



**Inteligencia Artificial, aprendizaje
automático y deep learning**

Integrantes:

Miguel Ángel Lenó Cárdenas - 2200644

Docente:

Gilber Alexis Corrales Gallego

**Facultad de Ingeniería ~ Ingeniería
Multimedia.**

**Redes Neuronales Artificiales y Deep
Learning
2024**

Resumen

En este trabajo, se presenta la implementación de un modelo de redes generativas adversariales condicionales (cGAN) basado en Pix2Pix, para la traducción de imágenes de siluetas de autos a imágenes a color con detalles. Se detalla la arquitectura del modelo, destacando la estructura específica de su generador y discriminador. A través de experimentos y ajustes en el procesamiento de datos y los parámetros de entrenamiento, el modelo logra resultados efectivos, aunque existen desafíos en cuanto a la precisión y calidad de los detalles generados.

Introducción

Las redes generativas adversariales condicionales (cGAN) han demostrado ser muy eficaces en la traducción de imágenes a imágenes. Estas redes aprenden a mapear una imagen de entrada a una imagen de salida, generando resultados en aplicaciones como colorización de imágenes en blanco y negro, generación de imágenes realistas a partir de bocetos, entre otros. Este proyecto explora la implementación de Pix2Pix, un modelo cGAN desarrollado específicamente para tareas de traducción de imágenes, para transformar imágenes de siluetas de autos en imágenes a color con detalles.

Marco Teórico

2.1 Redes Generativas Adversariales (GAN)

Las GAN están compuestas por dos redes neuronales en competencia: el generador, que intenta crear imágenes lo más realistas posibles, y el discriminador, que intenta distinguir entre imágenes reales y generadas. Este marco competitivo

permite que el generador aprenda a producir imágenes cada vez más precisas.

2.2 Redes Generativas Adversariales Condicionales (cGAN)

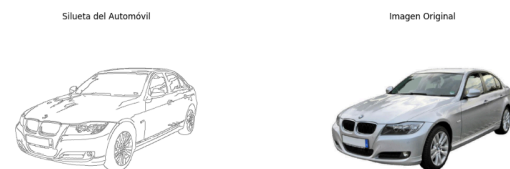
En una cGAN, el generador está condicionado a una entrada específica, lo cual permite controlar la salida. Esto es especialmente útil en tareas de traducción de imágenes, donde existe una relación directa entre la entrada y la salida. Pix2Pix utiliza este enfoque para aprender una correspondencia entre una imagen de entrada y una imagen de salida deseada.

2.3 Pix2Pix

Pix2Pix es una cGAN que utiliza un generador basado en U-Net y un discriminador PatchGAN. La arquitectura de U-Net ayuda al generador a preservar características de la imagen original mediante conexiones de salto, mientras que PatchGAN permite al discriminador evaluar la coherencia en pequeños parches de la imagen generada, en lugar de en toda la imagen.

Descripción del Problema

El objetivo de este proyecto es transformar imágenes de siluetas de autos en escala de grises en imágenes detalladas a color. Esta tarea presenta desafíos específicos debido a la necesidad de preservar la estructura y forma del auto en la silueta mientras se agregan detalles y colores realistas en la imagen generada. El problema es de particular interés en aplicaciones de diseño y simulación de automóviles.



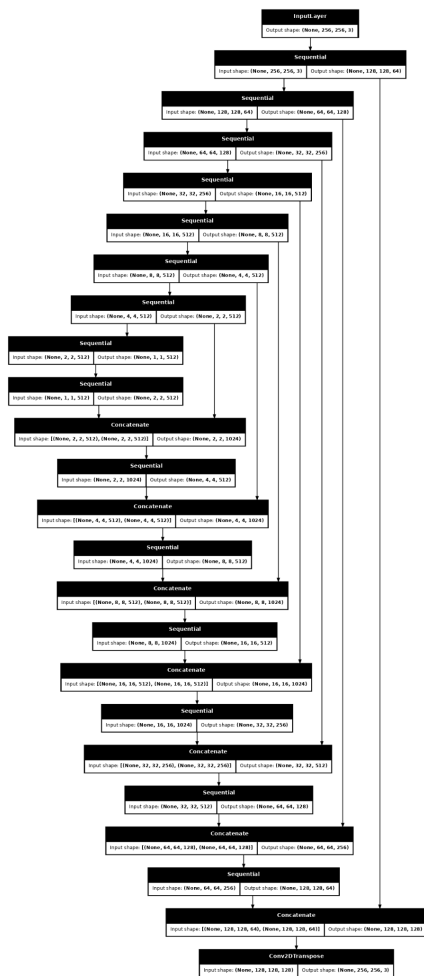
Planteamiento de la Solución

4.1 Organización y Preprocesamiento del Dataset

El dataset se organizó en dos carpetas: AutosIN para las imágenes en escala de grises y AutosOUT para las imágenes a color. Las imágenes se alinearon para asegurar que el modelo pudiera aprender una correspondencia uno a uno. Hubo desafíos iniciales con los formatos de imagen (PNG y JPEG) y se tuvo que convertir todas las imágenes a PNG para asegurar compatibilidad.

4.2 Arquitectura del Generador

El generador de Pix2Pix está basado en una U-Net, que es una red neuronal convolucional (CNN) especialmente diseñada para tareas de segmentación y traducción de imágenes. A continuación, se detalla cada componente:



Estructura del Generador U-Net

Capas de Codificación (Downsampling):

Estas capas reducen progresivamente la resolución de la imagen, capturando características de alto nivel.

Conv2D: Cada capa convolucional utiliza un filtro con tamaño 4x4 y un "stride" de 2, lo cual reduce la resolución de la imagen en un 50% en cada paso.

Batch Normalization: Normaliza la activación en cada paso para mejorar la estabilidad y la convergencia.

Leaky ReLU: La activación Leaky ReLU permite que las neuronas tengan una pequeña pendiente negativa, ayudando al flujo de gradientes.

Capas de Decodificación (Upsampling): Estas capas restauran la resolución original de la imagen utilizando convoluciones transpuestas.

Conv2DTranspose: Se utilizan filtros 4x4 para realizar el upsampling.

Batch Normalization y ReLU: Cada capa de upsampling tiene una normalización por lotes y utiliza ReLU como función de activación.

Dropout: Se aplica dropout en algunas capas para evitar el sobreajuste.

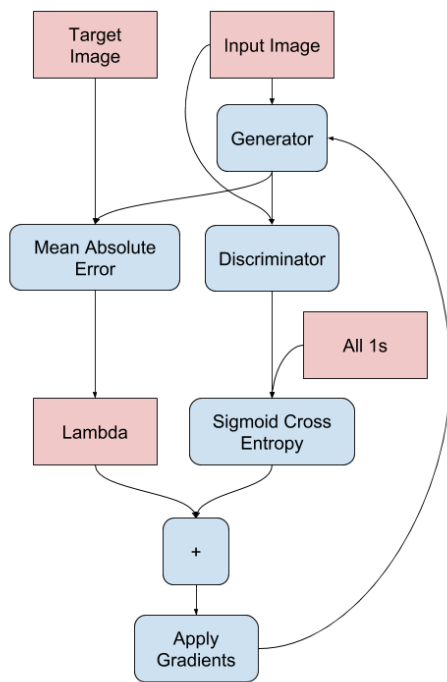
Conexiones de Salto (Skip Connections): Cada capa del codificador se conecta directamente con su contraparte en el decodificador. Esto permite que las características de nivel bajo de la imagen de entrada se mantengan, lo cual es crítico para preservar detalles en la imagen de salida.

Estructura del Generador en Números:

La red consta de 8 capas de codificación y 7 de decodificación.

Cada capa de downsampling aumenta el número de filtros: 64, 128, 256, 512, hasta 1024 en la capa más profunda.

En las capas de upsampling, el número de filtros disminuye de manera simétrica.

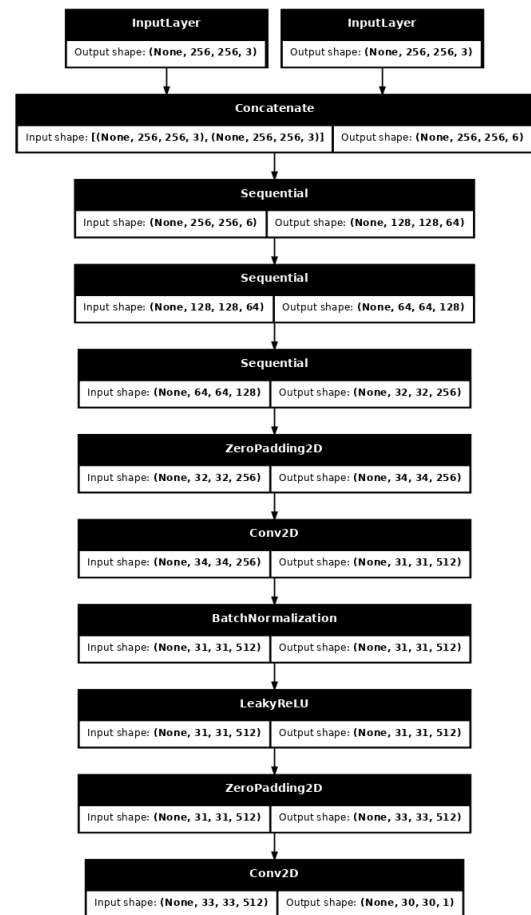


4.3 Arquitectura del Discriminador

El discriminador en Pix2Pix utiliza una arquitectura PatchGAN, diseñada para clasificar pequeños parches de la imagen como reales o generados en lugar de evaluar la imagen completa. Esta estrategia permite evaluar la coherencia local en la imagen.

Estructura del Discriminador PatchGAN

Concatenación de Imágenes: La imagen de entrada y la imagen generada se concatenan en el canal de profundidad, creando una entrada de tamaño (256, 256, 6).



Capas Convolucionales:

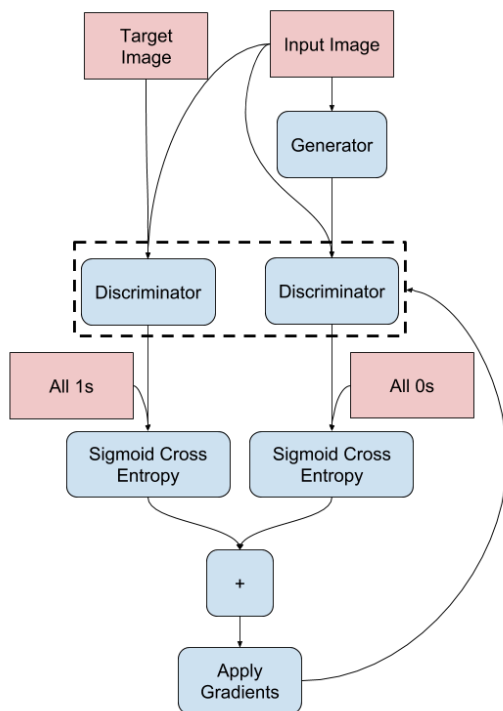
Cada capa convolucional tiene filtros de tamaño 4x4 y "stride" de 2, lo cual reduce la resolución.

La primera capa tiene 64 filtros, y este número se incrementa a 128, 256 y finalmente 512 en las capas posteriores. Cada capa incluye normalización por lotes y activación Leaky ReLU.

Evaluación de Parches:

La salida final es un mapa de probabilidad de (30, 30, 1), donde cada celda representa un parche de la imagen y su probabilidad de ser real.

Este diseño permite que el discriminador se enfoque en detalles locales, mejorando la calidad de la imagen generada.



4.4 Función de Pérdida

La función de pérdida en Pix2Pix combina:

Pérdida GAN: Una pérdida de entropía cruzada binaria que mide la habilidad del generador para engañar al discriminador.

Pérdida L1: Calcula el error absoluto medio entre la imagen generada y la imagen real, ayudando a que la imagen generada sea estructuralmente similar a la imagen de destino.

Resultados

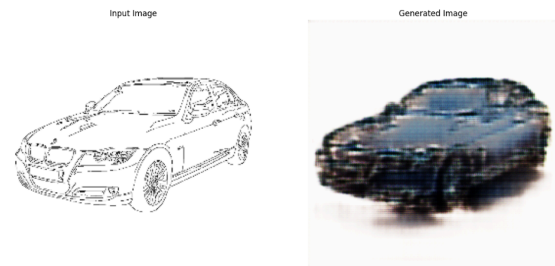
El modelo fue entrenado en Google Colab utilizando TensorFlow y Keras. A continuación, se presentan las métricas de pérdida promedio en el conjunto de prueba:

Pérdida del Generador: 78.9545

Pérdida del Discriminador: 1.5477

Ejemplos de Imágenes Generadas

Se incluyen ejemplos de imágenes generadas a partir de siluetas de autos en escala de grises (ver figuras). Las imágenes muestran un buen nivel de detalle, aunque se observan ciertos problemas de precisión en las áreas de contorno y detalles finos.



Problemas Encontrados

Error de Formato: Las imágenes en formatos distintos (JPEG y PNG) causaron problemas en la carga de datos. Este problema se resolvió convirtiendo todas las imágenes a PNG.

Alineamiento de Imágenes: La falta de alineamiento perfecto entre las imágenes de entrada y las de salida afectó la calidad de los resultados.

TensorBoard: Inicialmente, hubo problemas para visualizar las métricas en TensorBoard.

Esto se solucionó configurando correctamente la ruta de los archivos de registro.

Conclusiones

El modelo Pix2Pix es efectivo en la traducción de imágenes de siluetas a imágenes detalladas, y ofrece un enfoque prometedor para aplicaciones en diseño y simulación automatizada. Los resultados obtenidos destacan la importancia de un buen preprocesamiento de datos y de configuraciones de arquitectura.

adecuadas. En futuros trabajos, se podrían investigar técnicas adicionales de ajuste de hiperparámetros y refinamiento de detalles.

LINK DEL PROYECTO EN GOOGLE COLAB:

https://colab.research.google.com/drive/1Uyb2bR0v7kktsOFkrq40c_D5N_49TI5p?usp=sharing

Referencias

- Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-Image Translation with Conditional Adversarial Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 5967-5976).
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In Advances in neural information processing systems (pp. 2672-2680).