Universidad de O'Higgins

Tarea N° #4:
Informe #4

Sebastián Garrido

COM4402 – Introducción a Inteligencia Artificial
Escuela de Ingeniería, Universidad de O'Higgins
1, Diciembre, 2023

I. Resumen

En este trabajo, se busca aprender acerca de redes generativas, procesamiento, reconstrucción, transferencias y/o generación de imágenes por medio de diferentes métodos de aprendizaje profundo avanzado, en el cual, se escogen 3 de 4 métodos para la realización de este informe.

II. Introduction

En este informe, se hará presente la ejecución, procedimiento, análisis de resultados, estudios, definiciones y conclusiones relacionados al trabajo de aplicación de métodos de aprendizaje profundo avanzado, con enfoque a redes generativas y procesamiento (en términos generales) de imágenes que fue realizado. Su lectura se basará en brindar un marco teórico que defina conceptos claves para esta temática, la metodología que explique lo realizado de forma programática, detallando además las arquitecturas principales de cada código proporcionado, la revisión y exposición de resultados, el análisis del procedimiento y lo que se observó, las conclusiones generales de lo que se extrae a partir de lo realizado, su respectivo resumen, y finalmente la exposición de la bibliografía utilizada para hacer este informe.

III. *Marco teórico*

A. *Redes neuronales: "Una* ***red neuronal*** *es un modelo simplificado que emula el modo en que el cerebro humano procesa la información: Funciona simultaneando un número elevado de unidades de procesamiento interconectadas que parecen versiones abstractas de neuronas", esto también se puede apreciar cuando las redes neuronales son usadas para generar respuestas, cálculos, entre otras actividades que simulen el pensamiento humano mediante un proceso de aprendizaje profundo.*

B. *Overfiting: Un modelo neuronal en el que exista la presencia de Overfitting, será aquel donde se obtiene un error de entrenamiento relativamente bajo, y un error de validación relativamente alto.*

C. *Underfitting: Un modelo neuronal que tenga Underfitting como característica de sus datos, será aquel cuyos "errores tanto de entrenamiento como de validación son similares y relativamente altos"*

D. *Deep Learning: También conocido como aprendizaje profundo, corresponde a una forma de aprendizaje automático, "donde una máquina intenta imitar al cerebro humano utilizando redes neuronales artificiales con más de tres capas que le permiten hacer predicciones con una gran precision"*

E. *Redes neuronales convolucionales: Es un tipo regularizado de red neuronal de retroalimentación que procesa características mediante la optimización de filtros. " se distinguen de otras redes neuronales por su rendimiento superior con entradas de imagen, voz o señales de audio"*

F. *Autoencoder: Es un tipo especial de red neuronal que se entrena para copiar su entrada en su salida, o en otras palabras, que aprenda a generar en la salida el mismo dato de entrada.*

G. *Redes generativas adversarias: Una red generativa adversarias (GAN) es una arquitectura de aprendizaje profundo. Esta entrena dos redes neuronales de modo que compitan entre sí para generar nuevos datos más auténticos a partir de un conjunto de datos de entrenamiento determinado.*

H. *Transferencia neuronal de estilo: Es un algoritmo de Inteligencia Artificial que, modela y proyecta un estilo artístico sobre otra imagen cualquiera.*

I. *Función de activación: Tiene la labor de imponer un límite o modificar el valor resultado para poder proseguir a otra neurona. En otras palabras," es una función que transmite la información generada por la combinación lineal de los pesos y las entradas , es decir son la manera de transmitir la información por las conexiones de salida"*

J. *Normalización: "es la organización de datos de manera coherente para reducir la redundancia y mejorar la integridad de los datos", ayudando así a evitar errores y mejorar la eficiencia.*

K. *Pérdida (loss): La función de pérdida evalúa la desviación entre las predicciones y cálculos realizados por la red neuronal, y los valores reales de las observaciones utilizadas durante el aprendizaje." Cuanto menor es el resultado de esta función, más eficiente es la red neuronal."*

L. *Pooling: Es una operación que generalmente se aplica entre dos capas de deconvolución, y tiene como objetivo reducir el tamaño de las imágenes, y a la vez, preservar sus características más esenciales.*

M. *Hiperparámetro: es un parámetro cuyo valor se utiliza para controlar el proceso de aprendizaje.*

N. *Matriz de confusion: Tiene la función de "valorar cómo de bueno es un modelo de clasificación basado en aprendizaje automático", caracterizándose principalmente en mostrar explícitamente cuando una clase se confunde con otra, permitiendo trabajar de forma separada con diferentes tipos de errores.*

O. *Accuracy: corresponde al porcentaje de clasificaciones correctas que un modelo de aprendizaje entrenado logra, dentro de todas las que ejecuta.*

P. *Epocs (épocas): "Es el número total de iteraciones de todos los datos de entrenamiento en un ciclo para entrenar el modelo de aprendizaje automático".*

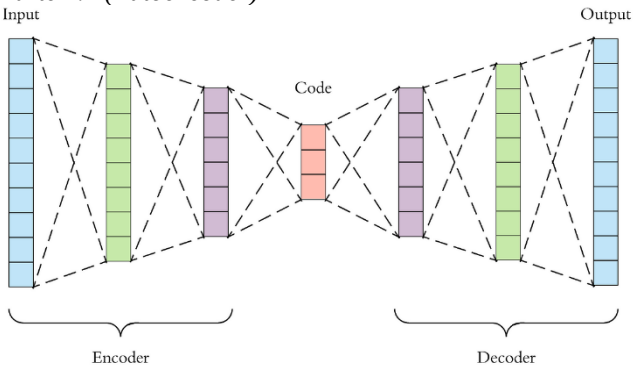
IV. *METODOLOGÍA*

Considerando el Desarrollo de la tarea, cabe detallar en el sentido y explicación del código.

A. *Parte 0*

Se inicia por decidir qué códigos escoger para realizar esta tarea, donde se eligieron las actividades de Autoencoders, GAN, y CNN.

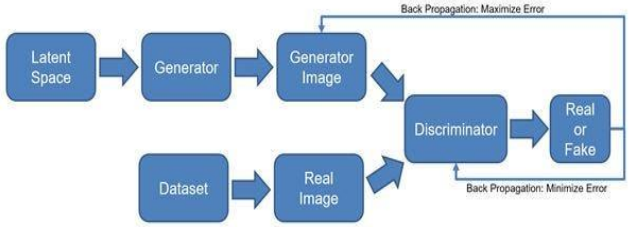
B. *Parte 1.1 (Autoencoder)*



Esto corresponde a la representación gráfica de la arquitectura principal que utiliza el código

La arquitectura principal de un autoencoder consta de dos partes esenciales: el encoder y el decoder. En la imagen, observamos cómo el encoder reduce la dimensionalidad de la entrada original (en este caso, imágenes Fashion MNIST) a una representación latente de menor dimensión. El decoder realiza el proceso inverso, reconstruyendo la imagen original a partir de la representación latente.

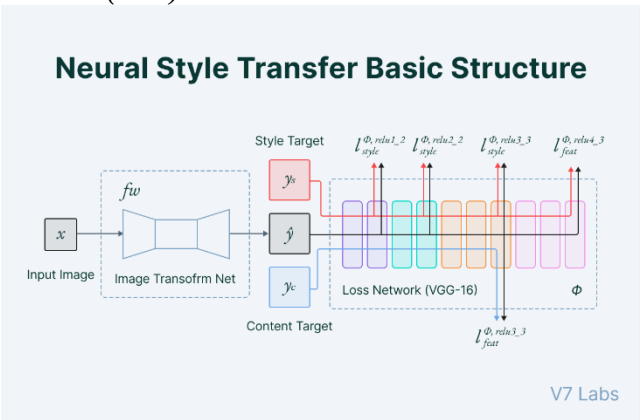
C. *Parte 2.1 (GAN)*



Esto corresponde a la representación gráfica de la arquitectura principal que utiliza el código

Las GAN constan de dos componentes clave: el generador *y el* discriminador. *El* generador *crea imágenes, mientras que el* discriminador *evalúa su autenticidad. La arquitectura utiliza una red adversarial, donde ambos modelos se entrenan simultáneamente. Durante el entrenamiento, el* generador *mejora su capacidad para engañar al* discriminador, *y este último mejora en distinguir entre imágenes reales y generadas.*

D. *Parte 3.1 (CNN)*



Esto corresponde a la representación gráfica de la arquitectura principal que utiliza el código

El modelo de transferencia de estilo artístico utiliza la arquitectura VGG19 preentrenada. VGG19 es una red neuronal convolucional (CNN) que consta de capas de convolución y activación. En el contexto de la transferencia de estilo, se seleccionan capas intermedias para representar el contenido y el estilo de las imágenes. El modelo utiliza estas representaciones para optimizar una imagen de contenido, de modo que combine el contenido de

una imagen de referencia y el estilo de otra. El algoritmo utiliza el descenso de gradiente y una pérdida que incluye términos de estilo, contenido y variación total para lograr la transferencia estilística deseada.

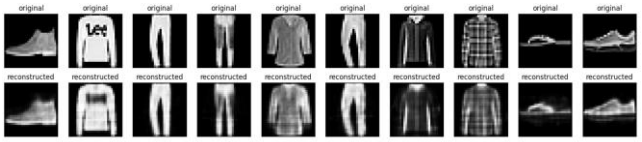
V. *RESULTADOS*

Considerando la metodología anteriormente explicada, ahora consta exhibir los resultados:

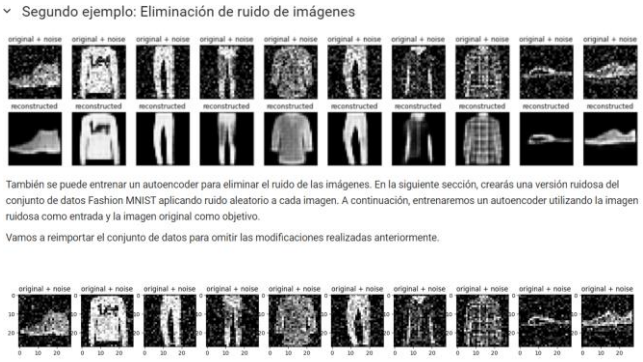
<u>Código 1 (autoencoder):</u>

Ejemplo #1 (Cuadro 1)

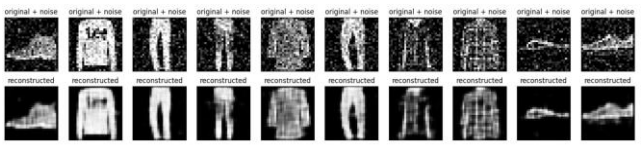
De este primer ejemplo, su resultado es el siguiente: (Cuadro 2)



Segundo ejemplo (Cuadro 3)

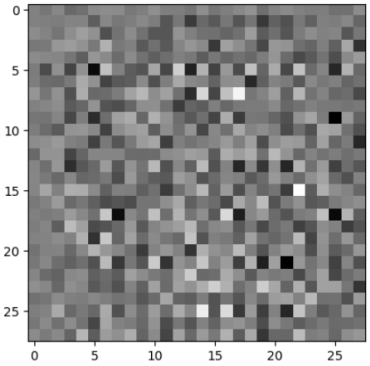


El resultado del segundo ejemplo es el siguiente: (Cuadro 4)

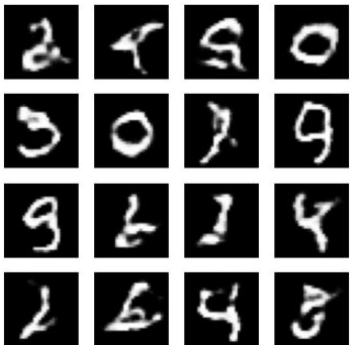


<u>Código 2 (GAN):</u>

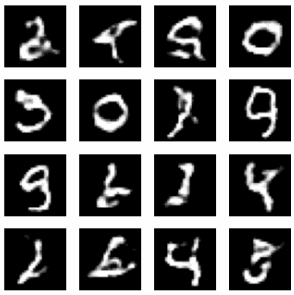
Resultado arrojado por el generador sin entrenar: (Cuadro 5)



Resultado del entrenamiento (Cuadro 6)

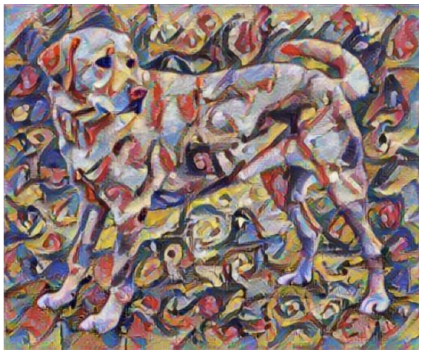


El código mismo solicitó la creación de un gif que documenta el entrenamiento en cuestión, cuyo resultado es el siguiente: (Cuadro 7)



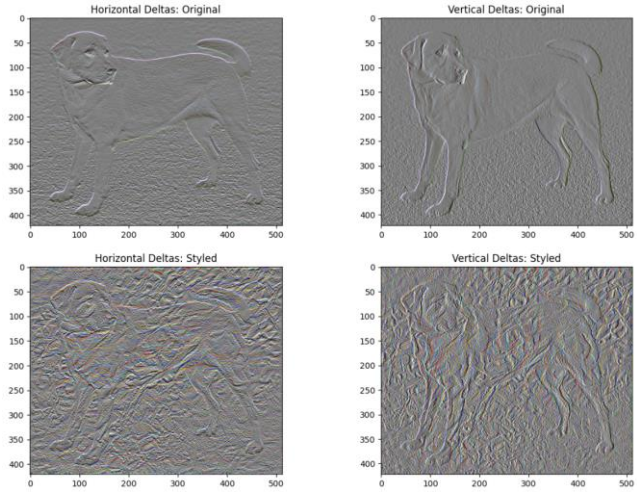
<u>Código 3 (CNN):</u>

El primer resultado viene de la mano del uso de TF-Hub para la transferencia rápida de estilos (Cuadro 8)

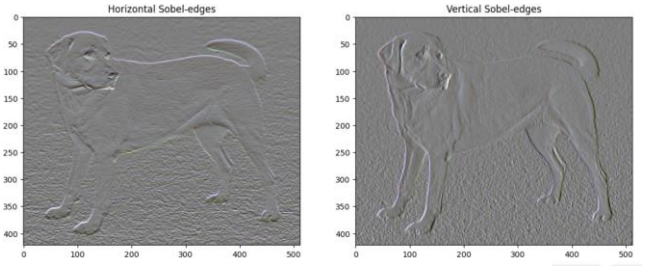


Resultado de construir el modelo VGG19. (Cuadro 9)

Resultados con variable de pérdida total de variación, Original vs estilizada (Cuadro 10)



Resultado con detector de bordes Sobel para obtener un proceso más afinado: (Cuadro 11)



Resultado final de la fusión de imágenes con el uso de la variable de pérdida total de variación (Cuadro 12)

VI. *Discusión*

<u>Código 1 (autoencoder):</u>

En el primer código y algoritmo de Autoencoder, se exponen 2 ejemplos, el primero de estos basándose en el autoencoder básico, que no incluye parámetros de ruido y cuyo procedimiento consta de comprimir y reconstruir imágenes. (ver cuadro 1)

En el primer ejemplo (ver cuadro 2), se puede apreciar un notorio deterioro de la imagen en relación a su calidad original, ya que se ve mucho más borroso y sus características son menos apreciables. No obstante, sigue siendo positivamente distinguible, y salvo por los pantalones con manchas, el resto de imágenes conserva su forma, y parcialmente sus patrones originales

En el segundo ejemplo (ver cuadro 3) se incorpora el uso de ruido aleatorio en el conjunto original de fotografías, y con un autoencoder entrenado y convolucional se buscará volver a la imagen original como objetivo.

En el resultado del segundo ejemplo (ver cuadro 4) se puede apreciar que el uso de ruido y en el resultado arrojado para esta iteración, profundiza aún más en el deterioro de la calidad de imagen al ser borrosa, deforme en ciertos casos, e incluso, menos fiel a la original debido a que el procesamiento de ruido provoca que ciertos elementos se omitan de la imagen original, como sucede en el caso del segundo pantalón, en las camisas y en las sandalias.

Código 2 (GAN):

En el segundo código de redes generativas adversarias, se utilizan 2 modelos a entrenar simultáneamente, el artista (generador de imágenes aparentemente reales) y el crítico (discriminador de imágenes respecto de su veracidad). Dada aquella breve introducción, se inicia exhibiendo el resultado arrojado por el generador sin entrenar (ver cuadro 5).

Se puede observar claramente que el generador no es capaz de hacer una imagen coherente o real siquiera.

Es entonces que luego de generar el discriminador, parámetros diversos, filtros y funciones, se procede a entrenar el mismo modelo, con ruido aleatorio, y durante 50 épocas que buscarán parecerse a los dígitos MNIST, el cual tardó nada más ni nada menos que 10 minutos para otorgar un resultado bastante interesante (ver cuadro 6).

Se puede apreciar con claridad que se logró el objetivo para la mayoría de las imágenes en cuestión, puesto que se pueden definir diversos símbolos, letras y números a partir de estos, siendo en su mayoría casos exitosos.

Cabe mencionar, el código mismo solicitó la creación de un gif que documenta el entrenamiento en cuestión(ver cuadro 7).

Código 3 (CNN):

En el tercer código y para el algoritmo de red neuronal convolucional aplicada para la transferencia neuronal de estilo, se busca (en pocas palabras) tomar 2 imágenes y mezclarlas para que la salida se parezca a la imagen de contenido, con su respectivo patrón o "estilo artístico".

Con esto en mente, y considerando la subida de ambas imágenes a usar para este código, el primer resultado viene de la mano del uso de TF-Hub para la transferencia rápida de estilos (ver cuadro 8).

Se puede apreciar notoriamente que está "hecho a la rápida", ya que hay un patrón o elemento que repercute en toda la imagen sin importar el patrón o la figura que se busca adaptar, además de ver deformidades respecto a los contornos de la figura del perro. Adicionalmente, su procedimiento tarda 22 segundos.

Luego de este caso, se definen representaciones de contenido y estilo. Y con esto hecho, se procede a construir el modelo VGG19.

Su refinamiento y entrenamiento le permite optimizar detalladamente a partir de la cantidad de tiempo que el proceso tarde en llevarse a cabo, por ejemplo, he aquí una optimización rápida de 7 segundos, comparada con una de 1 minuto (71 segundos), (ver cuadro 9).:

Como se puede apreciar, a más tiempo, más detalles e inclinación al patrón artístico proliferan en el resultado de la optimización, dando a lugar imágenes artísticas como las apreciadas en la parte superior.

Finalmente, se introduce un parámetro de "pérdida total de variación" para así reducir artefactos de alta frecuencia en la imagen de salida, y posteriormente generar una optimización más suavizada y con menos ruido.

Para ambos ejes en la imagen original y estilizada, se aprecia el siguiente análisis: (ver cuadro 10).

Y en el siguiente caso, se implementa un detector de bordes Sobel para obtener un resultado más afinado (ver cuadro 11).

Finalmente, la aplicación de todo este procedimiento de pérdida total de variación arroja una imagen bastante más prolija, notoria, definida y fiel a la imagen a la que se le quiere implementar el patrón artístico, en este caso, el perro (ver cuadro 12), siendo significativamente más detallada su visualización tanto en su forma, como en el patrón implementado. (Tiempo est. 75.7 segundos).

VII. *Conclusiones Generales*

Conclusiones Generales: Ofrece un resumen de los hallazgos, su relevancia y posibles aplicaciones, colocándolos en el contexto más amplio de la investigación

Tomando en cuenta lo exhibido en los puntos anteriores, se puede concluir que el modelar, probar, ejecutar y entrenar modelos puede llegar a dar resultados muy sorprendentes dependiendo del objetivo a buscar, y lo que los códigos elegidos han de realizar, con esto se quiere empezar por determinar que no hay un mejor código que otro, ya que son funcionalidades distintas.

Dicho esto, respecto a lo que se ha aprendido del primer código, es que la aplicación de ruido a largo plazo podría ayudar a robustecer un modelo en el procesado de imágenes de alta fidelidad y complejidad, siempre que esté bien entrenado. De lo contrario, un autoencoder convolucional y entrenado será la mejor alternativa para el procesado de imágenes.

Sobre el segundo tópico trabajado, se ha acatado que es absolutamente necesario permitir a la máquina trabajar en base a un entrenamiento extenso, de lo contrario, no logrará el objetivo de "crear nuevas figuras realistas" a partir de lo que se le entregue, esto debido a que una generación sin entrenamiento no brindará más que una imagen pixelada y sin sentido.

Finalmente, en el caso del tercer código, se puede concluir que para “hacer arte” y combinar dos fotografías con tal de adaptar un patrón artístico en algo convencional, se debe garantizar la creación de un modelo específico, y su procesamiento, con posterior entrenamiento, a partir de variables de pérdida total de datos. De esa forma, se completará lo solicitado de la forma más fidedigna a las solicitudes del usuario.

VIII. *Resumen*

En resumen, se puede observar en el trabajo realizado una interesante, variada y en cierto aspecto compleja ejecución del entrenamiento de modelos, para cualquiera de las situaciones concebidas en torno a la complejidad de la precisión buscada, yendo de menos a más a medida que se iban añadiendo más variables o elementos en la ejecución. Con esto dicho, se observa una ligera demora mayor en aquellos que tengan que lidiar con temas de ruido, entremanientos generativos entrenados, y construcción de modelos en conjunto de variables de pérdidas de datos, permitiendo mejores resultados a expensas de una mayor demora y potencial costes de rendimiento. Finalmente, sin mayores complicaciones se puede determinar que el ejercicio dado en el primer código de Autoencoder, no es necesario que se cedan complejidades para obtener un resultado decentemente esperado, a diferencia del necesario entrenamiento para el código de redes generativas adversarias, y el tercer código de aplicación de transferencias neuronales de estilo. Por otro lado, esto permitió que la máquina pudiera “hacer arte” de la mejor forma, tanto permitiendo que esta fuera la creadora, como también en el acto de combinar imágenes.

IX. *Bibliografía*

1. ***[1]*** *IBM. "Redes neuronales (SPSS Modeler)". [En línea]. Disponible en: https://www.ibm.com/docs/es/spss-modeler/saas?topic=networks-neural-model. Accedido el 23 de octubre de 2023.*
2. ***[2]*** *CodificandoBits. "Underfitting y Overfitting: Conceptos Esenciales". [En línea]. Disponible en: https://www.codificandobits.com/blog/underfitting-y-overfitting/#:~:text=Un%20modelo%20con%20underfitting%20es,uno%20de%20validaci%C3%B3n%20relativamente%20alto.. Accedido el 23 de octubre de 2023.*
3. ***[3]*** *Datademia. "¿Qué es el Deep Learning y qué es una Red Neuronal?". [En línea]. Disponible en: https://datademia.es/blog/que-es-deep-learning-y-que-es-una-red-neuronal. Accedido el 23 de octubre de 2023.*
4. ***[4]*** *DataScientest. "Perceptrón: qué es y para qué sirve". [En línea]. Disponible en: https://datascientest.com/es/perceptron-que-es-y-para-que-sirve. Accedido el 23 de octubre de 2023.*
5. ***[5]*** *Telefonica Tech. "Función de Activación en Redes Neuronales". [En línea]. Disponible en: https://aiofthings.telefonicatech.com/recursos/datapedia/funcion-activacion#:~:text=Una%20funci%C3%B3n%20de%20activaci%C3%B3n%20es,por%20las%20conexiones%20de%20salida.. Accedido el 23 de octubre de 2023.*
6. ***[6]*** *Universidad Internacional de la Rioja. "Backpropagation". [En línea]. Disponible en: https://www.unir.net/ingenieria/revista/backpropagation/. Accedido el 23 de octubre de 2023.*
7. ***[7]*** *Telefonica Tech. "Cómo interpretar la matriz de confusión: Ejemplo Práctico". [En línea]. Disponible en: https://telefonicatech.com/blog/como-interpretar-la-matriz-de-confusion-ejemplo-practico. Accedido el 23 de octubre de 2023.*
8. ***[8]*** *Ediciones ENI. "Inteligencia Artificial Fácil: Machine Learning y Deep Learning Prácticos". [En línea]. Disponible en: https://www.ediciones-eni.com/libro/inteligencia-artificial-facil-machine-learning-y-deep-learning-practicos-9782409025327/la-prediccion-con-neuronas. Accedido el 23 de octubre de 2023.*
9. *[9] CodificandoBits, "Autoencoders: Explicación y Tutorial en Python," CodificandoBits, [En línea]. Disponible en: https://www.codificandobits.com/blog/autoencoders-explicacion-y-tutorial-python/#qu%C3%A9-es-un-autoencoder. Accedido el 11 de noviembre de 2023.*
10. *[10] Amazon Web Services, "Generative Adversarial Network (GAN) - Qué es y cómo funciona," AWS, [En línea]. Disponible en: https://aws.amazon.com/es/what-is/gan/#:~:text=Una%20red%20generativa%20antag%C3%B3nica%20(GAN,de%20datos%20de%20entrenamiento%20determinado. Accedido el 11 de noviembre de 2023.*
11. *[11] La Máquina Oráculo, "Transferencia Neuronal de Estilo," La Máquina Oráculo, [En línea]. Disponible en: https://lamaquinaoraculo.com/deep-learning/transferencia-neuronal-de-de-estilo/. Accedido el 11 de noviembre de 2023.*

12. ***[12]*** *Iguazio. "Model Accuracy in Machine Learning". [En línea]. Disponible en: https://www.iguazio.com/glossary/model-accuracy-in-ml/#:~:text=AI%20accuracy%20is%20the%20percentage,is%20often%20abbreviated%20as%20ACC.. Accedido el 23 de octubre de 2023.*
13. ***[13]*** *Huawei Enterprise. "¿Qué es Epoch en Machine Learning?". [En línea]. Disponible en: https://forum.huawei.com/enterprise/es/%C2%BFQu%C3%A9-es-Epoch-en-Machine-Learning/thread/667232453749784577-667212895009779712. Accedido el 23 de octubre de 2023.*

14. [14] IBM, "Convolutional Neural Networks," IBM, [En línea]. Disponible: https://www.ibm.com/es-es/topics/convolutional-neural-networks. [Accedido: 11 Nov 2023].

15. [15] J. I. Blanco, "Por qué la normalización es clave e importante en machine learning y ciencia de datos," Medium, [En línea]. Disponible: https://jorgeiblanco.medium.com/por-qu%C3%A9-la-normalizaci%C3%B3n-es-clave-e-importante-en-machine-learning-y-ciencia-de-datos-4595f15d5be0#:~:text=La%20normalizaci%C3%B3n%20se%20refiere%20a,errores%20y%20mejorar%20la%20eficiencia. [Accedido: 11 Nov 2023].

16. [16] Amazon Web Services, "Hyperparameter Tuning," AWS, [En línea]. Disponible: https://aws.amazon.com/es/what-is/hyperparameter-tuning/. [Accedido: 11 Nov 2023].

17. [17] DataScientest, "Convolutional Neural Network (CNN)," DataScientest, [En línea]. Disponible: https://datascientest.com/es/convolutional-neural-network-es#:~:text=Capa%20de%20Pooling%20(POOL)%3A,preservar%20sus%20caracter%C3%ADsticas%20m%C3%A1s%20esenciales. [Accedido: 11 Nov 2023].