

**Reporte: Clasificación de Imágenes de Patos y Gansos utilizando Redes Neuronales  
Convolucionales (CNN)**

**TC3007C.501**

Inteligencia artificial avanzada para la ciencia de datos 2

Gpo 501

**Docentes**

Dr. Benjamín Valdés Aguirre

Ma. Eduardo Daniel Juárez Pineda

Dr. Ismael Solis Moreno

Dr. José Antonio Cantoral Ceballos

Dr. Carlos Alberto Dorantes Dosamantes

**Integrante**

Dafne Fernández Hernández A01369239

## 1. Introducción

En los últimos años, el campo de la visión por computadora ha experimentado avances significativos gracias a la aplicación de redes neuronales profundas, en particular las Redes Neuronales Convolucionales (CNN). Este tipo de red ha demostrado ser especialmente eficaz en la tarea de clasificación de imágenes debido a su capacidad para aprender patrones espaciales complejos en datos visuales.

Este proyecto tiene como objetivo principal desarrollar un modelo de clasificación de imágenes que sea capaz de distinguir entre dos tipos de aves: patos y gansos. Para alcanzar este objetivo, empleamos una arquitectura CNN conocida como ResNet18, que ha sido ampliamente utilizada en la literatura de visión por computadora y ha demostrado ser eficiente en tareas de clasificación.

La importancia de este proyecto radica en su aplicación potencial en monitoreo de vida silvestre, estudios de ecología y biología, y en herramientas de conservación. Además, el trabajo sirve como un ejemplo práctico de los pasos involucrados en el diseño, entrenamiento, evaluación e implementación de un modelo de deep learning.

---

## 2. Contexto del Trabajo

En ecología computacional, la clasificación de imágenes de especies es una tarea crucial para el monitoreo de la biodiversidad. La identificación de aves, en particular, plantea un reto debido a la similitud visual entre algunas especies. En nuestro caso, la distinción entre patos y gansos resulta desafiante para métodos tradicionales debido a sus características visuales similares, como el color y la forma del cuerpo.

Para resolver este problema, empleamos una Red Neuronal Convolutiva. Las CNN son adecuadas para esta tarea debido a su capacidad de aprender patrones en datos espaciales (imágenes) mediante el uso de convoluciones y agrupaciones que detectan bordes, texturas y formas. La arquitectura seleccionada fue ResNet18, que introduce la idea de bloques residuales para mitigar el problema del desvanecimiento del gradiente en redes profundas.

### Bloques Residuales

Un bloque residual permite que el modelo aprenda la identidad de la entrada, facilitando el entrenamiento de redes profundas. Matemáticamente, un bloque residual puede representarse como:

$$y = F(x\{W_i\}) + x$$

donde  $y$  es la salida del bloque,  $x$  es la entrada,  $F(x\{W_i\})$  es la transformación aprendida,  $W_i$  representa los pesos de la red.

---

## 3. Marco Técnico

### Herramientas Utilizadas

- Lenguaje de Programación: Python
- Librerías:
  - PyTorch: Para el desarrollo y entrenamiento de redes neuronales.
  - Torchvision: Para manipulación de datasets e importación de arquitecturas preentrenadas.
  - Matplotlib: Para la visualización de los resultados de las predicciones.

## Arquitectura de la Red

La arquitectura utilizada fue ResNet18, una CNN preentrenada en el dataset ImageNet y adaptada para la clasificación de dos clases: "duck" y "goose". Modificamos la última capa completamente conectada para ajustarse a nuestro problema de clasificación binaria. Esta arquitectura se representa de la siguiente manera:

1. Capas Convolucionales: Las capas convolucionales permiten extraer características espaciales de la imagen. Cada capa aplica filtros para detectar patrones específicos.
2. Bloques Residuales: Como se mencionó en la sección anterior, los bloques residuales mejoran la capacidad de la red para entrenar en profundidad.
3. Capa de Clasificación: La última capa tiene dos neuronas, correspondientes a las clases "duck" y "goose".

Matemáticamente, cada capa convolucional aplica una transformación de la forma:

$$y_{i,j} = \sum_{m=-k}^k \sum_{n=-k}^k x_{i+m, j+n} \cdot w_{m,n}$$

donde  $y_{i,j}$  es el valor resultante en la posición  $(i, j)$  de la característica de salida,  $x_{i+m, j+n}$  es el valor de la imagen de entrada en la posición  $(i+m, j+n)$ , y  $w_{m,n}$  representa los pesos del filtro.

---

## 4. Descripción de los Datos

### Fuente de los Datos

El dataset consiste en imágenes de patos y gansos organizadas en dos carpetas: duck y goose. Cada carpeta contiene varias

imágenes que fueron preprocesadas y aumentadas para mejorar la robustez del modelo.

### Preprocesamiento y Augmentación de Datos

Para aumentar el tamaño del conjunto de datos y mejorar la generalización del modelo, aplicamos técnicas de augmentación. Los métodos aplicados incluyeron:

- Rotate: Rotación aleatoria de las imágenes.
- Brightness y Contrast: Ajustes de brillo y contraste para imitar variaciones de iluminación.
- Color Jitter: Modificación de colores para mejorar la robustez en condiciones de color variables.
- Sharpen: Mejora de la nitidez para resaltar detalles.

La augmentación de datos se aplicó a un 30% del conjunto de entrenamiento, generando variaciones que permitieron al modelo aprender características visuales más diversas y reducir el overfitting.

---

## 5. Metodología

### Preprocesamiento de Imágenes

Cada imagen fue redimensionada a 224x224 píxeles y normalizada con los valores de media y desviación estándar del dataset ImageNet. Este preprocesamiento garantiza que las imágenes tengan un tamaño y un rango de valores adecuado para ser procesadas por la red ResNet18.

### Entrenamiento del Modelo

1. Función de Pérdida:

- Utilizamos la Entropía Cruzada (CrossEntropyLoss) como función de pérdida, que se define como:

$$L_{CE} = - \sum_{c=1}^C y_c \log(\hat{p}_c)$$

donde  $C$  es el número de clases,  $y_c$  es la etiqueta real, y  $\hat{p}_c$  es la probabilidad predicha para la clase  $c$

## 2. Algoritmo de Optimización:

- Empleamos el optimizador Adam debido a su capacidad para ajustar la tasa de aprendizaje adaptativamente. La actualización de pesos en Adam sigue la fórmula:

$$\theta_t + 1 = \theta_t - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}$$

donde  $\theta_t$  son los pesos en el paso  $t$ ,  $\alpha$  es la tasa de aprendizaje,  $\hat{m}_t$  y  $\hat{v}_t$  son las primeras y segundas estimaciones de momento.

## 3. Proceso de Entrenamiento:

- Entrenamos el modelo durante 5 epochs con un tamaño de lote de 32. La precisión en el conjunto de validación se calculó al final de cada epoch, permitiendo monitorear el rendimiento del modelo.

## Implementación de Predicciones

El modelo final fue implementado para realizar predicciones en imágenes seleccionadas aleatoriamente de las carpetas duck y goose. Cada vez que el modelo identifica un ganso, se pregunta al usuario si desea continuar.

## 6. Resultados

### Desempeño del Modelo

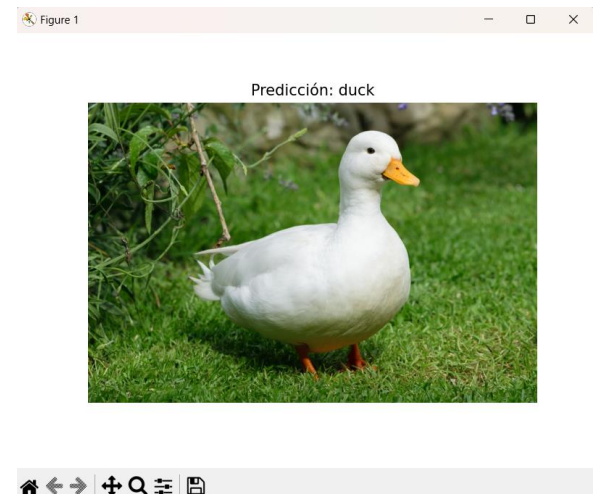
El modelo alcanzó una precisión en el conjunto de validación del 96.63%, lo cual indica que el modelo logró distinguir de manera precisa entre patos y gansos.

### Ejemplos de Predicciones

A continuación, se muestran ejemplos de imágenes clasificadas con el modelo, incluyendo su predicción correcta:

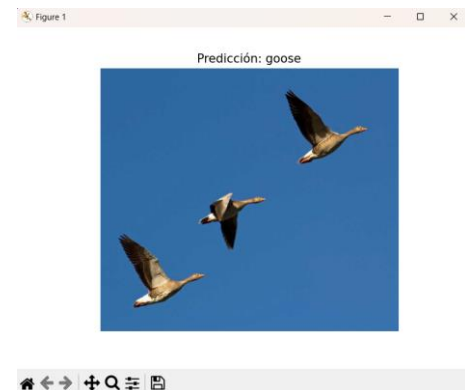
- Imagen de un pato:

- Predicción: Duck



- Imagen de un ganso:

- Predicción: Goose



---

## 7. Análisis y Discusión

### Overfitting y Underfitting

Dado que la precisión en entrenamiento y validación fue cercana, el modelo no mostró un overfitting significativo. La augmentación de datos ayudó a reducir el overfitting, permitiendo una generalización adecuada.

### Limitaciones

- Sesgo (Bias): El modelo podría estar sesgado hacia ciertas posiciones o condiciones de luz. Sin embargo, no se observó un sesgo significativo hacia una clase específica.
- Limitaciones del Dataset: Un mayor volumen de imágenes y variabilidad en condiciones ambientales podría mejorar el rendimiento del modelo.

---

## 8. Conclusión y Observaciones

El modelo ResNet18 modificado logró clasificar con éxito imágenes de patos y gansos con una alta precisión. La augmentación de datos y el preprocesamiento jugaron un papel importante en la capacidad del modelo para generalizar en nuevas imágenes.

### Futuras Mejoras

- Aumentar el dataset para incluir más variabilidad de las especies en condiciones de iluminación y ángulos diferentes.
- Implementar arquitecturas más complejas o explorar técnicas de ensemble para mejorar la precisión en casos más complejos.

Este trabajo demuestra el potencial de las CNN en la clasificación de imágenes y su aplicación en monitoreo de especies.

---

## Referencias

### 1. Dataset de Imágenes de Patos

- N. Alicenkbaytop. (2023). *Duck Images Dataset*. Kaggle. Disponible en: <https://www.kaggle.com/datasets/alicenkbaytop/duck-images>

### 2. Dataset de Imágenes de Gansos

- CV Datasets. (2023). *Goose Image Classification Dataset*. Disponible en: <https://images.cv/dataset/goose-image-classification-dataset>