Universidad del Valle de Guatemala
Facultad de Ingeniería
Deep Learning y Sistemas Inteligentes

Proyecto Final

Autor/Carné:
Fabiola Contreras, 22787
Diego Duarte, 22075
José Marchena, 22398
Sofía Velásquez, 22049
María José Villafuerte, 22129

Catedrático:
Luis Alberto Suriano Saravia

Sección 10

Guatemala, 2025

Índice

I. Descripción del problema

La colorización de imágenes en escala de grises es un problema complejo en el campo de la visión por computadora que requiere no solo entender la estructura y contenido de una imagen, sino también inferir colores realistas basándose en el contexto semántico. Este desafío tiene aplicaciones prácticas significativas en la restauración de fotografías históricas, mejora de material cinematográfico antiguo, procesamiento de imágenes médicas y satelitales. Tradicionalmente, este proceso se realizaba manualmente por artistas especializados, requiriendo horas de trabajo meticuloso y conocimiento experto para asignar colores coherentes a cada región de la imagen.

El problema radica en que la conversión de color a escala de grises es una transformación con pérdida de información: múltiples combinaciones de colores pueden resultar en el mismo tono de gris. Por lo tanto, la colorización automática debe aprender patrones complejos sobre qué colores son naturales para diferentes objetos y contextos (por ejemplo, el cielo tiende a ser azul, el pasto verde, la piel en tonos cálidos), mientras mantiene coherencia espacial y semántica en toda la imagen.

Nuestro objetivo es desarrollar un sistema basado en deep learning que pueda realizar este proceso de manera automática, generando colorizaciones realistas y visualmente convincentes.

II. Análisis

Desafíos Técnicos

El problema de colorización presenta varios desafíos técnicos que requieren un análisis cuidadoso. Primero, existe una ambigüedad inherente: una imagen en escala de grises no contiene suficiente información para determinar de manera única los colores originales. Por ejemplo, un objeto rojo brillante y uno azul oscuro pueden tener el mismo valor de gris. Segundo, el modelo debe aprender relaciones semánticas complejas entre objetos y sus colores típicos, así como mantener coherencia en regiones adyacentes. Tercero, los colores deben ser no solo técnicamente correctos sino también perceptualmente plausibles para el observador humano.

Enfoque con Redes Generativas Adversarias (GANs)

Las Redes Generativas Adversarias (GANs) han demostrado ser efectivas para este problema debido a su capacidad de generar imágenes realistas mediante un proceso de entrenamiento adversarial. En nuestro caso, utilizamos una arquitectura condicional (conditional GAN o cGAN) donde el generador aprende a predecir colores condicionado en la imagen en escala de grises, mientras que el discriminador aprende a distinguir entre colorizaciones reales y generadas. Esta competencia adversarial impulsa al generador a producir colores cada vez más realistas.

Además, trabajamos en el espacio de color LAB en lugar de RGB, lo que permite separar la información de luminancia (canal L, que ya poseemos) de la información

cromática (canales a y b, que debemos predecir), simplificando la tarea del modelo y mejorando la calidad de los resultados.

III. Propuesta de solución

Arquitectura GAN para Colorización

Nuestra propuesta consiste en implementar una Red Generativa Adversaria (GAN) especializada para colorización automática, compuesta por dos redes neuronales profundas que trabajan en conjunto: un Generador basado en U-Net y un Discriminador tipo PatchGAN. El generador recibe como entrada el canal L (luminancia) de una imagen en espacio LAB y genera los canales a y b (crominancia), efectivamente "coloreando" la imagen. El discriminador, por su parte, recibe imágenes LAB completas (ya sean reales o generadas) y evalúa su realismo a nivel local mediante patches, proporcionando retroalimentación al generador sobre qué tan convincentes son sus colorizaciones.

Componentes Clave del Sistema

El Generador VGG16-UNet utiliza una arquitectura híbrida que combina las fortalezas de redes pre-entrenadas con conexiones tipo U-Net. El codificador aprovecha los pesos pre-entrenados de VGG16, una red convolucional profunda entrenada en ImageNet, que ya posee conocimiento robusto sobre características visuales como bordes, texturas y patrones de objetos. Este codificador consta de 4 etapas de encoding más un bottleneck, donde las dos últimas etapas (codificador 4 y bottleneck) tienen sus pesos descongelados para permitir fine-tuning específico a la tarea de colorización. El generador también incorpora un mecanismo de atención que enfatiza las características más relevantes del bottleneck, y acepta información condicional de clase mediante un vector one-hot de 8 dimensiones (correspondiente a las categorías: avión, auto, gato, perro, flor, fruta, motocicleta, persona), permitiendo que el modelo ajuste sus predicciones de color según el tipo de objeto presente. El decodificador utiliza skip connections estilo U-Net que concatenan características del codificador con las del decodificador en cada nivel, y aplica bloques residuales después de cada etapa de upsampling para refinar los detalles y producir colorizaciones más precisas.

El Discriminador PatchGAN evalúa la imagen en regiones locales de aproximadamente 34×34 píxeles, produciendo una matriz de 14×14 decisiones sobre el realismo de diferentes partes de la imagen. Este enfoque local es superior a un discriminador global porque captura mejor la coherencia de texturas y colores en regiones específicas, mientras es computacionalmente más eficiente. La arquitectura consta de 5 capas convolucionales con kernel 4×4 y reducción progresiva de dimensiones espaciales (128→64→32→16→15→14), utilizando BatchNormalization en capas intermedias y activaciones LeakyReLU con pendiente

negativa de 0.2. Siguiendo las mejores prácticas, la primera capa omite la normalización por lotes para permitir mayor flexibilidad en el aprendizaje de características básicas, mientras que la capa de salida no utiliza función de activación ya que se trabaja con BCEWithLogitsLoss. El discriminador recibe imágenes LAB completas de 3 canales: el canal L (luminancia) que preserva la estructura de la imagen, y los canales a (eje verde-rojo) y b (eje azul-amarillo) que contienen la información cromática a evaluar.

Estrategia de Entrenamiento

El entrenamiento sigue un proceso adversarial donde alternamos la actualización del discriminador y del generador. El discriminador se entrena para distinguir correctamente entre imágenes reales (del dataset) y falsas (generadas), mientras que el generador se entrena para "engañar" al discriminador produciendo colorizaciones cada vez más realistas.

Crucialmente, la función de pérdida del generador combina dos componentes: una pérdida adversarial (que mide qué tan bien engaña al discriminador) y una pérdida L1 (que mide la similitud pixel a pixel con los colores reales), ponderada con un factor $\lambda=100$. Esta combinación asegura que las colorizaciones sean tanto perceptualmente realistas como técnicamente precisas.

IV. Datos

Dataset Original

El dataset utilizado proviene de: https://www.kaggle.com/datasets/prasunroy/natural-images?resource=download, un conjunto compilado de 6,899 imágenes distribuidas en 8 clases distintas

- Airplane
- Car
- Cat
- Dog
- Flower
- Fruit
- Motorbike
- Person

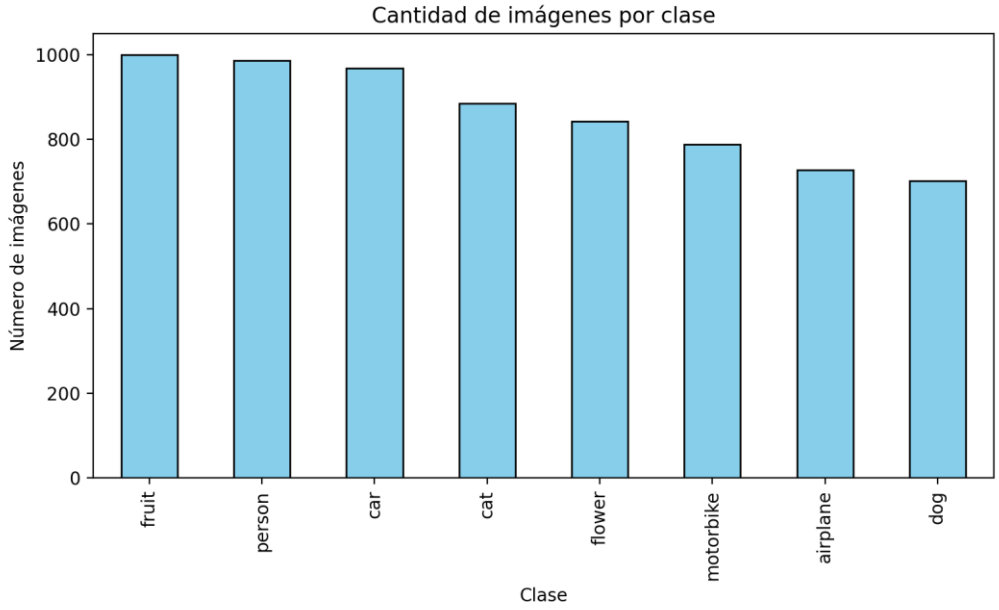
Estas imágenes fueron tomadas de distintos datasets públicos (como ImageNet, PASCAL VOC, y Stanford Cars) y unificadas en este conjunto de referencia.

El dataset fue diseñado originalmente para estudiar los efectos de degradaciones en arquitecturas de redes neuronales profundas (Roy et al., 2018).

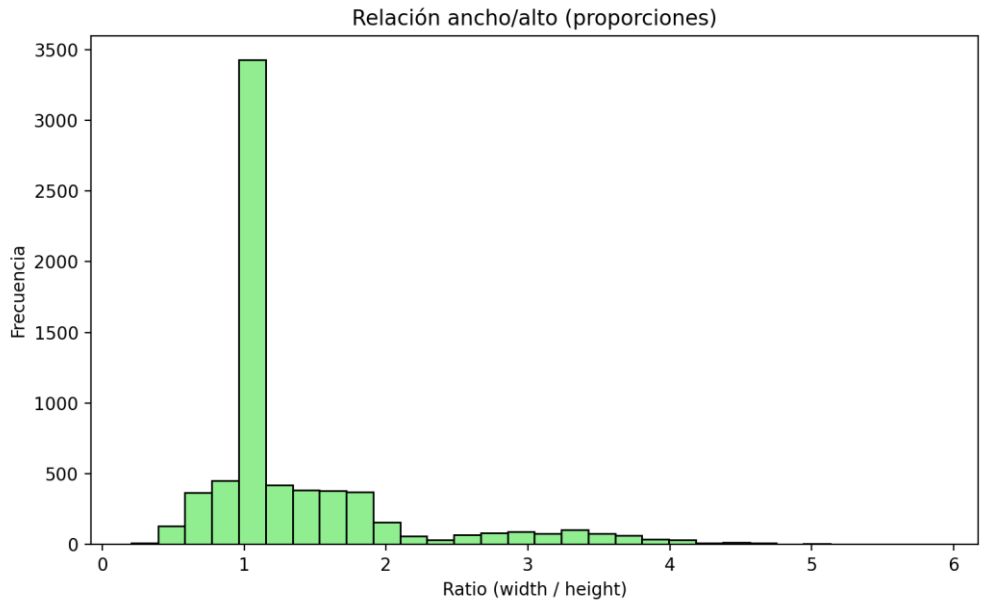
Análisis Exploratorio

Antes de aplicar cualquier preprocesamiento, se realizó un análisis exploratorio del conjunto de imágenes original para conocer su distribución, variedad y dimensiones.

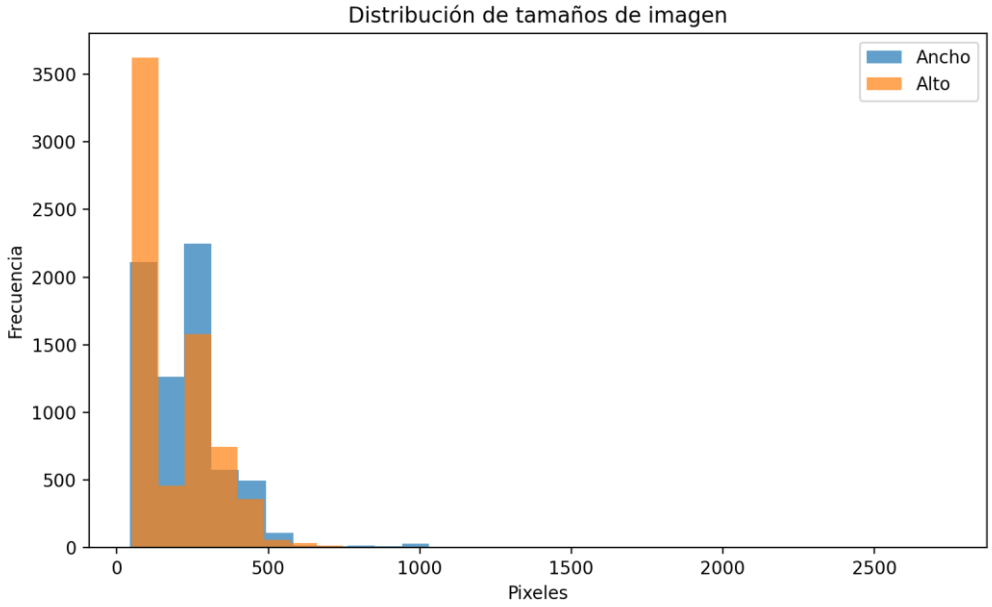
El dataset cuenta con 6,899 imágenes distribuidas en 8 clases. La clase con mayor cantidad de imágenes es fruit (1,000), mientras que la de menor cantidad es dog (702). En general, las clases presentan un número equilibrado de muestras, lo que favorece el entrenamiento del modelo sin necesidad de balanceo adicional.



El tamaño de las imágenes presenta una gran variabilidad, tanto en resolución como en proporciones. Los tamaños van desde 43×50 px hasta 2737×2665 px, con un promedio de 235×198 px.



El análisis de la relación ancho/alto arroja un promedio de 1.36, indicando que la mayoría de las imágenes son ligeramente apaisadas. Además, algunas clases presentan tamaños muy uniformes (como *car* y *fruit*, con imágenes fijas de 100×100 px), mientras que otras presentan una alta dispersión (como *flower*, con tamaños que van desde 51×59 hasta 2737×2665 px).



Procesamiento

Dado que las imágenes provenían de distintas fuentes y presentaban resoluciones y proporciones variables, se realizó un proceso de normalización y estandarización para garantizar coherencia en los datos de entrada del modelo.

1. **Redimensionamiento**

   Las imágenes fueron redimensionadas a un tamaño uniforme de 128×128 píxeles.

   Para evitar distorsión en las imágenes rectangulares, se aplicó un método que mantiene la proporción original y rellena con bordes negros cuando es necesario (padding), asegurando que todas las muestras tuvieran el mismo tamaño cuadrado.

2. **Conversión de formato y nombres**

   Todas las imágenes se guardaron en formato .jpg, con nombres estandarizados siguiendo la convención:

   gray_img00001-1.jpg, gray_img00002_1.jpg, … gray_imgNNNNN_M.jpg.

   Esto permitió una gestión más ordenada y evitó conflictos entre nombres provenientes de distintas carpetas o clases.

3. **Normalización de color**

   Las imágenes se convirtieron a escala de grises para ser utilizadas como entrada del modelo de colorización.

   Esto reduce los tres canales de color (RGB) a un único canal de intensidad, conservando la estructura y los bordes de la imagen.

4. **Aumento de datos (Data Augmentation)**

   Para incrementar la diversidad del conjunto de entrenamiento y mejorar la capacidad de generalización del modelo, se aplicaron transformaciones aleatorias a las imágenes, incluyendo:

   - Rotaciones aleatorias de ±20°
   - Volteos horizontales y verticales
   - Variaciones de brillo y contraste
   - Escalado y recortes menores

   Estas modificaciones permiten que el modelo se exponga a distintas variaciones de cada imagen, reduciendo el riesgo de sobreajuste (overfitting) y mejorando su robustez ante nuevas imágenes.

5. **Almacenamiento final**

   El conjunto procesado resultante mantiene un tamaño uniforme de 128×128 y conserva la estructura por clases.

   Cada imagen está disponible tanto en color como en escala de grises, generando pares alineados que servirán como entrada y referencia para el entrenamiento del modelo de colorización.

Dataset Final

Tras el preprocesamiento, el dataset quedó completamente uniforme y preparado para el entrenamiento del modelo de colorización. Se generaron 6,899 imágenes en total, donde cada imagen original se transformó en un par perfectamente alineado: una versión a color (RGB) y su correspondiente versión en escala de grises, que servirá como entrada para el modelo. Todas las imágenes fueron redimensionadas a 128×128 píxeles, aplicando padding cuando fue necesario para mantener las proporciones originales y evitar distorsiones en los objetos.

Cada archivo fue renombrado siguiendo el nuevo formato color_00001_1.jpg y gray_00001_1.jpg, donde el prefijo indica el tipo de imagen (color o gris), el número central representa un identificador secuencial y el último número corresponde a la categoría del conjunto original (por ejemplo: 1–airplane, 2–car, 3–cat, 4–dog, 5–flower, 6–fruit, 7–motorbike, 8–person).

Las imágenes se organizaron en dos carpetas principales:

- ImagesProcessed/color → contiene las versiones a color.
- ImagesProcessed/gray → contiene las versiones en escala de grises.

Gracias a este esquema, cada par de imágenes está perfectamente alineado pixel a pixel y categorizado de manera clara, lo que garantiza que el generador de la GAN disponga de la información estructurada necesaria para aprender la colorización de forma eficiente. Este dataset final es balanceado, consistente y listo para su uso en las etapas de entrenamiento y validación del modelo.

V. Herramientas aplicadas

Generador

1. **Transfer Learning con VGG16**

   El generador implementado aprovecha transfer learning mediante el uso de pesos pre-entrenados de VGG16 (Simonyan & Zisserman, 2014) en el

codificador. VGG16 es una red convolucional profunda entrenada en millones de imágenes de ImageNet, que ha demostrado capacidad excepcional para extraer características visuales jerárquicas: desde bordes y texturas en capas tempranas hasta patrones semánticos complejos en capas profundas. Esta estrategia es particularmente valiosa para nuestro problema de colorización, ya que el modelo ya posee conocimiento sobre la apariencia visual de objetos comunes (cielos, vegetación, piel, etc.), reduciendo significativamente el tiempo de entrenamiento y mejorando la calidad de los resultados especialmente cuando se trabaja con datasets de tamaño moderado como el nuestro (6,899 imágenes).

La arquitectura del codificador extrae características en 4 niveles de abstracción creciente, más un bottleneck: enc1 (capas VGG 0-4, características de 64 canales), enc2 (capas 4-9, 128 canales), enc3 (capas 9-16, 256 canales), enc4 (capas 16-23, 512 canales) y bottleneck (capas 23-30, 512 canales). Implementamos una estrategia de congelamiento selectivo donde las primeras tres etapas del codificador mantienen sus pesos pre-entrenados congelados (requires_grad=False), mientras que enc4 y el bottleneck son descongelados para fine-tuning. Esta decisión se basa en la observación de que las capas tempranas aprenden características genéricas que se transfieren bien entre tareas, mientras que las capas profundas requieren adaptación específica al dominio de colorización. Para adaptar el input de 1 canal (grayscale) más 8 canales (clase one-hot) a los 3 canales RGB que espera VGG16, utilizamos una capa convolucional adaptadora de 1×1 que mapea de 9 a 3 canales.

2. **Colorización Condicional por Clase**

   Una característica distintiva del generador es su naturaleza condicional: además de la imagen en escala de grises, el modelo recibe un vector one-hot de 8 dimensiones que codifica la categoría del objeto principal (0: avión, 1: auto, 2: gato, 3: perro, 4: flor, 5: fruta, 6: motocicleta, 7: persona). Este vector se expande espacialmente a un mapa de características del mismo tamaño que la imagen de entrada y se concatena con el canal L antes de pasar por el adaptador. La colorización condicional permite que el modelo ajuste sus predicciones según el contexto semántico: por ejemplo, puede aprender que los gatos suelen tener colores en gamas de grises, naranjas o blancos, mientras que las flores presentan una diversidad cromática mucho mayor. Este enfoque mejora significativamente la coherencia semántica de las colorizaciones y permite cierto control sobre el proceso generativo.

3. **Mecanismo de Atención**

Después del bottleneck, implementamos un módulo de atención simple que permite al modelo enfocarse en las características más relevantes para la colorización. Este módulo consta de dos convoluciones 1×1 con una capa intermedia de reducción de dimensionalidad (factor 8): la primera reduce de 512 a 64 canales con activación ReLU, y la segunda expande de vuelta a 512 canales con activación Sigmoid. La salida del módulo de atención son pesos entre 0 y 1 que se multiplican elemento a elemento con las características del bottleneck, amplificando regiones importantes y suprimiendo información irrelevante. Este mecanismo de atención ayuda al modelo a concentrarse en áreas que requieren decisiones cromáticas más cuidadosas, como rostros, objetos principales, o regiones con transiciones de color complejas.

4. **Decodificador con Skip Connections y Bloques Residuales**

   El decodificador sigue una arquitectura inspirada en U-Net con skip connections que concatenan características del codificador con las del decodificador en cada nivel de abstracción correspondiente. Cada etapa de decoding (up1-up4) realiza: (1) upsampling bilineal 2× de las características del nivel anterior, (2) concatenación con las skip connections del codificador en ese nivel, (3) dos convoluciones 3×3 con BatchNorm y ReLU para procesar las características concatenadas. Esta estructura permite recuperar detalles espaciales finos que podrían perderse en el bottleneck comprimido. Adicionalmente, después de cada etapa de upsampling aplicamos un bloque residual que consiste en dos convoluciones 3×3 con una conexión residual (skip connection dentro del mismo nivel). Los bloques residuales demostraron mejorar el flujo de gradientes durante el entrenamiento y permiten al modelo aprender refinamientos sutiles en los colores predichos, resultando en transiciones más suaves y detalles más finos. La capa final es una convolución 1×1 que mapea de 32 canales a 2 canales (a y b), seguida de activación Tanh para normalizar la salida al rango [-1, 1].

Discriminador

1. **Arquitectura PatchGAN**

   El discriminador implementado se basa en la arquitectura PatchGAN, introducida originalmente en el paper "Image-to-Image Translation with Conditional Adversarial Networks" (Pix2Pix, Isola et al., 2017). A diferencia de los discriminadores tradicionales que producen una única clasificación para toda la imagen, PatchGAN evalúa la imagen mediante patches (regiones locales), generando una matriz de predicciones que indican el realismo de diferentes zonas de la imagen. Esta aproximación presenta ventajas

significativas: requiere menos parámetros que un discriminador completo, captura mejor la consistencia local de texturas y colores, y resulta más efectiva para tareas de traducción imagen-a-imagen como la colorización.

La arquitectura específica implementada, para nuestras imágenes de 128×128 píxeles, consta de 5 capas convolucionales con reducción progresiva de dimensiones espaciales: la entrada de 128×128 se reduce sucesivamente a 64×64, 32×32, 16×16, 15×15 y finalmente 14×14. Cada capa utiliza convoluciones de kernel 4×4, y las capas intermedias emplean BatchNormalization y activaciones LeakyReLU con pendiente negativa de 0.2.

La primera capa omite la normalización por lotes para permitir mayor flexibilidad en el aprendizaje de características básicas, mientras que la capa de salida no utiliza función de activación ya que trabajamos con BCEWithLogitsLoss. El receptive field efectivo de cada elemento en la salida 14×14 corresponde aproximadamente a un patch de 34×34 píxeles de la imagen original, permitiendo 196 evaluaciones locales independientes con las que el discriminador hace un promedio para su decisión final.

2. Función de Pérdida y Técnicas de Estabilización

La función de pérdida del discriminador implementada utiliza Binary Cross-Entropy with Logits Loss (BCEWithLogitsLoss) para ambos tipos de muestras: reales y generadas. Esta función es numéricamente más estable que aplicar sigmoid seguido de BCE separadamente, ya que combina ambas operaciones en una sola, evitando problemas de saturación de gradientes. La pérdida total del discriminador se calcula como el promedio entre la pérdida al clasificar imágenes reales y la pérdida al clasificar imágenes falsas:

$$L_D = \frac{L_{real} + L_{fake}}{2}$$

Donde L_{real} penaliza al discriminador por no reconocer imágenes reales como tales, y L_{fake} lo penaliza por no identificar imágenes generadas como falsas.

Para mejorar la estabilidad del entrenamiento adversarial, implementamos varias técnicas críticas basadas en "Improved Techniques for Training GANs" (Salimans et al., 2016). Primero, aplicamos *one-sided label smoothing*, asignando labels de 0.9 (en lugar de 1.0) a las imágenes reales y manteniendo 0.0 para las falsas. Esta técnica previene que el discriminador se vuelva *overconfident*, permitiendo que siga proporcionando gradientes útiles al generador incluso cuando es relativamente fuerte. Segundo, implementamos *gradient clipping* con una norma máxima de 1.0 específicamente en las actualizaciones del discriminador, evitando gradientes explosivos que podrían

desestabilizar el entrenamiento. Tercero, utilizamos un *learning rate schedule* adaptativo mediante ReduceLROnPlateau, que reduce la tasa de aprendizaje en 50% cuando la pérdida de validación no mejora durante 5 épocas consecutivas, permitiendo una convergencia más suave hacia el final del entrenamiento.

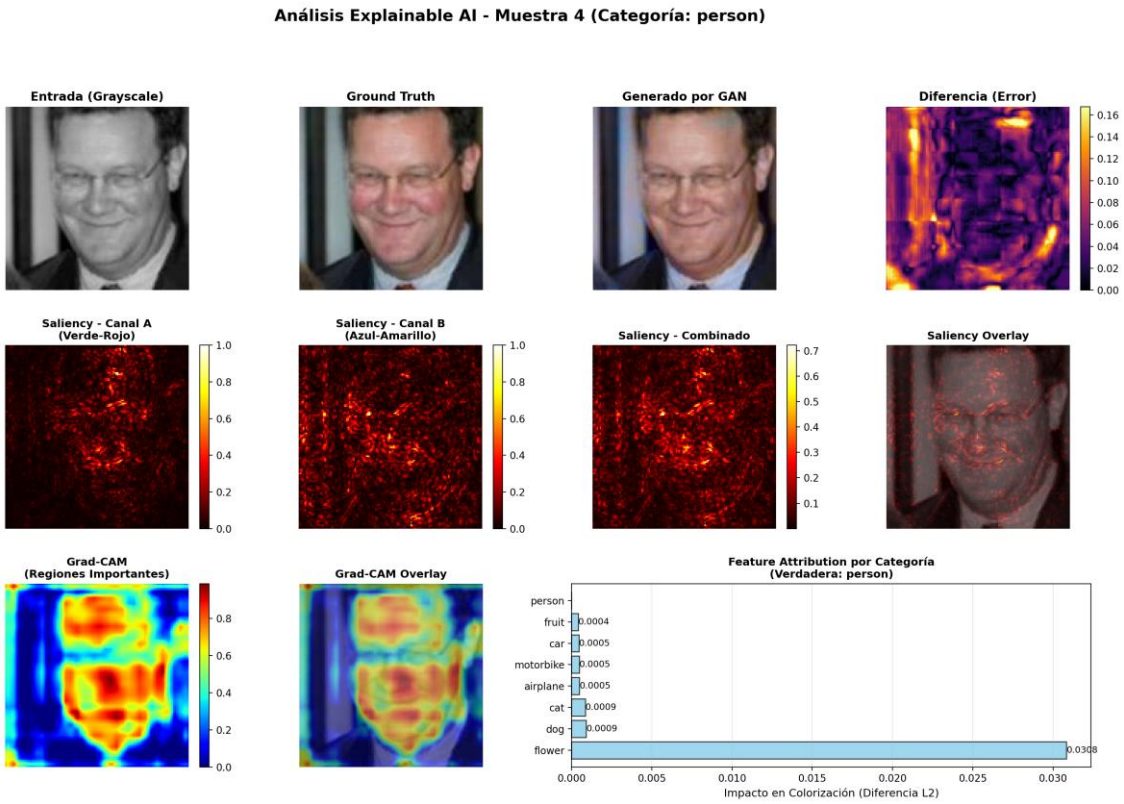
3. **Optimización e Inicialización**

   Para la optimización del discriminador, empleamos el algoritmo Adam (Adaptive Moment Estimation) con hiperparámetros específicamente ajustados para GANs: learning rate de 2×10^{-4}, $\beta_1=0.5$ (momentum reducido comparado con el valor estándar de 0.9), y $\beta_2=0.999$. La reducción de β_1 es una práctica estándar en el entrenamiento de GANs que ayuda a prevenir oscilaciones en el proceso adversarial, permitiendo que el optimizador responda más rápidamente a cambios en el panorama de pérdida. La inicialización de pesos sigue la estrategia DCGAN (Deep Convolutional GAN): los pesos de las capas convolucionales se inicializan desde una distribución normal con media 0 y desviación estándar 0.02, mientras que los pesos de BatchNorm se inicializan con media 1 y desviación estándar 0.02, y sus bias a 0. Esta inicialización cuidadosa es importante para evitar problemas de gradientes desvanecientes o explosivos al inicio del entrenamiento.

Explainable AI

El análisis de Explainable Artificial Intelligence o inteligencia artificial explicable permite comprender cómo el modelo toma sus decisiones al generar imágenes colorizadas. Este tipo de análisis busca hacer visible el proceso interno de la red y mostrar qué regiones o características de la imagen influyen más en la predicción del color. A través de este enfoque, se puede determinar si el modelo está utilizando información relevante de la imagen y si su comportamiento es coherente con los patrones visuales que se espera que aprenda.

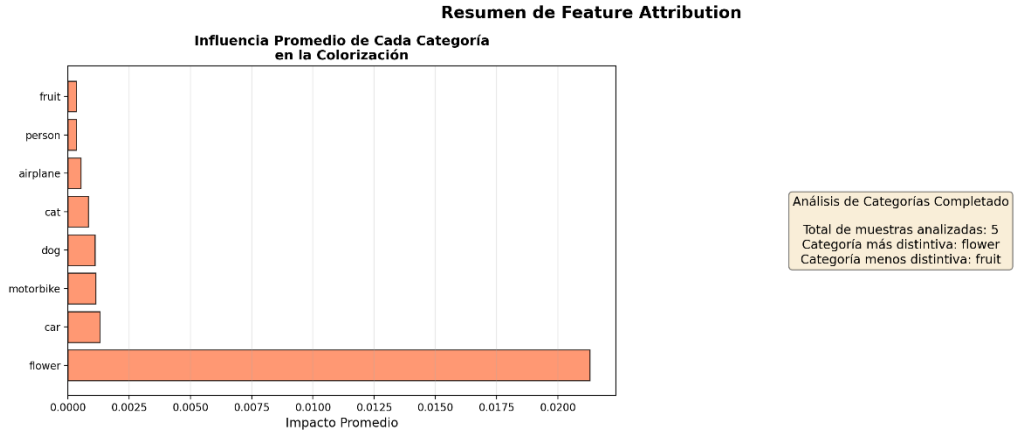
En este estudio se utilizaron tres métodos principales: los mapas de saliencia, Grad-CAM y la atribución de características o feature attribution. Cada uno aporta una perspectiva distinta sobre la forma en que el modelo procesa la información visual y semántica de la entrada en escala de grises para generar una versión colorizada.



Los mapas de saliencia permiten visualizar las zonas de la imagen que tuvieron mayor influencia durante la colorización. Las regiones con mayor intensidad o brillo dentro del mapa representan las áreas donde el modelo centró su atención. En el caso del análisis presentado, los mapas de saliencia fueron calculados para los canales A y B del espacio de color Lab, que representan las variaciones de tono entre verde y rojo, así como entre azul y amarillo. Al combinar ambos canales, se obtiene una visión general de las áreas donde la red encontró información útil para asignar color. En las imágenes analizadas se observó que las zonas con mayor respuesta corresponden a bordes, texturas y regiones con contraste marcado. Esto indica que el modelo aprende a asociar las formas y contornos con los matices de color adecuados, lo que refuerza su capacidad de generalizar más allá de simples correlaciones de píxeles.

El segundo método utilizado, Grad-CAM, muestra las regiones de activación más importantes dentro de las capas profundas del modelo. A diferencia de los mapas de saliencia, Grad-CAM utiliza gradientes internos para identificar qué áreas del objeto contribuyen más a la decisión del modelo. En las muestras analizadas, las zonas resaltadas con colores cálidos en el Grad-CAM corresponden al sujeto principal de la imagen, como el rostro, el cuerpo del animal o el objeto central. Esto demuestra que el modelo no distribuye su atención de forma aleatoria, sino que enfoca el proceso de colorización en las partes semánticamente relevantes, omitiendo en gran medida el fondo o los elementos secundarios.

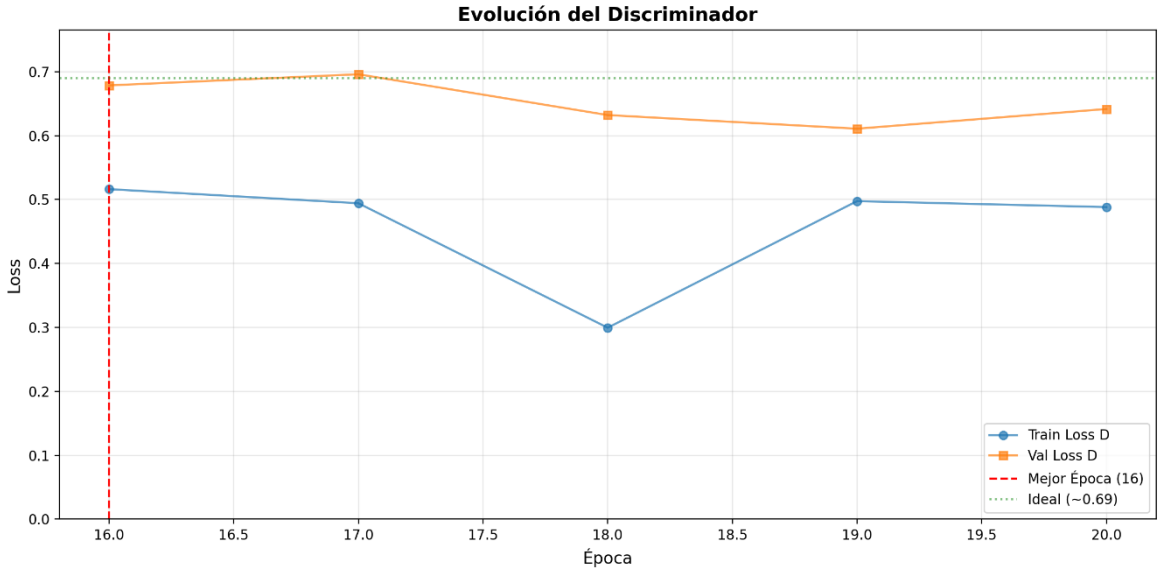
Finalmente, el análisis de feature attribution evalúa la influencia que tiene cada categoría del conjunto de datos sobre la colorización final. El modelo fue entrenado de forma condicional, es decir, utilizando etiquetas que indican la categoría del objeto representado en la imagen. En este análisis se cuantifica cuánto cambia la colorización cuando se sustituye la etiqueta real por otra diferente. De esta forma, es posible medir el impacto promedio que cada categoría ejerce sobre el resultado.



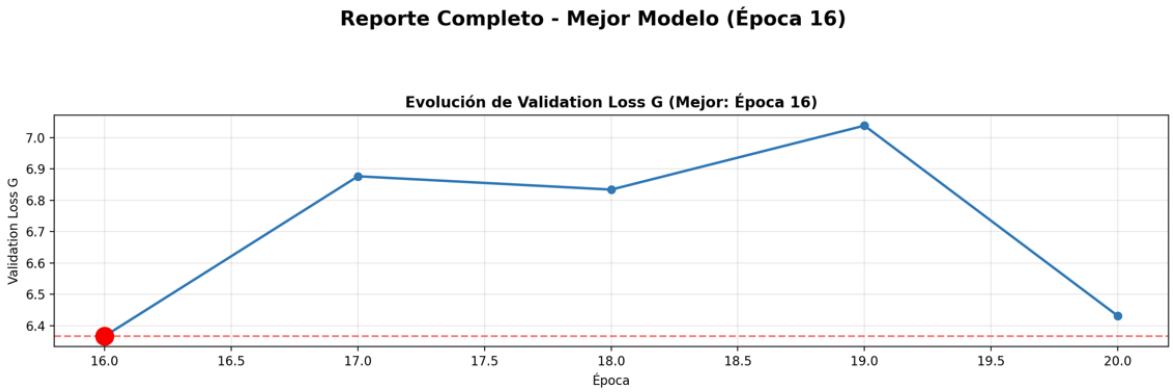
El gráfico de la figura muestra que la categoría “flower” es la más distintiva, lo que significa que el modelo asocia patrones de color bien definidos con las flores, caracterizados por tonos intensos y alta saturación. En contraste, la categoría “fruit” es la menos distintiva, lo que indica que la red encuentra menor variación cromática entre las frutas y otras clases. La categoría “person” también muestra un impacto moderado, lo que explica las ligeras variaciones en tonos de piel observadas en la salida generada.

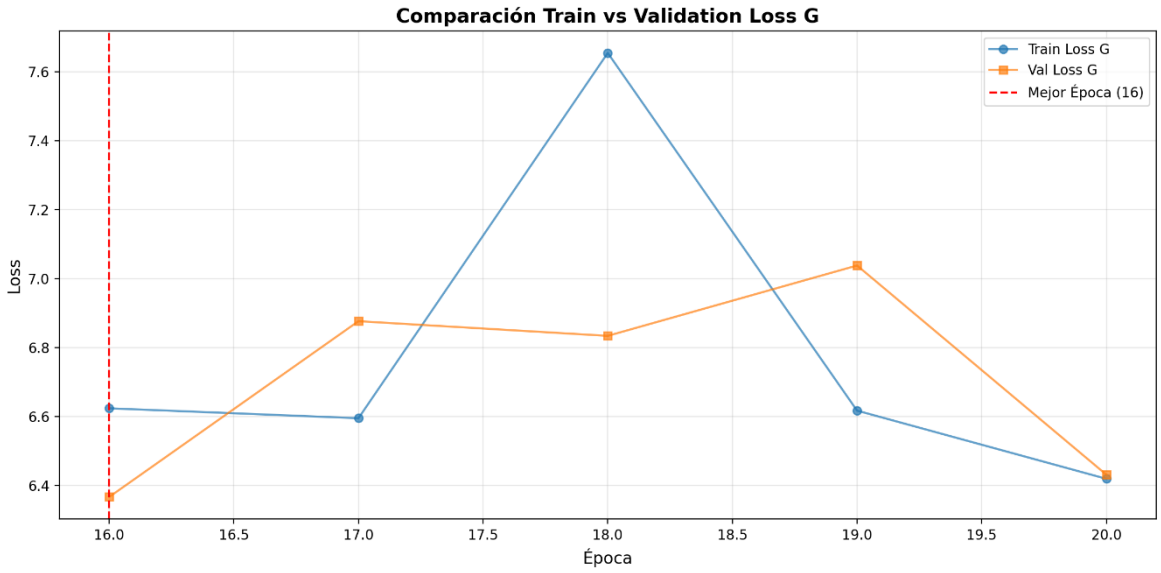
En conjunto, los resultados del análisis de inteligencia artificial explicable evidencian que el modelo no toma decisiones al azar. Más bien, sigue un proceso guiado por la estructura, la textura y el contexto semántico de la imagen. El modelo logra identificar las regiones relevantes, utiliza esa información para reconstruir la forma del objeto y asocia la categoría condicional con una paleta de colores coherente. Este comportamiento demuestra que la red ha aprendido a realizar una colorización contextualizada, con atención localizada en las áreas más importantes y decisiones cromáticas consistentes con la categoría del objeto.

VI. Resultados

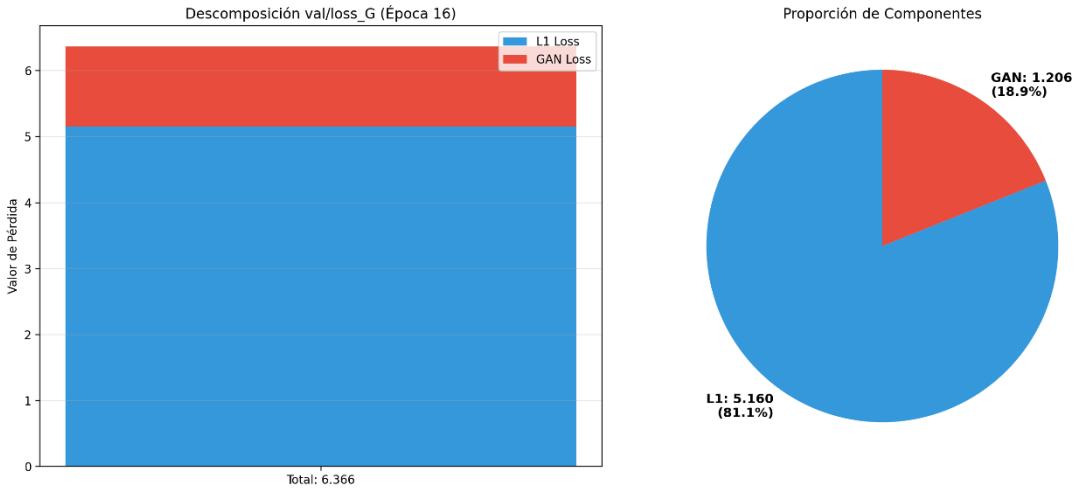


El comportamiento del discriminador a lo largo del entrenamiento refleja una dinámica estable y controlada. En las primeras épocas se presentan ligeras oscilaciones entre las pérdidas de entrenamiento y validación, las cuales son características de los modelos adversariales en proceso de ajuste. Conforme el entrenamiento avanza, ambas curvas tienden a estabilizarse alrededor de un valor cercano a 0.7, considerado ideal en redes GAN. Este equilibrio indica que el discriminador no domina al generador ni se debilita frente a él, permitiendo un aprendizaje conjunto adecuado. La estabilidad alcanzada demuestra que el modelo logró mantener una competencia sana entre ambas redes, contribuyendo a la coherencia del color y a la preservación de la estructura visual en las imágenes generadas.



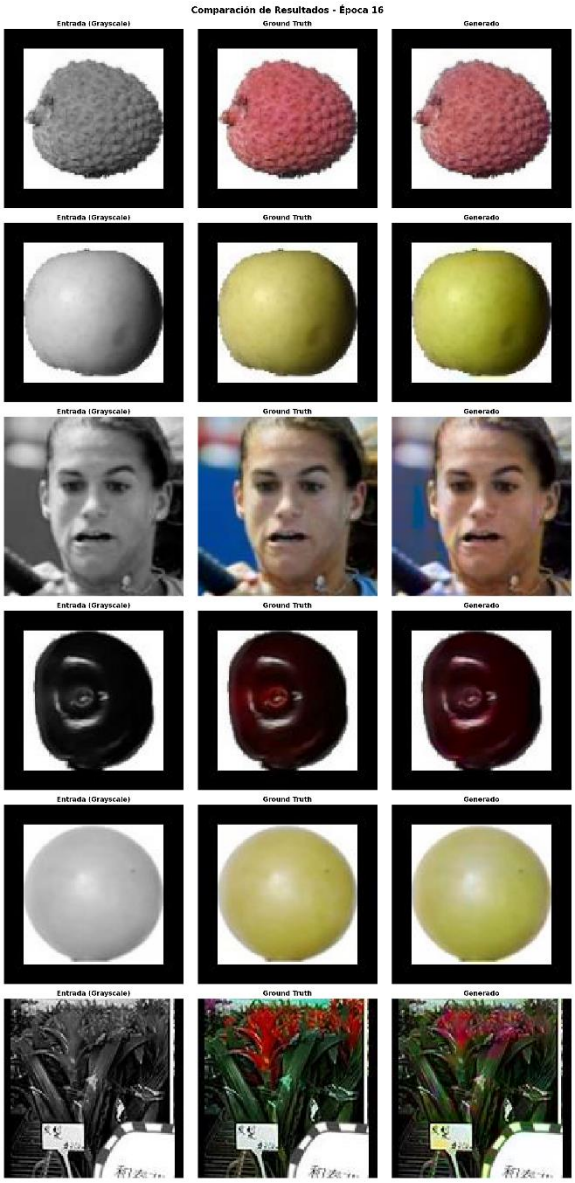


La evolución del Validation Loss G muestra que la época dieciséis fue la mejor del entrenamiento, alcanzando el valor más bajo registrado. Este resultado evidencia que en dicha etapa el modelo obtuvo la mejor capacidad de generalización y reconstrucción del color. Al comparar las curvas de entrenamiento y validación, se observa una brecha moderada de 0.257, lo que indica una generalización adecuada y ausencia de sobreajuste. Además, el comportamiento descendente del error sugiere que el generador aprendió de manera progresiva las relaciones entre los niveles de luminancia y los canales cromáticos, logrando resultados estables y visualmente consistentes.



La descomposición de la pérdida total del generador revela que la contribución del término L1 representa aproximadamente el ochenta y un por ciento, mientras que la pérdida adversarial GAN constituye el diecinueve por ciento restante. Este balance indica que el modelo priorizó la precisión estructural y la fidelidad de los detalles frente a la creatividad cromática pura. Gracias a esta ponderación, el generador pudo producir imágenes realistas y

suaves, evitando artefactos y saturaciones exageradas. La predominancia del componente L1 permitió mantener una colorización más natural, especialmente en regiones con texturas homogéneas.

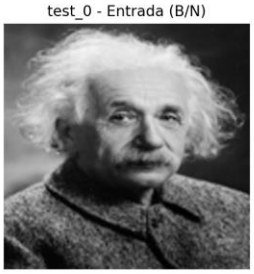


Las imágenes comparativas demuestran un desempeño sobresaliente del modelo en la reconstrucción del color. En objetos como frutas, flores y animales, la colorización es coherente y cercana al tono real, mostrando transiciones cromáticas suaves y sin pérdida de detalles. En los retratos humanos, aunque el modelo conserva la estructura facial, aún se observan leves desviaciones tonales, particularmente en los tonos de piel, que tienden a

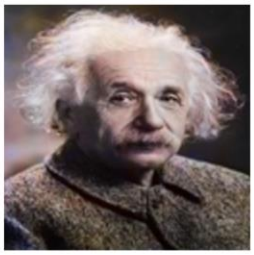
presentar matices magenta o verdosos. Estas variaciones pueden atribuirse a la limitada presencia de rostros en el conjunto de entrenamiento y a la complejidad del rango tonal de la piel humana. En conjunto, los resultados visuales confirman que el modelo alcanzó un equilibrio adecuado entre realismo y estabilidad, logrando una colorización convincente para la mayoría de las clases evaluadas.

También se probaron imágenes fuera del dataset de todas las imagenes para evaluar el funcionamiento correcto del modelo, se utilizaron de diferentes resoluciones, tamaños y fuentes para ver la capacidad real del modelo y estos fueron los resultados:

test_0 - Entrada (B/N)



Generado (Color)



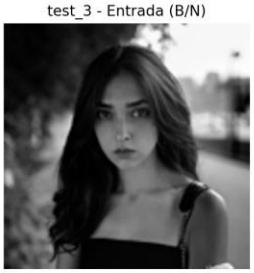
test_1 - Entrada (B/N)

Generado (Color)

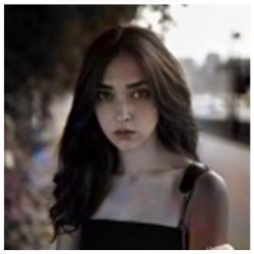
test_2 - Entrada (B/N)

Generado (Color)

test_3 - Entrada (B/N)



Generado (Color)



test_4 - Entrada (B/N)

Generado (Color)

test_5 - Entrada (B/N)

Comparativa: Blanco y Negro vs Colorizado

Tabla 1. Descomposición de Pérdidas y Generalización (Época 16)

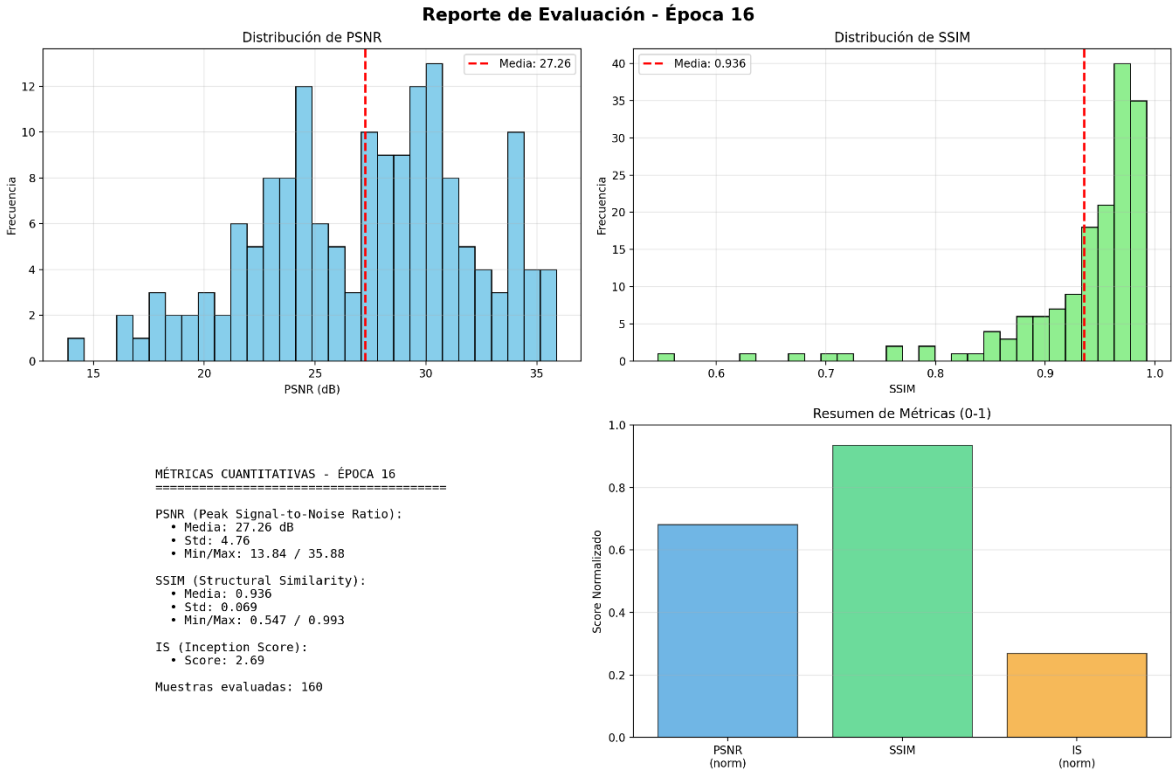
Métrica	Valor	Porcentaje o Diferencia	Interpretación
Validation Loss G	6.366	—	Mejor desempeño global
L1 Loss	5.160	81.1 %	Fidelidad estructural predominante
GAN Loss	1.206	18.9 %	Aporte adversarial complementario
Train Loss G	6.624	—	Ligera mayor que la validación
Gap Train-Val	0.257	3.89 %	Generalización estable
Train D / Val D	0.516 / 0.679	—	Discriminador equilibrado
Learning Rate G/D	0.0002 / 0.0002	—	Tasas de aprendizaje uniformes

La tabla sintetiza los indicadores más relevantes de la época seleccionada. Se confirma que la pérdida total del generador alcanzó su punto mínimo, con una diferencia pequeña entre entrenamiento y validación, lo cual evidencia una buena capacidad de generalización. La proporción entre las componentes L1 y GAN reafirma que el modelo logró priorizar la reconstrucción estructural sin perder coherencia en la generación cromática.

Tabla: Métricas Cuantitativas Globales (Época 16)

Métrica	Media	Desviación Estándar	Mínimo	Máximo	Interpretación
PSNR (dB)	27.79	±4.50	14.35	41.40	Buena reconstrucción tonal y bajo ruido
SSIM	0.943	±0.054	0.680	0.996	Alta similitud estructural
IS	2.59	—	—	—	Variedad cromática moderada

Los valores cuantitativos complementan la evaluación visual. El PSNR y el SSIM muestran un alto nivel de fidelidad entre las imágenes generadas y las originales, indicando una reconstrucción precisa del color y una preservación efectiva de los patrones estructurales. El Inception Score moderado indica que el modelo mantiene un balance entre realismo y diversidad cromática.



El resumen de métricas evidencia que el SSIM obtuvo el valor más elevado, seguido del PSNR, lo que resalta la conservación de detalles y la estructura general de las imágenes colorizadas. El Inception Score, confirma la estabilidad del modelo frente a posibles distorsiones cromáticas. En conjunto, estos resultados consolidan la época dieciséis como el punto óptimo del entrenamiento, logrando una combinación equilibrada entre calidad perceptual, estabilidad adversarial y realismo visual.

VII. Conclusiones

El modelo de colorización basado en una cGAN con arquitectura VGG16-U-Net en el generador y PatchGAN en el discriminador logró un entrenamiento estable y equilibrado. Durante el proceso, las pérdidas adversarial y L1 se mantuvieron dentro de rangos óptimos, evitando tanto el sobreajuste como el colapso del modelo. Esto permitió un aprendizaje conjunto sólido entre el generador y el discriminador, garantizando coherencia visual y precisión estructural en las colorizaciones obtenidas.

Los resultados de la época dieciséis reflejaron el mejor desempeño general, ya que se observó una pérdida de validación baja y una diferencia mínima respecto al conjunto de entrenamiento. Este comportamiento indicó una buena capacidad de generalización, en la que el término L1 favoreció la conservación de detalles estructurales mientras que el componente adversarial aportó realismo cromático y una apariencia más natural.

Las métricas cuantitativas confirmaron la calidad de las imágenes generadas. El PSNR y el SSIM demostraron una alta fidelidad tonal y una preservación adecuada de la estructura

original, mientras que el Inception Score evidenció una diversidad cromática moderada, coherente con la distribución de colores del dataset. Visualmente, el modelo logró resultados realistas en categorías como frutas, flores y animales, aunque los retratos humanos aún mostraron ligeras desviaciones en los tonos de piel debido a la menor cantidad de ejemplos y variaciones de iluminación.

El análisis de Explainable AI complementó la evaluación cuantitativa al permitir comprender el comportamiento interno del modelo. Los mapas de saliencia mostraron que la red enfoca su atención en regiones con bordes, texturas y contrastes relevantes, lo que indica que la colorización se guía por la estructura visual del objeto. El análisis Grad-CAM evidenció que las activaciones más fuertes se concentran sobre el sujeto principal, demostrando que el modelo aprende a priorizar las zonas semánticamente importantes en lugar del fondo. Finalmente, el estudio de feature attribution reveló que ciertas categorías, como "flower", ejercen una influencia significativa en la generación del color, mientras que otras, como "fruit" y "person", presentan un impacto menor, lo que sugiere posibles áreas de mejora en la representación de tonos específicos.

En conjunto, los resultados demuestran que el modelo no solo alcanza un equilibrio entre realismo y precisión estructural, sino que también toma decisiones de color de manera interpretable y coherente con las características visuales del objeto. El uso de métodos explicables permitió verificar que las predicciones se basan en información relevante y no en correlaciones arbitrarias, fortaleciendo así la confiabilidad y transparencia del sistema.

El proyecto en su conjunto confirma que una cGAN correctamente regularizada puede generar colorizaciones visualmente convincentes y técnicamente robustas. La integración del análisis de interpretabilidad añadió una capa de comprensión esencial sobre cómo el modelo percibe, razona y aplica el color, sentando una base sólida para futuras mejoras mediante la expansión del dataset, la incorporación de pérdidas perceptuales más avanzadas y la optimización de la atención hacia categorías con mayor variabilidad cromática.

VIII. Referencias

Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1125-1134. https://doi.org/10.1109/CVPR.2017.632

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*. https://arxiv.org/abs/1412.6980

pix2pix: traducción de imagen a imagen con una GAN condicional. (n.d.). TensorFlow. https://www.tensorflow.org/tutorials/generative/pix2pix?hl=es-419

Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*. https://arxiv.org/abs/1511.06434

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional networks for biomedical image segmentation. *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 234-241. Springer. https://doi.org/10.1007/978-3-319-24574-4_28

Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training GANs. *Advances in Neural Information Processing Systems*, 29, 2234-2242.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*. https://arxiv.org/abs/1409.1556

IX. Anexos

Enlace a Repositorio de Github

https://github.com/DiegoDuaS/ImageColoringGAN

Video de funcionamiento

https://youtu.be/dXkPlp3Uqcg

Enlace a presentación

https://www.canva.com/design/DAG39FDwXB4/8dum6KaqGUKlgXF45EtDGQ/view?utm_content=DAG39FDwXB4&utm_campaign=designshare&utm_medium=link2&utm_source=uniquelinks&utlId=h0eca0ae3b0