

UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería

Deep Learning – Alberto Suriano



Proyecto. Generación de imágenes pixel art

Daniel Alfredo Rayo Roldán, 22933

Flavio André Galán Donis, 22386

Links Importantes

Repositorio: <https://github.com/DanielRasho/FaceToPix>

Video: <https://youtu.be/7SMqa37SH2o>

Datasets: [PaperData](#)

<https://docs.google.com/presentation/d/1jO-GV4ParFwhhSP4bvVJ-4vNZPRVTzqmBOhZ-M3cUHQ/edit?slide=id.p#slide=id.p>

Descripción del problema

Con el renacimiento de los videojuegos indies en la última década se ha vuelto a expresar interés en una forma de arte usada extensamente hace unas décadas: el pixelart, un estilo artístico resultado de las carencias tecnológicas de la época (1980-1995), donde las pantallas ofrecían bajas resoluciones y una pequeña gama de colores (Kordic, 2025). Como indica Kordic en su artículo parte de las razones por el renacimiento del pixelart parte de la nostalgia hacia aquellos juegos de la infancia sino sobre todo por su minimalismo, haciéndolo perfecto de desarrollar para estudios pequeños.

Sin embargo, a pesar de eso sigue siendo un método que requiere de un artista con experiencia y tiempo para ser llevado ejecutado, condiciones que muchas veces no son deseadas al inicio de desarrollo donde se quieren prototipar las ideas principales y explorar el estilo de arte que se utilizará.

Análisis

Hacer pixelart no es tan simple como seleccionar una imagen y reducir su resolución a 32 x 32px, al ser un proceso durante el cual se pierde mucha información, y donde se debe decidir qué información abstraer y cual eliminar, características que un algoritmos de escalado no tiene en cuenta.

Ante ello se requiere una solución que sea capaz de extraer las características de una imagen original y ser capaz de transformarla a su versión pixelart manteniendo la esencia de la imagen original, una tarea que podría ser llevada a cabo por un red neuronal profunda. Sin embargo, el principal problema son los datos para entrenar una red de este tipo, en la comunidad del pixelart rara vez se realiza una obras que buscan ser la traducción de una obra ya existente, es decir, el contar un conjunto de datos de traducciones uno a uno no es viable, por tanto se necesita una arquitectura capaz de generar imágenes pixelart sin contar con un ejemplo objetivo con el que comparar.

Propuesta de solución

Para resolver el problema descrito se optó por implementar un modelo generativo que fuera capaz de convertir imágenes de rostros anime a pixelart, con la intención de ser un primer paso para poder generar imágenes de diferentes fuentes. Ante ello se encontraron 2 arquitecturas prometedoras que se explicaran a continuación.

CycleGAN

Hace aproximadamente una década Jun-Yan Zhu, et. al. (2017) publicaron una variantes de las conocidas redes generativas adversarias, también conocidas como GAN, en aquel momento Jun-Yan estaba explorando un problema similar de poder transformar imágenes de un dominio a otro, en concreto, transformar imágenes de caballos en cebras, a partir de conjuntos de datos no relacionados.

La variante consistía en utilizar dos redes adversarias que se apoyaban entre sí, la idea en síntesis partía de la naturaleza de las redes GAN de transformar entradas de un dominio X a otro Y de acuerdo con lo que dicta el Discriminador de la red, esto puede ser suficiente en problemas donde lo único que importa es conseguir la forma del dominio destino Y , pero insuficiente cuando se quiere preservar información sobre los datos originales; para solventarlo se agrega una segunda red GAN, cuyo trabajo es revertir lo hecho por la primera GAN, transformando una imagen de dominio Y a X como se puede observar en la siguiente figura.

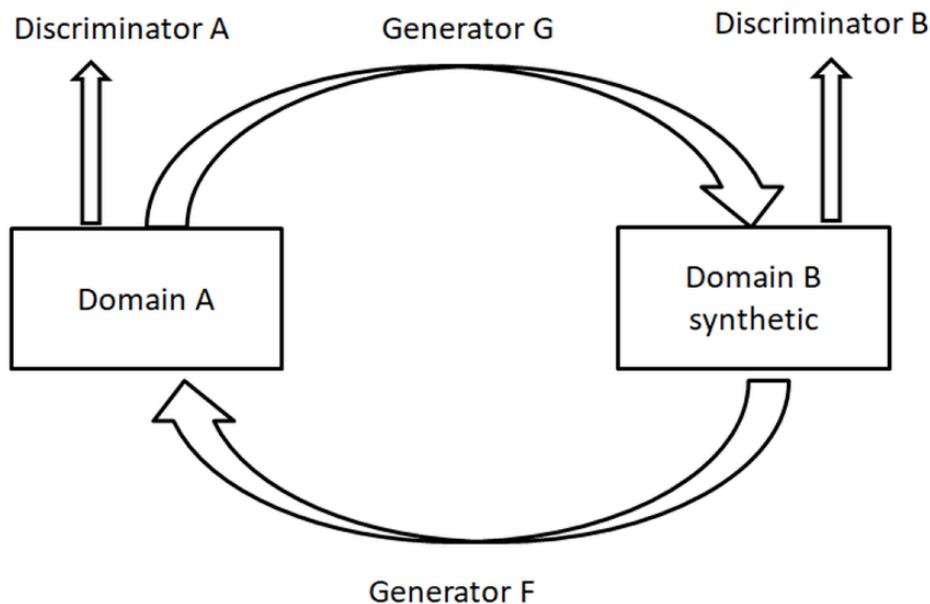


Figura 1. Diagrama general de la arquitectura de una red CycleGAN (Lan Lan, et.al , 2010)

De esta manera no solo se busca traducir de un dominio a otro sino también se busca enseñar a la red que solo que debe generar imágenes que al ser revertidas sean idénticas a la imagen general, es decir mantener el contenido. Y al ser redes independientes, también pueden

manejar conjuntos de datos no relacionados. Por su naturaleza cíclica de entrenamiento fueron llamadas CycleGAN.



Figura 2. Ejemplo de transformación de caballos a cebras utilizando una CycleGAN. Extraído de Zhu, et.al. 2017

A continuación se describe la función de pérdida con la que se ajustan a ambos generadores.

Pérdida de los generadores

$$L = L_{GAN X \rightarrow Y} + L_{GAN Y \rightarrow X} + L_{Consistencia}$$

Donde:

$L_{GAN X \rightarrow Y}$ = Error de generar imagen del comunito X al Y dada por el discriminador.

$L_{GAN Y \rightarrow X}$ = Error de generar imagen del comunito Y al X dada por el discriminador.

$L_{Consistencia}$ = Error dado por comparar la imagen revertida con la imagen original.

Visto de otra forma es entrenar algo similar a un Encoder - Decoder.

Redes Style Transfer

Como segundo candidato se encuentran las redes style-transfer que también parten de un método de entrenamiento poco convencional. La idea en síntesis parte de que toda imagen contiene 2 componentes : el contenido y el estilo, siendo el primero el que guarda los conceptos esenciales de la imagen, mientras que el segundo los detalles distintivos (Gatys & Ecker & Bethge, 2015).

La idea de este tipo de redes generativas es dadas 2 imágenes, generar una imagen que tenga la fusión del contenido de una imagen y el estilo de otra. Para ello la función de error está dada por :

$$L_{total} = L_{estilo} + L_{contenido}$$

Pérdida de contenido

Se parte de una red de clasificación pre entrenada, comúnmente una VGG, se empieza con una imagen generada por ruido aleatorio, que es pasada en la red VGG, y se observa sus activaciones en la última capa de convolución, luego se hace lo mismo con la imagen original. El error de contenido es la diferencia entre las activaciones de la imagen real y la generada (Gatys & Ecker & Bethge, 2015).

Pérdida de estilo

Se supone que el estilo se encuentran en detalles específicos esparcidos a lo largo de la imagen, estas características están esparcidas a lo largo de diferentes capas de convolución. Así que en este caso se extraen las activaciones de las diferentes capas y canales, obteniendo una lista de tensores, cada uno de estos canales guarda información de alguna característica. Las características normalmente están relacionadas, así que se revisa la correlación entre las diferentes características usando un producto punto, dando lugar a un mapa de Gramm. Se extraen los mapas de Gramm para la imagen original y generada y la diferencia es únicamente la pérdida de estilo (Gatys & Ecker & Bethge, 2015).



Figura 3. Ejemplo de transferencia de estilo utilizando redes de Style-Transfer . Extraido de Gatys & Ecker & Bethge, . 2015

En este tipo de red cada vez que se genera una nueva imagen, la red base debe ser calibrada a las 2 nuevas imágenes de entrada, y el entrenamiento consiste en reducir los errores de estilo y contenido, que solamente dependen de las imágenes de entrada, así que no requieren de conjuntos de datos y funcionan con imágenes desparejadas.

En el presente reporte se decidió implementar ambas, para ver su desempeño de las redes GAN y Style-Transfer para la generación de imágenes pixelart a partir de rostros anime.

Descripción de solución

Se extrajeron aproximadamente 5000 imágenes pixelart de 3 fuentes diferentes:

- <https://pixeljoint.com/> : caracterizado por tener imágenes muy pequeñas.
- <https://lospec.com/gallery/> : alta variedad de estilos de arte.
- <https://pixie.haus/gallery> : imágenes con un estilo pixelart parecido, la mayoría con transparencia incluida.

Y aproximadamente 20,000 imágenes de rostros anime, todas de 128 x 128 píxeles. Dado que las imágenes de pixel-art eran de diferentes tamaños, fueron reescaladas antes de entrar a la red para tener el tamaño de 128 x 128. Realmente durante el entrenamiento solamente se llegaron a usar unas 1000 imágenes de cada conjunto de datos, dado que se descubrió que después de las 20 épocas la presentación de varios ejemplos a la red disminuyó su rendimiento (en el caso de la CycleGAN).

En cuanto a los modelos, se implementaron ambos modelos; el CycleGAN fue probado contra los 3 diferentes conjuntos de datos, mientras que el style transfer con imágenes aleatorias de cualquier dataset. A continuación se muestra la arquitectura a gran nivel de ambas.

Arquitectura Cycle Gan

Generador		
Input	128 x 128 x 3	
Convolución	128 x 128 x 64	Downscaling
Convolución	64 x 64 x 128	
Convolución	32 x 32 x 256	
9 capas residuales	32 x 32 x 256	
Convolución	32 x 32 x 256	Upscaling
Convolución	64 x 64 x 128	
Convolución	128 x 128 x 64	
Tahn	128 x 128 x 3	

Discriminador	
Input	128 x 128 x 3
Convolución	64 x 64 x 64
Convolución	32 x 32 x 128
Convolución	16 x 16 x 256
Convolución	8 x 8 x 512
Zero Padding	9 x 9 x 512
Convolución	8 x 8 x 1

El discriminador sigue el formato de PatchGAN, es decir que agrega conclusiones diferentes secciones de la imagen, en vez de una conclusión general de toda la imagen (Gozde Unal, 2018).

Arquitectura Style-Transfer

Modelo Base	VGG-19 (16 capas de convolución)
Capas de Estilo	primeras 5 capas de convolución en sus primeros canales
Capa de Contenido	4ta capa de convolución en su segundo canal.
Iteraciones	10,0000 por par de imágenes.

Herramientas aplicadas

Extracción de imágenes

- Node : Lenguaje de programación
- Axios : Descarga de imágenes
- Cheerio : Parseo de archivos HTML

Entrenamiento de la red

- Pytorch : Framework para la creación de modelos.
- PIL: Preprocesamiento de imágenes.
- Matplotlib: Generación de gráficos.
- Numpy : Manipulación de Tensores.

Resultados

Los resultados para ambos modelos no fueron totalmente satisfactorios, los modelos GAN se caracterizaban por ser buenos innovando con paletas de colores, y agregar

CycleGAN

Probando con diferentes datasets y tamaños de épocas se descubrió que alrededor de la época 20 con baches de 150 imágenes los discriminadores ya habían asimilado la esencia de ambos estilos, mientras que el error total de las redes generadoras había bajado abruptamente y a partir de ahí su error disminuye lentamente.

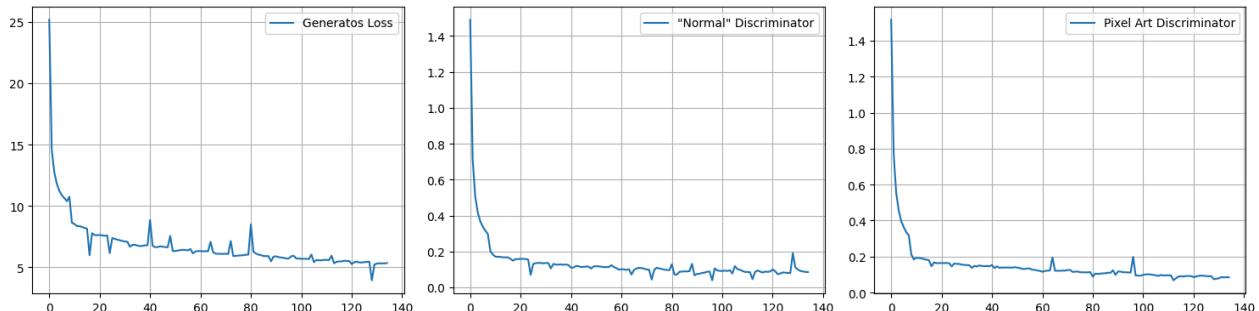


Figura 4. Curvas de aprendizaje de la red generativa CycleGAN a lo largo de las épocas.. Elaboración propia.

Sin embargo, a pesar de la pérdida evidente de error a lo largo de las épocas, las imágenes generadas no evocaban al arte pixel art, en todo caso la red había aprendido un método que el pixelart consiste en un patrón de baldosas con colores intermitentes, dando algunas imágenes aceptables mientras en otros casos resultados muy exagerados, y esto se hacía más evidente con el paso de las épocas.

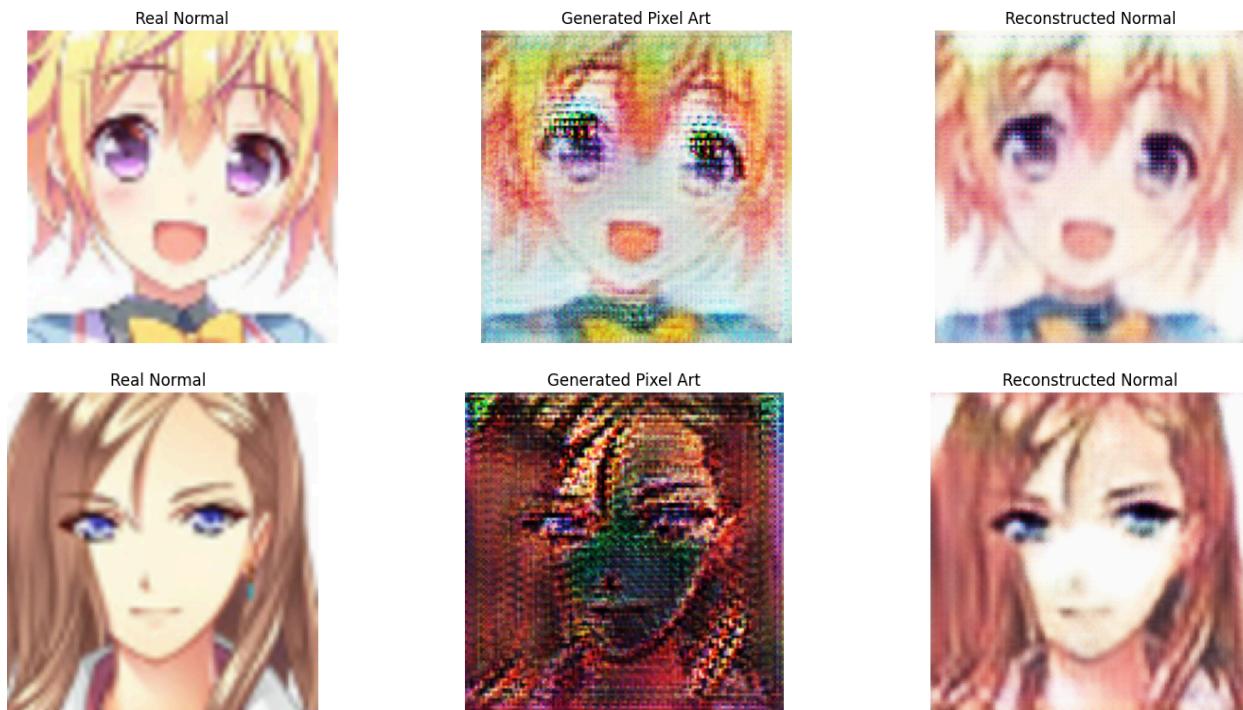


Figura 4. Imágenes generadas por el modelo a lo largo de las épocas. Elaboración propia.

Se tuvo la hipótesis que pudo estar por la diversidad de estilos en las imágenes fuente, pero sobre todo porque el tamaño de un pixel en cada obra, tenía un tamaño diferente, es decir un píxel en una obra podría representar 4 en otra, y el modelo no fue capaz de distinguir la característica. Se probó nuevamente con un dataset donde el tamaño de los píxeles era similar al igual que la paleta de colores, sin embargo el modelo fue capaz de encapsular todos los rostros en la misma imagen de una moneda, y esto ocurría a pesar de diferentes corridas de entrenamiento.

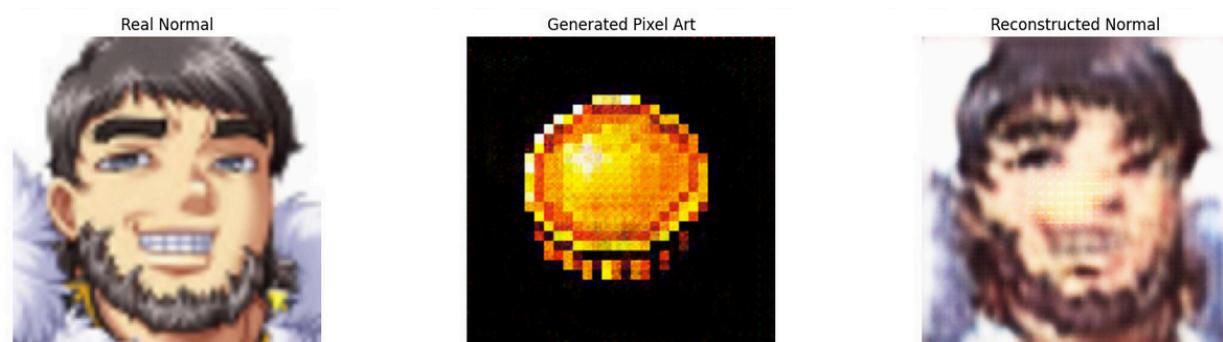
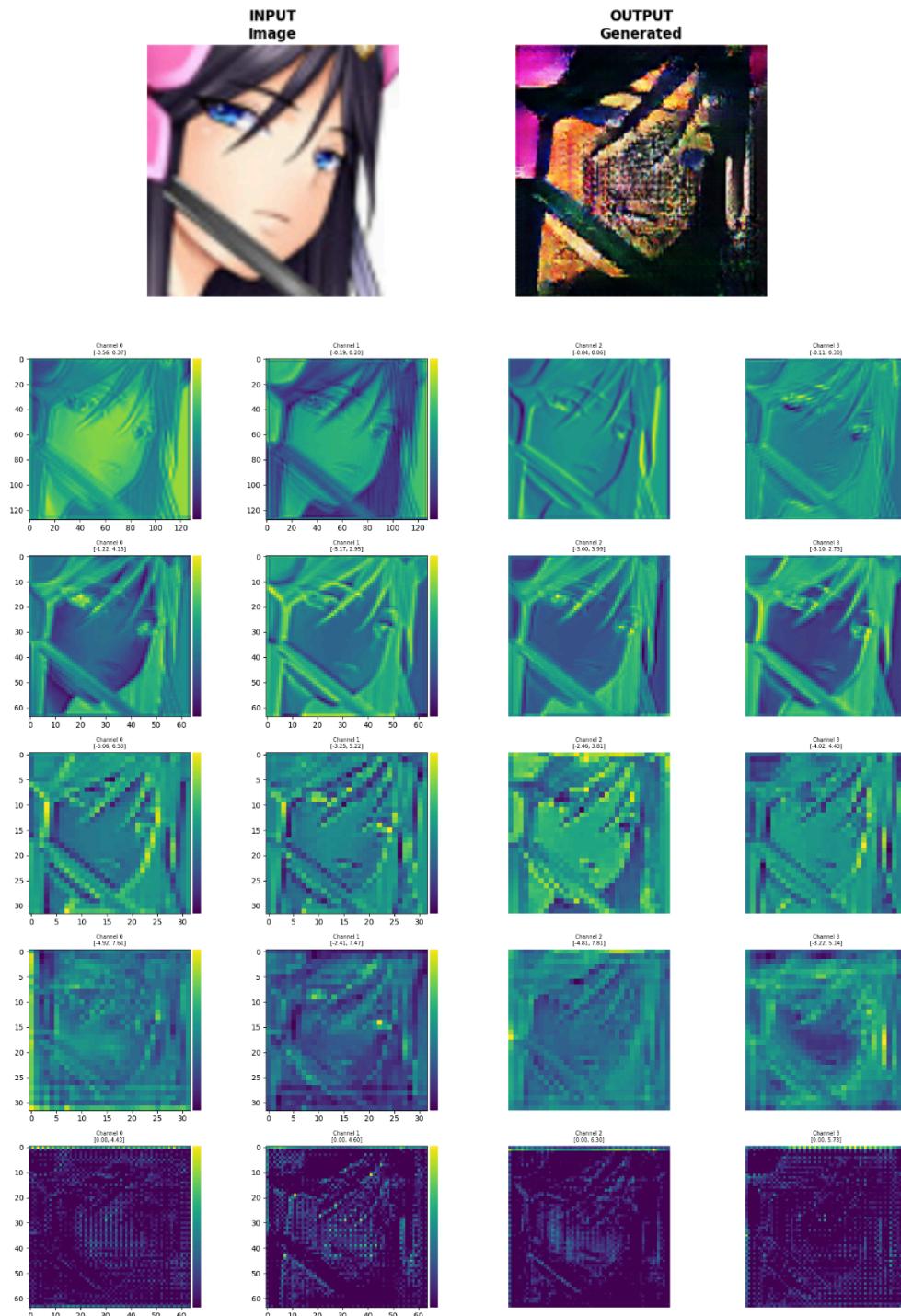


Figura 5. Generación de imágenes con dataset de estilo parecido, y alta presencia de secciones transparentes. Elaboración propia.

Se cree que el posible culpable de estas imágenes incorrectas partía de la sección de upscaling de la red GAN durante la cual se observa el patrón de azulejo distintivo de las imágenes generadas, como se puede ver en los mapas de calor de la siguiente figura.



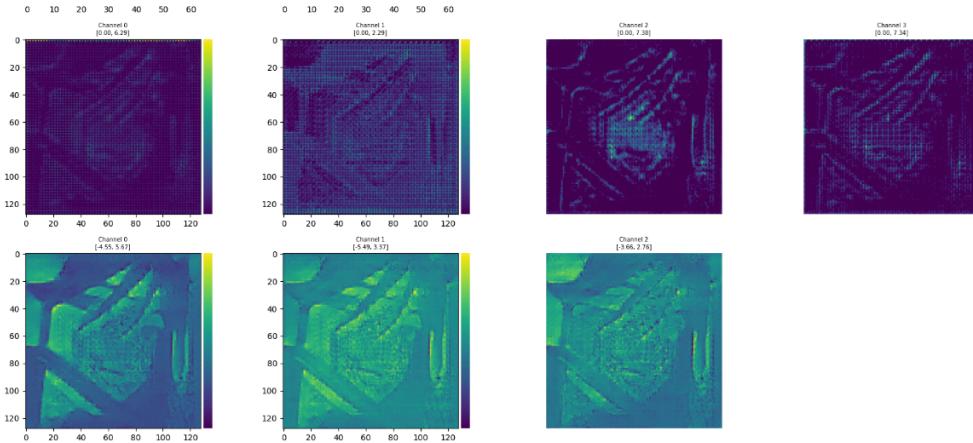


Figura 5. Mapas de activación para las diferentes capas de convolución del generador. Cada fila representa un capa de convolución, y cada columna uno de los canales de dicha capa, colores más amarillos indican mayor valores de activación. Elaboración propia.

En el proceso se descubrió que probablemente las capas de downscaling podrían haber sido suficientes para traducir la imagen a su versión pixelart al mostrar ser capaces de poner atención a secciones específicas de la imagen.

Style-Transfer

Para la red style transfer, se observó que era buena para transcribir la paleta de colores en la imagen resultando pero sin ser capaz de transferir el estilo pixelado que hace tan distintivo al estilo de arte. Sin embargo su tiempo de entrenamiento era considerablemente menor (en el orden de minutos) comparado al CycleGAN (horas de entrenamiento).



Figura 5. Generación de imágenes con dataset de estilo parecido, y alta presencia de secciones transparentes. Elaboración propia.

Conclusión

A pesar que no se obtuvieron resultados satisfactorios, se descubrieron resultados que podrían mejorar futuras iteraciones del modelo.

- Las redes CycleGAN tuvieron mejor rendimiento con datasets con baja presencia de canales de transparencia.
- Una posible mejora, sería remover la sección de Upscaling de la red Generadores de pixelart en la CycleGAN, no solo haciéndolas más rápidas de entrenar sino probablemente más certeras.
- Las redes de Style Transfer mostraron ser buenas para transferir paletas de colores, más no el estilo pyxelart, probablemente se puede mejorar al cambiar las capas de convolución estudiadas.

Bibliografía

Gatys, L. A., Ecker, A. S., & Bethge, M. (2015). A neural algorithm of artistic style. En *arXiv [cs.CV]*. <http://arxiv.org/abs/1508.06576>

Kordic, A. (2025, marzo 24). *What exactly is pixel art and how did it come back to life ?* Artsper Magazine; Artsper. <https://blog.artsper.com/en/a-closer-look/art-movements-en/pixel-art/>

Lan, L., You, L., Zhang, Z., & Fan, Z. (s/f). *Generative Adversarial Networks and Its Applications in Biomedical Informatics*. Researchgate.net. Recuperado el 10 de noviembre de 2025, de https://www.researchgate.net/figure/The-architecture-of-Cycle-GAN_fig2_341320720

Unal, G., & Demir, U. (s/f). *Patch-Based Image Inpainting with Generative Adversarial Networks*. Researchgate.net. Recuperado el 10 de noviembre de 2025, de https://www.researchgate.net/publication/323904616_Patch-Based_Image_Inpainting_with_Generative_Adversarial_Networks

Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired image-to-image translation using cycle-consistent adversarial networks. En *arXiv [cs.CV]*. <http://arxiv.org/abs/1703.10593>