

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA



FACULTAD DE CIENCIAS

**Clasificación en imágenes del borrego cimarrón (*Ovis canadensis*)
mediante el uso de características de aprendizaje profundo**

TESIS PROFESIONAL

QUE PARA OBTENER EL TÍTULO DE
LICENCIADO EN CIENCIAS COMPUTACIONALES

PRESENTA:

VARGAS FELIPE MANUEL DE JESÚS

Ensenada, Baja California, México, Octubre de 2020

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

FACULTAD DE CIENCIAS

**Clasificación en imágenes del borrego cimarrón (*Ovis canadensis*)
mediante el uso de características de aprendizaje profundo**

TESIS PROFESIONAL

PRESENTA:

VARGAS FELIPE MANUEL DE JESÚS

Aprobada por:



Dr. Luis Miguel Pellegrin Zazueta

Presidente del Jurado



Dr. Everardo Gutiérrez López

Secretario



Dr. Omar Álvarez Xochihua

1er. Vocal

Agradecimientos

Primeramente, agradezco a mi familia por todo su apoyo, por enseñarme a dirigirme ante la vida con dignidad, respeto y esmero. Los amo profundamente, y les agradezco todo el inmenso apoyo, manifestado de muchísimas maneras, que me han brindado durante todo este tiempo, así como la paciencia que me han tenido.

De igual manera, agradezco a mis mejores amigos por siempre estar ahí para mí y brindarme todo su apoyo en todo momento, dándome las palabras necesarias siempre que tenía dudas. Los amo.

Quiero agradecer a los amigos que hice durante mi vida universitaria. Especialmente a mis compañeros de carrera, con quienes pasé muy buenos momentos, siempre había apoyo ante cualquier duda, gracias por estos buenos cuatro años.

Al Dr. Luis Miguel Pellegrin Zazueta, por aceptar ser mi director de tesis, por darme la oportunidad de colaborar en este proyecto, por mostrar ese deseo de mejora continua, por la motivación a querer superarme y no conformarme, muchas gracias.

Tanto al Dr. Everardo Gutiérrez López como al Dr. Omar Álvarez Xochihua les agradezco por aceptar ser mis sinodales y por todos sus comentarios realizadas durante la etapa de revisión de la tesis.

Además, el Dr. Everardo Gutiérrez López fue mi tutor a lo largo de la licenciatura, al cual, quiero agradecerle por siempre estar al pendiente tanto de mí como de mis compañeros, ser un guía durante la toma de decisiones académicas, apoyarme en mi

vida estudiantil y ser un gran profesor, muchas gracias.

Agradezco tanto a los profesores como a las profesoras que tuve durante toda la carrera, por su profesionalismo y su deseo de que los estudiantes mejoren, todo esto no hubiera sido posible sin ustedes, muchas gracias.

De la misma manera, agradezco a la Universidad Autónoma de Baja California, y a la Facultad de Ciencias, por darme la oportunidad de estudiar esta licenciatura.

Resumen

Resumen de la tesis de Vargas Felipe Manuel de Jesús, presentada como requisito parcial para la obtención de la Licenciatura en Ciencias Computacionales. Ensenada, Baja California, México. Octubre de 2020.

Clasificación en imágenes del borrego cimarrón (*Ovis canadensis*) mediante el uso de características de aprendizaje profundo

El monitoreo de la vida silvestre en áreas geográficas es esencial para la conservación del patrimonio biológico. Hoy en día, el estudio de especies de animales en su hábitat natural es extenuante, pero esto ha sido amortiguado con la llegada de las cámaras trampa. La información recopilada por estos dispositivos comprende una secuencia de imágenes, que se activan por medio de sensores, y permite el monitoreo de manera simultánea de varias áreas geográficas sin perturbar la fauna. Como consecuencia de éstos avances para el estudio de especies de animales, se genera una gran cantidad de imágenes que aún deben clasificarse manualmente, convirtiéndolo nuevamente en un proceso costoso y agotador.

Como solución para realizar esta tarea, se propone un método computacional que se encargue de esto. El cual, se encargará de que de manera automática se etiquetaran imágenes capturadas con cámaras trampa como una herramienta de apoyo para el análisis del comportamiento animal.

Este trabajo de tesis se enfoca en el análisis del *Ovis canadensis* más conocido como borrego cimarrón, una especie que habita en el noroeste de México, Estados Unidos y Canadá. La importancia de esta especie radica en que es una especie emblemática de gran valor histórico, cultural y social.

Para realizar el método computacional previamente mencionado, se adaptó una metodología basada en redes neuronales convolucionales para la caracterización, mientras que, la clasificación de las imágenes que representan al borrego cimarrón se lleva a cabo por algoritmos de aprendizaje automático. Para la realización de éste método, se construyeron tres diferentes bancos de imágenes, uno está compuesto de imágenes sintéticas recopiladas de la Web, otro está compuesto de imágenes reales captadas por las cámaras trampa, y el último es un banco de imágenes compuesto por partes de los dos anteriores.

El método está basado en un modelo generado apartir del entrenamiento de un conjunto de imágenes sintéticas recopiladas de la web y luego es evaluado en imágenes realistas capturadas por cámaras trampa en el hábitat natural de la especie. El trabajo explora diferentes escenarios de la tarea del reconocimiento del borrego cimarrón en imágenes captadas por cámaras trampa. Esto es así para saber si es factible simplificar dicha tarea. Se lograron clasificaciones de 99 % y 88 % según la complejidad del escenario de evaluación.

Resumen aprobado por:



Dr. Luis Miguel Pellegrin Zazueta

Director de Tesis

Palabras clave: Borrego cimarrón, cámaras trampa, aprendizaje automático, aprendizaje profundo, clasificación de imágenes, clasificación de animales

Índice general

| | |
|--|----------|
| Agradecimientos | II |
| Resumen | IV |
| Índice de figuras | IX |
| Índice de tablas | XI |
| 1. Introducción | 1 |
| 1.1. Antecedentes y Justificación | 1 |
| 1.2. Hipótesis | 4 |
| 1.3. Objetivo general | 4 |
| 1.4. Objetivos específicos | 5 |
| 1.5. Metas | 5 |
| 2. Marco teórico | 7 |
| 2.1. Introducción al aprendizaje automático | 7 |
| 2.1.1. Aprendizaje supervisado | 9 |
| 2.1.2. Métricas para evaluar un modelo de aprendizaje automático . | 11 |
| 2.1.3. Algoritmos de aprendizaje automático | 13 |
| 2.2. Extracción de características | 18 |
| | VI |

| | | |
|-----------|--|-----------|
| 2.2.1. | Extracción de características tradicional | 20 |
| 2.2.1.1. | <i>Histogram of Oriented Gradient</i> (HOG) | 20 |
| 2.2.1.2. | <i>Scale Invariant Feature Transform</i> (SIFT) | 21 |
| 2.2.1.3. | <i>Local Binary Patterns</i> (LBP) | 22 |
| 2.2.2. | Extracción de características basada en aprendizaje profundo . | 24 |
| 2.2.3. | Redes residuales | 27 |
| 3. | Trabajos relacionados | 30 |
| 3.1. | Trabajos tradicionales | 30 |
| 3.2. | Trabajos que usan aprendizaje automático | 32 |
| 3.2.1. | Trabajos de categorización | 33 |
| 3.2.2. | Trabajos de identificación individual | 34 |
| 3.3. | Recapitulación | 35 |
| 4. | Metodología | 37 |
| 4.1. | Pasos a seguir | 38 |
| 4.2. | Construcción de las bases de datos | 39 |
| 4.2.1. | Base de datos sintética | 40 |
| 4.2.1.1. | Credenciales de las bases de datos | 41 |
| 4.2.1.2. | Adquisición y selección | 41 |
| 4.2.2. | Base de datos real | 43 |
| 4.3. | Extracción de características a través de aprendizaje profundo | 47 |
| 4.4. | Escenarios de clasificación para reconocimiento del borrego cimarrón . | 48 |
| 5. | Experimentos y Resultados | 51 |
| 5.1. | Algoritmos de aprendizaje automático | 51 |
| 5.2. | Particiones de datos utilizadas en los experimentos | 53 |

| | |
|---|-----------|
| 5.3. Resultados | 56 |
| 5.3.1. Base de datos sintética | 57 |
| 5.3.2. Base de datos real | 59 |
| 5.3.3. Base de datos cruzada | 62 |
| 5.4. Discusión | 66 |
| 6. Conclusiones | 69 |
| 6.1. Trabajo a futuro | 71 |
| A. Permiso del Valle de los Cirios | 72 |
| Bibliografía | 73 |

Índice de figuras

| | |
|--|----|
| 2.1. Proceso de aprendizaje automático para separar correo <i>spam</i> | 8 |
| 2.2. Ejemplo de clasificación con SVM | 14 |
| 2.3. Ejemplo de cómo KNN realiza la clasificación. | 15 |
| 2.4. Ejemplo de una red neuronal artificial. | 15 |
| 2.5. Ejemplo de un árbol de decisión. | 17 |
| 2.6. Gradientes e histogramas de una imagen | 20 |
| 2.7. Ejemplo de SIFT | 22 |
| 2.8. Operación de la unidad de textura de LBP (Shan et al., 2009). | 23 |
| 2.9. Obtención del histograma que funciona como descriptor de textura en LBP (Shan et al., 2009). | 23 |
| 2.10. Arquitectura de una red neuronal convolucional con dos capas convo- lucionales, dos capas <i>pooling</i> y una capa totalmente conectada. | 24 |
| 2.11. Ejemplo de <i>kernel</i> para la detección de aristas (figura extraída de Wikipedia). | 26 |
| 2.12. Ejemplo de las operaciones <i>max pooling</i> y <i>average pooling</i> en una ma- triz 4×4. | 27 |
| 2.13. Conexión de acceso directo (He et al., 2016). | 28 |
| 4.1. Proceso de la metodología del trabajo. | 38 |

| | |
|--|----|
| 4.2. Imágenes de la base de datos sintética | 43 |
| 4.3. Localización del Área de Protección de Flora y Fauna Valle de los Cirios | 44 |
| 4.4. Imágenes de la base de datos real | 46 |
| 4.5. Imágenes fallidas de la base de datos real | 47 |
| 4.6. Proceso de extracción de características a partir de <i>ResNet-50</i> | 48 |
| 5.1. Matrices de confusión de tres clases en la base de datos sintética . . . | 58 |
| 5.2. Matrices de confusión de múltiples clases en la base de datos sintética | 59 |
| 5.3. Matrices de confusión de tres clases en la base de datos real | 60 |
| 5.4. Matrices de confusión de múltiples clases en la base de datos real . . | 61 |
| 5.5. Matrices de confusión de tres clases en la base de datos cruzada . . . | 64 |
| 5.6. Matrices de confusión de múltiples clases en la base de datos cruzada | 65 |
| 5.7. Comparación de los resultados entre modelos | 66 |

Índice de tablas

| | |
|---|----|
| 3.1. Comparación de los trabajos relacionados | 33 |
| 4.1. Base de datos sintética. | 42 |
| 4.2. Cantidad de imágenes de las cámaras trampa | 45 |
| 5.1. Número de imágenes usadas para el entrenamiento y prueba por cada animal en la base de datos sintética. | 54 |
| 5.2. Número de imágenes usadas para el entrenamiento y prueba por cada animal en la base de datos real. | 55 |
| 5.3. Número de imágenes usadas para el entrenamiento y prueba por cada animal en la base de datos cruzada. | 56 |
| 5.4. Resultados obtenidos por los algoritmos de aprendizaje automático en los escenarios de clasificación binaria en la base de datos sintética. . . | 57 |
| 5.5. Resultados obtenidos por los algoritmos de aprendizaje automático en la clasificación de tres clases en la base de datos sintética. | 57 |
| 5.6. Resultados obtenidos por los algoritmos de aprendizaje automático en la clasificación de clases múltiples en la base de datos sintética. | 58 |
| 5.7. Resultados obtenidos por los algoritmos de aprendizaje automático en los escenarios de clasificación binaria en la base de datos real. | 59 |

| | |
|--|----|
| 5.8. Resultados obtenidos por los algoritmos de aprendizaje automático en la clasificación de tres clases en la base de datos real. | 60 |
| 5.9. Resultados obtenidos por los algoritmos de aprendizaje automático en la clasificación de clases múltiples en la base de datos real. | 61 |
| 5.10. Resultados obtenidos por los algoritmos de aprendizaje automático en los escenarios de clasificación binaria en la base de datos cruzada. . . | 62 |
| 5.11. Resultados obtenidos por los algoritmos de aprendizaje automático en la clasificación de tres clases en la base de datos cruzada. | 63 |
| 5.12. Resultados obtenidos por los algoritmos de aprendizaje automático en la clasificación de clases múltiples en la base de datos cruzada. | 64 |

Capítulo 1

Introducción

1.1. Antecedentes y Justificación

Las tareas de identificación y rastreo de animales son procesos necesarios que se llevan a cabo para reconocer con exactitud especies individualmente, ya sean de uso doméstico, como el ganado, o aquellas en su hábitat natural. Inicialmente en la ganadería, para llevar a cabo la identificación (por ejemplo, reses, cerdos, ovejas, caballos, etc.) se comenzó a utilizar un sistema de rastreo mediante marcas o tatuajes hechos de hierro candente (Ward, 1958). Con el paso del tiempo surgieron nuevos dispositivos como el RFID (por sus siglas en inglés, *radio frequency identification*) que eran puestos como ‘aretes’ en el animal y usados como etiquetas de identificación e incorporando sistemas de rastreo. El sistema de marcas empleado en la ganadería ha sido una forma de controlar la posible propagación de enfermedades (Reyes-Moncayo et al., 2011), y al mismo tiempo establecer una confianza para futuros consumidores del ganado (Ireland, 2005). Sin embargo, las prácticas de etiquetado en el ganado han cambiado, al tratarse de métodos invasivos, evidentemente pueden dañar el tejido del animal y por lo tanto causar dolor. Incluso, a largo plazo las

prácticas de dichos métodos de identificación pueden afectar la apariencia del animal, su interacción social, así como otros comportamientos que debilitan su estado natural de supervivencia (Lay et al., 1992).

En años recientes, han surgido nuevos métodos no invasivos para la identificación de animales, en donde resalta el uso de la biometría (Kühl y Burghardt, 2013). La biometría hace referencia al estudio para el reconocimiento inequívoco de seres, y está basada en la toma de medidas estandarizadas que pueden ser conductuales o físicas. Por ejemplo, tres métodos biométricos físicos ampliamente usados para la identificación de animales son el estampado de hocico (*muzzle print*), exploración de retina (*retina scan*), y reconocimiento de iris (*iris recognition*). No obstante, aunque los métodos basados en biometría proveen una identificación eficaz y a bajo costo pueden no ser aplicables a todo tipo de animales, tal es el caso de los animales de vida silvestre.

En cambio, para llevar a cabo un monitoreo de la biodiversidad de la vida silvestre se han empleado dispositivos digitales, conocidos como cámaras trampa. Para estos casos, expertos en biología de la conservación colocan estos dispositivos de grabación equipados de sensores de movimiento que realizan tomas automáticamente. Los lugares idóneos para colocar los dispositivos están definidos por bebederos de agua o rutas de caza de los animales determinados por expertos. Después de un periodo de tiempo, se recogen las imágenes capturadas por los dispositivos, y se procede a un reconocimiento, donde se define si existe o no un animal en la escena; después viene la clasificación de los animales, que consiste en categorizarlos por diferentes especies. Tradicionalmente, las tareas de reconocimiento y clasificación de animales se realizan de manera manual por expertos, siendo estas labores arduas y extenuantes debido a las características de las tomas que dificultan los procesos, por ejemplo imágenes

producidas con oclusiones, baja resolución, cambios en iluminación, perspectiva, etc.

La identificación de animales en su hábitat natural es una tarea de gran importancia para el control de mortandad de diversas especies, así como su conservación en ambientes naturales. Entre los casos exitosos de identificación automática de animales se encuentra la del tigre de Bengala (Sharma et al., 2005), cuya identificación se logró a partir del reconocimiento de los patrones de sus rayas empleando solo imágenes. En términos computacionales, las tareas de reconocimiento y clasificación están fuertemente relacionadas. Lo que permite el uso de una o ambas para la realización de una tarea de este estilo.

En el presente proyecto se introduce un método para facilitar la clasificación automática del borrego cimarrón (*Ovis canadensis*) a través de imágenes obtenidas por cámaras trampa (*camera trap*), instaladas en una localidad rural ubicada en el Área de Protección de Flora y Fauna Valle de los Cirios. La clasificación se realiza de manera automática e incluye el reconocimiento de animales que sean captados por las cámaras. Para tales efectos se han aprovechado los avances que se han logrado a través del paradigma de computación conocido como aprendizaje profundo.

Bajo el paradigma de aprendizaje profundo se trabaja en el diseño de arquitecturas de redes neuronales artificiales que permiten modelar abstracciones de los datos de entrada, con el objetivo de discriminarlos (Bengio et al., 2013). A diferencia del aprendizaje tradicional, las arquitecturas diseñadas mediante aprendizaje profundo reciben los datos sin preprocesar y realizan tareas de extracción de características automáticamente. Además, los modelos entrenados bajo el paradigma de aprendizaje profundo pueden ser utilizados en un dominio distinto, mostrando capacidad para transferir el conocimiento adquirido (o por su denominación en inglés *transfer learning*). Para llevar a cabo lo anterior, la imagen es procesada por el modelo aprendido

(representado por la arquitectura de la red), y activaciones arrojadas por las capas de la red son usadas para representar a la imagen por medio de un vector (Krizhevsky et al., 2012). Lo anterior plantea un nuevo escenario en donde los modelos entrenados pueden representar datos para diferentes tareas que pueden o no, tener relación con la tarea original para la cual fueron creados.

En los últimos años se han presentado importantes contribuciones que han sido divisorias en diferentes tareas relacionadas con imágenes usando el paradigma de aprendizaje profundo. Esto ha permitido, hasta cierto punto, automáticamente organizar y descubrir información a partir del contenido visual de las imágenes, ya sea por medio de su clasificación (Krizhevsky et al., 2012), el reconocimiento de objetos (Redmon et al., 2016), o la generación automática descriptiva (Karpathy y Fei-Fei, 2015). Por tales motivos, se plantea al paradigma de aprendizaje profundo como un buen punto de partida para realizar la investigación propuesta.

1.2. Hipótesis

La variabilidad en las imágenes capturadas por medio de cámaras trampa puede ser modelada a través de aprendizaje profundo para la clasificación automática del borrego cimarrón (*Ovis canadensis*).

1.3. Objetivo general

Proponer y desarrollar un método de clasificación automática del borrego cimarrón a partir de imágenes digitales extraídas de fuentes de monitoreo.

1.4. Objetivos específicos

1. Construir un banco de imágenes del borrego cimarrón obtenidas a partir de la web para el entrenamiento de modelos de clasificación. Y un segundo banco de imágenes reales provenientes de cámaras trampas para evaluación.
2. Proponer un método basado en aprendizaje profundo para la clasificación del borrego cimarrón, donde, al tener el modelo de inferencia no sea necesaria la intervención de expertos para la clasificación.
3. Realizar una evaluación del método propuesto para la clasificación de imágenes del borrego cimarrón.

1.5. Metas

1. Contribuciones científicas.
 - Un método de clasificación experimental que pueda emplearse para el reconocimiento automático del borrego cimarrón a partir de imágenes capturadas por cámaras para su uso en la conservación y monitoreo.
2. Impacto estatal y regional.
 - La presente propuesta tiene como objetivo, fortalecer por medio de ciencia aplicada a la Estrategia Estatal para la Conservación y el Manejo Sustentable del borrego cimarrón en Baja California publicada en el 2012 por la Secretaría de Protección al Medio (de Protección al Ambiente, 2012), con la introducción de métodos automatizados para el reconocimiento y la clasificación de especies animales.

- Por otro lado, la problemática a resolver ha surgido de la iniciativa del proyecto “Propuesta de modelo de aprovechamiento sustentable del borrego cimarrón (*Ovis canadensis*) a través de la observación en Baja California” (Guevara, 2017), dirigida por el Dr. Aldo Antonio Guevara Carrizales. Es una iniciativa para desarrollar una propuesta que tiene como objetivo el promover la conservación de esta especie y su hábitat, con la esperanza de que haya un decremento en la competencia, el furtivismo y la eventual sustitución del hábitat de la especie.
- Así como el desarrollo de prácticas que fomenten la generación de recursos económicos que favorezcan a los dueños del hábitat donde ostenta el borrego cimarrón, así como el incremento en la cultura de sustentabilidad ambiental para las comunidades rurales de Baja California. En el apéndice *A*, se anexa la carta donde se acepta el proyecto por parte del Área de Protección de Flora y Fauna Valle de los Cirios.

Capítulo 2

Marco teórico

El marco teórico que se desarrolla a continuación, revisa los conceptos necesarios para el entendimiento del desarrollo de este trabajo. Primero, en la sección 2.1 se habla sobre el aprendizaje automático, y se revisan algunos algoritmos que se utilizan en este campo de estudio, así mismo las métricas que tradicionalmente son usadas para medir el rendimiento de los modelos obtenidos a partir del entrenamiento de los algoritmos. Posteriormente, en la sección 2.2, se explica la tarea de extracción de características, su importancia en el reconocimiento de imágenes, y se mencionan algunos métodos tradicionales. Por último, se revisa la extracción de características por medio de aprendizaje profundo, enfoque que fue utilizado en este trabajo.

2.1. Introducción al aprendizaje automático

Para resolver un problema computacional, se necesita definir un algoritmo. Un algoritmo es una secuencia de instrucciones que se encarga de transformar la entrada en la salida. Un ejemplo de esto, sería un algoritmo que es capaz de ordenar números. La entrada es una lista de números, y la salida es la lista de números ordenados.

Sin embargo, para algunas tareas, aún no se tiene definido un algoritmo capaz de realizarlas. Por ejemplo, distinguir entre correos electrónicos *spam* y legítimos. Se sabe que la entrada es un correo electrónico, donde en un caso simple es sólo una cadena de texto. Se sabe que la salida es un sí o un no indicando si el correo electrónico es o no *spam*. Pero aún no se sabe cómo transformar la entrada a la salida para esta tarea. El aprendizaje automático puede solucionar este problema. Para ello, se compilan correos electrónicos de ejemplo, algunos de los cuales se sabe que son *spam*, y lo que se quiere es "aprender" lo que los caracteriza como *spam*. En otras palabras, se busca extraer automáticamente el algoritmo para esta tarea (Alpaydin, 2010). En la Figura 2.1 se puede observar un ejemplo de esto.

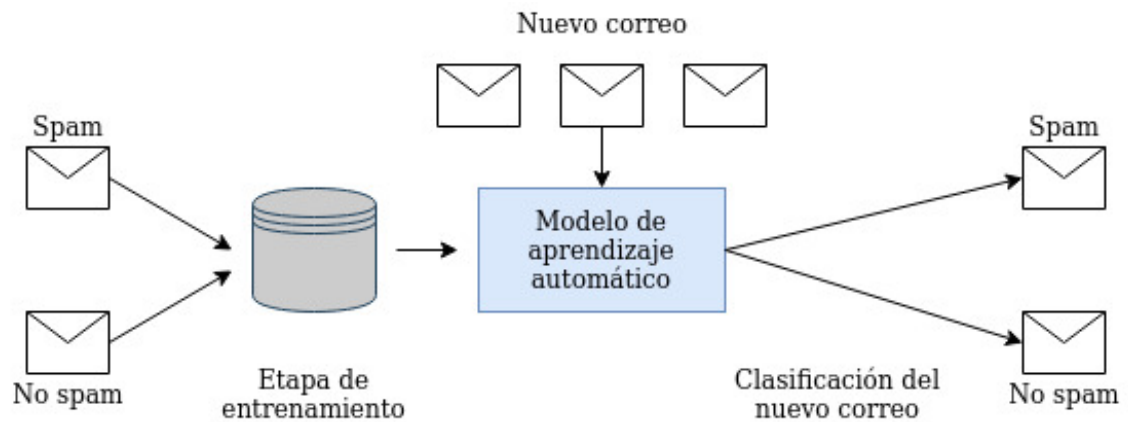


Figura 2.1. Proceso de aprendizaje automático para separar correo *spam*.

Se define aprendizaje automático como un conjunto de métodos que automáticamente pueden detectar patrones en los datos, y luego, usar los datos descubiertos (normalmente llamado conjunto de entrenamiento) para predecir datos nunca antes vistos (normalmente llamado conjunto de prueba), o para ejecutar otro tipo de decisiones (Murphy, 2012). Esto es una alternativa bastante atractiva porque facilita la creación de métodos computacionales para la solución de problemas que no cuentan con un algoritmo que los resuelva. En la última década, el uso del aprendizaje au-

tomático se ha extendido rápidamente: se utiliza en la búsqueda web, filtros de correo no deseado, sistemas de recomendaciones, colocación de anuncios, calificación crediticia, detección de fraude, comercio de acciones, diseño de medicamentos y muchas otras aplicaciones (Domingos, 2012).

El aprendizaje automático utiliza la teoría de la estadística en la construcción de modelos matemáticos, donde la tarea central es hacer inferencia a partir de una muestra de datos. Para esto, hay dos pasos principales. Primero, el entrenamiento, se necesitan algoritmos eficientes para resolver el problema de optimización, así como para almacenar y procesar la gran cantidad de datos que generalmente se tienen. Segundo, una vez que se aprende un modelo, su representación y solución algorítmica para su inferencia también debe ser eficiente (Alpaydin, 2010).

2.1.1. Aprendizaje supervisado

El aprendizaje automático usualmente se divide en tres enfoques principales: supervisado, no supervisado y por refuerzo. En este caso sólo se enfocará en el primero ya que es el que se emplea en este trabajo.

El objetivo del aprendizaje supervisado es el de encontrar una función que, dada una entrada, proporcione la salida esperada. Formalmente la tarea se define como un conjunto etiquetado de pares $\mathcal{S} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$. Donde \mathcal{S} es el conjunto de entrenamiento, N es el número de ejemplos de entrenamiento, \mathbf{x}_i es un ejemplo de entrenamiento representado como un vector de dimensionalidad D , e y_i es la clase a la que pertenece \mathbf{x}_i .

En la configuración más simple, cada entrada de entrenamiento \mathbf{x}_i está formada por números llamados características o atributos. Sin embargo, \mathbf{x}_i , podría ser un objeto complejo, como una imagen, un correo electrónico, un grafo, entre otras cosas.

En el enfoque supervisado, la forma de la salida y_i es una variable categórica o nominal de un conjunto finito, $y_i \in \{1, \dots, C\}$. Retomando el ejemplo de la sección anterior, *spam* o no. Donde C es el número de clases. Si $C = 2$, se le llama clasificación binaria (en dicho caso, se asume que $y \in \{0, 1\}$); si $C > 2$, se le llama clasificación de múltiples clases.

Una forma de formalizar el problema es como función de aproximación. Suponemos que $y = f(\mathbf{x})$ para alguna función desconocida f , y el objetivo del aprendizaje es estimar f dado un conjunto de entrenamiento etiquetado, y luego hacer predicciones usando $\hat{y} = \hat{f}(\mathbf{x})$, donde se usa el símbolo $\hat{\cdot}$ para denotar una estimación. La predicción se lleva a cabo sobre entradas nuevas, es decir, las que no se han visto antes, a esto se llama generalización, ya que predecir una respuesta del conjunto de entrenamiento es fácil ya que se puede sólo buscar la respuesta (Murphy, 2012).

Construir un modelo de clasificación implica llevar a cabo experimentación. Donde es necesario construir varios modelos para los que no necesariamente se sabe cuál de ellos será el "mejor". Para garantizar el mejor resultado posible el experimento debe diseñarse adecuadamente y realizar una evaluación del modelo, lo cual depende del objetivo del proceso de modelado definido al inicio del proyecto. Otros problemas son, por ejemplo, la capacidad de explicar cómo se hizo una predicción, también pueden ser importantes y deben tenerse en cuenta al evaluar los modelos generados. Siempre que sea posible, cuando dos o más modelos dan resultados comparables, se debe seleccionar el modelo más simple (Myatt, 2006).

El rendimiento de un modelo para la clasificación se puede medir mediante un rango de métricas de evaluación que incluyen la exactitud, tasa de positivos verdaderos y falsos, precisión y sensibilidad, métrica F, área bajo la curva (ROC) y puntaje de Brier. Cada una de estas métricas de evaluación tiene una interpretación técnica

clara que puede vincularse a casos de uso particulares. Además, existen relaciones bien definidas entre muchas de estas métricas de evaluación (Flach, 2019).

Las métricas antes mencionadas son obtenidas de un modelo cuando éste fue probado con el conjunto de datos de prueba. Hay diferentes métodos para particionar el total de datos en dos conjuntos distintos, los llamados conjuntos de entrenamiento y prueba. Algunas se muestran a continuación:

- *Hold-out* (Myatt, 2006): El conjunto de datos se divide en un cierto porcentaje para el conjunto de entrenamiento y el resto es para el de prueba, lo ideal es que el porcentaje del conjunto de entrenamiento sea mayor que el de prueba. Por ejemplo, un 25 % de los datos se asigna al conjunto de prueba y el 75 % restante se asigna al de entrenamiento.
- Validación cruzada de k pliegues (Tan et al., 2005): En este método, cada registro se usa la misma cantidad de veces para el entrenamiento y exactamente una vez para las pruebas. El enfoque es segmentar los datos en k particiones de igual tamaño. Durante cada ejecución del algoritmo de aprendizaje automático, se elige una de las particiones para la prueba, mientras que el resto se usa para entrenar exactamente una vez. Este método genera k modelos diferentes, por ende, el error total se obtiene al sumar los errores de cada uno de los modelos obtenidos.

2.1.2. Métricas para evaluar un modelo de aprendizaje automático

Dada una predicción del modelo, esta puede ser definida como: verdadero positivo (TP , por sus siglas en inglés) que son aquellas instancias positivas que se han

identificado correctamente; verdadero negativo (TN , por sus siglas en inglés) que son aquellas instancias negativas que han sido rechazadas correctamente; falso positivo (FP , por sus siglas en inglés) aquellas instancias negativas que han sido incorrectamente clasificadas como positivas; y falso negativo (FN , por sus siglas en inglés) son aquellas instancias positivas que han sido incorrectamente clasificadas como negativas. A partir de éstas definiciones existen métricas de evaluación que permiten medir el rendimiento de cada uno de los modelos generados:

- Exactitud. Proporción de las instancias clasificadas correctamente. Un inconveniente de esta métrica es que no es una buena medida del rendimiento cuando hay un gran desbalance de clases, como es el caso en este trabajo.

$$ACC = (TP + TN)/(TP + TN + FP + FN). \quad (2.1)$$

- Precisión. Proporción de las instancias que realmente pertenecen a una clase dividido entre el total de instancias clasificadas como dicha clase.

$$P = TP/(TP + FP). \quad (2.2)$$

- Sensibilidad. Proporción de instancias clasificadas en una cierta clase dividido por el total real en esa clase.

$$R = TP/(TP + FN). \quad (2.3)$$

- Métrica F. Una métrica combinada por precisión y sensibilidad. Se calcula así:

$$F1 = (2 * P * R)/(P + R). \quad (2.4)$$

Tomando en cuenta que las bases de datos pueden estar desbalanceadas, puede ser usado el promedio ponderado para las métricas de precisión y sensibilidad, luego, a partir de estas métricas ponderadas se calcula la métrica F. El promedio ponderado se calcula usando:

$$w_m = \frac{(m_{c_1} * |c_1|) + \dots + (m_{c_i} * |c_i|)}{|c_1| + \dots + |c_i|}, \quad (2.5)$$

donde m es la métrica (es decir, sensibilidad o precisión) obtenido por la clase c y $|c|$ es el número de instancias en la clase.

2.1.3. Algoritmos de aprendizaje automático

A continuación, se repasarán los algoritmos de aprendizaje automático utilizados en este trabajo.

- *Support Vector Machine* (SVM) (Cortes y Vapnik, 1995). Construye un modelo que asigna nuevos ejemplos a una categoría u otra en un espacio binario. Los ejemplos se asignan a ese mismo espacio y se predice que pertenecen a una categoría basada en el lado en el que caen. Estos lados, se generan a partir de un hiperplano óptimo. Éste último se define como la función de decisión lineal con el margen máximo entre los vectores de soporte de las dos clases, como se puede apreciar en la Figura 2.2. Para construir estos hiperplanos óptimos, solo hay que tener en cuenta una pequeña cantidad de datos de entrenamiento, los llamados vectores de soporte, que determinan este margen. Algo interesante es que éstos algoritmos pueden realizar eficientemente una clasificación no lineal utilizando lo que se llama *kernel trick*, que mapea implícitamente sus entradas en espacios de características de alta dimensión.

Una implementación eficiente de SVM es *LibLinear* (LL) propuesta en (Fan

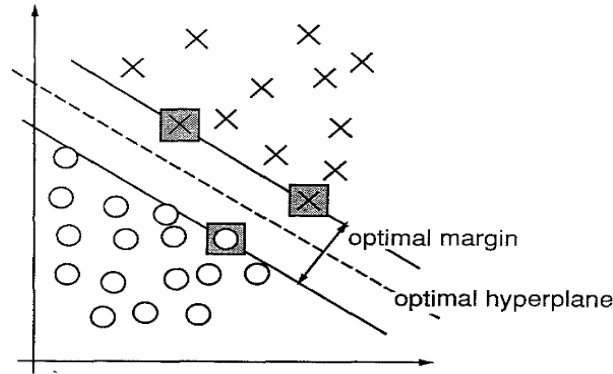


Figura 2.2. Un ejemplo de un problema separable en un espacio binario. Los vectores de soporte (las líneas que traspasan a los cuadrados grises), definen el margen de mayor separación entre las dos clases (Cortes y Vapnik, 1995).

et al., 2008). Durante el entrenamiento se resuelve el problema de optimización utilizando el método de descenso coordinado o un método Newton de región de confianza. Ofrece muy buenos tiempos, tanto en entrenamiento como en prueba.

- *K-Nearest Neighbor* (KNN) (Silverman y Jones, 1989). En este algoritmo, el modelo está representado por la propia regla de distancia (en este caso, la distancia euclidiana). El principio detrás de éste método es encontrar un número predefinido de datos de entrenamiento más cercanos en distancia al dato de prueba y predecir la clase a partir de la clase mayoritaria en las muestras encontradas. Un ejemplo se puede observar en la Figura 2.3. El número de muestras puede ser una constante definida k , o puede variar según la densidad local de puntos. La distancia euclidiana se define como:

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2} \quad (2.6)$$

donde \mathbf{p} y \mathbf{q} son las instancias puestas en comparación.

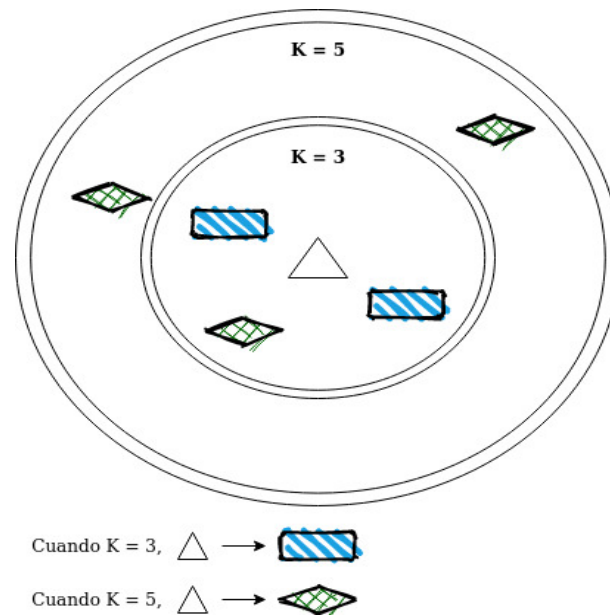


Figura 2.3. Ejemplo de cómo KNN realiza la clasificación. Cuando $K = 3$, el triángulo (dato de prueba) obtiene la categoría de rectángulo azul, y cuando $K = 5$ obtiene la categoría de rombo verde.

- Red neuronal artificial (ANN) (Rosenblatt, 1958). Está basado en las redes neuronales biológicas. Tal como se muestra en la Figura 2.4, es una red de

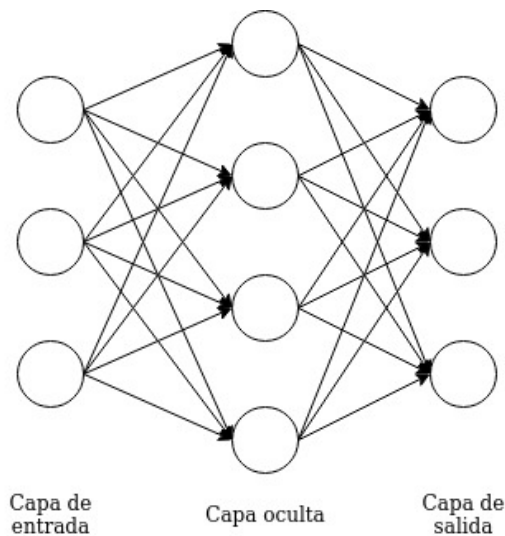


Figura 2.4. Ejemplo de una red neuronal artificial. La capa de entrada cuenta con tres nodos, la oculta con cuatro nodos, y la de salida con tres nodos.

nodos (también llamados neuronas, de ahí su nombre), que está conformada de varias capas, una de entrada, una o más ocultas, y una de salida. Cada uno de los nodos de cada capa están conectados con todos los nodos de la capa posterior, y cada una de esas conexiones tiene un peso, además, cada capa cuenta con una función de activación junto a un valor de sesgo. La forma en que se procesan los datos, es que la capa de entrada recibe el vector de entrada (características en una instancia) y va transfiriendo los datos hacía la siguiente capa por medio del producto punto. Esto se define como:

$$z = f\left(\sum_i^n w_i x_i + b\right) \quad (2.7)$$

donde f es la función de activación, \mathbf{x} es el vector de la capa actual, \mathbf{w} es el vector de los pesos de la capa actual, b es el sesgo de la capa actual y z son los datos que irán a la siguiente capa.

El objetivo del algoritmo es crear un modelo donde se establezcan relaciones entre los datos de entrenamiento y el peso de los nodos. Algo interesante del algoritmo, es que durante el período de entrenamiento lo que va aprendiendo es a minimizar el error generado al comparar los valores de los datos de entrenamiento con los valores obtenidos por el algoritmo.

- *Naïve Bayes* (NB) (Mitchell, 1997). Utiliza el teorema de Bayes para hacer varias hipótesis de un dato de prueba sobre a qué clase pertenece, esto lo hace mediante la probabilidad de que cada una de sus característica pertenezcan a cada una de las diferentes clases. Al terminar de calcular las probabilidades, selecciona la hipótesis con la mayor probabilidad y le asigna esa clase al dato de prueba. Esto lo hace leyendo el conjunto de entrenamiento. Formalmente

se puede escribir de tal manera que el dato de prueba es representado por un vector \mathbf{x} que contiene n características (variables independientes), y asigna a esta instancia las probabilidades

$$p = (C|x_1, \dots, x_n) \quad (2.8)$$

para cada una de las N clases posibles C_N .

- **Árbol de decisión (DT)** (Quinlan, 1993). Este algoritmo utiliza el conjunto de entrenamiento, para crear una estructura de datos de árbol que se puede utilizar para clasificar nuevos datos. En la Figura 2.5 se ilustra un ejemplo. Cada nodo interno contiene un criterio, cuyo resultado se usa para decidir qué rama seguir a partir de ese nodo.

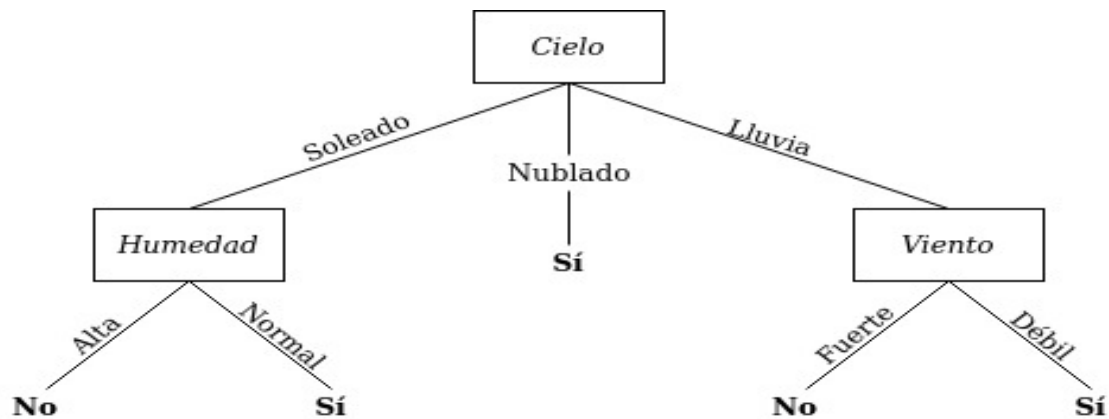


Figura 2.5. Ejemplo de un árbol de decisión. Este árbol clasifica si las condiciones climáticas son aptas o no para jugar al tenis (Mitchell, 1997).

Por otro lado, los nodos hoja indican una clase. Utiliza fórmulas basadas en la teoría de la información para evaluar qué tan bueno es un criterio; en particular, se elige el criterio que extrae la máxima cantidad de información de un conjunto de casos, dada la restricción de que solo se probará una característica. Para

evaluar la calidad de un criterio se utiliza una función de ganancia, que a su vez, utiliza la función de entropía de la información, y se muestran a continuación:

$$Entropia(S) = - \sum_{c \in S} \frac{c}{|S|} \log_2 \frac{c}{|S|} \quad (2.9)$$

donde $\frac{c}{|S|}$ es la frecuencia relativa de la clase c en el conjunto de entrenamiento S . Por otro lado, la función de ganancia se puede escribir de la siguiente manera:

$$Ganancia(S, A) = Entropia(S) - \sum_{i \in Instancias(A)} \frac{S_i}{|S|} \cdot Entropia(S_i) \quad (2.10)$$

donde A es un característica y S_i es el subconjunto de ejemplo de S que cuenta con la característica A .

2.2. Extracción de características

En las áreas del reconocimiento de patrones y el procesamiento de imágenes, la extracción de características es una forma de reducción de la dimensionalidad. El principal objetivo de la extracción de características es el de obtener la mayor cantidad de información relevante del dato original, y representar dicha información en un espacio de menor dimensionalidad. Por ejemplo, cuando los datos de entrada del algoritmo son demasiado grandes para ser procesados y se sospecha que son redundantes (muchos datos, pero poca información), los datos de entrada se transformarán en un conjunto reducido de características (también llamado vector de características). Si las características extraídas se eligen con cuidado, se espera que se extraiga la información relevante de los datos de entrada para realizar la tarea deseada utilizan-

do esta representación reducida en lugar de la entrada de tamaño completo (Kumar y Bhatia, 2014).

En el enfoque tradicional para el reconocimiento de imágenes, un extractor de características es diseñado a mano y recopila información relevante de la entrada eliminando las variaciones irrelevantes. Después de extraer características se utiliza un clasificador para elegir a qué clase corresponden las características encontradas (Hijazi et al., 2015).

Los extractores de características deben cumplir dos criterios importantes (Grauman y Leibe, 2011):

- El proceso debe ser repetible y preciso, de modo que se extraigan las mismas características de dos imágenes que muestran el mismo objeto.
- Al mismo tiempo, las características deben ser distintivas, de modo que las diferentes estructuras de imagen se puedan distinguir entre sí.

Normalmente se requiere un número suficiente de regiones de características para cubrir el objeto destino, de modo que pueda ser reconocido bajo una oclusión parcial. Esto se logra mediante el siguiente procedimiento de extracción de características (Grauman y Leibe, 2011):

1. Encontrar un conjunto de puntos clave distintivos.
2. Definir una región alrededor de cada punto clave de manera invariante a escala o afín.
3. Extraer y normalizar el contenido de la región.
4. Calcular un descriptor de la región normalizada.
5. Unir los descriptores locales.

2.2.1. Extracción de características tradicional

A continuación se explicarán algunos algoritmos que no utilizan aprendizaje profundo para la extracción de características en imágenes.

2.2.1.1. *Histogram of Oriented Gradient (HOG)*

En este algoritmo, la idea básica es que la apariencia y la forma del objeto local a menudo se pueden caracterizar bastante bien por la distribución del gradiente de intensidad local o las direcciones de los bordes, incluso sin un conocimiento preciso de las posiciones correspondientes del gradiente o del borde (Mao et al., 2010).

HOG permite aprovechar de forma eficiente la información del gradiente a partir de combinar esta información en forma de histogramas orientados locales, que se calculan en celdas de tamaño pequeño que se distribuyen uniformemente por toda la imagen. Un ejemplo se puede observar en la Figura 2.6.



Figura 2.6. Gradientes e histogramas de una imagen: Izquierda) Imagen original, Centro) Gradientes de la imagen, Derecha) Histograma de Gradientes Orientados (Ramos, 2016).

Los histogramas proporcionan información de las orientaciones de los contornos

que dominan cada una de las posiciones de la imagen. De esta manera, se permite distinguir la forma de los objetos presentes en una imagen y se observa la frontera entre un objeto y el otro (Ramos, 2016).

2.2.1.2. *Scale Invariant Feature Transform (SIFT)*

Este método transforma los datos de la imagen en coordenadas invariantes a la escala en relación con las características locales. Esto lo hace a través de diferentes etapas de cálculo utilizadas para generar el conjunto de características de la imagen. Las etapas se muestran a continuación (Lowe, 2004):

1. Detección extrema de espacio de escala: se busca en todas las escalas y ubicaciones de las imágenes. Se implementa de manera eficiente mediante el uso de una función de diferencia de Gauss para identificar posibles puntos de interés que son invariables para la escala y la orientación.
2. Localización de puntos clave: en cada ubicación candidata, se ajusta un modelo detallado para determinar la ubicación y la escala. Los puntos clave se seleccionan en función de las medidas de su estabilidad.
3. Asignación de orientación: se asignan una o más orientaciones a cada una de las ubicaciones de los puntos clave en función de las direcciones del gradiente de la imagen local. Todas las operaciones futuras se ejecutan en datos de la imagen que se han transformado en relación con la orientación, la escala y la ubicación asignadas para cada característica, lo que proporciona invariabilidad a estas transformaciones.
4. Descriptor de punto clave: los gradientes locales de la imagen se miden con la escala seleccionada en la región alrededor de cada punto clave. Estos se trans-

forman en una representación que permite niveles significativos de distorsión de forma local y cambio en la iluminación.

Un ejemplo se puede observar en la Figura 2.7.



Figura 2.7. Izquierda) Dos imágenes de objetos a buscar. Centro) Imagen de prueba con mucha oclusión. Derecha) Resultado de SIFT. El rectángulo dibujado muestra la ubicación de los objetos a buscar, y los pequeños cuadros indican los puntos clave que son usados para el reconocimiento (Lowe, 2004).

2.2.1.3. *Local Binary Patterns (LBP)*

Este método funciona para el análisis de textura basado en lo que se conoce como unidad de textura, donde una imagen de textura puede ser caracterizada por su espectro de textura. Como se muestra en Figura 2.8, una unidad de textura está representada por ocho elementos, cada uno de los cuales tiene uno de dos posibles valores obtenidos de una vecindad de 3×3 píxeles. De esta manera, a cada uno de los vecinos del píxel del centro se les asigna un valor binario, dependiendo si es mayor o menor que éste último, obteniendo así un vecindario llamado "vecindario umbral".

En total, hay $2^8 = 256$ posibles unidades de textura que describen patrones

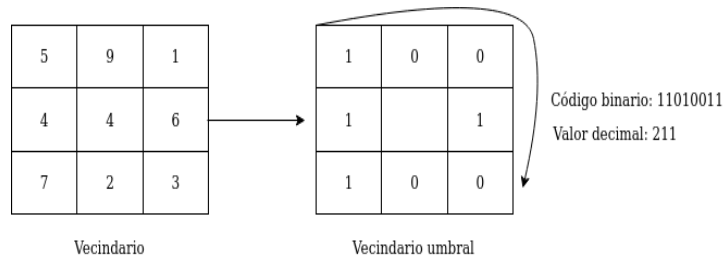


Figura 2.8. Operación de la unidad de textura de LBP (Shan et al., 2009).

espaciales de tres niveles en un vecindario de 3×3 . La aparición de la distribución de unidades de textura calculadas sobre una región se llama espectro de textura. Los valores de los píxeles en el vecindario umbral se multiplican por sus pesos dados. Finalmente, los valores de los ocho píxeles se suman para obtener el número de esta unidad de textura (Ojala et al., 1994).

Con los códigos binarios obtenidos, se crea un histograma de la imagen, como se aprecia en la Figura 2.9, para recoger las ocurrencias de diferentes patrones binarios (Pietikäinen, 2005). El histograma de 256 bandejas se usa como un descriptor de textura. Los números binarios derivados codifican primitivas locales que incluyen diferentes tipos de bordes curvos, puntos, áreas planas, etc. (Shan et al., 2009).

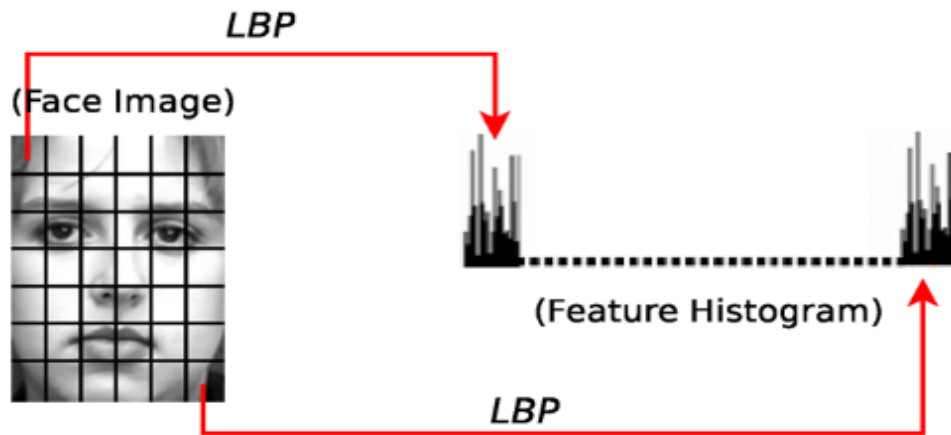


Figura 2.9. Obtención del histograma que funciona como descriptor de textura en LBP (Shan et al., 2009).

2.2.2. Extracción de características basada en aprendizaje profundo

El aprendizaje de la representación es un conjunto de métodos que permiten a una computadora ser alimentada con datos crudos y automáticamente descubrir las representaciones para detección o clasificación. Los métodos basados en aprendizaje profundo son métodos de aprendizaje de la representación con múltiples niveles, obtenidos al componer módulos simples pero no lineales que transforman la representación en un nivel bajo (comenzando con el dato crudo) en una representación en un nivel más alto, ligeramente un nivel más abstracto (LeCun et al., 2015). En la Figura 2.10 se puede observar un ejemplo de una arquitectura de una red neuronal convolucional.

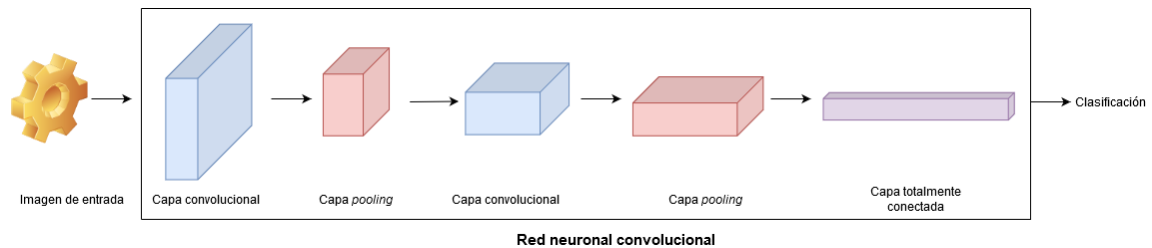


Figura 2.10. Arquitectura de una red neuronal convolucional con dos capas convolucionales, dos capas *pooling* y una capa totalmente conectada.

En el enfoque de aprendizaje profundo, la extracción de características y la clasificación han sido integradas como parte de un solo proceso *framework* de aprendizaje, el cual ha llegado a superar a los métodos tradicionales de selección y extracción de características (Yosinski et al., 2014). La principal idea del aprendizaje profundo es la de descubrir múltiples niveles de representación, donde las características de alto nivel representen una semántica más abstracta de los datos. Un ingrediente clave del aprendizaje profundo en la clasificación de imágenes es el uso de arquitecturas convolucionales. La inspiración del diseño de la red neuronal convolucional proviene

de la estructura del sistema visual de los mamíferos (Guo et al., 2017). Este tipo de redes hacen suposiciones fuertes y en su mayoría correctas sobre la naturaleza de las imágenes (Krizhevsky et al., 2012).

Para las tareas de clasificación, en las redes neuronales convolucionales, las capas superiores de representación amplifican aspectos de la entrada que son importantes para la discriminación y suprimen las variaciones irrelevantes. Una imagen, por ejemplo, viene en forma de una matriz de píxeles, y las características aprendidas en la primera capa de representación típicamente representan la presencia o ausencia de bordes en orientaciones y ubicaciones particulares en la imagen. La segunda capa normalmente detecta motivos al detectar disposiciones particulares de los bordes, independientemente de las pequeñas variaciones en las posiciones de los bordes. La tercera capa puede ensamblar motivos en combinaciones más grandes que corresponden a partes de objetos familiares, y las capas posteriores detectan objetos como combinaciones de estas partes. El aspecto clave del aprendizaje profundo es que estas capas de características no están diseñadas sino que se aprenden de los datos mediante un procedimiento de aprendizaje de propósito general (LeCun et al., 2015).

Las redes neuronales convolucionales incluyen principalmente tres tipos de capas, la convolucional, la de *pooling* y la totalmente conectada. A continuación se explican estos tipos de capas (Guo et al., 2017):

1. Capa convolucional. Esta capa es el núcleo de este tipo de redes, la cual, tiene conexiones locales y pesos de características compartidas. El objetivo de la capa convolucional es aprender representaciones de características de las entradas. Consta de varios mapas de características. Cada neurona del mismo mapa de características se usa para extraer características locales de diferentes posicio-

nes en la capa anterior, pero para las neuronas individuales, su extracción son características locales de las mismas posiciones en el mapa de características anterior. Para obtener una nueva característica, los mapas de características de entrada se combinan primero con un *kernel* aprendido y luego los resultados se pasan a una función de activación no lineal. Se pueden obtener diferentes mapas de características mediante la aplicación de diferentes *kernels*. Un ejemplo de un *kernel* se puede ver en la Figura 2.11.

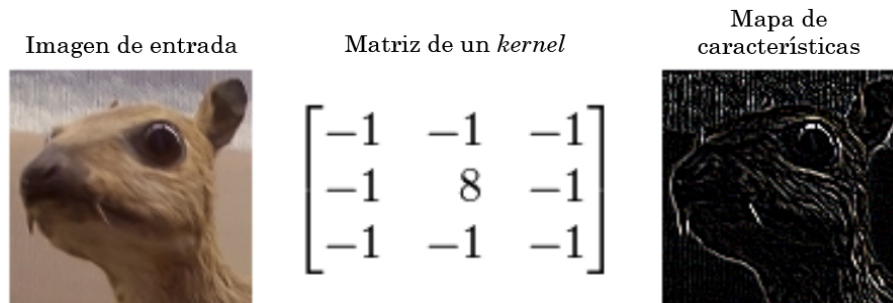


Figura 2.11. Ejemplo de *kernel* para la detección de aristas (figura extraída de [Wikipedia](#)).

2. Capa *pooling*. Esta capa se usa como un extractor de características secundario, puede reducir las dimensiones de los mapas de características y aumentar la robustez del proceso de extraer características. Generalmente, se coloca entre dos capas convolucionales. El tamaño de los mapas de características en esta capa se determina de acuerdo con el paso de movimiento de los *kernels*. Las operaciones típicas de *pooling* son la *average pooling* y *max pooling* (estas operaciones se pueden ver en la Figura 2.12). Se pueden extraer las características de alto nivel de las entradas apilando varias capas convolucionales y capas *pooling*.
3. Capa totalmente conectada. En general, el clasificador de las redes neuronales convolucionales es una o más de éste tipo de capas. Se toman todas las neuronas

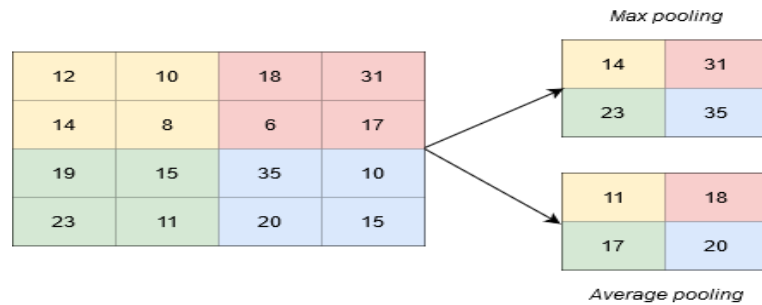


Figura 2.12. Ejemplo de las operaciones *max pooling* y *average pooling* en una matriz 4×4 .

de la capa previa y se conectan a cada una de las neuronas de la capa actual. No hay información espacial preservada en capas completamente conectadas. La última capa completamente conectada es seguida por una capa de salida que determina la clase a la que pertenece la imagen. En pocas palabras, es una red neuronal artificial que clasifica las características encontradas.

El aprendizaje profundo explota las propiedades de muchas señales naturales que son jerarquías compositivas, en las que las características de nivel superior se obtienen al componer las características de nivel inferior. En las imágenes, las combinaciones locales de bordes forman motivos, los motivos se ensamblan en partes y las partes forman objetos (LeCun et al., 2015).

2.2.3. Redes residuales

Las redes neuronales convolucionales profundas han llevado a una serie de avances para la clasificación de imágenes. Dichas redes integran naturalmente características de nivel bajo, medio y alto. Estos niveles de características pueden enriquecerse por el número de capas apiladas, es decir, la profundidad. Impulsado por la importancia de la profundidad, surge una pregunta: ¿Las redes aprenden mejor al apilar más capas?

En el aprendizaje profundo existe un problema llamado degradación de la precisión. Este problema trata de que, conforme se aumenta la profundidad de una red, se tiene el problema de que se va distorsionando el aprendizaje, produciendo que la precisión empeore conforme se agregan más capas a la red, y esto conduce a un mayor error de entrenamiento (He et al., 2016).

Una solución para este problema consiste en aplicar lo que se denomina como aprendizaje residual. Esto consiste en dividir la arquitectura de la red en varios bloques de capas, entonces, se replica la entrada del bloque anterior a la salida del bloque actual, de este modo conserva lo aprendido en capas anteriores y se agrega lo aprendido en las capas subsiguientes. Esto se le conoce como conexiones de acceso directo (He et al., 2016). Se puede observar un bloque residual en la Figura 2.13.

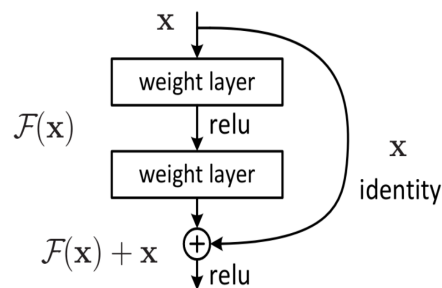


Figura 2.13. Conexión de acceso directo (He et al., 2016).

Aunque en este trabajo se utilizaron redes residuales, también se realizaron experimentos preliminares con otras dos tipos de redes, las cuales son InceptionV3 (Szegedy et al., 2016) y VGG19 (Simonyan y Zisserman, 2015). Éstas se explican a continuación:

- VGG19. VGG significa *Visual Geometry Group* en Oxford. VGG19 es una de las variantes del modelo VGG que consta de 19 capas: 16 capas de convolución, 3 capas totalmente conectadas, 5 capas *max pooling* y 1 capa *SoftMax*.

- InceptionV3. Es una red que consta de 48 capas de profundidad. Esta arquitectura apila 11 módulos *inception*, donde cada módulo consiste de capas *pooling* y filtros convolucionales con la función de activación ReLU (*Rectified linear units*).

Capítulo 3

Trabajos relacionados

En este capítulo, se presentan trabajos relacionados al estudio de animales. Se inicia con la sección 3.1, donde se comentan métodos tradicionales de campo para el estudio del borrego cimarrón. Posteriormente, en la sección 3.2, se habla de métodos enfocados en aprendizaje automático mediante el uso de imágenes, tanto para el reconocimiento de diferentes especies de animales como en la identificación individual dentro de algunos de ellos. Para finalizar, en la sección 3.3 se hace una recapitulación sobre lo visto y el camino que se está llevando actualmente el análisis visual de especies de animales en la vida silvestre.

3.1. Trabajos tradicionales

La importancia del estudio del borrego cimarrón se remarca en la creación de diferentes asociaciones para su conservación. En 1957 se celebró la Primera Reunión Anual del Consejo del Borrego Cimarrón del desierto (*Ovis canadensis nelsoni*) (<https://www.desertbighorncouncil.com/>) con el propósito de promover el avance del conocimiento y el bienestar a largo plazo de estos animales. Año tras año, se

presentan alrededor de una docena de estudios en la reunión anual que informan sobre su estado en los diferentes lugares donde reside, y también informan sobre estrategias de gestión diseñadas para mejorar las directrices de gestión actuales.

El modelado tradicional de la distribución del borrego cimarrón implica la definición de transectos que abarcan áreas extensas (Ruiz-Mondragón et al., 2018), principalmente debido a las enormes dimensiones de su hábitat. Las características del hábitat preferido por esta especie dependen de la elevación con montañas bajas, pendiente, rugosidad, terrenos rocosos y cortados por cañones (Montoya y Martínez-Gallardo, 2017). En consecuencia, ha habido monitoreo a través de rutas terrestres (Escobar-Flores et al., 2015; Ruiz-Mondragón et al., 2018), y también mediante observación aérea (Lee y Alcalá-Galvan, 2013).

La primera estimación poblacional del borrego cimarrón en México se realizó en 1959 calculando 2,000 individuos (Martínez-Gallardo et al., 2017). Desde entonces, se han llevado a cabo numerosos estudios. Desafortunadamente la densidad de población de la especie ha disminuido. En 1992, durante un censo de 5 meses que cubría un área de 900 km² a través de transectos, se contabilizaron 120 individuos; en 1995, mediante un censo aéreo que cubría 3.095 km², se identificaron 276 especímenes; en 1999, se estudiaron las mismas áreas mediante un censo aéreo que identificó a 282 especímenes; y años después, en 2008, a través de cinco censos terrestres realizados por 10 observadores durante 11 días, se identificaron sólo a 66 especímenes (Martínez-Gallardo et al., 2017).

El estudio de población más reciente fue realizado por Ruiz-Mondragón et al., 2018, utilizan un modelo de nicho ecológico con el objetivo de identificar el rango de distribución potencial del borrego cimarrón. El estudio está fechado de enero a junio de 2016, inspeccionando áreas desde la Sierra de Juárez en la frontera con los Estados

Unidos, hasta la Sierra de San Juan de la Costa adyacente a la Bahía de la Paz en el estado de Baja California. De este estudio, 116 de 196 registros corresponden a evidencia indirecta, como heces y huellas, mientras que el resto, es decir, 80 registros, fueron observaciones directas tomadas del censo aéreo de 2012 (Martínez-Gallardo et al., 2017). El censo aéreo anterior del borrego cimarrón puede consultarse en el informe de (Manterola-y-Piña, 2000).

De esta sección, queda claro que el monitoreo manual del borrego cimarrón es una tarea desalentadora que es crítica para la supervivencia de la especie y su ecosistema. Sin lugar a dudas, las herramientas automáticas para el apoyo de la tarea de monitoreo e incluso para la toma de decisiones automatizada pueden tener un impacto positivo al reducir la carga de trabajo de los biólogos o cubrir áreas más grandes que de otra manera no serían posibles. Este es precisamente el objetivo de este trabajo. En la siguiente sección se revisaron trabajos relacionados en esa dirección.

3.2. Trabajos que usan aprendizaje automático

Hasta donde se tiene conocimiento, no hay estudios previos que hayan intentado automatizar el reconocimiento de imágenes del borrego cimarrón (*Ovis canadensis*). No obstante, existen varios casos de estudio dedicados al desarrollo de herramientas para el análisis visual de animales en la vida silvestre. Por lo tanto, este apartado revisa el trabajo relacionado sobre métodos de aprendizaje automático para la automatización del monitoreo y análisis de diversas especies de animales. La revisión se dividió en enfoques centrados en especies animales en general y en técnicas que tratan especies en específico. La Tabla 3.1 muestra los trabajos revisados en esta sección. El trabajo de (Beery et al., 2018) es el único que presenta resultados para cada tarea. En cambio, los demás trabajos presentan diferentes métricas de evaluación.

Tabla 3.1. Comparación de los trabajos relacionados. En la columna *tarea*, R significa reconocimiento y D es de detección.

| Trabajo | Propuesta | Tarea | No. Especies de animales | Características | Clasificador | Métrica |
|------------------------|------------|-------|--------------------------|-----------------|--------------|--|
| Yu et al., 2013 | General | R | 57 | SIFT+LBP | SVM | Exactitud 82 % |
| Beery et al., 2018 | General | D+R | 15 | CNN | RCNN | R: Error Top-1 14.92 % D: mAP 95.04 % |
| Yousif et al., 2019 | General | D | 3 | CNN | DCNN | Exactitud 93.38 % |
| Falzon et al., 2019 | General | D+R | 8 | CNN | CNN | Exactitud 98.556 % |
| Miguel et al., 2016 | Específico | D+R | (1) Leopardo | SP | ML | - |
| Timm et al., 2018 | Específico | R | (1) Jaguar | SIFT | CNN | Exactitud 91.5 % |
| Körschens et al., 2018 | Específico | D+R | (1) Elefante | CNN | SVM | Exactitud 56 % |
| Buehler et al., 2019 | Específico | R | (1) Jirafa | HOG | SVM | Radio de error promedio 10.9 % |

3.2.1. Trabajos de categorización

En (Yu et al., 2013), se trabajó en el reconocimiento de imágenes de cámaras trampa. Utilizaron 10,598 imágenes de 57 especies de dos diferentes hábitats, un bosque tropical (Isla Barro Colorado, Panama) y un bosque templado (Parque nacional Hoge Veluwe, Países bajos). La propuesta consiste en representar las imágenes utilizando dos diferentes descriptores visuales tradicionales SIFT y LBP, entonces, se creó un histograma a partir de los dos conjuntos de características extraídas. Finalmente, un modelo fue generado a partir de un entrenamiento del algoritmo *Support Vector Machine* (SVM). Por otro lado, el trabajo en (Yousif et al., 2019) es dedicado a la detección de animales, también a partir de cámaras trampa. Primero, se detectaba el movimiento de los animales en las imágenes utilizando un método para la sustracción del fondo. Segundo, una *Deep Convolutional Neural Network* (DCNN) fue usada para la segmentación la región del animal. Finalmente, la clasificación fue ejecutada en tres diferentes clases (humano, animal o fondo).

Similarmente, en (Beery et al., 2018) se realizó detección y clasificación de 15 diferentes especies de animales. Para la detección, se utilizó Faster-RCNN (Ren et al., 2015), el cual fue entrenado utilizando dos diferentes *backbones*, ResNet-101

e Inception-ResNet-v2. En el lado de la clasificación, se utilizó InceptionV3 (Szegedy et al., 2016). En el trabajo de (Norouzzadeh et al., 2018), también se utilizó aprendizaje profundo para la identificación, y clasificación automática de 48 diferentes especies de animales. Los modelos fueron entrenados utilizando la base de datos de *Snapshot Serengeti* (Swanson et al., 2015), la cual cuenta con 3.2 millones de imágenes.

En (Falzon et al., 2019), se introduce un sistema llamado *ClassifyMe* para identificar animales en imágenes de cámaras trampa. *ClassifyMe* es un software que utiliza un marco de aprendizaje automático, los usuarios pueden solicitar la detección de animales aplicando modelos aprendidos de conjuntos de datos que también proporcionan los mismos usuarios.

En (Shahinfar et al., 2020), se explora el uso de diferentes arquitecturas de aprendizaje profundo para evaluar el equilibrio entre el número de clases y la cantidad de muestras de imágenes a utilizar.

3.2.2. Trabajos de identificación individual

Hay numerosos estudios dedicados a animales en específico. En el trabajo de (Miguel et al., 2016) se investiga al leopardo de las nieves, un animal enlistado como vulnerable porque su población global es baja. El método propuesto consiste de tres pasos: un secuencia de imágenes es la entrada, luego el fondo es substraído por medio de *median filtering*, las características son extraídas utilizando una técnica de superpíxeles, el paso final es usar *feature thresholding* para la detección de movimiento por medio de dos diferentes estrategias basadas en *Mean of Differences* (MoD), y *Difference of Means* (DoM). Otro felino que es de interes es el jaguar. El método propuesto en (Timm et al., 2018), considera la identificación individual del jaguar y

la clasificación de los 10 animales más comunes encontrados en la imágenes de las cámaras trampa que utilizaron. Para la tarea de clasificación utilizaron dos modelos de InceptionV3 entrenados con diferentes conjuntos. Para la identificación del jaguar, se utilizaron los descriptores SIFT y *Random Sample Consensus* (RANSAC) para encontrar características que correspondan entre los puntos de interes extraídos.

La investigación en (Körschens et al., 2018) es un trabajo de identificación a nivel individual en elefantes. Para la detección, se utilizó YOLO (Redmon et al., 2016), el cual fue entrenado con imágenes de elefantes obtenidas a partir de Flickr. Para la tarea de identificación, se utilizó ResNet50 como extractor de características. Finalmente, un SVM fue entrenado con las características extraídas. Un trabajo parecido es el de (Buehler et al., 2019), el cual realiza la detección en jirafas. En este trabajo el descriptor visual HOG es utilizado para la representación de imágenes y un SVM fue entrenado para realizar la detección.

3.3. Recapitulación

A partir de la revisión de este capítulo, y hasta donde se tiene conocimiento, el borrego cimarrón no ha sido considerado en estudios para el análisis visual de animales, a pesar de su relevancia e impacto potencial. Por lo tanto, esta propuesta es novedosa y podría tener un impacto positivo en el análisis de esta especie en particular. Específicamente, el impacto local podría ser enorme, ya que el borrego cimarrón se ha estudiado principalmente en México mediante la inspección directa del hábitat de la especie. En este sentido, las áreas protegidas podrían incorporar cámaras trampa cerca de los recursos de agua para monitorear el estado de su población.

Además, de la revisión se puede apreciar que el aprendizaje profundo se está convirtiendo en una metodología establecida para el análisis del comportamiento

animal. Esto se debe principalmente a la efectividad de este tipo de modelos. Sin embargo, para lograr resultados satisfactorios con la mayoría de los métodos de aprendizaje profundo, se requieren grandes cantidades de datos. Este no es el caso del análisis de borrego cimarrón a partir de imágenes de cámaras trampa, porque las imágenes que representan a esta especie son escasas. Por lo tanto, se debe recurrir a metodologías alternativas, como la presentada en este trabajo a través de extracción de características, para aliviar la necesidad de grandes cantidades de imágenes etiquetadas.

Capítulo 4

Metodología

El presente capítulo de la metodología está dividido en diferentes partes. En la sección 4.1 comienza con una explicación de los pasos a realizar para cumplir con los objetivos propuestos en las secciones 1.3 y en 1.4. Después, en la sección 4.2 se describe la construcción de las bases de datos empleadas. Primero en la subsección 4.2.1 se habla sobre la construcción de una base de datos sintética por medio de una recolección de imágenes de la Web con el propósito de generar un depósito inicial de datos del borrego cimarrón y otras especies de animales que están relacionadas. Posteriormente, se menciona el preprocesamiento que se realizó para la construcción de dicha base de datos. Luego, en la subsección 4.2.2 se habla sobre la obtención de imágenes a partir de cámaras trampa del borrego cimarrón y la zona de donde se obtuvieron estas imágenes. Más adelante, en la sección 4.3 se menciona la extracción de características a partir de un modelo de aprendizaje profundo pre-entrenado. El capítulo termina con la sección 4.4, donde se mencionan los diferentes escenarios en los que se trabajó para medir la funcionalidad de los modelos generados a partir de diferentes algoritmos de aprendizaje automático que nos permiten la clasificación automática de imágenes de borrego cimarrón y otros animales.

4.1. Pasos a seguir

En la Figura 4.1 se ilustra el proceso de la metodología propuesta de manera gráfica.

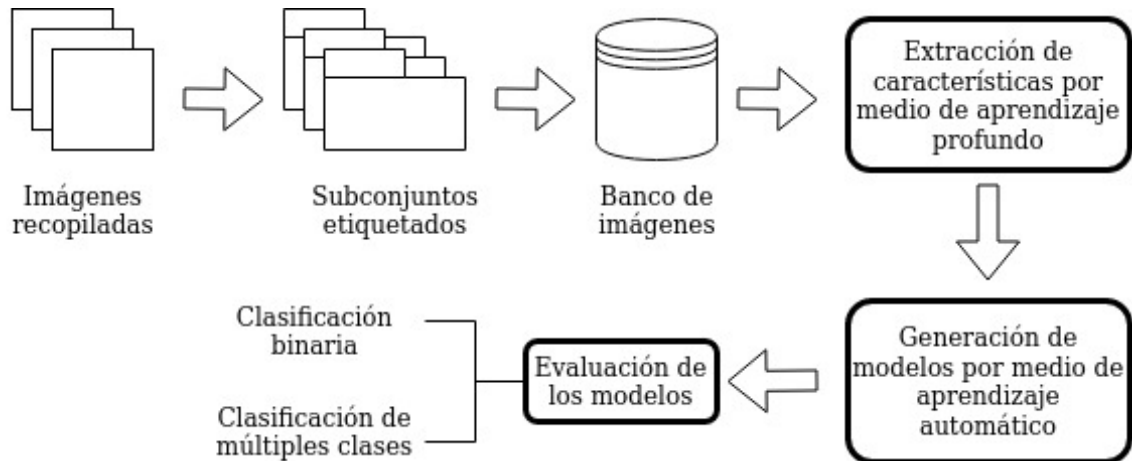


Figura 4.1. Proceso de la metodología del trabajo.

El **primer paso** del proyecto involucra la recopilación de datos y su procesamiento. Los repositorios iniciales a usar son colecciones de imágenes extraídas de la Web, e imágenes inéditas proporcionadas por las cámaras trampa tomadas por los distintos dispositivos instalados en zonas de observación o avistamiento del borrego cimarrón.

Para los bancos de imágenes se trabajarán las siguientes actividades:

- Recopilación de datos. Esto consiste en la obtención de los datos a utilizar en el proyecto por medio de diferentes fuentes.
- Etiquetado de subconjuntos de datos para evaluación. Es decir, una carpeta donde sólo haya imágenes del borrego cimarrón. Esto se hace separando las imágenes en diferentes carpetas, donde cada carpeta tenga imágenes de una sola categoría.

- Indexado y análisis. Involucran el procesamiento del banco de imágenes mediante el uso de arquitecturas de redes pre-entrenadas por aprendizaje profundo. Esto es por medio de una transformación de las imágenes a un espacio vectorial para ser manipuladas de manera más sencilla, es decir, el vector de características. Se experimentará con algunas para elegir la que mejores resultados dé.

El **segundo paso** de la metodología se centra en la generación de modelos de inferencia a través de la evaluación de algoritmos de aprendizaje automático para la clasificación del borrego cimarrón:

- Se proponen métodos que aprovechan la utilización de características extraídas a partir de arquitectura pre-entrenada en aprendizaje profundo que permitan el reconocimiento del borrego cimarrón.

Durante el **último paso** se evalúa el alcance de la metodología propuesta para la clasificación automática del borrego cimarrón. Esto se hace por medio de la definición de escenarios que tienen como objetivo medir la complejidad de la tarea, con la finalidad de que se permita obtener un mejor desempeño para el reconocimiento del borrego cimarrón.

4.2. Construcción de las bases de datos

La creación y uso de la base de datos sintética tiene como objetivos el evaluar representaciones, algoritmos, escenarios y lo más importante, el complementar y/o aliviar problemas de inferencia que puedan originarse en la base de datos real. Por otro lado, la base de datos real es a la que se busca dar la solución de reconocer al borrego cimarrón en imágenes tomadas por cámaras trampa.

Primero se trabajó con la base de datos sintética, en la subsección 4.2.1, para poder saber qué tan complicada puede llegar a ser la tarea de reconocer al borrego cimarrón. Además, también funciona para saber si esta base de datos se está construyendo adecuadamente para el entrenamiento de modelos que se probarán con la base de datos real (presentada en la subsección 4.2.2).

4.2.1. Base de datos sintética

Para construir la base de datos sintética con la mayor cantidad de imágenes necesarias para producir buenos modelos, se consultaron diferentes fuentes utilizando palabras clave relacionadas en dos idiomas diferentes (español e inglés). Primero se llevo a cabo una búsqueda de conjunto de datos a utilizar, encontrando *iWildCam 2019 Competition* (Beery et al., 2019). Se trata de una competencia para la detección y categorización de diferentes especies de animales a partir de imágenes de cámaras trampa. Consta de varias bases de datos, pero sólo una indicaba tener imágenes del borrego cimarrón. De aquí, se descargaron los conjuntos de datos de iNat Idaho y Smaller CCT (CCT), donde iNat Idaho consta de 250 imágenes del borrego cimarrón, pero sólo 122 eran útiles. Aun así, se descargaron porque contenían imágenes de especies de animales relacionadas al borrego cimarrón. Después, se consultaron otros sitios Web: iNaturalist (iNaturalist.org, Accessed [2019]) y Naturalista (Naturalista, Accessed [2020]) que están especializados en el registro de especies animales avistadas. Otra fuente considerada fue Flickr (Flickr, Accessed [2020]), que es uno de los depósitos de imágenes más grandes de la Web. Además, se usó a *Caltech Camera Traps* (Beery, Accessed [2019]), de donde se descargó la base de datos Small ECCV18 (ECCV18). En la siguiente subsubsección (4.2.1.2) se hace mención de la cantidad de imágenes obtenidas para ésta base de datos.

4.2.1.1. Credenciales de las bases de datos

Las bases de datos iNat Idaho, CCT y ECCV18 cuentan con la licencia *Community Data License Agreement (CDLA)*. En la cual, su punto 1.2 establece el “uso computacional”, el cual, incluye la aplicación de cualquier técnica analítica computacional, cuyo propósito es el análisis de cualquier dato en forma digital para generar información sobre los datos tales como patrones, tendencias, correlaciones, inferencias, percepciones y atributos. Por parte de iNaturalist y Naturalista, permiten el uso de sus datos con fines investigativos con la condición de citar los sitios Web. Por otro lado, Flickr ofrece una interfaz de programación de aplicaciones (API, por sus siglas en inglés) que permite realizar alguna acción o acceder a alguna característica o contenido que el sitio provee. En este caso fueron imágenes de diferentes especies de animales. La [API de Flickr](#) está disponible para uso no comercial por parte de desarrolladores externos, y el uso comercial es posible mediante un acuerdo previo.

4.2.1.2. Adquisición y selección

Con respecto a la selección de especies animales relacionadas y adicionales a estudiar, se consultó a dos expertos en el tema: Dr. Guillermo Romero Figueroa y Dr. Aldo Antonio Guevara Carrizales, ambos son investigadores del área de biología de la Facultad de Ciencias en la Universidad Autónoma de Baja California (UABC). También se llevo una consulta de una guía de referencia de especies animales representativas del estado de Baja California ([Guevara-Carrizales et al., 2016](#)). De lo anterior, se consideró al lince, el coyote, el venado, el puma y aves en general (por ejemplo, águila, cathartes aura, cuervo, búho y paloma) como especies de estudio relacionadas al borrego cimarrón.

En las bases de datos CCT, ECCV18, iNat Idaho venía un archivo en formato

JSON en el que se podían observar anotaciones de las imágenes, como el nombre, la categoría a la que pertenece, en caso de contar con *bounding box* indicar dónde se encuentra, un id, entre otras cosas. Es importante mencionar que el nombre de las imágenes no había sido modificado al realizar cada una de las bases de datos, es decir, todas las imágenes seguían con el nombre que les asignó la cámara trampa que las tomó, esto significa que para saber qué había en la imagen se tenía que ver o buscar sus anotaciones en el archivo *JSON*, por lo tanto, se realizó un *script* para leer cada uno de dichos archivos y poder separar las imágenes en su respectiva categoría.

Como resultado, se reunió un total de 6,744 imágenes para el borrego cimarrón. Además, se juntaron 22,608 imágenes de diferentes especies animales relacionadas con él. También, se obtuvieron 6,744 imágenes que no contienen a un animal, y están etiquetadas como "Fondo". La Tabla 4.1 muestra la fuente y el número de imágenes de cada especie animal. Igualmente, se puede observar el número de imágenes recopiladas por cada fuente. Dos anotadores inspeccionaron todas las imágenes descartando aquellas con derechos de autor, imágenes corruptas de baja resolución, imágenes modificadas y duplicadas. En la Figura 4.2 se pueden observar imágenes de los animales

Tabla 4.1. Base de datos sintética.

| Animales | No. Imágenes por fuente | | | | | | Total |
|--------------|-------------------------|-----------|--------------|--------------|--------------|--------------|---------------|
| | iNat | Natu | Flickr | iNat Idaho | CCT | ECCV18 | |
| BC | 3,212 | 44 | 3,366 | 122 | - | - | 6,744 |
| Puma | 33 | - | 1,131 | 53 | 24 | - | 1,241 |
| Coyote | - | - | - | 1,161 | 1,372 | 2,492 | 5,025 |
| Venado | - | - | - | 3,049 | 3,374 | 92 | 6,515 |
| Lince | - | - | - | 673 | 1,113 | 2,377 | 4,163 |
| Aves | 91 | - | 4,645 | - | - | 928 | 5,664 |
| Fondo | - | - | - | - | 3,454 | 3,290 | 6,744 |
| Total | 3,336 | 44 | 9,142 | 5,058 | 9,337 | 9,179 | 36,096 |

*iNat = iNaturalista y *Natu = Naturalista.

de diferentes fuentes. De igual manera, se puede ver que las imágenes 4.2a y 4.2c fueron tomadas por fotógrafos, a comparación de las demás, que se puede observar que son tomadas por cámaras trampa.

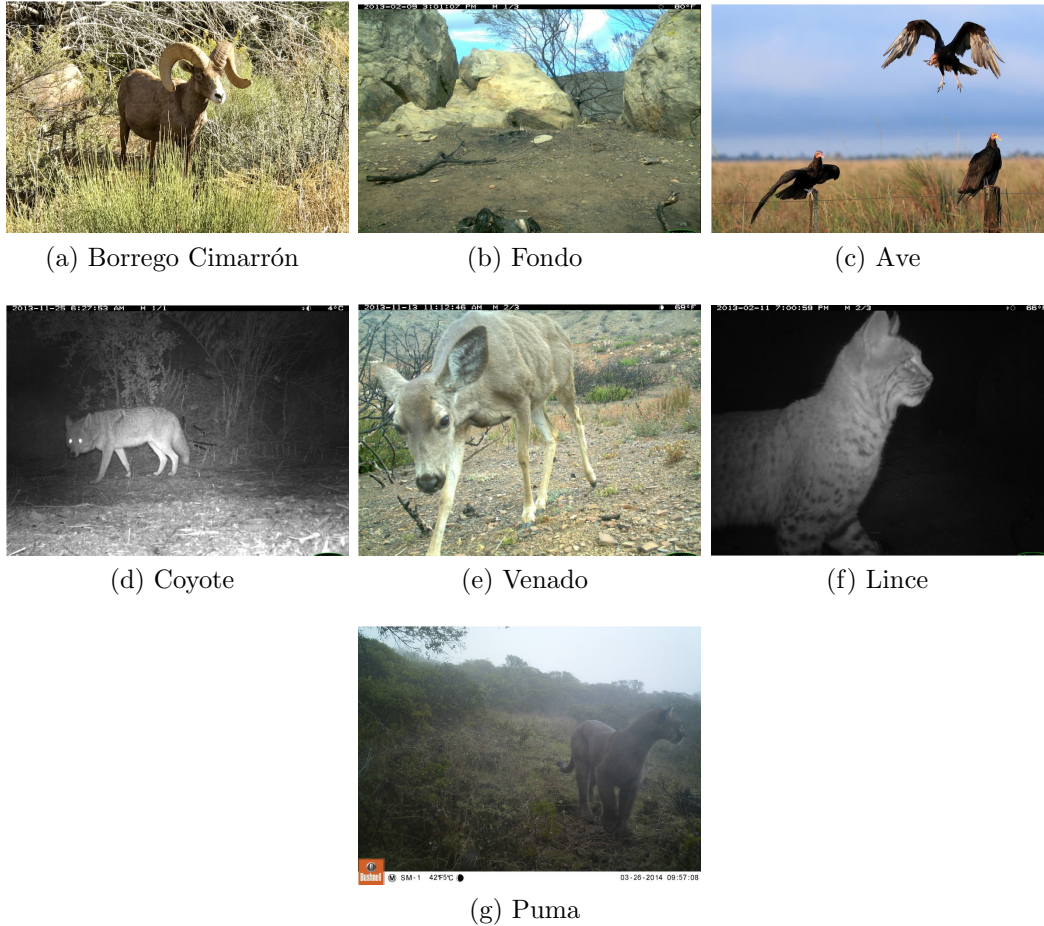


Figura 4.2. (a) Imagen obtenida a partir de iNaturalist. (b) Imagen obtenida a partir de ECCV18. (c) Imagen obtenida a partir de Flickr. (d) Imagen obtenida a partir de CCT. (e) Imagen obtenida a partir de ECCV18. (f) Imagen obtenida a partir de CCT. (g) Imagen obtenida a partir de iNat Idaho.

4.2.2. Base de datos real

La problemática consiste en identificar al borrego cimarrón a partir de imágenes tomadas por cámaras trampa. Para evaluar la metodología propuesta para el

reconocimiento del borrego cimarrón y otros animales, se recolectaron imágenes directamente de cámaras trampa ubicadas cerca de fuentes de agua naturales. Las observaciones se realizaron desde noviembre de 2018 hasta septiembre de 2019, donde se utilizaron cinco cámaras trampa equipadas con visión nocturna. Cada una de las cámaras fue configurada para que cuando se active el sensor se tome una secuencia de tres fotografías con una resolución de 12mp, y se iba registrando la fecha, hora y temperatura.

Las imágenes fueron provistas por el Dr. Aldo Antonio Guevara Carrizales me-

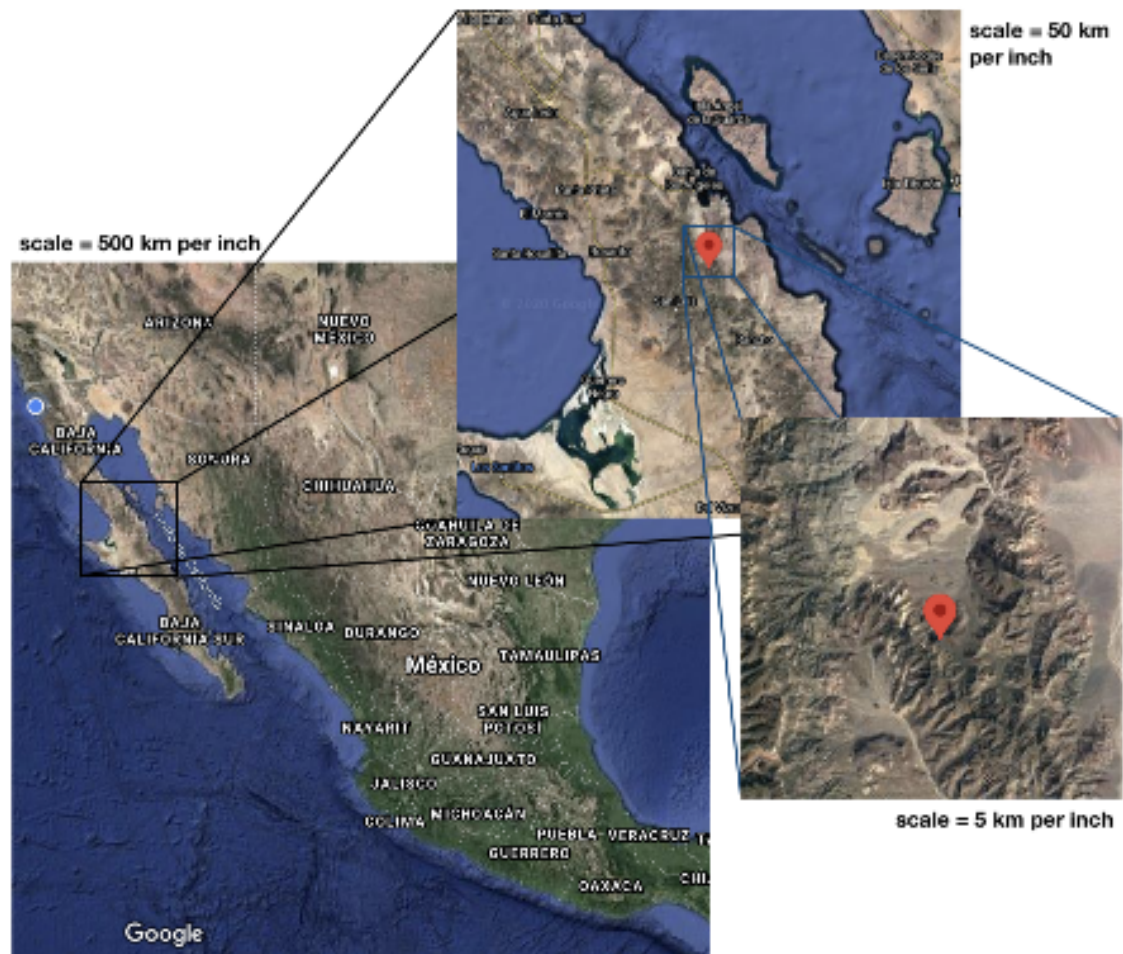


Figura 4.3. Localización del Área de Protección de Flora y Fauna Valle de los Cirios

diante el programa de monitoreo del borrego cimarrón con cámaras trampa instaladas en el Rancho Triple A, ubicado a 37 km al sur de Bahía de los Ángeles, en el Área de Protección de Flora y Fauna Valle de los Cirios, el cual, se encuentra en las coordenadas N 28° 35'35", W 113° 29'02". En la Figura 4.3 se puede observar un mapa de la localización específica del lugar. La elección de este lugar fue gracias al estudio previo del borrego cimarrón en (Martínez-Gallardo et al., 2017) que revela una población numerosa en dicha zona. Además de esta especie, en esta zona es posible observar venados, pumas, lince, zorros, buitres, entre otras especies de animales.

Aproximadamente 18,000 imágenes fueron recolectadas a partir de las cámaras trampa. El número de imágenes es relativamente pequeño para un método automático, sin embargo, el etiquetado que se realizó manualmente implicó un gran esfuerzo. Entre las especies de animales detectadas en las imágenes, se observa el borrego cimarrón, el lince, el venado, el coyote, el zorro y aves de diferentes tamaños como águilas, buitres, cuervos y palomas.

La Tabla 4.2 agrupa a las especies de animales detectadas por las cámaras trampa. Por simplicidad, se agrupó a los diferentes tipos de aves en una sola categoría, esto es así, porque el borrego cimarrón es la especie principal de este trabajo. Se tuvieron

Tabla 4.2. Cantidad de imágenes de las cámaras trampa

| Animales detectados | | |
|----------------------------|---------------------|--------------------|
| Categoría | No. Imágenes | Porcentajes |
| Borrego Cimarrón | 775 | 4.3 |
| Lince | 51 | 0.28 |
| Coyote | 2319 | 12.89 |
| Venado | 12 | 0.06 |
| Fondo | 2492 | 13.85 |
| Aves | 11972 | 66.5 |
| Falla | 381 | 2.12 |
| Total | 18002 | 100 |

que descartar 333 imágenes, porque presentaban animales domésticos como caballos y perros, dichos animales fueron fotografiados cuando las cámaras eran instaladas, de igual manera, entre las imágenes descartadas había personas. Se debe de observar que en la Tabla 4.2, el borrego cimarrón sólo representa el 4.3% del total de las imágenes, aunque es un área protegida para esta especie animal, se pueden observar diferentes especies de animales. Mientras que en la Figura 4.4 se muestran imágenes de cada uno de los animales que se encontraron en las cámaras trampa, en la Figura 4.5 se pueden observar imágenes dañadas de las cámaras trampa.

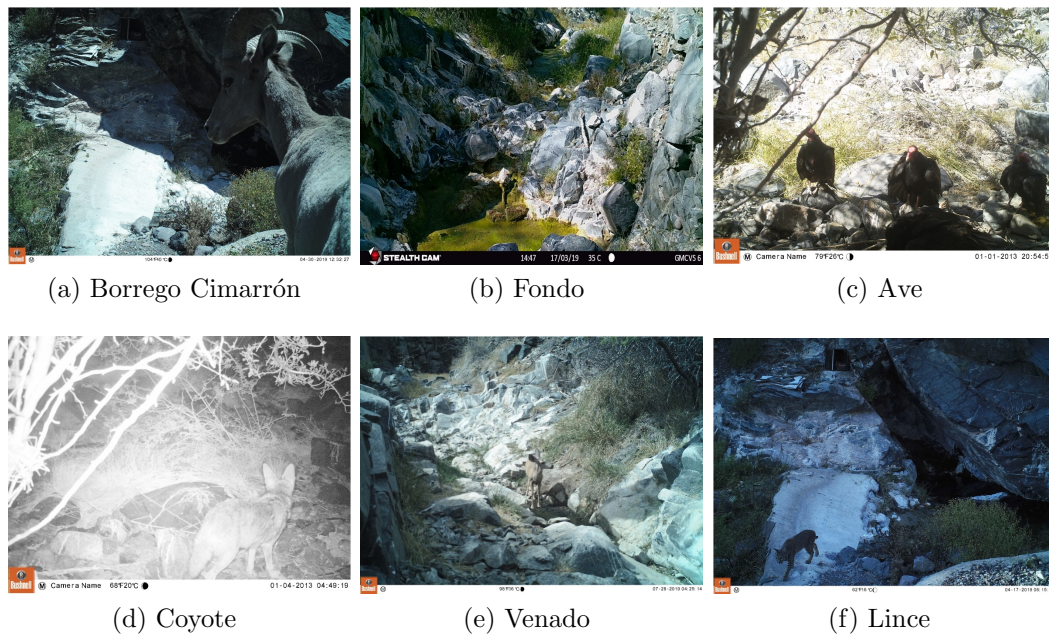


Figura 4.4. (a) Imagen de un borrego cimarrón. (b) Imagen del fondo. (c) Imagen de un ave. (d) Imagen de un coyote. (e) Imagen de un venado. (f) Imagen de un lince.

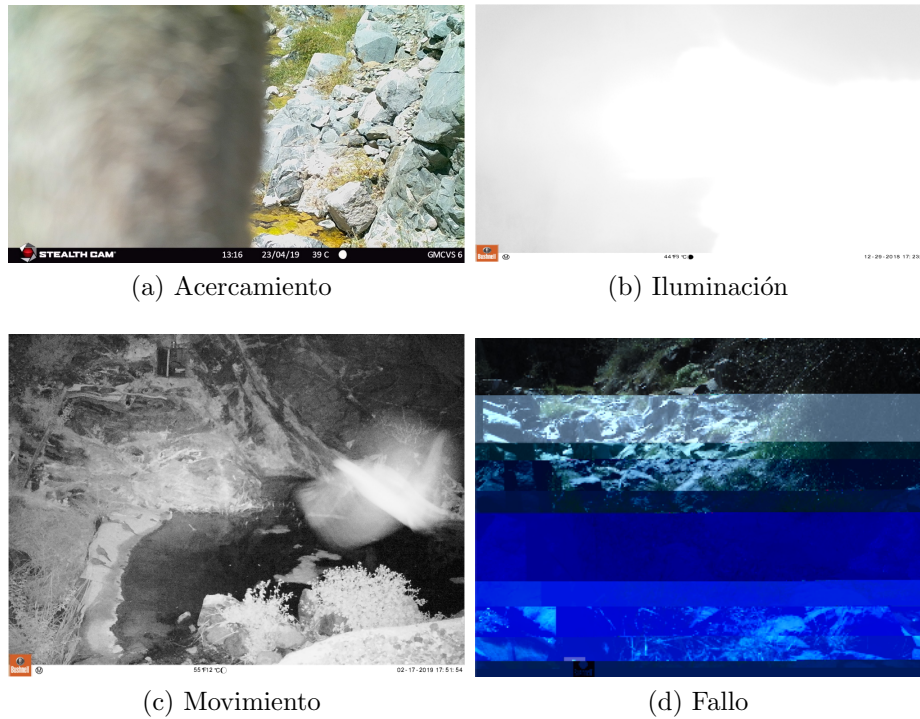


Figura 4.5. (a) En algunas ocasiones los animales se encuentran demasiado cerca de la cámara. (b) Si el animal está muy cerca, el flash de la cámara puede hacer que la imagen salga así. (c) Cuando el animal se encuentra en movimiento, es normal que la imagen salga así. (d) Algunas veces las cámaras pueden fallar, tomando fotografías como esta.

4.3. Extracción de características a través de aprendizaje profundo

Este proceso consiste en la obtención de los vectores de características de las imágenes que construyen las bases de datos generadas. Esto es, para poder utilizar dichos vectores en los algoritmos de aprendizaje automático y de esta forma se generen los modelos que harán la tarea de reconocer al borrego cimarrón.

Se experimentó con diferentes redes neuronales convolucionales pre-entrenadas antes de elegir una que funcionara de base para la extracción de características. Aunque, estas redes convolucionales fueron entrenadas usando datos de ImageNet,

cada una de ellas ofrece diferentes capacidades según el tiempo necesario para ser procesadas y la representación de características (tamaño del vector de características y poder discriminatorio) que pueden ser decisivas con el objetivo en cuestión.

Luego de realizar experimentos preliminares utilizando un pequeño conjunto de la base de datos sintética, se eligió *ResNet-50* (He et al., 2016) para este estudio porque mostró la mejor compensación entre efectividad y eficiencia. La implementación utilizada fue obtenida a partir de la API de *Keras* de la librería de *Tensorflow* (Tensorflow, 2015). En la Figura 4.6 se puede ver la idea del proceso que se realiza para la extracción de características.

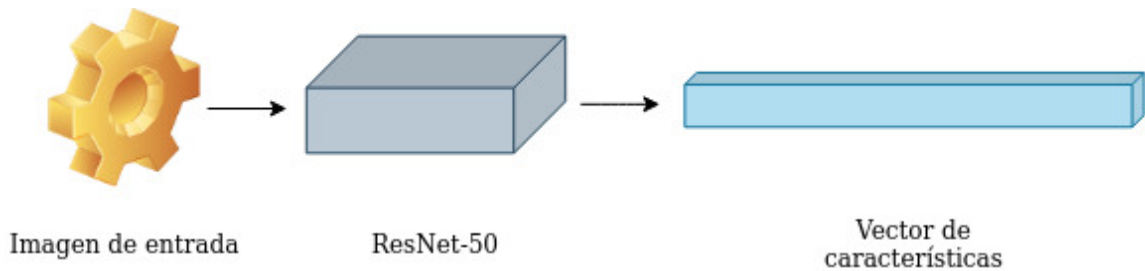


Figura 4.6. Proceso de extracción de características a partir de *ResNet-50*.

4.4. Escenarios de clasificación para reconocimiento del borrego cimarrón

Los escenarios planteados en este trabajo son para medir la dificultad de la tarea, y con el objetivo de definir aquellos escenarios propicios que permitan obtener un mejor desempeño para el reconocimiento del borrego cimarrón. Por un lado, la base de datos sintética, funciona bien para tener una idea de que tan complicado puede ser la tarea antes mencionada, de igual manera, funciona bien para observar qué tanto puede influir la calidad de las imágenes. Por otro lado, la base de datos real,

es la tarea a resolver y esta muestra ser desafiante, ya que presenta un desbalance en los datos, lo que dificulta el aprendizaje de los modelos.

A continuación, se presentan los diferentes escenarios considerados, cada uno de ellos tiene un objetivo y una complejidad específica, en seguida se describen:

- **Clasificación binaria.** Dado que el borrego cimarrón es la especie animal de interés en este trabajo, se definen tres escenarios diferentes que utilizan una clasificación binaria. Todos ellos enfocados en la búsqueda del borrego cimarrón y pueden aplicarse en consecuencia a la diversidad de la información:
 1. Borrego cimarrón vs Fondo. El conjunto de datos está compuesto sólo por imágenes del borrego cimarrón y el fondo. Se considera fondo a cualquier imagen de una cámara trampa que no contenga a algún animal.
 2. Borrego cimarrón vs Otros animales. En este caso, el conjunto de datos está compuesto por el borrego cimarrón y los otros cinco animales considerados, pero la clasificación aún se define como binaria.
 3. Borrego cimarrón vs Otros animales + Fondo. En este otro caso, el conjunto de datos está compuesto por el borrego cimarrón y los otros cinco animales considerados más imágenes del fondo, aún se considera una clasificación binaria, ya que sólo hay dos clases definidas.
- **Clasificación multiclase.** En este escenario, cada especie animal, además del caso de Fondo, se considera individualmente. El propósito de este escenario es proporcionar un segundo nivel de análisis. La complejidad en comparación con la tarea binaria es mayor debido a la diversidad en las imágenes, y también por la cantidad desequilibrada de muestras de cada especie animal en los conjuntos de datos.

- **Evaluación cruzada.** En este tipo de evaluación, se utiliza la base de datos sintética para entrenar, pero la evaluación del modelo generado se realiza con la base de datos real. Por lo tanto, la efectividad podría verse gravemente afectada por la diferencia entre la fuente y el objetivo. La idea de este escenario es la de observar qué tan eficaz resulta ser el entrenamiento de modelos a partir de imágenes con buena iluminación, bien enfocadas, en donde el animal se encuentre centrado, entre otras cosas. Para que sean probados a partir de imágenes con posibles problemas de iluminación, sin un lugar específicamente enfocado, el animal se puede encontrar en cualquier lugar de la imagen, entre otras cosas. Otra opción para este escenario, es la de utilizar un nuevo conjunto de entrenamiento compuesto tanto de imágenes de la base de datos sintética como la real. Esto es con la finalidad de complementar y/o aliviar problemas de inferencia que tiene la base de datos real.

Capítulo 5

Experimentos y Resultados

En este capítulo se presenta una evaluación completa en cada una de las bases de datos utilizadas. En la sección 5.1 se describen las configuraciones empleadas en los algoritmos de aprendizaje automático utilizados para la generación de modelos. Después en la sección 5.2, se da una descripción de las configuraciones utilizadas para los experimentos realizados en los escenarios mencionados previamente en la sección 4.4. Posterior a eso, la sección 5.3 se dividió en tres subsecciones: la de la base de datos sintética, la de la real y la de la evaluación cruzada. En cada una se habla sobre los resultados obtenidos por cada modelo en cada uno de los escenarios en los que se trabajó. Por último, en la sección 5.4 se discute sobre los resultados obtenidos.

5.1. Algoritmos de aprendizaje automático

Para la generación de modelos de aprendizaje para la clasificación automática de imágenes, se experimentó con algoritmos tradicionales de aprendizaje automático a partir de los principales paradigmas implementados en el *framework* Weka (Hall

et al., 2009). Los algoritmos utilizados son los siguientes, y se utilizaron los parámetros que tiene Weka por defecto:

- SVM. El tipo de *kernel* utilizado fue *radial basis function* y su grado fue de 3, la condición de terminación es 0.001, el valor de costo que maneja es de 1.0, el valor de gamma es $\frac{1}{2048}$ (es el tamaño del vector de características), el peso por defecto para cada clase es de 1. Además, se experimentó normalizando los datos, y no haciéndolo y se reportó el que dio mejores resultados.
- *Lib Linear* (LL). La función de penalidad utilizada es *L2-regularized L2-loss* y el parámetro ϵ de la función de pérdida es 0.1, el valor de sesgo es de 1.0, la condición de terminación es 0.001, el valor de costo que maneja es de 1.0, el valor de gamma es $\frac{1}{2048}$ (es el tamaño del vector de características), el peso por defecto para cada clase es de 1, por último, la cantidad de iteraciones que se utilizan para generar el modelo es de 1000. Además, se experimentó normalizando los datos, y no haciéndolo y se reportó el que dio mejores resultados.
- ANN. Se usó el algoritmo del perceptrón emulando la técnica *fine-tuning*, se entrenó en 10 épocas. Se utilizó un ratio de aprendizaje de 0.001, el algoritmo de optimización es el descenso del gradiente estocástico, cuenta con una capa de entrada de 2,048 nodos (es el tamaño del vector de características) y cantidad de nodos que utiliza la capa de salida depende de la cantidad de clases que se maneje, además, esta capa utiliza la función de activación *SoftMax* para determinar la clase de las instancias de prueba.
- KNN. Se experimentó con $k = 3|5$ vecinos junto a la distancia euclidiana y se reportó el que dio mejores resultados.
- *Naive Bayes* (NB). Al ser características numéricas, los datos se tienen que

discretizar. Dicha discretización se lleva a cabo por medio de una distribución gaussiana.

- *Decision Tree* (DT). Utiliza un factor de confianza utilizado para un posible podado del árbol, dicho valor es de 0.25. También se configuró para que el número mínimo de instancias por hoja sea 2.

5.2. Particiones de datos utilizadas en los experimentos

Se definieron condiciones justas para la evaluación, por lo que, se utilizó tanto en la clasificación binaria como en la clasificación de clases múltiples, dos tercios de los datos para el entrenamiento, y un tercio para probar. Esta sencilla configuración ayuda a tener una mejor interpretación de la exactitud, que puede ser interpretada erróneamente en bases de datos desbalanceadas debido al sobreajuste.

Siendo más específico, dos tercios del entrenamiento en la clasificación binaria usando la base de datos sintética corresponden a 4,496 instancias por clase, y un tercio como prueba son 2,248 instancias por clase. Por lo tanto, durante el entrenamiento se utilizan 8,992 instancias para generar un modelo, y durante la prueba se evalúan 4,496 instancias. Además, cuando se compara el Borrego cimarrón con otros animales en este escenario de clasificación binaria, se consideró el mismo número de muestras por cada especie animal en el entrenamiento y también la prueba (899 y 449 muestras por especie animal, respectivamente). De igual manera, en el último escenario donde se combinan la clase 'Fondo' más otros animales, el número de muestras de cada animal es igual (449 y 224 muestras por especie animal, respectivamente).

En la Tabla 5.1 se muestran las particiones en las que se dividieron los datos

para el experimento de la clasificación de múltiples clases utilizando la base de datos sintética, en donde las imágenes utilizan la mayor cantidad posible para cada especie animal.

Tabla 5.1. Número de imágenes usadas para el entrenamiento y prueba por cada animal en la base de datos sintética.

| Especies de animales | No. Imágenes usadas para: | |
|----------------------|---------------------------|--------|
| | Entrenar | Probar |
| Borrego Cimarrón | 4,496 | 2,248 |
| Fondo | 4,496 | 2,248 |
| Puma | 828 | 413 |
| Lince | 2,776 | 1,387 |
| Venado | 4,344 | 2,171 |
| Coyote | 3,350 | 1,675 |
| Ave | 3,376 | 1,887 |

En el caso de la base de datos real, en los escenarios de clasificación binaria se utilizaban todas las imágenes que entraban al escenario, es decir, para el escenario del Borrego Cimarrón vs el Fondo, se utilizaron 517 imágenes de entrenamiento de la clase Borrego Cimarron, y 1,662 de la clase Fondo, para la prueba, fueron 258 y 830 respectivamente. En el escenario del Borrego Cimarrón vs Otros animales, de igual manera, se utilizaron todas las imágenes que caen en este escenario, es decir, 9,572 imágenes de entrenamiento de la clase Otros animales y 4,784 imágenes para probar, la clase del Borrego Cimarrón mantuvo la misma cantidad de imágenes que se explicó anteriormente. En el último escenario, para la clase Fondo + Otros animales, se utilizaron 11,234 imágenes para el entrenamiento y 5614 para probar, la clase del Borrego Cimarrón mantuvo la misma cantidad de imágenes.

En la Tabla 5.2 se muestran las particiones en las que se dividieron los datos para el experimento de la clasificación de múltiples clases utilizando la base de datos real, en donde las imágenes utilizan la mayor cantidad posible para cada especie animal.

Tabla 5.2. Número de imágenes usadas para el entrenamiento y prueba por cada animal en la base de datos real.

| Especies de animales | No. Imágenes usadas para: | |
|----------------------|---------------------------|--------|
| | Entrenar | Probar |
| Borrego Cimarrón | 517 | 258 |
| Fondo | 1,662 | 830 |
| Lince | 34 | 17 |
| Venado | 8 | 4 |
| Coyote | 1,548 | 773 |
| Ave | 7,982 | 3,990 |

La evaluación cruzada de bases de datos se realizó uniendo dos tercios de la base de datos sintética con un tercio de la base de datos real para el entrenamiento, por el lado de la prueba, se unió un tercio de la base de datos sintética con dos tercios de la base de datos real.

En el caso de la base de datos cruzada la idea es la misma que en caso de la base de datos real, en los escenarios de clasificación binaria se utilizaban todas las imágenes que entraban al escenario, es decir, para el escenario del Borrego cimarrón vs el Fondo, se utilizaron 4,754 imágenes de entrenamiento de la clase Borrego cimarrón, y 2,765 de la clase Fondo, para la prueba, fueron 5,326 y 3,910 respectivamente. En el escenario del Borrego Cimarrón vs Otros animales, de igual manera, se utilizaron todas las imágenes que caen en este escenario, es decir, 19,458 imágenes de entrenamiento de la clase Otros animales y 17,105 imágenes para probar, la clase del Borrego cimarrón mantuvo la misma cantidad de imágenes que se explicó anteriormente. En el último escenario, para la clase Fondo + Otros animales, se utilizaron 24,781 imágenes para el entrenamiento y 21,015 para probar, la clase del Borrego cimarrón mantuvo la misma cantidad de imágenes.

En la Tabla 5.3 se muestran las particiones de los datos para este escenario. En el entrenamiento, se utilizó dos tercios del total de imágenes de la base de datos

sintética más un tercio del total de imágenes de la base de datos real. En cambio, para probar el modelo, sólo se usaron dos tercios del total de imágenes de la base de datos real. Además, se experimentó entrenando modelos que sólo usaron imágenes de la base de datos sintética y se probaron usando sólo imágenes de la base de datos real.

Tabla 5.3. Número de imágenes usadas para el entrenamiento y prueba por cada animal en la base de datos cruzada.

| Especies de animales | No. Imágenes usadas para: | |
|----------------------|---------------------------|--------|
| | Entrenar | Probar |
| Borrego Cimarrón | 4,754 | 517 |
| Fondo | 5,326 | 1662 |
| Lince | 2,793 | 34 |
| Venado | 4,348 | 8 |
| Coyote | 4,123 | 1548 |
| Ave | 7,366 | 7,982 |

5.3. Resultados

En esta sección se habla sobre los resultados de los experimentos realizados en las diferentes bases de datos. Dichos experimentos fueron realizados en cada una de ellas, y fueron evaluados mediante los escenarios explicados en la sección 4.4. La sección está dividida en tres subsecciones. La subsección 5.3.1 habla de los resultados obtenidos en la experimentación sobre la base de datos sintética, la subsección 5.3.2 es para el caso de la base de datos real, y por último, la subsección 5.3.3 trata sobre la evaluación cruzada, la cual se propuso un escenario con el objetivo de mejorar la clasificación del problema antes mencionado, y también sirve para reducir el etiquetado manual de las imágenes.

5.3.1. Base de datos sintética

La Tabla 5.4 muestra los resultados obtenidos en los escenarios de clasificación binaria, en la cual, se reporta la métrica de exactitud. Los resultados muestran un rendimiento comparable alcanzado por cada algoritmo. Se considera que las características extraídas ayudan a modelar este tipo de escenarios sin problemas.

Tabla 5.4. Resultados obtenidos por los algoritmos de aprendizaje automático en los escenarios de clasificación binaria en la base de datos sintética.

| Clasificación binaria | Porcentajes de la exactitud | | | | | |
|--|-----------------------------|-------|-------|-------|-------|-------|
| | SVM | LL | ANN | KNN | NB | DT |
| Borrego Cimarrón vs Fondo | 99.95 | 99.93 | 99.88 | 98.35 | 96.90 | 99.15 |
| Borrego Cimarrón vs Otros animales | 96.64 | 95.86 | 95.10 | 93.57 | 89.05 | 87.05 |
| Borrego Cimarrón vs Fondo + Otros animales | 96.68 | 96.28 | 95.55 | 93.68 | 90.68 | 88.85 |

Como complemento al escenario de clasificación binario, se realizó un experimento donde hay tres clases, Borrego cimarrón, Fondo y Otros animales. Esto es con el objetivo de ver si es factible el dividir así las clases. La Tabla 5.5 muestra los resultados obtenidos del experimento previamente mencionado.

Tabla 5.5. Resultados obtenidos por los algoritmos de aprendizaje automático en la clasificación de tres clases en la base de datos sintética.

| Métricas | Algoritmos evaluados | | | | | |
|--------------|----------------------|--------------|------|------|------|------|
| | SVM | LL | ANN | KNN | NB | DT |
| Exactitud | 95.19 | 95.21 | 92.5 | 90.2 | 85.2 | 86.0 |
| Precisión | 95.2 | 95.2 | 92.5 | 90.6 | 85.7 | 86.0 |
| Sensibilidad | 95.2 | 95.2 | 92.5 | 90.3 | 85.3 | 86.0 |
| F1 | 95.2 | 95.2 | 92.5 | 90.1 | 85.4 | 86.0 |

Para complementar la Tabla 5.5, en la Figura 5.1 se muestran las matrices de

confusión de los mejores resultados. Se puede observar que los resultados aún se mantienen altos a pesar de que la complejidad subió un poco.

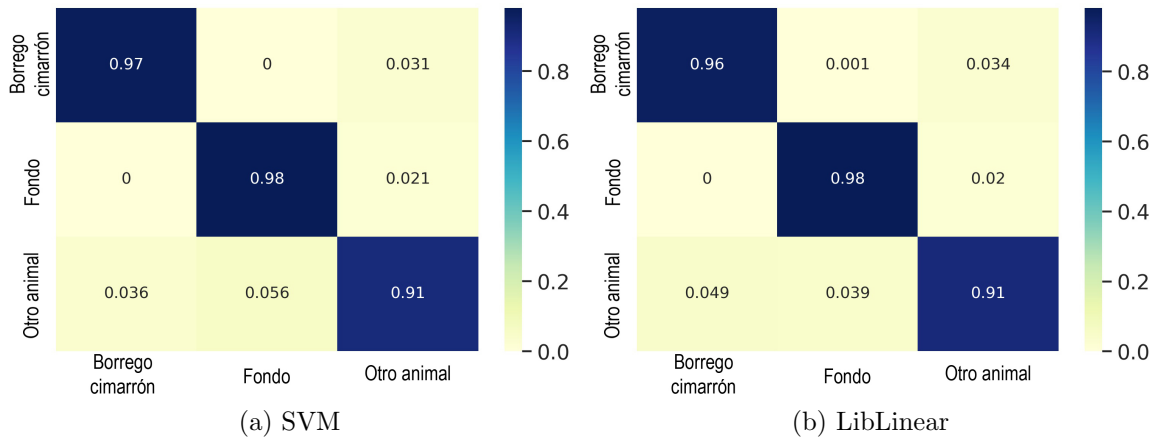


Figura 5.1. Matrices de confusión de los mejores resultados en la clasificación de tres clases en la base de datos sintética.

La Tabla 5.6 presenta los resultados obtenidos para el escenario de clasificación de múltiples clases considerando siete clases diferentes. Los mejores resultados alcanzados son cercanos al 90 %, y los más bajos se encuentran alrededor del 72 %.

Tabla 5.6. Resultados obtenidos por los algoritmos de aprendizaje automático en la clasificación de clases múltiples en la base de datos sintética.

| Métricas | Algoritmos evaluados | | | | | |
|--------------|----------------------|-------------|------|------|------|------|
| | SVM | LL | ANN | KNN | NB | DT |
| Exactitud | 88.9 | 89.7 | 86.3 | 86.0 | 72.0 | 75.9 |
| Precisión | 88.8 | 89.5 | 86.4 | 86.0 | 72.3 | 76.0 |
| Sensibilidad | 89.0 | 89.8 | 86.3 | 86.1 | 72.1 | 76.0 |
| F1 | 88.8 | 89.6 | 86.3 | 85.8 | 71.7 | 76.0 |

Para complementar la Tabla 5.6, en la Figura 5.2 se muestran las matrices de confusión de los mejores resultados. Se puede observar una disminución en el rendimiento debido al aumento de la complejidad. Sin embargo, se observa que la clasificación se conserva bien en la mayoría de las clases consideradas, especialmente en la clase del

borrego cimarrón. En cambio, la clase del puma es la menos favorecida, y también la clase minoritaria (ver Tabla 5.1).

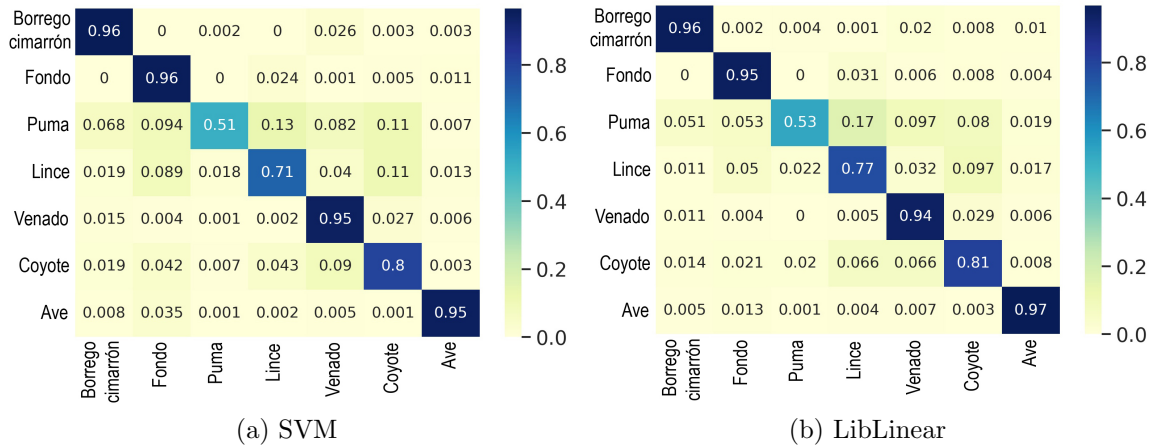


Figura 5.2. Matrices de confusión de los mejores resultados en la clasificación de múltiples clases en la base de datos sintética.

5.3.2. Base de datos real

La tabla 5.7 informa los resultados obtenidos en los escenarios de clasificación binaria. A partir de estos resultados, se puede observar un rendimiento competitivo y comparable alcanzado por cada algoritmo. Esto muestra un desempeño robusto

Tabla 5.7. Resultados obtenidos por los algoritmos de aprendizaje automático en los escenarios de clasificación binaria en la base de datos real.

| Clasificación binaria | Porcentajes de la exactitud | | | | | |
|--|-----------------------------|--------------|-------|-------|-------|-------|
| | SVM | LL | ANN | KNN | NB | DT |
| Borrego Cimarrón vs Fondo | 94.12 | 93.69 | 93.09 | 90.67 | 85.06 | 89.11 |
| Borrego Cimarrón vs Otros animales | 98.35 | 98.29 | 94.73 | 97.35 | 93.22 | 94.89 |
| Borrego Cimarrón vs Fondo + Otros animales | 98.39 | 98.55 | 93.97 | 97.56 | 93.52 | 96.44 |

y flexibilidad en el uso de características pre-entrenadas para identificar al borrego cimarrón.

La tabla 5.8 informa los resultados obtenidos cuando se consideran tres clases. A comparación de los resultados obtenidos a partir de los escenarios binarios, hay una clara disminución de más del 10% en el rendimiento cuando se trata de tres clases.

Tabla 5.8. Resultados obtenidos por los algoritmos de aprendizaje automático en la clasificación de tres clases en la base de datos real.

| Métricas | Algoritmos evaluados | | | | | |
|--------------|----------------------|-------------|------|------|------|------|
| | SVM | LL | ANN | KNN | NB | DT |
| Exactitud | 80.6 | 76.6 | 72.0 | 67.1 | 62.8 | 72.6 |
| Precisión | 75.8 | 76.2 | 73.5 | 71.8 | 77.4 | 73.6 |
| Sensibilidad | 80.6 | 76.7 | 72.1 | 67.1 | 62.9 | 72.6 |
| F1 | 77.4 | 76.4 | 72.7 | 69.2 | 66.9 | 73.1 |

Para complementar la Tabla 5.8, en la Figura 5.3 se muestran las matrices de confusión de los mejores resultados. Se puede observar que hay complicaciones entre las clases Fondo y Otros animales.

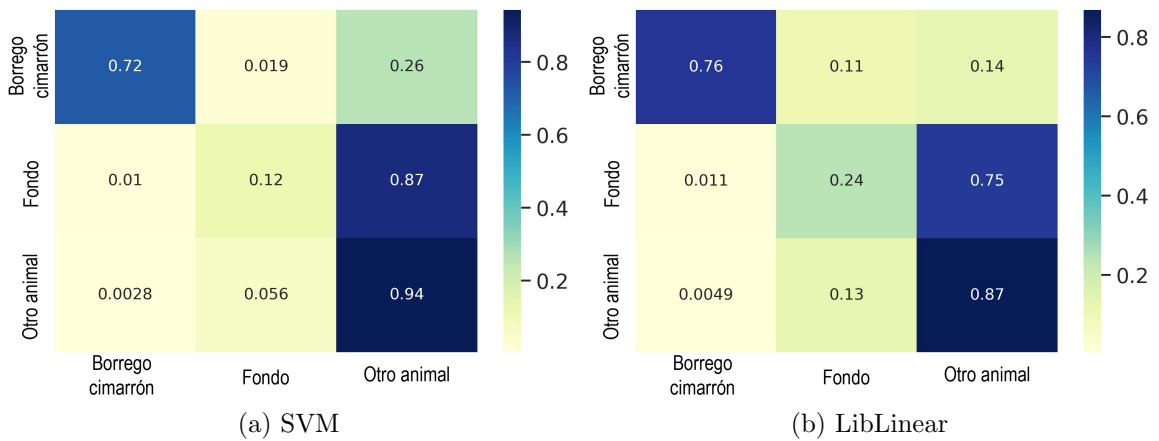


Figura 5.3. Matrices de confusión de los mejores resultados en la clasificación de tres clases en la base de datos real.

En la tabla 5.9 se informan los resultados en el escenario de clasificación de

múltiples clases utilizando la base de datos real. La mayoría de los resultados están alrededor del 85 %, sin embargo, existe un desequilibrio claro y severo entre las seis clases consideradas (ver Tabla 5.2) que afecta a algunos algoritmos. Por ejemplo, el modelo de SVM clasifica erróneamente ambas clases minoritarias, es decir, *Lince* y *Venado*.

Tabla 5.9. Resultados obtenidos por los algoritmos de aprendizaje automático en la clasificación de clases múltiples en la base de datos real.

| Métricas | Porcentajes de las métricas | | | | | |
|--------------|-----------------------------|--------------|-------|-------|-------|-------|
| | SVM | LL | ANN | KNN | NB | DT |
| Exactitud | 84.36 | 87.73 | 82.16 | 85.77 | 66.14 | 78.62 |
| Precisión | – | 87.30 | 83.80 | 85.80 | 78.70 | 78.60 |
| Sensibilidad | 84.40 | 87.70 | 82.20 | 85.70 | 66.10 | 78.60 |
| F1 | – | 87.50 | 82.90 | 85.70 | 69.10 | 78.60 |

La Figura 5.4 muestra las matrices de confusión de los mejores resultados de los algoritmos en la Tabla 5.9. El mejor rendimiento se alcanza con el modelo generado por LL, que muestra problemas de clasificación en las clases *Lince* y *Fondo*. También, usando el algoritmo KNN, se puede observar un rendimiento similar con problemas

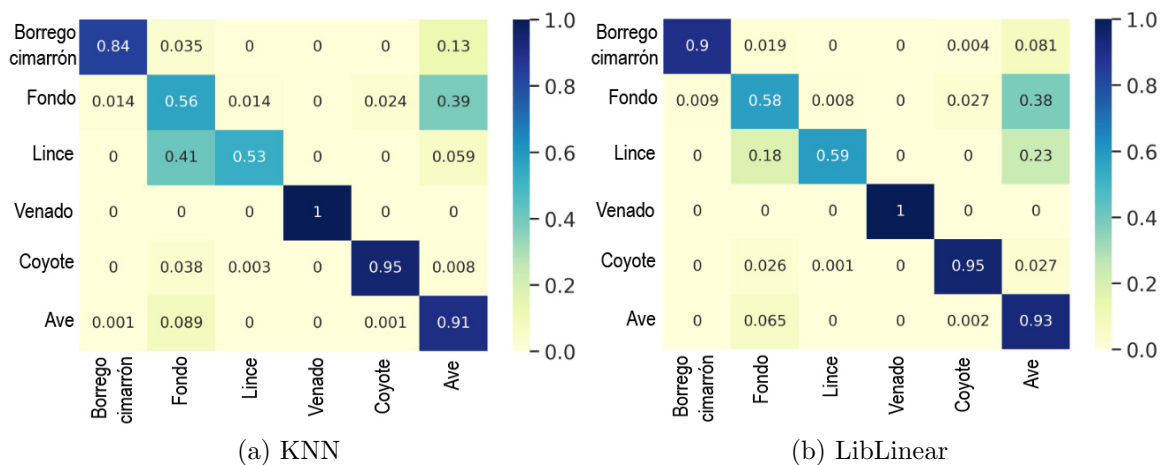


Figura 5.4. Matrices de confusión de los mejores resultados en la clasificación de múltiples clases en la base de datos real.

en las mismas clases. Recordando que la clase *Lince* es la segunda clase minoritaria con solo 51 muestras correspondientes a 0.2% en el conjunto de datos.

5.3.3. Base de datos cruzada

Para proporcionar alternativas cuando los datos de las cámaras trampa no están equilibrados o son escasos para una especie animal en particular, se experimentó con la combinación de datos sintéticos con datos reales de las cámaras trampa. Otra ventaja de adoptar esta alternativa es la reducción del etiquetado manual inicial de las imágenes.

El experimento inicial utiliza solo imágenes sintéticas para el entrenamiento y para probar utiliza imágenes de la base de datos real. Con esta configuración, se encontró una gran diferencia de fuentes que conducen a un bajo rendimiento. Se obtuvo un 58% de exactitud alcanzado con un modelo ANN, el cual fue el que tuvo el mejor rendimiento. Por lo tanto, se decidió agregar datos reales en los datos de entrenamiento.

La Tabla 5.10 muestra los resultados de la clasificación binaria en la propuesta de evaluación cruzada. Al comparar los resultados obtenidos con la Tabla 5.7, merece

Tabla 5.10. Resultados obtenidos por los algoritmos de aprendizaje automático en los escenarios de clasificación binaria en la base de datos cruzada.

| Clasificación binaria | Porcentajes de la exactitud | | | | | |
|---|-----------------------------|-------|-------|-------|-------|-------|
| | SVM | LL | ANN | KNN | NB | DT |
| Borrego Cimarrón vs Fondo | 96.97 | 96.55 | 96.55 | 94.86 | 95.50 | 95.06 |
| Borrego Cimarrón vs Otros animales | 98.15 | 97.56 | 96.09 | 96.91 | 90.97 | 94.28 |
| Borrego Cimarrón vs Fondo + Otros animales | 98.41 | 98.00 | 96.44 | 97.25 | 92.78 | 94.77 |

la pena mencionar que, el rendimiento muestra mejoría en el escenario de Borrego cimarrón vs Fondo. Además, es importante señalar que, aunque el número de muestras de entrenamiento ha aumentado, también lo ha hecho el número de muestras al probar, esta vez se evalúan más muestras de las imágenes captadas por cámaras trampa.

La Tabla 5.11 muestra los resultados de la clasificación de tres clases en la evaluación cruzada. Al comparar los resultados obtenidos con la Tabla 5.8, es necesario señalar que, el rendimiento se incrementa en todos los casos cuando se consideran tres clases. Además, al igual que en el caso del escenario binario, el número de muestras de entrenamiento ha aumentado al igual que el número de muestras al probar.

Tabla 5.11. Resultados obtenidos por los algoritmos de aprendizaje automático en la clasificación de tres clases en la base de datos cruzada.

| Métricas | Algoritmos evaluados | | | | | |
|--------------|----------------------|------|------|------|------|------|
| | SVM | LL | ANN | KNN | NB | DT |
| Exactitud | 88.0 | 86.2 | 83.3 | 81.3 | 68.1 | 80.6 |
| Precisión | 87.6 | 85.9 | 83.4 | 82.5 | 79.6 | 80.5 |
| Sensibilidad | 88.1 | 86.2 | 83.4 | 81.4 | 68.2 | 80.6 |
| F1 | 87.3 | 86.0 | 83.4 | 81.8 | 70.3 | 80.6 |

Complementando los resultados mostrados en la Tabla 5.11, en la Figura 5.5 se presentan las matrices de confusión del experimento cuando se consideran tres clases. Se puede observar que tanto a los modelos generados por SVM y LL les va bien con las clases *Borrego cimarrón* y *Otro animal*, pero, tiene algo de problemas con la clase *Fondo*.

La evaluación de la clasificación de múltiples clases se presenta en la Tabla 5.12. La mayoría de los resultados que se obtuvieron utilizando solo una fuente (ver Tabla 5.9) se han reducido un 10%. Sin embargo, es necesario indicar que se utilizaron el doble de muestras para el conjunto de prueba, y estos solo provienen de la base

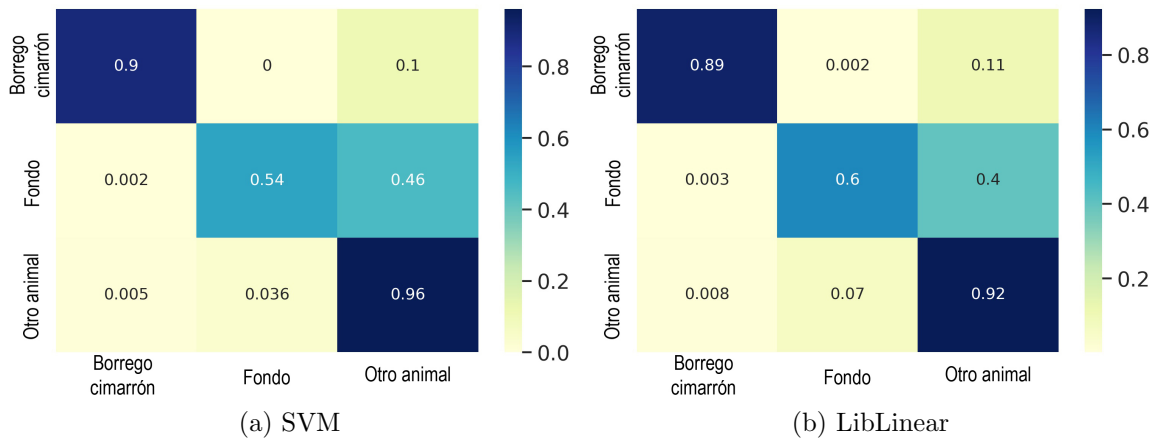


Figura 5.5. Matrices de confusión de los mejores resultados en la clasificación de tres clases en la base de datos cruzada.

de datos real. Los resultados obtenidos son suficientes para indicar que es posible aprovechar el uso de imágenes sintéticas en el entrenamiento de modelos para este caso. Haciendo de esta manera que el etiquetado manual de las imágenes llegue a bajar un 77.47 %.

Tabla 5.12. Resultados obtenidos por los algoritmos de aprendizaje automático en la clasificación de clases múltiples en la base de datos cruzada.

| Métricas | Porcentajes de las métricas | | | | | |
|--------------|-----------------------------|--------------|-------|-------|-------|-------|
| | SVM | LL | ANN | KNN | NB | DT |
| Exactitud | 77.47 | 74.78 | 73.95 | 66.92 | 62.21 | 64.19 |
| Precisión | 73.10 | 73.80 | 73.50 | 70.50 | 73.60 | 66.80 |
| Sensibilidad | 77.50 | 74.80 | 74.00 | 66.90 | 62.20 | 64.20 |
| F1 | 74.30 | 74.20 | 73.40 | 68.40 | 65.40 | 65.30 |

Complementando los resultados mostrados en la Tabla 5.12, en la Figura 5.6 se presentan las matrices de confusión del experimento. Se pueden ver los problemas enfrentados, el modelo generado por SVM está sobreajustado a las clases mayoritarias, y las clases como *Lince*, *Venado* y *Fondo* se ignoran. En cambio, el modelo de LL presenta resultados más equilibrados, pero hay un margen claro para mejoras.

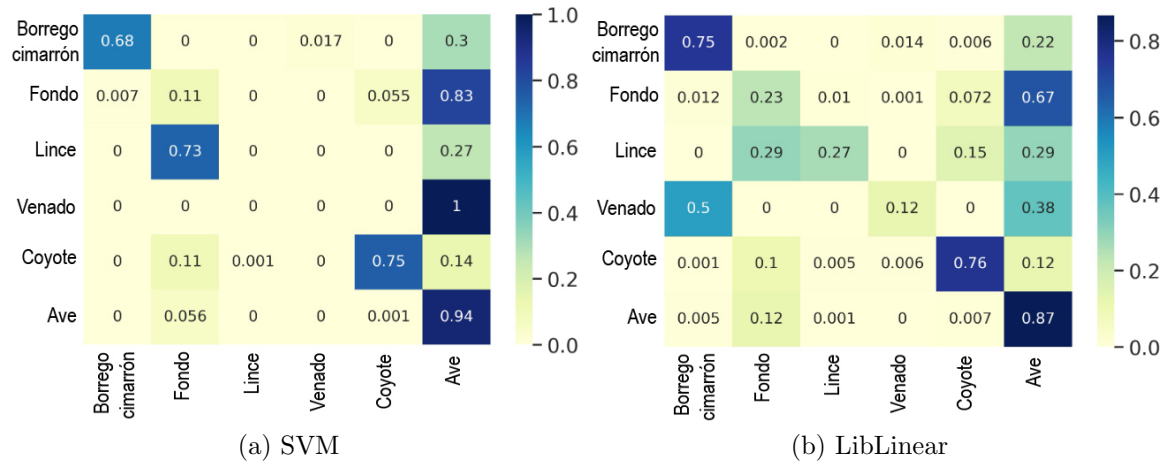


Figura 5.6. Matrices de confusión de los mejores resultados en la clasificación de múltiples clases en la base de datos cruzada.

En la Figura 5.7 se muestra una comparación de las predicciones obtenidas por diferentes algoritmos en la evaluación de la base de datos real y el escenario de evaluación cruzada. La sub-figura 1 muestra una escena difícil en la que aparentemente no se muestra nada, sin embargo, ambos escenarios presentan una clasificación incorrecta ya que un borrego cimarrón se ve parcialmente en el lado derecho. De igual manera, la sub-figura 2 muestra una escena difícil, pero en este caso, el modelo *LibLinear* entrenado y probado con la base de datos real logró clasificar correctamente la imagen. La sub-figura 3 muestra que, en algunos casos, al introducir ejemplos sintéticos, la predicción realizada por algunos modelos puede mejorarse y en otros no, como en el caso de SVM, donde pasa de un resultado correcto a uno fallido. Finalmente, la sub-figura 4 muestra un caso en el que ambos escenarios se clasifican correctamente.

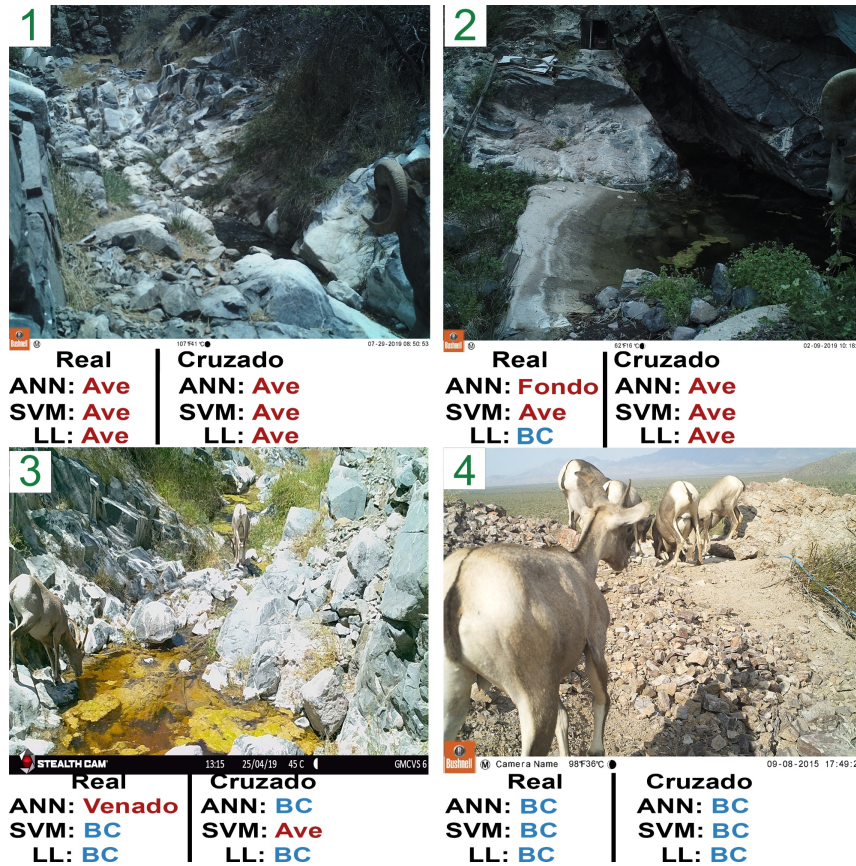


Figura 5.7. Comparación de los resultados entre modelos de la base de datos real y cruzada en el escenario de clasificación de múltiples clases. *BC = Borrego Cimarrón

5.4. Discusión

Los diferentes escenarios de clasificación binaria muestran que es posible reconocer correctamente al borrego cimarrón en datos sintéticos y reales. Aunque, todos los modelos en este escenario ofrecen resultados competitivos, se puede realizar la elección del algoritmo en función del tiempo necesario para la generación del modelo o el tiempo para realizar la clasificación. Por ejemplo, en una prueba simple del tiempo necesario para la generación del modelo cuando se entrena en el escenario binario de *Borego cimarrón vs Fondo + Otros animales*: LL usa 13.54 segundos, ANN 186.3 segundos y SVM 1,207.5 segundos. Mientras tanto, KNN no usa el tiempo en

la generación sino en la predicción.

En el caso del escenario de clasificación de tres clases, también se obtuvieron buenos resultados para el caso de datos sintéticos y reales para el reconocimiento del borrego cimarrón, pero, en el caso de la base de datos real, tiene complicaciones al diferenciar la clase *Fondo* de la clase *Otro animal*. Este problema aún se mantiene al utilizar la base de datos sintética, pero muestra una gran mejoría al diferenciar las clases.

En el escenario de clasificación de múltiples clases se muestra un buen rendimiento con una exactitud menor del 90 %. Algo que se puede observar es que los resultados logrados en la base de datos real al considerar este escenario son mejores con respecto a considerar tres clases. Curiosamente, ésta relación se invierte al usar la base de datos cruzada.

Experimentando con datos reales, en algunos casos, el desequilibrio de las muestras recolectadas conduce a sobreajustar la generación del modelo ignorando clases completamente minoritarias. Con respecto a esta preocupación, se presentó un experimento preliminar que involucra la mezcla de muestras de imágenes sintéticas y reales a través de una evaluación cruzada.

De lo anterior, se encontró que en el escenario de clasificación binaria es posible mantener un rendimiento competitivo. La clasificación de tres clases hace evidente que puede haber complicaciones al diferenciar clases, pero puede distinguir adecuadamente al borrego cimarrón. En cambio, la clasificación de múltiples clases muestra que es una posible forma de mitigar la necesidad del etiquetado manual.

A pesar de que en los estudios mencionados en la sección 3.2 utilizan bases de datos diferentes, se puede hacer una pequeña comparación de los resultados obtenidos. El ejemplo más cercano sería en (Yousif et al., 2019) donde realizan la tarea de

detección para tres clases y obtienen una exactitud del 93.556%. En este trabajo, para el experimento de tres clases, con la base de datos sintética se obtiene una mejor exactitud, la cual es del 95.21% (ver Tabla 5.5). Tanto para el caso de la base de datos real, y de la cruzada, la exactitud baja, siendo de 80.6% y de 88.0% respectivamente (ver Tablas 5.8-5.11). Regresando a los demás trabajos, se puede notar que varios de ellos tienen resultados por encima del 90%, para este caso, se estuvo cerca. Por ende, se puede decir que el método propuesto en este trabajo no va mal encaminado. Sería apropiado probar los métodos utilizados en dichos trabajos con las bases de datos empleadas en éste estudio para ver si se pueden obtener resultados más cercanos y experimentar para la realización de nuevos métodos.

Una curiosidad que se puede mencionar, es que, en el escenario de múltiples clases en la base de datos cruzada, el modelo generado por SVM mostraba un mejor rendimiento por medio de las métricas utilizadas, pero la matriz de confusión muestra el sobreajuste de dicho modelo hacía las clases mayoritarias, por lo que, no estaría mal el uso de los modelos generados por LL para llevar a cabo el etiquetado de imágenes en este escenario, dado que muestra un mayor equilibrio.

Capítulo 6

Conclusiones

A pesar de que el problema original es el reconocimiento automático del borrego cimarrón, se decidió durante el trabajo analizar la tarea de clasificación automática del borrego cimarrón y otras especies de animales a través de imágenes capturadas por medio de cámaras, haciendo de esta manera que el problema se hiciera más complejo. Aún así, se obtuvieron buenos resultados.

Durante el desarrollo de este trabajo se generaron diferentes modelos entrenados y probados en diferentes bases de datos para ser evaluados por medio de distintos escenarios para medir su rendimiento, y entre ellos se distinguieron aquellos que ofrecían mejores resultados en cada escenario.

El método propuesto es idóneo para el reconocimiento del borrego cimarrón y otras especies de animales relacionadas. Para alcanzar este fin, se recopilaron dos conjuntos de datos diferentes. Un conjunto de datos sintéticos con especies relacionadas compuesto por diversos conjuntos de datos y complementados por fotos recuperadas de fuentes Web. Y un conjunto de datos reales compilado por cámaras trampa ubicadas en un área protegida. Se evaluaron escenarios diversos, y a partir de este trabajo, se encontró lo siguiente:

- La clasificación binaria del borrego cimarrón permite aliviar el costo asociado a la inspección manual de las cámaras trampa. En este caso, el modelo aprende a caracterizar al borrego cimarrón, mientras que todo lo demás lo manda a la otra clase.
- La clasificación de tres clases también mostró que es posible el reconocimiento del borrego cimarrón, pero a su vez, demostró que es posible que los modelos generados tengan problemas al diferenciar clases.
- Se encontró plausible la clasificación de múltiples clases para analizar más en profundidad el comportamiento del borrego cimarrón coexistiendo con especies de animales diferentes que habitan en el área, sin alterar su ecosistema.
- Se demostró que el uso de imágenes controladas podría ser una forma de mitigar el desequilibrio de las muestras en conjuntos de datos reales. En este sentido, se proporcionó un estudio preliminar a través de una evaluación cruzada entre los dos conjuntos de datos usados. Sin embargo, haría falta un estudio adicional para lograr una transición fácil entre datos sintéticos y reales.

El resultado de todos estos experimentos ha puesto en evidencia que las características obtenidas a partir de una red pre-entrenada en aprendizaje profundo han mostrado ser eficientes para representar imágenes del borrego cimarrón. Haciendo de esta manera que la hipótesis (ver la sección 1.2) no sea rechazada.

Las cámaras trampa proporcionan una gran variedad de imágenes, lo cual, permite explorar la generalización de modelos obtenidos por medio de aprendizaje automático. Por ende, el método propuesto alienta a la introducción de enfoques parecidos para monitorear al borrego cimarrón y otras especies en su ecosistema, haciendo de esta manera posible el estudio de ésta especie sin alterar sus hábitos de compor-

tamiento.

El reconocimiento automatizado del borrego cimarrón que se describe en este trabajo puede ayudar en gran medida a futuras investigaciones que utilicen cámaras trampa y estén dedicadas a ésta especie o a las otras que también habitan en el Área de Protección de Flora y Fauna Valle de los Cirios, específicamente hablando en el Rancho Triple A. Haciendo de esto, el logro más significativo de este trabajo de tesis.

6.1. Trabajo a futuro

Para futuros trabajos, se podría explorar la efectividad del método a través de una extensión de este trabajo que incorpore ensamble de clasificadores y por medio de una votación se podría decidir la clase a la que pertenece la imagen. También se podría hacer una revisión de la fusión de características para una mejor clasificación de las especies de animales. Una idea adicional, podría ser la de aplicar un aumento en la cantidad de los datos recolectados por medio de las cámaras trampa al realizar una pequeña transformación a cada una de las imágenes, por ejemplo, que se le aplique un modo espejo, un acercamiento a la imagen, un giro de una cierta cantidad de grados, entre otras cosas. Esta técnica es llamada *data augmentation*. Dicha técnica funciona para que los modelos sean entrenados con más datos y nuevas perspectivas de las imágenes.

Otro enfoque en el que se podría trabajar, es en la tarea de detección. El cual consiste en conocer la ubicación exacta del elemento buscado en la imagen. La desventaja de esto, es que se tendría que hacer un etiquetado previo de todo el conjunto de datos, donde por medio de coordenada se indique en que parte se encuentra el animal en la imagen.

Apéndice A

A QUIEN CORRESPONDA

PRESENTE

Permiso del Valle de los Cirios

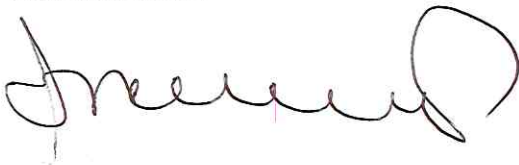
Asunto: Carta de apoyo

Por medio de la presente hacemos constar que se nos ha informado de la propuesta del proyecto "HACIA LA IDENTIFICACIÓN AUTOMÁTICA DEL BORREGO CIMARRÓN (*Ovis canadensis*) Y OTRAS ESPECIES PARA EL MANEJO Y CONSERVACIÓN DE VIDA SILVESTRE" por parte del Dr. Luis Pellegrin, adscrito a la Facultad de Ciencias de la Universidad Autónoma de Baja California y colaboradores.

Hemos analizado y encontrado que el desarrollo del proyecto representa un beneficio de nuestro interés para el proceso de monitorio y conservación del borrego cimarrón. Actualmente se busca promover la conservación de esta especie y su hábitat. Así como el desarrollo de prácticas que fomenten la generación de recursos económicos que favorezcan a los dueños del recurso y el incremento en la cultura de sustentabilidad ambiental para las comunidades rurales de Baja California.

Por tales motivos soportamos la pertinencia del desarrollo del proyecto, y expedimos la presente para los fines que se consideren pertinentes

ATENTAMENTE



BIOL. VÍCTOR GELASIO SÁNCHEZ SOTOMAYOR
DIRECTOR DEL APFF VALLE DE LOS CIRIOS

C.c.p. Archivo



Bibliografía

- [1] Ethem Alpaydin. *Introduction to Machine Learning*. The MIT Press, 2nd ed^{ón}., 2010. ISBN 026201243X.
- [2] Sara Beery. Caltech camera traps (cct). Available from <https://beerys.github.io/CaltechCameraTraps/>, Accessed [2019].
- [3] Sara Beery, Dan Morris, y Pietro Perona. The iwildcam 2019 challenge dataset. *arXiv preprint arXiv:1907.07617*, 2019.
- [4] Sara Beery, Grant Van Horn, y Pietro Perona. Recognition in terra incognita. En Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, y Yair Weiss, eds., *Computer Vision – ECCV 2018*, págs. 472–489. Springer International Publishing, Cham, 2018. ISBN 978-3-030-01270-0.
- [5] Y. Bengio, A. Courville, y P. Vicent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.
- [6] Patrick Buehler, Bill Carroll, Ashish Bhatia, Vivek Gupta, y Derek E. Lee. An automated program to find animals and crop photographs for individual recognition. *Ecological Informatics*, 50:191–196, 2019. doi:10.1016/j.ecoinf.2019.02.003.

- [7] Corinna Cortes y Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995. ISSN 0885-6125. doi:10.1023/A:1022627411411.
- [8] Secretaría de Protección al Ambiente. Estrategia estatal para la conservación y el manejo sustentable del borrego cimarrón (*ovis canadensis cremnobates*) en baja california. <http://www.spabc.gob.mx/wp-content/uploads/2017/10/ESTRATEGIA-ESTATAL-CONSERVACION-MANEJO-SUSTENTABLE-DEL-BORREGO-CIMARRON.-2012.pdf>, 2012.
- [9] Pedro Domingos. A few useful things to know about machine learning. *Commun. ACM*, 55(10):78–87, 2012. ISSN 0001-0782.
- [10] Jonathan G. Escobar-Flores, Sergio Álvarez Cárdenas, Raúl Valdez, Jorge Torres Rodríguez, Sara Díaz-Castro, Aradit Castellanos-Vera, y Roberto Martínez Gallardo. Detección de las preferencias de hábitat del borrego cimarrón (*Ovis canadensis*) en Baja California, mediante técnicas de teledetección satelital. *Therya*, 6:519 – 534, 2015. ISSN 2007-3364.
- [11] Greg Falzon, Christopher Lawson, Ka-Wai Cheung, Karl Vernes, Guy A. Ballard, Peter J. S. Fleming, Alistair S. Glen, Heath Milne, Atalya Mather-Zardain, y Paul D. Meek. Classifyme: A field-scouting software for the identification of wildlife in camera trap images. *Animals*, 10(1):58, 2019. ISSN 2076-2615. doi:10.3390/ani10010058.
- [12] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, y Chih-Jen Lin. Liblinear: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874, 2008. ISSN 1532-4435.
- [13] Peter Flach. Performance evaluation in machine learning: The good, the bad,

- the ugly, and the way forward. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:9808–9814, 2019. doi:10.1609/aaai.v33i01.33019808.
- [14] Flickr. Available from <https://www.flickr.com/>, Accessed [2020].
- [15] Kristen Grauman y Bastian Leibe. *Visual Object Recognition*. Morgan & Claypool Publishers, 1st ed^{ón}., 2011. ISBN 1598299689.
- [16] A. Guevara. Propuesta de modelo de aprovechamiento sustentable del borrego cimarrón (*ovis canadensis*) a través de la observación en baja california, 2017. http://www.fasol-ac.org/bc_17/propuesta-de-modelo-de-aprovechamiento-sustentable-del-borrego-cimarron-ovis-canadensis-a-traves-de-la-observacion-en-baja-california/.
- [17] Aldo A. Guevara-Carrizales, Gorgonio Ruiz-Campos, Jonathan Escobar-Flores, y Roberto Martínez-Gallardo. Mamíferos terrestres de las ecorregiones áridas del estado de baja california. En *Mamíferos de Baja California, México*, cap. 3, págs. 63–90. IB-UNAM, AMMAC & U. de Guanajuato, México, 2016.
- [18] Tianmei Guo, Jiwen Dong, Henjian Li, y Yunxing Gao. Simple convolutional neural network on image classification. En *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)*, págs. 721–724. 2017.
- [19] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, y Ian H. Witten. The weka data mining software: an update. *SIGKDD Explorations*, 11(1):10–18, 2009.
- [20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, y Jian Sun. Deep residual lear-

- ning for image recognition. En *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, págs. 770–778. 2016.
- [21] Samer L. Hijazi, Rishi Kumar, y Chris Rowen. Using convolutional neural networks for image recognition by. 2015. URL <https://ip.cadence.com/uploads/901/cnn-wp-pdf>.
- [22] iNaturalist.org. inaturalist research-grade observations. Available from <https://www.inaturalist.org>, Accessed [2019].
- [23] GS1 Ireland. Beef traceability case study. https://www.gs1si.org/Portals/0/GS1_Dokumentacija/GS1_Resitve/Sledljivost/BeefTraceabilityCaseStudy_Ireland.pdf, 2005.
- [24] Andrej Karpathy y Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. En *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015.
- [25] Matthias Körschens, Björn Barz, y Joachim Denzler. Towards automatic identification of elephants in the wild. En *ICEI 2018: 10th International Conference on Ecological Informatics- Translating Ecological Data into Knowledge and Decisions in a Rapidly Changing World*, tomo Poster. Jena, Germany, 2018. doi:10.22032/dbt.37903.
- [26] Alex Krizhevsky, Ilya Sutskever, y Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. En F. Pereira, C. J. C. Burges, L. Bottou, y K. Q. Weinberger, eds., *Advances in Neural Information Processing Systems 25*, págs. 1097–1105. Curran Associates, Inc., 2012.

- [27] G. Kumar y P. K. Bhatia. A detailed review of feature extraction in image processing systems. En *2014 Fourth International Conference on Advanced Computing & Communication Technologies*, págs. 5–12. 2014.
- [28] Hjalmar S. Kühl y Tilo Burghardt. Animal biometrics: quantifying and detecting phenotypic appearance. *Trends in Ecology & Evolution*, 28(7):432 – 441, 2013.
- [29] Donald C. Lay, Ted H. Friend, Ken K. Grissom, Cynthia L. Bowers, y Michael E. Mal. Effects of freeze or hot-iron branding of angus calves on some physiological and behavioral indicators of stress. *Applied Animal Behaviour Science*, 33(2):137 – 147, 1992.
- [30] Yann LeCun, Y. Bengio, y Geoffrey Hinton. Deep learning. *Nature*, 521:436–44, 2015.
- [31] Raymond Lee y Carlos H. Alcala-Galvan. Status of desert bighorn sheep in sonora, baja california, and baja california sur, mexico - 2013. En *Desert Bighorn Council Transactions*, tomo 52, págs. 40 – 43. 2013.
- [32] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004. ISSN 0920-5691. doi:10.1023/B:VISI.0000029664.99615.94.
- [33] C. Manterola-y-Piña. Censo aéreo de poblaciones de borrego cimarrón (*Ovis canadensis crembonates*) en Baja California, México. Inf. téc., Unidos para la Conservación A.C., 2000. URL <http://www.conabio.gob.mx/institucion/proyectos/resultados/InfU017.pdf>. SNIB-CONABIO Project Report No. U017.

- [34] L. Mao, M. Xie, Y. Huang, y Y. Zhang. Preceding vehicle detection using histograms of oriented gradients. En *2010 International Conference on Communications, Circuits and Systems (ICCCAS)*, págs. 354–358. 2010. doi: 10.1109/ICCCAS.2010.5581983.
- [35] Roberto Martínez-Gallardo, Jorge Alaniz-García, Aldo A. Guevara-Carrizales, y Salvador González-Guzmán. Monitoreo poblacional del borrego cimarrón (*ovis canadensis*) en baja california en el año 2010. En R. Eaton-González, A.A.G. Carrizales, J. Tapia-Mercado, G.R. Campos, y J.A. García, eds., *Estudios sobre el borrego cimarrón en el noroeste de México*, cap. 4. Universidad Autónoma de Baja California, 2017. ISBN 978-607-607-376-6. URL <https://libreriaabc.com/products/estudios-sobre-el-borrego-cimarron-en-el-noroeste-de-mexico>.
- [36] A. Miguel, S. Beery, E. Flores, L. Klemesrud, y R. Bayrakcismith. Finding areas of motion in camera trap images. En *2016 IEEE International Conference on Image Processing (ICIP)*, págs. 1334–1338. 2016.
- [37] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., USA, 1 ed^{ón}., 1997. ISBN 0070428077.
- [38] Mariana Montoya y Roberto Martínez-Gallardo. Evaluación del hábitat del borrego cimarrón (*ovis canadensis*) en la sierra san pedro mártir, baja california, méxico. En R. Eaton-González, A.A.G. Carrizales, J. Tapia-Mercado, G.R. Campos, y J.A. García, eds., *Estudios sobre el borrego cimarrón en el noroeste de México*, cap. 5. Universidad Autónoma de Baja California (UABC), 2017. ISBN 978-607-607-376-6. URL <https://libreriaabc.com/collections/ciencias-naturales/>.

- [39] Kevin Patrick Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012. ISBN 0262018020.
- [40] Glenn J. Myatt. *Making Sense of Data: A Practical Guide to Exploratory Data Analysis and Data Mining*. Wiley-Interscience, USA, 1st ed^{ón}., 2006. ISBN 047007471X.
- [41] Naturalista. CONABIO Available from <http://www.naturalista.mx>, Accessed [2020].
- [42] Mohammad Sadegh Norouzzadeh, Anh Nguyen, Margaret Kosmala, Alexandra Swanson, Meredith S. Palmer, Craig Packer, y Jeff Clune. Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. *Proceedings of the National Academy of Sciences*, 115(25):E5716–E5725, 2018. ISSN 0027-8424. doi:10.1073/pnas.1719367115.
- [43] T. Ojala, M. Pietikainen, y D. Harwood. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. En *Proceedings of 12th International Conference on Pattern Recognition*, tomo 1, págs. 582–585. IEEE Computer Society, 1994.
- [44] Matti Pietikäinen. Image analysis with local binary patterns. En Heikki Kalviainen, Jussi Parkkinen, y Arto Kaarna, eds., *Image Analysis*, págs. 115–118. Springer Berlin Heidelberg, 2005. ISBN 978-3-540-31566-7.
- [45] J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993. ISBN 1558602380.
- [46] José Antonio Ramos. *Generación de algoritmos para clasificación de obstáculos*.

- Trabajo de fin de grado, Universidad Carlos III de Madrid. Departamento de Ingeniería de Sistemas y Automática, 2016.
- [47] Joseph Redmon, Santosh Divvala, Ross Girshick, y Ali Farhadi. You only look once: Unified, real-time object detection. En *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.
- [48] Shaoqing Ren, Kaiming He, Ross Girshick, y Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. En C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, y R. Garnett, eds., *Advances in Neural Information Processing Systems 28*, págs. 91–99. Curran Associates, Inc., 2015.
- [49] Hector I. Reyes-Moncayo, Ana B Vacca-Casanova, y Agustín Gongora-Orjuela. Diseño de un sistema de identificación electrónica y su potencial uso en la trazabilidad de la carne bovina. *ORINOQUIA*, 15:90 – 98, 2011.
- [50] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, págs. 65–386, 1958.
- [51] Enrique de Jesús Ruiz-Mondragón, Guillermo Romero-Figueroa, Mario Alberto García-Aranda, Eloy Alejandro Lozano-Cavazos, y Raúl Valdez. Potential distribution model of *ovis canadensis* in northern baja california, mexico. *Therya*, 9:219–226, 2018. ISSN 2007-3364. doi:10.12933/therya-18-571.
- [52] Saleh Shahinfar, Paul Meek, y Greg Falzon. 'how many images do i need?' understanding how sample size per class affects deep learning model performance metrics for balanced designs in autonomous wildlife monitoring. *Ecological Informatics*, 57:101085, 2020. ISSN 1574-9541. doi:https://doi.org/10.1016/j.ecoinf.2020.101085.

- [53] Caifeng Shan, Shaogang Gong, y Peter W. McOwan. Facial expression recognition based on local binary patterns: A comprehensive study. *Image and Vision Computing*, 27(6):803 – 816, 2009. ISSN 0262-8856. doi:<https://doi.org/10.1016/j.imavis.2008.08.005>.
- [54] Sandeep Sharma, Yadvendradev Jhala, y Vishwas B. Sawarkar. Identification of individual tigers (*panthera tigris*) from their pugmarks. *Journal of Zoology*, 267(1):9–18, 2005.
- [55] B. W. Silverman y M. C. Jones. E. fix and j.l. hodges (1951): An important contribution to nonparametric discriminant analysis and density estimation: Commentary on fix and hodges (1951). *International Statistical Review / Revue Internationale de Statistique*, 57(3):233–238, 1989. ISSN 03067734, 17515823. URL <http://www.jstor.org/stable/1403796>.
- [56] Karen Simonyan y Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. En *International Conference on Learning Representations*. 2015.
- [57] AB Swanson, M Kosmala, CJ Lintott, RJ Simpson, A Smith, y C Packer. Snapshot serengeti, high-frequency annotated camera trap images of 40 mammalian species in an african savanna. *Scientific Data*, 2, 2015. doi:<https://doi.org/10.1038/sdata.2015.26>.
- [58] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, y Z. Wojna. Rethinking the inception architecture for computer vision. En *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, págs. 2818–2826. 2016. ISSN 1063-6919. doi:10.1109/CVPR.2016.308.

- [59] Pang-Ning Tan, Michael Steinbach, y Vipin Kumar. *Introduction to Data Mining*. Addison-Wesley Longman Publishing Co., Inc., USA, 2005. ISBN 0321321367.
- [60] Tensorflow. Large-scale machine learning on heterogeneous systems. 2015. URL <http://tensorflow.org/>. Software available from tensorflow.org.
- [61] Mikayla Timm, Subhransu Maji, y Todd Fuller. Large-scale ecological analyses of animals in the wild using computer vision. En *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2018.
- [62] Fay E. Ward. En *The Cowboy at Work: All About His Job and How He Does It*. Hastings House, 1958.
- [63] Jason Yosinski, Jeff Clune, Yoshua Bengio, y Hod Lipson. How transferable are features in deep neural networks? En *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS'14*, pág. 3320–3328. MIT Press, Cambridge, MA, USA, 2014.
- [64] Hayder Yousif, Jianhe Yuan, Roland Kays, y Zhihai He. Animal scanner: Software for classifying humans, animals, and empty frames in camera trap images. *Ecology and Evolution*, 9(4):1578–1589, 2019. doi:10.1002/ece3.4747.
- [65] Xiaoyuan Yu, Jiangping Wang, Roland Kays, Patrick A. Jansen, Tianjiang Wang, y Thomas Huang. Automated identification of animal species in camera trap images. *EURASIP Journal on Image and Video Processing*, 2013(1):52, 2013. ISSN 1687-5281. doi:10.1186/1687-5281-2013-52.