UACJ INVESTIGACIÓN

Título del Proyecto de Investigación
al que corresponde el Reporte Técnico:

Algoritmo de clasificación en un ambiente distribuido para la identificación de enfermedades en imágenes médicas

Tipo de financiamiento

Sin financiamiento

Fecha de Inicio: 06/01/2020
Fecha de Término: 30/06/2022

Tipo de Reporte

[ ] Parcial

[X] Final

Autor (es) del reporte técnico:

Rogelio Florencia Juárez
Gilberto Rivera Zárate
Julia Patricia Sánchez Solís
Francisco López Orozco
Vicente García Jiménez

Algoritmo de clasificación en un ambiente distribuido para la identificación de enfermedades en imágenes médicas

Resumen del reporte técnico en español (máximo 250 palabras)

En este trabajo se aborda el problema de clasificación de estilos fotográficos con el uso de redes neuronales convolucionales, el cual está enfocado a analizar la parte estética de la imagen (estilo) y no a su contenido. Dicho problema ya ha sido abordado en trabajos previos, sin embargo, los resultados no son tan favorables. En el presente, se estudian distintas arquitecturas y configuraciones de redes convolucionales con el fin de mejorar los resultados existentes y/o encontrar las posibles causas del bajo desempeño de los algoritmos.

Primeramente, se recolectó un *dataset* que permitiera trabajar con aspectos estilísticos de la imagen. Posteriormente, se probaron distintas arquitecturas de redes convolucionales para seleccionar aquella que diera un mejor resultado. Una vez hecho lo anterior, se probaron distintas técnicas para mejorar el desempeño del modelo seleccionado como: *transfer learning*, *data augmentation*, y *dropout*. Finalmente, se muestran los resultados obtenidos de entrenar las distintas arquitecturas, la comparación con los trabajos previos y ciertas observaciones que sugieren algunas de las causas que impiden que el modelo mejore su desempeño.

Resumen del reporte técnico en inglés (máximo 250 palabras):

This work addresses the problem of classifying photographic styles using convolutional neural networks, which focus on analysing the aesthetic part of the image (style) and not its content. This problem has already been addressed in previous works; however, the results are unfavourable. In this work, different configurations of convolutional networks are studied to improve the existing results and/or find the possible causes of the low performance of the algorithms.

First, a dataset was collected that would allow working with stylistic aspects of the image. Subsequently, different convolutional networks architectures were tested to select the one that gave the best result. Once the above was done, different techniques were tested to improve the performance of the selected model, such as transfer learning, data augmentation, and dropout. Finally, the results obtained from training the different architectures are shown. The comparison with previous works and specific observations suggests some causes that prevent the model from improving its performance.

Palabras clave: redes neuronales convolucionales, transfer learning, keras, estilo, imagen.

Usuarios potenciales (del proyecto de investigación)

Fotógrafos profesionales.

Reconocimientos

Agradecimiento a la División Multidisciplinaria en Ciudad Universitaria de la Universidad Autónoma de Ciudad Juárez, al alumno Andre Martin Vera Valdez con matrícula 154007 por su esfuerzo y dedicación en culminar el proyecto

1. Introducción

Normalmente, el contenido de la imagen puede ser considerado su aspecto más importante, pues sin contenido no habría imagen. Sin embargo, el aspecto estético (estilo) se ha estado estudiando en los últimos años en el área de inteligencia artificial mediante la clasificación de estilos con algoritmos de *machine learning* [1], [2], [3]. Sin embargo, en los trabajos antes mencionados se muestran resultados que no superan el 58.1% en sus métricas de validación.

Lo mencionado anteriormente presenta un área de oportunidad debido a que los resultados presentados no son precisamente buenos. Por esta razón, en este proyecto se propone una red neuronal convolucional que mejora los resultados obtenidos en algunos de los trabajos previos. El proceso de desarrollo de la red neuronal, que fue dictado por la metodología propuesta en [4], permitió que se encontraran algunas de las causas que dan

origen al bajo desempeño de los algoritmos utilizados para la clasificación de estilos fotográficos; todos ellos relacionados con los datos.

2. Planteamiento

2.1 Antecedentes

La fotografía es una actividad que prácticamente forma parte de nuestro día a día; principalmente por el fácil acceso que se tiene a una cámara. Otra de las razones de que esto sea así, es que tomar una foto es una tarea muy sencilla, solo tienes que poner un sujeto delante de la cámara y presionar el botón de disparar.

Una vez establecido lo anterior, se puede decir que el sujeto (o sujetos) es el elemento principal de una fotografía, es el objeto que está dentro del marco, por ejemplo: una persona, un auto, una planta, un animal, etc. El sujeto es el motivo por el cual se toma una foto [5].

Otro elemento presente en una fotografía es el estilo, y es el que interesa en la presente investigación. El estilo es la forma en la que se toma a la fotografía, si el sujeto es el "qué", el estilo es el "cómo". Es el resultado de ciertas decisiones técnicas propias del fotógrafo en cuanto a la composición de la imagen, la distancia focal, el tiempo de exposición, y la iluminación [5].

Una de las distintas aplicaciones de las técnicas de *machine learning* es el reconocimiento de imágenes [6]. Una vez definidos los elementos que componen a una fotografía (sujeto y estilo), se puede suponer que es posible que se puedan aplicar técnicas de *machine learning* para el reconocimiento tanto de sujetos como de estilos. A continuación, se presentan algunos trabajos relacionados con ello.

El primer paso para realizar un modelo de clasificación es recolectar los datos, en este caso, las imágenes. El *dataset Aesthetic Visual Analysis* (AVA) [1], es un conjunto de fotografías recolectadas del sitio *dpchallenge.com*¸ contiene alrededor de 250,000 imágenes, las cuales tienen anotaciones relacionadas con los siguientes aspectos:

- **Calificación de estética:** Es una calificación dada por los usuarios del sitio web de donde se obtuvieron las imágenes, esto con el fin de entrenar clasificadores que

sean capaces de predecir lo buenas o malas que son las fotografías en base a una nota numérica.

- **Anotaciones semánticas:** Son etiquetas que describen el contenido de las imágenes. Estas pueden ser utilizadas para realizar reconocimiento de sujetos dentro de las fotografías, algunas de las etiquetas son; naturaleza, arquitectura, y flora.
- **Anotaciones de estilo:** Son una serie de 14 estilos asociados a un subconjunto de imágenes dentro del *dataset*, algunos de ellos son; larga exposición, macro, y regla de tercios.

Además de haber hecho la recolección de imágenes en [1], se presentaron algunos métodos de clasificación para cada una de las características mencionadas anteriormente. Para la clasificación de imágenes por estilo, se utilizó el método *Support Vector Machine* y se obtuvo un *Mean Average Precision* (mAP) del 53.85%.

Más tarde en [2], se utilizó el *dataset* AVA para realizar también clasificación de estilos con un algoritmo de clasificación lineal y mediante métodos de extracción de características como *color histogram* o *GIST*, siendo el método más efectivo el de una *red neuronal convolucional* (CNN por sus siglas en inglés) con un mAP del 57.9% y logrando con la con la fusión de múltiples características un mAP del 58.1%. Además, en este trabajo se propuso otro *dataset* basado en imágenes obtenidas de la red social Flickr, siendo los resultados no tan satisfactorios como los de AVA, logrando apenas un mAP del 33.6%.

El reconocimiento de estilo no está limitado solo al área de la fotografía, también puede ser aplicado a otro tipo de imágenes como lo pueden ser pinturas. En el trabajo presentado por [3], se desarrolla una CNN para la clasificación de estilos en pinturas, los estilos clasificados por la red son: impresionismo, expresionismo, posimpresionismo, surrealismo, simbolismo, cubismo, y abstracción pospictórica. La efectividad obtenida de la red para este caso es del 51%.

Como se puede observar en los trabajos previos, la precisión de los modelos de clasificación de estilos no es precisamente la mejor. Por esta razón, es necesario

identificar otras técnicas, o configuraciones de las mismas, con el fin de ir mejorando los resultados existentes.

2.2 Marco teórico

2.2.1 Fotografía

Según [7], la fotografía se define como, "un acto a través del cual se produce la grabación de una situación luminosa, en un lugar y momento determinado: es la huella de la acción de la luz". La definición indica que el elemento esencial para realizar una fotografía es la luz, sin ella no hay fotografía.

Como lo menciona Freeman [5], uno de los elementos que están presentes dentro de cualquier fotografía es el estilo. En pocas palabras, el estilo es definido como la manera en la que se toma una fotografía, por supuesto, esto afecta al aspecto visual de la imagen. Además, en otro de los libros de Freeman [8], se menciona que el estilo puede ser intencionado o no intencionado.

Figura 1: Fotografía con *motion blur*

En la Figura 1 se muestra una foto que está movida, lo cual le da cierta apariencia "borrosa", este estilo es el denominado *motion blur*. Esta foto fue tomada durante la Segunda Guerra Mundial, este movimiento que se aprecia en la foto es ocasionado por las circunstancias de la situación, por lo cual, se puede decir que el estilo no necesariamente fue intencionado [8]. El hecho de que el estilo pueda ser no intencionado resulta interesante, pues hoy en día, que la fotografía está al alcance de cualquier persona, es muy probable que existan muchas fotos hechas por personas que no son fotógrafos profesionales y que simplemente estén "perdidas" en la base de datos de algún sitio web.

A continuación, se definen algunos estilos fotográficos presentados en [5]:

- *Motion blur:* Es cuando en la foto se puede apreciar el movimiento del sujeto y esto genera un desenfoque en ella.
- *Shallow DOF:* Esta técnica consiste en hacer que cierta parte de la fotografía se encuentre desenfocada.
- *High contrast:* Se refiere a cuando existe una diferencia muy grande entre las zonas más oscuras y las más claras de la imagen.
- *Vanishing point:* Cuando en una foto convergen dos o más líneas paralelas a un mismo punto.

En la Figura 2 se muestran ejemplos de los diferentes estilos mencionados anteriormente.

Figura 2: Motion blur, Shallow DOF, High contrast y Vanishing point

2.2.2 Machine Learning

En la actualidad el *Machine Learning* (en español, Aprendizaje Automático) forma parte de la vida cotidiana de la mayoría de las personas, algunas pruebas de ello son: la forma en la que un servicio de correo electrónico clasifica el spam, las sugerencias personalizadas que utilizan plataformas como Netflix para hacer recomendaciones, o el orden en el que Facebook decide mostrar las notificaciones a los usuarios. Lo anterior quiere decir que es muy probable que el *Machine Learning* esté involucrado de alguna manera cada vez que alguien utiliza una computadora o dispositivo móvil [9].

El término *Machine Learning* se refiere a un conjunto de algoritmos que permiten a una computadora resolver una tarea por su propia cuenta, a través de inferencias que esta hace sobre un conjunto de datos. Al utilizar un algoritmo de este tipo se pretende que una computadora pueda llegar a la solución de un problema por su propia cuenta, es decir, que el algoritmo dé con dicha solución sin ser programado explícitamente [9].

Existen diferentes tipos de aprendizaje que pueden ser implementados mediante *Machine Learning*, algunos de los más relevantes se muestran a continuación.

- *Aprendizaje supervisado:* Es el tipo de algoritmo que se utiliza cuando se conoce el resultado al que se quiere llegar, es decir, la respuesta que se espera del algoritmo es conocida. Esto se logra mediante un conjunto de datos previamente etiquetado [10].
- *Aprendizaje no supervisado:* Es cuando el algoritmo no conoce el resultado al que tiene que llegar, por lo tanto, este se encargará de encontrar similitudes entre el conjunto de datos. Es por esto que este tipo de algoritmos son usados para agrupar datos [10].
- *Aprendizaje por refuerzo:* En este tipo de aprendizaje se utiliza un sistema de recompensas, después de una serie de acciones el algoritmo recibe una recompensa positiva o negativa sobre la cual aprende. Una analogía que ejemplifica esto, es la forma de entrenar a un animal para realizar cierta tarea [10].

Deep Learning

El *Deep Learning* (Aprendizaje Profundo, en español) es un área del *Machine Learning* que se utiliza redes neuronales de múltiples capas para resolver tareas que requieren la entrada de cantidades muy grandes de datos. Algunas sus aplicaciones más comunes son: reconocimiento de voz, procesamiento de lenguaje natural, visión artificial, entre otras [11].

2.2.3 Redes Neuronales Artificiales

Las *Redes Neuronales Artificiales* (RNA) son modelo de mapeo no lineal, que están inspirados en el funcionamiento de las neuronas biológicas de los seres humanos, de tal manera que el propósito de una RNA es poder realizar tareas que las redes neuronales biológicas pueden hacer [12], por ejemplo, reconocimiento de voz o de imágenes.

Según Vlad [12], una red neuronal está compuesta por un conjunto de entradas, un proceso interno y un conjunto de salidas. Las entradas son los datos “crudos” que procesará la red. El proceso interno está determinado por la arquitectura de la red y lo que hace es mapear las entradas hacia la salida de la red. Por último, la salida es el resultado de la red, es la inferencia que esta hizo sobre los datos que se le pasaron como entrada. La Figura 3 muestra la representación gráfica de una red neuronal.

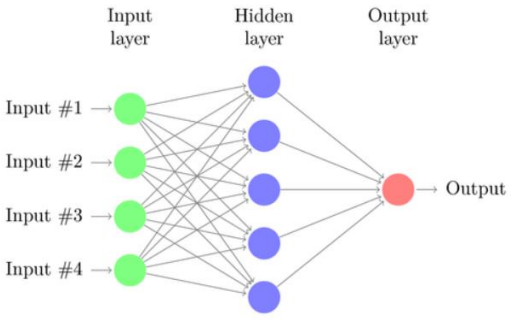


Figura 3: Arquitectura de una red neuronal

La neurona

Para comprender mejor a las RNA es necesario entender su unidad más simple, la neurona, que también es conocida como nodo. Una neurona recibe *entradas* (x) que son procesadas según la suma de sus *pesos* (w), esto es llamado *activación*, posteriormente la neurona pasa el valor obtenido a una *función de activación* para producir un valor de salida [12], como se muestra en la Figura 4.

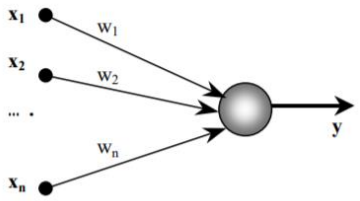


Figura 4: Neurona

La activación *A* está dada por la siguiente sumatoria:

$$A = \sum_{i=1}^{n} x_i w_i + \theta$$

Donde x_i son los valores de entrada, w_i son los pesos correspondientes y θ es un valor de *bias*, este es asimilado como el peso de la conexión a una neurona que siempre produce un valor de salida igual a 1. Por otra parte, la función de activación depende de la arquitectura de la red neuronal que se esté implementado, esta función permite introducir la *no linealidad* a la salida de la neurona. Algunas de las funciones de activación más comunes son: *Sigmoid*, *Softmax*, y *ReLU* [12].

2.2.4 Redes Neuronales Convolucionales

Las redes neuronales convolucionales son un tipo de RNA que permiten procesar datos que son de naturaleza de "vector" [13]. Por ejemplo, una imagen, que puede ser representada como un vector de pixeles de dos dimensiones. La *convolución* hace referencia a una operación matemática que se utiliza para procesar los datos de entrada, la cual se explica más adelante.

Si bien es posible utilizar las redes convolucionales para procesar datos que no sean imágenes, debido a la naturaleza de este proyecto, los siguientes conceptos serán abordados bajo la premisa de que las CNN se utilizaran para procesar imágenes.

Capa de entrada

Como toda red neuronal, lo primero que se necesitan son datos de entrada, en este caso la entrada sería una imagen que puede ser representada como una matriz de pixeles de dos dimensiones, si la imagen es a blanco y negro, o 3 matrices si es a color (Figura 5), uno para cada canal de color (RGB) [14].

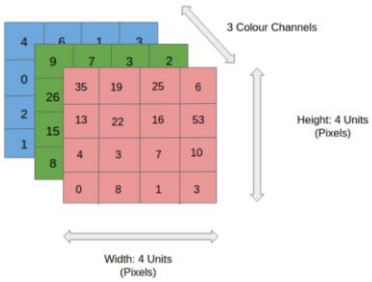


Figura 5: Representación de los tres canales de color de una imagen

Capa de convolución

Estas son las capas ocultas de la red, y son las encargadas de realizar la operación de convolución. Para realizar dicha operación se requiere de un *filtro* o *kernel*, este no es más que una matriz de menor tamaño que la de entrada y que será inicializada con ciertos valores, el filtro será el encargado de identificar características en la imagen. Por ejemplo, si se tiene un filtro de 3×3, para realizar la convolución este recorrerá cada bloque de 3×3 de la matriz de entrada y realizará un producto punto con dicho bloque, el valor resultante será almacenado en una nueva matriz. El proceso anterior debe continuar hasta que se

haya terminado de procesar cada bloque de 3×3 de la matriz de entrada, al finalizar, se obtendrá una nueva matriz más pequeña que la original, este es el resultado de la convolución [15]. En la Figura 6 se aprecia la imagen de entrada, el filtro y el resultado de la convolución.

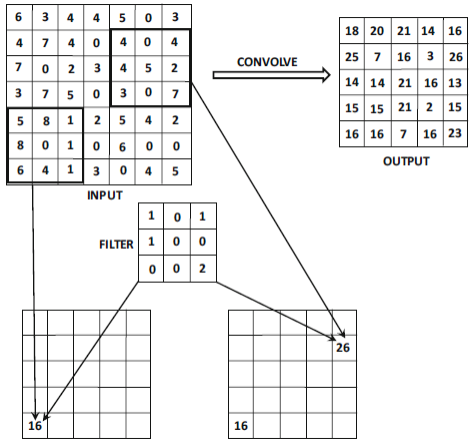


Figura 6: Operación de convolución

Algunas consideraciones que se deben tener es que una capa de convolución puede tener más de un filtro, a esto se le llama *profundidad*. Además, a cada filtro se le puede asignar un paso, esto es el número de saltos que hará cada vez que recorra la matriz de entrada. Cada vez que se hace una convolución la imagen se reduce en tamaño, sin embargo, es posible agregar cierto *padding* (Figura 7) a la imagen, el cual se refiere a agregar columnas y filas extras rellenadas con ceros para así mantener el tamaño original de la imagen [14].

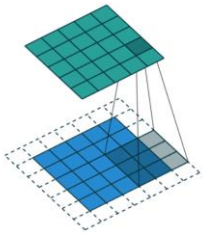


Figura 7: Convolución con *padding*

Capa de pooling

En esta capa se realiza una operación de *pooling* (o agrupamiento, en español) que se encarga de reducir más la imagen, para disminuir la potencia de cómputo necesaria para seguir procesando la imagen. Además, con el agrupamiento también se asegura que las

características obtenidas no estarán ligadas a una posición especifica dentro de la imagen [14].

Existen dos tipos de agrupación, *Max Pooling* y *Average Pooling*, el primero regresa el valor máximo de la parte que fue cubierta por el filtro sobre la imagen original, mientras que el segundo regresa el valor promedio. El tipo de agrupamiento más comúnmente utilizado es el *Max Pooling*, pues además de reducir la imagen, permite descartar activaciones ruidosas dentro de la red [14]. Los distintos tipos de *pooling* son representados en la Figura 8.

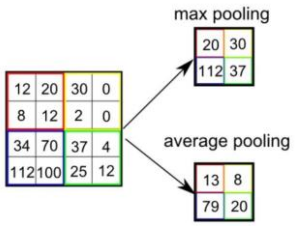


Figura 8: Tipos de *pooling*

Capa completamente conectada

Esta es la capa final de la red y consiste en un aplanamiento de las matrices que resultaron de las capas anteriores, lo cual quiere decir que los datos ahora estarán en un solo vector donde cada elemento será una entrada de una red neuronal completamente conectada que mediante una función de activación producirá los valores de salida [15].

3. Objetivos (general y específicos)

Objetivo general:

Desarrollar una red neuronal convolucional para la clasificación de estilos fotográficos.

Objetivos específicos:

- Definir las funcionalidades y resultados esperados de la red neuronal convolucional.
- Recolectar un *dataset* con el que sea posible trabajar con estilos fotográficos.
- Analizar el estado del arte actual sobre los algoritmos utilizados para clasificar estilos fotográficos.

- Diseñar y entrenar una red neuronal convolucional que permita obtener los mejores resultados posibles.
- Evaluar el desempeño de la red utilizando métricas establecidas en la literatura y comparar los resultados con los del estado del arte.

4. Metodología

Para lograr los objetivos se implementó una red neuronal convolucional que fue entrenada para clasificar estilos fotográficos.

Primeramente fue necesario recolectar los datos, es decir, un conjunto de imágenes con las que se pueda trabajar. Para esto se usó el *dataset* AVA [1], que es un conjunto de datos establecido en la literatura, este fue usado para entrenar la red. Dicho *dataset* cuenta con un subconjunto de fotografías que están etiquetadas por estilo, estos fueron los datos con los que se trabajó.

Una vez que se obtuvieron los datos, se diseñaron y probaron distintas configuraciones para lograr el mejor desempeño posible de la CNN, esta recibe como entrada una imagen y produce como salida la clasificación de su estilo, en la Figura 9 se muestra el flujo de datos de la aplicación propuesta. Para la implementación de la red se utilizó el lenguaje de programación *Python* con la librería *Keras* y *TensorFlow* como *backend* de esta.

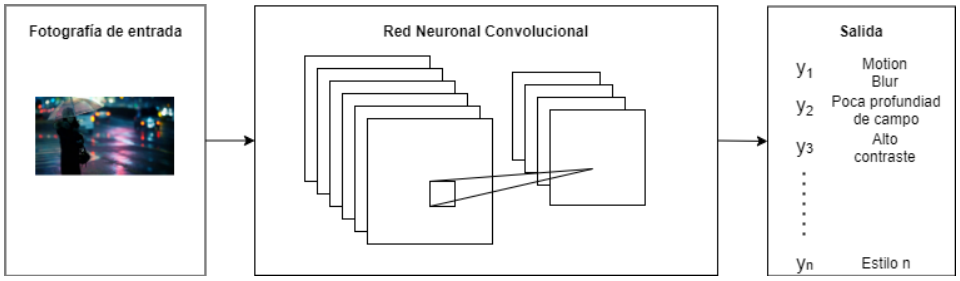


Figura 9: Bosquejo de la solución propuesta

Debido a la naturaleza experimental de los proyectos basados en *Machine Learning*, puede ser difícil hacerlos encajar en una de las metodologías tradicionales o ágiles de desarrollo de software. Por esta razón, para el desarrollo del presente proyecto se utilizó la metodología denominada *The nested loop model of the neural network development process*, propuesta por David Rodvold [4], que fue diseñada para trabajar con proyectos

de redes neuronales (Figura 10). En esta sección se detalla el trabajo realizado en cada una de las etapas de la metodología.

4.1 Requerimientos de la red, metas y restricciones

En esta fase de la metodología se definen las especificaciones de la red tales como; la salida que se espera obtener, la precisión deseada del algoritmo y algunas otras restricciones como lo pueden ser el lenguaje de programación, el tiempo de ejecución, el espacio en memoria, entre otras [4].

A continuación se presentan las especificaciones de la red neuronal convolucional:

- Para el entrenamiento de la red se utilizó el *dataset* AVA [1].
- La red neuronal fue capaz de clasificar los 14 distintos estilos fotográficos.
- Cada imagen solo es clasificada dentro de un estilo fotográfico.
    - Para evaluar el desempeño de la red se utilizó la métrica, y se superó el resultado del trabajo en [2], que fue del 58.1%.
- Para la implementación se utilizó el lenguaje de programación *Python* y la librería *Keras* y *Tensorflow* como su *backend*, por medio del servicio en la nube de *Google Colab*.

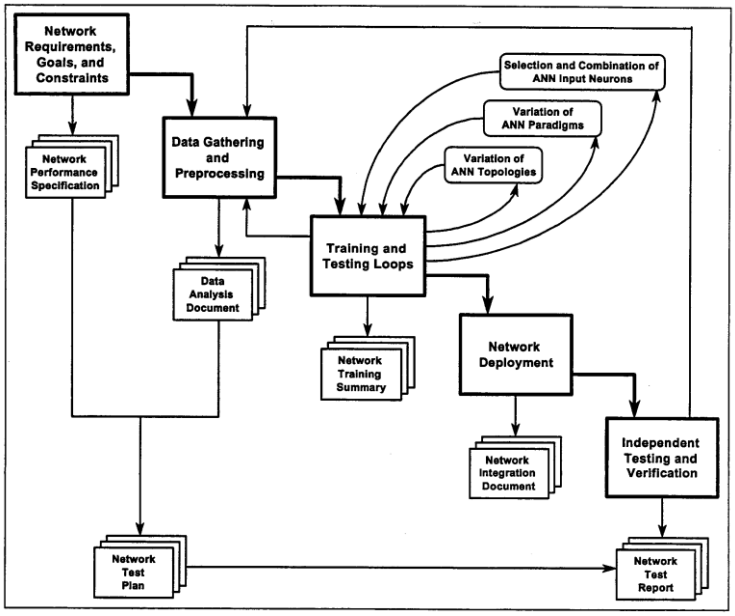


Figura 10: Representación gráfica de la metodología

4.2 Recolección de los datos y preprocesamiento

Esta es la etapa en la que se recolectan los datos con los que se van a trabajar. El objetivo es especificar cualquier información relevante a la naturaleza de los datos como: fuentes de obtención, formato, tamaño, etc. [4].

Descripción de los datos

El conjunto de datos utilizado fue el denominado AVA [1]. Algunas de sus características son las siguientes:

- El *dataset* es multipropósito y cuenta con 255,510 imágenes en formato *JPG*.
- Las imágenes fueron recolectadas del sitio *dpchallenge.com* que es una web especializada en concursos de fotografía.
- El total de imágenes que se contemplan en el *dataset* AVA para la clasificación de estilos son 14,079.
- Los 14 estilos fotográficos que contempla son: *complementary colors*, *duotones*, *hdr*, *image grain*, *light on white*, *long exposure*, *macro*, *motion blur*, *negative image*, *rule of thirds*, *shallow DOF*, *silhouttes*, *soft focus*, y *vanishing point*.

Obtención de los datos y preprocesamiento

Una vez descargado el *dataset* de [19], se separaron por estilos las imágenes correspondientes, pues el conjunto no estaba separado. Para hacer esto se utilizaron dos archivos de texto, mismos que venían incluidos como archivos adicionales a las imágenes. El primer archivo contenía el nombre de la imagen y el segundo el estilo correspondiente, para hacer la separación de las imágenes se realizó un *script* de *Python*.

Si bien el trabajo presentado en [1] describe un total de 14,079 imágenes etiquetadas por estilos, para el desarrollo del presente proyecto solo se utilizaron solo 11,253. Lo anterior debido a que en el conjunto de *train* (80% del total) cada imagen pertenecía solo a un estilo pero en el conjunto de *test* (20% del total) las imágenes estaban etiquetadas con más de un estilo.

Finalmente, una vez que se separaron las imágenes por estilo se crearon las carpetas *train*, *test*, y *valid*. Para el conjunto de entrenamiento se contemplaron 80% de las 11,253 imágenes mientras que para los conjuntos de pruebas y validación el 10% para cada uno. Para realizar esta tarea se utilizó otro *script* de *Python*. La cantidad de imágenes por clase y por conjunto se muestran en la Tabla 1.

Para preparar las imágenes antes de mandarlas a la CNN los pixeles fueron normalizados para tener valores entre 0 y 1, en lugar de 0 y 255. Además, las imágenes fueron redimensionadas a una medida de 256×256 pixeles. Para lograr esto, se utilizó el módulo *ImageDataGenerator* de *Keras* [20], como se muestra la Figura 11.

Con el fin de alterar el estilo de la imagen lo menos posible, no se utilizó ningún otro tipo de preprocesamiento, ya que esto podría generar ruido al clasificador.

Tabla 1: Cantidad de imágenes por estilo

Estilo	Train	Test	Valid
Complementary colors	608	76	76
Duotones	832	104	104
HDR	253	31	31
Image grain	537	67	67
Light on white	768	96	96
Long exposure	540	67	67
Macro	1087	135	135
Motion blur	390	48	48
Negative image	614	76	76
Rule of thirds	889	111	111
Shallow DOF	947	118	118
Silhouettes	660	82	82
Soft focus	454	56	56
Vanishing point	432	54	54

```
train_batches = ImageDataGenerator(rescale=1.0/255.0) \
                .flow_from_directory(directory=train_path, target_size=(256,256), batch_size=16)
valid_batches = ImageDataGenerator(rescale=1.0/255.0) \
                .flow_from_directory(directory=valid_path, target_size=(256,256), batch_size=16)
test_batches = ImageDataGenerator(rescale=1.0/255.0) \
                .flow_from_directory(directory=test_path, target_size=(256,256), batch_size=100, shuffle=False)
```

```
Found 9011 images belonging to 14 classes.
Found 1121 images belonging to 14 classes.
Found 1121 images belonging to 14 classes.
```

Figura 11: Objetos para crear los lotes de imágenes para entrenar

4.3 Ciclos de entrenamiento y prueba

Es la etapa media de la metodología y una de las más importantes. Es un proceso iterativo en el que se tienen que probar distintas configuraciones de la red para llegar al resultado deseado. En cada iteración de estos ciclos se pueden hacer cambios a la configuración de la red para gradualmente acercarse a los resultados deseados [4].

CNN a la medida

La primera arquitectura de CNN que se utilizó fue una diseñada a la medida. Para esto se utilizaron algunos de los principios de diseño como los presentados en [21] y [22]. Estos principios pueden ser: crecimiento gradual de la red, comenzar con filtros pequeños, utilizar filtros de tamaño impar, entre otros.

En esta etapa del proyecto el modelo que dio mejores resultados estaba compuesto por tres capas convolucionales con 32, 64 y 128 filtros respectivamente, con 1 capa de *max pooling* entre cada una de estas, como se aprecia en la Figura 12. En cuanto al optimizador se utilizó *Adam* con un *learning rate* de 0.00001.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 52, 52, 32)        2432
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 26, 26, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 26, 26, 64)        51264
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 13, 13, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 13, 13, 128)       73856
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 7, 7, 128)         0
_________________________________________________________________
flatten (Flatten)            (None, 6272)              0
_________________________________________________________________
dense (Dense)                (None, 14)                87822
=================================================================
Total params: 215,374
Trainable params: 215,374
Non-trainable params: 0
_________________________________________________________________
```

Figura 12: Arquitectura de la CNN construida a la medida

Los resultados después del entrenamiento de esta red no fueron muy buenos, pues después de 100 épocas de entrenamiento apenas se alcanzaba un 37.56% de *accuracy* y el modelo ya comenzaba a presentar señales de *overfitting*, como se puede ver en la Figura 13, donde el *accuracy* del conjunto de entrenamiento tiende a aumentar pero no ocurre lo mismo para el de validación.

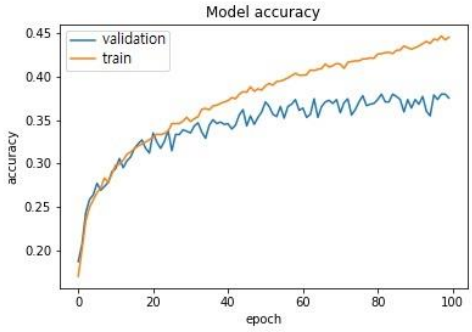


Figura 13: *Accuracy* de la CNN a la medida

Utilización de arquitecturas existentes

Debido a los largos períodos de tiempo que implicaría el construir y entrenar nuevas redes, se optó por usar algunas de las ya existentes. Las arquitecturas que se probaron fueron *VGG19* [23], *DenseNet201* [24], y *MobileNetV2* [25]. Dichos modelos pueden ser instanciados directamente desde *Keras* utilizando el módulo *applications* [26], y una vez hecho esto se crea un nuevo modelo que tendrá como "primera capa" todo el contenido del modelo instanciado, y como capa de salida una capa densa de 14 unidades, correspondientes a los estilos fotográficos, como se aprecia en la Figura 14.

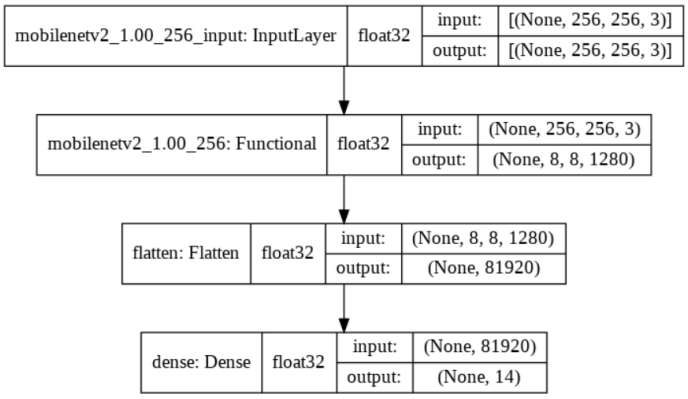


Figura 14: Representación del modelo MobileNetV2

Para los tres modelos se establecieron los mismos hiperparámetros; *Adam* como optimizador, *learning rate* de 6.5×10^{-6}, y entrenados por 12 épocas, esto con el fin de probarlos en igualdad de condiciones y verificar cuál era el que tenía mejor desempeño. Estos modelos fueron entrenados con el *dataset* AVA completamente desde cero con los pesos inicializados aleatoriamente. El modelo que obtuvo un mejor desempeño fue *DenseNet201* con un *accuracy* del 32.02%.

Transfer learning

Una vez terminado el ciclo de entrenamiento de la fase anterior, los resultados hasta el momento no estaban cerca de lo esperado. El hecho de que las arquitecturas existentes como *VGG19* y *DenseNet201* tampoco generaran buenos resultados era un indicio de que el problema estaba en los datos; estos no eran suficientes para entrenar un modelo desde cero. Bajo esta premisa, se utilizó la técnica de *transfer learning* la cual consiste en utilizar un modelo previamente entrenado y "transferir" ese aprendizaje a un nuevo modelo [27].

Para implementar esta técnica se utilizaron los mismos modelos de la fase anterior: *VGG19, DenseNet201,* y *MobileNetV2*. Dichos modelos ya han sido pre-entrenados con el conjunto de datos de *ImageNet,* [28], por lo que se cargaron los pesos aprendidos de ese *dataset*, los cuales son proporcionados por Keras.

El *transfer learning* puede ser implementado de distintas maneras, por ejemplo: utilizar el modelo pre-entrenado directamente en el nuevo dominio, hacer un ajuste fino del modelo "congelando" algunas de sus capas para que no se vean afectadas durante el entrenamiento, o utilizar una parte del modelo para integrarlo en uno nuevo [27].

Después de realizar distintas pruebas con algunas de las técnicas mencionadas anteriormente, la que mejores resultados dio fue el de utilizar y entrenar el modelo completo sin "congelar" ninguna de sus capas (Figura 15), y manteniendo un *learning rate* igualmente bajo (6.5×10^{-6}) con el fin de no perder lo que la red ya había aprendido de *ImageNet*. Nuevamente, al igual que en la etapa anterior, el modelo con el mejor desempeño fue *DenseNet201,* como se aprecia en la Figura 16, con un *accuracy* del 51.92%.

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
vgg19 (Functional)           (None, 8, 8, 512)         20024384
_________________________________________________________________
flatten (Flatten)            (None, 32768)             0
_________________________________________________________________
dense (Dense)                (None, 14)                458766
=================================================================
Total params: 20,483,150
Trainable params: 20,483,150
Non-trainable params: 0
_________________________________________________________________
```

Figura 15: Parámetros del modelo VGG19

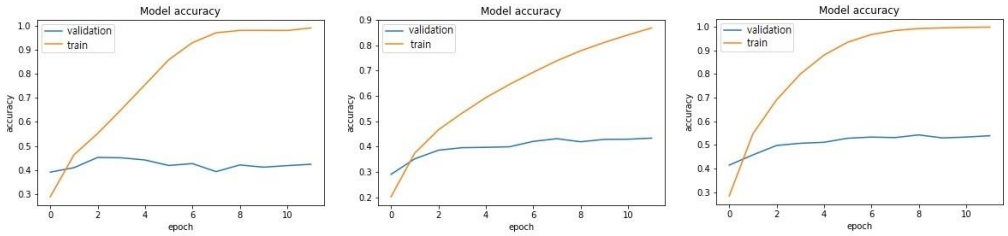


Figura 16: Transfer learning con VGG19, MobileNetV2 y DenseNet201

Data augmentation y Dropout

El modelo pre-entrenado *DenseNet201* mostraba resultados que ya se acercaban a los planteados en la fase de requerimientos, por esta razón se decidió trabajar sobre él y utilizar técnicas que permitieran reducir el *overfitting*, pues este era su principal problema.

Data augmentation consiste en generar más datos a partir de los existentes, algunas de las opciones para realizar este aumento son; girar la imagen, invertirla, proyectarla sobre sus ejes, hacer zoom, entre otras [29]. Para tratar de evitar que las transformaciones de las imágenes afectaran a su estilo al realizar el aumento, solo se hicieron dos transformaciones por cada imagen, se proyectaron tanto en su eje horizontal como vertical, como se aprecia en la Figura 17.

Después del aumento, el *dataset* terminó con **33,759** imágenes. Adicionalmente, se construyó otro conjunto pero esta vez solo se aumentaron aquellas clases que tenían menos de 1,000 muestras hasta alcanzar dicha cantidad, y así tratar de lograr un mejor balance en las clases, el conjunto resultante de este aumento fue de **14,656** imágenes. Para hacer este aumento se utilizó la librería *OpenCv* [30].

Figura 17: Imagen aumentada

Otra de las técnicas utilizadas para reducir el *overfiting* es el *dropout*, que es una forma de establecer en 0 y de manera aleatoria cierto porcentaje de las neuronas de una o más capas durante el entrenamiento de la red [31]. Para la implementación de esta técnica se

utilizó la clase *Dropout* de *Keras*, la cual viene incluida en el módulo *layers*; a la arquitectura de *DenseNet201* se le agregaron dos nuevas capas completamente conectadas antes de la capa de salida, y en medio de cada una de ellas se agregó una capa de *dropout* con una tasa del 40%. La arquitectura del modelo *DenseNet201* modificado se muestra en la Figura 18.

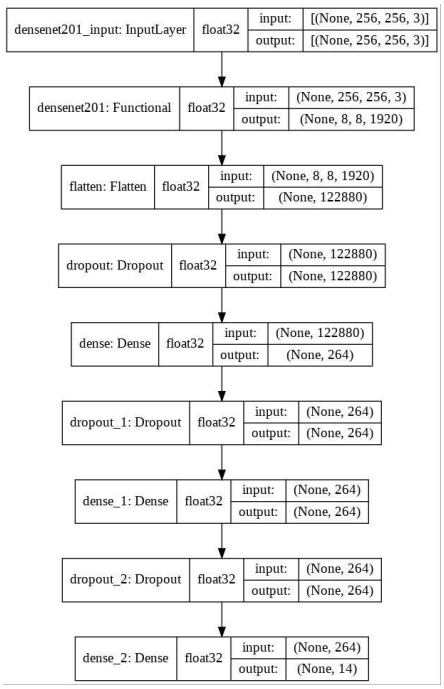


Figura 18: Modelo DenseNet201 modificado con *dropout*

Modificación de las entradas

Al finalizar los ciclos de entrenamiento de la etapa anterior, ya se tenía un modelo que cumplía con los requerimientos establecidos, la red ya mostraba un *mAP* del **60.69%**. Sin embargo, estos resultados no eran considerablemente buenos.

La metodología utilizada menciona que cuando ya se han probado distintas configuraciones y arquitecturas de redes neuronales, y aun así no se obtienen buenos resultados; esto puede ser debido a los datos, por lo que es necesario analizarlos para tener un mejor entendimiento de ellos y hacer modificaciones para mejorar el desempeño de la red [4].

Los resultados obtenidos mostraban que las clases con peor *accuracy* eran *motion blur* y *rule of thirds*, con 29% y 31% respectivamente. Por esta razón, se hizo un análisis para tratar de encontrar si el bajo desempeño en estas clases podría ser debido a los datos.

El estilo *motion blur* presentaba un alto grado de confusión con el estilo de *long* exposure. En su definición, ambos estilos hacen referencia a movimiento dentro de la imagen [8], [5]. En la Figura 19 se aprecian las similitudes en los dos estilos, ambas presentan movimiento, sin embargo, en *long exposure* no necesariamente hay un desenfoque completo, pues las teclas del piano están perfectamente enfocadas.

Figura 19: Imágenes etiquetadas como *motion blur* y *long exposure*

Por otra parte, la *regla de tercios* (traducción de *rule of thirds*) es un estilo que habla acerca de la posición en la que se coloca el sujeto principal de una imagen, esta dice que el encuadre se debe dividir en tres partes iguales tanto horizontales como verticales y colocar el sujeto a fotografiar en alguna de las intersecciones de estas divisiones [32]. Este estilo puede presentar problemas en el clasificador, pues al tratarse de posición, se puede dar el caso de que este estilo se encuentre embebido en alguna de las otras categorías. En la Figura 20 se muestra una imagen que corresponde al estilo de *shallow DOF*, sin embargo, aplicando la *regla de tercios* se observa que también se cumple.

Figura 20: Imagen etiquetada como *shallow DOF*

Algo similar a lo que ocurría en las clases de *motion blur* y *long exposure* también sucedía con *macro* y *shallow DOF* (en español, *poca profundidad de campo*); pues la fotografía *macro* puede implicar una reducción de la profundidad de campo en algunos

casos [33]. En la Figura 21 se aprecia cómo ambas imágenes tienen poca profundidad de campo, esto es, el efecto de desenfoque en el fondo [5].

Figura 21: Imágenes etiquetadas como *macro* y *shallow DOF*

Para verificar el impacto del "ruido" que podrían introducir al clasificador las clases antes mencionadas, se entrenó nuevamente el modelo que ya se tenía pero esta vez removiendo las clases *rule of thirds, motion blur,* y *shallow DOF*. Los resultados obtenidos se presentan en la Sección 5 del presente documento.

4.4 Despliegue de la red

Después de que la red ha cumplido con todos los requerimientos, se tiene que implementar dentro de una aplicación host, es decir, esta tiene que ser embebida en otro sistema [4]. Debido a la naturaleza del presente proyecto en la que no se trabajó en un entorno empresarial, para esta etapa se realizó una interfaz en consola en la que se pudiera cargar el modelo y utilizarlo para clasificar imágenes individualmente.

4.5 Pruebas y verificación independiente

En esta etapa se realiza una verificación de todas las fases previas. Se deben verificar desde la recolección de los datos hasta el despliegue de la red, para asegurarse de que todo funciona según lo especificado. Aquí es donde se utilizan las distintas métricas que existen para evaluar la efectividad de la red [4].

Para validar los resultados de este proyecto se utilizó la métrica *mAP* y se compararon los resultados con los obtenidos con algunos trabajos previos en los que se utilizó el mismo *dataset* y la misma métrica, los resultados se presentan en Sección 5.

5. Resultados

En la Tabla 2 se presenta una comparativa de las distintas configuraciones de CNN entrenadas. Se encontró que el modelo que obtuvo el mejor desempeño fue el denominado *DenseNet201 + dropout* y el aumento de imágenes completo. Por otra parte, el modelo de *MobileNetV2* es el que presenta el peor desempeño. Si bien, la *CNN a la medida* no presenta el peor desempeño, si requiere el mayor número de *épocas* para alcanzar sus resultados.

Tabla 2: Resultados obtenidos de las distintas configuraciones de CNN probadas

Modelo	Pesos pre-entrenados	Data augmentation	Épocas	mAP
CNN a la medida	No	No	100	31.75%
VGG19	No	No	12	26.33%
DenseNet201	No	No	12	27.10%
MobileNetV2	No	No	12	12.19%
VGG19	Imagenet	No	12	37.83%
DenseNet201	Imagenet	No	12	52.15%
MobileNetV2	Imagenet	No	12	42.41%
DenseNet201 + dropout	Imagenet	No	20	59.77%
DenseNet201 + dropout	Imagenet	Balanceado	20	60.10%
DenseNet201 + dropout	**Imagenet**	**Completamente**	**10**	**60.69%**

La Figura 22 muestra la matriz de confusión de modelo *DenseNet201 + dropout* que tuvo los mejores resultados. Como se puede observar, el estilo *motion blur* es el que peor desempeño tiene y presenta un alto grado de confusión con el estilo *long exposure*. Por otra parte, *rule of thirds*, es el segundo peor y sus grados de confusión están repartidos entre todas las clases. Otro par de estilos en los que se puede apreciar un alto grado de confusión entre sí, son *macro* y *shallow DOF*.

En la Tabla 3 se muestra el *mAP* del modelo *Densenet201 + dropout* entrenado sobre el *dataset* completamente aumentado y con el *transfer learning* de *imagenet*, esta vez, quitando las clases *rule of thirds, motion blur,* y *shallow DOF*. Como se puede observar, conforme se fueron quitando clases el *mAP* del modelo aumentó. De esta manera, un modelo con 11 clases tiene una mejora del 12.92% con respecto al de 14 clases.

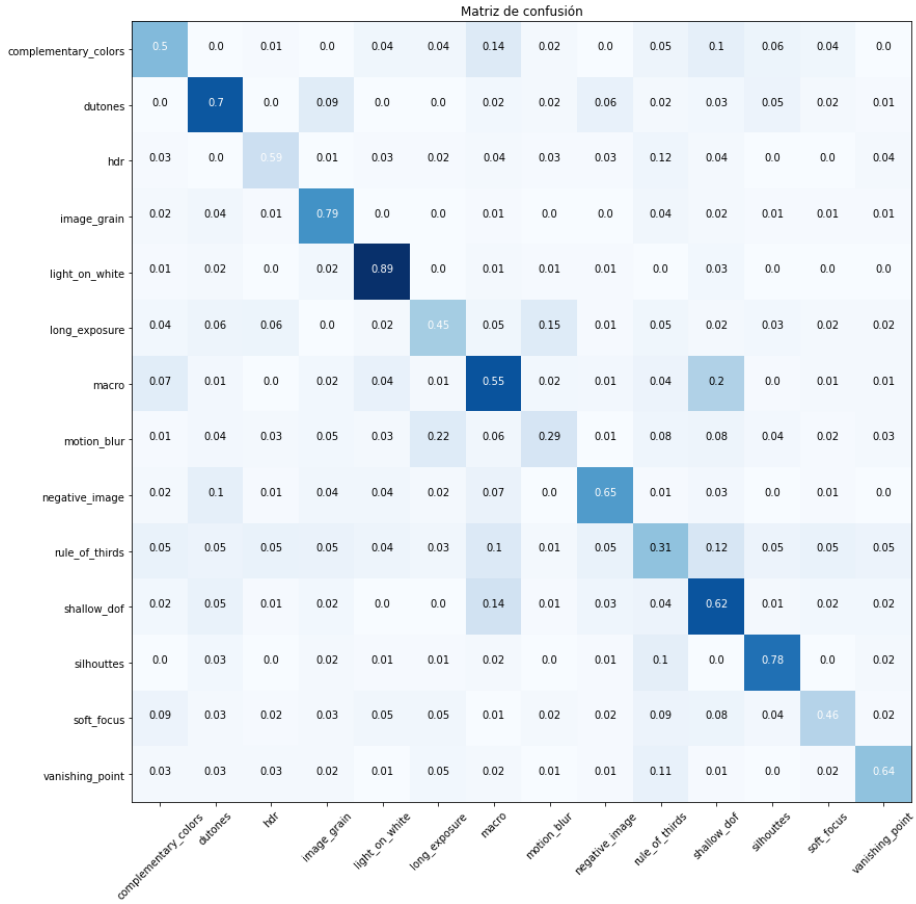


Figura 22: Matriz de confusión del modelo *DenseNet201 + dropout*

Tabla 3: Resultados del modelo DenseNet201 + dropout entrenado con menos clases

Clases removidas	mAP
Ninguna	60.69%
Rule of thirds	65.32%
Rule of thirds + motion blur	68.66%
Rule of thirds + motion blur + shallow DOF	**73.61%**

En la Figura 23 se muestra la matriz de confusión resultante de quitar las clases *rule of thirds*, *motion blur* y *shallow DOF*. Al haber removido las clases problemáticas con *long exposure* y *macro*, estas reciben una mejora del 15% en su exactitud, comparándolos con los resultados de la Figura 22.

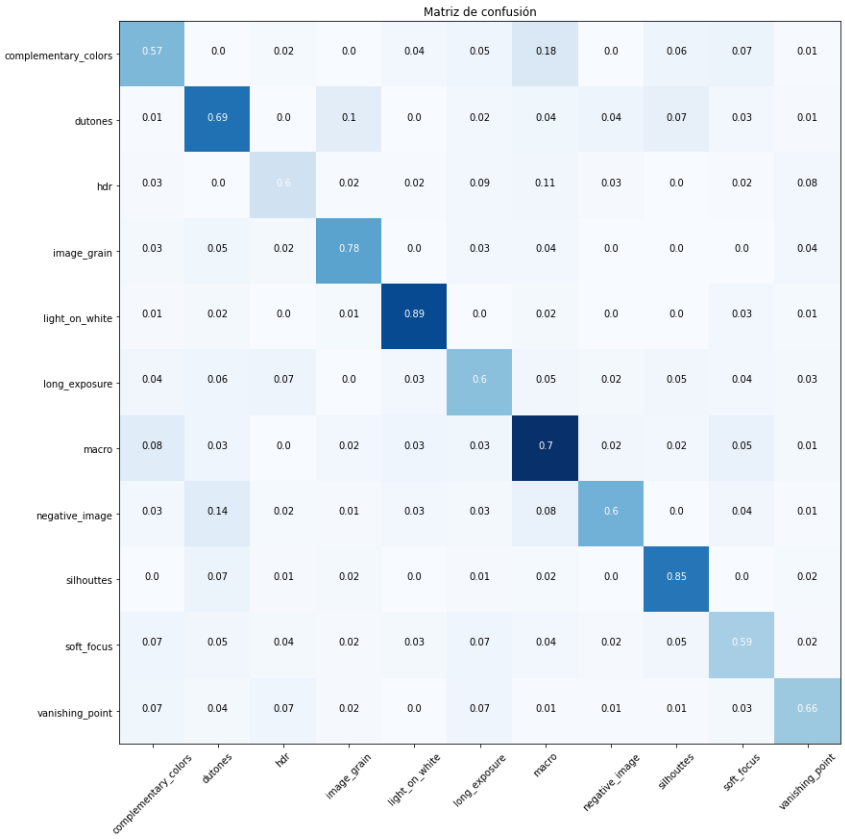


Figura 23: Matriz de confusión de *DenseNet201 + dropout* entrenado con 11 clases

Con el fin de obtener una mejor estimación de la habilidad del modelo para predecir datos no vistos durante el entrenamiento, se utilizó el método de *cross validation* denominado *stratified k-folds* [34]. Se utilizaron 10 *folds* sobre el conjunto de datos aumentado completamente y los resultados de entrenar el modelo *DenseNet201 + dropout* se muestran en la Tabla 4.

Tabla 4: Resultados del 10 *folds cross validation*

Fold	mAP
1	60.05%
2	61.58%
3	59.83%
4	61.93%
5	61.96%
6	60.69%
7	56.70%
8	59.63%
9	58.22%
10	60.46%
Promedio	60.10%

Si bien, 11,253 imágenes pueden parecer muchas para entrenar una red neuronal convolucional, los resultados demuestran lo contrario, ya que aquellos modelos que se entrenaron sin ningún tipo de *transfer learning* se comportaron peor que aquellos que si utilizaban pesos previamente aprendidos de otro *dataset*. Además, entrenar modelos desde cero requiere un tiempo mayor de entrenamiento.

Por otra parte, en cuanto a las técnicas utilizadas para reducir el *overfitting*, el *dropout* resulta más efectivo que el *data augmentation*. Lo anterior puede ser debido al dominio de aplicación, pues al aumentar los datos se podría estar modificando el estilo de la imagen, y esto podría impactar negativamente en el desempeño del modelo.

A pesar de que se mejoró el *mAP* del presentado en [1] y [2], la mejora es de apenas del 6.85% y 2.59% respectivamente. Según lo observado durante el desarrollo del proyecto y sus resultados, la principal causa de tan poco aumento puede estar en los datos, ya que el análisis presentado en el Capítulo III sugiere que algunos de los estilos nos son mutuamente excluyentes, es decir, una imagen puede contener más de un estilo, aunado a esto se encuentra la similitud entre clases como *motion blur* y *long exposure*. Lo anterior se ve reforzado con el hecho de que al momento de quitar las clases "ruidosas" el modelo mejora su desempeño.

6. Productos generados

- Se redactó un artículo titulado, "*Clasificación de estilos fotográficos utilizando una Red Neuronal Convolucional*", aceptado para publicación en la revista *Research in Computing Science*, ISSN 1870-4069.
- Se presentó una ponencia en el 22º CONGRESO INTERNACIONAL DE CIENCIAS DE LA COMPUTACIÓN CORE 2022 realizado del 26-30 septiembre del 2022.
- Se realizó una tesis de licenciatura con la que el estudiante Andre Martin Vera Valdez logró su grado de Ingeniero en Software.

7. Conclusiones

En este proyecto se presentó la implementación de diferentes CNNs para clasificar fotografías de acuerdo con su estilo fotográfico. Se entrenaron ocho modelos de clasificación, uno de ellos fue una CNN simple y los demás se implementaron utilizando las CNNs preentrenadas VGG19, DenseNet201, y MobileNetV2, transfer learning y estrategias para reducir el overfitting.

Los modelos se entrenaron en 11,253 imágenes anotadas con 14 estilos fotográficos del dataset AVA, las cuales se normalizaron entre 0 y 1 y se redimensionaron a 256×256 píxeles. Además, se utilizó data augmentation para crear dos nuevos conjuntos de imágenes. En el primero se transformó completamente el conjunto de imágenes, resultando 33,759 imágenes (completo). En el segundo se aumentaron las clases con menos de 1,000 imágenes con la finalidad de balancear las clases, resultando 14,691 imágenes (balanceado).

Los resultados demuestran que el mejor modelo fue DenseNet201 + dropout + transfer learning + data augmentation completo obtuvo el mejor desempeño, alcanzando un mAP de 60.69%.

Al analizar los resultados se identificó a través de la matriz de confusión que tres estilos fotográficos pudieron haber generado ruido en el modelo, los cuales fueron rule of thirds, motion blur y shallow DOF. Al eliminar estos estilos del conjunto de imágenes y volver a entrenar el modelo, éste alcanzó un mAP de 73.61%. Esto debido a que existe similitud entre estos tres estilos y a que una fotografía puede pertenecer a más de un estilo en AVA, lo cual sugiere abordar la clasificación de estilos fotográficos desde un enfoque multietiqueta.

Como trabajo futuro se sugiere recolectar y/o construir un dataset con una gran cantidad de imágenes para verificar si existe una mejora al entrenar CNNs simples sin la necesidad de utilizar transfer learning. Además, se pretende abordar la clasificación de estilos fotográficos desde un enfoque multietiqueta.

8. Referencias

[1] N. Murray, L. Marchesotti y F. Perronnin, «AVA: A large-scale database for aesthetic visual analysis,» de *2012 IEEE Conference on Computer Vision and*

Pattern Recognition, Providence, RI, 2012.

[2] S. Karayev, M. Trentacoste, H. Han, A. Agarwala, T. Darrell, A. Hertzmann y H. Winnemoeller, «Recognizing Image Style,» de *British Machine Vision Conference*, 2014.

[3] I. Pérez Roldán, *Clasificación de obras de arte por estilo artístico usando redes neuronales convolucionales,* proyecto de fin de grado, Universisdad Politécnica de Madrid, 2019.

[4] D. Rodvold, «A software development proccess model for artificial neural networks in critical applications,» de *International Joint Conference on Neural Networks*, Washington, DC, 2002.

[5] M. Freeman, The photographer's mind, Lewes, UK: Elsevier, 2011.

[6] Y. LeCun, Y. Bengio y Geoffrey Hinton, «Deep learning,» de *Nature 521*, 2015.

[7] A. P. Martínez Lanz Durán, *Memorias: fotgrafía pictórica, tesis*, Cholula, Puebla: Universidad de las Américas de Publa, 2003.

[8] M. Freeman, El estilo en fotografía, Madrid, España: H. Blume, 1986.

[9] P. Domingos, The master algorithm, New York: Basci Books, 2015.

[10] M. Myszczynska, P. Ojamies, A. M. Lacoste, D. Neil, A. Saffari, R. Mead, G. M. Hautbergue, J. D. Holbrook y L. Ferraiuolo, «Applications of machine learning to diagnosis and treatment of neurodegenerative diseases,» *Nature Reviews Neurology*, vol. 16, pp. 440-456, 2020.

[11] L. Rouhiainen, Inteligencia artificial, 101 cosas que debes saber hoy sobre nuestro futuro., Barcelona, España: Planeta, 2018.

[12] V. Constantin Cardei, «A neural network approach to colour constancy, Ph. D thessis,» Simon Fraser University, Burnaby, Canada, 2000.

[13] I. Goodfellow, Y. Bengio y A. Courville, Deep Learning, MIT Press, 2016.

[14] S. Saha, «A Comprehensive Guido to Convolutional Neural Networks - the ELI5 way,» Towarsd data science, 15 Diciembre 2018. [En línea]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53. [Último acceso: 05 October 2020].

[15] C. Aggarwal, Neural networks and deep learning, Cham, Switzerland: Springer, 2018.

[16] TensorFlow, «Tensor Flow Doc,» 2017. [En línea]. Available: https://tensorflowdoc.readthedocs.io/es/latest/. [Último acceso: 5 Octubre 2020].

[17] Keras, «Keras documentation,» [En línea]. Available: https://faroit.com/keras-docs/1.2.0/. [Último acceso: 5 Octubre 2020].

[18] Colab, «¿Qué es Colaboratory?,» [En línea]. Available: https://colab.research.google.com/notebooks/intro.ipynb#scrollTo=5fCEDCU_qrC0. [Último acceso: 5 Octubre 2020].

[19] N. Murray, L. Marchesotti y F. Perronnin, «AVA: A Large-Scale Database for Aesthetic Visual Analysis,» 13 Noviembre 2016. [En línea]. Available: https://github.com/mtobeiyf/ava_downloader. [Último acceso: 12 Noviembre 2020].

[20] Keras, «Image data preprocessing,» Keras, [En línea]. Available: https://keras.io/api/preprocessing/image/. [Último acceso: 10 Marzo 2021].

[21] S. Ramesh, «Towards Data Science,» 7 Mayo 2018. [En línea]. Available: https://towardsdatascience.com/a-guide-to-an-efficient-way-to-build-neural-network-architectures-part-ii-hyper-parameter-42efca01e5d7. [Último acceso: 4 Enero 2021].

[22] H. H. Seyyed , R. Mohammad , F. Mohsen , S. Mohammad y A. Ehsan , «Towards Principled Design of Deep Convolutional Networks: Introducing SimpNet,» 17 Febrero 2018. [En línea]. Available: https://arxiv.org/abs/1802.06205. [Último acceso: 2 Enero 2021].

[23] S. Karen y Z. Andrew, «Very deep convolutional networks for large-scale image recognition,» de *International Conference on Learning Representations*, Toulon, France, 2015.

[24] G. Huang, Z. Liu, L. Van Der Maaten y W. Kilian, «Densely Connected Convolutional Networks,» de *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2018.

[25] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov y L.-C. Chen, «MobileNetV2: Inverted Residuals and Linear Bottlenecks,» de *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018*, Salt Lake City, Utah, 2018.

[26] Keras, «Keras Applications,» Keras, [En línea]. Available: https://keras.io/api/applications/. [Último acceso: 10 Enero 2021].

[27] J. Brownlee, «Transfer Learning in Keras with Computer Vision Models,» Machine Learning Mastery, 18 Agosto 2020. [En línea]. Available: https://machinelearningmastery.com/how-to-use-transfer-learning-when-developing-convolutional-neural-network-models/. [Último acceso: 20 Marzo 201].

[28] L. Fei-Fei, J. Deng, O. Russakovsky, A. Berg y K. Li, «Imagenet,» Imagenet, 19 Mayo 2015. [En línea]. Available: http://www.image-net.org/about. [Último acceso: 18 Abril 2021].

[29] TensorFlow, «Aumento de datos,» TensorFlow, 19 Marzo 2021. [En línea]. Available: https://www.tensorflow.org/tutorials/images/data_augmentation. [Último acceso: 10 Abril 2021].

[30] OpenCv, «OpenCV modules,» OpenCv, [En línea]. Available: https://docs.opencv.org/master/. [Último acceso: 10 Abril 2021].

[31] Keras, «Dropout layer,» Keras, [En línea]. Available: https://keras.io/api/layers/regularization_layers/dropout/. [Último acceso: 14 Enero 2021].

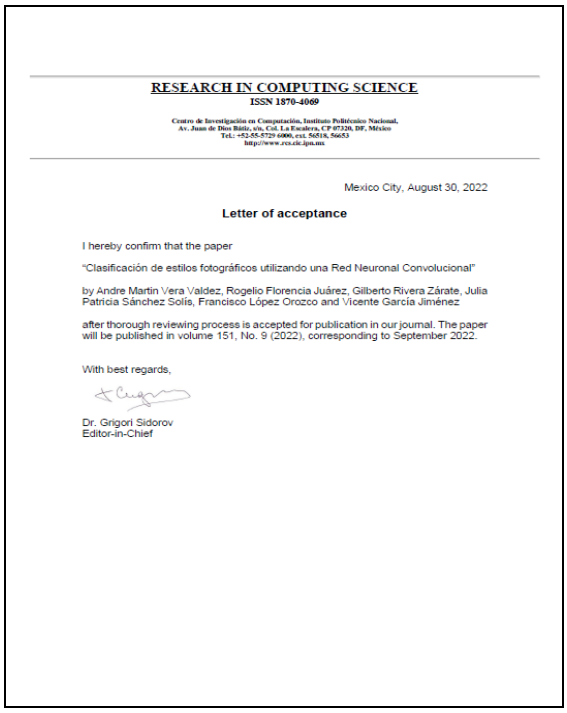
[32] S. A. Amirshahi, . U. G. Hayn-Leichsenring, D. Joachim y C. Redies, «Evaluating the Rule of Thirds in Photographs and Paintings,» *Art & Perception,* vol. 2, pp. 163-182, 2014.

[33] R. Sheppard, Macro Photography From Snapshots To Great Shots, Peachpit Press, 2015.

[34] Scikit Learn, «3.1. Cross-validation: evaluating estimator performance,» [En línea]. Available: https://scikit-learn.org/stable/modules/cross_validation.html. [Último acceso: 20 Abril 2021].

9. Anexos

- Se anexa la imagen de la carta de aceptación del artículo "*Clasificación de estilos fotográficos utilizando una Red Neuronal Convolucional*" aceptado para publicación en la revista *Research in Computing Science*, ISSN 1870-4069.
- Se anexa la primera página del artículo aceptado.
- Se anexa el acta del examen de grado de la tesis de licenciatura con la que el estudiante *Andre Martin Vera Valdez* logró el grado de Ingeniero en Software.
- Por último, se anexa la constancia de la ponencia realizada en el 22º CONGRESO INTERNACIONAL DE CIENCIAS DE LA COMPUTACIÓN CORE 2022 realizado del 26-30 septiembre del 2022.



RESEARCH IN COMPUTING SCIENCE

ISSN 1870-4069

Centro de Investigación en Computación, Instituto Politécnico Nacional,
Av. Juan de Dios Bátiz, s/n, Col. La Escalera, CP 07320, DF, México
Tel.: +52-55-5729 6000, ext. 56518, 56653
http://www.cic.ipn.mx

Mexico City, August 30, 2022

Letter of acceptance

I hereby confirm that the paper

"Clasificación de estilos fotográficos utilizando una Red Neuronal Convolucional"

by Andre Martin Vera Valdez, Rogelio Florencia Juárez, Gilberto Rivera Zárate, Julia Patricia Sánchez Solís, Francisco López Orozco and Vicente García Jiménez

after thorough reviewing process is accepted for publication in our journal. The paper will be published in volume 151, No. 9 (2022), corresponding to September 2022.

With best regards,

Dr. Grigori Sidorov
Editor-in-Chief

Clasificación de estilos fotográficos utilizando una Red Neuronal Convolucional

Andre Martin Vera Valdez[1], Rogelio Florencia Juárez[1][0000−0002−5208−6577], Gilberto Rivera Zárate[1][0000−0002−2365−4651], Julia Patricia Sánchez-Solís[1][0000−0001−5514−5081], Francisco López-Orozco[1][0000−0002−2917−5088], and Vicente García Jiménez[1][0000−0003−2820−2918]

Departamento de Ingeniería Eléctrica y Computación, División Multidisciplinaria de Ciudad Universitaria, Universidad Autónoma de Ciudad Juárez
{al154007@alumnos.uacj.mx
{rogelio.florencia,gilberto.rivera,julia.sanchez,francisco.orozco,vicente.jimenez}@uacj.mx

Resumen La fotografía es una actividad que muchas personas realizan cotidianamente para capturar en una imagen momentos importantes de sus vidas. Los fotógrafos agregan etiquetas manualmente a sus imágenes al subirlas a sitios web con la finalidad de describir aspectos importantes, como el estilo fotográfico, impulsando de esta manera su visibilidad en las búsquedas que realizan los usuarios. Sin embargo, etiquetar manualmente cada fotografía se vuelve una tarea tediosa que consume demasiado tiempo cuando se trata de una gran cantidad de imágenes, además de requerir conocimientos profundos en fotografía. Este artículo presenta la implementación de diferentes redes neuronales convolucionales para clasificar fotografías de acuerdo con 14 estilos fotográficos contenidos en el dataset AVA. Se entrenaron ocho modelos diferentes: a) un modelo de una red neuronal convolucional simple; b) tres modelos basados en *VGG19*, *DenseNet201* y *MobileNetV2*; c) tres modelos mediante *aprendizaje por transferencia* y d) un modelo que, en base al mejor de los anteriores, fue adicionado con estrategias para reducir el *sobrajuste*. Los resultados indican que el mejor desempeño se obtuvo al utilizar *aprendizaje por transferencia* sobre *DenseNet201* adicionado con estrategias para reducir el *sobrajuste*, alcanzando un *promedio medio de precisión* de 60.69 %.

Keywords: Estilos fotográficos · Redes neuronales convolucionales · Aprendizaje por transferencia · Aumento de datos · Sobrajuste

1. Introducción

La fotografía es una actividad que forma parte de nuestro día a día; principalmente por el fácil acceso que se tiene a una cámara y a que es una tarea muy sencilla de realizar, ya que solo se tiene que poner un sujeto delante de una cámara y presionar el botón de *disparar*. El sujeto (o sujetos) es el elemento principal de una fotografía que está dentro del marco, por ejemplo: una persona,

27/5/2021 Acta Titulación

UNIVERSIDAD AUTÓNOMA DE CIUDAD JUÁREZ

Instituto de Ingeniería y Tecnología

EVALUACIÓN DE EXAMEN PROFESIONAL INTRACURRICULAR
NIVEL: LICENCIATURA

Fecha: 31/May/2021
Horario: 12:30 a 14:00
Salón:

TEMA: **Red Neuronal Convolucional para la Clasificación de Estilos Fotográficos**

La evaluación del examen profesional intracurricular consta de 4 partes:
1.- Exposición por parte de los alumnos (aprox. 20 minutos).
2.- Réplica por parte del jurado.
3.- Comentarios y/o recomendaciones.
4.- Entrega de resultados.

Nombre del alumno: **VERA VALDEZ ANDRE MARTIN**
Matrícula : **154007**

Calificación Maestro de la materia (30%)	30
Calificación Director del trabajo (40%)	40
Calificación del Jurado (30%)	28
Total	98

El trabajo tiene impacto en las áreas de:

Social	Si ___	No X
Tecnológico	Si X	No ___
Económico	Si ___	No X
Medio ambiente	Si ___	No X

Director del trabajo — DR. ROGELIO FLORENCIA JUÁREZ
Jurado — DR. VICENTE GARCÍA JIMÉNEZ
Maestro de la materia — DR. GILBERTO RIVERA ZÁRATE
DR. GILBERTO RIVERA ZÁRATE
DR. FRANCISCO LÓPEZ OROZCO

UACJ
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA Y COMPUTACIÓN

localhost/tis/ActaTit.php?var1=154007 1/1

INSTITUTO POLITÉCNICO NACIONAL
CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN

otorgan la presente

CONSTANCIA

a

Andre Martin Vera Valdez

Por presentar la ponencia titulada:
"Clasificación de estilos fotográficos utilizando una Red Neuronal Convolucional"
llevada a cabo el día 27 de septiembre de 2022
en el marco del 22º Congreso Internacional de Ciencias de la Computación CORE 2022
en modalidad presencial.

Dr. Francisco Hiram Calvo Castro, CIC
Director Interino del CIC

Dr. Victor Gabriel Reyes Macedo
Organizador del CORE22

CORE

9.1 Taxonomía de los Roles de Colaborador (con las actividades logradas)

Roles	Definición de los roles	Nombre de él(la) investigador(a)	Figura	Grado de contribución	Actividades logradas durante el proyecto	Tiempo promedio semanal (en horas) dedicado al proyecto
Responsabilidad de la dirección del proyecto.	Coordinar la planificación y ejecución de la actividad de investigación.	Rogelio Florencia Juárez	Director	Principal	Reporte técnico del estudiante de Seminario de Titulación II Redacción de artículo enviado al CORE 2022	10
Responsabilidad de supervisión	Elaborar la planificación de las actividades de la investigación (cronogramas y controles de seguimiento).	Rogelio Florencia Juárez Gilberto Rivera Zárate	Director Supervisor	Principal Principal	Propuesta del proyecto en Seminario de Titulación I	2
Realización y redacción de la propuesta	Preparación, creación y redacción de la propuesta de investigación, revisión de coherencia del texto, presentación de los datos y la normatividad aplicable para garantizar el cumplimiento de los requisitos.	Julia Patricia Sánchez Solís Francisco López Orozco	Redactor de propuesta Redactor de propuesta	Principal Principal	Propuesta del proyecto en Seminario de Titulación I Presentación de la propuesta del estudiante ante el comité tutorial	5
Desarrollo o diseño de la metodología	Contribuir con el diseño de la metodología, modelos a implementar y el sustento teórico, empírico y científico para la aplicabilidad de los instrumentos en la ejecución del proyecto.	Julia Patricia Sánchez Solís Vicente García Jiménez	Diseñador metodología Diseñador metodología	Principal Principal	Desarrollo del software (Redes neuronales convolucionales)	5
Recopilación / recolección de datos e información	Ejecuta las estrategias propuestas en acciones encaminadas a obtener la información, haciendo la recopilación de datos y la inclusión de la evidencia en el proceso.	Francisco López Orozco Vicente García Jiménez	Recopilador de datos Recopilador de datos	Principal Principal	Dataset AVA, utilizado para entrenar las redes neuronales convolucionales	2
Elaboración del análisis formal de la investigación	Aplicar métodos estadísticos, matemáticos, computacionales, teóricos u otras técnicas formales para analizar o sintetizar los datos del estudio. Verifica los resultados preliminares de cada etapa del análisis, los experimentos implementados y otros productos comprometidos en el proyecto.	Julia Patricia Sánchez Solís Vicente García Jiménez	Analista de Datos Analista de datos	Principal Principal	Preprocesamiento de las imágenes del dataset AVA. Validación de los resultados de las redes neuronales convolucionales	5

Preparación, creación y/o presentación de los productos o entregables	Preparar la redacción del reporté técnico de avance parcial y el reporte técnico final. Se hace la revisión crítica, la recopilación de las observaciones y comentarios del grupo de investigación. Y finalmente se procede a la edición del documento a entregar.	Rogelio Florencia Juárez Gilberto Rivera Zárate Julia Patricia Sánchez Solís Francisco López Orozco Vicente García Jiménez	Editor reporte técnico Editor reporte técnico Editor reporte técnico Editor reporte técnico Editor reporte técnico	Principal Apoyo Apoyo Apoyo Apoyo	Elaboración de este reporte técnico de proyecto interno sin financiamiento	5

9.1.1 Estudiantes participantes en el proyecto

Nombre de estudiante(s)	Matrícula	Tiempo promedio semanal (en horas) dedicado al proyecto	Actividades logradas en la ejecución del proyecto
Andre Martin Vera Valdez	154007	16 horas	• Redacción de su reporte técnico de investigación con el que logró el grado de ingeniero. • Desarrollo del software del proyecto.