UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
SANTIAGO- CHILE

"TRANSFERENCIA DE ESTILO BIMODAL DE TEXTO A IMÁGENES USANDO MODELOS GENERATIVOS PROFUNDOS"

DIEGO IGNACIO GUTIÉRREZ SILVA

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: Marcelo Mendoza
Profesor Correferente: Pedro Godoy

Agosto – 2022

DEDICATORIA

A mi familia, amigos y abuela que se encuentra en el cielo. Estoy seguro de que estará feliz viéndome cumplir uno de los objetivos más importantes de mi vida.

AGRADECIMIENTOS

- A mi familia, especialmente a mis padres, los cuales me apoyaron en cada una de mis decisiones y gracias a ellos soy quien soy hoy en día. Este trabajo en gran parte es por ellos.
- A mis amigos y especialmente a María Jose, los cuales siempre estuvieron presente apoyándome para lograr mis objetivos a lo largo de estos años. Sin ellos probablemente nunca hubiese podido llegar hasta acá.
- A todos los profesores que tuve en la carrera, y en especial, a Pedro Godoy, Jose Luis Martí, Elizabeth Montero y otros profesores que me brindaron todos sus conocimientos a lo largo de la carrera siendo un excelente profesor(a) y gran persona, dispuestos a ayudar en lo que necesité.
- A mi profesor guía Marcelo Mendoza, por su confianza, ayuda y enseñanzas en este proceso largo proceso. Gracias por acompañarme en este desarrollo y mostrarme un área de trabajo super linda e importante en la informática.

RESUMEN

El campo de las redes neuronales se ha convertido en una de las áreas más importantes de la inteligencia artificial por su capacidad de solucionar problemas comunes con una gran precisión. Esto ha llevado a abordar nuevas arquitecturas y modelos para problemas más complejos como es el caso de la transferencia de estilo neuronal. En este problema, se busca generar una imagen mezclando el estilo de una y el contenido de otra. En esta memoria se propone una nueva investigación y metodología para realizar transferencia de estilo bimodal utilizando como entrada un texto. La metodología consiste en tres sub-modelos donde inicialmente se recupera una imagen de contenido usando una representación multimodal de imágenes y texto en un mismo espacio latente a través de una proyección de sus representaciones. Luego, se extrae la imagen de estilo a través de un modelo de *Image Retrieval*, para finalizar con un modelo generativo que permite generar imágenes artísticas combinando el estilo y contenido de ambas imágenes a través de una optimización de sus funciones de *loss*. De esta forma, se logran recuperar imágenes semánticamente similares a las descripciones, logrando buenas medidas de precisión (*Median rate)* en la recuperación de imágenes del *dataset SemArt*. También, se logran obtener imágenes de buena calidad en el modelo de transferencia de estilo neuronal, mezclando correctamente el estilo de una imagen con el contenido de otra dependiendo de los pesos utilizados. Por último, se plantean los trabajos futuros a realizar en el modelo y la documentación para poder replicar el sistema.

Palabras Clave— Modelos profundos; Transferencia; Arquitectura; Estilo; Contenido.

The field of neural networks has become one of the most important areas in artificial intelligence due to their great capacity of solving common problems with great precision. This had led to the propposal of novel arquitectures and models in order to tackle more complex problems as neural style transfer. In this problem, the goal is to generate an image, mixing the style from one of them with the content from the other. In this article we propose a novel research and metodology to achieve bimodal style transfer using text as input. The metodology consists in three sub models where we initially retrieve one content image and a text, which are then mapped into a multimodal common latent space through the projection of their attributes. Then, an image is extracted through an image retrieval model, to conclude with a generative model which allows to create artistic images, by the combination of content and style from both images by the optimization of their loss functions. Thus, this work retrieve semantically similar images with respect to the query description, achieving great precision rates (Median rate) in image retrieval applied to the SemArt dataset. Additionally, the transfer style neural model preserves the image's high quality, combining style and content in a correct manner dependings on the weights used. Finally, we discuss future work with respect to the model and the system is documented in order to replicate the experimentation.

Palabras Clave— Deep models; Transfer; Architecture; Style; Content.

GLOSARIO

UTFSM: Universidad Técnica Federico Santa María.

ANN: Redes neuronales artificiales

GAN: Red generativa adversaria

CNN: Red neuronal convolucional.

VGG: *Visual Geometry Group*

Resnet*: Residual Net*

CML: *Cosine Margin Loss*

BOW: *Bag of Words*

ConvNet: Redes convolucionales

BGR: Escala de colores (*Blue*, *Green*, *Red)*

INDICE DE CONTENIDOS

INDICE DE FIGURAS

INDICE DE TABLAS

INTRODUCCIÓN

El campo de las redes neuronales se ha convertido en una de las áreas más importantes de la inteligencia artificial por su capacidad de solucionar problemas con una gran precisión, resultando en una herramienta utilizada para tareas comunes como lo es la clasificación de imágenes, detección de objetos, diagnósticos de salud. Este campo de investigación ha ido en crecimiento debido al aumento exponencial de la cantidad de información que se produce diariamente y a las capacidades computacionales que han ido creciendo.

En las artes, específicamente en las pinturas, los humanos han dominado la habilidad para poder crear experiencias visuales a través de la composición de una interacción compleja entre el contenido y estilo otorgado a una imagen. Hasta la actualidad, la computación no ha logrado superar las capacidades de las personas para poder generar arte de igual o mayor calidad que la de un pintor. Sin embargo, en otras áreas claves de la percepción visual, como el reconocimiento de objetos y rostros humanos, se han demostrado modelos que permiten resolver estas tareas. Estos modelos son conocidos como redes neuronales profundas.

Los humanos al trabajar en las áreas artísticas buscan representar en las pinturas sus pensamientos, visiones, sentimientos o ideas. Estos pensamientos muchas veces pueden ser difíciles de ser representados, ocasionando frustración y horas de trabajo por parte de la persona.

Normalmente al trabajar en el área de generación, uno define previamente cómo es el contenido que quiere representar en la imagen, consiguiendo tardar desde horas hasta días en el proceso de ideación de como representar sus ideas en un dibujo.

En esta memoria se explora un modelo artificial basado en una red neuronal profunda, creada para poder generar imágenes artísticas de alta calidad perceptiva a partir de la idea escrita de una persona.

El documento se divide en 5 capítulos. En primer lugar, el capítulo 1 define el problema, la solución y los objetivos de la investigación. El capítulo 2 detalla los conceptos necesarios para la creación del modelo, pasando desde la definición básica de las redes neuronales hasta los modelos más profundos. El tercer capítulo define la propuesta de solución, la cual se divide en tres subproblemas a resolver. En el cuarto capítulo se implementa la solución propuesta y se muestran los resultados obtenidos en la investigación. El último capítulo contiene las conclusiones obtenidas a partir del desarrollo y resultados logrados.

CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA

1.1 Situación actual

Dado el crecimiento que ha existido en los últimos años respecto a la cantidad de datos que se manejan y producen, como también la capacidad que poseen las computadoras para poder trabajar con ellos, el área de las redes neuronales (ANN) ha logrado solucionar cada vez una mayor cantidad de problemas con gran precisión como por ejemplo clasificación de imágenes [Yin19], reconocimiento de voz [Gupta18], detección de objetos o transferencia de estilo. El éxito en su precisión se debe a la gran capacidad de aprender automáticamente una tarea a partir de ejemplos que se le pasa a la máquina. Por esto, uno de los factores más importantes para que una red neuronal tenga éxito es contar con un volumen de datos adecuado para el aprendizaje de la máquina. Si el conjunto de entrenamiento no es representativo, equilibrado ni lo suficientemente grande, por más que se cuente con la mejor arquitectura de una ANN, no logrará aprender una tarea con una buena precisión.

El campo de la transferencia de estilo es relativamente nuevo en el área de las redes neuronales generativas siendo su principal uso para la generación de nuevas imágenes a partir de dos imágenes de entrada ingresadas por parte del usuario. A medida de los años, la investigación en estos campos ha ido en aumento, logrando ser uno de los desafíos más novedosos dentro de las redes neuronales.

1.2 Información del problema

Actualmente existe una limitada capacidad para generar datos artísticos (pinturas) a partir de humanos. Esto supone un problema a la hora de querer generar nuevas imágenes en gran escala. Normalmente al trabajar en el área de generación artística, uno define previamente cual es el contenido que quiere representar en la imagen, pudiendo tardar desde horas hasta días en el proceso de ideación de como representar sus ideas en un dibujo.

Las informaciones a las que accedemos a diario son realizadas en diferentes estilos combinados. Las imágenes comúnmente vienen asociados a descripciones, tags, audios u otros tipos de contenidos. La tendencia más destacada es fusionar información de diferentes modalidades para formar una mejor representación sobre lo que vemos. Por ejemplo, los libros de literatura regularmente llevan incluido imágenes artísticas en su portada o dentro del texto permitiendo una mejor representación de las emociones que el autor quiere dar a sentir en su contenido. La información multimodal ha tenido diversas investigaciones de sus aplicaciones en campos como clasificación de imágenes o recuperación de información, pero no se ha abordado profundamente su aplicación en la transferencia de estilo bimodal.

1.3 Solución

Para solucionar este problema se implementará un modelo generativo profundo utilizando redes neuronales que permita a una máquina aprender la tarea de construir una pintura a partir de un texto ingresado por el humano, permitiendo que se generen ejemplos artificiales para su posterior uso personal en la tarea que se requiera.

Para llevar esto a cabo, la máquina debe aprender diferentes contenidos y estilos de pinturas que los artistas reflejan en sus imágenes artísticas junto a la descripción que se puede entregar sobre la misma. Para esto se trabajará con un conjunto de datos bimodal donde se tiene la imagen, el título de la pintura y un pequeño comentario que describa lo observado en el contenido de la obra.

La red que se implementará se basa principalmente en tres módulos. El primero consiste en que la red aprenda representaciones conjuntas de texto e imágenes en un mismo espacio latente para poder recuperar una imagen de contenido a partir de un *input* de texto. El siguiente módulo es donde se recupera la imagen de estilo a partir de un sistema de recuperación *Image2Image*. El módulo final es donde ocurre la transferencia de estilo. Aquí, usando una imagen de estilo y contenido extraídas de los módulos anteriores, se realiza la transferencia de estilo neuronal a partir de un método de optimización de una función de *loss* que combina tanto el estilo como el contenido de una imagen.

Poder solucionar este problema permitirá que se resuelvan los problemas de escalabilidad que existen actualmente para trabajar en la generación de grandes volúmenes de datos artísticos (específicamente imágenes), además de ayudar con el proceso creativo a autores u otras personas que quieren representar de manera visual sus ideas. Finalmente, esta investigación podrá lograr abrir un nuevo campo de investigación en áreas de inteligencia artificial sobre la transferencia de estilo bimodal de texto a imágenes.

1.4 Objetivos

1.4.1 Objetivo general

- Implementar un modelo generativo profundo que permita generar una imagen artística nueva a partir de un texto utilizando transferencia de estilo.

1.4.2 Objetivos específicos

- Analizar los modelos generativos y sus respectivas representaciones que incluyan volúmenes de datos bimodales (texto e imágenes).
- Construir un modelo que permita representar texto e imágenes en un único espacio latente.

- Observar la calidad de las imágenes generadas por el modelo generativo implementado.
- Documentar los resultados para investigaciones futuras en el área de inteligencia artificial.

CAPÍTULO 2: MARCO CONCEPTUAL

2.1 Redes neuronales artificiales

2.1.1 Definición

Una red neuronal artificial (*ANN*) es un paradigma de aprendizaje y procesamiento automático inspirado en el funcionamiento del sistema nervioso humano. Las *ANN* se esfuerzan en reconocer las relaciones subyacentes en un conjunto de datos a través de un proceso que imita la forma en que funciona el cerebro humano [James20]. En este sentido, las redes neuronales se refieren a sistemas neuronales, ya sea de naturaleza orgánica o artificial. Respecto a su composición, están compuestas por múltiples capas que se conectan a través de la conexión entre las neuronas (nodos) de distintas capas. Cada neurona toma como entrada las salidas de las neuronas de capas antecesoras, cada una de esas entradas se multiplica por un peso y mediante una función de activación se calcula la salida. La unión de todas las neuronas interconectadas forma una red neuronal artificial.

2.1.2 Capas

Toda red neuronal se divide en tres capas claves:

- Capa de entrada (*input layer*): Se encarga de recibir y transformar la información que entra a la red (datos de entrada), para luego transmitir esta información a la primera de las capas ocultas.
- Capas ocultas (*hidden layer*): Son capas invisibles para sistemas externos. Permiten a la red generar una representación propia de los datos recibidos. Aquí se realizan todos los cómputos necesarios para generar un resultado y ser enviado a la capa de salida.
- Capa de salida (*output layer*): Es la capa final de una red y se encarga de recoger la representación creada por la red para luego transformarla en una salida legible para el usuario.

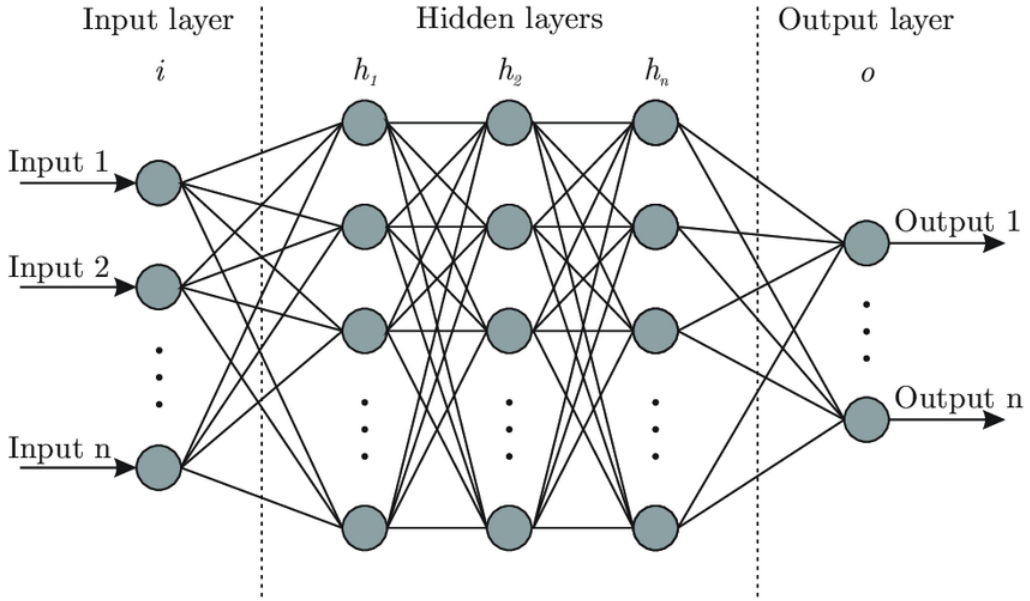


Figura 1: Modelo de una red neuronal con múltiples capas. Fuente: https://wandb.ai/site/articles/fundamentals-of-neural-networks

2.2 Elementos de una red neuronal

2.2.1 Neurona

La neurona artificial es el componente básico de las redes neuronales. Su función es modelar el funcionamiento de una neurona biológica a través de una función matemática. La salida de una neurona es enviada a las otras neuronas que se encuentren conectadas. Uno de los primeros modelos de una neurona artificial fue definido por McCulloch: [McCulloch43]

$$N = \sigma(\sum w_{\mathrm{i}}\, x_i + b)$$

Donde cada símbolo corresponde a:

σ: Función de activación.

W: Vector de pesos. Se encarga de regular la importancia de cada entrada.

X: Vector de datos de entrada.

B: Valor del bias o sesgo (threshold).

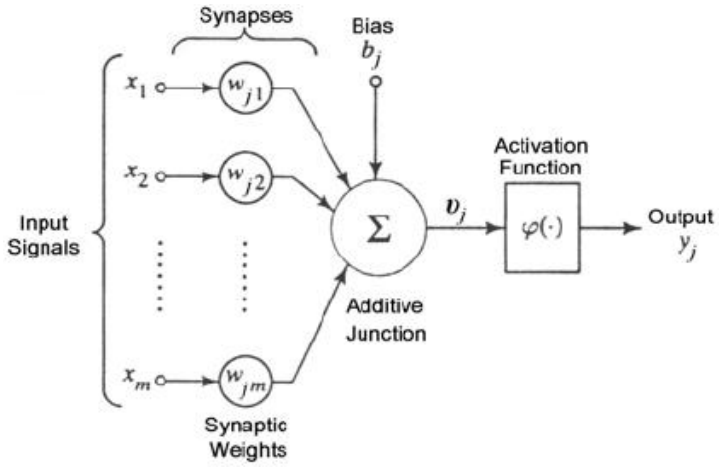


Figura 2: Esquema de una neurona artificial. Fuente: https://www.researchgate.net/figure/Artificial-Neuron-models-and-its-parts-Source-Adapted-from-Haykin-1994_fig2_229036664

2.2.2 Función de activación

La función de activación o de transmisión es usada para simular una respuesta de una neurona biológica y obtener un output "y" a partir de una señal de entrada. Las funciones de activación no lineales han aportado a las redes neuronales la capacidad de diferenciar la información que no puede ser clasificada linealmente en el espacio de datos [Ding18]. Las funciones de activación no lineales más comunes se analizan a continuación:

- **Sigmoidal**

Es una de las funciones más utilizadas y está definida por la siguiente función:

$$f(x) = \frac{1}{1 + e^{-x}}$$

Donde el dominio de x está en todos los reales mientras que la salida esta acotada entre valores de cero y uno.

- ***ReLU***

ReLU se caracteriza por asignar un valor de cero para cualquier valor de entrada negativo mientras que para valores positivos se comporta de manera lineal. Esta función es más rápida para ser computada dado que no se necesita realizar funciones exponenciales. La función de activación *ReLU* se define de la siguiente manera:

$$f(x) = \max(0, x)$$

- **Tangente hiperbólica**

La función matemática de la tangente hiperbólica es similar a la sigmoidal, pero puede lograr tiempos de convergencia más rápidos y con un menor error de clasificación. Por otro lado, el cálculo de las derivadas de la función hiperbólica es más complicada que en la sigmoidal. La ecuación de esta función está dada por:

$$f(x) = \frac{(\mathrm{e}^{\mathrm{x}} - e^{-x})}{(\mathrm{e}^{\mathrm{x}} + e^{-x})}$$

2.2.3 Entrenamiento o Aprendizaje de una red

El proceso para llevar a cabo el aprendizaje de una red neuronal se denomina entrenamiento, el cual consiste en ajustar los pesos de cada una de las neuronas para resolver el problema para la cual se creó la red. El método más usado es *Backpropagation***.** Este algoritmo ajusta los pesos de las neuronas en cada iteración a través de un algoritmo que calcula el gradiente de una función de *loss* con respecto a las variables del modelo. Por otra parte, *Stochastic Gradient Descent* (*SGD*), es un algoritmo de optimización que busca minimizar la función de *loss* de un modelo predictivo en un conjunto de entrenamiento. Para realizar esta tarea el algoritmo busca el conjunto de variables de entrada para hallar el valor mínimo de una función objetivo, al cual se lo denomina mínimo de la función.

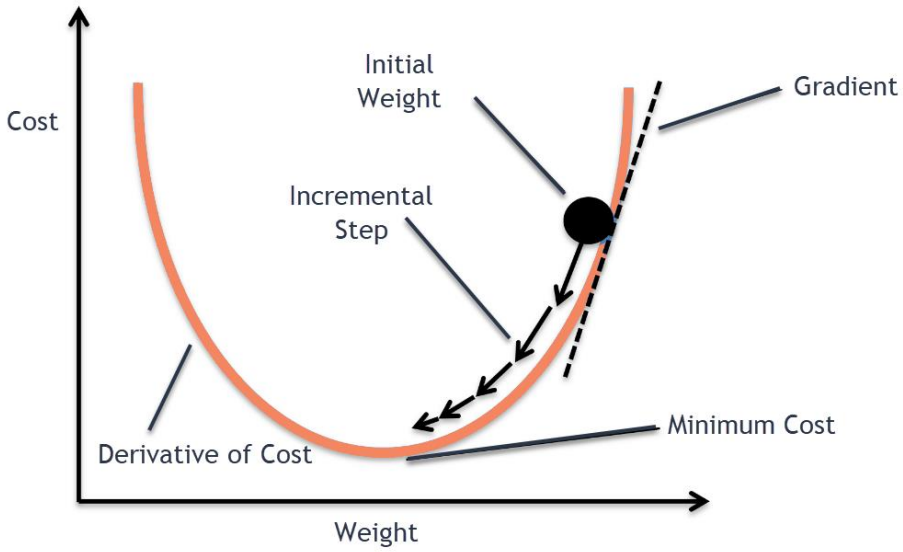


Figura 3: Gradiente Descendente. Fuente: https://medium.com/@divakar_239/stochastic-vs-batch-gradient-descent-8820568eada1

2.3 Redes convolucionales

Las redes neuronales convolucionales (*ConvNet* / *CNN*) son redes que se especializan en problemas de computación visual, teniendo un buen desempeño en tareas de clasificación de imágenes, reconocimiento de patrones en vídeos, detección de objetos, segmentación semántica, entre otras tareas. Estas redes son análogas a las redes neuronales tradicionales en su composición, ya que también están compuestas por neuronas, capas y pesos que se auto optimizan a través de entrenamiento [O'shea15].

La mayor diferencia entre las redes convolucionales y las redes neuronales tradicionales es que las primeras se utilizan en el campo de reconocimiento de patrones dentro de las imágenes. Esto permite utilizar características específicas de las imágenes en la arquitectura, como por ejemplo la multidimensionalidad. En el caso de una imagen, está utiliza tres dimensiones, las cuales son el alto, el ancho y los canales de colores. Esto quiere decir que la red toma como entrada los píxeles de una imagen. Si hay una imagen a color con 28 píxeles de alto y ancho respectivamente, necesitaríamos 3 canales (RGB) y se usarían 28x28x3 = 2352 neuronas de entrada.

2.3.1 Arquitectura de una red neuronal convolucional

Las *CNN* están compuestas de tres tipos de capas. Estas son capas convolucionales, capas de pooling y capas *fully-connected***.** Una vez que estas capas se apilan se forma la arquitectura de una red neuronal convolucional [O'shea15].

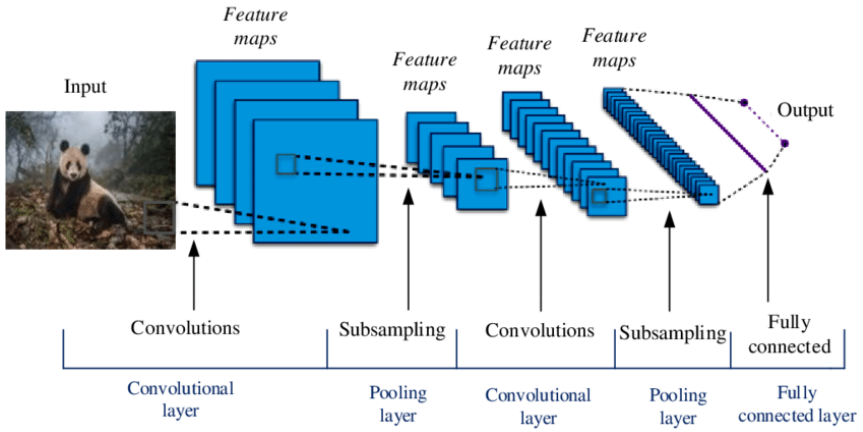


Figura 4: Ejemplo de la arquitectura de una red convolucional. Fuente: https://medium.com/@lchandratejareddy/convolutional-neural-networks-6ad55d9bf446

La funcionalidad básica de una *CNN* se puede dividir en cuatro áreas claves:

1. Al igual que en otros tipos de redes, la capa de entrada contendrá los valores de los píxeles de la imagen.

2. La capa convolucional determinará la salida de las neuronas que están conectadas a regiones locales de la entrada a través del cálculo del producto escalar entre sus pesos y la región conectada al volumen de entrada.
3. La capa de *pooling* se encargará de realizar un *downsampling* de la dimensionalidad de una entrada, reduciendo el número de parámetros y preservando los más importantes.
4. Las capas *fully-connected* realizarán las mismas funciones que se encuentran en las ANN e intentarán realizar una clasificación a través de los puntajes de las activaciones de las neuronas.

- **Capas convolucionales**

Las capas convolucionales juegan un papel clave en el funcionamiento de las *CNN*. Convolución es una operación matemática para unir dos conjuntos de información. En el caso de las capas convolucionales la convolución se aplica a unos datos de entrada (*input data)* usando *un* filtro *(kernel),* para producir un mapeo de características (*feature map)* [Chandra20].

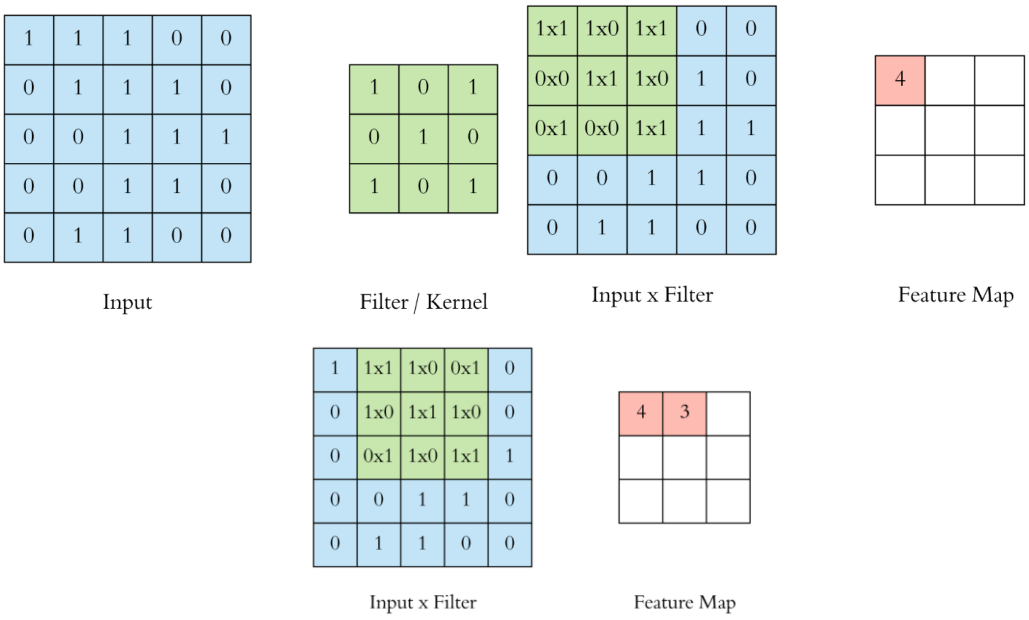


Figura 5: Ejemplo de operación convolucional sobre unos datos de entrada. Fuente: https://medium.com/@lchandratejareddy/convolutional-neural-networks-6ad55d9bf446

Las capas convolucionales también pueden reducir la complejidad del modelo a través de la optimización del *output*. Estos se optimizan mediante tres parámetros: *depth, stride* y *zero-padding.*

- **Capas de *pooling***

Las capas de *pooling* tienen el objetivo de reducir la dimensionalidad de la representación y, por lo tanto, reducir la cantidad de parámetros y complejidad computacional del modelo. Esta capa opera de manera similar a las operaciones convolucionales. Se debe seleccionar un tamaño del filtro (*kernel*), normalmente de 2x2 y un *stride.* Basado en el tipo de *pooling* que se selecciona, el filtro calcula el *output* en el campo respectivo. Existen diversos enfoques como *max pooling, average pooling*, entre otros.

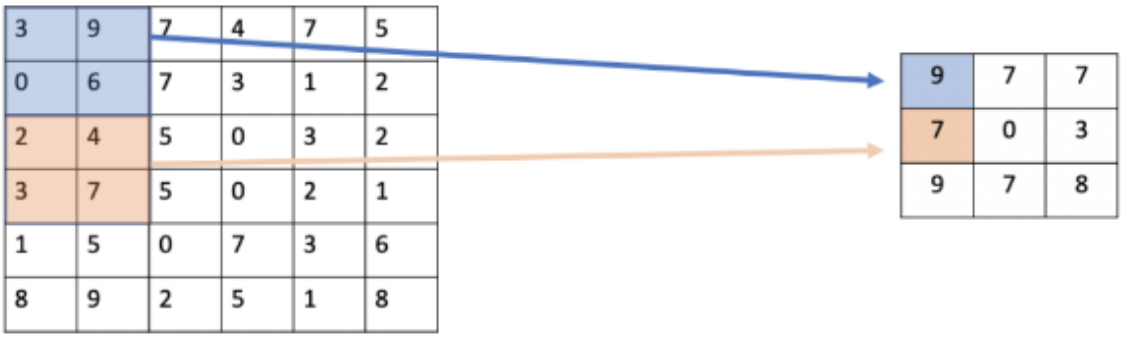


Figura 6: Ejemplo de la operación de una capa de “max-pooling”. Fuente: https://programmathically.com/what-is-pooling-in-a-convolutional-neural-network-cnn-pooling-layers-explained/

- **Capas fully-connected**

Las capas *fully-connected* contienen neuronas que están directamente conectadas a las neuronas de las dos capas adyacentes, sin estar conectadas a ninguna capa dentro de ellas. Esto es análogo a la forma en que las neuronas se organizan en una ANN.

2.4 Modelos generativos profundos

2.4.1 Descripción

En esta memoria se trabajará principalmente utilizando modelos generativos para generar nuevas imágenes. Este tipo de modelos describe como un dataset puede ser generado a partir de un modelo probabilístico. A diferencia de un modelo discriminatorio donde se busca determinar si una imagen existente pertenece o no a cierta clase, los modelos generativos buscan generar una nueva imagen que no existe anteriormente, pero que se vea tan similar a una real que no se pueda distinguir [Foster19]. Para esto se debe tener un conjunto de ejemplos para entrenar la máquina (conjunto de entrenamiento), la cual tendrá la misión de aprender la distribución de los datos, todo esto aplicado dentro de un

paradigma de aprendizaje no supervisado. Existen dos arquitecturas que se han instalado como pilares para los modelos generativos: *autoencoder* variacionales (*VAEs*) y Redes generativas adversarias (*GANs*). Este trabajo se enfoca primordialmente en definir la segunda arquitectura.

2.4.2 Redes generativas adversarias (*GANs*)

Las redes *GANs* representan un gran cambio en el diseño de la arquitectura de una red neuronal profunda normal. Esta nueva arquitectura coloca a dos redes una contra la otra en un entrenamiento para producir modelos generativos que normalmente son difíciles de entrenar (Kalin18). Existen diversas ventajas a la hora de utilizar este tipo de redes como la capacidad de generalizar con datos limitados, concebir una nueva imagen a partir de un dataset pequeño y hacer que los nuevos datos sean más realísticos. Este problema es bastante importante en aprendizaje profundo (*Deep learning*) debido a que muchas técnicas requieren una gran cantidad de datos. Usando este tipo de redes, para algunos ejercicios se puede llegar a requerir solo un diez por ciento de los datos que serían necesarios para otra red.

Una red *GAN* se compone de dos redes:

- Red discriminadora (D): Basada en un modelo discriminatorio, esta red se encarga de detectar si la imagen que se le pasa fue creada artificialmente por el modelo generativo o es parte del conjunto de entrenamiento de la red.

- Red generativa (G): Basada en modelos generativos, este tipo de red busca crear nuevas imágenes o datos similares a los del conjunto de entrenamiento con el fin de confundir a la red discriminadora.

2.4.3 Entrenamiento de una *GAN*

Las redes *GANs* se entrenan basado en un modelo matemático *minimax*. Esto consiste en entrenar las dos redes una contra la otra. La red generativa (G) es entrenada con el fin de maximizar la probabilidad de que la red discriminadora (D) pueda equivocarse en reconocer si la imagen fue generada artificialmente o no. Por otra parte, el objetivo de D es optimizarse hasta un valor de 0.5, donde el discriminador no pueda distinguir entre imágenes reales y generadas. En la figura 7 se muestra el flujo de una GAN.

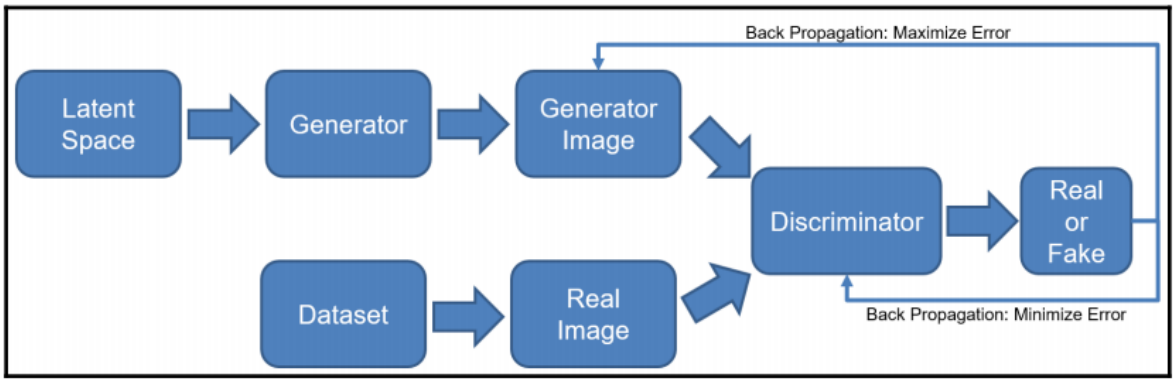


Figura 7: Flujo de una red generativa adversaria. Fuente: [Kalin18]

2.4.4 Aplicaciones de una GAN

Una GAN se pueda utilizar para una variedad de tareas dependiendo de la necesidad. Existen tres aplicaciones grandes de estas redes: Transferencia de estilo, creación de nuevas escenas (*DCGAN*) y mejoramiento de datos simulados (*SimGAN*). El documento se centrará en la primera aplicación.

2.4.5 Transferencia de estilo

La transferencia de estilo es una de las aplicaciones más utilizadas por parte de las redes *GANs.* Consiste en una técnica de visión computacional que permite transformar un dato de entrada en un nuevo dato que pertenezca a un conjunto de datos con un estilo correspondiente. Se puede aplicar tanto a la transferencia de imágenes, texto u audio, entre otros.

2.4.6 Transferencia de estilo imagen a imagen

La transferencia de estilo de imagen a imagen es la más común dentro de las aplicaciones de la transferencia de estilo. Esta permite convertir una fotografía al estilo de un artista, permitiendo crear nuevas imágenes que parezcan realizadas por el pintor. Una de las ventajas de esta técnica es que no necesita una gran cantidad de ejemplos para su entrenamiento. Hoy en día no existe una gran cantidad de ejemplos de imágenes con sus estilos, lo que provoca contar con un conjunto de datos muy limitado. Existen dos principales técnicas relacionadas a la transferencia de estilo de imágenes artísticas: redes *CycleGAN* y transferencia de estilo neuronal.

- ***CycleGAN***

La *CycleGAN* [JY17] es una adaptación de la arquitectura de una red *GAN* y permite enseñarle a un modelo como convertir una fotografía en una pintura con un estilo particular (o viceversa). Esto permite generar pinturas que se vean como si un artista la hubiese

realizado. Respecto a su arquitectura, una *CycleGAN* está compuesta por cuatro modelos, dos generadores y dos discriminadores. Los generadores se encargan de convertir imágenes de un dominio a otro mientras que los discriminadores son entrenados para poder medir si la imagen creada por el generador es convincente. Para trabajar con este tipo de red, es necesario contar con dos conjuntos de datos de entrenamiento diferentes, por ejemplo, imágenes de paisajes y obras de Monet. Una *CycleGAN* permitirá crear imágenes de ambos conjuntos, obteniendo nuevas fotografías y obras artificialmente como se muestra en la figura 8.

Figura 8: Ejemplo de una CycleGAN luego del entrenamiento de la red. Fuente: [Foster19]

- **Transferencia de estilo Neuronal**

La segunda técnica es conocida como "*Neural Style Transfer*" [Gatys15] y se caracteriza por transferir el estilo de una sola imagen en una imagen base. Para esto minimiza dos funciones de *loss*, las cuales penalizan el modelo por extraer gran parte del contenido de la imagen base, como también extraer demasiado estilo de la imagen que se le quiere transferir. Esta técnica es utilizada por muchas aplicaciones que permiten a un usuario modificar una de sus fotos con diferentes estilos de dibujo.

Figura 9: Ejemplo de Transferencia de estilo neuronal. Fuente: [Foster19]

2.4.7 Transferencia de estilo texto a texto

La transferencia de estilo texto a texto, conocida como *Text Style Transfer (TST)* es importante para la generación de lenguaje natural donde se controla el cambio de ciertos atributos como la emoción, el humor, cortesía, mientras se preserva el contenido del texto. Por ejemplo, se puede aplicar para realizar una transferencia de una frase informal (¡"Come and sit!") en una frase formal ("Please consider taking a seat"). [Jin21] Es muy importante saber diferencia el estilo y el contenido de un texto. La primera es por definición lingüística donde las características lingüísticas no funcionales se clasifican (formalidad, emoción, humor, etc..). En contraste, la segunda se basa en el contenido central del texto.

A través de la evolución de los modelos generativos profundos se le ha vuelto a dar mucha atención al procesamiento del lenguaje natural en esta época, por lo que la transferencia de estilo de texto se ha convertido en una de las mayores aplicaciones de las técnicas de visión computacional.

2.5 Image Retrieval

2.5.1 Definición y estructura

Image Retrieval o recuperación de imágenes corresponde a uno de los problemas más motivadores y de rápido crecimiento dentro del ambiente de la tecnología multimedia. Corresponde al problema de encontrar una imagen de una colección o base de datos basado en los rasgos o características de una imagen de entrada (*query image).*

Normalmente, el problema de recuperación de imágenes se basa en la similitud visual entre las imágenes, buscando la más cercana a la imagen de entrada. En problemas más complejos se puede recuperar la imagen basada en la similitud del estilo o calidad de las imágenes. Para poder comparar las imágenes se debe transformar la data de los píxeles de

cada una dentro de un espacio latente donde la representación de la imagen reflejará las características de ella.

Cada una de las imágenes en el espacio latente estarán más cerca o lejos de otra dependiendo de la similitud entre ellas. Dos imágenes cercanas estarán cerca una de la otra en el espacio mientras que dos imágenes opuestas en rasgos se encontrarán a una mayor distancia. A partir de esta regla que entrena el modelo la recuperación de imágenes se centrará en recuperar la imagen más cercana del espacio latente dada la representación de la *query.* Para esta tarea *Image Retrieval* utiliza la técnica de la búsqueda del vecino más cercano (*nearest neighbor search)*. [Bhattacharyya22]

2.5.2 Técnicas de Image Retrieval

Dada la riqueza y gran utilidad de este problema, se han desarrollado diversas técnicas para realizar *image retrieval*. Algunas de ellas, se pueden ver en la figura 10. La investigación se centrará en la recuperación de imágenes basada en texto y la recuperación de imágenes basada en contenido. [Shubhankar16]

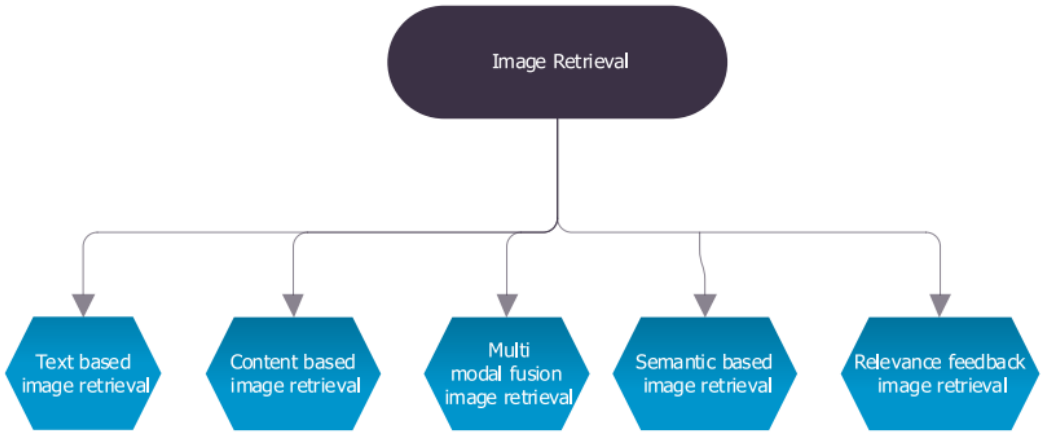


Figura 10: Técnicas de Image Retrieval. Fuente: [Shubhankar16]

- **Text based Image Retrieval**

La recuperación de imágenes basada en texto consiste en un sistema de recuperación de imágenes a partir de la información textual especificada en una *query*. En esta tarea se debe tener un *dataset* bimodal que contenga imágenes junto a su correspondiente etiqueta, título, descripción u otra información. El modelo se debe encargar de representar tanto las imágenes y textos en un mismo espacio latente para poder recuperar la imagen más cercana a partir de un texto de entrada.

- **Content based Image Retrieval**

La recuperación de imágenes basada en contenido es la técnica más común y corresponde a la tarea de encontrar una imagen a partir de la similitud entre el contenido visual de una imagen utilizando las características de ella. Un modelo de extracción de características es utilizado para recuperar los rasgos más importantes de las imágenes de una base de datos. Ejemplos de las características extraídas son la textura, color, tamaño, entre otras.

Los 4 pasos claves para este tipo de problema son:

1. Definir la extracción de características de las imágenes

En esta fase se debe decidir la forma de realizar la extracción de las características de una imagen, decidiendo enfocarse en el color, el aspecto, la textura, entre otras opciones. Normalmente en esta fase se suele utilizar redes pre-entrenadas como es el caso de una red neuronal convolucional *VGG19,* compuesta de 19 capas o una red neuronal residual Resnet152, compuesta de 152 capas.

2. Indexación del *dataset*

Luego de definir la forma en que se extraerán las características de las imágenes, se debe contar con un conjunto de imágenes, *dataset*, que no hayan sido entrenadas en el paso previo. Una vez definido, se aplica la extracción de características a cada una de las imágenes y se guardan los vectores resultantes para poder compararlos posteriormente en la búsqueda.

3. Definición de la métrica de similitud

Teniendo el conjunto de vectores de características se deben comparar a través de una métrica para poder definir cuan distante es una imagen a otra en términos de similitud. Usualmente se ocupa la distancia euclidiana, la distancia coseno o la distancia chi-cuadrado. La opción por utilizar depende altamente del *dataset* utilizado y el tipo de características que se están extrayendo de cada una de las imágenes.

4. Búsqueda

El paso final consiste en realizar una búsqueda. Un usuario enviará una imagen de consulta al sistema y su trabajo consistirá en extraer las características de la imagen consultada, utilizando el mismo método del primer paso. Posteriormente se aplicará la función de similitud para comparar el vector de entrada con el conjunto de vectores previamente indexados. A partir de ahí, el sistema puede devolver los resultados más relevantes de acuerdo con la función definida en el tercer paso.

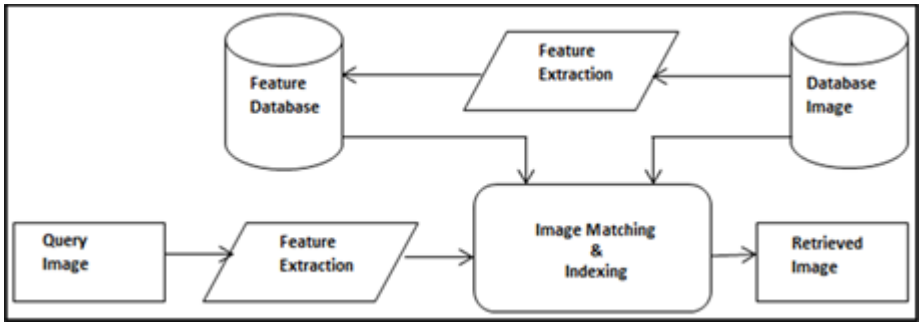


Figura 11: Arquitectura general de un sistema de recuperación de imágenes.
Fuente: https://www.scirp.org/journal/paperinformation.aspx?paperid=107008

2.6 Estado del arte

En [Alqahtani19] se revisan las diferentes aplicaciones basadas en imágenes que tienen las redes *GANs*. Se destacan la generación de imágenes de alta calidad, reconstrucción de partes de imágenes, aumentar la resolución de imágenes, detección de objetos y/o personas, manipulación de atributos faciales y transferencia de imagen a imagen. Esta última aplicación permite transferir el estilo de una imagen de entrada al estilo de una imagen de salida.

[Isola17] muestra que el modelo *pix2pix* ofrece una posible solución a la transferencia de estilo similar a las *CycleGAN*. Además de aprender a hacer un mapeo de una imagen de entrada a una de salida, *pix2pix* construye una función de *loss* para abordar el entrenamiento. Este modelo ha demostrado efectividad en sus resultados, pero posee problemas al tener que requerir en el conjunto de entrenamiento pares de imágenes iguales en contenido, pero con diferentes estilos.

Respecto a transferencia de estilo de texto a texto, [Foster19] menciona como las redes neuronales recurrentes pueden ser aplicadas para generar secuencias de texto que copien un particular estilo de escritura. Se exploran dos tipos de redes recurrentes: *LSTM* (*Long short-term memory)* y *GRU* (*Gated Recurrent Unit*). Estas técnicas son aplicables en un gran rango de problemas como son la generación de preguntas-respuestas, traducción o resumen de un texto.

Todo este tipo de técnicas se han utilizado para transferencia de imagen a imagen o texto a texto, pero no abordan una transferencia de estilo bimodal. Para poder implementar una solución de este tipo se debe tener en cuenta cómo manejar texto e imagen en un mismo espacio latente.

[Liao14] propone un modelo (*BM-LDA*) cuyo propósito es aprender una representación unificada de datos que vengan tanto de imágenes o textos. Basado en que las imágenes y sus respectivos textos comparten un mismo tópico, el modelo aprende una distribución de probabilidad en el espacio latente de las variables que permita poder unir ambas modalidades.

CAPÍTULO 3: PROPUESTA DE SOLUCION

La propuesta de solución consiste en implementar un modelo generativo profundo utilizando redes neuronales que permita generar imágenes nuevas a partir de un texto ingresado por el usuario. Esta solución generará nuevos ejemplos artificiales para su posterior uso personal o como parte de nuevos conjuntos de datos de entrenamiento.

La solución se divide en tres fases importantes. La primera consiste en realizar una recuperación de imagen de contenido a través de un *input* de texto utilizando un *dataset* bimodal (texto e imagen) para entrenar el modelo. Para ello, el modelo debe poder representar las imágenes y los textos en un mismo espacio latente. Posteriormente, utilizando un *dataset* diferente de únicamente imágenes, se obtiene la imagen de estilo más cercana a la resultante en la primera fase para luego poder ser utilizada como *input* del modelo de transferencia de estilo. Finalmente, con las dos imágenes anteriores se realiza una transferencia de estilo neuronal (*neural style transfer)* para generar una imagen nueva a través de un modelo entrenado que combina dos funciones de *loss*, contenido y estilo, en una función de *loss* combinada. Estas tres partes se muestran en la figura 12 y se detallará por separado la arquitectura utilizada para cada uno de los modelos.

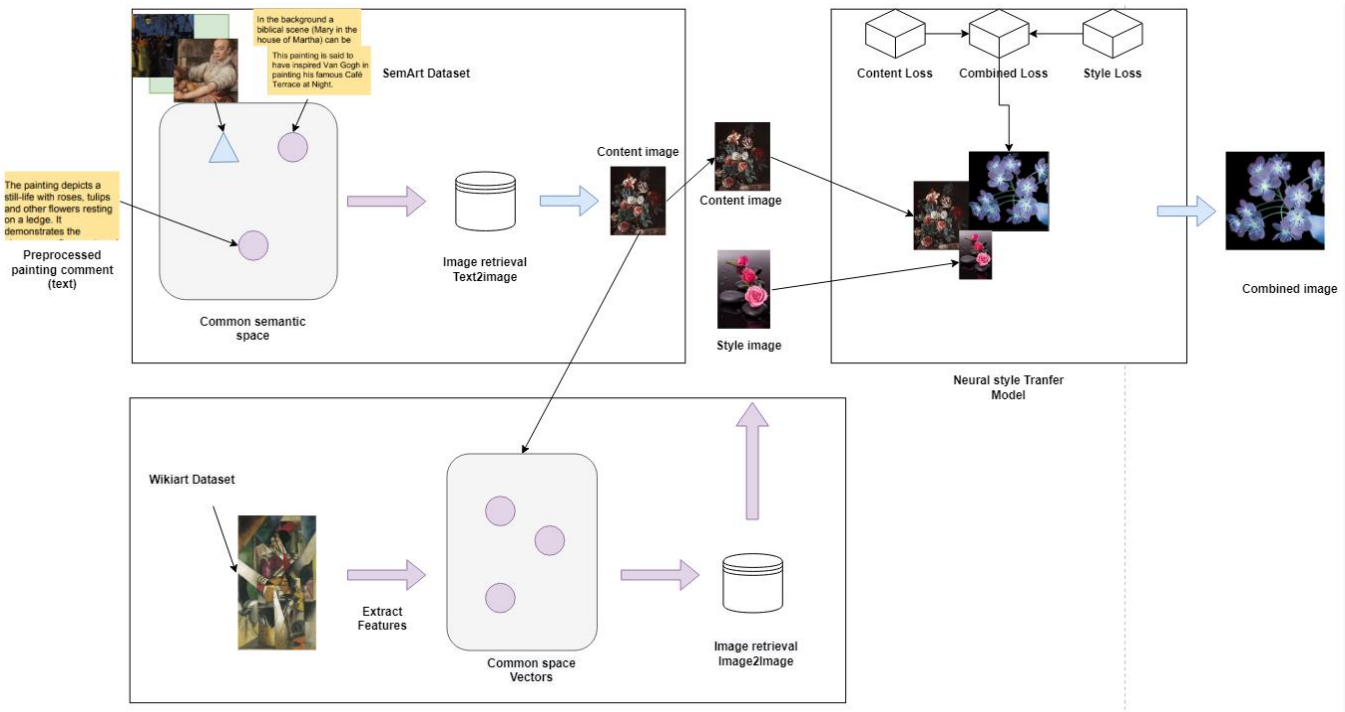


Figura 12: Propuesta de solución. Fuente: Elaboración propia

3.1 Recuperación de imagen de contenido

La primera de las etapas como se menciona anteriormente consiste en recuperar el contenido de una imagen artística a través de un texto preprocesado utilizado como *input*. Esta tarea requiere realizar un modelo *Text2Art* que permita codificar en un mismo espacio latente semántico tanto los textos artísticos como las pinturas para luego realizar una búsqueda utilizando una métrica de similitud entre el *texto* y las imágenes.

Para llevar a cabo esto se trabaja con el *SemArt* [Noa18], un *dataset* multimodal que contiene datos artísticos de imágenes y sus respectivos atributos obtenidos de una web de galería de arte (WGA), la cual contiene más de 44 mil imágenes. Contiene desde pinturas antiguas del siglo octavo hasta arte más moderno del siglo 20.

Figura 13: Dos ejemplos del *dataset SemArt*. Fuente [Noa18]

3.1.1 Estructura del *dataset*

Luego de hacer una limpieza de las imágenes que no contenían descripción se obtuvo un *dataset* de 21372 imágenes, las cuales estaban formadas por una imagen, un texto y un conjunto de atributos que se señalan en la tabla 1. Toda la información del *dataset* se guarda en un archivo *CSV.*

Al realizar un análisis de los datos se obtiene que en total existen 3300 autores diferentes aproximadamente, donde el más frecuente es Vincent Van Gogh con un poco más de 300 pinturas. Por otra parte, respecto a los títulos, existen alrededor de 15000 diferentes, lo que implica que un 38% de las pinturas presentan un título repetido. Dentro de los repetidos, el más común es “Self-Portrait”. Respecto a las escuelas artísticas, hay 26 escuelas a lo largo del *dataset* donde la más repetida es la italiana y la que esta menos representada es la finlandesa. Alguna de las distribuciones de tiempo, tipo de pinturas y escuelas artísticas se representan en la figura 14.

Con respecto a las descripciones de las pinturas, la mayoría de los comentarios poseen menos de 100 palabras.

Tabla 1: Lista de atributos del *dataset SemArt*. Fuente: Elaboración Propia.

Imagen (ubicación del archivo .jpg)
Descripción
Autor
Título
Fecha
Técnica
Tipo
Escuela
Tiempo

El *dataset* es dividido de manera aleatoria en un conjunto de entrenamiento, validación y test siguiendo una distribución del 90%, 5% y 5%, generando 19.234, 1.069 y 1.069 imágenes respectivamente divididas en tres archivos CSV junto a sus atributos.

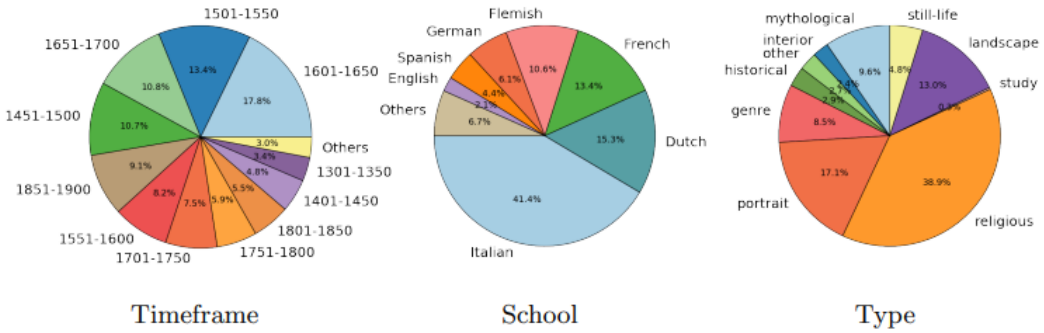


Figura 14: Distribución de los datos de diferentes atributos del *dataset SemArt.* Fuente: [Noa18]

3.1.2 Representación de texto e imágenes en un mismo espacio latente

Para poder realizar la recuperación de imágenes por medio de texto se propone realizar un *encoding de* cada uno de los textos e imágenes en sus vectores respectivos para luego ser proyectados en un mismo espacio latente a través de una función de transformación donde se aplica una función de similitud. Dada una colección de pinturas K, el K-ejemplo está compuesto por tres atributos $(img_k, title_k, desc_k)$, donde img_k corresponde a la pintura, $title_k$ su título y $desc_k$ a la descripción de la imagen artística.

Cada uno de estos tres atributos debe pasar por una función de *encoding* $(f_{img}, f_{title}, f_{desc})$ para poder mapear los datos de entrada en vectores de representación separados. Estos vectores se representan por $i_k, t_k y\ d_k$ para la imagen, título y descripción respectivamente:

$$i_k = f_{img}(img_k;\ \delta_{img})$$

$$t_k = f_{title}(title_k;\ \delta_{title})$$

$$d_k = f_{desc}(desc_k;\ \delta_{desc})$$

donde $\delta_{img}, \delta_{title}\ y\ \delta_{desc}$ corresponden a los parámetros de cada función de *encoding*.

Para trabajar con solo un vector de *encoding* para la representación de texto, se debe realizar una unión entre el vector de los títulos y descripciones a través de un vector de comentario c_k que resulta en la concatenación de las dos representaciones.

$$c_k = t_k \oplus d_k$$

Una vez que se tiene el *encoding* visual, i_k, y el de los comentarios,c_k, ambos deben pasar a través de funciones de transformación T_{vis} y T_{text} respectivamente para poder proyectarlos en un espacio multimodal común.

Los vectores de proyección se obtienen de la siguiente ecuación:

$$p_k^{vis} = T_{vis}(i_k\ ; \delta_{vis})$$

$$p_k^{text} = T_{text}\ (c_k; \delta_{text})$$

donde $\delta_{vis} y\ \delta_{text}$ corresponden a los parámetros de cada función de transformación.

3.1.3 Similitud en un espacio multimodal

Definir la similitud entre cualquier texto e imagen implica calcular la distancia entre ambas proyecciones:

$$d(p_k^{vis}, p_j^{text}) = d(T_{vis}(i_k ; \delta_{vis}), T_{text}(c_j; \delta_{text}))$$

La misión de un modelo de representación multimodal es aprender las funciones de *encoding* $f_{img}, f_{title}, f_{desc}$ y las funciones de transformación T_{vis} y T_{text} de modo que la distancia dada por la ecuación anterior entre un texto y una imagen del mismo ejemplo sea menor a cualquier distancia con otra pintura del conjunto K:

$$d(p_k^{vis}, p_k^{text}) < d(p_j^{vis}, p_k^{text}) \; \forall \; k, j \; K$$

Para evaluar la tarea de recuperación de imágenes a partir de un texto se define la tarea de obtener la imagen más cercana perteneciente al conjunto K para una *query "k"* que contenga tanto el título como la descripción de una pintura:

$$output_{img} = min \; d \; img_{j \in K}(p_j^{vis}, p_k^{text})$$

Donde p_k^{text} corresponde a la proyección del texto ingresado por parte del usuario luego de pasar por la función de encoding y de transformación respectiva.

3.2 Recuperación de imagen de estilo

La segunda etapa se basa en lograr recuperar la imagen de estilo más cercana por medio de la imagen resultante anteriormente. Para ello se propone un modelo de *Image Retrieval* que permita codificar las imágenes de un *dataset* y una de entrada en un mismo espacio latente extrayendo sus características más importantes para luego realizar una búsqueda utilizando una métrica de distancia entre los atributos de cada imagen.

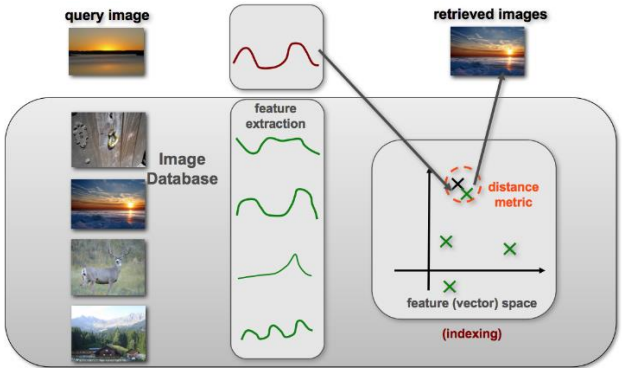


Figura 15: Diagrama de *Image Retrieval*. Fuente [Po-Chi21]

3.2.1 Estructura del *dataset*

A diferencia del *dataset* anterior, para esta etapa se utiliza uno compuesto de una menor cantidad de atributos dado que no se requiere de la descripción de cada una de las imágenes. Este se compone de 79.433 pinturas, acompañadas de su título, estilo y fecha de creación. Dentro de los estilos más comunes se encuentra el impresionismo, realismo, cubismo, romanticismo, entre otros. Por otra parte, respecto a la fecha de la pintura, todas estas obras son modernas, variando entre los años 1800 y 2000 aproximadamente. El *dataset* es generado a partir de *WikiArt*, una página encargada de coleccionar obras artísticas para el acceso de cualquier persona en el mundo. Dentro de ella existen más de 250.000 trabajos y 3000 artistas.

Figura 16: Dos ejemplos de imágenes de estilo dentro del *dataset*. Fuente: Elaboración propia

3.2.2 Extracción de características de las imágenes

Para extraer las características de las imágenes se propone un algoritmo de extracción de características de las imágenes en sus respectivos vectores y luego comparar cada uno de ellos con un vector de características extraído de una imagen de entrada. Esta comparación se realiza utilizando una métrica de distancia entre dos vectores.

Cada una de las imágenes del *dataset K debe* pasar por la siguiente función de *encoding*:

$$i_k = f_{img}(img_k)$$

Donde img_k corresponde a la imagen k $\in$ K y f_{img} a la función de extracción de características. Esta última puede ser a través de la extracción basada en los colores de la imagen, en la textura, tamaño u utilizando métodos más profundos como es el caso de una *Residual net* (*Resnet*) o una *Visual Geometry Group* (*VGG-19*).

Posterior a este proceso, se debe realizar la misma función de extracción para la imagen de entrada img_j:

$$i_j = f_{img}(img_j)$$

Para evaluar la tarea de recuperación de imágenes se define y propone la tarea de lograr obtener la imagen más cercana perteneciente al conjunto K para una *query* de entrada *j* utilizando una función de distancia *d* que puede ser distancia coseno, cuadrada, absoluta, entre otras.

$$output_{img} = min\ d\ (i_{k\in K}, i_j)$$

3.3 Transferencia de estilo neuronal

La tercera y última etapa corresponde a la transferencia de estilo neuronal. Aquí es donde, a través de métodos de aprendizaje profundo, se compondrá nuevas imágenes combinando el contenido semántico de una imagen con el estilo artístico de otra.

Para poder llevar a cabo esta técnica de aprendizaje automático se propone una solución que involucra tres imágenes:

1. Imagen de contenido: Es la que se obtiene en la primera etapa, a través de un modelo de *Text2Image*.

2. Imagen de estilo: Corresponde a la imagen recuperada en la segunda etapa utilizando un modelo de recuperación de imágenes Image2Image.

3. Imagen de entrada: Inicializada como la imagen de contenido, es la que se busca transformar en una nueva imagen de salida a través de un modelo de optimización que busca minimizar la distancia entre el contenido de la imagen con la primera y la distancia del estilo con la segunda.

Gran parte del problema radica en tomar las representaciones internas de las características aprendidas por una red convolucional (*CNN*) utilizada para clasificación de imágenes u detección de objetos con el objetivo de obtener representaciones separadas del estilo y contenido de cualquier imagen. Para ello se utilizan redes pre-entrenadas como puede ser el caso de una red *VGG-16* O *VGG-19,* las cuales permiten desarrollar representaciones internas independiente del contenido y estilo contenidas en una imagen dada.

Una vez resuelto el problema de las representaciones de contenido e imagen, se propone e implementa un método de optimización para generar una imagen nueva que vuelva a combinar el contenido y el estilo.

Dado esto, no se requiere definir una nueva arquitectura de red neuronal para la transferencia de estilo. Se propone un algoritmo en el que se pueda definir una red pre-entrenada en un *dataset* amplio, como es el caso de *ImageNet*, y una función de *loss* que permita optimizar nuestro objetivo de obtener una transferencia de estilo.

La función de *loss* se puede dividir en dos: *loss* de contenido y *loss* de estilo. La *loss* de contenido se basa en que imágenes con contenido similar tendrán una representación similar en las capas más altas de la red, dado que estas capas profundas representan características de mayor escala y, por lo tanto, tienen una representación de mayor nivel del contenido de la imagen. De manera similar, la *loss* de estilo se define como la distancia entre dos representaciones. Esta distancia es entre una imagen de estilo y una imagen de salida.

Se propone representar ambas funciones de *loss* en una función combinada cuyo objetivo es que el contenido de la imagen final sea similar al de la imagen de contenido y que el estilo de ella sea semejante al de la imagen de estilo de entrada.

$$L_{total}(c, s, x) = \alpha * L_{contenido}(s, x) + \beta * L_{estilo}(s, x)$$

Donde alfa y beta son los pesos para el contenido y estilo respectivamente, “c” corresponde a la imagen de contenido, “s” la imagen de estilo y "x" la imagen de entrada que se itera. La función se minimiza utilizando técnicas de optimización definidas.

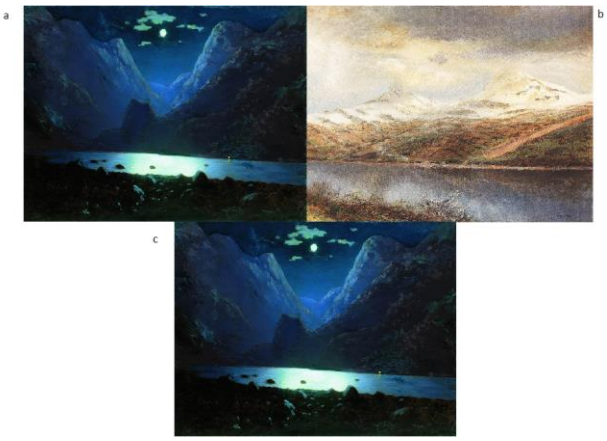


Figura 17: Representación de las tres imágenes a utilizar para el modelo de transferencia de estilo neuronal. Se observa la imagen de contenido (a), la imagen de estilo a aplicar (b) y la imagen de entrada (c), inicializada como la imagen de contenido antes de la primera iteración. Fuente: Elaboración propia.

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN

El proyecto busca implementar un modelo de transformación de estilo neuronal que permita transferir el estilo desde un texto inicial hacia una imagen de salida nueva. El modelo se separa en tres: uno para la recuperación de imagen de contenido, otro para la recuperación de imagen de estilo y, por último, un modelo para la transferencia de estilo entre las dos primeras imágenes. Se estudiarán los resultados de cada uno de ellos por separado, así como también su comportamiento dependiendo de los atributos, *encodings* y redes utilizadas.

4.1 Recuperación de imagen de contenido

Para realizar la recuperación de imágenes primero se realiza un *encoding* de cada uno de los textos e imágenes en sus vectores respectivos. Esto requiere aprender una representación tanto de las imágenes como textos por separado.

4.1.1 *Encoding* visual

Cada pintura de nuestro *dataset* se representa como un vector i_k usando redes neuronales convolucionales (*CNN).* Las CNN se caracterizan por ser utilizadas para aplicaciones de detección de objetos o clasificación de imágenes debido al uso de capas convolucionales que permiten reducir la dimensionalidad de las imágenes sin perder información importante de ella. En esta tarea se utiliza una *Residual Net* de 50 capas (*Resnet50)* pre-entrenada con el *dataset* de *ImageNet. Resnet* permite aprender funciones complejas de manera más eficiente que otras redes, ya que no sufre del "*degradation problem"*. Este problema señala que la eficiencia y performance de un modelo cae a medida que se aumenta la profundidad de la arquitectura de la red. *Resnet* utiliza *"skip-connections"* o saltos de conexiones para conectar el *input* de una capa con el *output* de una más profunda.

La arquitectura de esta red posee 4 etapas, tal como se ve en la figura 18. La red recibe una imagen de entrada y realiza una convolución inicial seguida de una capa de max-pooling usando tamaños del kernel de 7x7 y 3x3 respectivamente. A continuación, en la primera etapa se tienen tres bloques, los cuales contienen tres capas convolucionales cada uno. A medida que se progresa de una etapa a otra el tamaño del kernel aumenta. Finalmente, luego de la última etapa se utiliza una *average pooling layer* seguida de una *fully-connected layer* que contiene 1000 nodos con una función de activación *softmax.* Para nuestro problema de obtener el *encoding* visual se utiliza el *output* de la última capa de la red.

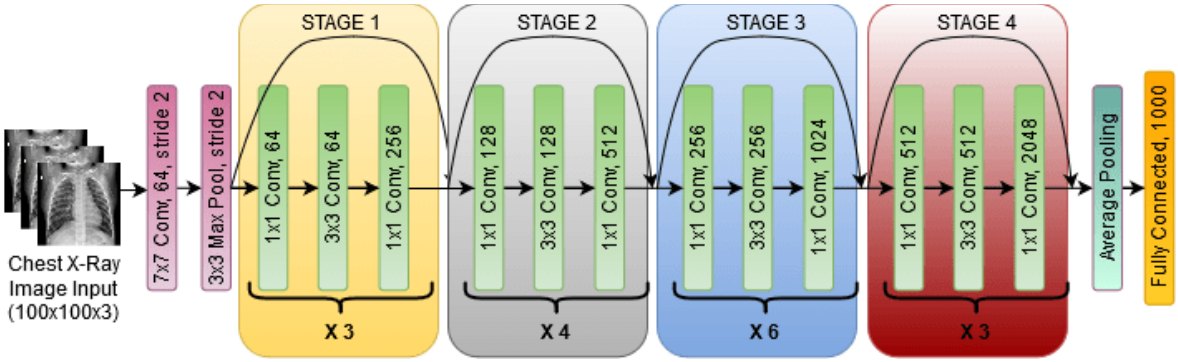


Figura 18: Arquitectura de Resnet50. Fuente: [Khan20]

Cada una de las imágenes del dataset *SemArt* antes de pasar por la etapa de entrenamiento pasa por un proceso de transformación:

1. La imagen se re-escala a un tamaño de 256x256.
2. Se realiza un proceso de centrado de la imagen re-escalada.
3. Se utiliza *data-augmentation* haciendo un proceso de centrado aleatorio con un tamaño de 224. Además, se realiza un *flip* de la imagen horizontalmente de manera aleatoria.
4. Por último, se normaliza la imagen realizando una sustracción tanto de la media como la desviación estándar con respecto al *dataset* ImageNet con el que se pre-entrena la red.

4.1.2 *Encoding* Textual

Con respecto a la información textual de cada pintura del *SemArt*, las descripciones son codificadas en un vector d_k y los títulos en un vector t_k. Para obtener el *encoding* final, ambos vectores son concatenados en un vector resultante c_k.

En el caso de la codificación de las descripciones, primero se arma un vocabulario V_d. V_d contiene todas las palabras que aparecen al menos diez veces en todo el conjunto de entrenamiento, eliminando también las *stopwords.* El vector de las descripciones se obtiene utilizando la técnica de *Bag-of-words (BOW).* Esta técnica permite codificar los textos en términos de su frecuencia en el documento (*term frequency – inverse document frequency o tf-idf).* Esta medida numérica expresa cuán relevante es una palabra dentro de un corpus. El vocabulario luego de la limpieza resulta de 9758 palabras.

Para los títulos de las pinturas, se sigue un proceso similar, armando un vocabulario V_t. A diferencia del vocabulario anterior, acá no se eliminan las palabras con una frecuencia menor a diez dentro del texto. Esto es debido a que la cantidad de palabras en los títulos es mucho menor a las descripciones, provocando que el vocabulario sea mucho menor. Pese

a esto, al igual que en V_d, se eliminan las *stopwords* dentro del corpus. Los títulos también se codifican utilizando el método de *Bag-of-words*. El vocabulario luego de la limpieza es de 9205 palabras.

Es importante mencionar que, dado que el *SemArt* es un *dataset* de pinturas en todo el mundo, las descripciones y títulos son en inglés, por lo que los comentarios de entrada en la experimentación deben ser en este idioma.

4.1.3 Transformación multimodal

Tanto los *encoding* visuales i_k como los textuales c_k codifican el contenido en dos espacios diferentes. Se utiliza un modelo de transformación multimodal para trasladar ambas representaciones en un espacio común. En este espacio, ambas informaciones visuales y textuales pueden ser comparadas en términos de una función de similitud.

El modelo utilizado es el de *Cosine Margin Loss (CML).* Esta arquitectura de aprendizaje profundo está entrenada para aprender los *encoding* visuales y textuales y sus proyecciones al mismo tiempo. Cada imagen codificada por la *ResidualNet pasa por una capa fully-connected* con un tamaño del *embedding de 28* acompañada por una función de activación tangente hiperbólica y una capa de normalización l_2 para proyectar las características visuales en un espacio dimensional común D, obteniendo el vector de proyección p_k^{vis}. De manera similar, cada vector de comentarios c_k es la entrada dentro de una red con la misma estructura que la anterior: una capa *fully-connected* con función de activación tangente hiperbólica y una capa de normalización l_2. Lo anterior genera el vector de proyección p_k^{text} que se encuentra dentro del mismo espacio dimensional D.

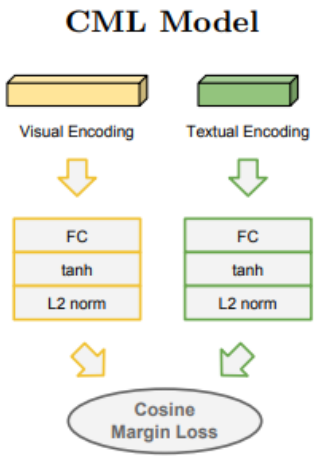


Figura 19: Modelo de transformación multimodal. Fuente: [Noa18]

La función de activación hiperbólica está dada por la siguiente ecuación:

$$\tanh(\mathrm{x}) = \frac{\mathrm{e}^{\mathrm{x}} - \mathrm{e}^{-\mathrm{x}}}{e^{x} + e^{-x}}$$

El modelo *CML* es entrenado con todos los pares de imágenes y textos coincidentes ($k = j$) y no coincidentes ($k \neq j$) utilizando una función de *loss*:

$$L_{CML}\left(p_k^{text}, p_j^{vis}\right) = \begin{cases} 1 - \cos\left(p_k^{text}, p_j^{vis}\right), & k = j \\ \max\left(0, cos\left(p_k^{text}, p_j^{vis}\right) - m\right) & k \neq j \end{cases}$$

Donde $\cos(x, y)$ corresponde a la similitud coseno entre dos vectores normalizados y m al margen de la función de *loss*. Este último valor oscila entre -1 y 1, siendo sugerido un valor entre 0 y 0.5. Para el entrenamiento se utiliza un valor de m = 0.1.

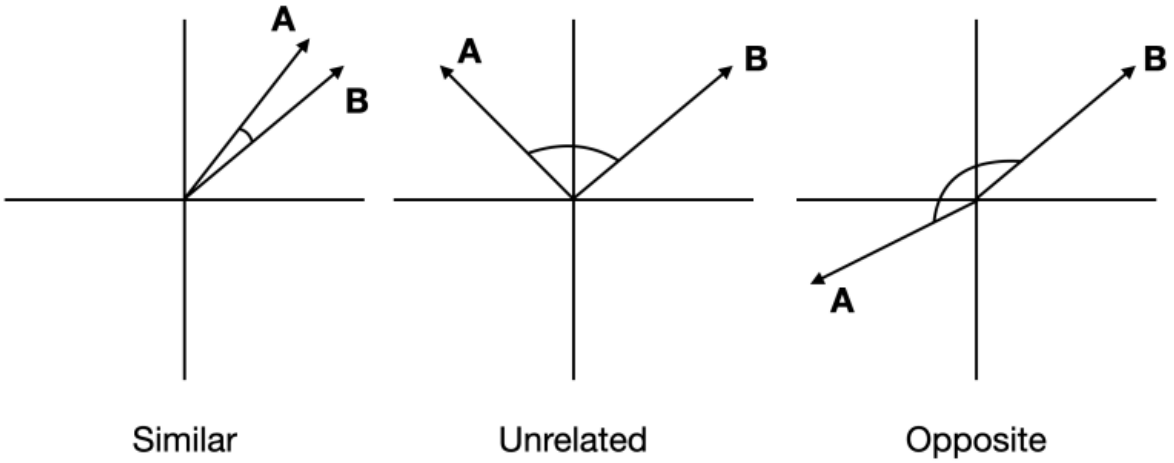


Figura 20: Tres ejemplos de similitud entre dos vectores usando el angulo coseno.
Fuente: https://researchdatapod.com/how-to-calculate-similarity-python/

La similitud coseno utilizada es una medida de similitud que se calcula entre dos vectores distintos de cero dentro del espacio interno del producto que mide el coseno del ángulo entre ellos. La función proporciona un valor igual a 1 si el ángulo comprendido entre ambos vectores es cero, es decir, si ambos apuntan a un mismo lugar. Su rango de valores se encuentra en el intervalo cerrado [-1,1]. La fórmula para calcularlo está dada por la siguiente ecuación:

$$\cos\left(\vec{a}, \vec{b}\right) = \frac{\vec{a} * \vec{b}}{\|\vec{a}\|\|\vec{b}\|}$$

4.1.4 Experimentos y resultados

Para la experimentación se trabaja con el conjunto de entrenamiento del dataset *SemArt* compuesto 19.234 imágenes junto a su descripción y título. También se utiliza un set de imágenes de validación compuesta de 1069 imágenes. Para la codificación de imágenes se inicializa la red *Resnet50* con sus pesos pre-entrenados para clasificación de imágenes. Cada una de las imágenes pasa por el proceso de transformación mencionado anteriormente. En la sección de codificación de texto se trabaja con los vocabularios V_d y V_t utilizando la técnica de bag-of-words (BOW) para codificar cada una de las descripciones y títulos.

Para el modelo de transformación multimodal de *Cosine Margin Loss* se utiliza el método de optimización de gradiente descendente *Adam Optimizer* con un *learning rate de 0.0001* y un margen de 0.1. El entrenamiento utiliza un *batch_size* de 28. La comparación entre los vectores de imágenes y texto proyectados se realiza mediante la similitud coseno descrita en la sección anterior. Se realizan 10 iteraciones (*epochs),* entregando los resultados que se muestran en la tabla 2 y las figuras 20 y 21.

La evaluación del modelo consiste en ranquear las pinturas de acuerdo con su similitud para un texto dado, aplicándose en todo el conjunto de prueba. Los resultados se basan en dos medidas. La primera medida se conoce como *Recall Rate at K* (R@K) con K variando entre 1,5 y 10. Esta consiste en la proporción de imágenes para la cual la imagen correspondiente de cada texto está en el top K de posiciones de acuerdo con el ranking de similitud obtenido de cada una de ellas. La segunda medida es la *Median Rank* (MR) y se basa en la mediana del ranking para todas las imágenes.

Tabla 2: Resultados de las iteraciones del modelo CML utilizado para la recuperación de imagen de contenido. Fuente: Elaboración propia.

N° iteración	MedRank (MR)	R @ 1	R @ 5	R@10
1	25	0.08	0.23	0.34
2	22	0.10	0.26	0.37
3	10	0.10	0.27	0.38
4	19	0.11	0.29	0.40
5	20	0.11	0.27	0.38
6	18	0.12	0.29	0.40
7	18	0.12	0.30	0.41
8	16	0.13	0.31	0.43
9	15	0.13	0.31	0.45
10	15	0.14	0.32	0.45

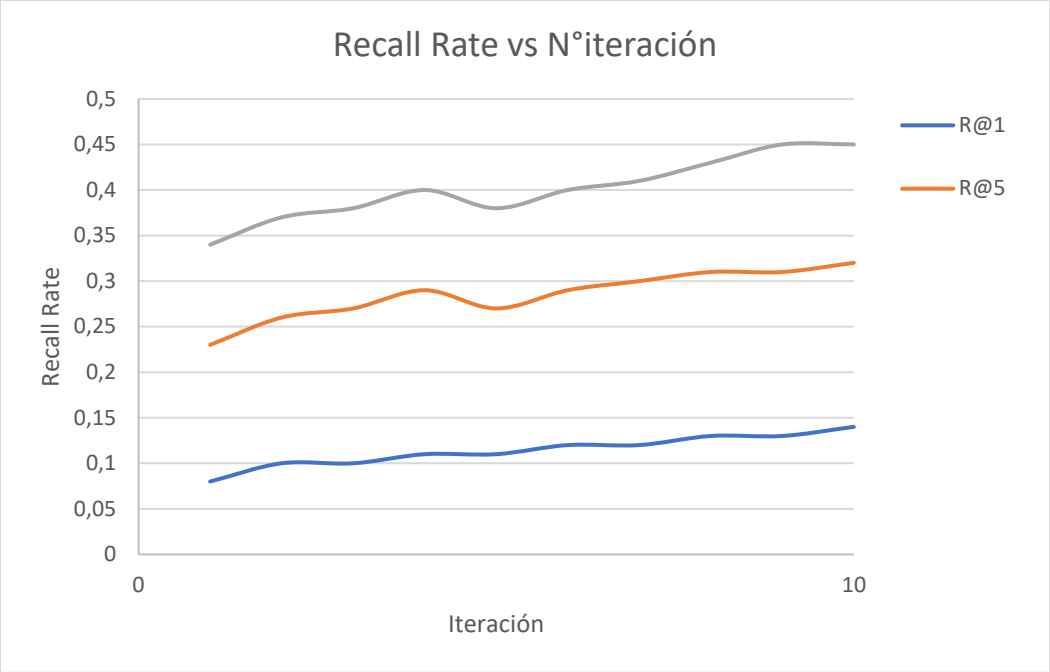


Figura 21: Gráfica de dispersión de resultados del *recall rate* (R@K) para cada iteración del modelo CML. Fuente: Elaboración propia.

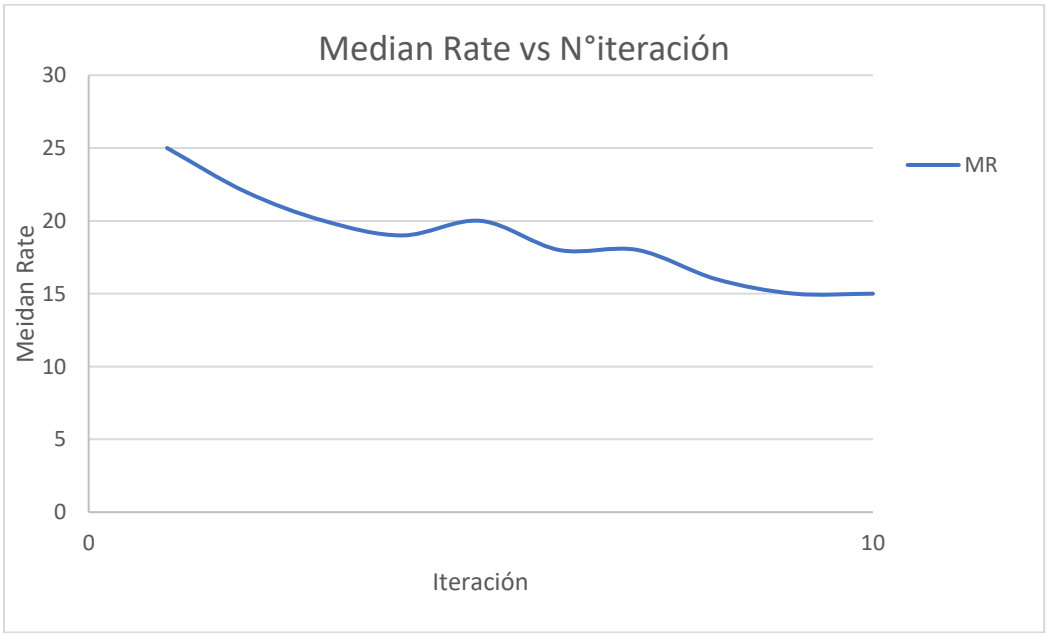


Figura 22: Gráfica de dispersión de resultados del *median rate* (MR) para cada iteración del modelo CML. Fuente: Elaboración propia.

La tabla 2 muestra los resultados del entrenamiento de la red. A partir de estos resultados se puede ver que, al realizar la recuperación de las imágenes más similares a un texto determinado, la imagen correspondiente de cada texto se encuentra en la posición quinceava de mediana dentro del conjunto de imágenes del set de datos. Respecto a las medidas de *recall*, se observa que en un 45% de las veces la imagen que sí corresponde al texto está dentro del top 10 de imágenes más cercana al input de entrada. Por otro lado, solo en un 15% aproximadamente de los casos la imagen más cercana corresponde a la verdadera. Esto puede deberse a que dentro del conjunto de imágenes existen muchas semánticamente similares al texto de entrada.

En las figuras 23 y 24 se muestran al lado izquierdo el top 5 de imágenes del conjunto de prueba más cercanas para un título y descripción de entrada. En el lado derecho se muestra la imagen real del conjunto de prueba que corresponde al texto de entrada. Cada una de las obras posee su distancia (similitud coseno), el título y descripción correspondiente.

Para el primer caso (figura 23), donde se describe diversas frutas sobre una cesta, aun estando la imagen verdadera correspondiente al texto en la posición doceava, las primeras cinco imágenes se pueden corresponder en gran medida con la descripción de entrada. El top de imágenes se mueve dentro de valores de similitud sobre 0.5, correspondiéndose con el intervalo que puede llegar a moverse la similitud coseno entre dos vectores similares.

El segundo caso (figura 24) describe de manera más compleja a Cristo en la cruz acompañado de la virgen y el evangelista San Juan. A diferencia de la anterior, las imágenes más cercanas solo se corresponden parcialmente con el texto de entrada, incluyendo a solo algunas de las personas descritas anteriormente. Pese a esto, el contenido semántico es similar al incluir un panel central que involucra a la virgen o cristo en ella en un contexto religioso. Para esta búsqueda, la similitud es menor, oscilando entre valores entre 0.45 y 0.55, pero encontrando la imagen verdadera de manera más rápida en la posición seis del ranking.

Título

Still-Life of Apples, Pears and Figs in a Wicker Basket on a Stone Ledge

Descripción

The large dark vine leaves and fruit are back-lit and are sharply silhouetted against the luminous background, to quite dramatic effect. Ponce's use of this effect strongly indicates the indirect influence of Caravaggio's Basket of Fruit in the Pinacoteca Ambrosiana, Milan, almost 50 years after it was created

Distancia: 0.73
Título: Still-Life with Fruits
Descripción: Notable among the artists Jan Davidsz. de Heem trained during his stay in Utrecht who then worked in his manner is the flower and fruit specialist Abraham Mignon

Distancia: 0.72
Título: Maiolica Bowl with Peaches, Grapes, and Bees
Descripción: In this version of Panfilo Nuvolone's celebrated composition of grapes and peaches, the succulent fruit, floating against a dark background, is arranged in a maiolica bowl and seen from a particularly high viewpoint. A mysterious light shines from above, casting a shadow around the bowl and not just to one side, as is most often the case in the other versions

Distancia: 0.70
Título: Still-Life with Fruit and Crystal Vase
Descripción: Intense colours characterize the still-lifes of Van Aelst, who worked for a time in France and Italy. In this painting, which was commissioned by Cardinal Giovan Carlo de' Medici, the intense blue of the table-cloth catches one's eye. The large crystal vase in the centre of the composition was probably in the possession of the Medici family. The painter placed his signature and the date on the hem of the white tablecloth: W.A.Aelst. 1652

Distancia: 0.69
Título: Still-Life
Descripción: The painting represents a still-life with fruit, fish and dead game arranged on table-top. It is signed and dated lower right: H Steenwyck/ 1640

Distancia: 0.69
Título: The Five Senses
Descripción: In this still-life the Five Senses are represented in the form of objects. Hearing is clearly given a greater value than all the other senses. The open hymn-book with the words of thanksgiving 'Laudate dominum' forms a clear contrast to the reprehensible game of cards and the empty purse beside it - two objects representing the sense of touch

Distancia: 0.60
Título: Still-Life of Apples, Pears and Figs in a Wicker Basket on a Stone Ledge
Descripción: The large dark vine leaves and fruit are back-lit and are sharply silhouetted against the luminous background, to quite dramatic effect. Ponce's use of this effect strongly indicates the indirect influence of Caravaggio's Basket of Fruit in the Pinacoteca Ambrosiana, Milan, almost 50 years after it was created

Figura 23: A la izquierda el ranking de Top 5 imágenes más cercana para el texto de entrada. A la derecha la distancia a la imagen correspondiente al texto de entrada, la cual se ubica en el lugar 12 del ranking. Fuente: Elaboración propia.

Título

The Galitzin Triptych: Christ on the Cross

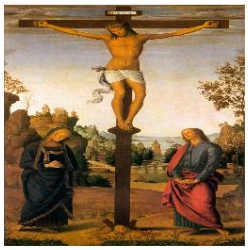
Descripción

The picture shows the central panel of the triptych representing Christ on the Cross between the Virgin and St John the Evangelist

Distancia: 0.52
Título: Triptych
Descripción:
The triptych with the Virgin and Child Enthroned in Vienna must have been painted before 1488. The present donor, dressed in a dark tabard, is not by Memling and conceals another figure who turns out to be none other than Abbot Jan Crabbe, who died in 1488. The wings show the two St Johns with the `living' figures of Adam and Eve in niches on the exterior (seen when the wings are closed).For the first time the painting includes the italianate motif of putti stretching festoons across the scene at the top. Similar figures appear on a couple of other occasions in Memling, including the 'Resurrection triptych (Paris, Musée du Louvre) and the Virgin Enthroned in Florence

Distancia: 0.52
Título: Triptych of Adriaan Reins
Descripción: This small triptych is the second altarpiece to be commissioned from Memling by a brother at St John's Hospital in Bruges, the first being the Floreins triptych. The donor has been identified as Adriaan Reins on the basis of the initials AR on the frame and the figure of St Adrian who protects him in the left wing. Reins was received into the Order in 1479, and died in 1490.The principal scene is a Lamentation on Golgotha beneath a glowering evening sky.

Distancia: 0.48
Título: St James of the Marches with Two Kneeling Donors
Descripción: St James of the Marches (1391-1476) (Italian: Giacomo della Marca) was an Italian Franciscan Friar Minor, preacher and writer)



Distancia: 0.46
Título: The Galitzin Triptych: Christ o the Cross
Descripción: The picture shows the central panel of the triptych representing Christ on the Cross between the Virgin and St John the Evangelist

Distancia: 0.47
Título: Trinity
Descripción: In a theme that was majestically treated by Masaccio a century earlier, Beccafumi does not retain an orderly sense of scale for his figures, so that God the Father and the crucified Christ are much smaller in relation to the side saints (Sts Cosmas and John the Baptist at left, Sts John the Evangelist and Damian at right). All of his figures seem to deny pure volumetric presence, this despite the fact that Beccafumi was accomplished both as a bronze sculptor and painter

Distancia: 0.47
Título: San Zeno Polyptych
Descripción: The central part represents the Madonna and Child Enthroned (212 x 125 cm), the left part shows Saints Peter and Paul, Saint John the Evangelist and Saint Zeno, while the right part Saints Benedict, Lawrence, Gregory and Saint John the Baptist (235 x 135 cm each). The predella paintings are copies, the originals are in the Musée du Louvre and in the Musée des Beaux-Arts of Tours.Despite the framing elements that divide them, the three main panels of the San Zeno Altarpiece form an unified picture space.

Figura 24: A La izquierda el ranking de top 5 imágenes más cercana para el texto de entrada. A la derecha la distancia a la imagen correspondiente al texto de entrada, la cual se ubica en el lugar 6 del ranking. Fuente: Elaboración propia.

Los dos primeros casos utilizan de entrada textos que forman parte del conjunto de prueba, los cuales contienen una imagen asociada para poder medir su desempeño en la capacidad de encontrar la imagen verdadera en un ranking de imágenes más cercana. Los siguientes dos casos corresponden a textos nuevos ingresados por un usuario, los cuales no tienen una imagen propia asociada ni se encuentran dentro de uno de los conjuntos de datos. Cada uno de estos textos pasa por el mismo proceso de *encoding* para posteriormente ser proyectado y comparado con cada una de las imágenes del *dataset*. Para ello, se aplica la misma función de similitud coseno utilizada anteriormente. Los resultados se muestran en la figura 25 y 26.

Para la figura 25, se ingresa un texto describiendo un paseo en bote acompañado de personas mientras se observa la ciudad iluminada por un sol radiante. La primera, tercera y cuarta imagen del ranking describen de manera casi perfecta el comentario ingresado, incluyendo una vista de un bote, una ciudad y un paisaje soleado. Por otra parte, pese a no mostrar específicamente un bote en la segunda y cuarta imagen, se rescatan otros aspectos del comentario como lo son la vista de una ciudad y la presencia de personas en la foto. Además, la segunda imagen permite la confusión entre el agua y la nieve de invierno que se muestra. La similitud entre el texto ingresado y las imágenes varía entre un 0.5 y 0.65, teniendo un desempeño similar a los textos del conjunto de prueba mencionados anteriormente.

En el caso de la figura 26, se ingresa un texto más simple, el cual menciona la vista panorámica de una montaña en un cielo nublado. Para este *input*, los cinco resultados incorporan montañas, siendo las más claras las imágenes 1, 4 y 5. Pese a ser un texto breve y de mayor interpretación, dado que solo se mencionan las palabras clave de montañas y clima nublado, los resultados arrojados son bastante positivos, entregando imágenes que cumplen con estas características. La mayor diferencia se da en la imagen 3, donde se incorpora un ángel arriba de las montañas. Esta se pudo ocasionar debido a la longitud del texto ingresado y a lo acotado que es el conjunto de imágenes con las que se realiza la búsqueda. En términos de similitud basada en distancia, es uno de los resultados con mayor exactitud variando sus valores entre 0.68 y 0.72, comparándose con los mejores resultados entregados por el conjunto de prueba para textos definidos en el *dataset* como fue el de la figura 23.

Título

A boat trip on a sunny day

Descripción

From a boat accompanied by people you can see the city illuminated by a radiant sun.

Distancia: 0.62
Título: The Giudecca Canal with the Zattere
Descripción: This view is dominated by the great marble bulk of the Church of Gesuati built in the 18th century

Distancia: 0.61
Título: Winter Landscape
Descripción: Two out of three winters in Holland during the 17th century saw particularly extended periods of frosts and snow. During the 1660s the level of snowfall increased significantly and two of the harshest winters were in 1662-63 and 1671-72. Dutch artists, among them Jacob van Ruisdael and Klaes Molenaer produced many winter landscape in this period. Molenaer's winter scenes are characterised by a great interest in the harmony and balance between the figures and their surroundings, more akin to the work of the Haarlem painter of the previous generation, Isack van Ostade, than the atmospheric landscapes of Jacob van Ruisdael.The landscape elements in the present scene are broadly repeated in other works by Molenaer.

Distancia: 0.55
Título: View of the Giudecca Canal
Descripción: The most original paintings by the Swedish artist depict the Venetian lagoons enlivened by boats. The architecture is represented down to the smallest detail. He frequently used the 'camera obscura' like in the case of this painting.

Distancia: 0.54
Título: The Bacino di San Marco
Descripción: The painting shows the profile of Venice as it appears from the present-day Giardini when approaching the city by boat from the Lido. To the far left is the island of San Giorgio Maggiore, with the church of the same name and the complex of the Benedictine monastery. Just to the right of the island appear the Magazzini del Sal, dazzlingly illuminated, the Dogana and Santa Maria della Salute on the spit of land known as the Dorsoduro. The Campanile of San Marco looms right on the vertical axis, counterbalancing the predominant horizontals. On the right, the Riva degli Schiavoni stretches beyond the church of the Pieta.The glassy green surface of the water with the dark bank in the foreground, the contrasting light on the various buildings and the snow-white clouds lend the painting an atmosphere all its own.

Distancia: 0.53
Título: View of the Campo San Zanipolo in Venice
Descripción: This painting is a view of the Campo San Zanipolo with the loggia temporarily erected outside the Scuola di San Marco for the benediction of Pope Pius VI on 19 May 1782.In May 1782, Pope Pius VI visited the city of Venice. The ceremonies and festivities on that occasion were recorded by Francesco Guardi in a series of four paintings. The city council commissioned Guardi to make four paintings representing the following subjects: the arrival of the pontiff near San Giorgio in Alga, the papal mass in SS Giovanni e Paolo, the papal audience and the blessing of the people on Campo SS Giovanni e Paolo.On the occasion of the Pope Pius VI blessing the crowd a temporary wooden loggia, designed by Antonio Codognato, was erected on the façade of the Scuola di San Marco.

Figura 25: Cinco imágenes más cercanas para el texto de entrada citado en la parte superior por parte de un usuario. Fuente: Elaboración propia.

Mountain View

Descripción

Panoramic view of the mountains in a cloudy sky

Distancia: 0.72
Título: View of Monticelli, near Tivoli
Descripción: Settled in Rome, Denis made studies of the landscape surrounding the city. He painted outdoors in all seasons, carefully studying and documenting the effects of light in various weather conditions. The present painting is described in an inscription on the reverse: "Le temps qui s'eclairci, peu à peu, après l'orage" ("The weather clears, little by little, after the storm")

Distancia: 0.71
Título: A Danish Coast
Descripción: The painting shows a view from Kitnas on Roskilde Fjord, Zeeland. When the painter decided to depict a Danish coastline in an unusually monumental format, the scene was to represent the quintessential characteristic of the Danish countryside: the extensive coastlines. The scene does not, in fact, depict a single, specific site, the main motif is based on cliffs that the artist saw and sketched at the Roskilde Fjord. He adapted these cliffs, making them rather more monumental, and combined them with scenes from elsewhere

Distancia: 0.71
Título: Mary Magdalen Raised by Angels
Descripción: Shortly after the completion of the ceiling in the Galleria of the Palazzo Farnese, Annibale and his pupils, who included Domenichino and Giovanni Lanfranco, were asked to provide decorations for a palazzetto behind Palazzo Farnese, across the Via Giulia. The decoration consisted of mainly landscapes. In keeping with the character of the building, a casino in a garden, they represented mythological episodes. A few years later, Cardinal Odoardo Farnese decided to add a small room (Camerino degli Eremiti) to the Palazzetto Farnese as a Christian retreat where he could withdraw for his devotion.

Distancia: 0.70
Título: Landscape with an Approaching Shower
Descripción: The painting depicts two figures by a stream in a landscape with an approaching shower

Distancia: 0.68
Título: The painting depicts two figures by a stream in a landscape with an approaching shower
Descripción: Glover had a house at Blowick Farm near Patterdale, at the foot of Ullswater Lake, and spent much of his time in the Lake District. He was so fond of the region that, when in Tasmania, where he settled from 1831, he reminiscently named his house Patterdale

Figura 26: Cinco imágenes más cercanas para el texto de entrada citado en la parte superior por parte de un usuario. Fuente: Elaboración propia.

4.2 Recuperación de imagen de estilo

Realizar la recuperación de la imagen de estilo requiere crear un *encoding* de cada una de las más de 70.000 imágenes del *dataset WikiArt.* Al igual que en el paso anterior, se debe aprender una representación vectorial de cada una de las imágenes para luego ser comparados mediante una métrica de distancia.

4.2.1 *Encoding* Visual

Cada imagen del d*ataset* representada por un vector i_k previamente a su codificación pasa por un proceso de:

1. Redimensión y centrado de la imagen a una de tamaño 256x256 píxeles.
2. Normalización de los píxeles dividiéndolos en 255.
3. Estandarización de los píxeles restándole la media de los tres canales. Esta media corresponde a un arreglo de [103.939, 116.779, 123.68]/255 en orden de escala de colores BGR.

Luego, se utiliza una *Residual Net* de 152 capas (*Resnet152)* pre-entrenada con el *dataset* de *ImageNet* con el objetivo de obtener el *encoding* de cada imagen. *La* arquitectura de esta red es similar a la *Resnet50* presentada anteriormente, diferenciándose por la cantidad de capas convolucionales utilizadas. Al igual que la Resnet50, esta red profunda utiliza "*skip-connections"* entre los distintos bloques para conectar el *input* de una capa con el *output* de otra.

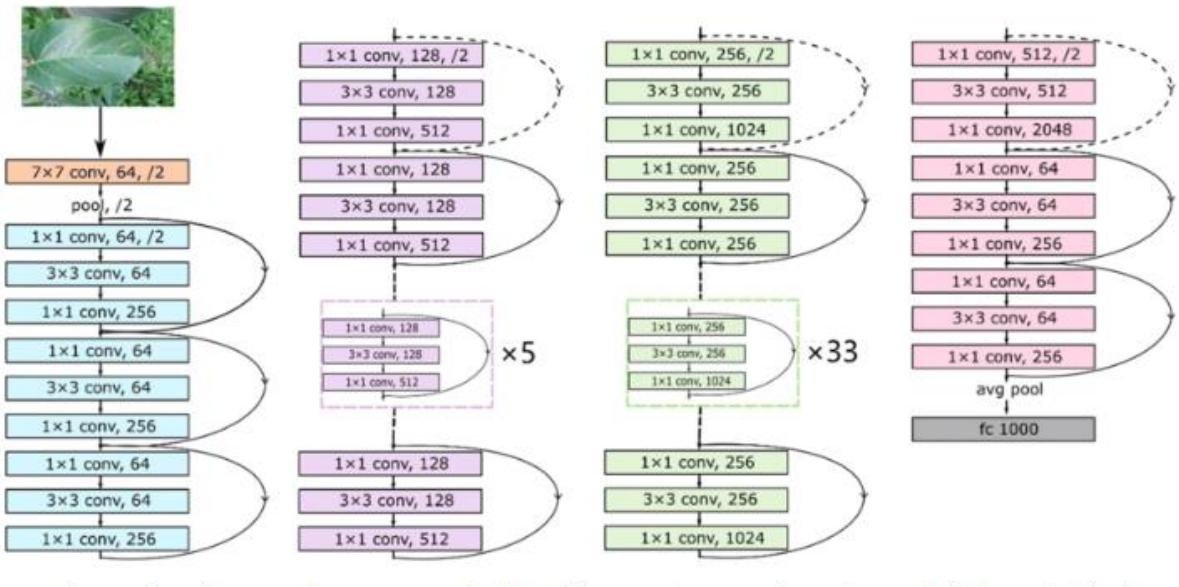


Figura 27: Arquitectura de una *Resnet152.* Fuente: [Chongke22]

4.2.2 Experimentos y resultados

La experimentación realizada se basa en el conjunto de imágenes extraídas del *dataset* de *WikiArt* compuesto de 79.433 imágenes. Cada una de ellas pasa por una función de codificación donde se le realiza la transformación mencionada anteriormente y se le aplica un modelo pre-entrenado de clasificación de imágenes *Resnet152*. El vector de *encoding* de cada imagen es el resultado de la extracción de la salida de la capa de *Average Pooling* de la red. Cada uno de los vectores i_k es almacenado en una lista "K" junto al detalle de la imagen correspondiente.

La imagen de entrada "j" que ingresa el usuario entra por el mismo proceso de codificación para poder almacenar su vector i_j en una variable. Luego, este vector se compara utilizando la métrica de distancia coseno entre el vector i_j y los vectores i_k pertenecientes a la lista "K". La métrica de distancia utilizada es la distancia coseno, la cual está dada por la fórmula:

$$\mathrm{d}_{cos}(\vec{a}, \vec{b}) = 1 - \frac{\vec{a} * \vec{b}}{\|\vec{a}\|\|\vec{b}\|}$$

Donde $\vec{a}, \vec{b}$ corresponden a vectores representativos de una imagen. Distancias más cercanas a cero implican que la similitud entre ambas imágenes es mayor.

Los resultados de la experimentación se muestran desde la figura 28 hasta la figura 31. En las dos primeras figuras se muestran las cinco imágenes más cercanas para cada imagen de entrada (a la izquierda de la figura). Para ambos casos se utilizan como *input* resultados de la sección anterior de recuperación de imagen de contenido como *input*, obtenidas a partir del texto de entrada de la figura 25 utilizando el modelo de la sección 4.1. Por otro lado, en las figuras 30 y 31 se ocupan como imagen de entrada dos de los resultados logrados en la figura 26.


Distancia: 0.104


Distancia: 0.109


Distancia: 0.113


Distancia: 0.117

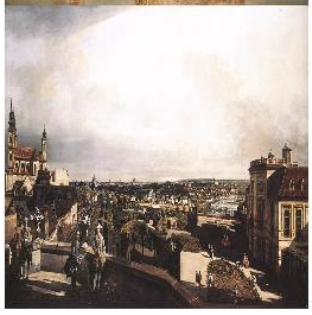

Distancia: 0.119

Figura 28: Top 5 de imágenes más cercanas a la imagen de entrada (izquierda). Fuente: Elaboración propia.

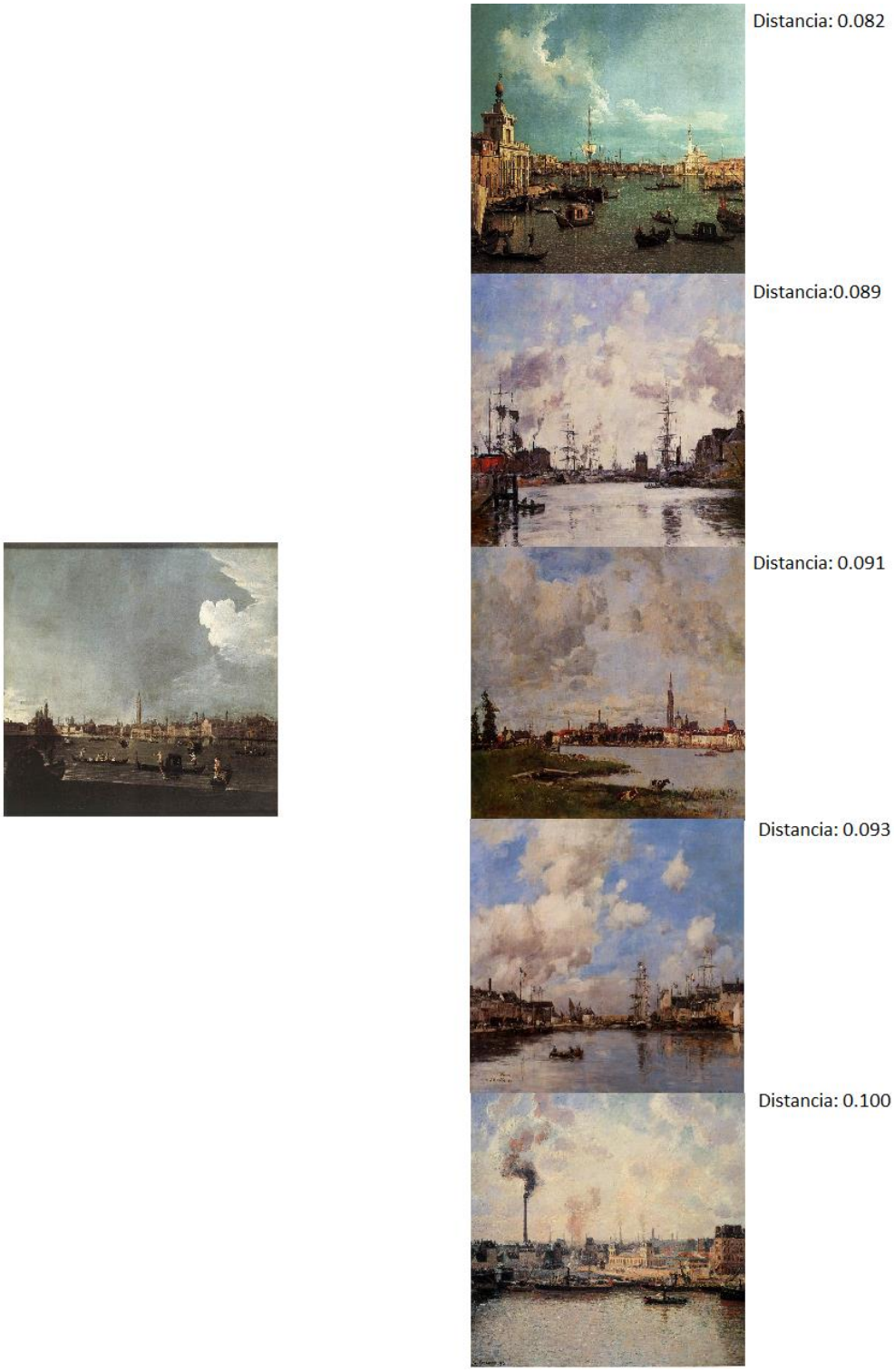


Figura 29: Top 5 de imágenes más cercanas a la imagen de entrada (izquierda). Fuente: Elaboración propia.

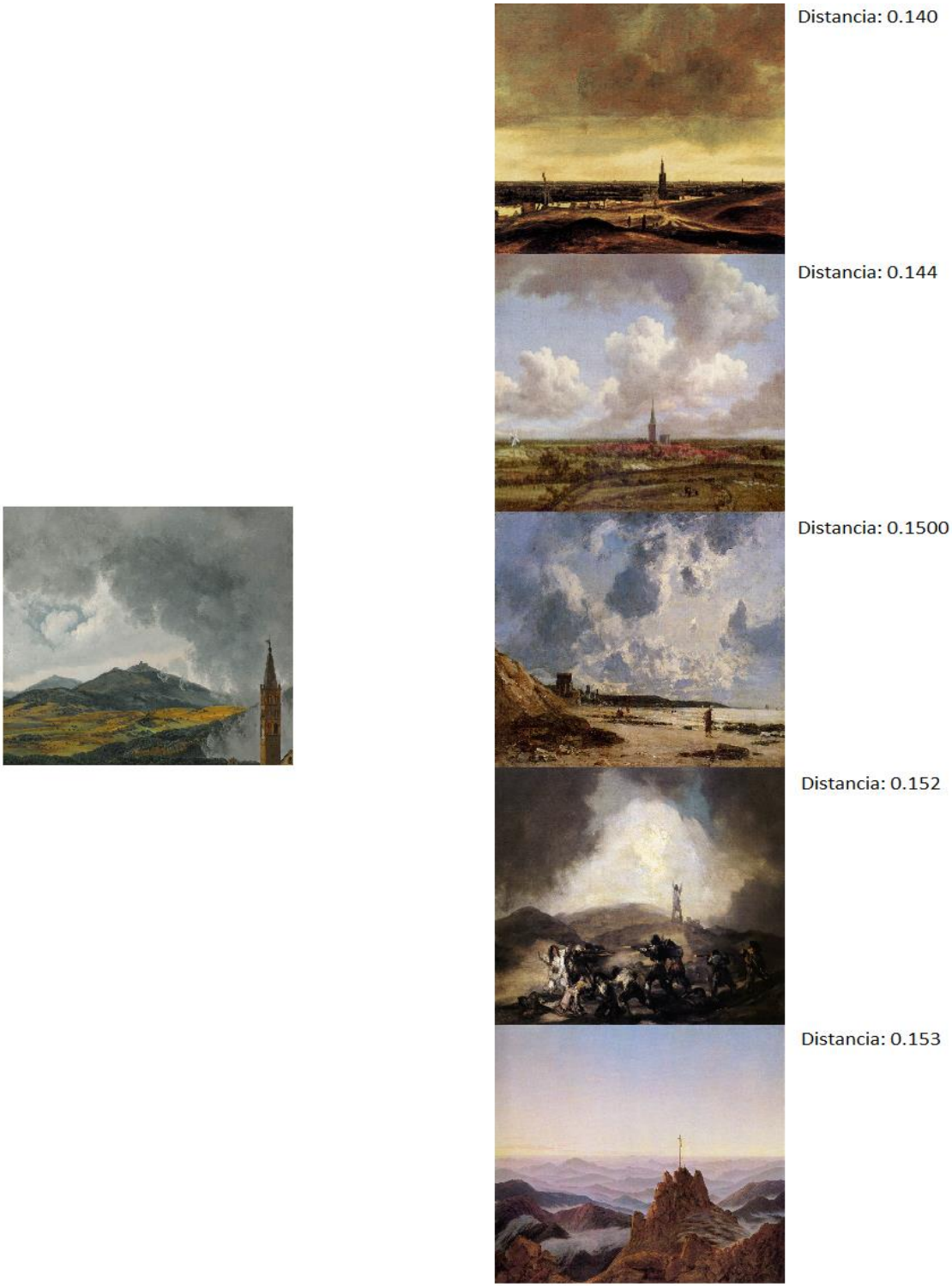


Figura 30: Top 5 de imágenes más cercanas a la imagen de entrada (izquierda). Fuente: Elaboración propia.

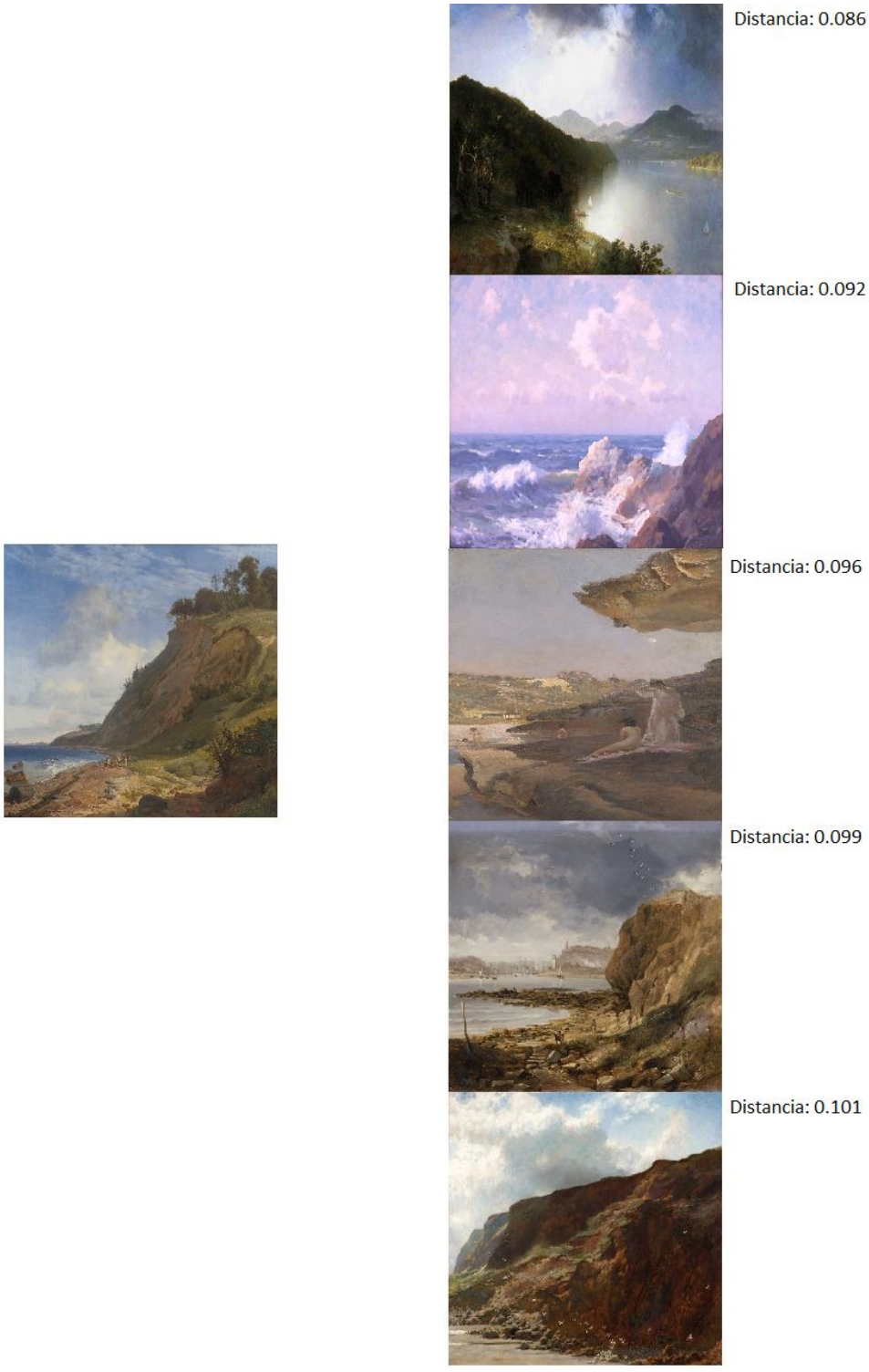


Figura 31: Top 5 de imágenes más cercanas a la imagen de entrada (izquierda). Fuente: Elaboración propia.

Para la figura 28, las cuatro primeras imágenes concuerdan con lo esperado a partir de la imagen de contenido, mientras que la quinta solo se asemeja en el fondo de la imagen. Pese a eso, la distancia entre la imagen de entrada y las cinco primeras es menor a 0.12, lo cual implica lo semánticamente similar de las imágenes.

En el caso de la figura 29, siendo una imagen de entrada con un contenido similar a la usada en la figura anterior, el sistema de recuperación logra encontrar cinco imágenes de estilo nuevas y concordantes con la imagen de contenido. Para este caso, la distancia entre ellas es incluso menor, variando entre valores menores a 0.1.

Por otra parte, en la figura 30, las imágenes de salida rescatan en mayor parte el cielo de la imagen de entrada, pero sin incorporar el paisaje de montañas detrás de ellas. Solo la cuarta y quinta reflejan en mayor medida montañas y el paisaje mostrado en la imagen. Estas diferencias provocan que la distancia entre las imágenes sea mayor a los anteriores casos, fluyendo entre valores de 0.14 y 0.16.

4.3 Transferencia de estilo neuronal

Los experimentos de esta investigación concluyen con la transferencia de estilo entre la imagen de contenido obtenida en la sección 4.1 y la imagen de estilo recuperada en la sección 4.2. En este proceso, utilizando modelos profundos, se generan nuevas imágenes combinando el contenido semántico de una obra con el estilo de otra. Transferir el estilo requiere inicialmente de tres imágenes: una imagen de contenido, una imagen de estilo y una imagen inicial o, de entrada. Esta última inicialmente corresponde a una copia de la imagen de contenido, la cual se irá modificando según el modelo de optimización definido posteriormente.

4.3.1 Preparación de los datos

Inicialmente, la imagen de contenido y estilo reciben un pre-procesamiento con el fin de dejarlas acorde al proceso de entrenamiento de una red *VGG.* Cada imagen se re-escala a un tamaño de 224x224, el mismo utilizado como *input* en las redes VGG-19. Además, son normalizados en una escala BGR bajo la media definida por el arreglo [103.939, 116.779, 123.68].

4.3.2 Representación del contenido y estilo de una imagen

Para obtener las representaciones de contenido y estilo de nuestras imágenes se extraen características de capas intermedias dentro del modelo. El modelo por utilizar es el de una red VGG-19, la cual esta pre-entrenada para clasificación de imágenes utilizando el *dataset* de *ImageNet*. Las capas menos profundas o de menor nivel se caracterizan por aprender características de bajo nivel como líneas o curvas, mientras que a medida que se va profundizando en las capas convolucionales más profundas o de mayor nivel se pueden

representar características de una escala mayor. Esto implica que para representar el contenido de una imagen se deben extraer capas intermedias de la red de mayor nivel.

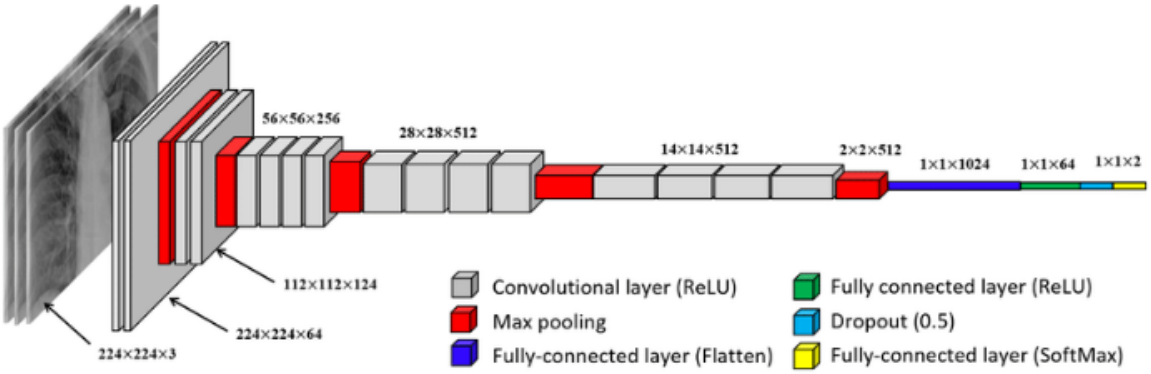


Figura 32: Arquitectura de una red VGG-19, compuesta de 5 bloques de capas convolucionales. Fuente: [Kamil21].

A diferencia de los dos modelos anteriores, en este se extraen las representaciones de capas intermedias dado que los resultados intermedios son los que permiten definir el estilo y contenido de una imagen. Para que una red convolucional como es el caso de una *Resnet50* o *VGG-19* realice clasificación de imágenes o detección de objetos debe comprender la imagen mediante sus píxeles. Esto conlleva tomar una imagen sin procesar (*input*) y generar transformaciones utilizando distintas capas convolucionales o de otro tipo para poder comprender características más complejas de la imagen.

La capa utilizada para la representación de la imagen de contenido corresponde a la segunda capa convolucional del quinto bloque (conv5_2) mientras que para la representación del estilo de la imagen se extraen cinco capas siendo cada una de ellas la primera convolucional de cada bloque: conv1_1, conv2_1, conv3_1, conv4_1, conv5_1. La selección de estas capas se da por la captura del espacio de características diseñada para la información de la textura de la imagen. Al incluir la correlación de características de múltiples capas se obtiene una representación multiescalar de la textura de la imagen.

4.3.3 Función de *loss*

La función de *loss* utilizada se divide en dos: *loss* de contenido y *loss* de estilo.

- ***Loss* de contenido**

La función de *loss* de contenido utilizada se representa por la siguiente ecuación:

$$L_{contenido}(c,x) = \sum_{i,j}(C_{ij}^{l}(c) - X_{ij}^{l}(x))^2$$

Donde "c" es la imagen de contenido, "x" la imagen de entrada, $C_{ij}^{l}(c)$ y $X_{ij}^{l}(x)$ corresponden a la representación de características de la capa "l" de las imágenes de contenido y entrada respectivamente, luego de ser alimentadas en la red VGG-19 pre-entrenada. En el experimento, la representación corresponde a la salida de la capa convolucional conv5_2 de la red.

- ***Loss* de estilo**

La función de *loss* de estilo utilizada posee una complejidad mayor al estar compuesta por la sumatoria de la *loss* de cada una de las cinco capas extraídas como representación del estilo de la imagen multiplicada por el peso de cada una.

$$L_{estilo}(s,x) = \sum_{l=0}^{L} w_l E_l$$

E_l simboliza la *loss* de una de las capas.

$$E_l = \frac{1}{4N_l^2 M_l^2} \sum_{i,j} (S_{ij}^{l}(c) - X_{ij}^{l}(x))^2$$

El peso de cada capa w_l en la experimentación es el mismo, resultando en un valor de 1/5. $S_{ij}^{l}(c)$ y $X_{ij}^{l}(x)$ corresponden a la matriz de Gram de la imagen de estilo y la imagen de entrada. La correlación entre las características está dada por la matriz de Gram $G^l \epsilon R^{N_l x N_l}$ donde G_{ij}^{l} es el producto interno entre dos mapeos de características i y j en la capa l. N_l es el número de mapeo de características y M_l corresponde al producto entre el ancho y alto del vector. Para un vector de (224,224,3) N_l = 3 y M_l = 224*224.

Ambas funciones se combinan en una función de *loss* general:

$$L_{total}(c,s,x) = \alpha * L_{contenido}(s,x) + \beta * L_{estilo}(s,x)$$

Alfa representa el peso del contenido y beta el peso del estilo en la imagen. Ambos parámetros son ajustables, tomando un valor para el experimento de 1e3 y 1-2 respectivamente.

4.3.4 Proceso de optimización

El proceso final corresponde a la optimización de la imagen inicial o de entrada utilizando el método de gradiente descendente con un optimizador Adam para minimizar la función de *loss* total. En este proceso no se ajustan los pesos asociados a la red, si no que se entrena la imagen de entrada para iterativamente minimizar la función de *loss*. La imagen de contenido es comparada con la imagen de entrada a través de la función de *loss* de

contenido mientras que la imagen de estilo se compara con la imagen de entrada a través de su función correspondiente.

4.3.5 Resultados

Utilizando la red, los pesos, variables, funciones de *loss* y el método de optimización señalados anteriormente se obtienen los resultados mostrados desde las figuras 33 a 39. Cada una de las figuras contiene una imagen de salida, la cual es el resultado de 1000 iteraciones del modelo de optimización utilizando gradiente descendente con optimizador Adam para la imagen de contenido y estilo correspondiente.

Las figuras se estructuran de la siguiente manera:

1. Se muestran las imágenes de contenido y de estilo a ser utilizadas.
2. De izquierda a derecha, se muestra la imagen resultante del proceso de optimización cada cien iteraciones.
3. Más abajo, se muestra la imagen de salida luego de mil iteraciones.

En las figuras 33 y 34 se aprecia como la imagen de contenido agarra el estilo del fondo tanto del cielo como del agua de la segunda imagen, generando una con nuevos colores, con una calidad del contenido menor, perdiéndose parte de los detalles de los barcos y de las casas. Ambas imágenes de contenido y de estilo utilizadas son las obtenidas en los experimentos anteriores mostrados en las figuras 25, 28 y 29.

Por otra parte, las figuras 35 y 36 muestran la misma imagen de contenido siendo aplicada por dos estilos diferentes. Para los dos casos, el contenido principal de las imágenes, las montañas, toma la textura de la imagen de estilo de manera correcta. Sin embargo, el cielo pierde calidad al mezclar los colores de ambas imágenes.

Por último, las figuras 37 y 38 muestran los resultados de la transferencia de estilo para las mismas imágenes de contenido y estilo, invirtiendo únicamente en una de ellas los pesos alfa y beta de la función de *loss* de contenido y estilo respectivamente. En la figura 37 se puede apreciar que el contenido de la imagen (la montaña en el lado derecho de la imagen), pierde menos calidad que la de la figura 38, donde prepondera más el estilo de la foto en la salida. Esto concuerda con los pesos utilizados para el estilo y contenido de una imagen, donde uno pondera más que el otro.

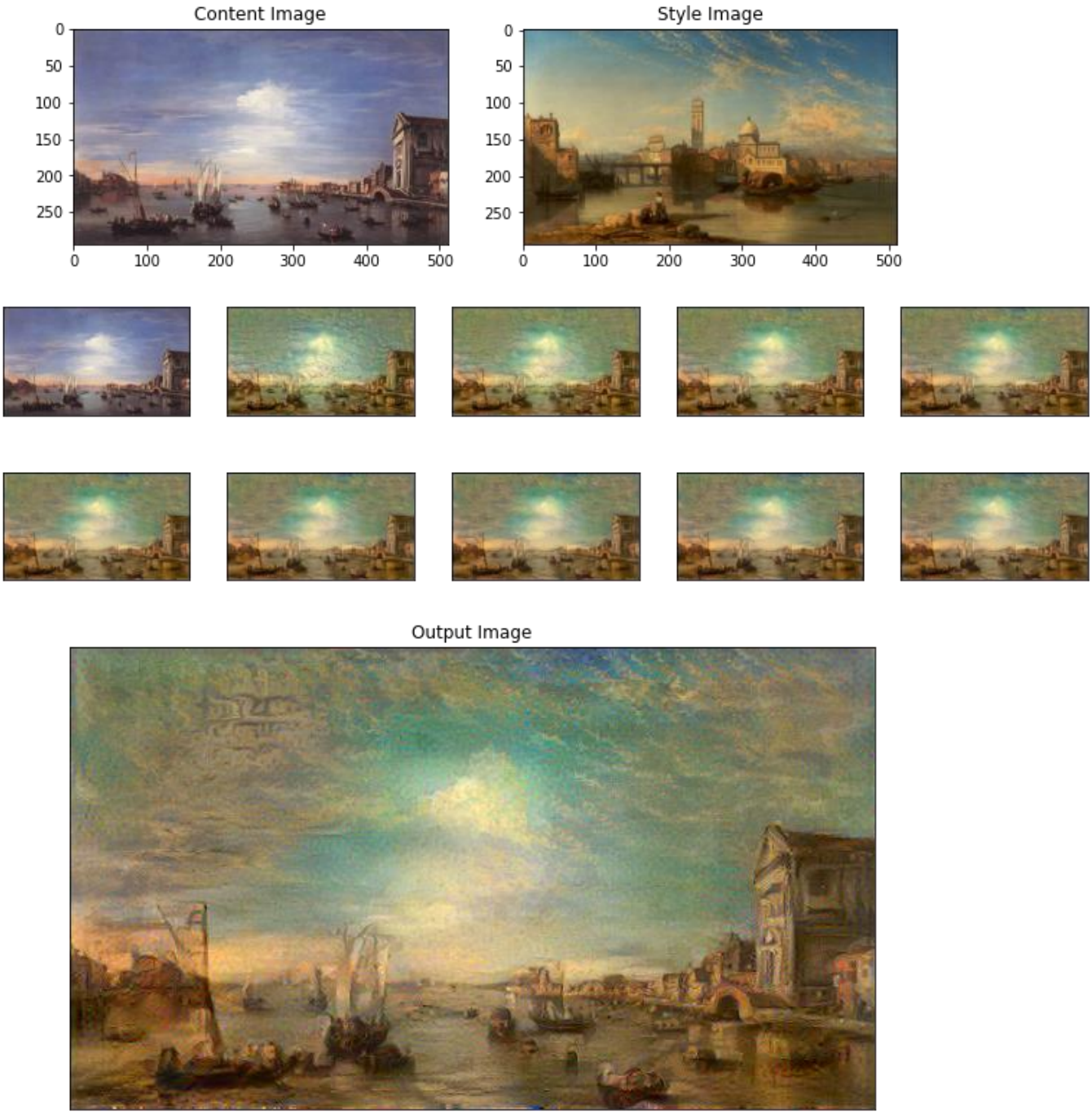


Figura 33: Resultado de la transferencia de estilo de la imagen de contenido y de estilo extraídas de las figuras 25 y 28. Se muestra la imagen de salida cada 100 iteraciones y la imagen final. Fuente: Elaboración propia.

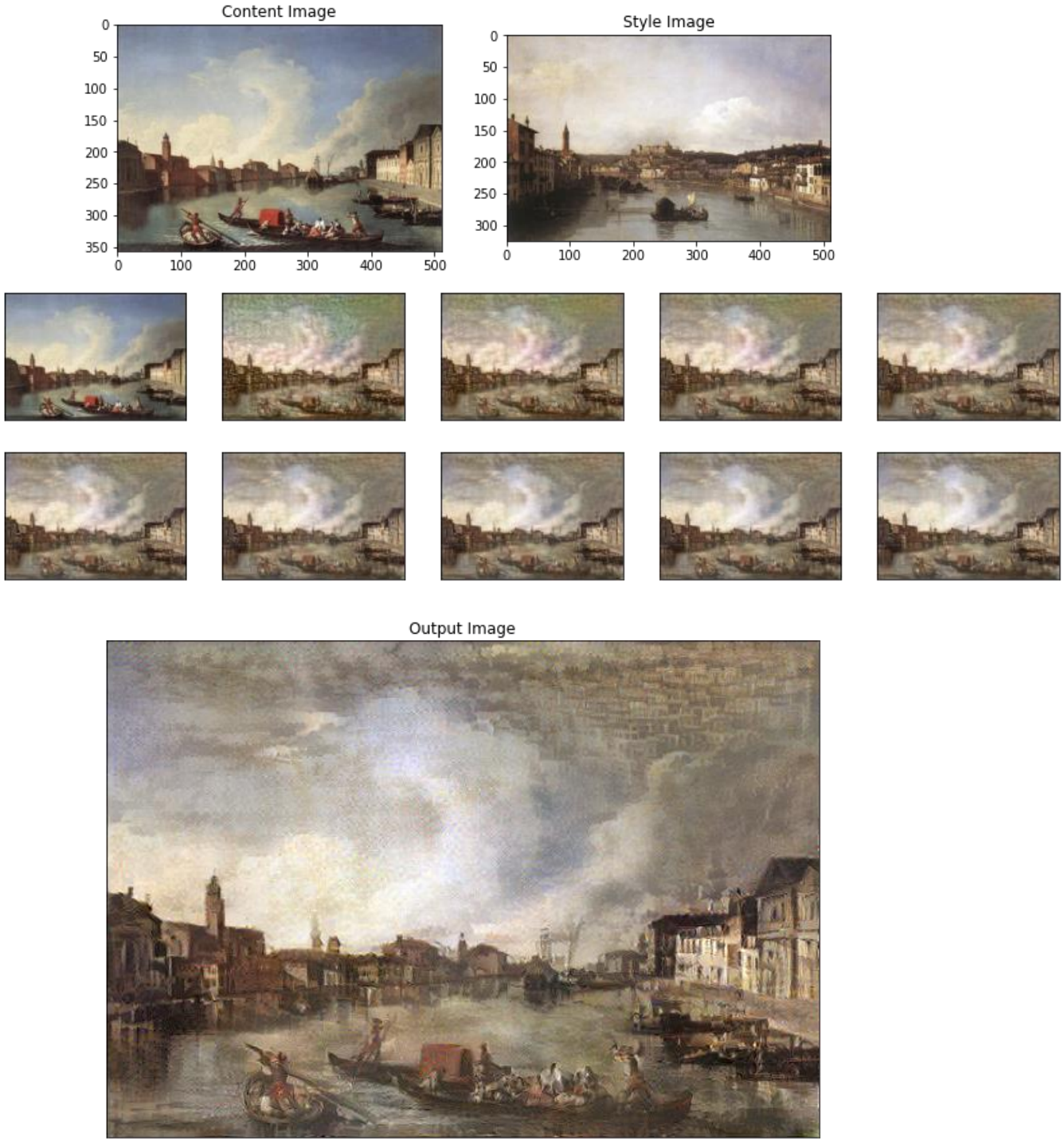


Figura 34: Resultado de la transferencia de estilo de la imagen de contenido y de estilo extraídas de las figuras 25 y 29. Se muestra la imagen de salida cada 100 iteraciones y la imagen final. Fuente: Elaboración propia.

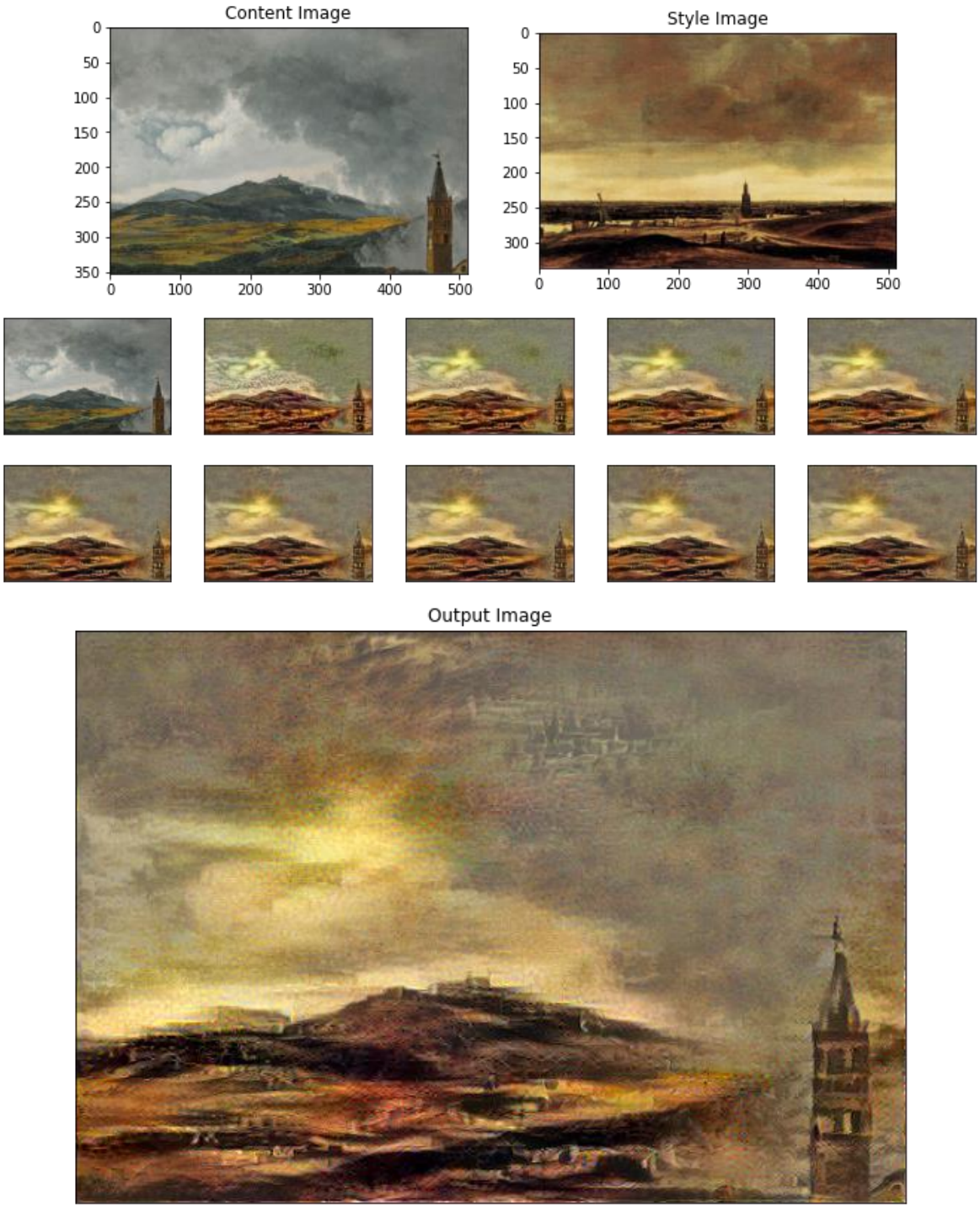


Figura 35: Resultado de la transferencia de estilo de la imagen de contenido y de estilo extraídas de las figuras 26 y 30. Se muestra la imagen de salida cada 100 iteraciones y la imagen final. Fuente: Elaboración propia.

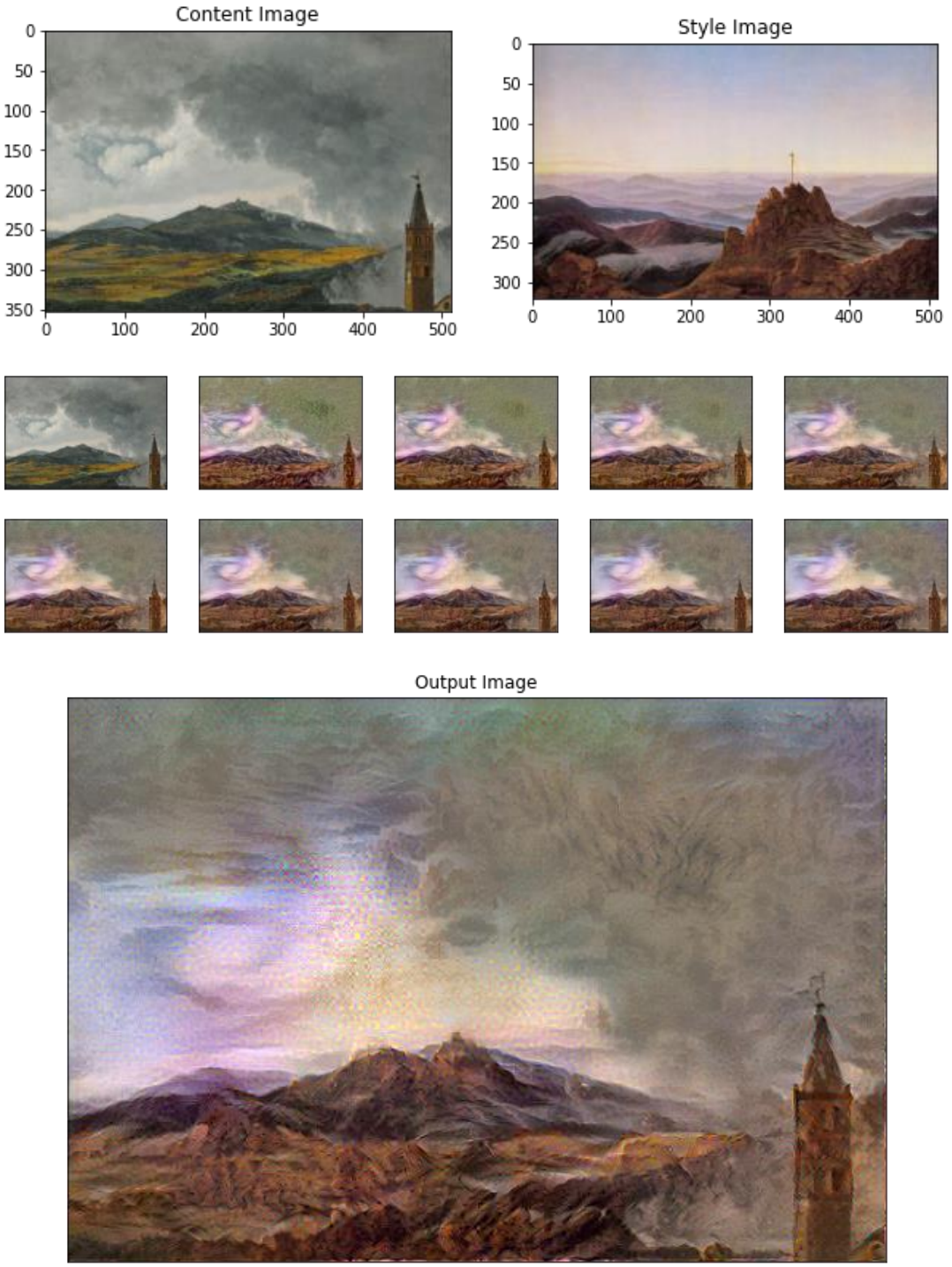


Figura 36: Resultado de la transferencia de estilo de la imagen de contenido y de estilo extraídas de las figuras 26 y 30. Se muestra la imagen de salida cada 100 iteraciones y la imagen final. Fuente: Elaboración propia.

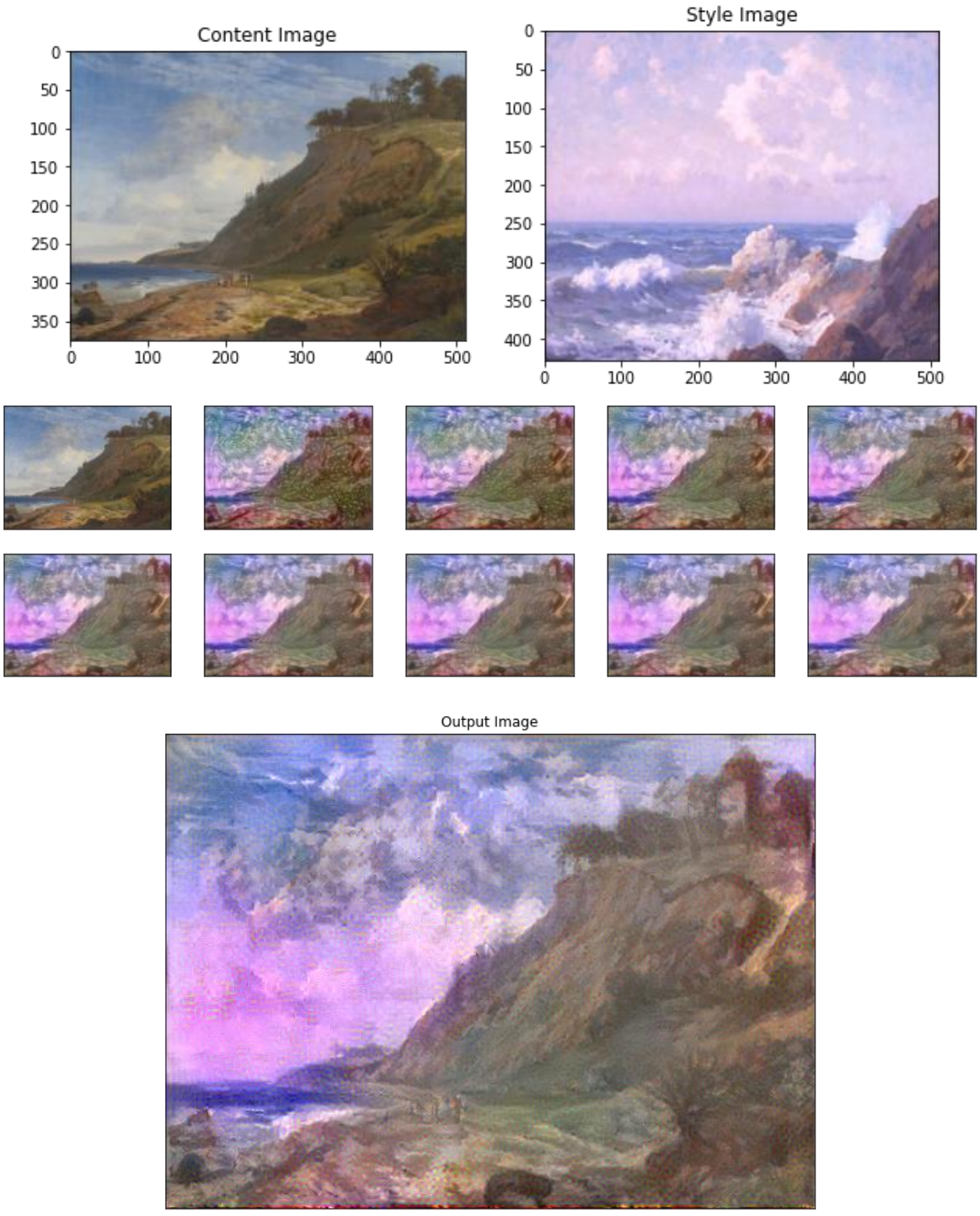


Figura 37: Resultado de la transferencia de estilo de la imagen de contenido y de estilo extraídas de las figuras 26 y 31. Se muestra la imagen de salida cada 100 iteraciones y la imagen final. Fuente: Elaboración propia.

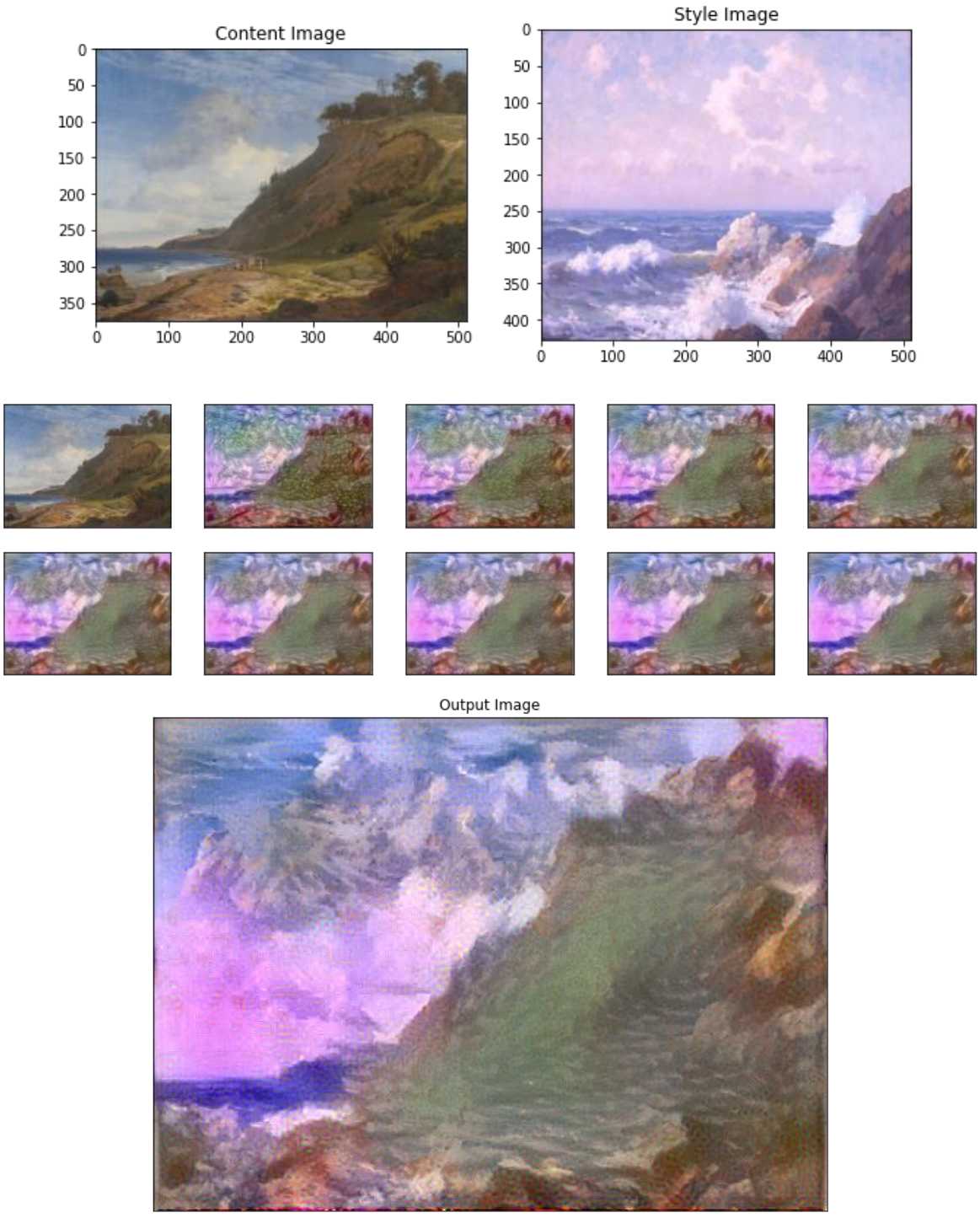


Figura 38: Resultado de la transferencia de estilo de la imagen de contenido y de estilo utilizados en la figura 36 invirtiendo los pesos de estilo y contenido. Se muestra la imagen de salida cada 100 iteraciones y la imagen final. Fuente: Elaboración propia.

Finalizando los experimentos y resultados de la solución propuesta en el documento, se realiza el proceso completo señalado en las tres secciones de este capítulo para un nuevo *input* de entrada. El título y comentario para generar la imagen se muestran a continuación:

Título:

'Flowers lovers colorful'

Descripción:

'Couple of lovers admiring a garden of colorful tulipans'

En la figura 39 se muestra la imagen de contenido entregada por el primer sub-modelo de recuperación de imágenes a través del texto citado anteriormente, acompañada de la imagen de estilo extraída del segundo sub-modelo de recuperación de imágenes. Finalmente, se muestra el resultado del modelo final al aplicar la transferencia de estilo entre ambas imágenes.

En la tabla 3 se resume el rendimiento del modelo en términos de los valores de la función de *loss* para la pérdida de contenido, estilo y total. Inicialmente, dado que la imagen de entrada es la misma que la de contenido, la *loss* de contenido es cero. Luego de mil iteraciones, se logra reducir exponencialmente la función de *loss* total a través del proceso de optimización utilizado.

Tabla 3: Valores de *Loss* de contenido, estilo y total para el experimento de transferencia de estilo del texto de entrada con título '*Flowers lovers colorful'* y descripción *'Couple of lovers admiring a garden of colorful tulipans'*

Iteración	*Loss* de contenido	*Loss* de estilo	*Loss* total
0	0	1.20e08	1.20e08
100	1.11e06	1.68e06	2.79e06
200	7.11e05	6.20e05	1.33e06
300	5.07e05	3.44e05	8.51e05
400	3.97e05	2.46e05	6.43e05
500	3.34e05	2.01e05	5.35e05
600	2.96e05	1.76e05	4.73e05
700	2.72e05	1.61e05	4.34e05
800	2.55e05	1.51e05	4.07e05
900	2.43e05	1.44e05	3.87e05
1000	2.32e05	1.35e05	3.67e05

Figura 39: Resultado de la transferencia de estilo de la imagen de contenido y de estilo recuperadas a partir del título: 'Garden lovers colorful' y descripción: 'Couple of lovers admiring a garden of colorful tulipans'. Fuente: Elaboración propia.

4.3.6 Cambio de parámetros y resultados

En la segunda parte de la experimentación se trabaja con distintos parámetros de alfa y beta en la función de *loss* a minimizar y cambiando el algoritmo de optimización Adam por *L-BFGS (Limit Memory Broyden-Fletcher-Goldfarb-Shanno Algorithm*). Los pesos aplicados de alfa y beta utilizados son de 0.025 y 1, teniendo una proporción inversa respecto a la experimentación anterior. La cantidad de iteraciones se reduce a diez debido al funcionamiento del algoritmo de optimización utilizado, el cual ejecuta cada función de evaluación de la *loss* y el gradiente una mayor cantidad de veces por iteración.

El experimento repite las imágenes de contenido y estilo utilizadas anteriormente con la finalidad de poder comparar ambas imágenes de salida utilizando otros algoritmos de optimización y pesos.

En las dos primeras figuras (40 y 41) se muestra como la imagen de contenido toma el estilo del agua de la segunda imagen, generando una nueva imagen con colores más apagados y con un cielo menos claro. A diferencia de la experimentación anterior, se logra mantener con mejor calidad el contenido de la imagen original.

Para el caso de las figuras 42 y 43 ocurre lo mismo que en las dos figuras anteriores. La imagen de salida logra modificar el color de las montañas, superficie y torre, manteniendo con una mayor calidad el color del cielo al ser comparado con la experimentación anterior. Al igual que en las figuras anteriores, el cambio de estilo es menor comparado a la experimentación anterior.

Por último, las figuras 44 y 45 muestran los resultados de la transferencia para las mismas imágenes de contenido y estilo de la experimentación anterior, también invirtiendo los pesos alfa y beta de la función de loss de contenido y estilo. La figura 44 muestra una mayor pérdida en el contenido de la imagen, desfigurándose el cielo al modificar el estilo. Por otra parte, en la figura 45 se puede ver una mejor calidad de la imagen, manteniendo una mayor calidad de la imagen y modificando el estilo del cielo siguiendo las características de la imagen de estilo.

Al analizar como un conjunto todas las imágenes de la segunda experimentación, se puede ver mayor calidad en cada una de las fotos, sufriendo de menos problemas de desfiguración en las imágenes de salida. Sin embargo, el experimento anterior muestra una mayor modificación en los estilos de cada imagen de contenido, logrando una mayor transferencia del estilo artístico a la imagen.

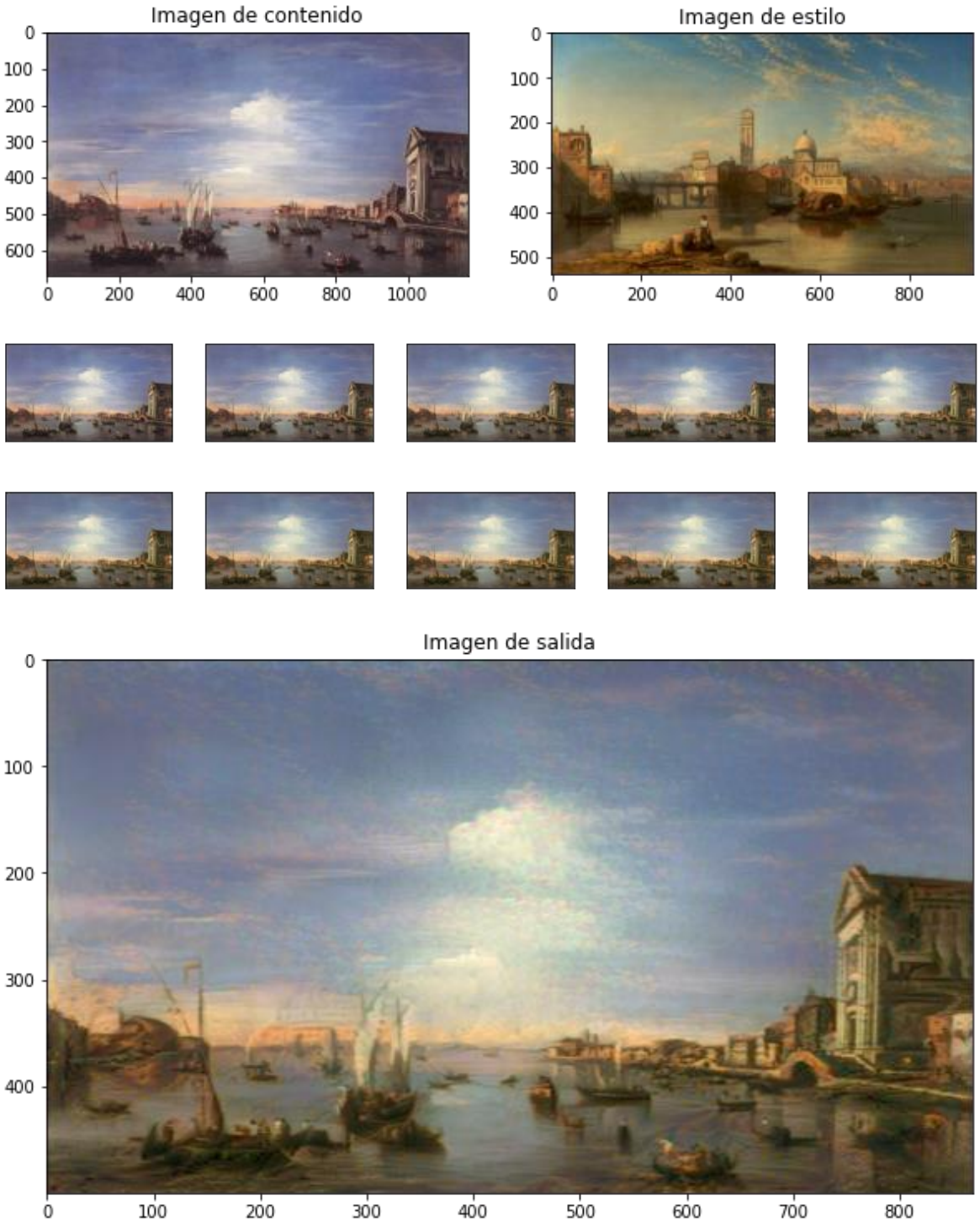


Figura 40: Resultado de la transferencia de estilo de la imagen de contenido y de estilo extraídas de las figuras 25 y 28. Se muestra la imagen de salida de cada iteración y la imagen final. Fuente: Elaboración propia.

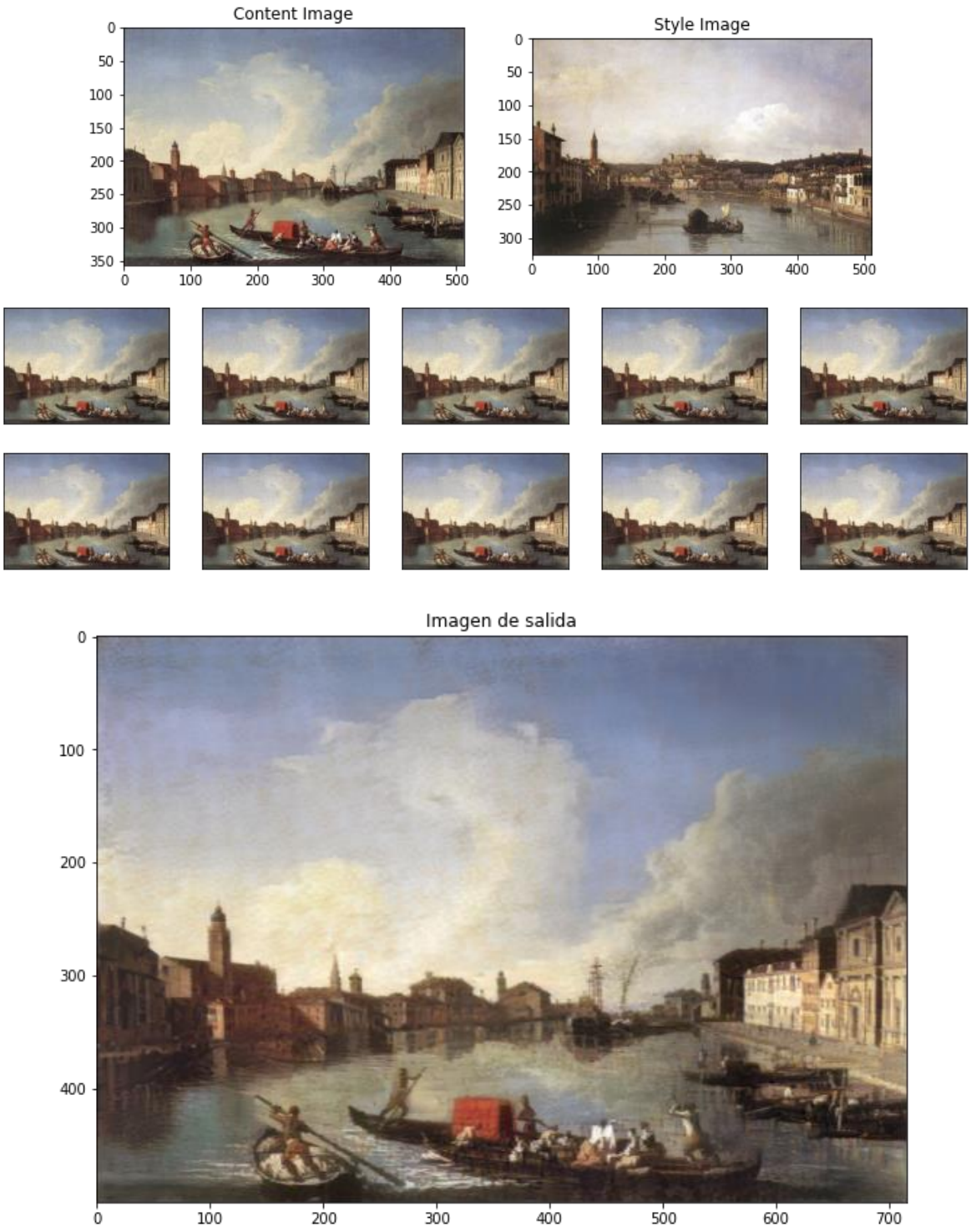


Figura 41: Resultado de la transferencia de estilo de la imagen de contenido y de estilo extraídas de las figuras 25 y 29. Se muestra la imagen de salida de cada iteración y la imagen final. Fuente: Elaboración propia.

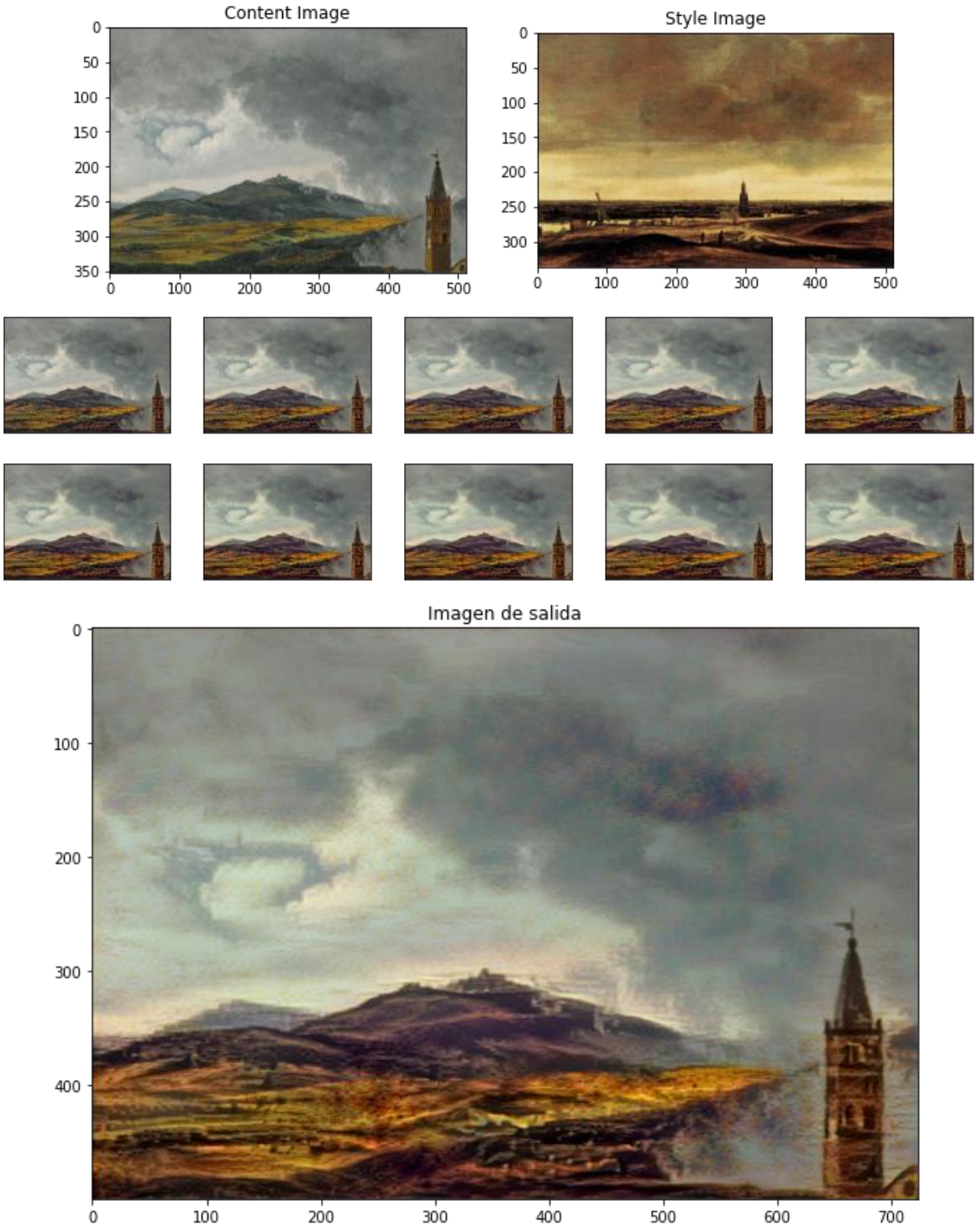


Figura 42: Resultado de la transferencia de estilo de la imagen de contenido y de estilo extraídas de las figuras 26 y 30. Se muestra la imagen de salida de cada iteración y la imagen final. Fuente: Elaboración propia.

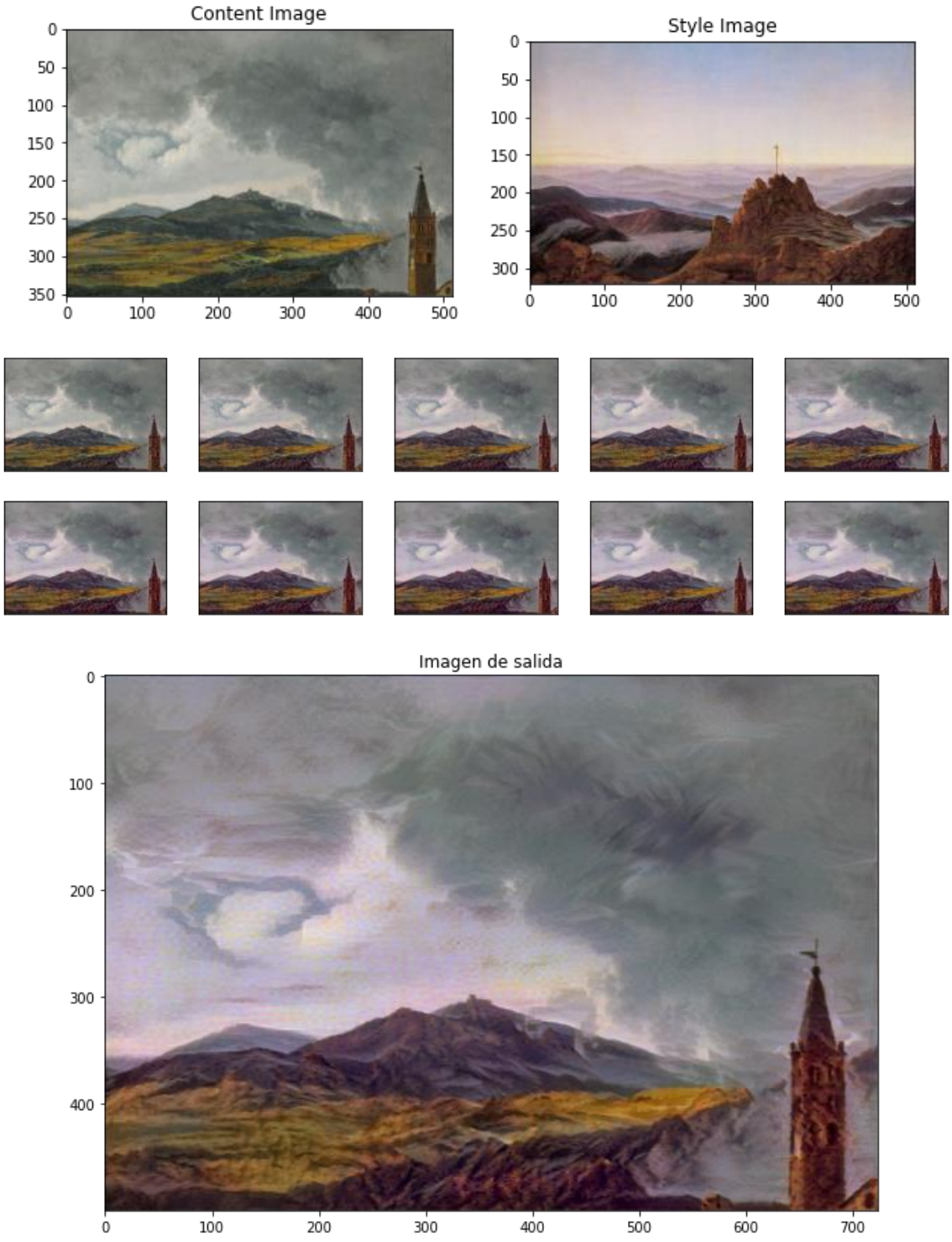


Figura 43: Resultado de la transferencia de estilo de la imagen de contenido y de estilo extraídas de las figuras 26 y 30. Se muestra la imagen de salida de cada iteración y la imagen final. Fuente: Elaboración propia.

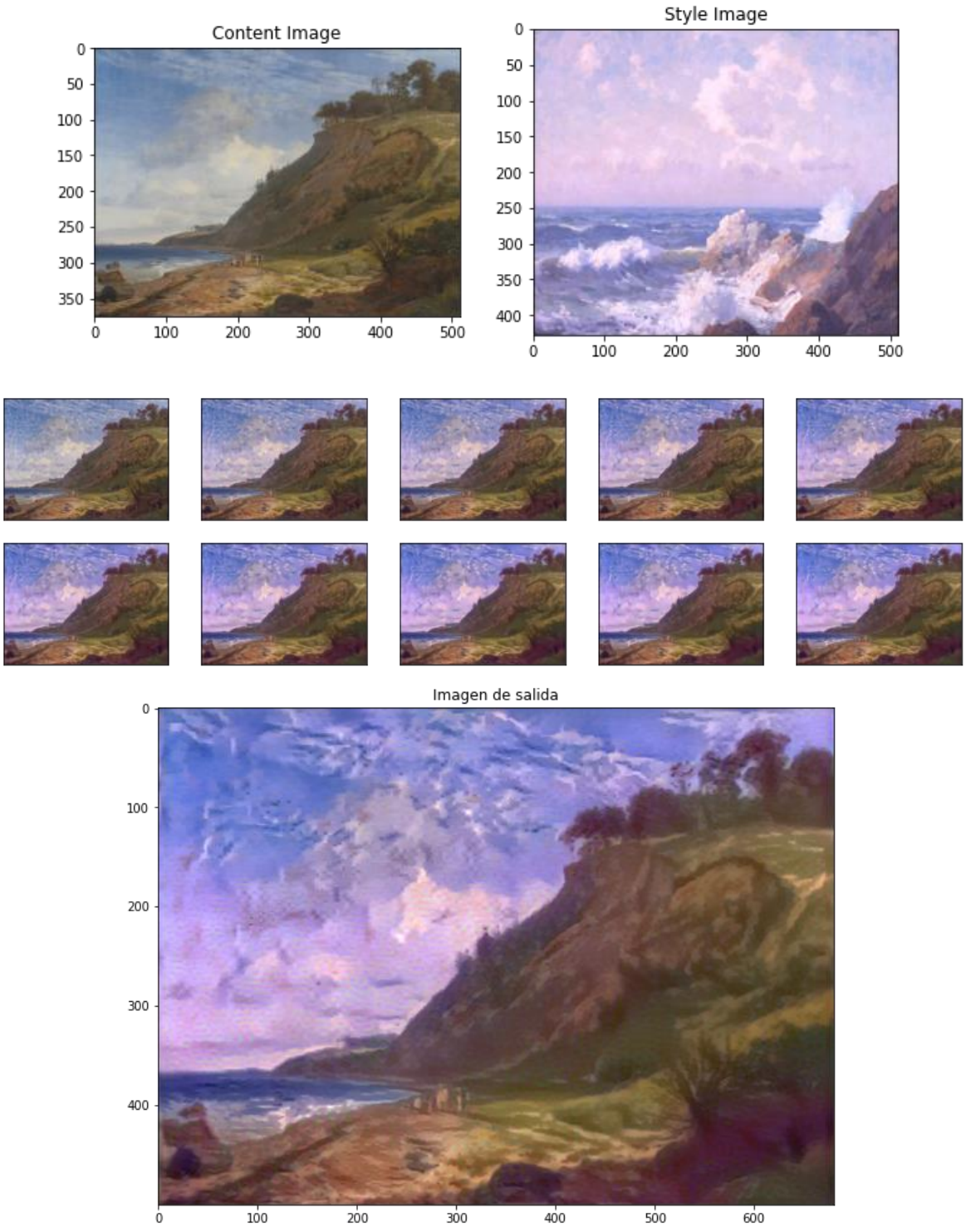


Figura 44: Resultado de la transferencia de estilo de la imagen de contenido y de estilo extraídas de las figuras 26 y 31. Se muestra la imagen de salida de cada iteración y la imagen final. Fuente: Elaboración propia.

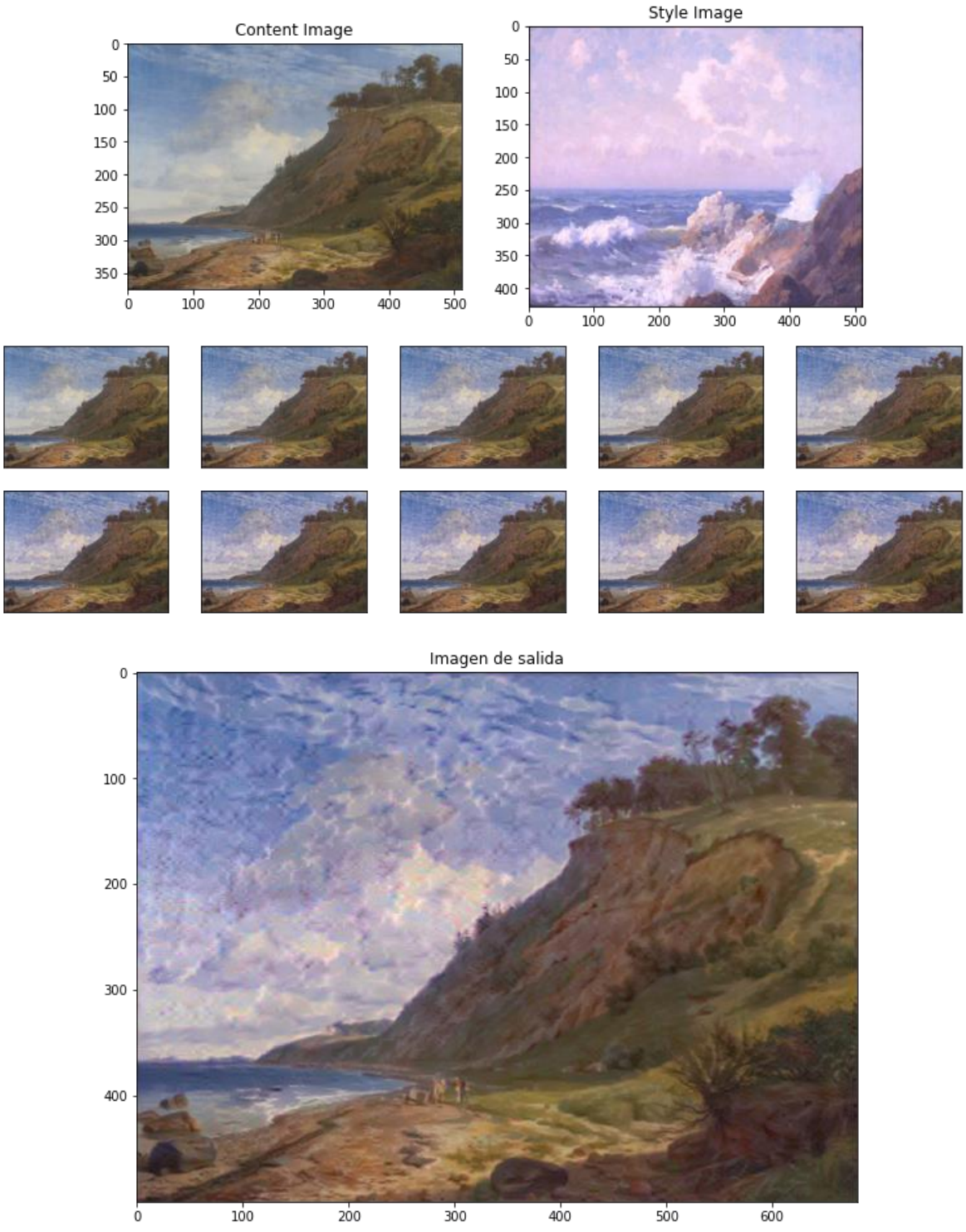


Figura 45: Resultado de la transferencia de estilo de la imagen de contenido y de estilo utilizados en la figura 34 invirtiendo los pesos de estilo y contenido. Se muestra la imagen de salida de cada iteración y la imagen final. Fuente: Elaboración propia.

Título:

'Flowers lovers colorful'

Descripción:

'Couple of lovers admiring a garden of colorful tulipans'

En la figura 46 se muestra el resultado de la transferencia de estilo para el título y descripción señalados arriba. Los resultados, a diferencia de la anterior experimentación, entregan una mejor calidad de imagen, manteniendo detalles del color del fondo, pero cambiando su textura. Además, se puede ver el cambio del color del jarrón y las flores al color de la imagen de estilo. Comparando con la experimentación anterior para la misma imagen, al igual que en los anteriores resultados, la imagen de salida es de mejor calidad por lograr rescatar solo las partes importantes de la imagen de estilo (el color de las flores y jarrón), sin cambiar el fondo de la imagen.

En la tabla 4 se resume el rendimiento del modelo en términos de los valores de la función de *loss* total. Inicialmente, dado que la imagen de entrada es la misma que la de contenido, la *loss* corresponde a la del estilo de la imagen. Luego de diez iteraciones, se logra reducir la función de *loss* total a través del proceso de optimización utilizado.

Tabla 4: Valores de *Loss* total para el segundo experimento de transferencia de estilo del texto de entrada con título '*Flowers lovers colorful'* y descripción *'Couple of lovers admiring a garden of colorful tulipans'*

Iteración	*Loss* total
1	7.36e09
2	3.70e07
3	2.72e06
4	2.22e05
5	1.85e05
6	1.61e05
7	1.42e05
8	1.27e05
9	1.16e05
10	1.08e05

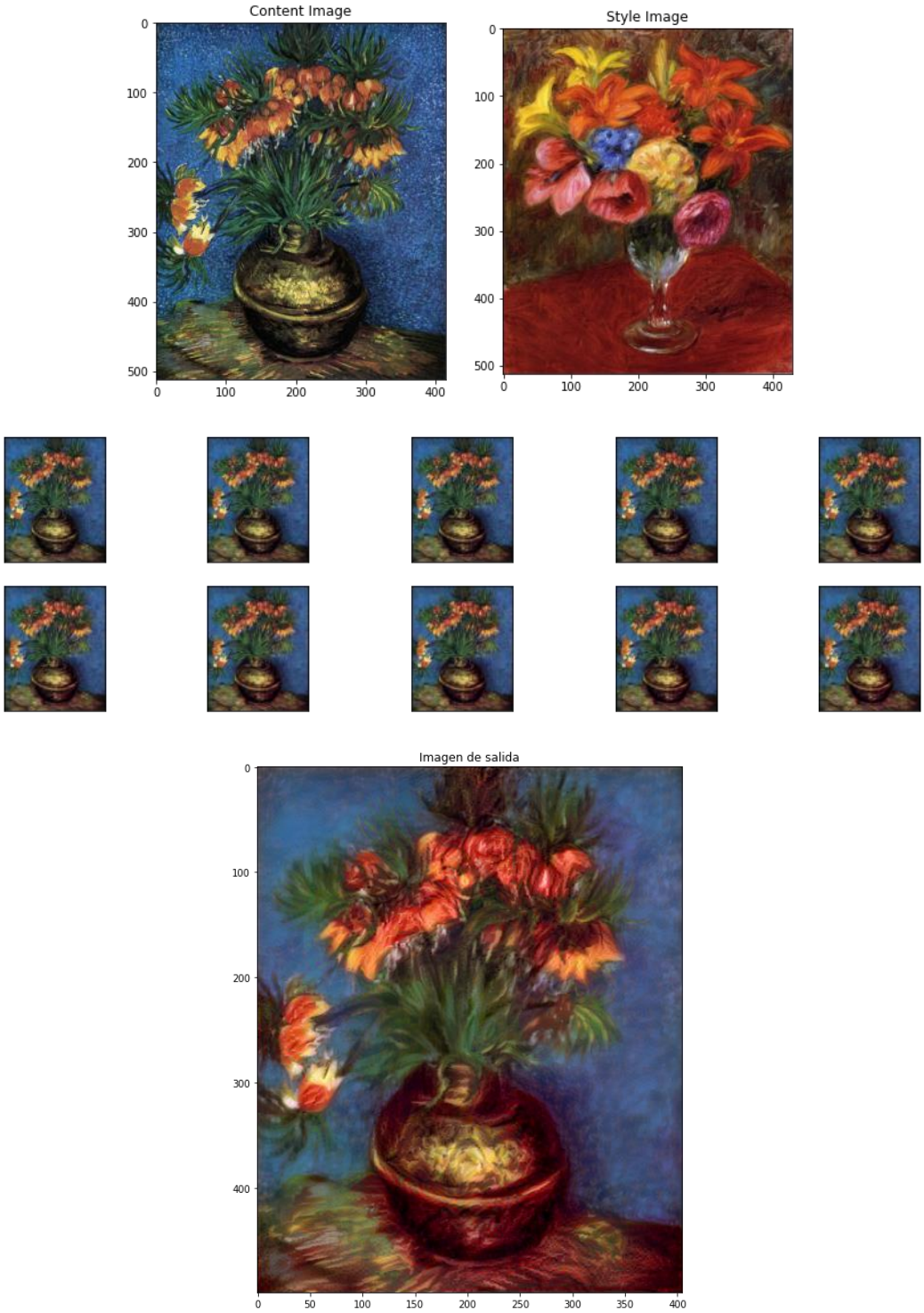


Figura 46: Resultado de la transferencia de estilo de la imagen de contenido y de estilo recuperadas a partir del título: 'Garden lovers colorful' y descripción: 'Couple of lovers admiring a garden of colorful tulipans'. Fuente: Elaboración propia.

CAPÍTULO 5: CONCLUSIONES

En esta sección se concluirá sobre diferentes temas tratados en el documento:

- Conclusiones sobre la calidad de las implementaciones de cada modelo propuesto en la solución.
- Trabajos futuros, documentación y como mejorar los resultados.

5.1 Conclusiones de las implementaciones

Los experimentos realizados para la primera sección de recuperación de imágenes bimodal utilizando texto han demostrado funcionar correctamente para ciertos casos y regular para otros.

Lograr representar un texto y una imagen de manera que puedan ser comparados en su parecido es y fue un desafío importante. Primero, se debe codificar cada imagen y texto del *dataset.* El uso de un modelo pre-entrenado como es el caso de la *Resnet50*, cuyos resultados en otros campos de clasificación de imágenes o detección de objetos es muy preciso, ayudó a tener una representación de las imágenes del *dataset* correcta, sin tener que ajustar los pesos del modelo. Pese a esto, la red se encuentra pre-entrenada con los pesos para el *dataset* de *ImageNet*, el cual contiene más de 14 millones de imágenes, de las cuales solo una pequeña parte se encuadran como imágenes artísticas como es el caso del *dataset SemArt* utilizado. Esto puede conllevar a que la precisión de la representación de las imágenes varie un poco y no sea igual de precisa que para otros casos en que se utiliza la *Resnet50.* Para el caso de los textos se logró representar de manera simple y funcional cada vector de comentarios utilizando un modelo basado en la frecuencia de los términos en el documento para determinar cuál relevante es una palabra para un documento. El vocabulario en inglés para los comentarios, luego de hacer una limpieza de stopwords y términos menos frecuentes está compuesto por alrededor de 10000 palabras para el *dataset* de entrenamiento.

El modelo de transformación multimodal de *Cosine Margin Loss* implementado logra codificar tanto el *encoding* de las imágenes como de los textos en un mismo espacio latente con buenos resultados, logrando que dos representaciones diferentes sean comparadas a través de una función de similitud coseno luego de pasar por una transformación en sus vectores.

A partir de los resultados del entrenamiento se puede concluir que el modelo de transformación permite encontrar los textos correspondientes a cada imagen correctamente dentro de las 10 imágenes más cercanas con alrededor de un 45% de precisión. Pese a ser un valor bajo el 50%, las imágenes encontradas para un texto en su

mayoría coinciden semánticamente en su contenido, siendo posibles ser confundidas por el modelo dado que el texto coincide con la representación de varias imágenes en el *dataset.* El modelo permitió encontrar, dentro de la comparación de veinte mil imágenes con el texto, en una media de posición de 15, la imagen correcta. Además, se debe considerar que el conjunto de entrenamiento está compuesto de alrededor de 20.000 imágenes artísticas donde muchas de ellas son similares en su contenido.

A la hora de evaluar en un conjunto de prueba utilizando textos nuevos, se logran encontrar imágenes semánticamente similares al contenido del texto ingresado por el usuario. Pese a ello, dependiendo del texto ingresado, las imágenes encontradas pueden variar en su contenido por dos factores. El primer factor es que las imágenes incluyen el contenido del texto, pero con un extra de contenido que no fue mencionado en el texto inicial. Esto ocurre por la longitud del texto ingresado, el cual no especifica claramente el contenido principal de la imagen. El segundo factor es que dentro del repositorio de imágenes de prueba no se encuentre ninguna imagen similar a lo ingresado por el usuario. El tamaño del *dataset* de prueba no es tan amplio para poder abarcar todos los contenidos dado que los *dataset* existentes de imágenes artísticas en su gran mayoría no poseen una descripción, reduciéndose a una cantidad menor comparada a otros conjuntos de imágenes utilizadas en otros problemas.

La segunda tarea realizada de recuperación de imagen de estilo fue un proceso más simple que el primero dado que ya se cuenta con la estructura para realizar un *encoding* de las imágenes. A diferencia de la anterior, se utilizó la estructura de una *Resnet152* pre-entrenada con el mismo *dataset* de *ImageNet* para la representación de las imágenes.

A partir de los resultados logrados, se logra encontrar imágenes similares a la imagen de entrada, donde la distancia coseno entre ella y cada una de las cinco más cercanas es menor a 0.15 en la mayor parte de los ejemplos.

A diferencia del *dataset* anterior, en este caso no es necesario contar con un conjunto de imágenes acompañado de texto, por lo que se consiguió trabajar con un *dataset* cuatro veces mayor al anterior, con un total de 80.000 imágenes.

El tamaño del *dataset* permitió resolver grandes partes de los problemas de la primera etapa donde no se encontraba una imagen similar al *input* de entrada dado que el conjunto de imágenes era menor. En este caso, la gran mayoría de imágenes recuperadas concuerdan con la imagen de entrada, siendo solo una parte de ellas no similares. Esto puede deberse a confusiones en la representación de distintos fondos, como es el caso donde se confunde un fondo de agua con nieve. Otro factor también puede ser el tamaño de las imágenes de entrada, el cual permite que no se capture con la mejor calidad la representación de cada

píxel de la imagen original al tener que ser redimensionadas para el proceso de entrenamiento.

Dado esto, podemos concluir que el segundo proceso de recuperación unimodal es más limpio y preciso de realizar que una recuperación de imágenes bimodal a través de texto debido el tamaño del *dataset* de entrenamiento, el cual juega una vital importancia en el desempeño de modelos de redes neuronales.

Por último, la última tarea de este documento consistió en realizar la transferencia de estilo entre dos imágenes, una de contenido y una de estilo, para lograr generar una nueva imagen mezclando ambas características.

Los resultados obtenidos se logran a través de un método de optimización de las representaciones de contenido y estilo de cada imagen, utilizando una red pre-entrenada *VGG-19* y una función de *loss* para ambos tipos de representaciones. A partir de los resultados de las experimentaciones se puede concluir que dependiendo del método de optimización y de los pesos otorgados a cada función de pérdida, el modelo puede entregar dos tipos de resultados:

1. El primer resultado con un foco en representar mejor la transferencia de estilo a la imagen de contenido, perdiendo parte de la calidad visual de los objetos representados y sufriendo algunas desfiguraciones en algunos casos, pero logrando transferir en mayor parte el estilo de imagen.
2. El segundo resultado busca generar las imágenes de mayor calidad posible en su contenido transfiriendo solo parte del estilo de la otra imagen. Esta transferencia se centra en los cambios de colores y texturas para los contenidos más importantes de la imagen (objetos o personas), dejando de manera más similar el resto del fondo de la imagen.

Dado esto, se puede concluir que la tarea de transferencia de estilo permite un "*trade-off*" entre el estilo y contenido de una imagen, dándole oportunidades al usuario de generar imágenes con una mayor transferencia o no dependiendo lo que el usuario requiera.

De modo general, al terminar los experimentos de los tres sub-modelos, se puede concluir que se logró en buenas condiciones implementar un modelo profundo que permite generar nuevas imágenes a partir de un texto ingresado. La calidad de la imagen depende en gran factor de los parámetros utilizados, siendo posible ser ajustados para cada caso con la finalidad de asegurar un buen resultado final. Además, se logró el objetivo de construir una representación bimodal de texto e imágenes en un mismo espacio latente con buenos resultados, el cual era uno de los desafíos más importantes en esta memoria.

5.2 Trabajo futuro y documentación

Como trabajos futuros, el más plausible a desarrollar es generar o recuperar un *dataset* bimodal con una mayor cantidad de imágenes. Esto permitirá resolver los problemas que se tuvieron en la primera parte, donde el contenido recuperado en algunos casos no concordaba con el texto de entrada. Para ello, se debe trabajar utilizando *data mining* para recuperar el comentario y título asociado a cada imagen artística encontrada en sitios como *WikiArt* o *Wikipedia*. Además, utilizando mejores recursos computacionales, se podrá iterar más de 10 veces la fase de entrenamiento del modelo, generando mejores resultados que los obtenidos en la experimentación llegando tal vez a resultados mayores al 50% de precisión para el top 10 de imágenes más cercanas.

Otro estudio importante será la posibilidad de trabajar con textos de entrada en español. Encontrar comentarios artísticos en este idioma en gran escala no es posible en la actualidad, por lo que para trabajar con textos en español se debiese trabajar un módulo de traducción de los textos ingresados. Al ser comentarios artísticos, estos están compuestos de un vocabulario más específico que el cotidiano, por lo que los resultados pueden empeorar a la hora de utilizar una traducción general de las palabras.

En el modelo de transferencia de estilo, sería interesante estudiar nuevas posibilidades de redes a ser utilizadas para la recuperación del estilo y contenido de una imagen. Para ello, se puede trabajar utilizando la arquitectura de redes más profundas como es el caso de una *Resnet50*, la cual posee una mayor cantidad de capas que la utilizada (*VGG-19).* Esto implica obtener una mayor cantidad de capas para recuperar el estilo de una imagen y contar con una capa más profunda para la extracción del contenido.

También, se puede estudiar si modificando las funciones de *loss* dándole un mayor peso a una de las capas de estilo dependiendo de la profundidad en que se encuentre, mejora los resultados del modelo de optimización.

Por último, en un trabajo futuro, es interesante levantar una plataforma web donde el usuario pueda escribir un texto y le genere la transferencia de estilo bimodal utilizando el modelo propuesto. Además, se pueden generar distintas opciones para que el usuario elija cual utilizar. Cada búsqueda puede mostrar cinco imágenes de contenido, donde al seleccionar una se le sugieren las cinco imágenes de estilo para ella. Así, el usuario tendrá una amplia gama de opciones de generar imágenes artísticas. Para un mismo texto se pueden generar más de 50 imágenes diferentes combinando diferentes estilos y contenidos de imágenes.

Toda la documentación de esta memoria se encuentra disponible en [Gutiérrez22], la cual es de uso público con los fines que decida la persona. En este repositorio se adjunta cada uno de los tres sub-modelos. Dos de ellos se encuentran disponibles en un *Google Colab*,

para lo cual no se necesita instalar ninguna dependencia en el computador. El tercer modelo requiere su instalación en el sistema operativo de Windows o Linux, junto a las librerías de *torchvision* y *tensorflow* señaladas.

REFERENCIAS BIBLIOGRÁFICAS

[Alqahtani19] Hamed Alqahtani, Manolya Kavakli-Thorne y Gulshan Kumar. 2019. Applications of Generative Adversarial Networks (GANs): An Updated Review

[Ding18] B. Ding, H. Qian and J. Zhou. 2018. "Activation functions and their characteristics in deep neural networks," *2018 Chinese Control And Decision Conference (CCDC)*, Shenyang,

[Foster19] David Foster. 2019. Generative Deep Learning. Teaching Machines to Paint, Write, Compose and Play.

[Gatys15] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2015. A Neural Algorithm of Artistic Style

[Gupta18] A. Gupta and A. Joshi, "Speech Recognition Using Artificial Neural Network," 2018 International Conference on Communication and Signal Processing (ICCSP), Chennai, 2018, pp. 0068-0071.

[Isola17] Isola P, Zhu JY, Zhou T, Efros AA (2017) Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1125–1134

[James20] James Chen, Michael J Boyle “Neural Network” [online], 2020, Disponible en https://www.investopedia.com/terms/n/neuralnetwork.asp

[JY17] Zhu JY, Park T, Isola P, Efros AA (2017) Unpaired image-toimage translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision, pp 2223–2232

[Kalin18] Josh Kalin. 2018. Generative Adversarial Networks Cookbook

[Liao14] Xiaofeng Liao, Qinshang Jiang, Wei Zhang y Kai Zhang. 2014. BiModal Latent Dirichlet Allocation for Text and Image.

[McCulloch43] W. McCulloch, Warren S y Pitts, “A logical calculus of the ideas immanent in nervous activity,” The bulletin of mathematical biophysics, vol. 5, no. 4, pp. 115–133, 1943

[Rumelhart85] G. E. y. W. R. J. Rumelhart, David E Y Hinton, “Learning internal representations by error propagation,” California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.

[Yin19] Qiwei Yin, Ruixun Zhang, y Xiauli Shao. 2019. CNN and RNN mixed model for image classification

[Chandra20] Chanda Reddy, Convolutional Nueral Networks [online], 2020, Disponible en https://medium.com/@lchandratejareddy/convolutional-neural-networks-6ad55d9bf446

[O'shea15] Keiron O'Shea, Ryan Nash. 2015. An introduction to Convolutional Nueral Networks.

[Jin21] Di Jin, Zhiting Hu, Rada Mihalcea, Zhijing Jin, Olga Vechtomova. 2021. Deep Learning for Text Style Transfer: A Survey

[Shubhankar16] K.Shubhankar Reddy, K.Sreedhar. 2016. Image Retrieval Techniques: A survey

[Bhattacharyya22] Mayukh Bhattacharyya, Intro to image Retrieval with pytorch [online], 2022, Disponible en https://towardsdatascience.com/a-hands-on-introduction-to-image-retrieval-in-deep-learning-with-pytorch-651cd6dba61e

[Noa18] Noa Garcia, George Vogiatzis. 2018. How to Read Paintings: Semantic Art Understanding with Multi-Modal Retrieval

[Po-Chi21] Po-Chih Huang,A content-based image retrieval (CBIR) system [online],2021, Disponible en https://github.com/pochih/CBIR

[Khan20] M. M. R. Khan *et al*. 2020. Automatic Detection of COVID-19 Disease in Chest X-Ray Images using Deep Neural Networks

[Chongke22] Bi, Chongke & Wang, Jiamin & Duan, Yulin & Fu, Baofeng & Kang, Jia-Rong. 2022. MobileNet Based Apple Leaf Diseases Identification. Mobile Networks and Applications.

[Kamil21] Kamil, Mohammed. 2021. A deep learning framework to detect Covid-19 disease via chest X-ray and CT scan images. International Journal of Electrical and Computer Engineering.

[Gutiérrez22] Diego Gutierrez. Transferencia de estilo [online], 2022, Disponible en https://github.com/dguti97/Transferencia-de-estilo