UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

IDENTIFICACIÓN DE ESTILOS ARTÍSTICOS A TRAVÉS DE REDES CONVOLUCIONALES Y ANÁLISIS DE ARQUETIPOS

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN COMPUTACIÓN

OSVALDO NICOLÁS GARAY ROOS

PROFESOR GUÍA:
IVÁN SIPIRAN MENDOZA

MIEMBROS DE LA COMISIÓN:
BÁRBARA POBLETE LABRA
DANIEL CALDERÓN SAAVEDRA

SANTIAGO DE CHILE
2021

Resumen

La motivación de este trabajo surge de la posibilidad de encontrar un nuevo método de categorización de obras artísticas según su estilo artístico. Esto se debe a que, además de los fines prácticos que esto puede tener, tales como el etiquetamiento de obras artistas y el armado de sistemas de recomendaciones de arte, este problema presenta una técnica alternativa para la categorización de entes más abstractos como lo son los estilos artísticos.

El problema a resolver se basa en la implementación de un algoritmo de clasificación de estilos artísticos para obras de arte a través de la combinación de redes neuronales convolucionales con el análisis de arquetipos, algoritmo que luego se modifica en base la precisión obtenida durante diversos experimentos para alcanzar un desempeño superior, seguido de un análisis final de los resultados obtenidos.

La solución implementada consiste en diversos pasos. Se requiere entrenar una red neuronal con una base de datos con la información pertinente al problema, siendo esto una etiqueta de estilo artístico para cada imagen. Una vez entrenada esta red, se usan sus capas para extraer características de las imágenes de una base de entrenamiento, convirtiendo así estas imágenes en vectores de características. Estos vectores de características se agrupan según estilo artístico, donde luego se usa el análisis de arquetipos para obtener un número arbitrario de arquetipos asociados a cada estilo artístico. Finalmente, se itera sobre las imágenes de un conjunto de evaluación, en donde cada imagen es aproximada como una combinación convexa de los arquetipos de cada estilo, siendo la combinación de arquetipos de un mismo estilo con una mayor cercanía a la imagen original aquella que determina el estilo de la imagen a categorizar. De esta forma, se puede evaluar la cantidad de imágenes que fueron categorizadas de forma correcta para contar con la precisión final de la estrategia planteada.

Los resultados más relevantes consisten en la mejora de una configuración inicial con una precisión de un 32 % en una configuración final con una precisión de un 50 %, mientras que la red neuronal usada para calcular los arquetipos de esta configuración final alcanza una precisión de un 57 %. Este resultado se logra gracias a dos pasos particularmente importantes. El primer paso es la utilización de una capa específica de la red neuronal para la caracterización de las imágenes, a diferencia de la sugerencia hecha por trabajos anteriores que utilizan múltiples capas a lo largo de toda la red neuronal, lo que se traduce en una mejora de un 34 % a un 42 % de precisión. El segundo paso es el aumento de arquetipos de forma artificial, debido a que restricciones de memoria no permiten un uso mayor a una cantidad fija de arquetipos previo a este aumento artificial, lo que permite subir la precisión de un 42 % a un 50 %.

Tabla de Contenido

1. Introducción **1**

2. Marco teórico y estado del arte **4**

2.1. Marco teórico 4

2.1.1. Redes neuronales convolucionales 4

2.1.2. Análisis de arquetipos 5

2.1.3. Singular value decomposition 7

2.1.4. Algoritmo de Frank-Wolfe 7

2.1.5. Transfer learning 8

2.1.6. ImageNet 8

2.1.7. WikiArt 8

2.2. Estado del arte 9

2.2.1. Unsupervised Learning of Artistic Styles with Archetypal Style Analysis 9

2.2.2. Large-scale Classification of Fine-Art Paintings: Learning The Right Metric on The Right Feature 10

2.2.3. Recognizing Art Style Automatically in painting with deep learning . 10

2.2.4. Deep Transfer Learning for Art Classification Problems 11

2.2.5. Recognizing the Style of Visual Arts via Adaptive Cross-layer Correlation 13

2.2.6. Cross-Depiction Transfer Learning for Art Classification 13

2.2.7. Artist Identification with Convolutional Neural Networks 14

3. Problema **15**

3.1. Descripción 15
3.2. Relevancia 15
3.3. Requisitos de la solución 17
3.4. Caracterización de los resultados 17
3.5. Resultado esperable 18

4. Solución **19**

4.1. Implementación 19
4.1.1. Armar base de datos 19
4.1.2. Entrenar red neuronal convolucional 20
4.1.3. Convertir las imágenes en vectores de características 21
4.1.4. Reducir la dimensionalidad de los vectores de características 22
4.1.5. Calcular los arquetipos 22
4.1.6. Reconstrucción de imágenes a partir de arquetipos 23
4.1.7. Determinar categoría y calcular precisión 23

5. Resultados **24**

5.1. Experimentos exploratorios 24
5.1.1. Configuración estándar 24
5.1.2. Argumento “tmax” 26
5.1.3. Número de arquetipos 27
5.1.4. Tasa de aprendizaje 28
5.1.5. Reducción de dimensionalidad 28
5.1.6. Entrenamiento desde cero 29
5.1.7. Número de épocas 31
5.1.8. Entrenamiento de una a cinco épocas 32
5.1.9. Capas a utilizar 32
5.1.10. Número de capas a reentrenar 33

5.1.11. Reentrenamiento de las capas del último bloque 33
5.1.12. Ponderación por capas 34
5.1.13. Aumento de arquetipos 36
5.2. Experimento final 37
5.2.1. Configuración final 37
5.2.2. Entrenamiento con la base de datos completa 38
5.2.3. Matriz de confusión 38
5.2.4. Precisión y pérdida 42
5.2.5. Tipos de clasificación 44
5.3. Resumen de los resultados 45

6. Conclusión **47**

Bibliografía **51**

Índice de Tablas

2.1. Desglose base de datos 11

2.2. Resultados Rijksmuseum 12

2.3. Resultados Antwerp 12

5.1. Argumento “tmax” 27

5.2. Número de arquetipos 27

5.3. Tasa de aprendizaje 28

5.4. Reducción de dimensionalidad 29

5.5. Entrenamiento desde cero 29

5.6. Épocas 32

5.7. Desempeño durante las primeras cinco épocas. 32

5.8. Capas a utilizar 33

5.9. Cantidad de capas a reentrenar 33

5.10. Desempeño de entrenar las 5 últimas capas convolucionales por 20 épocas. . 34

5.11. Desempeño de distintas ponderaciones a una red que reentrena sus últimas cuatro capas convoluciones por 5 épocas. 35

5.12. Desempeño de distintas redes al usar distintas capas individuales para el análisis de arquetipos. 35

5.13. Desempeño de capas individuales de la red C al hacer el análisis de arquetipos. 36

5.14. Desempeño de utilizar distintas cantidades de arquetipos a partir de los 1000 arquetipos emulados. 37

5.15. Desempeño de la configuración final. 38

5.16. Comparación entre métricas de distancia. 44

Índice de Ilustraciones

2.1. Representación visual de la arquitectura VGG19, siendo todas las capas max pooling de 2x2, siendo todas las capas convolucionales de 3x3, con el número entre paréntesis representando la cantidad de veces que se repite la capa, el número sobre las flechas representando el tamaño de la matriz de entrada/salida, y N representando el número de clases a clasificar. 5

2.2. Representación gráfica de tres arquetipos para 20 puntos en un espacio de dos dimensiones. 6

3.1. Cuatro ejemplos de obras obtenidas de WikiArt.
obras mostradas son, de izquierda a derecha y luego de arriba a abajo: "Lamentation" de Giotto (*Proto Renaissance*), "The Martyrdom of St. Catherine" de Jan Cossiers (*Baroque*), "The Oxbow" de Thomas Cole (*Romanticism*), y "Two Centimeter Wavy Bands in Colors" de Sol LeWitt (*Post-Painterly Abstraction*). 16

5.1. Precisión de la red entrenada utilizando la configuración estándar. 25

5.2. Pérdida de la red entrenada utilizando la configuración estándar. 26

5.3. Precisión de la red entrenada desde cero utilizando una tasa de aprendizaje de 1×10^{-4}. 30

5.4. Pérdida de la red entrenada desde cero utilizando una tasa de aprendizaje de 1×10^{-4}. 31

5.5. Matriz de confusión de la red neuronal convolucional, con la clase actual a lo largo del eje vertical y la clase predicha a lo largo del eje horizontal. 40

5.6. Matriz de confusión del análisis de arquetipos, con la clase actual a lo largo del eje vertical y la clase predicha a lo largo del eje horizontal. 41

5.7. Matriz de confusión del análisis de arquetipos restando la matriz de confusión de la red neuronal convolucional. 42

5.8. Precisión de la configuración final de la red entrenada con la base de datos completa. 43

5.9. Pérdida de la configuración final de la red entrenada con la base de datos completa. 44

Capítulo 1

Introducción

Hoy en día existen diversos algoritmos de clasificación para música, videos y obras artísticas, cada uno con distintas estrategias de clasificación con sus respectivas ventajas y desventajas. Si bien ya existen algoritmos que hacen clasificación de estilos artísticos, en este trabajo se propone un algoritmo distinto a las alternativas ya conocidas, basado en una combinación de redes neuronales convolucionales con el análisis de arquetipos, con la esperanza de que esta propuesta obtenga resultados superiores a los de clasificación usando solo redes neuronales convolucionales. Se mencionan las redes neuronales convolucionales en particular ya que este trabajo se arma en base a estas mismas redes para hacer la clasificación.

A grandes rasgos, la propuesta actual busca convertir imágenes en vectores de características, utilizando la salida de capas especificas de una red neuronal convolucional como la base para obtener estos vectores de características, donde luego son utilizados para crear un conjunto de arquetipos para cada estilo a clasificar, y finalmente usar una combinación de estos mismos arquetipos para aproximarlos a la imagen a clasificar, retornando el estilo del conjunto de arquetipos que mejor aproxima a la imagen como el estilo de la imagen a clasificar.

Este trabajo de clasificación es relevante por diversas razones. Desde un punto de vista teórico, el aprendizaje de máquina en el mundo del arte resulta ser interesante debido a la diversidad de información asociada a las obras artísticas, ya que el estilo artístico de estas obras no se puede clasificar usando solo el contenido concreto de la imagen (entendiéndose como ejemplos de esto un tenedor, un perro o una casa), a diferencia de como suelen funcionar los algoritmos de clasificación de imágenes, si no que dependen también de características más abstractas que no necesariamente tienen un análogo en el mundo físico. Algunos ejemplos de estas características incluyen técnicas de trazado, uso de la perspectiva, paletas de colores (que no necesariamente se corresponden a la paleta de colores del objeto representado en el mundo real), y figuras abstractas, entre otras características más sutiles y complejas de describir. Desde un punto de vista práctico, se puede mencionar el etiquetamiento de forma eficiente para obras artísticas recién digitalizadas, así también como posibles aplicaciones en sistemas de recomendación de obras artísticas, e incluso implementar la propuesta actual en campos similares, tales como el diseño gráfico, el arte contemporáneo y las comisiones artísticas, que pueden contar con un número de obras considerablemente mayor a las utilizadas en este

trabajo.

El objetivo general de este proyecto es desarrollar, evaluar y analizar el desempeño del método de clasificación de estilos artísticos a través de la técnica basada en la combinación de redes neuronales convolucionales con análisis de arquetipos, utilizando la precisión como medida principal de evaluación de la propuesta, ya que es esta métrica la que permite evaluar el desempeño del algoritmo al momento de hacer la clasificación. Este trabajo hace además una comparación entre la técnica propuesta y una red neuronal convolucional típica para así poder analizar posibles ventajas y desventajas de cada método.

Entre los objetivos específicos, se pueden mencionar los siguientes:

- Estudiar el estado del arte de propuestas similares.
- Obtener la base de datos apropiada y reestructurarla para poder llevar a cabo la propuesta.
- Implementar la red neuronal convolucional.
- Desarrollar el algoritmo para convertir imágenes en sus vectores de características.
- Desarrollar el análisis de arquetipos para llevar a cabo la clasificación.
- Ejecutar el algoritmo usando una variada cantidad de posibles configuraciones.
- Analizar los resultados obtenidos de las distintas configuraciones.
- Refinar el algoritmo según los resultados obtenidos.
- Evaluar el desempeño final del algoritmo.
- Analizar y comparar el desempeño final del algoritmo con otras propuestas.

La idea de combinar redes neuronales convolucionales con análisis de arquetipos para clasificar obras de arte surge del trabajo de Wynen et al. [26] del año 2018, en donde se utilizan estas tecnologías para transferir al estilo de una imagen al contenido de otra imagen, razón por la cual se espera que estos arquetipos puedan ser útiles para la clasificación de estilos artísticos también. Usando como base el procedimiento utilizado en el trabajo anterior, se propone entonces una solución que cuenta con los siguientes requisitos. Primero, es necesario estudiar el estado del arte de propuestas similares para recolectar ideas y averiguar sobre configuraciones recomendadas. También es necesario contar con una base de datos apropiada, tanto por el contenido de esta como que tenga los metadatos pertinentes, que luego se debe reestructurar de tal forma que pueda ser utilizada por la red neuronal. La red a utilizar es una red neuronal convolucional llamada VGG19, y se usa tanto su variante predeterminada como una variante preentrenada a través de ImageNet. También es necesario contar con un algoritmo para convertir las imágenes de entrada en un vector de características, que en este caso buscan representar su estilo artístico. Este vector puede llegar a ser muy largo, razón por la cual podría ser necesario contar con un algoritmo de reducción de matrices. También es necesario implementar el algoritmo de elaboración de arquetipos.

Una vez cumplidos estos requisitos, se puede empezar la implementación del algoritmo. Esto consiste en reentrenar la red neuronal con la base de datos a clasificar. Una vez hecho

esto, se implementa un algoritmo que recibe las imágenes de entrenamiento de entrada, convirtiendo cada imagen en un vector de características armado usando las salidas de las capas de la red anterior. Estos vectores se apilan para formar una matriz, que luego se guarda en el disco duro para poder ser utilizado por otro algoritmo. El mismo paso anterior se aplica a las imágenes de evaluación. El siguiente algoritmo convierte las matrices anteriores en versiones reducidas. Para esto se toma la matriz de entrenamiento antes mencionada y se usa para entrenar el reductor de dimensionalidad de scikit-learn llamado TruncatedSVD. Luego, se usa este reductor para reducir tanto la matriz de entrenamiento como la matriz de evaluación a matrices de dimensionalidad significativamente menor. Estas matrices también son guardadas en el disco duro. Finalmente, se ejecuta el algoritmo de análisis de arquetipos. Para esto se usa la matriz de entrenamiento reducida para armar los arquetipos, lo que se logra segmentando esta matriz según estilo artístico. Luego se le aplica a todas las imágenes de entrenamiento del mismo estilo el algoritmo de análisis de arquetipos, lo que retorna una nueva matriz donde cada fila es un arquetipo del estilo a estudiar. Es importante mencionar que el número de arquetipos a utilizar por estilo es arbitrario, por lo que puede ser menor al número de imágenes que se cuentan por estilo. Luego, se itera entre las imágenes de la matriz de evaluación, donde una por una son pasadas por el algoritmo de combinación convexa de arquetipos, que es similar al algoritmo usado para crear los arquetipos anteriormente. Se hace entonces la combinación convexa de arquetipos que mejor aproxima al vector de características de la imagen original, y se categoriza a esa imagen como el estilo del conjunto de arquetipos que mejor aproxima esa imagen. Una vez aplicado esto a toda la matriz es posible medir el porcentaje de precisión final.

Con los resultados obtenidos es posible hacer comparaciones entre distintas configuraciones, lo que permite refinar el algoritmo hasta contar con los mejores algoritmos según su desempeño obtenido. Con estos últimos algoritmos es posible evaluar el desempeño general de la combinación de redes neuronales convolucionales con análisis de arquetipos, para finalmente hacer un análisis completo del método propuesto, incluyendo las posibles ventajas y desventajas de usar este método respecto al uso de una red neuronal convolucional típica.

Los resultados más relevantes consisten en que la configuración inicial del análisis de arquetipos alcanza una precisión de un 32 %, mientras que la configuración final alcanza una precisión de un 50 %. Esto se logró a través de diversas configuraciones y experimentos, siendo dos técnicas particulares aquellas que logran mejoras sustanciales. La primera técnica consiste en usar una sola de las capas finales de la red para convertir las imágenes en sus respectivos vectores de características, ya que las últimas capas resultan tener una mejor representación de estilo al usar el análisis de arquetipos, logrando así una mejora de un 34 % a un 42 % de precisión. La segunda técnica consiste en aumentar el número de arquetipos de forma artificial para hacer la categorización de las imágenes, logrando así una mejora de un 42 % a un 50 % de precisión.

Capítulo 2

Marco teórico y estado del arte

2.1. Marco teórico

A continuación se describen ciertas tecnologías que son relevantes para el trabajo actual, siendo las más importantes el método de análisis de arquetipos y las redes neuronales convolucionales.

2.1.1. Redes neuronales convolucionales

Las redes neuronales convolucionales, conocidas también como *convolutional neural network* (CNN) en inglés, son un tipo de red neuronal profunda usado frecuentemente en el área de visión computacional. Estas redes se basan en múltiples capas de filtros convolucionales que transforman la información de las imágenes en cada capa, en donde las neuronas de una capa están todas conectadas a las neuronas de la siguiente capa. Esta arquitectura está inspirada en procesos biológicos tales como la forma en la que se organiza la corteza visual de los animales.

Este trabajo utiliza una red neuronal convolucional específica, llamada VGG19, cuya configuración predeterminada recibe un input de $224 \times 224 \times 3$, es decir, una imagen RGB representada como una matriz de 224 de ancho y 224 de alto para cada uno de los tres canales de color RGB. Más específicamente, esta red utiliza capas convolucionales de 3×3 con activación ReLU, capas de *max pooling* de 2×2, capas *fully connceted* y una capa *softmax*. ReLU es una función de activación que retorna el argumento de la función si el argumento es positivo, y retorna 0 si el argumento no es positivo. *Max pooling* retorna el valor máximo para cada grupo de neuronas del tamaño indicado. Las capas *fully connected* son aquellas que conectan todas las neuronas de una capa a todas las neuronas de otra capa. La capa *softmax* generaliza una función logística a un vector con tal de retornar otro vector cuya suma de sus valores sean igual a 1, conservando así el orden de los valores del vector anterior, así como también un sentido de magnitud entre valores arbitrariamente grandes a valores acotados entre 0 y 1. Las capas se usan en el siguiente orden: dos capas convolucionales de profundidad 64, una capa de *max pooling*, dos capas convolucionales de profundidad 128,

una capa de *max pooling*, cuatro capas convolucionales de profundidad 256, una capa de *max pooling*, cuatro capas convolucionales de profundidad 512, una capa de *max pooling*, cuatro capas convolucionales de profundidad 512, dos capas *fully connected* de 4096 neuronas con activación ReLU, y finalmente, una capa *softmax*, también *fully connected*. Esta red es la misma red utilizada por Wynen et al. [26], trabajo en el cual está inspirado este mismo trabajo, razón por la cual se decide usar esa red en particular. Una representación visual de esta red se puede observar en la figura 2.1.

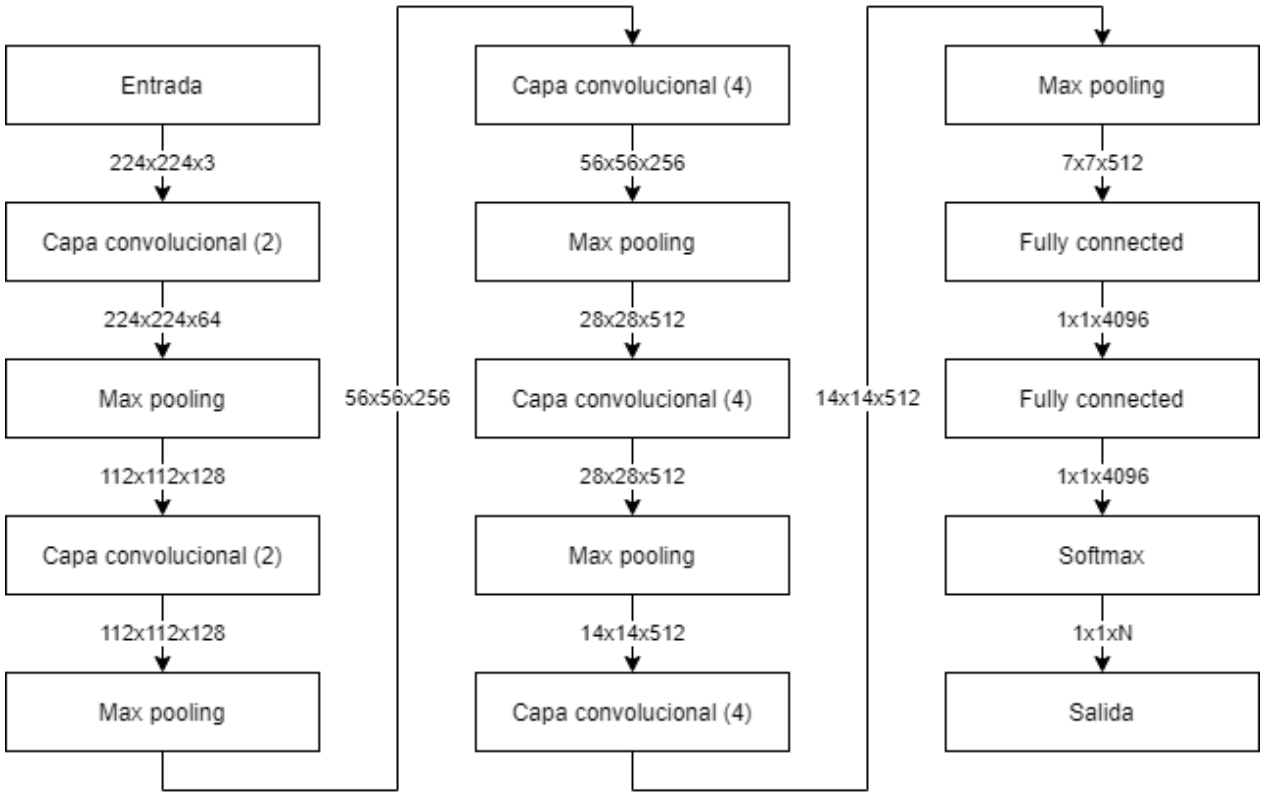

Figura 2.1: Representación visual de la arquitectura VGG19, siendo todas las capas max pooling de 2x2, siendo todas las capas convolucionales de 3x3, con el número entre paréntesis representando la cantidad de veces que se repite la capa, el número sobre las flechas representando el tamaño de la matriz de entrada/salida, y N representando el número de clases a clasificar.

2.1.2. Análisis de arquetipos

Según el trabajo original de Cutler y Breiman sobre análisis de arquetipos [5], el análisis de arquetipos es un método que busca representar un elemento como una combinación de elementos especiales del conjunto de datos, llamados arquetipos, siendo a su vez los arquetipos una combinación de elementos de alguna base de datos específica. En particular, estos arquetipos se arman a partir de reducir el error proveniente de reconstruir cada imagen de la base de datos con el número fijo de arquetipos especificados.

La intuición detrás del análisis de arquetipos es la idea de representar una colección

de datos a través de un conjunto reducido de ejemplares, los cuales contienen suficiente información para aproximar el conjunto de datos original.

De forma gráfica, los arquetipos se pueden representar como una cantidad de puntos arbitrarios de un espacio sobre el cual el área conformado por estos puntos mejor aproxima al área que conforman todos los datos a analizar. Se puede ver en la figura 2.2 que los puntos del espacio de dos dimensiones a analizar están representados con puntos rojos (en este caso 20), mientras que los arquetipos (en este caso tres) son los puntos que conforman las esquinas del triángulo azul, y se puede interpretar como el área que mejor cubre la totalidad de puntos del espacio. La propuesta entonces es escoger un número específico de arquetipos para cada estilo artístico, con el número de arquetipos a utilizar sujeto a experimentación, para luego poder determinar la cantidad adecuada de estos.

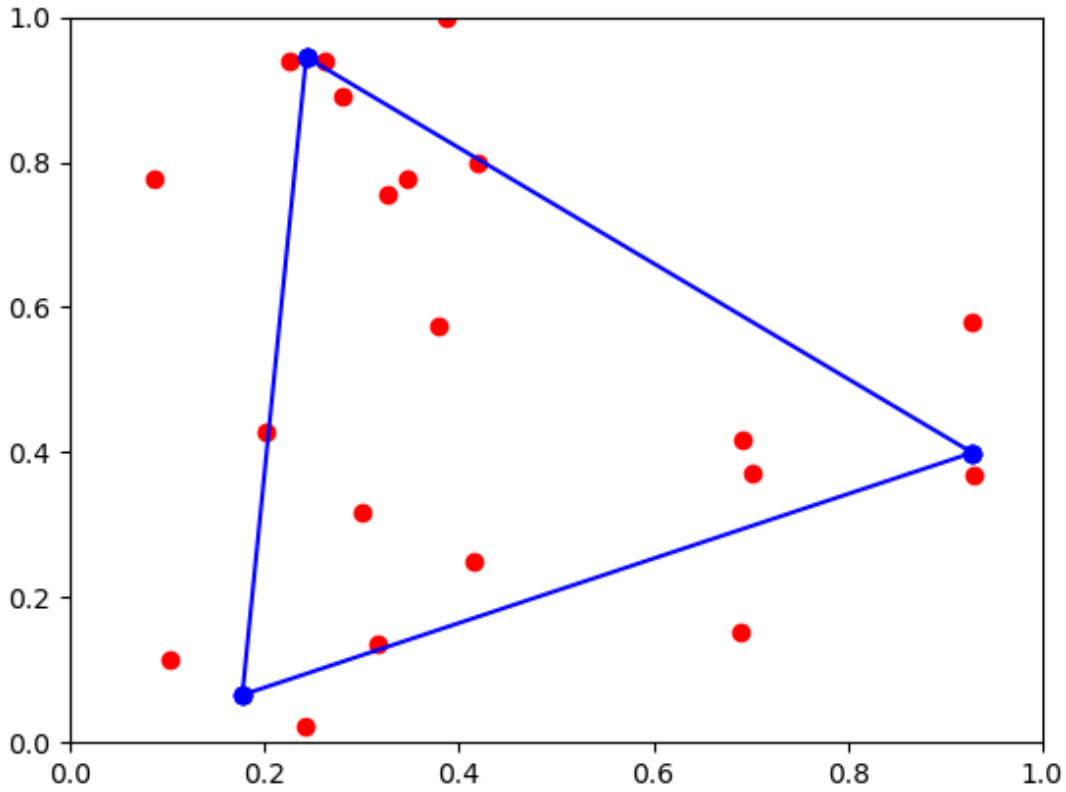

Figura 2.2: Representación gráfica de tres arquetipos para 20 puntos en un espacio de dos dimensiones.

Expresando el método con mayor rigurosidad, para hacer el análisis de arquetipos se requiere de un conjunto de datos. Este conjunto de datos es una pila de vectores de características $x_1, ..., x_n$, donde cada x_i es un vector con m características de los datos que se buscan estudiar (en este caso, los datos a estudiar son las imágenes, y los vectores característicos son aquellos que retorna la función de caracterización para cada imagen), y n representa el total

de muestras (en este caso, cada muestra representa una imagen). Luego, se declara un número arbitrario p de arquetipos a utilizar (en este caso, se usa el mismo número de arquetipos para cada estilo artístico). Una vez escogido el número p de arquetipos a utilizar, se procede a buscar los valores de α_{ik} tales que minimizan la expresión $\sum_i \|x_{\mathrm{i}} - \sum_{k=1}^{p} \alpha_{\mathrm{ik}} z_{\mathrm{k}}\|^2$, sujeto a que la suma de los α_{ik} sea 1 y cada uno de los α_{ik} sea mayor o igual 0, y con $z_{\mathrm{k}} = \sum_j \beta_{\mathrm{kj}} x_{\mathrm{j}}$, siendo $k = 1, ..., p$, y con β_{kj} sujeto a las mismas restricciones que α_{ik}. Una vez resuelto este problema de optimización, se guardan los vectores z_{k}, ya que estos son los arquetipos que se buscaba calcular. En otras palabras, la definición de z_{k}, que son los arquetipos a utilizar, restringe a los arquetipos a ser una combinación lineal de los datos a estudiar, mientras que la expresión a minimizar representa la combinación de arquetipos que más se acercan a los datos a través de una combinación convexa de estos mismos arquetipos. Lo anterior implica que la entrada para la función de creación de arquetipos son los vectores característicos x_{i} de un estilo artístico específico, y lo que retorna esta función son los vectores característicos z_{k} de los arquetipos para ese estilo artístico en particular. Es importante notar que el proceso necesario para poder retornar estos arquetipos implica resolver un problema de optimización no trivial para que los arquetipos se acerquen a los datos que se busca ejemplificar, lo cual se puede resolver a través de métodos numéricos ya establecidos en el área de *sparse coding* y *non-negative matrix factorization*, como los presentados por Mairal et al. en Sparse Modeling for Image and Vision Processing del año 2014 [12].

2.1.3. Singular value decomposition

Una *singular value decomposition* (SVD) es una factorización de una matriz A en un producto de matrices $U\Sigma V^H$, siendo U una matriz unitaria, Σ una matriz diagonal, y V^H siendo otra matriz unitaria [21]. La importancia de esta técnica radica en la posibilidad de calcular una aproximación a la matriz A como un producto de matrices $U\Sigma V^H$ cuyos rangos son significativamente menores al de la matriz A original. La reducción es tan eficiente que, en Wynen et al. [26], logran reducir un vector de características de 600 000 valores en uno de 4096 valores conservando el 99 % de su varianza. En este trabajo se usa una implementación de un SVD reducido de scikit-learn llamado Truncated SVD [17] para hacer la reducción de dimensionalidad.

Una característica importante de esta técnica es que, para la reducción de una matriz específica, es luego posible aplicar esta misma reducción a otra matriz, lo que se usa al momento de convertir un vector de características particular en un vector reducido y luego así ser comparable con la matriz original.

Un detalle importante a tener en consideración es que, para hacer la reducción de dimensionalidad, se requiere que el número de dimensiones a reducir sea menor al largo de la dimensión más corta de la matriz.

2.1.4. Algoritmo de Frank-Wolfe

Como los arquetipos del análisis de arquetipos son una combinación convexa de datos, y luego estos datos se aproximan como una combinación convexa de arquetipos, se requiere

de un algoritmo de optimización convexa para resolver este problema de optimización. Un algoritmo de optimización convexa es el algoritmo de Frank-Wolfe, que se utiliza en una implementación de Bauckhage [1] para calcular arquetipos, razón por la cual esta implementación es utilizada en este trabajo. Esta implementación funciona al iterar el algoritmo de Frank-Wolfe suficientes veces para el cálculo del vector Z que mejor aproxima la matriz $X \approx AZ$, luego el vector Y que mejor aproxima la matriz $X \approx XYZ$, y finalmente, se calcula la matriz A mediante la fórmula $A = XY$, donde X representa a los puntos a estudiar, A representa a los arquetipos, e Y y Z representan a los vectores que mejor logran las aproximaciones ya descritas. Lo anterior se justifica por el hecho de que el análisis de arquetipos aproxima una matriz X a un producto XYZ, donde A es igual al producto XY, y por lo tanto la expresión $X \approx XYZ$ se puede expresar también como $X \approx AZ$.

2.1.5. Transfer learning

Transfer learning, también conocido como *inductive transfer*, se refiere al problema de retener y aplicar el conocimiento aprendido en una tarea para resolver de forma eficiente otras tareas similares [18]. A lo largo de este trabajo, se hace referencia al uso de esta técnica a través del uso del término "reentrenamiento". Tanto para este trabajo como otros trabajos mencionados en este capítulo resulta común usar una red neuronal entrenada con ImageNet para mejorar el desempeño de esta red en la clasificación de características artísticas, debido a que luego se usa una base de obras artísticas para continuar el entrenamiento en la red anterior, y así llevar a cabo el *transfer learning*. Entre los trabajos de categorización de obras artísticas que usan esta técnica se pueden mencionar Lecoutre et al. [10] del año 2017, Sabatelli et al. [15] del año 2018, Sur y Blaine [22] del año 2017 y Viswanathan [24], también del año 2017.

2.1.6. ImageNet

ImageNet es una base de imágenes para una gran cantidad de objetos distintos, contando a la fecha con más de 14 millones de imágenes para más de 20 000 objetos distintos [7]. En trabajos de clasificación que utilizan redes neuronales es común usar redes preentrenadas con esta base de datos para implementar técnicas *transfer learning*, como se puede señalar en los trabajos de Lecoutre et al. [10], Sabatelli et al. [15], Chen y Yang [2], Sur y Blaine [22], y Viswanathan [24]. Un ejemplo de red que permite una configuración ya preentrenada con ImageNet para 1000 objetos distintos es la implementación de la arquitectura VGG19 por parte de la librería Keras [23].

2.1.7. WikiArt

WikiArt es una página web dedicada a las artes visuales que contiene unas 250 000 obras entre 3000 artistas distintos, estando las obras etiquetadas según artista, año, estilo, género, entre otros [25]. Resulta interesante mencionar que esta página cuenta con más de 200 estilos distintos, 35 de los cuales cuentan con al menos 1000 obras cada uno. Al ser una página web

de acceso público y con una enorme cantidad de obras a su disposición, su uso en el área del aprendizaje de máquina ha sido considerable, lo que se puede evidenciar por su uso en trabajos como Saleh y Elgammal [16], Lecoutre et al. [10], y Viswanathan [24].

2.2. Estado del arte

Los problemas de clasificación de arte han sido abordados por diversas propuestas a la fecha, diferenciándose tanto por el contenido que buscan clasificar (estilo, artista, periodo, medio, etc) como por la estrategia a utilizar para la clasificación (redes neuronales convolucionales, *transfer learning*, *adaptive cross-layer correlation*, etc). También se han abordado propuestas de representación de estilo y extracción de características que pueden llegar a ser de utilidad para problemas de clasificación de obras artísticas. A continuación se describe el estado del arte relevante a la propuesta de este trabajo.

2.2.1. Unsupervised Learning of Artistic Styles with Archetypal Style Analysis

Un trabajo particularmente relevante para esta propuesta es Unsupervised Learning of Artistic Styles with Archetypal Style Analysis de Wynen et al. [26] del año 2018. Esto se debe al hecho de que, si bien este trabajo no busca hacer una clasificación de imágenes, si busca transferir el estilo artístico de una imagen en otra, y para esto utiliza una combinación de representación de estilo artístico y análisis de arquetipos, siendo esta última técnica la relevante para definir los estilos a utilizar. Se cree entonces que estas técnicas pueden ser de utilidad para hacer una clasificación según estilo artístico.

Más específicamente, el trabajo desarrolla un algoritmo que retorna una matriz de características de una serie de capas específicas de una red neuronal convolucional para cada imagen a tratar, siendo la red utilizada una red VGG19 preentrenada con ImageNet. Luego define el estilo de la imagen como la concatenación del vector de los promedios y las desviaciones estándar asociadas a las matrices anteriores, para luego hacerle una reducción de dimensionalidad que permite al algoritmo aplicar el análisis de arquetipos. Esta reducción de dimensionalidad permite reducir el largo del vector de estilo a un largo de 4096 valores, y luego utiliza un número de arquetipos entre 32 y 256 para representar un estilo artístico. Una vez hecho esto, sigue el proceso de manipulación de estilo artístico. Para esto se requieren dos imágenes, una que determinará el contenido y otra que determinará el estilo de la imagen final. También se requiere de un codificador que retorna la matriz de características de una imagen al ingresarla por las capas especificadas de una red neuronal, y un decodificador que ha sido entrenado para revertir la matriz de características del codificador anterior, de tal forma que aplicarle el decodificador a la imagen codificada retorne una aproximación a la imagen original. Usando esta estrategia se puede luego tomar matrices de características asociadas tanto al contenido como al estilo de una imagen para luego crear una imagen usando el contenido de una obra y el estilo de otra usando la estrategia de decodificar el codificador, pero esta vez con contenido y estilos distintos.

2.2.2. Large-scale Classification of Fine-Art Paintings: Learning The Right Metric on The Right Feature

En un trabajo realizado en el año 2015 por Saleh y Elgammal [16] se busca clasificar obras artísticas según artista, estilo y género usando una gran variedad de técnicas para ello. Más específicamente, se usa una combinación de métricas y extracción de características, usando un tipo de extracción de característica para cada tipo de métrica utilizada. Las métricas utilizadas son Boost Metric Learning, Information Theoretic Metric Learning (ITML), Large Margin Nearest Neighbors (LMNN), Metric Learning for Kernel Regression (LMKR), Neighborhood Component Analysis (NCA), y una métrica de punto de referencia que consiste en entrenar un clasificador *one-vs-all* sobre el vector de características. Las características utilizadas son GIST, Classemes, Picodes, y características de CNN. Además de lo anterior, también se usan las técnicas Metric Learning, Feature Fusion y Metric Fusion para hacer la clasificación. Feature Fusion hace combinaciones de las cuatro características antes mencionadas, Metric Fusion hace una combinación de las cinco métricas antes mencionadas, y Metric Learning es la técnica donde se entrena una métrica según un tipo de característica en particular.

Los experimentos se evaluaron usando una base de datos de WikiArt, contantando esta con 81 449 pinturas entre 1119 artistas, 27 estilos y 45 géneros distintos. Para la clasificación de estilo y género se utiliza un mínimo de 1500 obras por cada una, mientras que para la clasificación de artista se utiliza al menos 500 obras por cada uno. Al experimentar con las técnicas anteriores, los resultados indican que las mejor combinación para Metric Learning es Boost con Classemes, tanto para clasificación de estilo con un 31.77 %, para clasificación de género con un 57.87 %, y para clasificación de artista con un 57.76 %. Cuando se trata de Metric Fusion, el mejor tipo de característica son los Classemes, con un 37.33 % para estilo, 58.29 % para género, y 59.37 % para artista. Finalmente, para Feature Fusión, el mejor tipo de métrica es LMNN para estilo con un 45.97 %, ITML para género con un 60.28 %, y LMNN nuevamente para artista con un 63.06 %. El trabajo concluye que los Classemes tienen el mejor desempeño entre los distintos tipos de características, que Boost e ITML mejor la precisión sobre todas las características, que LMNN logra el mejor desempeño al usar Feature Fusion, y que LMNN con Classemes no solo logra superar el estado del arte de la época, sino que además logra hacerlo reduciendo la dimensionalidad del vector de características en un 90 %.

2.2.3. Recognizing Art Style Automatically in painting with deep learning

El trabajo con mayor parecido a la propuesta actual es Lecoutre et al. [10], del año 2017, ya que también busca hacer clasificación según estilos artísticos utilizando redes neuronales. Este trabajo busca hacer una clasificación de estilos artísticos usando las redes AlexNet y ResNet (tanto ResNet34 como ResNet50). AlexNet es una red neuronal convolucional, mientras que ResNet es una red neuronal residual, y ambas redes fueron preentrenadas con ImageNet. Estas redes fueron reentrenadas con *transfer learning* utilizando 80 000 imágenes de WikiArt etiquetadas con 25 estilos artísticos distintos, y 40 000 imágenes de ErgSap etiquetadas con

14 estilos distintos que coinciden con los estilos de WikiArt. La implementación fue hecha en Python usando Keras y Tensorflow.

Usando el conjunto de evaluación de WikiArt, AlexNet obtiene una precisión de 37.8 %, mientras que la precisión de ResNet50 es de 49.4 %, ambos después de 50 épocas de entrenamiento, y reentrenando solo la última capa. También se hace el experimento de reentrenar 20 % de las últimas capas de ResNet50, número de capas que se determinó como el mejor evaluado entre entrenar todas las capas y solo la capa final, lo que resulta en una precisión de 60 %, y luego logra aumentar la precisión a un 61 % al usar volteo horizontal de la imagen como parte de *bagging*. Finalmente, al usar *data augmentation*, el resultado mejora a un 62 %. El *data augmentation* utilizado consiste en el volteo horizontal, rotación (de 0 a 90 grados), traslación horizontal y vertical (entre 0 y el ancho/alto de la imagen), y zoom (con un factor entre 1 y 2).

2.2.4. Deep Transfer Learning for Art Classification Problems

Sabatelli et al. [15], del año 2019, hace una comparación entre cuatro redes neuronales, a las cuales se les aplica dos tipos de *transfer learning*, las cuales son *off-the-shelf classification* y *fine-tuning*. Las redes neuronales que se usaron fueron VGG19, Inception-V3, Xception y ResNet50. *Off-the-shelf classification* es un tipo de *transfer learning* en donde solo se reentrena la última capa de una red ya entrenada, mientras que *fine-tuning* reentrena todas las capas de una red ya entrenada. Las obras para entrenar y evaluar se recopilan de la base de datos del Rijksmuseum y de la base de datos de la ciudad de Antwerp. Es importante especificar que la base de datos de Antwerp no está etiquetada según material de la obra.

El desempeño de estas redes se mide según tres tipos de categorías, las cuales son material (papel, oro, porcelana, etc), tipo (impresión, escultura, dibujo, etc) y artista de cada obra. El desglose entre el número de obras para cada categoría y el número de etiquetas de cada categoría para ambas bases de datos se puede ver en la tabla 2.1. De las bases de datos antes mencionadas se usa el 80 % para entrenamiento, el 10 % para la evaluación, y el 10 % para la validación. No se usa *data augmentation* para el entrenamiento, el *batch size* es de 32, y el optimizador es distinto según el tipo de *transfer learning*, en donde se utiliza RMSprop como optimizador para *off-the-shelf classification* (con un tasa de aprendizaje igual a 0.001, el valor del *momentum* igual a 0.9 y un *épsilon* igual a 1×10^{-8}), y SGD para el optimizador de *fine-tuning* (donde la tasa de aprendizaje es 0.001 y el *momentum* es de 0.9). El entrenamiento se detiene una vez que la evaluación de la pérdida deja de mejorar después de 7 épocas seguidas.

Categoría	Base de datos	Obras	Categorías
Material	Rijksmuseum	110 668	206
Material	Antwerp	-	-
Tipo	Rijksmuseum	112 012	1054
Tipo	Antwerp	23 797	920
Artista	Rijksmuseum	82 018	1196
Artista	Antwerp	18 656	903

Tabla 2.1: Desglose base de datos

Los resultados obtenidos de las redes preentrenadas con Imagenet reentrenadas sobre la base de datos del Rijksmuseum se pueden ver en la tabla 2.2, mientras que los resultados obtenidos de las redes preentrenadas con Imagenet ("A") o que recibieron *fine-tuning* de la base de datos del Rijksmuseum ("B") se muestran en la tabla 2.3, donde "OTS" indica que se usa *off-the-shelf classification* para el *transfer learning*, y "FT" indica que se usa *fine-tuning* para el *transfer learning*. Los porcentajes en negrita indican el mejor resultado entre las distintas arquitecturas para cada tipo de clasificación.

Categoría	Red	Off-the-shelf classification	Fine-tuning
Material	Xception	87.69 %	92.13 %
Material	InceptionV3	88.24 %	92.10 %
Material	ResNet50	86.81 %	**92.95 %**
Material	VGG19	**92.12 %**	92.23 %
Tipo	Xception	74.80 %	90.67 %
Tipo	InceptionV3	72.96 %	91.03 %
Tipo	ResNet50	71.23 %	**91.30 %**
Tipo	VGG19	**77.33 %**	90.27 %
Artista	Xception	10.92 %	51.43 %
Artista	InceptionV3	0.07 %	**51.73 %**
Artista	ResNet50	0.08 %	46.13 %
Artista	VGG19	**38.11 %**	44.98 %

Tabla 2.2: Resultados Rijksmuseum

Categoría	Red	A + OTF	B + OTF	A + FT	B + FT
Tipo	Xception	42.01 %	62.92 %	69.74 %	72.03 %
Tipo	InceptionV3	43.90 %	57.65 %	70.58 %	71.88 %
Tipo	ResNet50	41.59 %	**64.95 %**	76.50 %	**78.15 %**
Tipo	VGG19	42.01 %	62.92 %	69.74 %	72.03 %
Artista	Xception	48.52 %	**54.81 %**	58.15 %	58.47 %
Artista	InceptionV3	21.29 %	53.41 %	56.68 %	57.84 %
Artista	ResNet50	22.39 %	31.38 %	62.57 %	**69.01 %**
Artista	VGG19	49.90 %	53.52 %	54.90 %	60.01 %

Tabla 2.3: Resultados Antwerp

El mejor desempeño en la base de datos del Rijksmuseum para la categoría de materiales se logra con ResNet50 usando *fine-tuning* y obteniendo un 92.95 % de precisión. Para la categoría de tipo se logra el mejor desempeño usando ResNet50 con *fine-tuning* nuevamente, obteniendo un 91.30 % de precisión. Finalmente, para la categoría de artista la mejor precisión obtenida es de un 51.73 %, resultado que se logra con InceptionV3 haciendo *fine-tuning* también. El desempeño en la base de datos de Antwerp muestra resultados similares, ya que se obtiene un 78.15 % de precisión para la categoría de tipo usando ResNet50 con *fine-tuning* despues de un *fine-tuning* adicional sobre la base de datos del Rijksmuseum, mientras que para la categoría de artista se logra una precisión de 69.01 %, usando la misma configuración a la de la red en la categoría de tipo ya mencionada.

Sabatelli et al. concluyen que el uso de *fine-tuning* mejora el desempeño de las redes de forma significativa en comparación al uso de *off-the-shelf classification* para el *transfer learning* en todos los casos. También concluyen que el desempeño de las redes preentrenadas con ImageNet supera el desempeño de las redes entrenadas desde cero, y que hacer *fine-tuning* sobre una base de datos similar, como hacer *fine-tuning* sobre la base de datos del Rijksmuseum antes de reentrenar con los datos de Antwerp, resulta en una precisión aun mayor.

2.2.5. Recognizing the Style of Visual Arts via Adaptive Cross-layer Correlation

Chen y Yang [2], del año 2019, utiliza una red neuronal convolucional con *adaptive cross-layer correlation*, una técnica que consiste en calcular el producto interno entre características extraídas de distintas capas de la red convolucional, lo que permite mejorar la representación de estilo. Además, se utiliza una matriz de pesos entre las distintas capas de características, matriz que es actualizada durante el entrenamiento, para obtener resultados aun mejores. La red utilizada en este trabajo es VGGNet preentrenado con ImageNet.

El objetivo de este trabajo consiste en clasificar obras según su artista y estilo, contando con 4266 imágenes entre 91 artistas para el primero y 2338 imágenes entre 13 estilos para el segundo. Estas obras provienen de la base de datos Painting91 Los resultados obtenidos muestran que la precisión para la clasificación de artistas alcanza un 70.65 %, mientras que para la clasificación de estilos alcanza un 78.13 %. También se usan otras dos bases de datos, una de arquitectura y otra de moda, tomadas de un trabajo anterior y de Hipster Wars, respectivamente, el primero de 4786 imágenes entre 25 estilos distintos, alcanzando una precisión de 73.67 %, y el segundo de 1893 imágenes entre 5 estilos distintos, alcanzando una precisión de 80.53 %.

2.2.6. Cross-Depiction Transfer Learning for Art Classification

En el trabajo de Sur y Blaine [22] del año 2017 se busca hacer una clasificación de artistas. Esto lo logra utilizando la red neuronal ResNet18 preentrenada con ImageNet y luego aplicando *transfer learning* a la base de datos objetivo del Rijksmuseum, la cual consiste de 13 674 obras entre 25 artistas distintos. Lo que hace este trabajo distinto de propuestas anteriores es que además utiliza histogramas de gradientes orientados para extraer características de las imágenes, así como también una técnica de transferencia de estilos, usando capas de estilo que computan con la correlación espacial de valores de una imagen a través de una matriz de Gram.

Los resultados obtenidos de los experimentos ya mencionados se reducen en lo siguiente. Para el *transfer learning* sin ninguna de las técnicas adicionales se consigue una precisión de 82.5 %, mientras que para la técnica de histograma de gradientes de estilo la precisión lograda es de 59.5 %, y finalmente, para la técnica de transferencia de estilos, se logra una precisión de 63.5 %. Es importante mencionar que este último resultado se logró usando SqueezeNet

en vez de ResNet18.

2.2.7. Artist Identification with Convolutional Neural Networks

Viswanathan [24] desarrolla un algoritmo de clasificación de artistas para el año 2017 utilizando distintas variedades de redes neuronales convolucionales. La base de datos utilizada proviene de WikiArt, de la cual se usan 57 artistas con 300 imágenes para cada uno. Entre las redes utilizadas se incluye una red neuronal convolucional personalizada simple como punto de referencia, una red ResNet18 entrenada desde cero, y una red ResNet18 preentrenada con ImageNet y aplicando *transfer learning* usando la base de datos objetivo.

El resultado obtenido para la red usada como punto de referencia fue una precisión de 42.2 %, mientras que para ResNet18 desde cero fue de un 51.1 %, y para ResNet18 preentrenado con Imagenet la precisión fue de un 77.7 %. A partir de los distintos experimentos realizados el trabajo concluye que las características que el preentrenamiento con ImageNet logra extraer son particularmente relevantes para la clasificación de artistas.

Capítulo 3

Problema

3.1. Descripción

Este trabajo busca clasificar obras artísticas según estilo usando una combinación de análisis de arquetipos con redes neuronales convolucionales. Más específicamente, se usa la red neuronal para crear una serie de vectores de características para cada imagen, que luego son utilizados para crear arquetipos, los cuales se usan para aproximar la imagen a clasificar, y la combinación convexa del conjunto de arquetipos asociados a un estilo que mejor aproxima la imagen original es el estilo que se le predice a la imagen.

La red neuronal a utilizar está basada en la red neuronal convolucional llamada VGG19 [23] preentrenada para clasificar objetos de 1000 categorías distintas a partir de más de un millón de imágenes de ImageNet [7], una base de imágenes orientada a software de reconocimiento visual.

A esta red neuronal se le aplica *transfer learning* para clasificar según estilo artístico usando una base de datos de WikiArt [25], una enciclopedia de arte online que cuenta con más de 170 000 obras distintas, las cuales que ya vienen etiquetadas según estilo artístico. Dentro de esta base de datos se busca entrenar y clasificar entre los 25 estilos con una mayor cantidad de obras, contando así con al menos 1300 imágenes por cada estilo.

3.2. Relevancia

El uso del aprendizaje de máquina en el arte es relevante debido a la diversidad de información asociada a las obras artísticas, información que si bien no es de fácil acceso, puede ser de gran utilidad para diversos fines. Es tal su importancia que en los últimos años se han publicado diversos trabajos respecto a la clasificación de distintas propiedades de obras artísticas, tales como periodo artístico por Yang et al. [27] y género artístico por Saleh et al. [16].

Este problema que resulta ser interesante debido a que la clasificación de estilos no se basa

en la clasificación de objetos concretos, como seria el caso en la clasificación de herramientas, animales o edificios, ya que el estilo de una obra se basa en características más abstractas que no necesariamente tienen un análogo en el mundo físico, como lo son las paletas de color (que no siempre coinciden con la paleta de color del objeto a representar del mundo físico), el uso de la perspectiva, las técnicas de trazado y las figuras abstractas, entre otras características más sutiles y complejas de describir, sin que el objeto representado en la obra sea necesariamente relevante para determinar su estilo, a tal punto que el objeto puede incluso llegar a desaparecer cuando se trata de obras de arte abstracto. Esto se puede evidenciar en la Figura 3.1, donde se puede apreciar que dos pinturas que representan a un mismo contenido pueden pertenecer a estilos distintos debido a las distintas formas en las que este mismo contenido puede ser representado, siendo en este caso que el contenido de las obras es un grupo de personas mientras que los estilos corresponden al *Proto Renaissance* y al *Baroque*. Es este nivel de abstracción entonces lo que diferencia a este problema de clasificación de otros problemas del área.

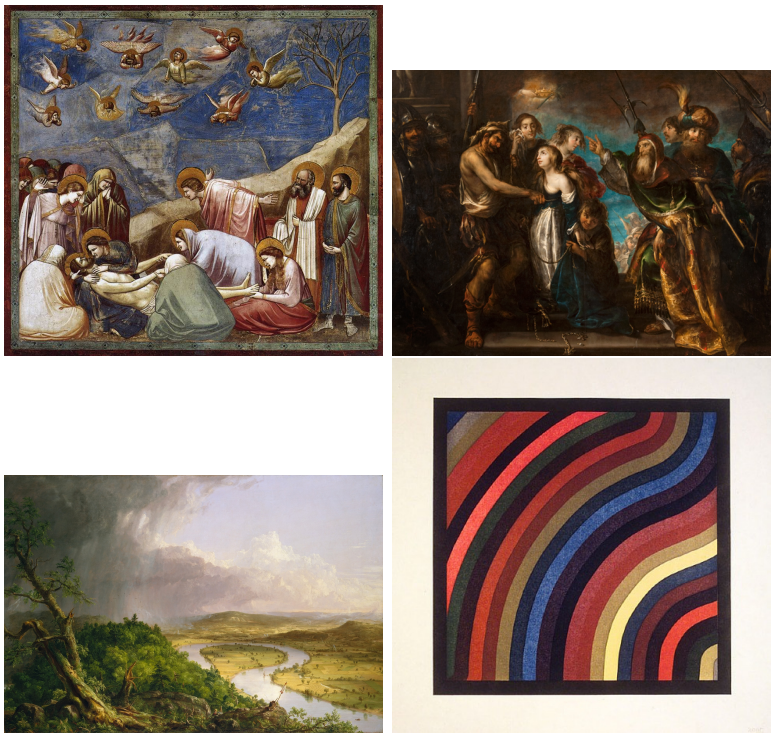

Figura 3.1: Cuatro ejemplos de obras obtenidas de WikiArt.
Las obras mostradas son, de izquierda a derecha y luego de arriba a abajo: "Lamentation" de Giotto (*Proto Renaissance*), "The Martyrdom of St. Catherine" de Jan Cossiers (*Baroque*), "The Oxbow" de Thomas Cole (*Romanticism*), y "Two Centimeter Wavy Bands in Colors" de Sol LeWitt (*Post-Painterly Abstraction*).

Si bien los algoritmos de clasificación según estilo artístico ya existen, tales como Lecoutre et al. [10] y Sabatelli et al. [15], esta propuesta difiere de propuestas previas por su uso de análisis de arquetipos al momento de hacer las clasificaciones. Se busca entonces explorar el potencial desempeño asociado a la combinación de redes convolucionales con el análisis de arquetipos.

Existen diversas aplicaciones posibles que se pueden desarrollar a partir de mejores algoritmos de clasificación de obras artísticas. Algunas de las aplicaciones que se han sugerido en trabajos anteriores incluyen la posibilidad de encontrar relaciones novedosas entre artistas por Sur. et al. [22], identificar falsificaciones por Viswanathan [24], e implementar sistemas de recomendaciones a partir de la exploración de características por Yang et al. [27]. Viswanathan también hace un énfasis en la importancia de poder etiquetar de forma eficiente obras artísticas recién digitalizadas a medida que empiezan a crecer las colecciones de arte online. Algunas aplicaciones que se sugieren en este trabajo incluyen sistemas de recomendaciones no solo para obras artísticas históricas, sino también para portales de arte contemporáneo, comisiones artísticas y diseño gráfico, tales como Artspace, ArtStation, Fiverr y DesignCrowd, que pueden contar con una cantidad de obras considerablemente mayor en comparación a las bases de datos de arte histórico, razón por la cual la identificación de estilos para estas plataformas podría llegar a cumplir un rol bastante relevante si se usa como la base de un sistema de recomendaciones para sus clientes.

3.3. Requisitos de la solución

La implementación de la solución se basa principalmente en una combinación de redes convolucionales con análisis de arquetipos, y su elaboración consiste de distintos pasos, cada uno con diferentes requisitos. Primero, es necesario armar la base de datos, para lo cual se requiere una fuente de donde sacar las imágenes, que en este caso es WikiArt. Luego se entrena una red neuronal a partir de los datos anteriores usando *transfer learning*, para lo cual se utiliza la red neuronal convolucional llamada VGG19 preentrenada con ImageNet. Se requiere también de un algoritmo para convertir las imágenes en un vector de características, para lo cual se utiliza un algoritmo presentado por Wynen et al. [26]. Después se reduce la dimensionalidad de este vector con el algoritmo TruncatedSVD de scikit-learn. Finalmente, se utiliza el algoritmo elaborado por Bauckhage [1] de cálculo de arquetipos para así armar los arquetipos a partir de los vectores anteriores y así recrear la imagen usando los arquetipos calculados, consiguiendo así una aproximación de estilo de la imagen original que se usa para la clasificación.

3.4. Caracterización de los resultados

A partir de este trabajo se espera analizar el desempeño obtenido de combinar el uso de redes neuronales convolucionales con el análisis de arquetipos a través de la precisión obtenida respecto a predecir los estilos artísticos de una base de datos de evaluación, siendo esta precisión definida como el número de aciertos dividido por el número total de predicciones. Se busca luego comparar este desempeño con el desempeño de una red neuronal convolucional respecto a la misma base de datos antes mencionada. En otras palabras, se decide analizar las distintas estrategias de clasificación según la precisión obtenida por cada configuración. El uso de la precisión como métrica de evaluación se justifica por el hecho de que es esta métrica la que permite evaluar que tan seguido el algoritmo acierta al estilo de las imágenes a clasificar, además de permitir una comparación entre el método de análisis de arquetipos con

el de las redes neuronales. También es importante mencionar que es esta métrica la que es utilizada en otros trabajos similares, como aquellos que son detallados en la sección "Estado del arte".

3.5. Resultado esperable

El resultado esperable de la solución propuesta es que se logre una precisión mayor a la que ofrece la red neuronal convolucional tradicional, sin embargo, es importante considerar la posibilidad de que este caso no se cumpla. Independiente de lograr el resultado esperable o no, se busca analizar los resultados obtenidos y evaluar las distintas estrategias implementadas al momento de hacer la clasificación planteada, y de esa forma discutir sobre las posibles ventajas y desventajas de la solución propuesta, además de encontrar técnicas útiles para mejorar esta misma solución.

Capítulo 4

Solución

Los métodos principales a utilizar son las redes neuronales convolucionales y el análisis de arquetipos, los cuales se detallan en el capitulo "Marco teórico y estado del arte".

Para cumplir el objetivo se utiliza una base de datos de más de 170 000 obras artísticas de una enciclopedia de arte online llamada WikiArt [25]. Las obras ya vienen catalogadas según estilo por el mismo sitio. Esta base de datos contiene obras de casi 200 estilos distintos, 100 de los cuales contienen al menos 100 obras distintas, y 35 de los cuales contienen más de 1000 obras distintas.

La red neuronal es preentrenada con ImageNet para luego aplicar *transfer learning* usando la base de datos antes mencionada, esto con la intención de obtener una red enfocada en clasificar estilos artísticos.

Otras tecnologías utilizadas son las librerías Keras para las redes neuronales, junto a SciPy y NumPy de Python para la programación en general.

Los experimentos se ejecutan en un servidor con 132 GB de RAM, 23 CPU i9-9920X, y dos GPU GeForce RTX 2080 Ti de 12 GB, siendo la GPU particularmente relevante al momento de entrenar las redes neuronales, ya que este hardware permite hacer el entrenamiento con mayor eficiencia en cuanto a tiempo se refiere.

4.1. Implementación

4.1.1. Armar base de datos

El primer paso para desarrollar la solución al problema consiste en obtener una base de datos con los metadatos apropiados, para lo cual se utiliza la base de datos de WikiArt [25], una enciclopedia online de artes visuales que permite a sus usuarios subir obras y etiquetarlas según artista, año, estilo, género y medio. Esta base de datos cuenta con más de 170 000 obras artísticas, catalogadas en casi 200 estilos distintos, 100 de los cuales contienen al menos 100 obras distintas, 35 de los cuales contienen más de 1000 obras distintas, y 25 de los cuales

contienen más de 1600 obras. En este trabajo se utilizan los 25 estilos con la mayor cantidad de obras. Para descargar esta base de datos se usa una herramienta llamada "Wikiart Retriever" [11]. Esta herramienta descarga las obras dentro de una carpeta con el año de producción de nombre, la cual queda guardada en una carpeta de artista. En otras palabras, la estructura es de una carpeta por cada artista, conteniendo esta carpeta una carpeta por cada año, donde finalmente se guardan las obras. Como las obras se usan para entrenar una red neuronal convolucional, es necesario limpiar la base de datos, reordenando la estructura de las carpetas para usar una carpeta por cada estilo artístico, donde finalmente se guardan las imágenes en las carpetas de sus respectivos estilos. Para lograr esto se utiliza un código que permite copiar las imágenes desde su estructura original hasta la estructura apropiada para hacer el entrenamiento, dividiendo estas nuevamente según carpeta de entrenamiento y carpeta de evaluación. Es importante notar que esto se debe hacer dos veces. La primera vez para entrenar y evaluar la red neuronal, usando 1000 imágenes para entrenar y 250 imágenes para evaluar. La segunda vez para calcular los arquetipos y evaluar la eficacia de estos, usando 250 imágenes aleatorias del conjunto de 1000 imágenes de entrenamiento de la red anterior para calcular los arquetipos, y las mismas 250 imágenes de valimiento de la red anterior para evaluar el desempeño del análisis de arquetipos. En esta ocasión se usan solo 1000 imágenes para el entrenamiento de la red neuronal con la intención de reducir el tiempo de ejecución de la red neuronal durante los experimentos de exploración, aunque para el experimento final se usa toda la base de datos disponible de todas formas. La razón de por que se usan 250 imágenes para el resto de las carpetas se debe al hecho de que, al momento de hacer la reducción de dimensionalidad a los vectores de características para el análisis de arquetipos, el algoritmo de reducción de dimensionalidad no logra contar con la memoria RAM suficiente cuando se usan más de 250 imágenes.

4.1.2. Entrenar red neuronal convolucional

Para la red neuronal convolucional se utiliza una configuración similar al trabajo de de Wynen et al. [26], siendo la arquitectura utilizada una red VGG19, ya que este trabajo es el que hace la transferencia de estilos de una imagen a otra, y por lo tanto se espera que una configuración similar a la de transferencia de estilos tenga un buen desempeño para categorizar estilos también.

Algunas de las variables con las cuales se experimenta son el número de épocas (entre 0 y 20), la tasa de aprendizaje (entre 10^{-4} y 10^{-6}), inicialización de pesos (inicialización aleatoria e "ImageNet" de Keras) y número de capas a reentrenar (entre todas y ninguna).

Algunas de las variables con las cuales no se experimenta son el *batch size* (igual a 32), el optimizador (Adam) y la función de pérdida (*categorical cross-entropy*). Adam es un optimizador de descenso de gradiente estocástico basado en estimaciones adaptables de momentum de primer y segunda orden, y su uso en este trabajo se justifica por el trabajo de Choi et al. [4], en donde se establece que optimizadores como Adam nunca tienen un desempeño inferior a optimizadores como SGD (una popular alternativa de optimizador), siempre y cuando los hiperparámetros estén debidamente configurados. *Cross-entropy* es una medida de la diferencia entre dos distribuciones probabilísticas para un conjunto de eventos dado, y en este trabajo se utiliza *categorical cross-entropy* como función de perdida debido al hecho de que

es esta función la que permite un calculo de la perdida cuando se hacen predicciones entre mas de dos clases que no representan un rango de números.

Para las imágenes a procesar se usa *data augmentation* de 30 grados de rotación, 20 % de traslación horizontal y vertical, 20 % de zoom, y voltear la imagen horizontalmente. Esta técnica se utiliza para mejorar el desempeño de la red, ya que el *data augmentation* logra aumentar la cantidad de datos disponibles para el entrenamiento haciendo ligeras modificaciones a las imágenes de entrada, siendo cada modificación un dato útil adicional para mejorar el desempeño de la red, sin que estas modificaciones alteren el estilo de las imágenes a utilizar.

4.1.3. Convertir las imágenes en vectores de características

El siguiente paso consiste en implementar una técnica utilizada en Wynen et al. [26] para lograr una representación vectorial del estilo artístico de la imagen. Este vector se consigue usando las capas de la red neuronal convolucional mencionada en la sección anterior. Más específicamente, y siguiendo la recomendación de Gatys et al. [6] mencionada en Wynen et al. [26], se usa la primera capa de cada bloque de capas convolucionales para lograr una representación de estilo, dando un total de cinco capas a utilizar.

Para convertir la imagen en su vector de características, se ingresa la imagen en la red neuronal y se guarda la salida que producen las capas antes mencionadas de esta red. La salida de cada capa se puede interpretar como una matriz de dimensiones p_l y m_l, donde p_l es el número de canales de la capa, m_l es el número de posiciones de pixeles de la capa, y l es la capa a guardar. Luego, se calculan las estadísticas de primer y segundo orden respecto a m_l, usando las definiciones 4.1 y 4.2, donde F es la matriz resultante de guardar la salida de las capas seleccionadas, lo que resulta en un vector de dimensión p_l y una matriz de dimensiones $p_l \times p_l$, respectivamente, antes de que esta matriz se aplane a un vector de dimensión p_l^2. A continuación, tanto al vector de la matriz aplanada de estadísticas de segundo orden como al vector de estadísticas de primer orden se le aplica una normalización, dividiendo estos vectores por $p_l * (p_l + 1)$, con la intención de reducir la sobrerrepresentación de las capas con un mayor número de parámetros. Finalmente, se concatenan estos últimos dos vectores para obtener el vector de características de la imagen. La decisión de utilizar un vector de características definido de esta forma proviene del trabajo de Li et al. [20], también mencionado por Wynen et al. [26], donde se demuestra que este vector tiene una capacidad destacable para representar estilos visuales. Sin embargo, es importante mencionar que la normalización es una sugerencia de Wynen et al. [26] por sobre el trabajo anterior, con la intención de reducir la sobrerrepresentación de las capas con un mayor número de parámetros.

Definición 4.1 *Estadísticas de primer orden*

$$\mu = \frac{1}{m_l} \sum_{j=1}^{m_l} F_l[j] \in \mathbb{R}^{p_l}$$

Definición 4.2 *Estadísticas de segundo orden*

$$\Sigma = \frac{1}{m_l} \sum_{j=1}^{m_l} (F_l[j] - \mu_l)(F_l[j] - \mu_l)^T \in \mathbb{R}^{p_l \times p_l}$$

Finalmente, se concatenan los vectores de estadísticas de primer y segundo orden de las cinco capas a utilizar en un solo vector. En este caso, el vector de características para cada imagen resulta ser de largo 611776, ya que las cinco capas a utilizar tienen los siguientes valores de p_l: $64, 128, 256, 512, 512$, lo que se traduce en que sus vectores de estadísticas de primer orden tengan los siguientes largos: $64, 128, 256, 512, 512$, y el largo de sus vectores de estadísticas de segundo orden tenga los siguientes largos: $64^2, 128^2, 256^2, 512^2, 512^2$, por lo que al concatenar todos estos largos el vector resultante es de largo 611776.

4.1.4. Reducir la dimensionalidad de los vectores de características

Como el vector de características es excesivamente largo, contando con más de 600 000 valores, es necesario aplicar una reducción de dimensionalidad, ya que en caso contrario el calculo de los arquetipos puede llegar a tomar un tiempo cercano a las 24 horas, sin si quiera obtener un desempeño superior, mientras que la reducción usada en este trabajo permite reducir este mismo calculo a no mas de 2 horas. Volviendo a seguir la recomendación de Wynen et al. [26], se decide reducir el largo del vector a 4096, ya que esto permite conservar el 99 % de la varianza del vector, tanto en el trabajo citado como en el trabajo actual. Para lograr esto se utiliza la función TruncatedSVD de scikit-learn [17], que aplica una reducción de dimensionalidad a un vector usando una versión condensada de *singular value decomposition* (SVD). Una restricción importante de la función es que su documentación indica que, para hacer la reducción de dimensionalidad a un valor k, es necesario contar con un conjunto de al menos k ejemplos para hacer la reducción. En este caso, como se usan 250 ejemplos para cada uno de los 25 estilos, se tiene un total de 6250 ejemplos, suficientes como para lograr la reducción a 4096 dimensiones. Es importante notar que el algoritmo TruncatedSVD ofrece un objeto que es entrenado con el conjunto de ejemplos que se le ingresa, pero luego puede ser utilizado para reducir la dimensionalidad de cualquier otro conjunto de imágenes usando la misma reducción que fue aplicada al conjunto de ejemplos anterior. En efecto, esto es lo que se hace al momento de hacer la reducción de la carpeta de evaluación del análisis de arquetipos: en vez de usar el algoritmo TruncatedSVD dos veces, este se usa solo en las imágenes de la carpeta de entrenamiento, y luego se utiliza el objeto SVD entrenado anteriormente para hacer la conversión de las imágenes de la carpeta de evaluación. Esto se hace para que sea la misma configuración de reducción que se aplique a tanto las imágenes de entrenamiento como de evaluación.

4.1.5. Calcular los arquetipos

Basándonos en la propuesta de Y. Chen et al. [3], se genera un conjunto arbitrario de arquetipos por cada estilo a utilizar. Para calcular los arquetipos se utiliza la implementación

desarrollada por Bauckhage [1] para el análisis de arquetipos a través del algoritmo de Frank-Wolfe, cuya función "AA" recibe como input una pila de vectores de características y un número k de arquetipos a calcular, y retorna una pila de arquetipos del largo especificado creados a partir de la pila ingresada. Esto se hace para todos los estilos a clasificar, resultando así en un número k de arquetipos para cada uno de los 25 estilos artísticos. El valor de k es arbitrario, razón por la cual se exploran distintos valores para esta variable, buscando el valor que logre una mayor precisión al momento de hacer la clasificación a partir del análisis de arquetipos. Es importante mencionar que el valor de k es un número entre cero y el número de imágenes a utilizar para armar los arquetipos, ya que no es posible calcular más arquetipos que el número de imágenes que se usan para armar estos arquetipos.

4.1.6. Reconstrucción de imágenes a partir de arquetipos

Nuevamente, se utiliza el código desarrollado por Bauckhage [1], pero esta vez solo se utiliza la función "fwUpdateZ", una función que utiliza el algoritmo de Frank-Wolfe para obtener un vector Z tal que $X \approx AZ$ a partir de X y A, donde X corresponde a la imagen a aproximar, A representa a los arquetipos de un estilo que se usaran para aproximar a la imagen, y Z representa al vector que mejor logra la aproximación a través de la multiplicación ya descrita. Lo que se busca entonces es la combinación lineal de los vectores de características de los arquetipos para cada estilo que logran hacer la mejor aproximación al vector de características de la imagen a clasificar, lo cual se determina calculando la distancia euclidiana entre el vector de la imagen original y el vector de la imagen aproximada, siendo entonces el estilo de los arquetipos que producen la menor distancia aquel que la función determina como el estilo de la imagen a clasificar. En otras palabras, dada una imagen, se itera por cada estilo, y usando solo los arquetipos de ese estilo, se retorna la distancia entre la imagen a clasificar y la combinación de arquetipos del estilo que mejor logra reconstruir la imagen original, donde finalmente se determina el estilo de la imagen original como el estilo que logra la menor distancia antes mencionada. La intuición detrás de esta propuesta es que cada estilo artístico tenga su propio conjunto de arquetipos que lo represente bien, por lo que el conjunto de arquetipos de un estilo que mejor represente a una nueva imagen debería ser aquel que contribuye mejor a su composición original, y por lo tanto, es este estilo al cual la imagen original tiene más probabilidades de pertenecer.

4.1.7. Determinar categoría y calcular precisión

Una vez determinado el estilo de cada imagen del conjunto de evaluación, en base a la menor distancia obtenida entre la imagen a clasificar y los arquetipos de estilo que mejor reconstruyen la imagen a clasificar, se convierten estos resultados en una matriz de confusión, y finalmente se calcula la precisión tomando la suma de la diagonal de la matriz de confusión y dividiendo este número por el número total de imágenes de evaluación para obtener la precisión final. Esta precisión es el valor a utilizar para evaluar el desempeño de las distintas configuraciones a explorar.

Capítulo 5

Resultados

5.1. Experimentos exploratorios

Se hicieron diversos experimentos con el objetivo de obtener el mejor resultado posible usando diversas configuraciones para la combinación de redes convolucionales y análisis arquetipos. Se busca experimentar con distintas variables, tales como tasa de aprendizaje, número de épocas, número de arquetipos, cantidad de capas a reentrenar, entre otros, las cuales se detallan a lo largo de este capítulo.

5.1.1. Configuración estándar

Los siguientes experimentos fueron probados con la siguiente configuración a menos que se especifique lo contrario. Se usa una red VGG19 entrenada durante 20 épocas, usando *data augmentation*, preentrenado con ImageNet, reentrenando todas las capas, con una tasa de aprendizaje de 1×10^{-5}, usando dos capas *fully connected* con 4096 neuronas cada una, usando 1000 imágenes para el entrenamiento de la red convolucional, 250 imágenes para la elaboración de los arquetipos (tomadas de las imágenes de entrenamiento para la red convolucional) y 250 imágenes para evaluar la precisión de tanto la red como el análisis de arquetipos (las mismas imágenes para tanto el caso de la red y los arquetipos). Más específicamente, el *data augmentation* consiste en rotaciones aleatorias entre un rango de -30 a 30°, desplazamiento vertical y horizontal de la imagen entre un rango de 0 a 20 %, zoom entre un rango de 0 a 20 %, y volteo horizontal de la imagen. Esta configuración suele tomar alrededor de 24 horas en ejecutar usando el hardware mencionado al principio del capítulo "Solución". La precisión de la red al usar esta configuración, siendo la precisión definida como el número de aciertos dividido por el número total de predicciones, es de un 43 % entre los 25 estilos artísticos, lo que resulta ser la mayor precisión de todas las redes evaluadas, razón por la cual se ha determinado esta red como la configuración estándar.

Una nota importante a aclarar es que la configuración estándar utiliza solo 1000 imágenes para cada estilo artístico en el entrenamiento, a pesar de que existe una cantidad considerablemente mayor para muchos de los estilos analizados. La razón de por que no se usan todas

las imágenes disponibles es porque con utilizar solo 1000 imágenes por estilo, el entrenamiento toma una hora por época, mientras que utilizar todas las imágenes disponibles toma 8 horas por época, razón por la cual entonces se decide hacer el experimento con todas las imágenes disponibles una vez que ya se han hecho los experimentos necesarios para acercarnos a la configuración ideal.

Los gráficos de precisión y pérdida durante el entrenamiento de la red estándar se pueden revisar en las figuras 5.1 y 5.2. Del gráfico de pérdida obtenido se puede concluir que la red ya habría alcanzado su óptimo una vez entrenada en torno a las 10 épocas, aunque aun así se decide seguir con la red entrenada después de 20 épocas debido a que, si bien la red ha sido levemente sobreentrenada, el desempeño resulta ser ligeramente superior en el gráfico de precisión y el análisis de arquetipos al entrenar la red durante estas 20 épocas, como se puede evidenciar en la sección "Número de épocas" de este capitulo.

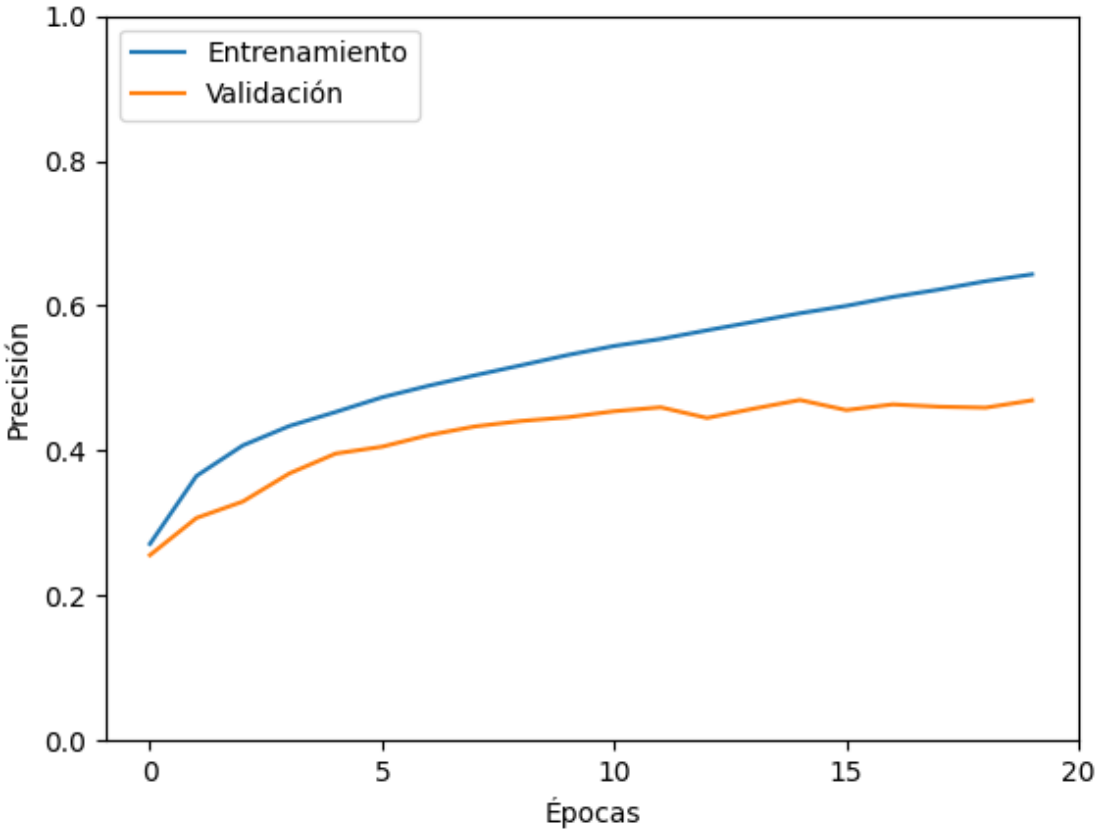

Figura 5.1: Precisión de la red entrenada utilizando la configuración estándar.

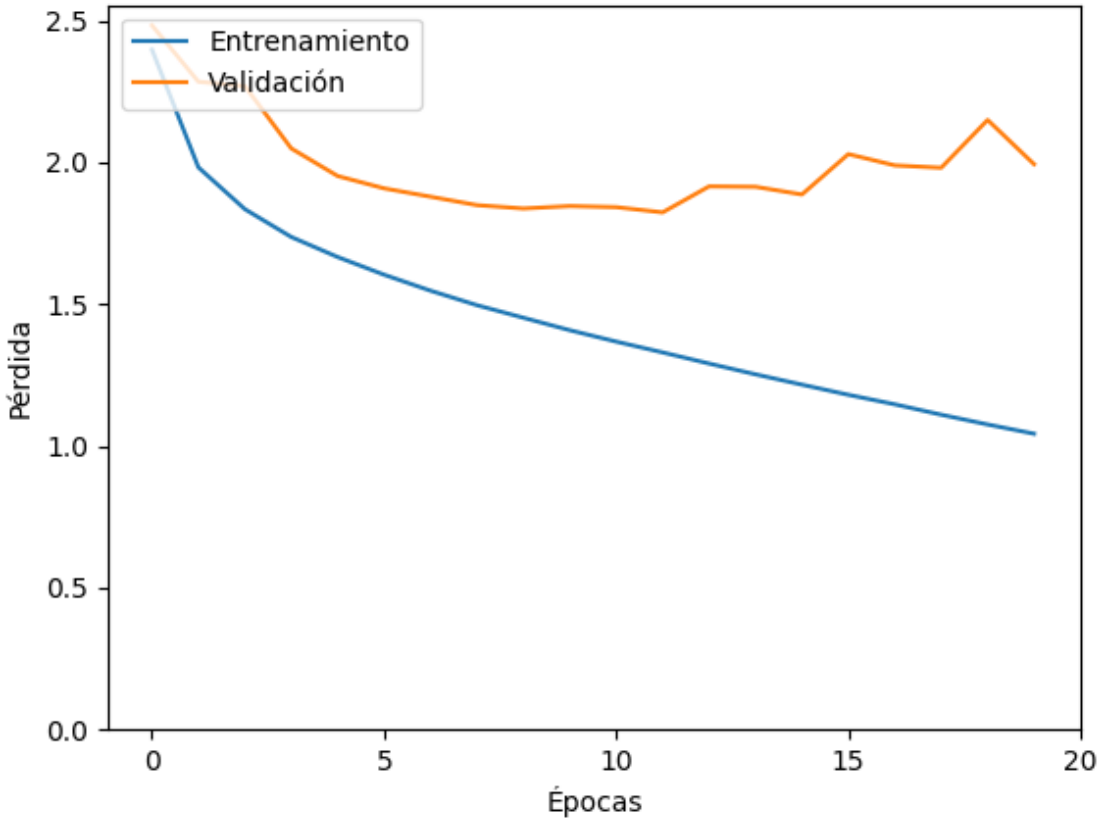

Figura 5.2: Pérdida de la red entrenada utilizando la configuración estándar.

5.1.2. Argumento "tmax"

La implementación de Bauckhage [1] para el análisis de arquetipos requiere de una función llamada "fwUpdateZ", que busca una combinación lineal de arquetipos que mejor aproxima a un vector particular, el cual depende de un argumento "tmax". Este argumento "tmax" indica el número de iteraciones del algoritmo Frank-Wolfe para encontrar esta combinación lineal. Si bien el "tmax" utilizado originalmente fue de 100, escogido por ser la recomendación de Bauckhage [1], los resultados obtenidos fueron menores a los esperados, razón por la cual se decide aumentar el número de iteraciones para verificar si la recomendación es apropiada para este trabajo, aunque esto no logra una mejora significativa, además de aumentar el tiempo de ejecución. Los resultados obtenidos se pueden ver en la tabla 5.1. A partir de los resultados obtenidos se concluye que usar un "tmax" igual a 100 es suficiente para este trabajo.

"tmax"	Precisión (250 arq.)
100	32.912 %
1000	32.480 %

Tabla 5.1: Argumento "tmax"

5.1.3. Número de arquetipos

A continuación se muestran los resultados según el número de arquetipos utilizados. Lo esperable hubiera sido que exista un número fijo de arquetipos que logre maximizar la precisión, sin embargo, como se puede ver en los resultados de la tabla 5.2, esta situación no ocurre, ya que el mejor resultado es de 32.768 % al usar la mayor cantidad de arquetipos posible, siendo que la precisión de la red utilizada para conseguir este resultado de un 43 %.

Una hipótesis de por que esto no se cumple es que el número ideal de arquetipos está por sobre los 250 arquetipos, pero esto no se puede comprobar con el equipamiento actual, ya que para contar con más de 250 arquetipos, es necesario utilizar más de 250 imágenes para hacer la reducción de dimensionalidad en la cual se basan los arquetipos, pero el equipamiento actual agota la memoria RAM al hacer la reducción anterior. Una forma alternativa de comprobar esta hipótesis podría ser haciendo la reducción a 250 imágenes, y luego usar el reductor entrenado con estas 250 imágenes para reducir un conjunto de imágenes considerablemente mayor. Este experimento se realiza en la sección "Aumento de Arquetipos".

Es importante mencionar que usar la mayor cantidad de arquetipos posibles necesariamente se traduce en que cada imagen es usada como un arquetipo. Es decir, si se arman los arquetipos usando 250 imágenes, y luego se arman 250 arquetipos a partir de estas imágenes, cada imagen resulta ser su propio arquetipo, razón por la cual no es necesario calcular los arquetipos. Aun así, sigue siendo necesario hacer la combinación convexa de estas imágenes, que usa un algoritmo similar a la de creación de arquetipos, para calcular qué tan bien aproximan estas imágenes a las imágenes a categorizar.

Número de arquetipos	Precisión
1	15.088 %
2	20.72 %
3	21.184 %
5	22.56 %
10	24.304 %
20	25.696 %
50	28.24 %
100	29.888 %
150	31.536 %
200	32.464 %
250	32.768 %

Tabla 5.2: Número de arquetipos

5.1.4. Tasa de aprendizaje

En vista de los resultados de la red neuronal (43 % de precisión), se propone la idea de que la tasa de aprendizaje podría no ser la adecuada, razón por la cual se decide hacer el mismo experimento con tasas de aprendizaje distintas, presentándose estas tabuladas en la tabla 5.3. De esta tabla se puede observar que disminuir la tasa de aprendizaje de 1×10^{-4} a 1×10^{-5} resulta en una mejora significativa, mientras que disminuir la tasa de aprendizaje de 1×10^{-5} a 1×10^{-6} no, tanto en la evaluación de la red neuronal, como en la evaluación de los arquetipos, razón por la cual se determina que la tasa de aprendizaje de 1×10^{-5} es la más adecuada.

Tasa de aprendizaje	Precisión (CNN)	Precisión (50 arq.)	Precisión (250 arq.)
1×10^{-4}	32 %	22.24 %	25.968 %
1×10^{-5}	43 %	27.952 %	32.96 %
1×10^{-6}	43 %	27.696 %	33.088 %

Tabla 5.3: Tasa de aprendizaje

5.1.5. Reducción de dimensionalidad

Si bien la solución propuesta considera un paso de reducción de dimensionalidad a vectores de largo superior superior a los 600 000 valores en vectores de solo 4096 valores, es posible omitir este paso y hacer el calculo de arquetipos a partir de los vectores originales, sin necesidad de hacer la reducción. Los resultados obtenidos entre usar y no usar la reducción de dimensionalidad para el análisis de arquetipos se muestran en la tabla 5.4. Si bien el desempeño del análisis de arquetipos es superior cuando se usa la reducción, la razón de esto se puede deber al hecho de que la implementación del análisis de arquetipos debería usar un argumento "tmax" considerablemente mayor a 100 para que los arquetipos armados en base a los vectores de largo mayor a 600 000 valores converjan. Sin embargo, aun usando un valor de "tmax" igual a 100, el tiempo de ejecución para el experimento sin reducción de dimensionalidad dura 30 horas, por lo que aumentar el valor de "tmax" podría tomar un tiempo considerablemente mayor a las 30 horas, mientras que el tiempo de ejecución para el experimento con reducción solo toma unas 4 horas. Sumado a lo anterior, una de las razones para usar vectores sin reducción de dimensionalidad es que se evita la restricción de memoria RAM en el número de imágenes a utilizar, aunque en este caso el tiempo de ejecución aumenta de forma proporcional al número de imágenes a utilizar, por lo que si se quisiera usar 1000 imágenes en vez de 250 imágenes, el tiempo de ejecución pasaría a ser mayor a las 120 horas, aun sin considerar el aumento en el tiempo de ejecución del argumento "tmax" para quizás obtener una precisión similar al método con reducción. También es importante considerar que en la sección "Aumento de arquetipos" de este capitulo se idea una estrategia para emular un mayor número de arquetipos sin requerir de un tiempo de ejecución considerablemente mayor como es el caso en el método sin reducción de dimensionalidad. Es por estas razones que se decide no hacer mas experimentos sin el método de reducción de dimensionalidad.

Reducción de dimensionalidad	Precisión (250 arq.)
Si	33.200 %
No	17.712 %

Tabla 5.4: Reducción de dimensionalidad

5.1.6. Entrenamiento desde cero

El siguiente experimento consiste en entrenar la red neuronal desde cero. En vez de tomar una red preentrenada con ImageNet, se decide tomar una red que no ha sido entrenada anteriormente, inicializando el valor de sus neuronas de forma aleatoria. Al hacer esto, y siguiendo la configuración estándar, se pueden examinar los resultados en la tabla 5.5, donde se puede observar que el desempeño del análisis de arquetipos es considerablemente menor en comparación al *transfer learning* de la red preentrenada con ImageNet. Esto se puede deber a que es necesario un entrenamiento por un número considerablemente mayor de épocas, debido a que tanto en los gráficos de precisión como de pérdida, que se pueden apreciar en las figuras 5.3 y 5.4, existe la evidencia de que el desempeño de la red seguía mejorando en la red que utiliza una tasa de aprendizaje de 1×10^{-4}, que es 10 veces mayor a la tasa de aprendizaje usado en casi todos los otros experimentos de este trabajo. También es importante considerar que el desempeño de esta red puede mejorar aun más al ir reduciendo paulatinamente la tasa de aprendizaje a medida que pasan las épocas. Sin embargo, se decide no proceder con estos experimentos. Las razón detrás de esta decisión incluye el hecho de que entrenar una red desde cero requiere del reentrenamiento de todas sus capas, además de demandar un entrenamiento por una cantidad de épocas considerablemente mayor a la de otras redes, y también podría requerir de distintas configuraciones alternativas para poder encontrar su desempeño ideal, siendo todas estas razones que aumentan el tiempo de ejecución de este tipo de redes de forma considerable, y por lo tanto, debido al tiempo total de ejecución requerido para entrenar estas redes, y debido a las restricciones de tiempo de este trabajo, es que estos experimentos no se realizan.

Tasa de aprendizaje	Prec. (CNN)	Prec. (50 arq.)	Prec. (150 arq.)	Prec. (250 arq.)
1×10^{-4}	28 %	16.608 %	18.448 %	19.040 %
1×10^{-5}	26 %	13.344 %	14.528 %	14.816 %

Tabla 5.5: Entrenamiento desde cero

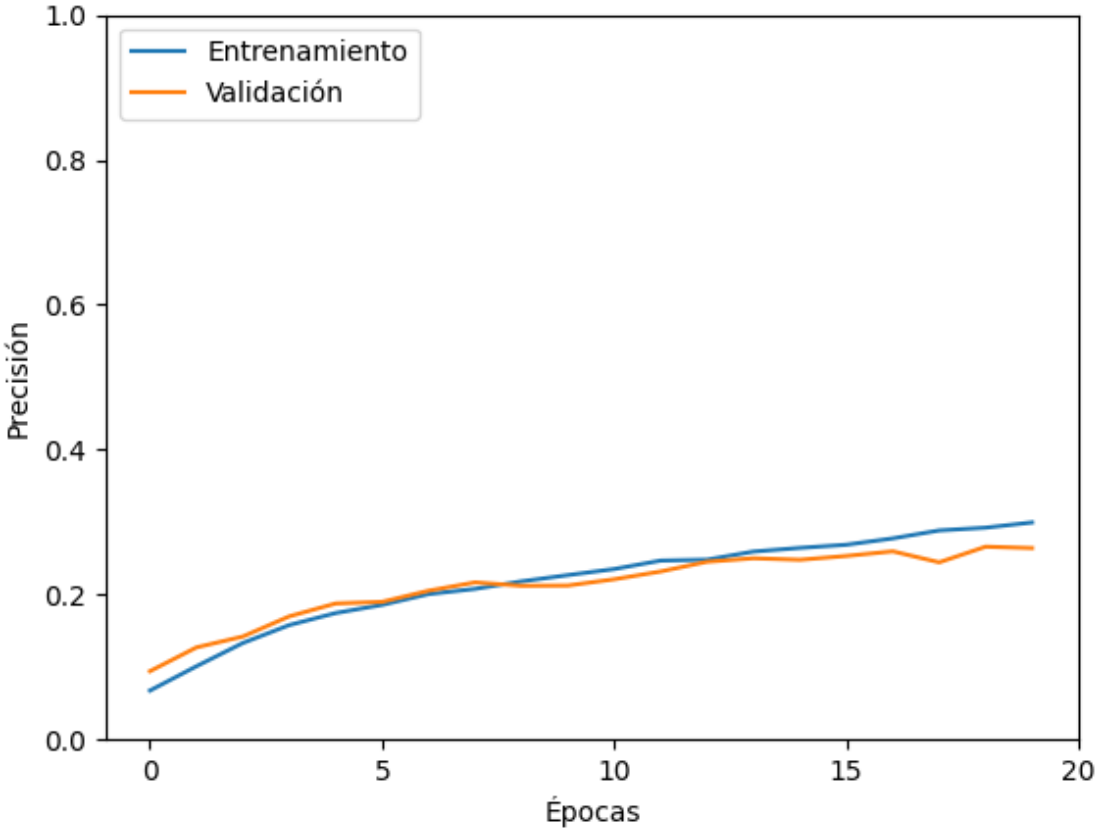

Figura 5.3: Precisión de la red entrenada desde cero utilizando una tasa de aprendizaje de 1×10^{-4}.

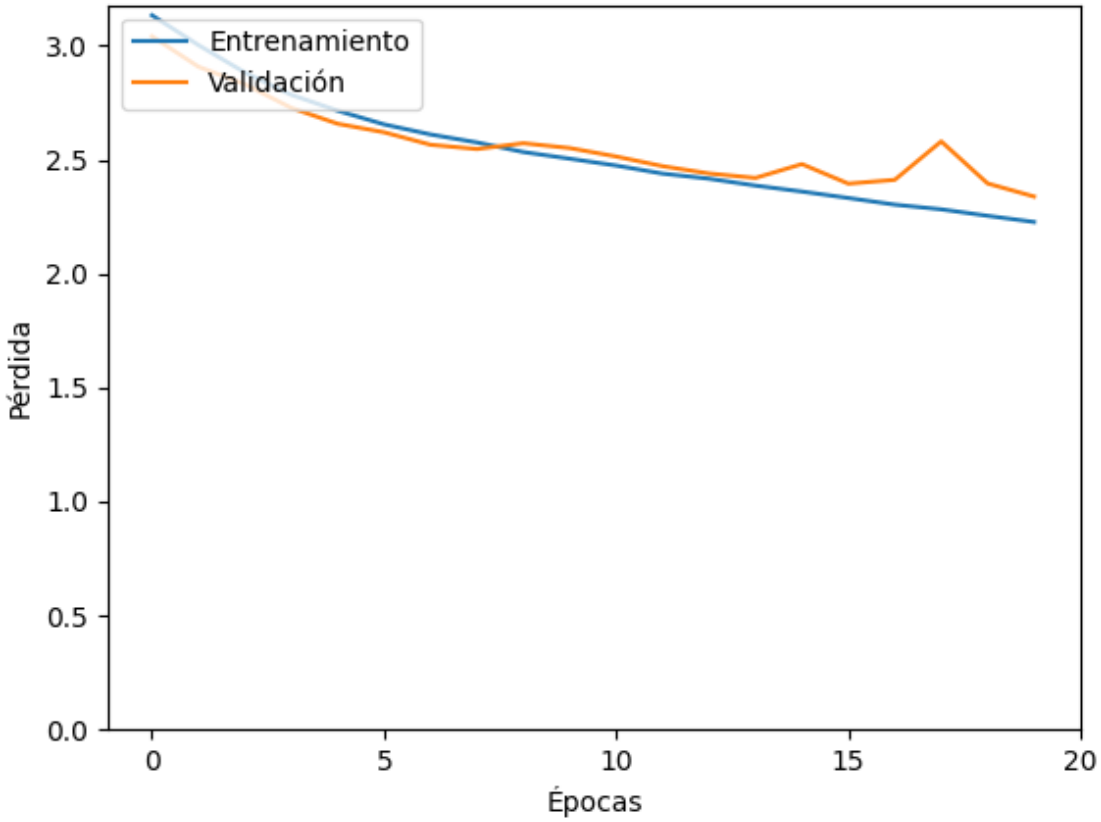

Figura 5.4: Pérdida de la red entrenada desde cero utilizando una tasa de aprendizaje de 1×10^{-4}.

5.1.7. Número de épocas

Se decide entrenar una red con la configuración estándar en periodos de cinco épocas y luego obtener el desempeño de cada periodo, resultado que se puede revisar en la tabla 5.6.

Sin embargo, los resultados no son los esperados. La hipótesis inicial es que a medida que se entrena la red con las imágenes según estilo artístico, el desempeño del análisis de arquetipos iba a mejorar, pero esto no ocurre. Es más, el desempeño del análisis de arquetipos empeora después de que la red es reentrenada, siendo por lo tanto superior el desempeño de los arquetipos armados con la red preentrenada con ImageNet sin haberle aplicado el *transfer learning*. Una hipótesis de por que esto ocurre es que distintas capas se entrenan de forma inadecuada, ya que pierden la caracterización que ofrece el preentrenamiento de ImageNet, y al mismo tiempo no logra entrenar suficiente la caracterización que ofrece el *transfer learning* realizado. De ser este el caso, las primeras capas se podrían congelar, ya que estas están bien entrenadas según ImageNet, siendo solo las últimas capas las que se deberían entrenar. Esta idea proviene de Lecoutre et al. [10], donde concluye que el mejor desempeño se logra al reentrenar solo el 20 % de las últimas capas, argumentando que las capas más profundas

suelen ser similares en distintos tipos de clasificación.

Otra observación es particularmente relevante por el hecho de que la precisión del análisis de arquetipos no parece mejorar de forma significativa después de entrenarlo por cinco épocas, y se cree que este resultado podría ocurrir después de entrenar con una sola época, razón por la cual se realiza un experimento entrenando por una sola época para confirmar esta hipótesis, y una vez hecho esto, se buscarán las razones detrás de este resultado.

Épocas	Prec. (CNN)	Prec. (50 arq.)	Prec. (150 arq.)	Prec. (250 arq.)
0	n/a	31.184 %	34.576 %	37.344 %
5	34 %	27.456 %	30.832 %	33.616 %
10	40 %	27.008 %	30.896 %	33.424 %
15	42 %	27.872 %	31.104 %	33.68 %
20	43 %	27.52 %	31.024 %	33.856 %

Tabla 5.6: Épocas

5.1.8. Entrenamiento de una a cinco épocas

Según lo discutido en la sección anterior, al revisar los resultados en la tabla 5.7 se comprueba la hipótesis de que basta con entrenar la red por solo una época para que el desempeño de la clasificación por arquetipos baje considerablemente.

Épocas	Prec. (CNN)	Prec. (250 arq.)
0	n/a	37.344 %
1	23.248	33.808 %
2	27.696	33.488 %
3	29.568	33.536 %
4	32.752	32.704 %
5	34.304	32.784 %

Tabla 5.7: Desempeño durante las primeras cinco épocas.

5.1.9. Capas a utilizar

La recomendación de Gatys et al. [6] es usar la primera capa de cada bloque de capas convolucionales para lograr una representación de estilo, sin embargo, se intenta usar la última capa para tratar de obtener mejores resultados, ya que estas últimas capas podrían ser las qué se entrenan mejor en el caso de estilos artísticos al momento de hacer la conversión a los arquetipos, pero como se puede ver en los resultados de la tabla 5.8, esto no se logra traducir en una mejora, razón por la cual se sigue utilizando la primera capa de cada bloque en los siguientes experimentos.

Capa	Precisión (CNN)	Precisión (250 arq.)
Primera	43 %	34.048 %
Última	43 %	30.800 %

Tabla 5.8: Capas a utilizar

5.1.10. Número de capas a reentrenar

En relación a una de las hipótesis mencionadas en la sección "Número de épocas", que menciona la posibilidad de que las primeras capas no se están entrenando de forma adecuada, razón por la cual la precisión del análisis de arquetipos es menor cuando se reentrena la red, se decide experimentar entrenando solo las últimas capas de la red, esperando así mantener las primeras capas bien entrenadas que se estarían reentrenando de forma inadecuada, y así reentrenar solo las últimas capas, que podrían ser más eficientes después de ser reentrenadas.

Considerando la arquitectura de la red VGG19 que se está usando para este trabajo, se puede apreciar que esta se divide entre varios bloques de capas convolucionales separados por capas de *max pooling*, por lo que se decide hacer tres experimentos: entrenar todas las capas, entrenar todas las capas desde los últimos dos bloques de redes convolucionales, entrenar todas las capas desde el último bloque de redes convolucionales, y finalmente, no entrenar ninguna capa, para confirmar si se verifica la hipótesis. Los resultados de estos experimentos se pueden ver en la tabla 5.9.

De los resultados se puede observar que el desempeño de la red que solo ha reentrenado las capas desde su último bloque tiene un desempeño superior a las redes que entrenaron una mayor cantidad de capas, además de lograr un desempeño similar a la red sin entrenar al usar 50, 150 y 250 arquetipos.

Capas a reentrenar	Prec. (CNN)	Prec. (50 arq.)	Prec. (150 arq.)	Prec. (250 arq.)
Ninguna capa	n/a	31.184 %	34.576 %	37.344 %
Último bloque	41 %	30.448 %	35.248 %	37.360 %
Últimos dos bloques	42 %	29.936 %	34.736 %	36.448 %
Todas las capas	43 %	27.520 %	31.024 %	33.856 %

Tabla 5.9: Cantidad de capas a reentrenar

5.1.11. Reentrenamiento de las capas del último bloque

Considerando el hecho de que la red con el mayor desempeño fue aquella a la cual se le entrenaron todas sus capas desde el último bloque de capas convolucionales en comparación a las redes que se entrenaron con un mayor número de capas, se propone analizar el desempeño de entrenar la red utilizando progresivamente menos capas de los últimos dos bloques de capas convolucionales, con la esperanza de encontrar un óptimo de capas a reentrenar.

Como se puede observar de los resultados mostrados en la tabla 5.10, el desempeño para el análisis de arquetipos no solo tiende a mejorar al reentrenar solo las últimas cuatro capas convolucionales, sino que es en este número de capas donde se encuentra el mejor desempeño, lo que también coincide con que las últimas cuatro capas convolucionales son justamente las capas del último bloque de capas convolucionales, a pesar de que el desempeño de la red neuronal mejora a medida de que se reentrena la mayor cantidad de capas. Es importante señalar que la mejora sigue siendo bastante pequeña, pero al menos se logra superar de forma consistente el desempeño de no reentrenar ninguna capa, a diferencia de experimentos anteriores donde se reenetrenan todas las capas y aun así el desempeño era inferior al de no reentrenar.

Últimas capas convolucionales	Prec. (CNN)	Prec. (250 arq.)
0	n/a	37.344 %
1	36.272 %	37.376 %
2	36.544 %	37.456 %
3	38.224 %	37.408 %
4	38.960 %	37.488 %
5	38.784 %	37.376 %
6	40.016 %	37.440 %
7	39.152 %	37.408 %
8	40.128 %	36.752 %

Tabla 5.10: Desempeño de entrenar las 5 últimas capas convolucionales por 20 épocas.

5.1.12. Ponderación por capas

En vista de los resultados obtenidos en la sección de reentrenamiento por capas, donde los resultados logran mejorar según las capas que se decide reentrenar, se propone usar distintas ponderaciones por cada capa para lograr la representación de estilo, dándole mayor o menor peso a las primeras o últimas capas. También se hace una ponderación en donde solo se considera una capa de las cinco seleccionadas originalmente. La red utilizada es la red que fue reenetrenada en sus últimas cuatro capas convolucionales durante 5 épocas. Los resultados se pueden ver en la tabla 5.11, donde x representa el vector de características, p representa la profundidad de la capa, y "bloque" hace referencia al conjunto de capas convolucionales consecutivas de la red.

Tipo de ponderación	Prec. (250 arq.)
x	37.760 %
x/p	37.696 %
$x/p/(p+1)$	37.440 %
$x*p$	37.712 %
$x*p*(p+1)$	37.792 %
Solo primera capa del último bloque	41.152 %
Solo última capa del último bloque	25.792 %
Solo última capa del penúltimo bloque	41.088 %
Solo primera capa de los últimos dos bloques	37.712 %
Solo última capa de los últimos dos bloques	41.136 %

Tabla 5.11: Desempeño de distintas ponderaciones a una red que reentrena sus últimas cuatro capas convoluciones por 5 épocas.

Como se puede observar de los resultados obtenidos, el mejor desempeño se logra utilizando solo la primera capa del último bloque. Es importante notar que la ponderación "Solo última capa del penúltimo bloque" corresponde a una capa que no es reentrenada, ya que esta capa esta fuera las últimas cuatro capas convolucionales que se reentrenan en este experimento.

En vista del buen desempeño al utilizar solo una capa para lograr una representación de estilo, se decide probar esta técnica entre distintas redes, cuyas precisiones se muestran en la tabla 5.12, donde la red A es una CNN preentrenada con ImageNet sin reentrenamiento, la red B es una CNN preentrenada con Imagenet reentrenada en sus últimas cuatro capas convolucionales por 20 épocas, y la red C es una CNN preentrenada con ImageNet reentrenada en todas sus capas por 20 épocas. De los resultados obtenidos se puede señalar que la red C obtiene el mejor desempeño, por lo que se propone analizar el desempeño de sus últimas 8 capas convolucionales de forma individual, y experimentar si combinando las dos mejores capas se logra un desempeño aun mejor.

Tipo de red	Tipo de ponderación	Prec. (250 arq.)a
Red A	Solo primera capa del último bloque	41.200 %
Red A	Solo última capa del último bloque	30.928 %
Red B	Solo primera capa del último bloque	41.888 %
Red B	Solo última capa del último bloque	31.552 %
Red C	Solo primera capa del último bloque	42.496 %
Red C	Solo última capa del último bloque	37.760 %
Red C	Solo primera capa de los últimos dos bloques	37.808 %
Red C	Solo última capa de los últimos dos bloques	41.232 %

Tabla 5.12: Desempeño de distintas redes al usar distintas capas individuales para el análisis de arquetipos.

Como se puede ver en los resultados de la tabla 5.13, la mejor capa es la primera del último bloque, lo que es consistente con los resultados obtenidos en los experimentos de

la tabla 5.10. También se intenta combinar dos capas, la primera del último bloque con la cuarta del último bloque, para ver si esto logra mejorar el resultado aun más, sin embargo, esto resulta en una precisión de 42.080 %, por lo que no se logra una mejora en comparación a la mejor de las dos capas.

Capa a utilizar	Prec. (250 arq.)
Cuarta del último bloque	37.760 %
Tercera del último bloque	39.648 %
Segunda del último bloque	40.368 %
Primera del último bloque	42.496 %
Cuarta del penúltimo bloque	41.088 %
Tercera del penúltimo bloque	41.264 %
Segunda del penúltimo bloque	39.808 %
Primera del penúltimo bloque	37.728 %

Tabla 5.13: Desempeño de capas individuales de la red C al hacer el análisis de arquetipos.

El hecho de que se logre un mejor desempeño usando las capas finales en vez de las capas iniciales sugiere que, para lograr una representación de estilo, el reentrenamiento reduce el desempeño de las primeras capas, pero mejora el desempeño de las últimas capas. Esto se puede deber a que las últimas capas de las redes neuronales suelen representar mejor las características más abstractas de las imágenes, lo que podría incluir estilo artístico, mientras que las primeras capas suelen representar mejor las características más concretas de la imagen. Sin embargo, es importante también considerar que la red usada para el reentrenamiento fue preentrenada con ImageNet, una base de datos compuesta en su mayoría por fotografías de objetos reales, razón por la cual las primeras capas podrían necesitar un mayor entrenamiento para adaptarse a las peculiaridades del mundo artístico, que no siempre imitan las características del mundo físico. Se sugiere hacer un entrenamiento desde cero para verificar si se logra una mejor caracterización con las primeras capas. Si bien este experimento ya se realizó, el desempeño general de la red fue considerablemente inferior a la red reentrenada, aunque esto puede deberse a que la red entrenada desde cero necesita entrenamiento por una mayor cantidad de épocas. Sin embargo, para este trabajo se decide no proceder con el experimento entrenado por una mayor cantidad de épocas. Las razones de esta decisión se detallan al final de la sección "Entrenamiento desde cero".

5.1.13. Aumento de arquetipos

Buscando verificar la hipótesis presentada en la sección "Número de Arquetipos" respecto a si se logra un mejor desempeño al usar la mayor cantidad de arquetipos posibles o si esto lo logra con algún número fijo de arquetipos, se decide llevar a cabo el experimento propuesto en esa sección. Este experimento consiste en usar el reductor entrenado con 250 imágenes, ya que este es el máximo número de imágenes que permiten los recursos computacionales actuales al ejecutar el reductor de dimensionalidad, para luego entregar 1000 imágenes (o 750 imágenes adicionales) que pasan a ser reducidas por ese reductor, y así emular un aumento en el número de arquetipos disponibles. Para este experimento se sigue usando el mejor método

encontrado en la sección "Ponderación por capas", que consiste en seleccionar solo la primera capa del último bloque para armar los arquetipos usando la red preentrenada con ImageNet que tuvo todas sus capas reentrenadas. Los resultados obtenidos usando estos 1000 arquetipos se pueden ver en la tabla 5.14.

Número de arquetipos	Precisión
1000	50.128 %
750	49.424 %
500	48.336 %
350	47.232 %
250	44.304 %
150	42.560 %
50	42.272 %

Tabla 5.14: Desempeño de utilizar distintas cantidades de arquetipos a partir de los 1000 arquetipos emulados.

Como se puede ver de los resultados obtenidos, el desempeño mejora al usar la mayor cantidad de arquetipos disponibles, lo que parece contrarrestar la hipótesis de que existe un número fijo de arquetipos para alcanzar el mejor desempeño, aunque también existe la posibilidad de que este número fijo sea mayor a 1000. Aun así, es importante mencionar que el mejor resultado obtenido, con un desempeño de un 50.128 %, es bastante significativo en relación al experimento anterior, cuyo mejor desempeño solo alcanza un 42.496 %. También es importante remarcar que en este experimento el análisis de arquetipos supera el desempeño de la red neuronal convolucional con el cual se arma este análisis de arquetipos, teniendo esta red un 43 % de precisión, lo que sugiere que, al momento de usar una igual cantidad de imágenes de entrenamiento (1000 en este caso), el desempeño del análisis de arquetipos supera a la red neuronal convolucional. Sin embargo, es importante recordar que el método de análisis de arquetipos presentado en este trabajo requiere de una mayor cantidad de recursos que la red neuronal para una misma cantidad de imágenes.

5.2. Experimento final

5.2.1. Configuración final

En el experimento final se decide usar la base de datos completa para el entrenamiento de la red, además de utilizar la configuración con el mejor desempeño obtenido en los experimentos exploratorios, siendo esta configuración la utilizada en la sección "Aumento de arquetipos".

La base de datos completa para este experimento cuenta con un total de 120 937 imágenes entre los 25 estilos artísticos usados, mientras que la base de datos usada en los experimentos anteriores contaba con 31 250 imágenes para el mismo número de estilos. El estilo con menos imágenes de esta base de datos es *Neo-Expressionism* con 1385, mientras que el estilo con más

imágenes es *Impressionism* con 15 381. Es importante mencionar que para este experimento final se eliminan las imágenes que pertenecen a más de un estilo artístico. Esta medida se justifica por el hecho de que, al tener más de un estilo por imagen, esta imagen aparece más de una vez en la base de datos, apareciendo una vez en la carpeta de cada estilo al que pertenece. Esto implica que al validar la clasificación de las imágenes, el clasificador necesariamente falla en al menos la mitad de los casos, debido a que el clasificador retorna la misma clase para las imágenes duplicadas, pero las imágenes duplicadas pertenecen todas a clases distintas.

La configuración para este experimento consiste en utilizar una red VGG19 preentrenada con ImageNet, a la cual se le aplica *transfer learning* a todas las capas con la base de datos antes mencionada usando *data augmentation*, una tasa de aprendizaje de 1×10^{-5}, y una división entre el conjunto de entrenamiento y evaluación en una proporción de 80 a 20, respectivamente. Para el análisis de arquetipos se utilizan 1000 imágenes de la base de datos de entrenamiento de la red para el aumento de arquetipos, y 250 de estas imágenes se utilizan para conformar la matriz base para la función de reducción de dimensionalidad. Los vectores de características se arman usando solo la salida de la primera capa del último bloque de capas convolucionales. La reducción de dimensionalidad se hace a 4096 dimensiones.

5.2.2. Entrenamiento con la base de datos completa

La precisión obtenida para cada configuración usando un conjunto de 250 imágenes de evaluación para cada uno de los 25 estilos distintos se puede revisar en la tabla 5.15. Cabe destacar que la precisión alcanzada por la red neuronal convolucional es de un 57 %, mientras que el mejor desempeño del análisis de arquetipos es de un 50.0 % al usar 1000 arquetipos aumentados. Es importante mencionar que el tiempo de ejecución para este experimento final toma cerca de 100 horas con el equipamiento mencionado al principio del capítulo "Solución", usando aproximadamente 4 horas y media por cada época para las 20 épocas de entrenamiento de la red neuronal, y unas pocas horas más para hacer la clasificación usando la técnica de análisis de arquetipos.

Número de arquetipos	Precisión
CNN	57 %
1000	50.0 %
750	48.768 %
500	47.392 %
250	44.064 %

Tabla 5.15: Desempeño de la configuración final.

5.2.3. Matriz de confusión

Usando la misma base de datos de 250 imágenes por estilo como conjunto de evaluación, se muestra la matriz de confusión para tanto la red neuronal convolucional como para el

análisis de arquetipos, los cuales se pueden ver en las figuras 5.5 y 5.6, respectivamente.

Luego, para visualizar con facilidad qué método logra destacar en qué estilo, se resta la matriz de confusión del análisis de arquetipos a la matriz de confusión de la red neuronal convolucional, cuyo resultado muestra con valores positivos un mejor desempeño para el análisis de arquetipos y con valores negativos el mejor desempeño para la red neuronal convolucional. Este resultado se puede ver en la figura 5.7. Tomando las clases con una diferencia mayor o igual a 25 aciertos entre la resta de las matrices anteriores, se puede señalar que la red neuronal convolucional tiene un mejor desempeño en los estilos *Art Nouveau (Modern)*, *Baroque*, *Expressionism*, *Realism*, *Romanticism* y *Surrealism*, mientras que el análisis de arquetipos tiene un desempeño superior en los estilos *Art Informel*, *Impressionism*, *Mannerism (Late Renaissance)*, *Minimalism* y *Pop Art*. Sin embargo, no parece haber una relación entre los distintos estilos de cada conjunto, salvo quizás que los estilos en los que la red neuronal convolucional funciona mejor tienden a estilos realistas (siendo *Expressionism* la excepción), mientras que el análisis de arquetipos tiende a estilos menos realistas (siendo *Mannerism (Late Renaissance)* la excepción), aunque para confirmar esta hipótesis es necesario hacer una experimentación más completa.

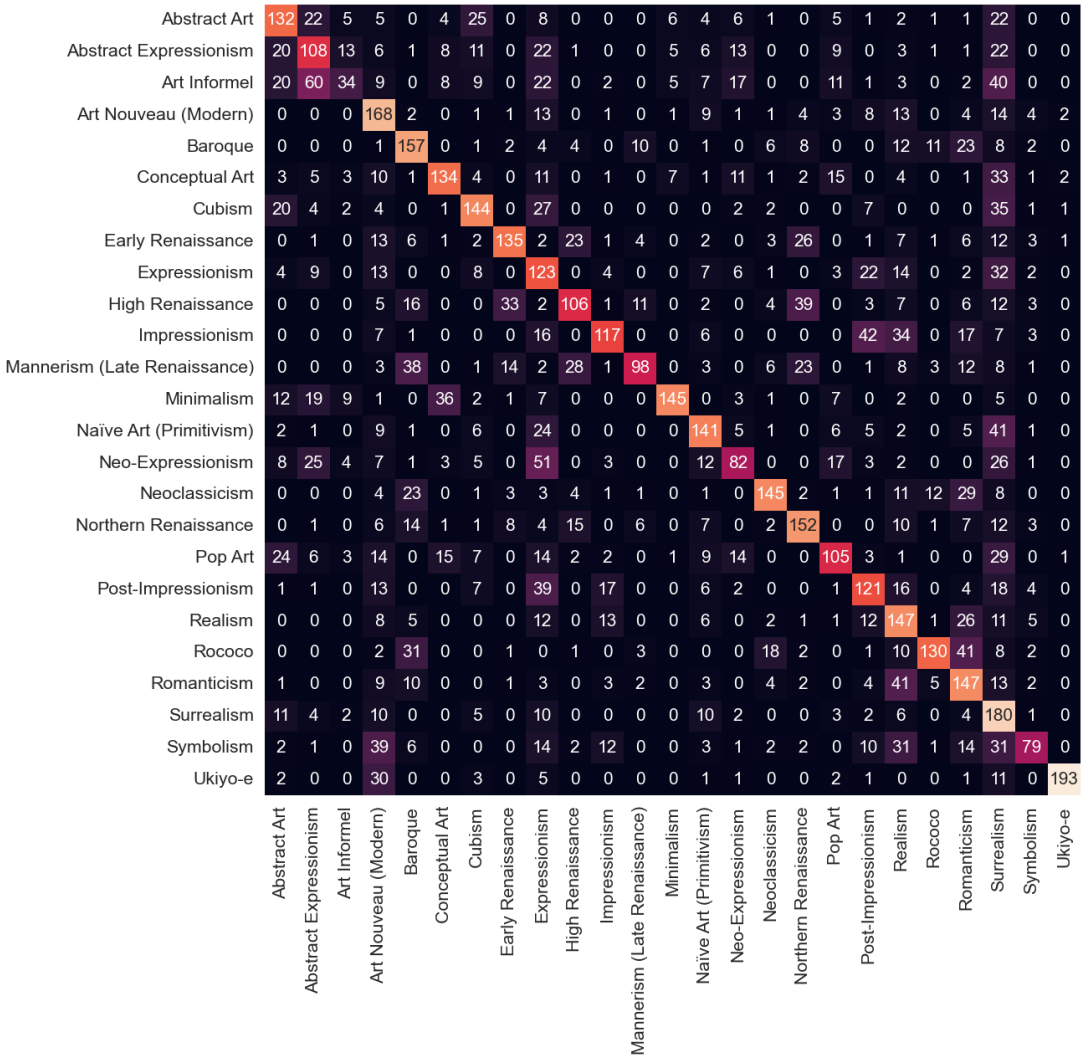

Figura 5.5: Matriz de confusión de la red neuronal convolucional, con la clase actual a lo largo del eje vertical y la clase predicha a lo largo del eje horizontal.

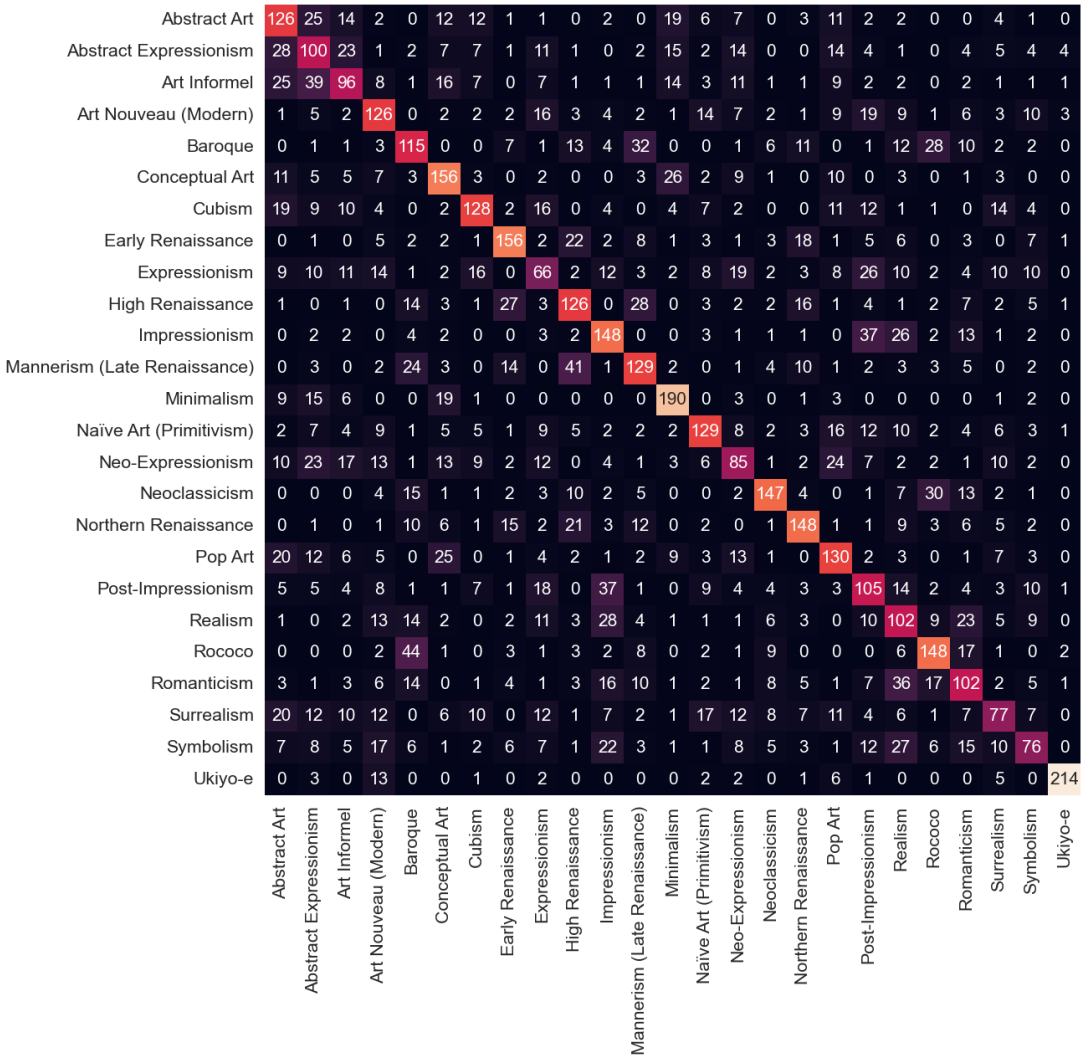

Figura 5.6: Matriz de confusión del análisis de arquetipos, con la clase actual a lo largo del eje vertical y la clase predicha a lo largo del eje horizontal.

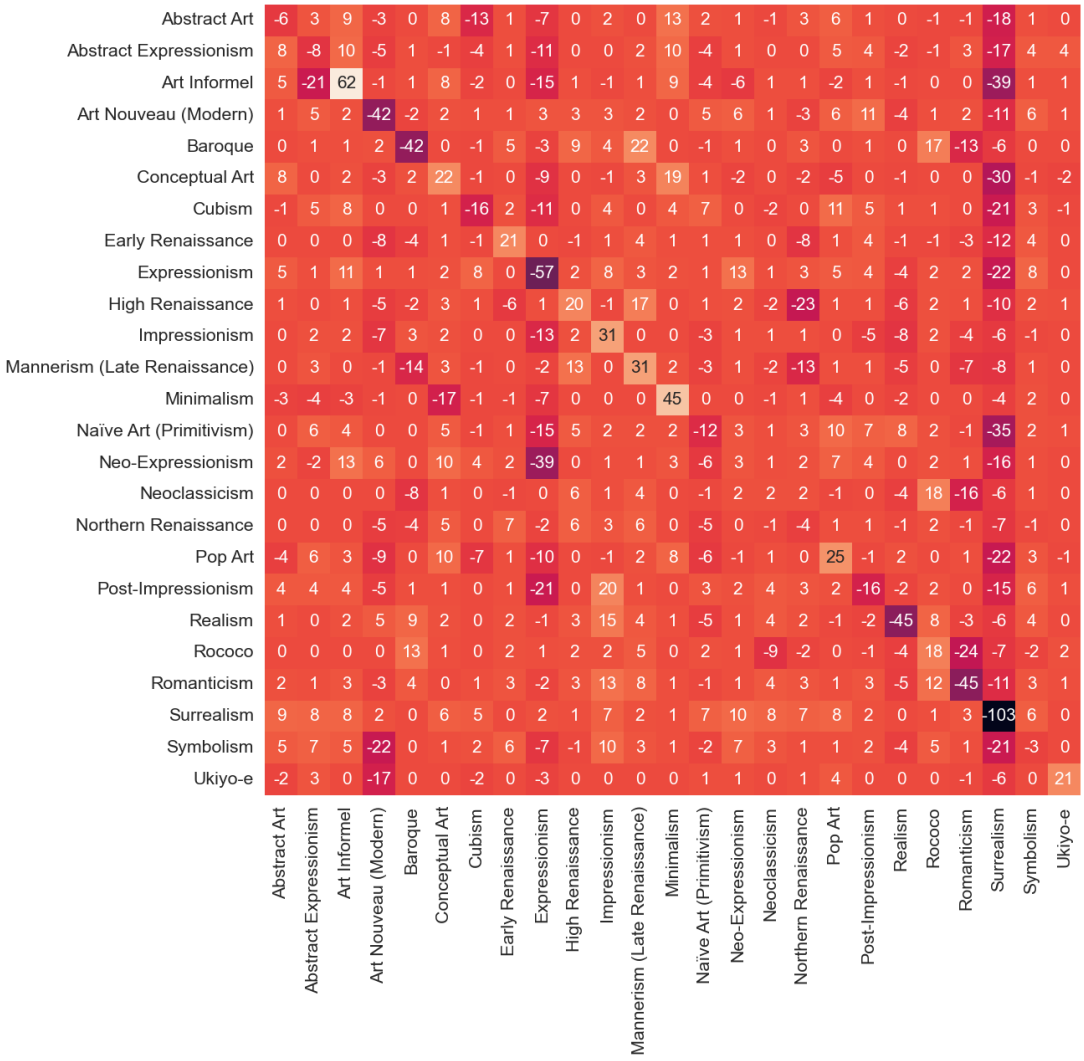

Figura 5.7: Matriz de confusión del análisis de arquetipos restando la matriz de confusión de la red neuronal convolucional.

5.2.4. Precisión y pérdida

En las figuras 5.8 y 5.9 se puede observar la precisión y la pérdida de esta configuración final, donde se puede señalar que después de las 20 épocas la red neuronal ya estaba cerca de su mejor desempeño, tanto en precisión como pérdida se refiere, demostrando así que un entrenamiento por un mayor número de épocas no hubiera significado una mejora considerable en el desempeño de la red, además de mostrar que la red no ha sido sobreentrenada tampoco.

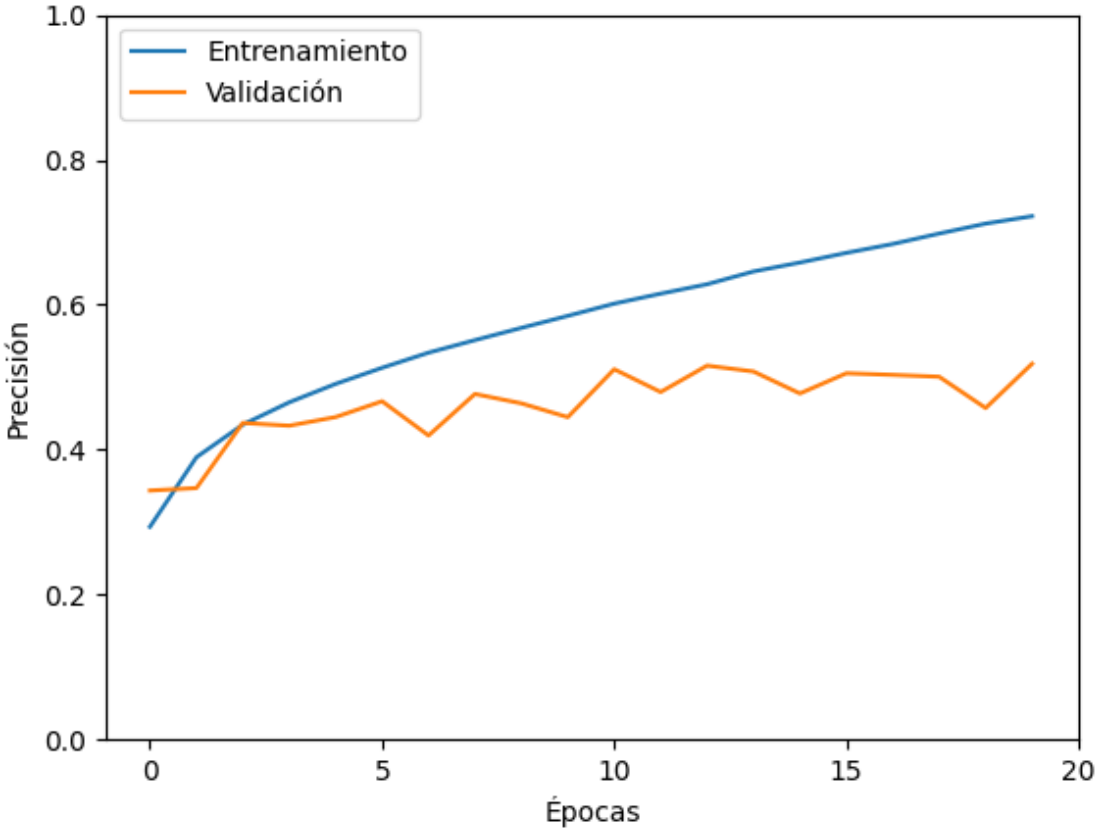

Figura 5.8: Precisión de la configuración final de la red entrenada con la base de datos completa.

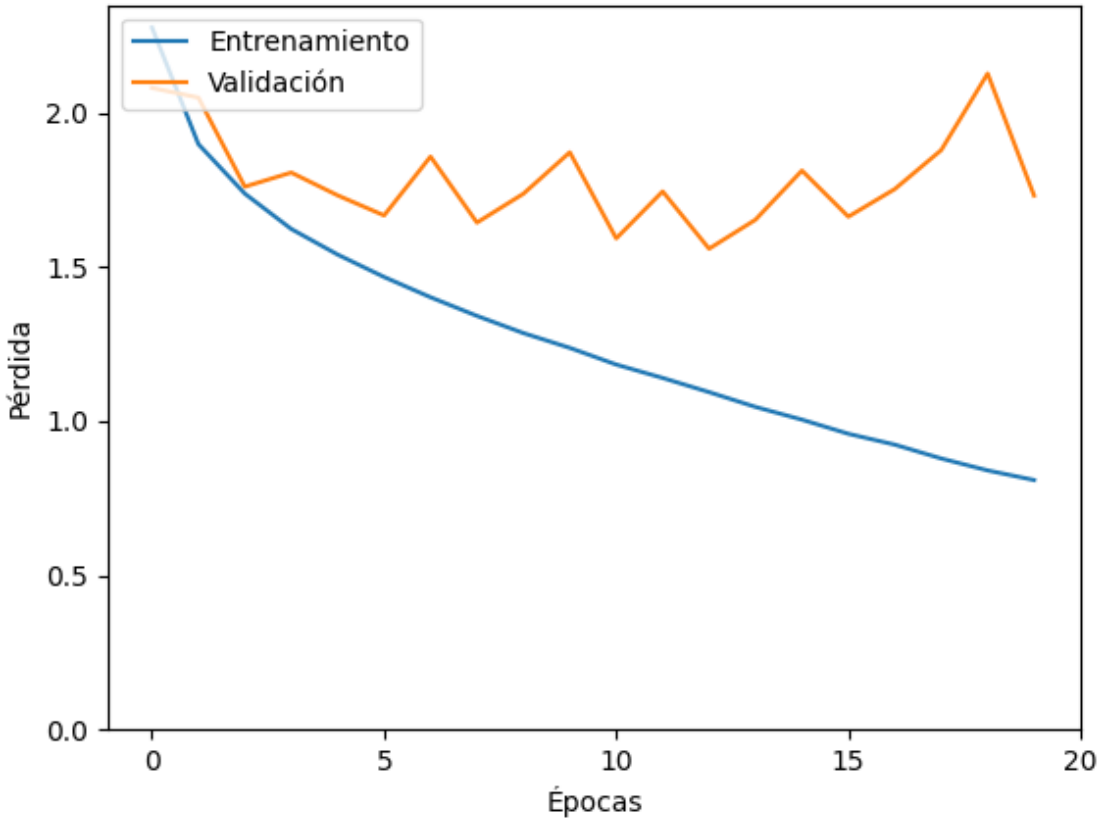

Figura 5.9: Pérdida de la configuración final de la red entrenada con la base de datos completa.

5.2.5. Tipos de clasificación

En este experimento se decide comparar el desempeño entre usar el método de reconstrucción de imágenes a partir de arquetipos con los métodos de distancia a la imagen más cercana y el método de distancia promedio entre todas las imágenes de cada estilo. Es importante recordar que, como el número de arquetipos a usar es igual al número de imágenes con las cuales se arman los arquetipos, los arquetipos son iguales a las imágenes usadas para armar los arquetipos.

Tipo	Precisión
Análisis de arquetipos	50.000 %
Imagen más cercana	34.064 %
Distancia promedio	7.696 %

Tabla 5.16: Comparación entre métricas de distancia.

Como se puede ver en los resultados de la tabla 5.16, el análisis de arquetipos tiene un desempeño bastante superior a tomar la imagen más cercana del conjunto de entrenamiento

de arquetipos, y el desempeño es aun mejor comparado con tomar el promedio de la distancia de las imágenes de cada estilo.

5.3. Resumen de los resultados

Entre los resultados mas relevantes se pueden mencionar aquellos obtenidos por dos técnicas en particular. La primera técnica es el uso de una sola capa convolucional (la primera capa del último bloque de capas convolucionales), en vez de las cinco sugeridas por un trabajo anterior (la primera capa de cada bloque de capas convolucionales), como base para armar los vectores de características. Esta técnica permite una mejora en la precisión de un 37.3 % a un 42.5 % para el método de análisis de arquetipos. La segunda técnica es el aumento de arquetipos, que usa un número fijo de imágenes para hacer la reducción de dimensionalidad, y luego usa el reductor de dimensionalidad configurado por ese conjunto de imágenes para reducir un número de imágenes adicionales considerablemente mayor. Esta técnica permite aumentar el número de arquetipos de 250 a 1000 para este trabajo, lo que se traduce en una mejora en la precisión de un 42.5 % de la técnica anterior a un 50.1 % para el método de análisis de arquetipos. De esta primera técnica se sugiere la posibilidad de que no existe un número fijo de arquetipos ideal para hacer la clasificación, a diferencia de la hipótesis planeada inicialmente, sin embargo, tampoco es posible hacer concluir esto de forma definitiva. Para la segunda técnica es interesante mencionar que se logra superar el desempeño de la red neuronal convolucional cuando ambas técnicas usan la misma cantidad de imágenes de entrenamiento. Sin embargo, esta técnica de análisis de arquetipos no escala tan bien al usar cada vez mas imágenes de entrenamiento debido a limitaciones de la etapa de reducción de dimensionalidad, que requiere una gran cantidad de memoria RAM mientras mayor sea el número de imágenes a utilizar, a diferencia de la red neuronal convolucional que no requiere de tal etapa, por lo que el análisis de arquetipos podría no ser tan útil cuando se trata de bases de datos mas grandes. Aun así, también existe la posibilidad de que exista alguna implementación de software que permita ejecutar esta reducción sin necesidad de agotar la memoria, por lo que podría ser provechoso desarrollar tal implementación alternativa.

Una observación interesante en el experimento de usar las cinco capas, cuando se compara entre reentrenar todas las capas y no reentrenar ninguna, es que el desempeño del análisis de arquetipos baja. Al mismo tiempo, al entrenar una sola capa (en particular, la ultima capa del último bloque), el desempeño del análisis de arquetipos sube. Esto significa que las primeras capas, al ser reentrenadas, pierden poder de representación al momento de hacer el análisis de arquetipos, mientras que las ultimas capas ganan poder de representación. Una posible razón de por que esto ocurre es que estas primeras capas, que son preentrenadas con una base de datos basada en su mayoría en fotografías del mundo real, y que suelen representar mejor características más concretas, podrían tener una dificultad para adaptarse a las peculiares características del mundo artístico.

Un experimento que podría ser interesante para un trabajo posterior, basado en un experimento del trabajo actual, es la elaboración de arquetipos en base a una red neuronal convolucional entrenada desde cero. Si bien se ese experimento se elabora durante este trabajo, sin dar un resultado superior a otras redes, no es posible hacer una conclusión definitiva al

respecto, ya que esa red requiere de un entrenamiento por un periodo de tiempo considerablemente mayor a las otras redes, entre otras razones que dificultan su ejecución. Sin embargo, un experimento así podría tener resultados positivos respecto a las redes anteriores, por lo que su elaboración podría beneficioso, aunque es importante tomar en cuenta el mayor costo en términos de tiempo de todas formas. Una razón de por que esta red podría funcionar mejor se debe a la dificultad que las primeras capas parecen tener para representar características del mundo artístico al ser reentrenadas, como se acaba de mencionar en la observación sobre el experimento de las cinco capas, por lo que entrenar estas capas desde cero podría mejorar la representación de estas características al hacer el análisis de arquetipos.

Otros resultados relevantes incluyen la observación de que usar la primera capa de cada bloque de capas convolucionales (y en particular la primera capa del último bloque) conllevan a un desempeño superior respecto al uso de las ultimas capas de cada bloque, que combinar capas para armar los arquetipos no se traduce en una mejora respecto a usar solamente la primera capa del último bloque, y que usar el análisis de arquetipos para clasificar las imágenes funciona considerablemente mejor que tomando tanto el estilo del vectores de características mas cercano como el de la distancia promedio de los vectores de características, siendo ambos vectores de características aquellos para armar los arquetipos.

Finalmente, en base a los resultados observados de la matriz de confusión del experimento final, se sospecha que el análisis de arquetipos parece tener un desempeño superior a la red neuronal cuando se trata de estilos de menor realismo, lo que puede estar relacionado con la utilización de una única capa del final de la red neuronal como base de los arquetipos, aunque esto no se puede confirmar con seguridad en este trabajo.

Capítulo 6

Conclusión

Para este trabajo se planteó el problema de clasificación y una posible solución para ese problema. Fue necesario investigar sobre trabajos relacionados al problema para así poder contar con una solución general y posibles configuraciones de referencia como punto de partida. También fue necesario encontrar una base de datos apropiada para poder llevar a cabo el análisis de la solución planteada. Una vez ideada la solución general, se implementaron las distintas tecnologías requeridas, apoyado del uso de diversas librerías, partiendo por la implementación de la red neuronal, que luego es usada para convertir las imágenes en vectores de características, seguido por la implementación de algoritmos para el análisis de arquetipos, con los cuales se emplea la categorización. Una vez terminada la configuración inicial, se hicieron diversas iteraciones de los experimentos y las configuraciones para llegar a un desempeño considerablemente superior.

Respecto a los objetivos planteados al principio de este trabajo, se puede mencionar que se pudo implementar todos los componentes requeridos por la solución, desde encontrar la base de datos apropiada hasta la elaboración de la red neuronal y la creación de los arquetipos, así como también evaluar el desempeño de esta solución usando distintas configuraciones, cuyos resultados se usaron para lograr una mejora efectiva en el desempeño del algoritmo. Sumado a lo anterior, se sostiene que durante el transcurso de este trabajo se logró un análisis apropiado del algoritmo a estudiar.

Si bien se creía que era posible usar el método de análisis de arquetipos para mejorar la precisión respecto a la red neuronal convolucional, esto solo se logró al usar una igual cantidad de imágenes de entrenamiento para tanto la red neuronal como el análisis de arquetipos, obteniendo una precisión de un 43 % y un 50 %, respectivamente. Sin embargo, al usar todas las imágenes posibles para las etapas de entrenamiento, y aprovechando todos los recursos disponibles, el desempeño de la red neuronal resultó ser superior al desempeño del análisis de arquetipos, siendo la precisión de estos métodos un 57 % y un 50 %, respectivamente. Esto se debe a que la red neuronal, a diferencia del análisis de arquetipos, no acaba con la memoria RAM disponible para el entrenamiento al usar una cantidad de imágenes considerablemente mayor, mientras que el análisis de arquetipos si lo hace al momento de hacer el paso de reducción de dimensionalidad, por lo que este último método estaba restringido a usar solo 1000 imágenes de entrenamiento por estilo, mientras que la red neuronal podía llegar a

usar entre 13 000 y 1000 imágenes de entrenamiento por estilo. Aun así, se logró mejorar el desempeño del análisis de arquetipos respecto a su configuración inicial, cuya precisión contaba con un 32 %, llegando a una precisión de un 50 % con su configuración final. Los factores más importantes para lograr esta mejora fueron usar solo una de las capas finales de la red para la conversión de la imagen a un vector de características, en vez de utilizar capas a lo largo de toda la red para lograr esto, e inflar el número de arquetipos artificialmente de 250 a 1000. El primer factor significó una mejora de un 34 % a un 42 % de precisión, mientras que el segundo factor permitió una mejora de un 42 % a un 50 % de precisión.

El hecho de que la precisión haya mejorado al usar más arquetipos sugiere que el número ideal de arquetipos para hacer la clasificación sea un número aun mayor, aunque no se puede determinar con seguridad si este número mayor es un número fijo o si el desempeño es mejor a medida que aumenta el número de arquetipos, sin que exista un número fijo ideal.

Respecto al uso de una única capa al final de la red para conformar los arquetipos, el hecho de que estas últimas capas hayan tenido mayor poder de predicción que las primeras capas después del reentrenamiento podría sugerir que este reentrenamiento afecta a las primeras capas de forma negativa, mientras que las últimas se ven afectadas de forma positiva. Esto podría deberse a que las primeras capas suelen representar mejor características más concretas, pero como la red usada para el reenetrenamiento fue preentreada con una base de datos basada en su mayoría en fotografías del mundo real, estas capas podrían tener una dificultad para adaptarse a las peculiares características del mundo artístico.

Se sugirió que se podría obtener un desempeño superior utilizando una red entrenada desde cero, es decir, sin reentrenarla, para así poder aprovechar una mayor cantidad de capas, dado el hecho de que el reentrenamiento pareció afectar de forma negativa a las primeras capas. Sin embargo, el desempeño de esta red fue considerablemente inferior al desempeño de la red reentrenada, aunque esto se puede deber al hecho de que la red necesitaba un número considerablemente mayor de épocas de entrenamiento respecto a la red reentrenada para poder alcanzar un desempeño similar.

Otra sospecha relevante respecto al desempeño de la red, aunque no confirmado, es que el análisis de arquetipos parece tener un desempeño superior a la red neuronal cuando se trata de estilos de menor realismo, lo que puede estar relacionado con la utilización de una única capa del final de la red neuronal como base de los arquetipos.

Uno de los aspectos relevantes de este trabajo consiste en que se trata de una implementación del análisis de arquetipos para la clasificación de estilos artísticos, método que no fue encontrado en trabajos anteriores. Se logró también encontrar una caracterización para las imágenes que se tradujo en una mejora significativa respecto a la caracterización sugerida en otros trabajos, y además se ideó un método para inflar artificialmente la cantidad de arquetipos a utilizar cuando no hay suficiente memoria RAM para hacer la reducción de dimensionalidad, lo que llevó a otra mejora considerable en el desempeño del algoritmo. Estas técnicas, junto a los distintos experimentos realizados, pueden ser de bastante utilidad para futuros proyectos que busquen explorar la técnica presentada. Sin embargo, es importante también mencionar que este método ha sido bastante costoso en cuanto a tiempo y RAM se refiere. Cuando se trata de tiempo, al menos el 90 % de las 100 horas de ejecución fueron utilizadas en el entrenamiento de la red neuronal, para lo cual el hardware más relevante

fue la GPU GeForce RTX 2080 Ti de 12 GB. En cuanto a la memoria, fue necesario usar casi toda la RAM de 132 GB para poder armar los arquetipos, debido al elevado consumo de RAM en la reducción de dimensionalidad hecha a los vectores de características de las imágenes.

Durante la elaboración de este trabajo fue posible aprender una variedad de lecciones respecto al trabajo actual, lecciones que en retrospectiva hubieran sido de utilidad al momento de elaborar este trabajo. La lección más importante es posiblemente la importancia de definir con anticipación un número acotado de experimentos sobre el cual se debe basar este trabajo, definidos en base a posibles configuraciones y variables a utilizar. Esto se debe a que durante el experimento actual los experimentos a tratar fueron definidos durante el transcurso del trabajo en base a experimentos anteriores, sin contar con un plan anterior que defina que experimentos realizar, lo que se traduce en un flujo constante de nuevos posibles experimentos a medida que se desarrolla el trabajo sin tener una meta fija de cuantos experimentos son suficientes para el trabajo. Otra lección importante consiste en la importancia de estudiar el estado del arte con mayor detención antes de implementar el software y ejecutar los experimentos, ya que en diversas oportunidades se perdió tiempo durante estas etapas por no tener claridad de las tecnologías y conceptos a utilizar, especialmente considerando lo útiles que fueron las sugerencias ofrecidas por estos trabajos anteriores para la elaboración de este trabajo. Como lección final, se menciona que, si bien la falta de planificación para los experimentos resulta en un desarrollo sin una meta clara, este método tuvo una consecuencia positiva de todas formas, ya que la idea de usar una capa particular para armar los vectores de características es una idea que surge a medida que se hicieron los experimentos en base a los resultados obtenidos, por lo que esta idea no se hubiera desarrollado al usar un plan de experimentos predeterminados. En retrospectiva, y en base a lo ya mencionado, un plan de desarrollo ideal hubiera consistido en una etapa estandarizada con experimentos predeterminados, seguido de una etapa de exploración con experimentos propuestos en base a los resultados de los experimentos de la etapa anterior.

Un trabajo futuro podría intentar una implementación de la técnica presentada actualmente usando una red neuronal entrenada desde cero, pero entrenado por una cantidad de épocas considerablemente mayor a la presentada en este trabajo, ya que se cree que una implementación de este tipo podría lograr una mejor representación de características en las primeras capas de la red, cuya representación que empeoró al momento de reentrenar la red preentrenada utilizada. También se podría tomar inspiración al combinar este método con el método presentado por Chen y Yang [2], que utiliza una red neuronal convolucional con *adaptive cross-layer correlation* que permite mejorar la representación de estilo, logrando así una precisión de 78.13 % entre 25 estilos arquitectónicos distintos. Adicionalmente, el desarrollo de una implementación de reducción de dimensionalidad que no requiera de un uso exacerbado de la memoria RAM podría ser de gran utilidad para superar algunas de las limitaciones mencionadas en este trabajo.

Bibliografía

[1] Christian Bauckhage. Numpy / scipy recipes for data science: Archetypal analysis via frank-wolfe optimization. 2020.

[2] Liyi Chen and Jufeng Yang. Recognizing the style of visual arts via adaptive cross-layer correlation. In *ACM International Conference on Multimedia*, 2019.

[3] Yuansi Chen, J. Mairal, and Z. Harchaoui. Fast and robust archetypal analysis for representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[4] Dami Choi, Christopher J. Shallue, Zachary Nado, Jaehoon Lee, Chris J. Maddison, and George E. Dahl. On empirical comparisons of optimizers for deep learning. 2020.

[5] Adele Cutler and Leo Breiman. Archetypal analysis. In *Technometrics Vol 36 No 4*, 1994.

[6] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. 2015.

[7] ImageNet. https://www.image-net.org/. 2021.

[8] Ioannis Karatzas. and Steven E. Shreve. *Brownian Motion and Stochastic Calculus*. Springer, Berlin, 2nd edition, 2000.

[9] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2014.

[10] Adrian Lecoutre, Benjamin Negrevergne, and Florian Yger. Recognizing art style automatically in painting with deep learning. In Min-Ling Zhang and Yung-Kyun Noh, editors, *Proceedings of the Ninth Asian Conference on Machine Learning*, volume 77 of *Proceedings of Machine Learning Research*, pages 327–342. PMLR, 2017.

[11] lucasdavid. Wikiart retriever, https://github.com/lucasdavid/wikiart/. 2018.

[12] Julien Mairal, Francis R. Bach, and Jean Ponce. Sparse modeling for image and vision processing. In *Foundations and Trends in Computer Graphics and Vision*, 2014.

[13] Philip Protter. *Stochastic Integration and Differential Equations*. Springer, 1990.

[14] Daniel Revuz and Marc Yor. *Continuous martingales and Brownian motion*. Number 293 in Grundlehren der mathematischen Wissenschaften. Springer, Berlin [u.a.], 3. ed edition, 1999.

[15] Matthia Sabatelli, Mike Kestemont, Walter Daelemans, and Pierre Geurts. Deep transfer learning for art classification problems. 2018.

[16] Babak Saleh and Ahmed Elgammal. Large-scale classification of fine-art paintings: Learning the right metric on the right feature. In *arXiv preprint, abs/1505.00855*, 2015.

[17] scikit-learn. Truncated singular value decomposition and latent semantic analysis. 2021.

[18] Daniel Silver and Kristin Bennett. Guest editor's introduction: Special issue on inductive transfer learning. volume 73, pages 215–220, 2008.

[19] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations*, 2015.

[20] Karen Simonyan and Andrew Zisserman. Universal style transfer via feature transforms. In *Advances in Neural Information Processing Systems*, 2017.

[21] G. W. Stewart. On the early history of the singular value decomposition. 1992.

[22] Debnil Sur and Ellen Blaine. Cross-depiction transfer learning for art classification. In *Technical report for CS 231A and CS 231N at Stanford*, pages 4321–4329, 2017.

[23] VGG16 and VGG19. https://keras.io/api/applications/vgg/. 2021.

[24] Nitin Viswanathan. Artist identification with convolutional neural networks. In *Technical report*. Stanford University, 2017.

[25] WikiArt. https://www.wikiart.org/en/about. 2020.

[26] Daan Wynen, Cordelia Schmid, and Julien Mairal. Unsupervised learning of artistic styles with archetypal style analysis. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, 2018.

[27] Sean Yang, Bum Mook Oh, Daniel Merchant, Bill Howe, and Jevin West. Classifying digitized art type and time period. In *Proceedings of International Journal for Digital Art History*, 2018.