



Universidad de
San Andrés

I302 – Aprendizaje Automático
y Aprendizaje Profundo

Reviviendo Recuerdos Familiares con Pix2Pix y DnCNN

Antonio Santiago Tepsich

atepsich@udesa.edu.ar

Máximo Gubitosi

mgubitosi@udesa.edu.ar

Motivación

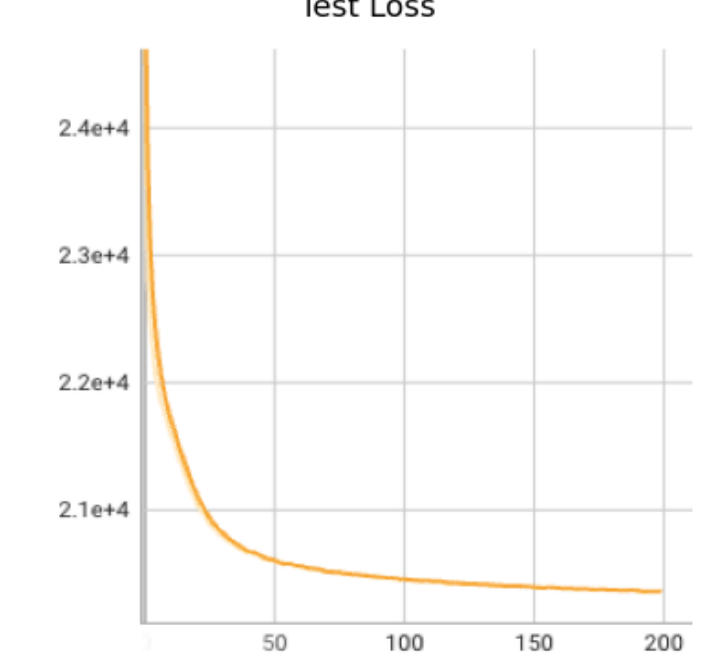
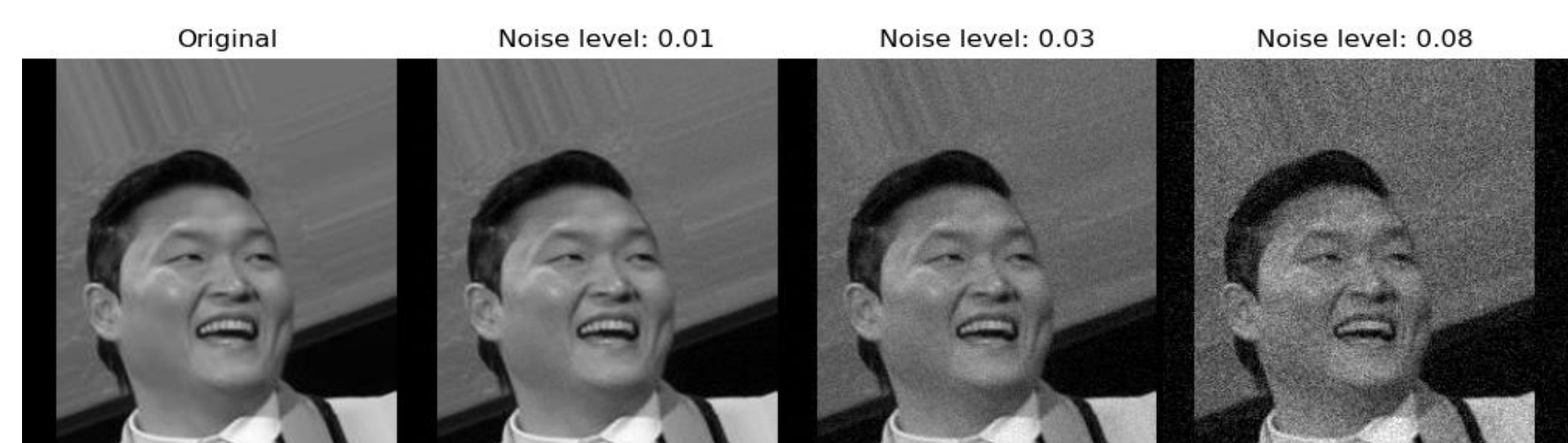
En la intersección de los recuerdos y la innovación tecnológica, las personas siempre buscamos mantener vivos distintos momentos. Las maneras de hacerlo fueron evolucionando, desde el uso de lienzos a las fotografías. Sin embargo, las limitaciones tecnológicas de la época relegaron muchos de estos recuerdos a escalas de grises, y los colores del momento original quedaron olvidados. Por eso el objetivo de este trabajo es revivir el color de nuestras Imágenes Familiares.

Discusiones

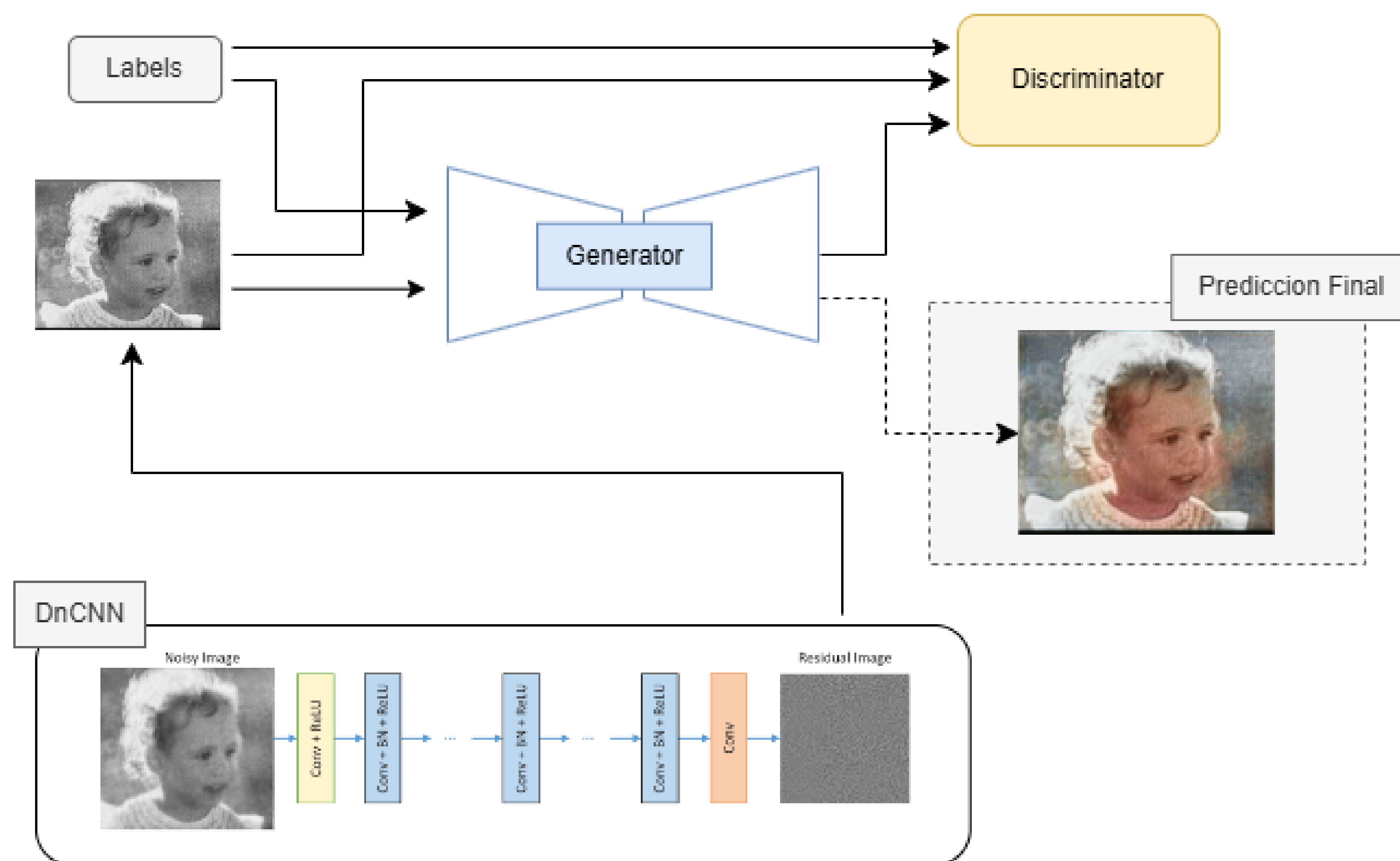
La DnCNN se usó para reducir el ruido y mejorar las imágenes de nuestro dataset. Sin embargo, esto no aumenta la resolución, por lo que persiste una diferencia de calidad entre estas imágenes reales y las utilizadas para entrenar el modelo.

Entrenando el Colorizador con un dataset variado se obtienen mejores resultados generales, mientras que utilizando un dataset con imágenes similares, proporciona un rendimiento superior en ese tipo específico de imágenes.

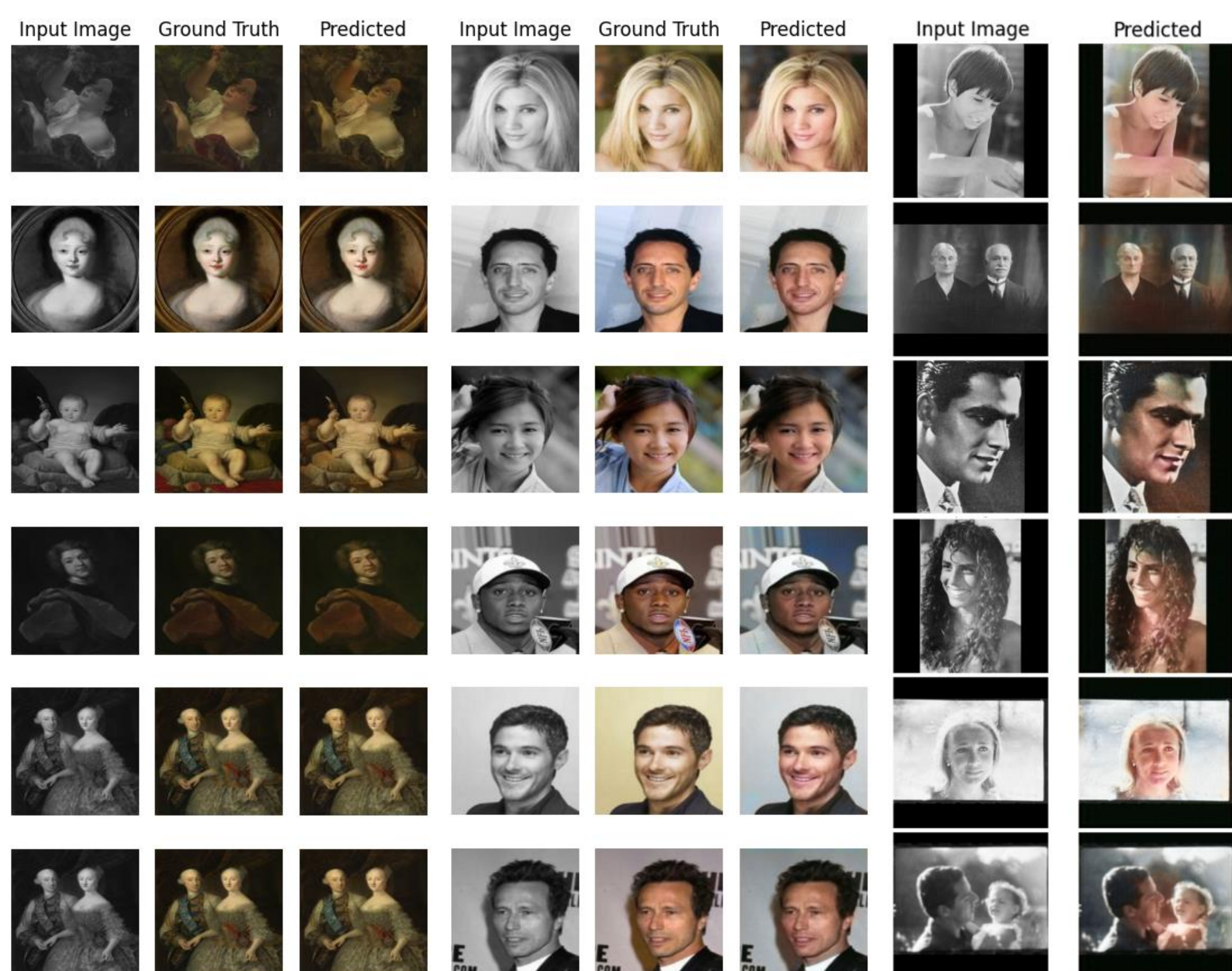
DnCNN



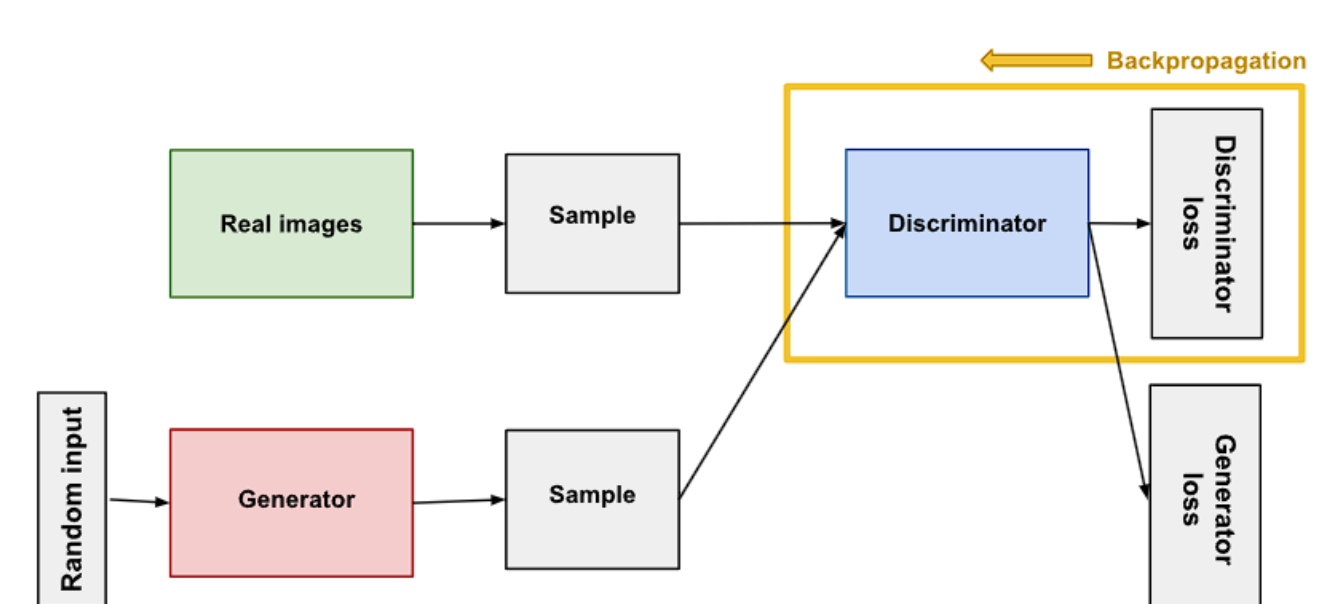
Arquitectura del Sistema



Resultados y Análisis



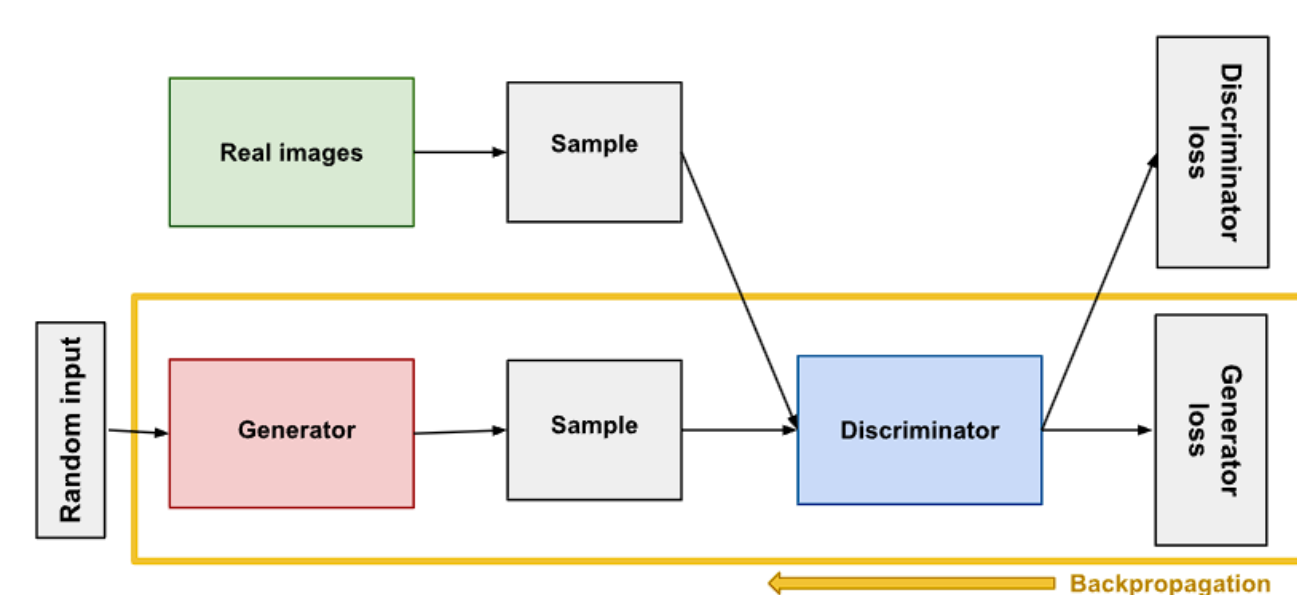
Pix2Pix



$$Loss_{disc} = -[E_{x \sim p_{data}(x)} \log D(x) + E_z \log(1 - D(G(z)))]$$

El Generador aprende a crear imágenes incorporando Feedback del Discriminador, con el objetivo de que este clasifique su salida como Real. El generador entrena creando nuevas imágenes, que luego son clasificadas por el Discriminador. La Loss del generador penaliza por no engañar al discriminador, y basada en esta clasificación, hace Backpropagation a través de ambas redes, actualizando únicamente los pesos del Generador.

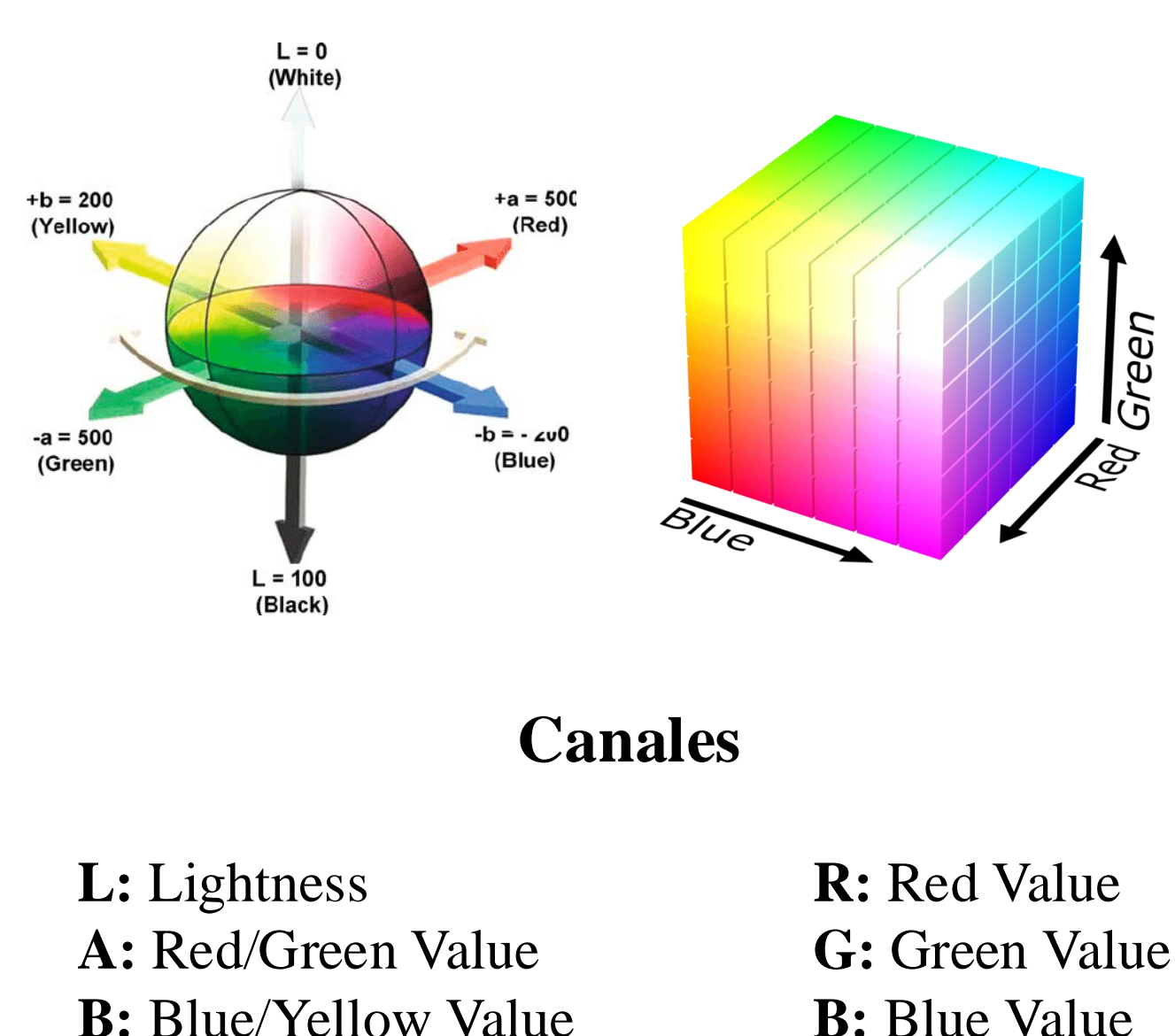
El Discriminador es un clasificador que intenta distinguir entre imágenes reales y creadas por el generador. En nuestro caso es una PathGAN y durante su entrenamiento, el generador mantiene sus pesos mientras genera las imágenes. La Loss del Discriminador lo penaliza por clasificar incorrectamente una instancia real como falsa o viceversa; y actualiza sus pesos a través de Backpropagation basada en esta Loss.



$$Loss_{gen} = -E_z \log(D(G(z))) + \lambda * E_z \|y - G(z)\|$$

$y = img. a \ color$, $z = img. en \ blanco \ y \ negro$

LAB vs RGB



Datasets

Dataset Propio: 84 imágenes antiguas digitalizadas de dimensiones y resolución variable de nuestras familias, en escala de grises.

CelebA Dataset: 202.599 imágenes de las caras de 10.177 celebridades.

Image Colorization Dataset: 25.000 imágenes variadas de tamaño 224x224 en formato LAB

Art Images, Paintings: 2186 imágenes únicamente a color y con tamaño variable

Data Augmentation

Buscamos que el modelo tenga un set de imágenes lo suficientemente diferentes para que no haga overfitting, pero sin que las imágenes originales sean inentendibles

- **Resize por medio de Interpolación:** pasamos del size original a 256x256
- **Ampliación + Cropping:** aumentamos las imágenes a 286x286 y recortamos de manera aleatoria un área de 256x256
- **Random Jitter:** de manera aleatoria la función elige si espejar o no la imagen

Entrenamiento



Trabajo Futuro

- **Utilizar LAB*** en el modelo Pix2Pix.
- **Reemplazar la arquitectura cGAN** por una CycleGAN o WGAN.
- Mejorar las **generaciones del fondo** de las imágenes agregando al entrenamiento un dataset específico de paisajes.
- Utilizar un **modelo pre-entrenado** con ImageNet (ej: ResNet16) y entrenarlo como generador.
- Utilizar además de la DnCNN para denoising, un **modelo de aumento de resolución**, como una SRGAN, ESRGAN o RCAN.