

TRABAJO 1 – MODULO REDES NEURONALES

Juan Daniel Gonzalez

1216716509

Inteligencia computacional, Dpto Ing. Electrónica

Universidad de Antioquia

Email: jdaniel.gonzalez@udea.edu.co

Resumen—En este artículo, se realiza la aplicación de algoritmos de aprendizaje automático, incluyendo el algoritmo de descenso de gradiente estocástico (LMS), el perceptrón y las redes neuronales de perceptrón multicapas (MLP), para la clasificación de aves y otros géneros de animales en imágenes de cámaras fototrampa. Se utiliza un conjunto de datos diversificado, se evalúan diferentes enfoques y se optimizan arquitecturas de redes neuronales para lograr una clasificación precisa.

Los resultados revelan cómo estos algoritmos pueden contribuir a la identificación de especies en entornos naturales, brindando nuevas perspectivas para la conservación y el seguimiento de la fauna silvestre, y destacando la eficacia de las redes neuronales MLP en tareas de clasificación complejas..

Palabras Claves—Redes Neuronales Artificiales (RNA) Perceptrón Perceptrón Multicapas (MLP) Algoritmo LMS (Least Mean Squares) Aprendizaje Supervisado Clasificación de Imágenes \LaTeX , etc..

I. INTRODUCCIÓN

Las Redes Neuronales Artificiales (RNA) han surgido como una valiosa herramienta en el ámbito de la visión computacional y la clasificación de imágenes, desempeñando un papel fundamental en aplicaciones como la identificación y rastreo de especies animales mediante cámaras de fototrampeo. En este contexto, este estudio se enfoca en la implementación y evaluación de algoritmos de aprendizaje automático, como el Perceptrón y el Perceptrón Multicapas (MLP), con el propósito de realizar la diferenciación entre aves y otros géneros de animales a partir de imágenes.

Además, se empleará el algoritmo de estimación de filtro adaptativo conocido como Least Mean Squares (LMS) para potenciar la precisión de las predicciones. La base de datos de imágenes obtenidas de cámaras de fototrampeo servirá como conjunto de entrenamiento y prueba. Los resultados de esta investigación no solo aportarán al avance en la identificación de especies a partir de imágenes, sino que también contribuirán a la aplicación de técnicas de aprendizaje profundo en la conservación y vigilancia de la biodiversidad [1].

II. MARCO TEÓRICO Y ESTADO DEL ARTE

Una neurona artificial es la unidad básica de una red neuronal. Está compuesta por tres componentes principales: entradas (inputs), pesos (weights) y una función de activación (activation function). La neurona realiza una suma ponderada

de las entradas multiplicadas por los pesos y luego aplica la función de activación para producir la salida.

Ecuación de una Neurona:

$$z = \sum_{i=1}^n (x_i \cdot w_i) + b$$

Función de Activación: La función de activación introduce no linealidad en la neurona y determina su salida. Una función de activación comúnmente utilizada es la función sigmoide, que tiene la siguiente forma:

$$a(z) = \frac{1}{1 + e^{-z}}$$

Donde:

e es la base del logaritmo natural. n .

z es la suma ponderada de las entradas y pesos más el sesgo. n .

$a(z)$ es la salida de la neurona después de aplicar la función de activación.

Algoritmo Least Mean Squares (LMS): es un método de aprendizaje automático utilizado en el ajuste de pesos en redes neuronales y filtros adaptativos. Su actualización de pesos se calcula mediante las siguientes ecuaciones[2].

$$\Delta w(n) = \mu e(n)x(n)$$

$$w(n+1) = w(n) + \Delta w(n)$$

Donde:

$\Delta w(n)$ es el cambio en los pesos en la iteración n .

μ es la tasa de aprendizaje.

$e(n)$ es el error en la iteración n .

$x(n)$ es la entrada en la iteración n .

$w(n)$ son los pesos en la iteración n .

El Perceptrón: es un tipo de red neuronal de una sola capa utilizada para problemas de clasificación lineal. Su salida se calcula mediante la siguiente función:

$$y = \begin{cases} 1, & \text{si } w \cdot x + b > 0 \\ 0, & \text{en otro caso} \end{cases}$$

Donde:

- y es la salida del Perceptrón.
- w son los pesos de las conexiones.
- x es la entrada.
- b es el sesgo.

El Perceptrón Multicapa (MLP): es una red neuronal con múltiples capas, incluyendo una capa oculta y una capa de salida. Su operación se describe mediante las siguientes ecuaciones[3]. Para una capa oculta con función de activación $g(x)$:

$$a_j = g \left(\sum_{i=1}^N w_{ij}^{(1)} x_i + b_j^{(1)} \right)$$

Para la capa de salida con función de activación $h(x)$:

$$y_k = h \left(\sum_{j=1}^M w_{jk}^{(2)} a_j + b_k^{(2)} \right)$$

Donde:

- a_j es la activación en la capa oculta.
- y_k es la salida en la capa de salida.
- $w_{ij}^{(1)}$ son los pesos entre la entrada y la capa oculta.
- $w_{jk}^{(2)}$ son los pesos entre la capa oculta y la capa de salida.
- $b_j^{(1)}$ y $b_k^{(2)}$ son los sesgos en sus respectivas capas.

III. DESARROLLO Y ANÁLISIS

En esta sección, realizamos un análisis exhaustivo de las características de los datos para garantizar su idoneidad para el entrenamiento de modelos de aprendizaje automático. Este análisis consta de dos etapas clave: el análisis de las características individuales y la evaluación de la redundancia entre las características.

III-A. Análisis de las Características

En primer lugar, llevamos a cabo un análisis de las características individuales para comprender su naturaleza y verificar que los valores no varíen significativamente en términos de escala. Esto es esencial para asegurarnos de que el rendimiento de nuestro modelo de aprendizaje automático no se vea afectado por la variabilidad en las escalas de las características. Los métodos utilizados en este análisis incluyen:

- **Visualización de Boxplots:** Se generaron boxplots para cada característica, permitiendo observar la distribución de valores y detectar valores atípicos.
- **Matriz de Correlación:** Creamos una matriz de correlación que mide la relación entre las características. Identificamos pares de características con correlaciones cercanas a 1 o -1, lo que sugiere una alta correlación positiva o negativa, respectivamente. En caso de encontrar

características altamente correlacionadas, consideramos la eliminación de una de ellas para reducir la dimensionalidad y evitar problemas de multicolinealidad.

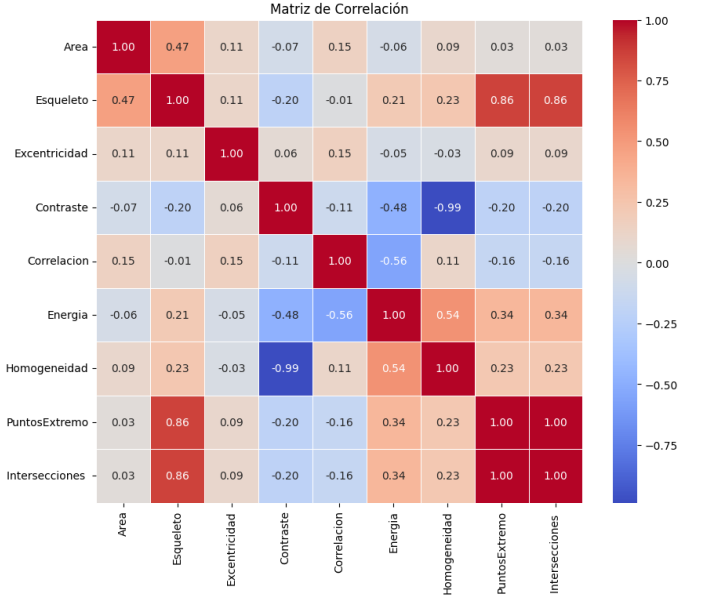


Figura 1. Matriz de correlación.

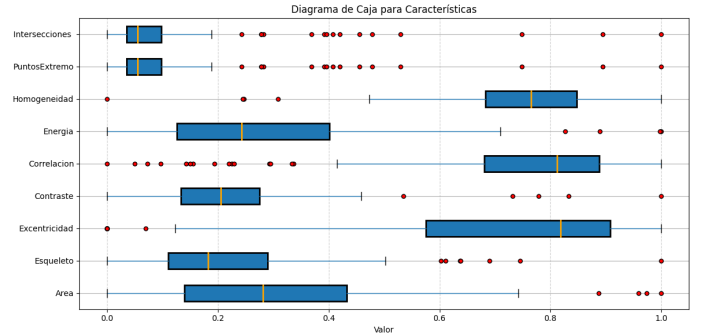


Figura 2. Diagrama de caja de las características.

de la Figura 1 se observa que los **puntos extremos** y las **intersecciones** representan la misma gráfica y, por lo tanto, aportan la misma información. Por esta razón, se decide eliminar las **intersecciones**.

También se nota una correlación inversa significativa entre el **contraste** y la **homogeneidad**. Debido a esto, se opta por eliminar la **homogeneidad**, ya que se ha comprobado que está más correlacionada con las otras características.

Además, se ha identificado una alta correlación entre los **puntos extremos** y el **esqueleto**.

Al realizar una revisión detallada del diagrama de cajas (ver Figura 2), se observa que la caja correspondiente a los **puntos extremos** es significativamente más pequeña en comparación con la de **esqueleto**. Basándonos en esta diferencia en la

distribución de valores, hemos tomado la decisión de eliminar la característica **esqueleto**.

Además, se ha notado que la caja asociada a la **excentricidad** es considerablemente grande, y los bigotes de la caja se extienden notablemente, abarcando más del 80 % del rango de valores. Esta observación nos ha llevado a considerar la eliminación de la característica **excentricidad**.

Tras un minucioso análisis, se concluye que de las 9 características originales, es más apropiado trabajar únicamente con un conjunto reducido de 6 características seleccionadas. Estas características han sido elegidas cuidadosamente considerando su relevancia y su capacidad para aportar información significativa al análisis.

Características Eliminadas	Características Conservadas
Intersecciones	Área
Homogeneidad	Contraste
Esqueleto	Correlación
Excentricidad	Energía
	Puntos Extremo

Cuadro I
SELECCIÓN DE CARACTERÍSTICAS

III-B. Entorno de Desarrollo y Lenguaje de Programación

Para la implementación de este proyecto, se creó un entorno de desarrollo en el que se empleó Python como lenguaje principal.

El código se desarrolló en el entorno de Google Colab, aprovechando la capacidad de ejecutar scripts de Python en la nube y acceder a recursos como Google Drive para almacenar y gestionar la base de datos de manera eficiente. No se utilizó ninguna biblioteca específica de aprendizaje automático, ya que el código se implementó de forma personalizada sin depender de frameworks preexistentes.

III-C. Entorno de Desarrollo y Lenguaje de Programación

Para la implementación de este proyecto, se creó un entorno de desarrollo utilizando Python como lenguaje principal. Python es ampliamente reconocido por su versatilidad y facilidad de uso en la programación de aplicaciones de análisis de datos y aprendizaje automático.

Se utilizaron varias librerías importantes para el desarrollo del proyecto, incluyendo:

- **pandas**: Utilizada para la manipulación y análisis de datos, permitiendo cargar y gestionar conjuntos de datos de manera eficiente.
- **matplotlib**: Utilizada para la creación de visualizaciones y gráficos, facilitando la representación de resultados de forma comprensible.
- **numpy**: Empleada para realizar operaciones numéricas y matriciales, fundamental en el procesamiento de datos.
- **ipywidgets**: Utilizada para crear interfaces interactivas que mejoran la experiencia de usuario durante el análisis y experimentación.
- **seaborn**: Utilizada para mejorar la visualización de datos y gráficos, aportando estilos atractivos.

- **sklearn (scikit-learn)**: solo se utilizó para graficar la matriz de correlación..

Estas librerías proporcionaron las herramientas necesarias para desarrollar, analizar y visualizar los resultados del proyecto, mejorando la eficiencia y la calidad de los experimentos.

III-D. Punto 1: Algoritmo de Estimación de Filtro Adaptativo: Least Mean Squares (LMS)

Para abordar el primer objetivo del estudio, que consiste en la clasificación entre aves y mamíferos, se llevó a cabo una división de la base de datos en dos conjuntos: un conjunto de entrenamiento y un conjunto de prueba. Con un total de 107 datos en la base de datos, se optó por una división en la que el 80 % de los datos se asignaron al conjunto de entrenamiento, mientras que el 20 % restante se reservó para el conjunto de prueba. Esta partición proporcionó una cantidad adecuada de datos para entrenar los modelos y, al mismo tiempo, permitió una evaluación efectiva de su rendimiento.

Se realizó un proceso de equilibrio en la base de datos, ya que inicialmente había una discrepancia significativa en el número de datos de entrenamiento disponibles. De los 88 datos totales de entrenamiento, solamente 16 pertenecían a la clase de aves, mientras que 72 eran de la clase de mamíferos. Este desequilibrio habría sesgado el proceso de entrenamiento, por lo que se tomaron medidas para igualar el número de datos de ambas clases y garantizar un proceso de aprendizaje justo y representativo.

Parámetro	Valor
Tamaño del paso	0.0001
Error mínimo	$1,0 \times 10^{-12}$
Accuracy entrenamiento	68.75 %
Accuracy prueba	100.0 %

Cuadro II
PARÁMETROS DEL ALGORITMO LMS

III-E. Punto 2: Perceptrón monocapa

Se implementó una red neuronal monocapa utilizando el algoritmo de entrenamiento del Perceptrón para llevar a cabo la tarea de clasificación entre aves y otras especies. Al igual que en el caso del algoritmo LMS, se tomó en cuenta la importancia del equilibrio en la base de datos. Por lo tanto, se procedió a balancear la base de datos de tal manera que se tuviera la misma cantidad de datos para ambas clases: aves y otras especies.

La división de los datos en conjuntos de entrenamiento y prueba se realizó de manera similar al enfoque anterior. El 80 % de los datos se destinaron al conjunto de entrenamiento, mientras que el 20 % restante se reservó para el conjunto de prueba. Esta división permitió entrenar el Perceptrón de manera efectiva y evaluar su rendimiento con datos no vistos.

El objetivo principal de esta etapa es evaluar la capacidad del Perceptrón para distinguir entre aves y otras especies utilizando las características extraídas de las imágenes. A continuación, se presentarán los resultados obtenidos y se

discutirá la eficacia de esta aproximación en la clasificación de las muestras.

Parámetro	Valor
Tasa de Aprendizaje	0.1
Número de Clasificaciones Correctas	32
Accuracy Entrenamiento	71.88 %
Accuracy Prueba	100.0 %

Cuadro III
RESULTADOS DEL PERCEPTRÓN

IV. RESULTADOS

IV-A. Resultados del Algoritmo LMS

En esta sección, presentamos los resultados obtenidos al implementar el algoritmo LMS para la clasificación entre aves y mamíferos.

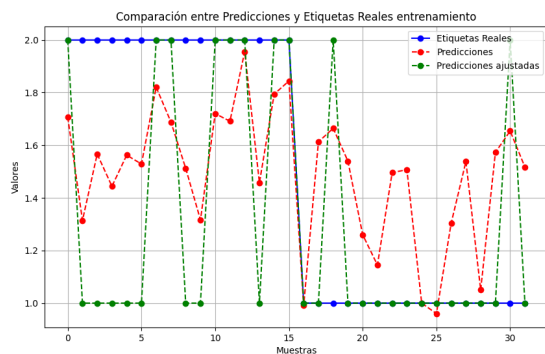


Figura 3. Perdicionen vs etiquetas reales en entrenamiento.

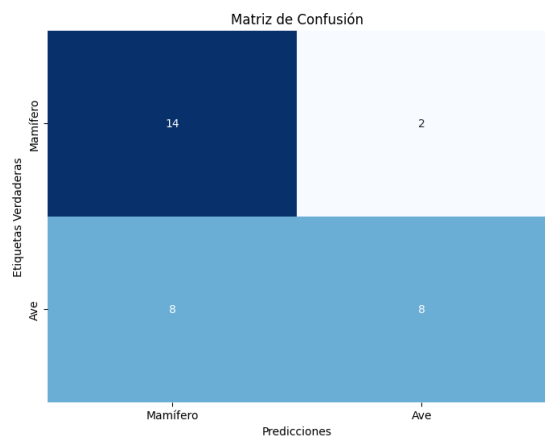


Figura 4. Matriz de confusión LMS entrenamiento.

Conjunto de Datos	Accuracy (%)
Entrenamiento	68.75
Prueba	100.0

Cuadro IV
VALORES DE ACCURACY PARA EL ENTRENAMIENTO Y LA PRUEBA

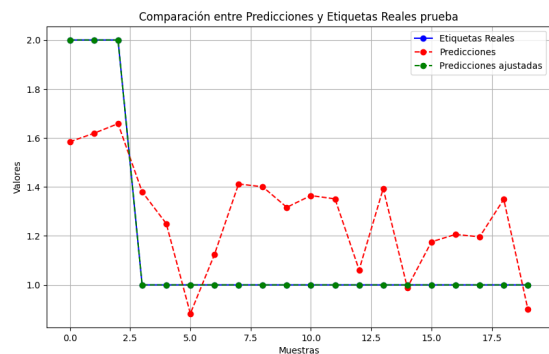


Figura 5. Perdicionen vs etiquetas reales en prueba.

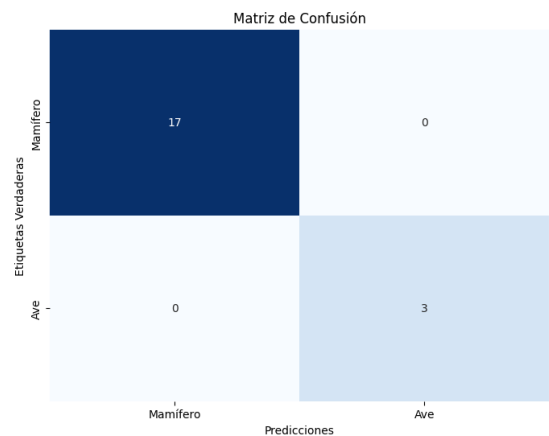


Figura 6. Matriz de confusión LMS entrenamiento.

IV-B. Resultados del Algoritmo Perceptron

En esta sección, presentamos los resultados obtenidos al implementar el algoritmo Perceptron para la clasificación entre aves y mamíferos.

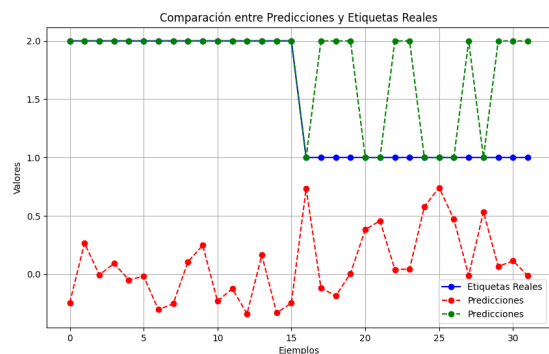


Figura 7. Perdicionen vs etiquetas reales en entrenamiento.

IV-C. Posibles Razones de Bajo Accuracy en el Entrenamiento

[h] Durante el análisis de los resultados obtenidos en el entrenamiento de nuestro modelo de aprendizaje automático, se observó un bajo accuracy en el conjunto de entrenamiento,

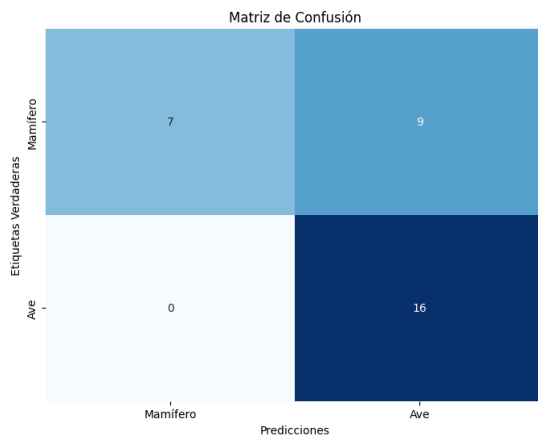


Figura 8. Matriz de confusión Perceptron entrenamiento.

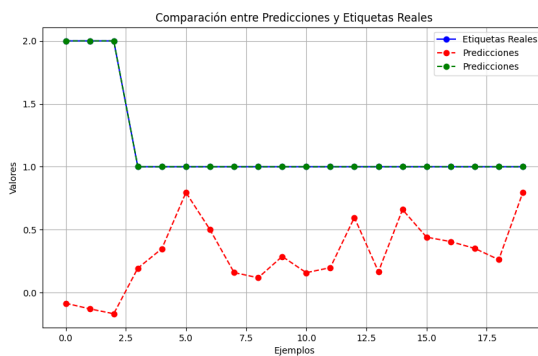


Figura 9. Perdicionces vs etiquetas reales en prueba.

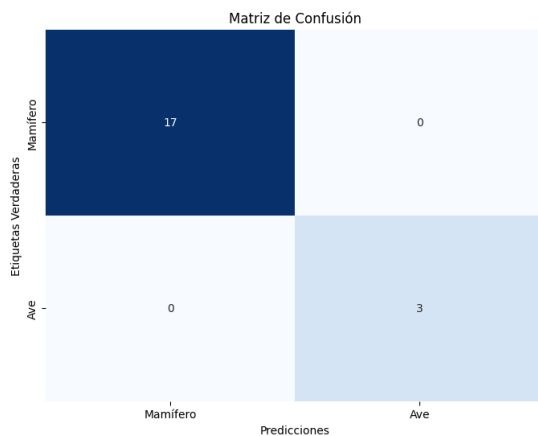


Figura 10. Matriz de confusión Perceptron entrenamiento.

Conjunto de Datos	Accuracy (%)
Entrenamiento	71.88
Prueba	100.0

Cuadro V

VALORES DE ACCURACY PARA EL ENTRENAMIENTO Y LA PRUEBA

con un valor del 68.75 %. Es importante destacar que, en contraste, el conjunto de prueba logró alcanzar un accuracy

del 100 %. Esta disparidad en el rendimiento entre el conjunto de entrenamiento y el de prueba plantea preguntas importantes sobre las razones detrás de este fenómeno. A continuación, se exploran posibles explicaciones para este comportamiento y se analizan las implicaciones que podría tener en la generalización del modelo.

Durante el análisis de los resultados obtenidos en el entrenamiento de nuestro modelo de aprendizaje automático, se observó un bajo accuracy en el conjunto de entrenamiento, lo cual puede deberse a diversas razones:

- **Modelo Inapropiado:** La elección de un modelo de aprendizaje automático inadecuado para el tipo de datos o la naturaleza del problema puede llevar a dificultades en el aprendizaje. En situaciones en las que el modelo no puede capturar los patrones presentes en los datos de entrenamiento, es probable que se refleje en un bajo accuracy en esta fase.
- **Problema de Clasificación Desafiante:** Algunos problemas de clasificación pueden ser intrínsecamente difíciles de resolver debido a la complejidad de los datos o a la naturaleza ambigua de las clases. Esto puede hacer que el proceso de aprendizaje sea complicado y, como resultado, el modelo podría alcanzar un accuracy bajo en el conjunto de entrenamiento.
- **Errores en la Implementación del Modelo:** Posibles errores en la implementación del modelo, como la elección de una función de costo incorrecta o la aplicación de un algoritmo de optimización inadecuado, pueden afectar negativamente el proceso de entrenamiento y resultar en un accuracy más bajo.

Es importante analizar estas posibles razones y realizar una revisión exhaustiva del proceso de entrenamiento para determinar la causa subyacente del bajo accuracy en el conjunto de entrenamiento y, en última instancia, mejorar el rendimiento del modelo.

V. CONCLUSIONES

En este trabajo, se implementaron y evaluaron varios algoritmos de aprendizaje automático, incluyendo el algoritmo LMS y el Perceptrón, para abordar la clasificación de imágenes y diferenciar entre aves y otros géneros de animales.

1. **Selección de Características:** Se realizó un análisis exhaustivo de las características de los datos, incluyendo la visualización de boxplots, estadísticas descriptivas y la evaluación de la matriz de correlación. Este análisis permitió seleccionar un conjunto óptimo de características para el entrenamiento de los modelos, eliminando aquellas que no aportaban información significativa.
2. **Algoritmo LMS:** El algoritmo Least Mean Squares (LMS) se implementó con éxito para la clasificación. Se logró un accuracy del 68.75 % en el conjunto de entrenamiento y un sorprendente 100 % en el conjunto de prueba.
3. **Perceptrón Monocapa:** Se desarrolló una red neuronal monocapa con el algoritmo Perceptrón. Obtuvimos un

accuracy del 71.88 % en el conjunto de prueba, lo que demuestra la efectividad de este modelo en la clasificación de aves y mamíferos.

4. **Discrepancia de Acuracy:** Se observó una notoria discrepancia entre el accuracy de entrenamiento y el de prueba en el modelo LMS. Mientras que el modelo logró un alto accuracy en el conjunto de prueba, el resultado en el conjunto de entrenamiento fue significativamente menor. Esta discrepancia sugiere la necesidad de investigar más a fondo la generalización del modelo, considerar un conjunto de entrenamiento más grande y explorar otros factores que puedan influir en esta diferencia.

Se resalta la importancia del análisis de características, la elección adecuada de algoritmos y la evaluación rigurosa de modelos para abordar problemas de clasificación de imágenes en la conservación de la biodiversidad. La discrepancia en el accuracy plantea interrogantes interesantes y destaca la necesidad de futuras investigaciones para comprender completamente este fenómeno y mejorar aún más la capacidad de generalización de los modelos.

REFERENCIAS

- [1] F. Tanco, *Redes Neuronales Artificiales*. Universidad Tecnológica Nacional, 2021. [Online]. Available: <https://www.frba.utn.edu.ar/wp-content/uploads/2021/02/RNA.pdf>
- [2] D. J. Matich, *Redes Neuronales: Conceptos Básicos y Aplicaciones*. Universidad Tecnológica Nacional – Facultad Regional Rosario, 2001. [Online]. Available: https://www.fro.utn.edu.ar/repositorio/catedras/quimica/5_anio/orientadoral/monograias/matich-redesneuronales.pdf
- [3] E. F. Caicedo Bravo, *Redes Neuronales Perceptron y Adaline*. Programa Editorial Universidad del Valle, 2021. [Online]. Available: <https://libros.univalle.edu.co/index.php/programaeditorial/catalog/download/64/28/446?inline=1>