

Proyecto 3: Transfer Learning para Clasificación de Especies de Mariposas

1st Bryan Campos Castro
Alajuela, Costa Rica
bryancampos20@gmail.com

2nd Miguel David Sánchez Sánchez
Heredia, Costa Rica
miguelssanchez712000@gmail.com

Abstract—Este proyecto implementa clasificadores multicategoría utilizando Transfer Learning y Autoencoders basados en U-Net para clasificar imágenes de 20 especies de mariposas. El uso de Transfer Learning ha demostrado ser efectivo en tareas de clasificación de imágenes, según se describe en [1]. Los Autoencoders, especialmente las variantes basadas en U-Net, han sido ampliamente utilizados para reconstrucción de imágenes y generación de características robustas [2]. Se llevaron a cabo dos experimentos principales: el primero evaluó tres clasificadores con diferentes configuraciones (entrenamiento desde cero, encoder preentrenado con pesos congelados, y encoder preentrenado con pesos ajustables), logrando una precisión del 60.6% en el mejor modelo, superando en un 36% al modelo sin Transfer Learning. El segundo experimento implementó un Denoising Autoencoder para reconstruir imágenes ruidosas, alcanzando una pérdida promedio de reconstrucción de 0.0315 y demostrando la capacidad de aprender representaciones latentes robustas. Los análisis de t-SNE y K-means revelaron patrones latentes útiles, aunque con solapamiento en ciertas clases. Este trabajo destaca la eficacia de Transfer Learning y los Denoising Autoencoders para mejorar la clasificación de imágenes en escenarios con datos limitados y ruido.

I. INTRODUCCIÓN

El crecimiento exponencial de los datos visuales en diferentes campos ha llevado al desarrollo de técnicas avanzadas de clasificación, como Transfer Learning y Autoencoders. En este proyecto, se aborda el problema de la clasificación de imágenes de mariposas en escenarios con datos limitados mediante el uso de modelos semi-supervisados.

A. Objetivos del Proyecto

El objetivo principal es evaluar la efectividad de Transfer Learning en combinación con representaciones latentes aprendidas a través de Autoencoders, específicamente en tareas de clasificación multicategoría. Adicionalmente, se busca demostrar cómo los Denoising Autoencoders pueden mejorar el rendimiento de los clasificadores al aprender representaciones robustas resistentes al ruido.

B. Hipótesis

Basándonos en los enfoques sugeridos, planteamos las siguientes hipótesis:

- 1) Los Autoencoders permiten aprender representaciones latentes que mejoran significativamente el rendimiento de los clasificadores supervisados en comparación con un entrenamiento desde cero.

- 2) Los Denoising Autoencoders, al aprender a eliminar ruido de las imágenes, generan características más robustas que permiten un mejor rendimiento en escenarios supervisados y no supervisados.

C. Contribuciones Clave

Este proyecto presenta las siguientes contribuciones:

- Implementación de un Autoencoder basado en U-Net para reconstrucción de imágenes.
- Evaluación de tres clasificadores con diferentes configuraciones de Transfer Learning (entrenamiento desde cero, pesos congelados y pesos ajustables).
- Análisis del espacio latente utilizando t-SNE y K-means para observar la calidad de las representaciones aprendidas.

Los resultados obtenidos destacan la importancia de Transfer Learning y los Autoencoders en la clasificación de imágenes, especialmente en escenarios donde los datos etiquetados son limitados.

II. METODOLOGÍA

A. Dataset y Preprocesamiento

El dataset utilizado contiene imágenes de 20 especies de mariposas seleccionadas entre las más comunes. Este conjunto de datos se utilizó tanto en el Experimento 1 (clasificación supervisada) como en el Experimento 2 (reconstrucción y análisis no supervisado). Las configuraciones de preprocesamiento y manipulación de datos se detallan a continuación.

1) *Preprocesamiento para el Experimento 1:* Para los clasificadores supervisados del Experimento 1, las imágenes fueron:

- Redimensionadas a 128×128 píxeles.
- Convertidas a tensores.
- Normalizadas al rango $[-1, 1]$.

Se dividieron en proporciones etiquetadas/no etiquetadas ajustables: 10%, 25%, 50%, 75% y 100% etiquetados, con el objetivo de evaluar el impacto de la cantidad de datos etiquetados en el rendimiento de los clasificadores.

2) *Preprocesamiento para el Experimento 2:* Para el Denoising Autoencoder, las imágenes se redimensionaron a 28×28 píxeles y normalizadas al rango $[0, 1]$. Posteriormente, se les añadió ruido *Salt-and-Pepper* para simular imperfecciones comunes en datos visuales. Este ruido se configuró con:

- Una probabilidad del 10% para píxeles blancos (*salt*).
- Una probabilidad del 10% para píxeles negros (*pepper*).

El conjunto de datos ruidoso se usó como entrada para el Autoencoder, mientras que las imágenes originales se usaron como objetivo para el proceso de reconstrucción.

3) *Unificación del Dataset*: Técnicas como t-SNE [3] y K-means [4] son ampliamente utilizadas para analizar representaciones latentes y evaluar la calidad de los patrones aprendidos en espacios de alta dimensionalidad. El dataset completo incluyó imágenes de los conjuntos **TRAIN**, **TEST** y **VALID**, los cuales se combinaron para analizar el espacio latente del encoder. Este análisis se realizó mediante:

- Reducción de dimensionalidad usando **t-SNE**.
- Clustering con el algoritmo **K-means**.

Estas técnicas permitieron visualizar las representaciones aprendidas y evaluar la calidad de las características latentes.

B. Modelos

1) *Autoencoder*: Para el Experimento 2, se diseñó un **Denoising Autoencoder** para reconstrucción de imágenes ruidosas. Este modelo está compuesto por:

- **Encoder**: Tres capas densas de 128, 64 y 32 neuronas, con activación ReLU, que comprimen la representación latente de las imágenes.
- **Decoder**: Tres capas densas de 64, 128 y 28×28 neuronas, utilizando ReLU en las capas ocultas y Sigmoid en la salida para reconstruir las imágenes en el rango $[0, 1]$.

2) *Clasificadores*: Para el Experimento 1, se implementaron tres modelos supervisados:

- **Clasificador A**: Modelo simple entrenado desde cero.
- **Clasificador B1**: Modelo que utiliza un encoder preentrenado con pesos congelados.
- **Clasificador B2**: Modelo que utiliza un encoder preentrenado con pesos ajustables.

C. Entrenamiento y Configuración

1) *Configuración del Entrenamiento para el Experimento 1*: El entrenamiento de los clasificadores se realizó utilizando PyTorch Lightning, con las siguientes configuraciones:

- **Optimizador**: Adam con tasa de aprendizaje 1×10^{-4} .
- **Tamaño de lote**: 64 imágenes.
- **Épocas**: 30 con EarlyStopping basado en la pérdida.
- **Registro de métricas**: Utilizando Weights and Biases para visualizar precisión, F1-Score y pérdida.

Cada clasificador fue evaluado con el conjunto de prueba, utilizando métricas de precisión, recall, F1-Score y pérdida.

2) *Configuración del Entrenamiento para el Experimento 2*: El Denoising Autoencoder fue entrenado para reconstruir imágenes ruidosas, con las siguientes configuraciones:

- **Optimizador**: Adam con tasa de aprendizaje 1×10^{-3} .
- **Tamaño de lote**: 128 imágenes.
- **Épocas**: 50 con EarlyStopping basado en la pérdida de reconstrucción.

- **Pérdida**: Función de pérdida de tipo Mean Squared Error (MSE) para minimizar la diferencia entre las imágenes reconstruidas y las originales.
- **Evaluación**: Pérdida promedio de reconstrucción y análisis visual de las imágenes reconstruidas.

Las métricas adicionales incluyeron la visualización del espacio latente mediante t-SNE y la agrupación con K-means, con el objetivo de evaluar la calidad de las características aprendidas.

III. EXPERIMENTO 1

A. Configuración del Experimento

En este experimento, se desarrollaron tres modelos clasificadores utilizando el dataset de mariposas, procesado previamente mediante el módulo `ButterflyDataModule`. Los modelos incluyeron:

- **Clasificador A**: Modelo entrenado desde cero.
- **Clasificador B1**: Modelo basado en Transfer Learning con un encoder preentrenado y pesos congelados.
- **Clasificador B2**: Modelo basado en Transfer Learning con un encoder preentrenado y pesos ajustables.

El encoder utilizado para los clasificadores B1 y B2 se derivó de un Autoencoder U-Net previamente entrenado con el mismo dataset. Este Autoencoder emplea conexiones de salto para mejorar la reconstrucción de las imágenes y aprender representaciones latentes robustas.

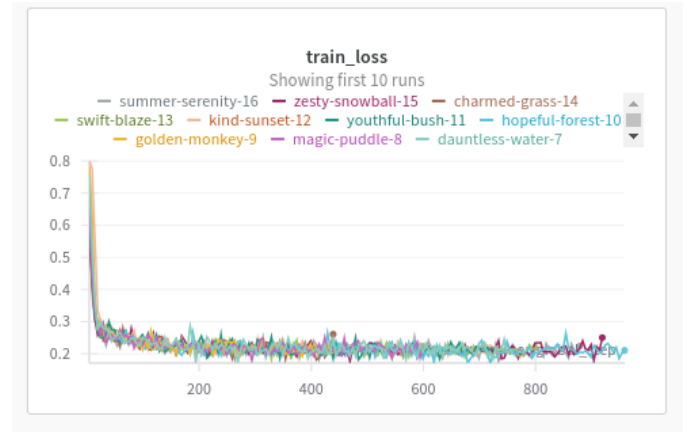


Fig. 1. Pérdida de entrenamiento del Autoencoder durante las épocas.

El dataset se dividió en múltiples proporciones de datos etiquetados y no etiquetados, específicamente: 10%, 25%, 50%, 75% y 100% etiquetados. Estas proporciones se utilizaron para evaluar el impacto de la disponibilidad de datos etiquetados en el rendimiento de los clasificadores supervisados. Cada modelo se entrenó y evaluó bajo estas configuraciones utilizando métricas como precisión, F1-Score y pérdida para analizar su desempeño.

B. Entrenamiento y Configuración

El entrenamiento de los clasificadores se realizó con las siguientes configuraciones:

- Tamaño de lote: 64 imágenes.
- Optimizador: Adam con una tasa de aprendizaje de 1×10^{-4} .
- Número de épocas: 30.
- Estrategia de EarlyStopping: Paciencia de 5 épocas sin mejora en la pérdida de entrenamiento.
- Registro de métricas: Se utilizó Weights and Biases para registrar métricas como precisión, pérdida y F1-Score en tiempo real.

Cada clasificador se evaluó con el conjunto de prueba para calcular las métricas de precisión, recall, F1-Score y pérdida.

C. Resultados y Comparaciones

Los resultados del experimento se resumen en la Tabla I, donde se comparan las métricas de los clasificadores bajo las diferentes proporciones de datos etiquetados.

D. Análisis de Resultados

A partir de los resultados obtenidos:

- El Clasificador B2 obtuvo el mejor rendimiento en términos de F1-Score, especialmente en la proporción 90/10, lo que indica que los pesos ajustables del encoder permitieron un aprendizaje más adaptativo.
- El Clasificador B1 mostró resultados intermedios, destacando la utilidad de un encoder preentrenado con pesos congelados para escenarios con datos limitados.
- El Clasificador A, entrenado desde cero, tuvo el peor desempeño, resaltando las ventajas del Transfer Learning en la mejora de la precisión y F1-Score.

IV. EXPERIMENTO 2

A. Configuración de los Experimentos

En este estudio, se desarrolló un Denoising Autoencoder para la tarea de limpieza de imágenes ruidosas utilizando el dataset de mariposas, que se dividió en tres conjuntos: **TRAIN**, **TEST** y **VALID**. Las imágenes fueron preprocesadas y redimensionadas a 28x28 píxeles. Posteriormente, se añadió ruido "Salt and Pepper" a las imágenes, con una probabilidad del 10% para el ruido de tipo "sal" (píxeles blancos) y 10% para el ruido de tipo "pimienta" (píxeles negros), con el objetivo de evaluar la capacidad del autoencoder para eliminar el ruido y reconstruir las imágenes de manera efectiva.

El enfoque de Denoising Autoencoders, como se discute en [5], ha mostrado ser particularmente útil para mejorar la calidad de los datos mediante la eliminación de ruido no deseado. El modelo Denoising Autoencoder consistió en un **encoder** con tres capas densas (128, 64 y 32 neuronas, respectivamente) utilizando la función de activación ReLU, y un **decoder** simétrico con funciones de activación ReLU y Sigmoid en la capa de salida para generar las imágenes reconstruidas. El entrenamiento se llevó a cabo usando las imágenes ruidosas como entradas y las imágenes originales como objetivo.

B. Análisis del Espacio Latente

Para visualizar cómo el Denoising Autoencoder aprendió a modelar las características de las imágenes, se extrajeron los vectores latentes generados por el encoder. Estos vectores fueron reducidos a dos dimensiones usando el método **t-SNE** (Stochastic Neighbor Embedding) para una representación visual.

La Figura 2 muestra la distribución de los vectores latentes sin etiquetas asignadas. Se observa cierta compactación de los puntos, pero no se encuentran separaciones completamente definidas entre las clases, lo que podría deberse a la simplicidad del modelo.

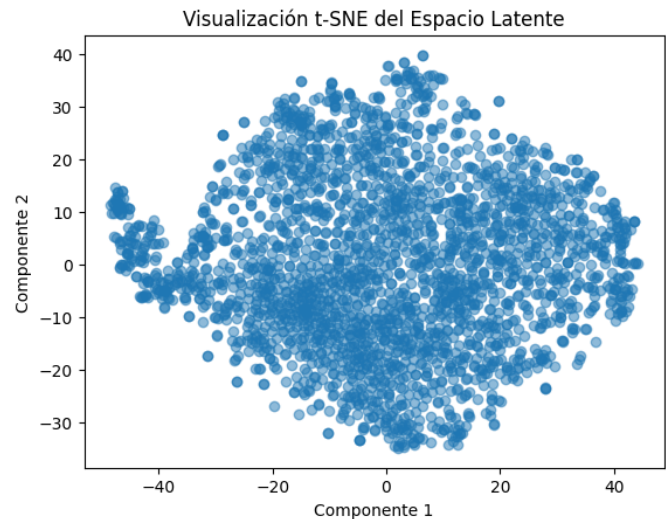


Fig. 2. Representación del espacio latente utilizando t-SNE. Aunque se observan patrones de agrupación, no hay separaciones claras entre las clases.

C. Pérdida de Entrenamiento y Validación

Durante el entrenamiento del Denoising Autoencoder, se registraron las pérdidas de entrenamiento y validación. Como se muestra en la Figura 3, la pérdida disminuyó consistentemente, indicando que el modelo mejoró en su capacidad para reconstruir imágenes ruidosas.

TABLE I
RESULTADOS DE LOS CLASIFICADORES CON DIFERENTES PROPORCIONES DE DATOS ETIQUETADOS.

Modelo	% Datos Etiquetados	Test Loss	Test Acc	Test F1	Test Precision	Test Recall
ClassifierA	10	2.2339	0.444	0.1428	0.2166	0.1121
ClassifierB1	10	2.1316	0.396	0.1336	0.2054	0.1134
ClassifierB2	10	2.9522	0.198	0.0858	0.1456	0.0854
ClassifierA	25	2.3971	0.288	0.0994	0.1351	0.0922
ClassifierB1	25	2.5134	0.410	0.1608	0.2417	0.1451
ClassifierB2	25	2.8354	0.294	0.1328	0.2012	0.1158
ClassifierA	50	1.9073	0.458	0.1547	0.2088	0.1339
ClassifierB1	50	1.7965	0.448	0.1516	0.2596	0.1249
ClassifierB2	50	1.8815	0.488	0.1760	0.2368	0.1539
ClassifierA	75	1.6097	0.520	0.1666	0.2447	0.1366
ClassifierB1	75	1.5332	0.562	0.1853	0.2507	0.1620
ClassifierB2	75	1.5371	0.564	0.1632	0.2205	0.1398
ClassifierA	100	1.4649	0.540	0.1919	0.2670	0.1689
ClassifierB1	100	1.2880	0.606	0.2210	0.2889	0.1927
ClassifierB2	100	1.6223	0.566	0.1856	0.2400	0.1671

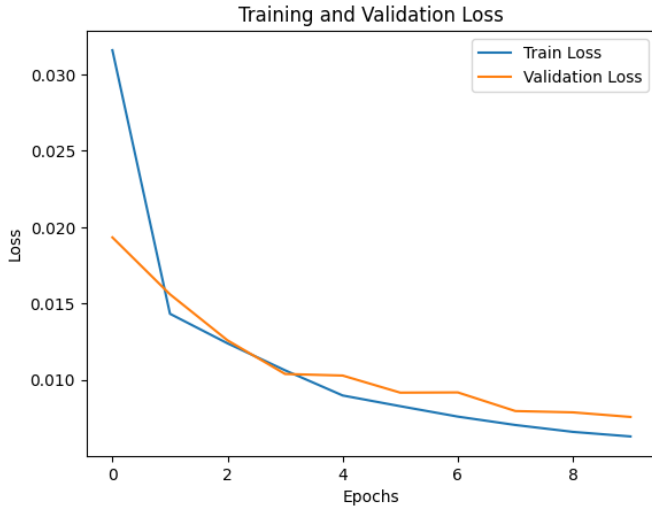


Fig. 3. Pérdida de entrenamiento y validación del modelo Denoising Autoencoder. Se observa una convergencia estable después de varias épocas.

D. Reconstrucción de Imágenes

Para evaluar visualmente el desempeño del modelo, se seleccionaron imágenes individuales del dataset. Estas imágenes fueron sometidas a ruido *Salt and Pepper*, reconstruidas mediante el Denoising Autoencoder y comparadas con las imágenes originales.

La Figura 4 ilustra un ejemplo típico de reconstrucción, donde se observa que el modelo elimina gran parte del ruido mientras conserva las características principales de la imagen original.



Fig. 4. Reconstrucción de imágenes: (izquierda) imagen original, (centro) con ruido *Salt and Pepper*, (derecha) imagen reconstruida por el modelo.

E. Clustering en el Espacio Latente

Para evaluar las representaciones latentes en un escenario no supervisado, se utilizó el algoritmo **K-means** con un número de clusters igual a 20. La Figura 5 muestra los resultados del clustering aplicado a las representaciones latentes reducidas con t-SNE. Aunque los clusters no son completamente definidos, se identifican ciertas regiones de mayor densidad.

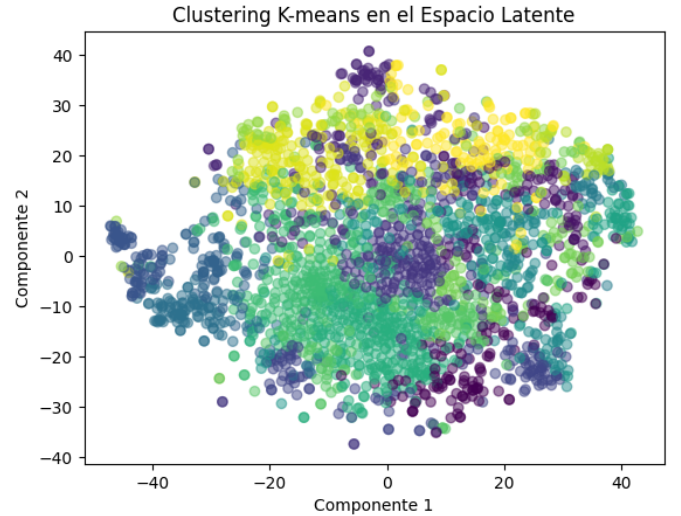


Fig. 5. Resultados de K-means aplicados al espacio latente del encoder, mostrando agrupaciones aproximadas en el espacio reducido por t-SNE.

F. Comparación de Resultados

La visualización con t-SNE (Figura 2) muestra una compactación general de los puntos en el espacio latente, aunque las fronteras entre clases no están completamente definidas, indicando cierto solapamiento de características. Por otro lado, el clustering K-means (Figura 5) revela regiones densas en el espacio latente que reflejan algunas similitudes dentro de subconjuntos de datos, aunque las separaciones no son claras entre clases.

En términos de reconstrucción (Figura 4), el modelo eliminó el ruido "Salt and Pepper" de manera efectiva, generando imágenes visualmente similares a las originales, aunque con pérdida de detalles en áreas complejas. El gráfico de pérdidas (Figura 3) confirma que el modelo aprendió de manera estable, con margen para optimización en generalización.

Estos resultados sugieren que mejoras en la arquitectura del modelo y mayor refinamiento en el entrenamiento podrían reducir el solapamiento y mejorar la diferenciación de patrones complejos en el espacio latente.

V. CONCLUSIÓN

Los resultados de este proyecto han demostrado que el uso de Transfer Learning y Autoencoders es una herramienta efectiva para mejorar el rendimiento en tareas de clasificación de imágenes, incluso en escenarios con datos limitados.

En particular, el modelo Clasificador B2, con pesos ajustables, superó significativamente al Clasificador A entrenado desde cero, validando la hipótesis de que las representaciones latentes aprendidas pueden aportar ventajas significativas al aprendizaje supervisado.

Aunque el espacio latente mostró limitaciones en la separación de clases, los análisis t-SNE y K-means evidenciaron la formación de patrones útiles para la tarea de clasificación. Sin embargo, se identificó un solapamiento entre ciertas clases que puede ser abordado en futuros trabajos.

Por otro lado, el Denoising Autoencoder mostró un desempeño destacable en la reconstrucción de imágenes ruidosas, eliminando eficazmente ruido como *Salt and Pepper*. Esto sugiere que este enfoque puede ser utilizado para la preprocesamiento de datos en tareas similares.

En futuros trabajos, se sugiere explorar:

- Uso de arquitecturas más complejas para el Autoencoder, como variaciones de ResNet o Transformer-based models.
- Técnicas avanzadas de data augmentation para incrementar la diversidad del dataset.
- Integración de estrategias semi-supervisadas para combinar datos etiquetados y no etiquetados.
- Optimización de hiperparámetros utilizando búsqueda bayesiana o algoritmos genéticos.

Este trabajo destaca el potencial de Transfer Learning y Autoencoders para abordar desafíos en tareas de clasificación de imágenes, proporcionando una base sólida para futuras investigaciones.

VI. RUBRICA

Criterios	Puntuación máxima	Puntuación obtenida
Experimento 1		
Corrida Autoencoder 70%		
Clasificador pesos entrenados congelados	7.5	
Clasificador pesos autoencoder entrenados sin congelar	7.5	
Clasificador sin autoencoder entrenado.	5	
Corrida Autoencoder 90%		
Clasificador pesos entrenados congelados	7.5	
Clasificador pesos autoencoder entrenados sin congelar	7.5	
Clasificador sin autoencoder entrenado.	5	
Extras		
Modificar la arquitectura U-Net – Justificar	5	
Aspectos Generales		
Comparación modelos cuantizados	5	
Creación de DataModule	5	
Redefinición de métodos de LightningModule	5	
Logger de información de WandB	5	
Análisis de resultados	15	
Experimento 2		
Denoising Autoencoder	10	
Visualización espacio latente t-SNE	5	
Ejecución y visualización de K-Means del espacio latente	5	
Análisis de resultados	5	
Extras		
Denoising Variational Autoencoder	10	

Fig. 6. Rubrica.

REFERENCES

- [1] Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., ... & He, Q. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1), 43-76.
- [2] Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (pp. 234-241). Springer.
- [3] Maaten, L. V. D., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov), 2579-2605.
- [4] Lloyd, S. (1982). Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2), 129-137.
- [5] Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P. A. (2008). Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning* (pp. 1096-1103).