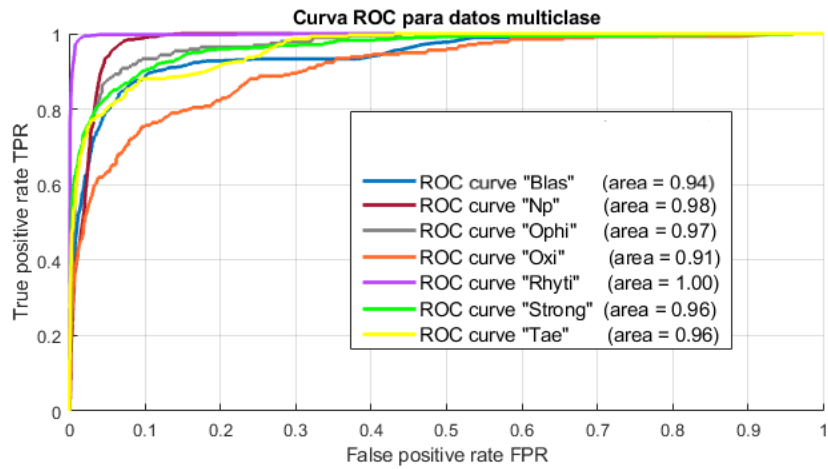


Figura 3.6: Gráfica ROC para el modelo SVM cúbico.

Se tiene el menor valor del Área Bajo la Curva ROC (AUC) para la clase oxi de 91 % mientras que el mayor valor de 100 % corresponde a la clase rhyti. Además, el valor micropromedio de los aportes de las clases antes de calcular la precisión promedio corresponde a 96 %.

Los valores de AUC están en el rango de 0 a 1, y los valores de AUC más altos indican un mejor rendimiento del clasificador.

Model 2 (Cubic SVM)

True Class \ Predicted Class	blas	np	ophi	oxi	rhyti	strong	tae
blas	619	51	12	17	15	42	4
np	23	1340		5		4	23
ophi	29		118	15	13	21	
oxi	68	8	16	135	11	37	1
rhyti	8			5	843	2	
strong	52	19	8	30	1	406	2
tae	25	98		2		4	156

Figura 3.7: Matriz de confusión para las categorías del clasificador

En la Figura 3.7, se puede ver la matriz de confusión del clasificador SVM cúbico utilizado. Como resultado, se observa en la matriz de confusión que la categoría "np" presenta una mejor predicción de las imágenes (las celdas diagonales muestran dónde coinciden la clase verdadera y la clase predicha). Mientras que las categorías de "tae" y "oxi" tienen mayor grado de confusión como se esperaba por la superposición en la gráfica de T-SNE. Las demás clases tienen una confusión moderada. En general, existe confusión entre todas las clases a excepción de la clase rhyti que presenta un alto grado de precisión. Es de recordar que la distribución de imágenes para las 7 clases es asimétrica. Un factor que influye para el aprendizaje del algoritmo. Al equilibrar la distribución de imágenes se estima mejorar el rendimiento de precisión, tal como se planteó en el aumento de datos en [9].

Con respecto a trabajos anteriores, se comparan en la Tabla 3.2 los resultados del trabajo presente con lo desarrollado en [26], que maneja una precisión de 98,66 %, frente al 84,4 % que se obtuvo en el presente trabajo que no utiliza una gran cantidad de recursos computacionales, ni se emplean redes profundas. Además, es de notar que se incluye una nueva categoría que consiste en no parásitos, siendo 7 categorías de predicción frente a las 6 categorías del trabajo contrastado.

Tabla 3.2: Comparación de clasificador de parásitos mediante BoVW y aprendizaje por transferencia (deep learning).

Modelo	Enfoque	Database	Data Augmentation (total de imágenes)	Clases	Precisión %
Visión Computacional	BoVW y SVM	5360	5360	7	84.40
Red pre entrenada	MobileNet	3616	10099	6	98.66
Red no pre entrenada	RNC personalizada	3616	10099	6	95.30

Además, se puede examinar que en [26] utilizaron bases de datos de 10099 imágenes. Mientras tanto en el presente trabajo, se empleó una base de datos compuesta por 5360 imágenes. Esto debido a que se aplicó el aumento de datos (Data Augmentation) como una técnica para redes neuronales con el fin de trabajar en la precisión. Es de recordar que en la visión computacional es una ventaja el trabajar con base de datos con menor cantidad de datos frente a redes neuronales.

3.3 CONCLUSIONES

En el presente trabajo de titulación, se propuso un sistema de clasificación de parásitos que emplea una etapa de preprocesamiento de imágenes para limitar el ruido de fondo, luego se extrajeron las características de estas imágenes con el modelo BoVW y finalmente para la clasificación, se utilizó un clasificador SVM de kernel cúbico con k-fold cross-validation. Al estudiar sobre el proceso manual de clasificación de parásitos luego de la recolección de muestras, fue posible aprovechar la ventaja del proceso de toma de muestras con el microscopio de forma estática para definir la técnica de extracción de características. La técnica SURF fue adecuada para el proceso ofreciendo mejor tiempo de respuesta y sin comprometer su robustez frente a cambios de luz o rotación.

Los resultados indican que se alcanzó un buen rendimiento del clasificador SVM cúbico, donde el AUC fue de 96% y la precisión de 84.4%. Consecuentemente, resulta eficiente para determinar la presencia de algún agente parasitario. No obstante, al determinar específicamente la clase de parásito existe una tasa menor como se visualizó en la matriz de confusión para la predicción errónea en las diversas categorías.

El modelo de Bolsa de Palabras Visuales (BoVW) se basó en el descriptor SURF que es invariante a la rotación y traslación. Sin embargo, este descriptor no es invariante a las transformaciones de proyección y afines, lo que implica que la cámara debe estar en un lugar fijo. Este proceso es ventajoso en la obtención de muestras en el microscopio que son fijos para la toma de muestras. Sin embargo, hay algunos problemas con BoVW, como la necesidad de seleccionar manualmente el tamaño y el número de celdas o la dependencia de un diccionario de palabras predefinido. Algunos enfoques más modernos, como el aprendizaje profundo, han demostrado tener resultados superiores en tareas de clasificación específicas.

El trabajo presente utiliza un extractor de características para la clasificación de imágenes de parásitos, existiendo dificultades respecto a la variabilidad en las imágenes y a la presencia de otros objetos o partículas que pueden dificultar la detección de parásitos. Sin embargo, con el enfoque adecuado y algunos ajustes, debería ser capaz de obtener un rendimiento aceptable en caso de que la tolerancia a los errores implique consecuencias aceptables en su aplicación. Ciertamente, deep learning es un framework donde el algoritmo aprende las características por sí mismo. No obstante, al incluir una nueva clase, y obtener resultados de precisión cercanos a los del trabajo que utiliza redes neuronales pre

entrenadas, se resalta el trabajo que brinda este modelo de aprendizaje.

Al obtener los mejores resultados con el modelo SVM de kernel cúbico es trascendental tener en cuenta que a medida que aumenta el grado del kernel polinómico, también aumenta la complejidad del modelo y el riesgo de sobre ajuste (es decir, que el modelo se ajuste demasiado a los datos de entrenamiento y no generalice bien a datos nuevos). Los mecanismos de segmentación de imágenes son de gran importancia, ya que la mayoría de las imágenes, debido a la naturaleza de su adquisición por los mecanismos coprológicos, presentaban ruido de fondo que afecta en el grado de precisión. Además, el manejo de las técnicas tanto en esta etapa como K-fold en bloques posteriores, implicaba que el tiempo de procesamiento sea mayor. Al usar el modelo SVM con el extractor BoVW, se minimiza recursos computacionales, en lugar de usar por ejemplo redes neuronales. De esta forma se ha abordado y aplicado los conceptos de aprendizaje supervisado.

Es importante destacar que t-SNE no es una técnica de clasificación y no garantiza una separación perfecta de los datos. La separación de los datos con t-SNE depende de la calidad de los datos y de la adecuación del modelo a los datos, por lo que es importante experimentar con diferentes parámetros y técnicas para encontrar la mejor solución.

3.4 RECOMENDACIONES

Es importante tener en cuenta que la precisión puede variar en función del conjunto de datos y del algoritmo utilizado. Por lo que puede ser necesario ajustar los parámetros o utilizar técnicas adicionales para mejorar los resultados. Se recomienda enfocar esfuerzos en identificar técnicas para procesar y analizar las imágenes, como la detección de objetos o el análisis de características de imágenes.

Frente al método actual de obtención de datos etiquetados de parásitos que demanda un trabajo intenso y experiencia, destaca la importancia de trabajos futuros que concentren esfuerzos en el aumento de datos como la aplicación de redes generativas tipo adversaria (GANS).

4 REFERENCIAS BIBLIOGRÁFICAS

4.1 BIBLIOGRAFÍA

- [1] O. T. Carvajal, "Lista actualizada de las lagartijas de Ecuador con comentarios acerca de su diversidad," *Revista Ecuatoriana de Medicina y Ciencias Biológicas*, vol. 32, no. 1-2, pp. 119–133, 2011.
- [2] M. E. Rodríguez Espinosa, "Identificación de parásitos intestinales presentes en reptiles en cautiverio en dos centros de manejo de fauna silvestre," B.S. thesis, Quito: UCE, 2015.
- [3] K. A. Núñez Alverca, "Identificación de parásitos con diferentes métodos coprológicos en muestras de reptiles en el vivarium de Quito," B.S. thesis, 2021.
- [4] X. Peng, L. Wang, X. Wang, and Y. Qiao, "Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice," *Computer Vision and Image Understanding*, vol. 150, pp. 109–125, 2016.
- [5] L. J. Sandoval Serrano *et al.*, "Algoritmos de aprendizaje automático para análisis y predicción de datos," *Revista Tecnológica*; no. 11, 2018.
- [6] P. Sodhi, N. Awasthi, and V. Sharma, "Introduction to machine learning and its basic application in python," in *Proceedings of 10th International Conference on Digital Strategies for Organizational Success*, 2019.
- [7] A. Gholamy, V. Kreinovich, and O. Kosheleva, "Why 70/30 or 80/20 relation between training and testing sets: A pedagogical explanation," 2018.
- [8] L. Suca and G. Gómez, "Visión Computacional," *Instituto Nacional de Astrofísica, Óptica y Electrónica INAOE*.
- [9] C. Parra, F. Grijalva, B. Núñez, A. Núñez, N. Pérez, and D. Benítez, "Automatic identification of intestinal parasites in reptiles using microscopic stool images and convolutional neural networks," *Plos one*, vol. 17, no. 8, p. e0271529, 2022.

- [10] M. Huang, W. Yu, and D. Zhu, "An improved image segmentation algorithm based on the otsu method," in *2012 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*. IEEE, 2012, pp. 135–139.
- [11] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006, vol. 4, no. 4.
- [12] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, 2010, pp. 270–279.
- [13] P. Panchal, S. R. Panchal, and S. Shah, "A comparison of sift and surf," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 1, pp. 323–327, 2013.
- [14] M. Hussain, S. K. Wajid, A. Elzaart, and M. Berbar, "A comparison of svm kernel functions for breast cancer detection," in *2011 eighth international conference computer graphics, imaging and visualization*. IEEE, 2011, pp. 145–150.
- [15] J. Gareth, W. Daniela, H. Trevor, and T. Robert, *An introduction to statistical learning: with applications in R*. Spinger, 2013.
- [16] S. Visa, B. Ramsay, A. L. Ralescu, and E. Van Der Knaap, "Confusion matrix-based feature selection." vol. 710, no. 1, 2011, pp. 120–127.
- [17] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.
- [18] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.
- [19] G. Ciaburro, *MATLAB for Machine Learning*. Packt Publishing Ltd, 2017.
- [20] K. Espinoza, D. L. Valera, J. A. Torres, A. López, and F. D. Molina-Aiz, "Combination of image processing and artificial neural networks as a novel approach for the identification of bemisia tabaci and frankliniella occidentalis on sticky traps in greenhouse agriculture," *Computers and Electronics in Agriculture*, vol. 127, pp. 495–505, 2016.

- [21] J. Cho, J. Choi, M. Qiao, C. Ji, H. Kim, K. Uhm, and T. Chon, "Automatic identification of whiteflies, aphids and thrips in greenhouse based on image analysis," *Red*, vol. 346, no. 246, p. 244, 2007.
- [22] M. I. Razzak, "Automatic detection and classification of malarial parasite," *International Journal of Biometrics and Bioinformatics (IJBB)*, vol. 9, no. 1, pp. 1–12, 2015.
- [23] A. Hanbury, "The morphological top-hat operator generalised to multi-channel images," in *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, vol. 1. IEEE, 2004, pp. 672–675.
- [24] B. Ramalingam, R. E. Mohan, S. Pookkuttath, B. F. Gómez, C. S. C. Sairam Borusu, T. Wee Teng, and Y. K. Tamilselvam, "Remote insects trap monitoring system using deep learning framework and iot," *Sensors*, vol. 20, no. 18, p. 5280, 2020.
- [25] Y. Ginoris, A. Amaral, A. Nicolau, M. Coelho, and E. Ferreira, "Recognition of protozoa and metazoa using image analysis tools, discriminant analysis, neural networks and decision trees," *Analytica Chimica Acta*, vol. 595, no. 1-2, pp. 160–169, 2007.
- [26] B. P. Núñez Alverca, "Clasificación automática de parásitos de reptiles mediante redes neuronales convolucionales: tecnologías de internet." B.S. thesis, Quito: EPN, 2022., 2022.
- [27] M. Paluszek and S. Thomas, *MATLAB machine learning*. Apress, 2016.
- [28] S. Sharma, S.-L. Peng, J. Agrawal, R. K. Shukla, and D.-N. Le, *Data, Engineering and Applications: Select Proceedings of IDEA 2021*. Springer Nature, 2022, vol. 907.

5 ANEXOS

ANEXO A. Instrucciones para ejecutar el código fuente

ANEXO B. Código principal Principal_parasitos.m

ANEXO A: Instrucciones para ejecutar el código fuente

URL del repositorio de datos

Por favor, verifique el acceso a este repositorio. El repositorio incluye el conjunto de datos para emplear con el script y la tabla de coordenadas de segmentación del parásito empleado. Además, se incluye otros scripts empleados en la investigación previa de redes neuronales desarrollada en el entorno de Python.

Dataset

¿Cómo empezar?

Este documento describe el uso del conjunto de datos y los comandos empleados en Matlab para reproducir los resultados del manuscrito titulado "Clasificación de imágenes de parásitos de reptiles mediante técnicas de Visión por Computador y Aprendizaje Automático". Las imágenes y videos tomados por el microscopio se encuentran en el directorio `./dataSETIS00/BASE1` (en adelante dataset BASE1). Las mismas imágenes tomadas por el microscopio y los fotogramas extraídos de los videos usando el editor de video Sony Vegas Pro 11 se pueden encontrar en el directorio `./dataSET1500/BASE2` (en adelante dataset BASE2). Este último conjunto de datos es el necesario para ejecutar el script.

Para la correcta ejecución de los scripts, siga las instrucciones a continuación:

1. Descargar la carpeta llamada "Images" y la carpeta "NP" del dataset BASE2. Asegúrese que exista un total de 3616 y 1744 imágenes respectivamente.
2. Agrupar las imágenes de Images en 6 carpetas de acuerdo a la etiqueta del parásito correspondiente. En el presente trabajo se asignó el nombre de BASE3, donde se encuentran las subcarpetas de las 6 categorías. Verificar que la distri-

bución de imágenes corresponde con la Tabla 2.1.

3. Se utilizó la versión de MATLAB R2022A, instalando "Computer Vision System Toolbox". Para ingresar a la aplicación "Classification Learner", se puede realizar mediante comando (ver ANEXO B) o en la pestaña de "APPS" en el bloque de Machine Learning.
4. Para manejar la función `imcrop`, `imsharpen`, `graythresh`, `imbinarize`, se debe añadir "Image Processing Toolbox". Para visualizar la gráfica de T-SNE con `tsne`, se debe incluir el paquete de "Statistics and Machine Learning Toolbox".
5. Para trabajar con BASE3, se debe modificar la dirección ingresada en "imagedatastore" en el código, según la afinidad del lector.
6. Para realizar la etapa de preprocesamiento, se debe crear previamente la carpeta donde se va a guardar las imágenes. Para el ejemplo se nombró la carpeta "Pre-procesado" y las subcarpetas respectivas. Además, se añade en este directorio la carpeta NP.
7. Hay varias variables en algunas funciones que se pueden cambiar según la prueba que desee realizar:
 - ❑ `imsharpen`: las variables `Radius` y `Amount`, para resaltar la máscara de enfoque y los bordes
 - ❑ `splitEachLabel`: La distribución de los conjuntos de entrenamiento y prueba en valores porcentuales.
 - ❑ `bagOfFeatures`: las variables de `Vocabulary`, `StrongestFeatures` y `PointSelection` para las características de la bolsa de palabras visuales.

Después de ejecutar el script mencionado anteriormente, se pueden exportar las variables de la bolsa de palabras visuales, y los bloques de entrenamiento y test codificados para realizar pruebas posteriores y enfocarse en la aplicación de Classification Learner. La matriz de confusión y las curvas ROC se muestran en la aplicación mencionada.

ANEXO B: Principal_parasitos.m

```
1  % SISTEMA DE CLASIFICACION DE PARASITOS EN HECES DE REPTILES ...
    % BASADOS EN VISION COMPUTACIONAL.
2  % Guillermo Yanascual
3  %Programa principal
4  clc
5  close all
6  clear all
7
8  %% Paso 1: configurar conjuntos de categorias de imagenes
9  % CREACION DEL ALMACENAMIENTO DE DATOS (datastore) DE LAS ...
    % IMAGENES ORIGINALES.
10 imds1 = imageDatastore('.\Base 3\', ...
    % "FileExtensions",[".jpg",".png"], ...
11     'IncludeSubfolders',true,'LabelSource','foldernames');
12
13 %% Paso 2: Preparar imagenes para preprocesar imagenes
14
15 % Se remueve direccion de las imagenes, conservando el nombre
16 str = imds1.Files;
17 match = wildcardPattern + "\"; %patr n perezoso
18 Files2search = erase(str,match);
19
20 % Coordenadas donde se ubica el parasito en la imagen
21 T = ...
    % readtable('baseImagesCORDSTOTAL.csv','PreserveVariableNames',true);
22 info = imds1.Labels;
23 % Reordenamiento de las coordenadas de vector numerico de 4 ...
    % elementos en la forma [xmin ymin width height]
24 T2 = movevars(T,[6,4,7,5 ],'after',3);
25
26 %Verificar que el nombre de la imagen coincida con el ...
    % rectangulo de
27 %coordenadas
28 filename = T2.filename;
```

```

29 [Lia,Locb] = ismember(Files2search,filename);
30
31 %% Crear nueva base de datos con las imagenes preprocesadas
32
33 for k = 1:length(Locb) %3616 imagenes (6 categorias)
34     img_ori1 = imds1.readimage(Locb(k));
35     %redimensionar imagenes al tamaño de 512x512
36     imgReSz = imresize(img_ori1, [512 512]);
37
38     %Recortar imagen redimensionada con las coordenadas cargadas
39     imgReSz1 = imcrop(imgReSz,table2array(T2(Locb(k),[4:7])));
40
41     % contraste para mejorar nitidez
42     imgReSz1 = imsharpen(imgReSz1, 'Radius',2, 'Amount',1);
43
44     %Binarizacion mediante OTSU
45     %imgReSz1 = rgb2gray(imgReSz1);
46     level = graythresh(imgReSz1);
47     BW = imbinarize(imgReSz1,level);
48     imgReSz1 = im2uint8(BW);
49
50     % CONFIGURACIONES PARA ALMACENAR LA IMAGEN REMOVIDA EL FONDO ...
51     EN SU
52     % RESPECTIVA CARPETA
53     if info(k)=='blas'
54         % Se especifica la direccion a donde va la imagen
55         ruta = ...
56             strcat('C:\Tesis\Preprocesado\blas\',Files2search(k));
57         % Se escribe la imagen en la direccion indicada
58         imwrite(imgReSz1,string(ruta));
59     end
60     if info(k)=='ophi'
61         % Se especifica la direccion a donde va la imagen
62         ruta = strcat('\Preprocesado\ophi\',Files2search(k));
63         % Se escribe la imagen en la direccion indicada
64         imwrite(imgReSz1,string(ruta))
65     end
66     if info(k)=='oxi'

```

```

65         % Se especifica la direccion a donde va la imagen
66         ruta = strcat('\Preprocesado\oxi\',Files2search(k));
67         % Se escribe la imagen en la direccion indicada
68         imwrite(imgReSz1,string(ruta))
69     end
70     if info(k)=='rhyti'
71         % Se especifica la direccion a donde va la imagen
72         ruta = strcat('\Preprocesado\rhyti\',Files2search(k));
73         % Se escribe la imagen en la direccion indicada
74         imwrite(imgReSz1,string(ruta))
75     end
76     if info(k)=='strong'
77         % Se especifica la direccion a donde va la imagen
78         ruta = strcat('\Preprocesado\strong\',Files2search(k));
79         % Se escribe la imagen en la direccion indicada
80         imwrite(imgReSz1,string(ruta))
81     end
82     if info(k)=='tae'
83         % Se especifica la direccion a donde va la imagen
84         ruta = strcat('\Preprocesado\tae\',Files2search(k));
85         % Se escribe la imagen en la direccion indicada
86         imwrite(imgReSz1,string(ruta))
87     end
88 end
89
90 %% Paso 3: configurar conjuntos de categorias de imagenes
91 % ALMACENAMIENTO DE DATOS DE LAS IM GENES PREPROCESADAS.
92 imds2 = imageDatastore('\Preprocesado\', ...
93     "FileExtensions",[".jpg",".png"], ...
94     'IncludeSubfolders',true,'LabelSource','foldernames');
95 %Conjuntos en datos de entrenamiento y validacion. 80 % de las ...
96     imagenes de cada conjunto para entrenamiento
97 % El resto, el 20 %, para los datos de validacion. Aleatorice ...
98     la division
99 [trainingSet, testSet] = splitEachLabel(imds2, 0.8, 'randomize');

```



```

100 %% SE LLAMA A LAS FUNCIONES ELABORADAS PARA APLICAR BAG OF ...
    VISUAL WORDS
101
102 % Objeto de vocabulario visual con el bloque de entrenamiento
103 bag = bagOfFeatures (trainingSet, ...
104     'VocabularySize', 500, 'StrongestFeatures', 0.8, ...
105     'PointSelection', 'Detector');
106 % Representar cada imagen de acuerdo a la bolsa BoVW
107 featureVector = encode(bag, trainingSet); %colocar el objeto ...
    de entrenamiento
108 featuretest = encode(bag, testSet); %colocar el objeto de test
109 %ETIQUETAS
110 labeltrain = trainingSet.Labels;
111 labeltest = testSet.Labels;
112
113 %% Paso 4: entrenar un clasificador de imagenes con una bolsa ...
    de palabras visuales
114 % Se utilizar el Aprendizaje de clasificacion "Classification ...
    Learner"
115 classificationLearner
116
117
118 %% Grafica t-SNE
119 % Se obtiene una matriz de dos dimensiones del argumento que es ...
    de alta dimension
120 featureVector_normalized = zscore(featureVector);
121 Y5 = tsne(featureVector_normalized);
122
123 figure
124 % Se grafica el diagrama de dispersion con los valores de la ...
    matriz de dos dimensiones
125 gscatter(Y5(:,1),Y5(:,2),labeltrain,'rkgcm','.',8)
126 grid on
127 title('Grafica t-SNE')

```