ISSN 1870-4069

Clasificación de estilos fotográficos utilizando una Red Neuronal Convolucional

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

Universidad Autónoma de Ciudad Juárez, Departamento de Ingeniería Eléctrica y Computación, División Multidisciplinaria de Ciudad Universitaria, México

{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 *sobreajuste*. Los resultados indican que el mejor desempeño se obtuvo al utilizar *aprendizaje por transferencia* sobre *DenseNet201* adicionado con estrategias para reducir el *sobreajuste*, alcanzando un *promedio medio de precisión* de 60.69 %.

Palabras clave: Estilos fotográficos, redes neuronales convolucionales, aprendizaje por transferencia, aumento de datos, sobreajuste.

Photographic Style Classification Using a Convolutional Neural Network

Abstract. Photography is an activity that many people perform on a daily basis to capture important moments of their lives in an image.

Photographers often add manual tags to their images when uploading them to websites in order to describe relevant aspects, such as the photographic style, thereby increasing their visibility in user searches. However, manually tagging each photograph becomes a tedious and time-consuming task when dealing with large volumes of images, in addition to requiring deep knowledge of photography. This article presents the implementation of different convolutional neural networks to classify photographs according to 14 photographic styles contained in the AVA dataset. Eight different models were trained: (a) a simple convolutional neural network model; (b) three models based on VGG19, DenseNet201, and MobileNetV2; (c) three models using transfer learning; and (d) a model that, based on the best of the previous ones, incorporated strategies to reduce overfitting. The results indicate that the best performance was obtained by applying transfer learning on DenseNet201, enhanced with strategies to reduce overfitting, achieving an average accuracy of 60.69

Keywords: Photographic styles, convolutional neural networks, transfer learning, data augmentation, overfitting.

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, un auto, una planta, un animal, etc. El sujeto es el motivo por el que se toma una foto [1].

Según [2], 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 [1], otro de los elementos que están presentes dentro de una fotografía es el *estilo fotográfico*. El estilo es la forma en la que se toma la fotografía, determinando el aspecto visual de la imagen; 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 [1]. Se puede decir que si el sujeto es el '*qué*', el estilo es el '*cómo*'. Además, Freeman [3] menciona que el estilo puede ser intencionado o no intencionado. El estilo fotográfico es importante para un fotógrafo porque está relacionado con los detalles, características, colores, etc., que caracterizan su trabajo y permiten diferenciarlo de entre otros fotógrafos. Entre los diferentes estilos fotográficos existentes se definen los siguientes [1]:

– *Motion blur*. Se aprecia el movimiento del sujeto y esto genera un desenfoque en ella.

- *Shallow DOF*. Cierta parte de la fotografía se encuentre desenfocada.
- *High contrast*. Existe una diferencia muy grande entre las zonas más oscuras y las más claras de la imagen.
- *Vanishing point*. Convergen dos o más líneas paralelas a un mismo punto.

El reconocimiento de imágenes es una de las distintas aplicaciones del aprendizaje automático [4] en donde se emplea un conjunto de métodos y técnicas para detectar y analizar imágenes con el fin de identificar lugares, personas, objetos, entre otros elementos que se encuentran dentro de éstas. Entre las diferentes tareas relacionadas con el reconocimiento de imágenes se pueden mencionar:

- Clasificación. Determina la clase a la que pertenece una imagen (ver [5]).
- Etiquetado. Identifica la presencia de varios objetos dentro de una imagen (ejemplo [6]).
- Detección. Localiza un objeto en una imagen, delimitándolo mediante un cuadro que se sitúa alrededor del objeto detectado (ver [7]).
- Segmentación. Es capaz de ubicar un elemento en una imagen al píxel más cercano (ejemplo [8]).

Recientemente, el reconocimiento de imágenes se ha empezado a utilizar en la identificación tanto de sujetos como de estilos fotográficos. Murray et al. [9] conformaron un *dataset* a gran escala con un conjunto de más de 250,000 imágenes utilizado para análisis visual estético que denominaron *Aesthetic Visual Analysis*, AVA. Los autores implementaron *Máquinas de Vectores de Soporte* lineales sobre un subconjunto de 14,079 imágenes de AVA relacionadas con 14 anotaciones de estilo fotográfico, obteniendo un *Promedio Medio de Precisión* (mAP, por sus siglas en inglés) de 53.85 %. Posteriormente, Karayev et al. [10] también utilizaron el *dataset* AVA para realizar clasificación de estilos fotográficos utilizando un algoritmo de clasificación lineal y métodos de extracción de características como histogramas de color o GIST, siendo una *Red Neuronal Convolucional* (CNN, por sus siglas en inglés) el método más efectivo con un mAP de 57.9 % y con la fusión de múltiples características un 58.1 %. Además, los autores propusieron otro *dataset* basado en imágenes recolectadas de la red social *Flickr*, obteniendo un mAP de 33.6 %. Celona et al. [11] propusieron una multired para la predicción automática de la estética de una imagen en base al análisis del contenido semántico, el estilo artístico y la composición de la imagen. La red propuesta está compuesta por una red preentrenada para la extracción de características semánticas, un perceptrón multicapa para la predicción de los atributos de la imagen y una hiper-red autoadaptativa para predecir los parámetros de la red dedicada a la estimación estética. Los autores reportaron una *exactitud* de un 80.75 %. Por otra parte, debido a que el reconocimiento de estilo no está limitado solo al área de la fotografía, en el trabajo de Pérez [12] se desarrolló una CNN para la clasificación de estilos en pinturas. La efectividad obtenida de la red fue un 51 %.

Hoy en día, tanto fotógrafos novatos como profesionales comparten sus contenidos fotográficos en sitios como *Instagram*, *Flickr*, *500px*, o *Shutterstock*,

ya sea simplemente por gusto o por una cuestión de comercialización. Al subir contenidos a estas plataformas, los fotógrafos tienen la opción de colocar manualmente etiquetas descriptivas a sus fotografías para posicionarlas mejor dentro de las búsquedas y lograr que sus contenidos lleguen a más personas, siendo el estilo fotográfico una de estas etiquetas. Esto se vuelve una tarea tediosa y tardada para el fotógrafo cuando se trata de una gran cantidad de fotografías lo cual resalta la necesidad de automatizar su clasificación de acuerdo con su estilo fotográfico.

En este artículo se describe la implementación de diferentes arquitecturas de CNNs para la clasificación de fotografías según su estilo fotográfico. Las CNNs fueron entrenadas sobre los 14 estilos fotográficos contenidos en el *dataset* AVA [9]. Integrar el modelo entrenado por la CNN en una aplicación podría ayudar a los fotógrafos a etiquetar automáticamente sus fotografías, además de no requerir conocimientos profundos en fotografía.

El artículo está estructurado de la siguiente manera. La Sección 2 presenta una descripción del conjunto de imágenes utilizado para entrenar las CNNs. La Sección 3 describe las diferentes arquitecturas de las CNNs entrenadas para la clasificación de fotografías en base a su estilo fotográfico. La Sección 4 muestra los resultados de los diferentes modelos entrenados y presenta una discusión sobre los resultados obtenidos. Por último, la Sección 5 menciona los hallazgos encontrados en este trabajo y sugiere algunos trabajos futuros.

2. Materiales y métodos

Las CNNs se entrenaron utilizando el *dataset* AVA [9]. Este *dataset* es multipropósito y contiene 255,510 imágenes en formato JPG. Las imágenes están anotadas en base a los siguientes aspectos:

- Calificación de estética. Calificación dada por los usuarios del sitio web de donde se obtuvieron las imágenes 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. 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. Contiene 14 estilos fotográficos asociados a 14,079 imágenes dentro del *dataset*. Los estilos que contempla son *Complementary colors*, *Duotones*, *High Dynamic Range*, *Image grain*, *Light on white*, *Long exposure*, *Macro*, *Motion blur*, *Negative image*, *Rule of thirds*, *Shallow DOF*, *Silhouttes*, *Soft focus*, y *Vanishing point*.

Debido a que el interés de este artículo se centra en la clasificación de estilos fotográficos, solo se contemplaron las imágenes anotadas con los 14 estilos fotográficos donde cada estilo se consideró como una etiqueta de clase distinta. No obstante, la clasificación de estilos fotográficos se abordó desde

Tabla 1. Distribución de las imágenes utilizadas en cada estilo fotográfico.

Estilo	Imágenes
Complementary Colors	760
Duotones	1,040
High Dynamic Range	315
Image Grain	671
Light on White	960
Long Exposure	674
Macro	1,357
Motion Blur	486
Negative Image	766
Rule of Thirds	1,111
Shallow DOF	1,183
Silhouettes	824
Soft Focus	566
Vanishing Point	540

una perspectiva multiclase (donde una fotografía pertenece a una sola clase); por lo que las imágenes del *dataset* que pertenecían a más de un estilo se descartaron, dando como resultado un total de 11,253 imágenes utilizadas para el entrenamiento de las CNNs. En la Tabla 1 se muestra la distribución de las imágenes utilizadas en cada estilo fotográfico.

3. Arquitectura propuesta

En primera instancia, la Subsección 3.1 describe el preprocesamiento realizado al conjunto de imágenes utilizado. En las siguientes secciones se presentan las arquitecturas implementadas. La Subsección 3.2 describe la implementación de una CNN simple. La Subsección 3.3 presenta el uso de tres CNNs preentrenadas, existentes en la literatura. La Subsección 3.4 muestra la arquitectura de tres CNN utilizando *transfer learning* a partir de las CNNs preentrenadas. Por último, la Subsección 3.5 describe el uso de estrategias para reducir el sobreajuste en el mejor modelo de los anteriores para incrementar su rendimiento.

3.1. Preprocesamiento de las imágenes

Las 11,253 imágenes que se tomarón del *dataset* AVA [9] se separaron en tres conjuntos, *Train*, *Test* y *Valid*. El 80 % de las imágenes se almacenaron en una carpeta llamada *Train*, el 10 % en una carpeta llamada *Test* y el resto en una carpeta llamada *Valid*. Dentro de estas carpetas, las imágenes se almacenaron en

Tabla 2. Imágenes utilizadas para el entrenamiento de las CNNs.

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	1,087	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

carpetas de acuerdo a su estilo fotográfico. La cantidad de imágenes utilizadas se muestran en la Tabla 2.

Posteriormente, los valores de los píxeles de las imágenes se normalizaron entre 0 y 1, en lugar de 0 y 255. Adicionalmente, las imágenes se redimensionaron a un tamaño de 256×256 píxeles utilizando el módulo *ImageDataGenerator* de *Keras* [13].

3.2. Red Neuronal Convolucional

Como primera instancia, se implementó una nueva CNN simple siguiendo algunos de los principios de diseño como los presentados en [14] y [15], tales como crecimiento gradual de la red, comenzar con filtros pequeños, utilizar filtros de tamaño impar, entre otros.

De entre las diferentes arquitecturas diseñadas, la que mejor resultado proporcionó estuvo compuesta por: 3 capas convolucionales con 32, 64 y 128 filtros; 1 capa de *max pooling* entre cada una de estas; 1 capa densa con 14 neuronas de salida, una por cada estilo fotográfico. En cuanto al optimizador se utilizó *Adam* con una tasa de aprendizaje de 0.00001. Cabe mencionar que los parámetros de las CNNs que se presentan en este artículo se definieron en base a prueba y error. La Figura 1 muestra la arquitectura de la CNN implementada.

Después de entrenar esta CNN durante 100 épocas, los resultados no fueron aceptables, ya que se alcanzó un *mAP* de 31.75 %. Además, el modelo comenzó a mostrar que se empezó a *sobreajustar* (overfitting) (ver Figura 2). Con esto se pensó que el bajo desempeño se pudo deber a que el número de imágenes utilizadas para entrenar la CNN no fueron suficientes.

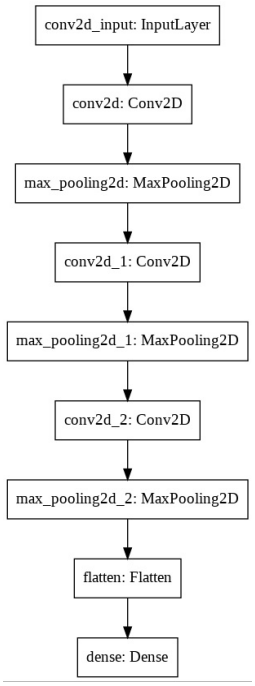

Fig. 1. Modelo de la CNN simple.

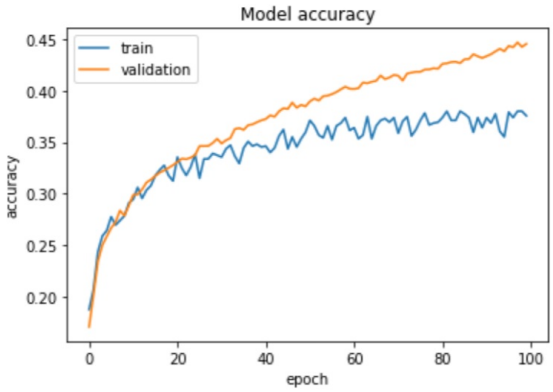

Fig. 2. Muestras de overfitting de la CNN simple.

3.3. Red Neuronal Convolucional preentrenada

Debido a que construir y entrenar nuevas CNNs requiere largos períodos de tiempo y grandes cantidades de imágenes, se optó por utilizar las CNNs

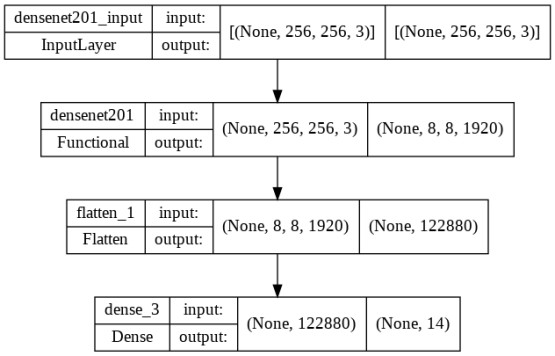

Fig. 3. Modelo basado en *DenseNet201*.

preentrenadas *VGG19* [16], *DenseNet201* [17], y *MobileNetV2* [18], existentes en la literatura.

Estos modelos se instanciaron desde *Keras* utilizando el módulo *applications* [19]. Para cada una de estas tres CNNs preentrenadas se creó un nuevo modelo que tiene como primera capa todo el contenido del modelo instanciado. Como salida de las tres CNNs se definió una capa densa de 14 neuronas, una tasa de aprendizaje (learning rate) de 0.0000065, se utilizó el optimizador *Adam* y se entrenaron por 12 épocas. Los modelos se entrenaron 12 épocas debido al indicio de overfitting mostrado en la Figura 2. El modelo que obtuvo el mejor desempeño fue *DenseNet201* con un *mAP* de 27.1 %. En la Figura 3 se presenta la arquitectura de esta CNN. Debido a que los resultados tampoco se consideraron como aceptables, se siguió pensando que posiblemente la causa fue el bajo número de imágenes.

3.4. Aprendizaje por transferencia

Con el fin de mejorar los resultados de clasificación que se habían obtenido hasta al momento, se utilizó la técnica de *aprendizaje por transferencia* (*transfer learning*) la cual consiste en utilizar un modelo previamente entrenado y '*transferir*' ese aprendizaje a un nuevo modelo [20].

Para implementar esta técnica se utilizaron los mismos modelos *VGG19*, *DenseNet201*, y *MobileNetV2*. Estos modelos de la literatura fueron preentrenados en *ImageNet* [21], por lo que se utilizaron los pesos aprendidos de ese *dataset*, los cuales son proporcionados por *Keras*.

Se puede implementar *transfer learning* de distintas maneras, por ejemplo: utilizar el modelo preentrenado 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 [20].

Después de realizar distintas pruebas con algunas de las estrategias mencionadas anteriormente, utilizar y entrenar el modelo completo sin 'congelar' ninguna de sus capas y mantener una tasa de aprendizaje de 0.0000065 fue la estrategia que dio mejores resultados. Nuevamente, al igual que en la etapa anterior, el modelo con el mejor desempeño fue *DenseNet201* con un mAP del 52.15 %.

3.5. Reducción del sobreajuste

Apesar de que el modelo basado en *DenseNet201 + transfer learning* obtuvo un mejor desempeño, se decidió aplicarle técnicas de reducción del *overfitting* con el fin de mejorar su desempeño.

En primera instancia, se utilizó la estrategia *dropout*, la cual consiste en establecer en 0 de manera aleatoria cierto porcentaje de las neuronas de una o más capas durante el entrenamiento de la red [24]. Para la implementación de esta técnica se utilizó la clase *Dropout* de *Keras*, la cual viene incluida en el módulo *layers*. Se 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 4. De esta manera, el modelo alcanzó un mAP de 59.77 %.

Posteriormente, se utilizó el *aumento de datos* (data augmentation), el cual consiste en generar nuevos datos a partir de los existentes [25]. Algunas de las opciones para aplicar *data augmentation* son girar la imagen, invertirla, proyectarla sobre sus ejes, hacer zoom, entre otras [26]. Debido a que algunas de estas estrategias podría afectar el estilo fotográfico de las imágenes, solo se hicieron dos transformaciones por cada imagen: proyecciones en su eje horizontal y proyecciones en su eje vertical. Estas dos transformaciones se utilizaron para crear dos nuevos conjuntos de imágenes. Para generar el primer conjunto se aplicaron a todas las imágenes, dando como resultado un total de 33,759 imágenes (completo). Para el segundo sólo se aplicaron a las clases con menos de 1,000 imágenes hasta alcanzar dicha cantidad con la finalidad de solo lograr una menor tasa de desbalance entre las clases, resultando un total de 14,691 imágenes (balanceado). Para aplicar *data augmentation* se utilizó la librería *OpenCV* [23]. En la Figura 5 se aprecia una imagen de ejemplo aplicando estas dos transformaciones. El modelo se entrenó nuevamente pero ahora utilizando cada uno de estos dos nuevos conjuntos de imágenes, alcanzando un mAP de 60.69 % en el primer conjunto y un 60.1 % en el segundo.

4. Resultados y discusiones

Para la implementación de los modelos descritos en la Sección 3 se utilizó el lenguaje de programación *Python* y las librerías *Keras* y *Tensorflow* por medio del servicio en la nube de *Google Colab*. El resto de esta sección se estructura de la siguiente manera. La Subsección 4.1 presenta los resultados de la evaluación

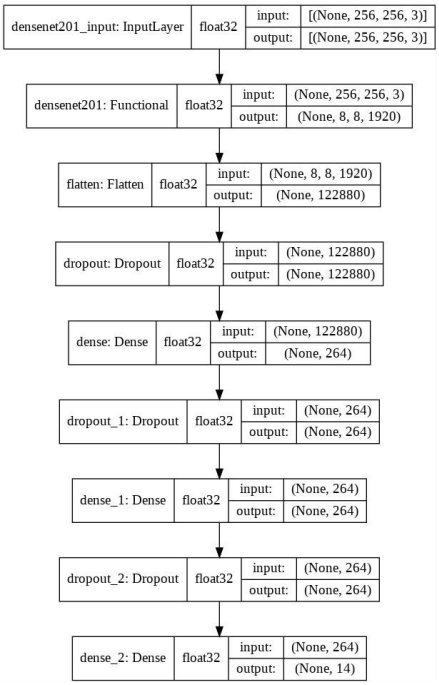

Fig. 4. Modelo *DenseNet201 + dropout + transfer learning*.

Fig. 5. Ejemplo de *data augmentation* de una imagen.

de los modelos desarrollados. La Subsección 4.2 presenta una discusión a partir de los resultados obtenidos.

4.1. Evaluación de los modelos desarrollados

En la Tabla 3 se presenta una comparativa del desempeño de los distintos modelos descritos en la Sección 3 en términos de la métrica *mAP*. La primera columna contiene el nombre del modelo entrenado; la segunda indica si se utilizó *transfer learning*; la tercera muestra el número de épocas utilizadas para entrenar cada uno de los modelos y, la última columna muestra el desempeño

Tabla 3. Resultados obtenidos de las distintas CNNs entrenadas.

Modelo	Transfer learning	Épocas	% mAP
CNN simple	No	100	31.75
VGG19	No	12	26.33
DenseNet201	No	12	27.1
MobileNetV2	No	12	12.19
VGG19	Imagenet	12	37.83
DenseNet201	Imagenet	12	52.15
MobileNetV2	Imagenet	12	42.41
DenseNet201 + dropout	**Imagenet**	**20**	**59.77**

Tabla 4. Resultados obtenidos del modelo *DenseNet201 + dropout + transfer learning + data augmentation.*

Modelo	Data augmentation	Épocas	% mAP
DenseNet201 + dropout	No	20	59.77
DenseNet201 + dropout	Balanceado	20	60.1
DenseNet201 + dropout	**Completo**	**10**	**60.69**

de cada modelo utilizando la métrica *mAP*. Como se puede observar, el modelo *DenseNet201 + dropout + transfer learning* obtuvo el mejor desempeño con un *mAP* de 59.77 %. Por otra parte, el modelo *MobileNetV2* obtuvo el peor desempeño con un 12.19 %.

La Tabla 4 muestra los resultados del mejor modelo, *DenseNet201 + dropout + transfer learning* utilizando los dos conjuntos de imágenes aumentados, descritos en la Subsección 3.5. Como se puede ver, haber entrenado éste modelo con ambos conjuntos de imágenes mejoran su desempeño con un *mAP* de 60.1 % y de 60.69 %.

La Figura 6 muestra la matriz de confusión del modelo *DenseNet201 + dropout + transfer learning + data augmentation* completo. Como se puede observar, las clases con un bajo número de predicciones correctas fueron *motion blur* con un 29 % y *rule of thirds* con 31 %. Por tal motivo, se realizó un análisis para determinar si el bajo desempeño de la red se debió a los datos.

El modelo confundió el estilo *motion blur* con el estilo *long exposure* posiblemente porque ambos estilos hacen referencia a movimiento dentro de la imagen [3], [1]. En la Figura 7 se aprecia la similitud entre estos estilos. Como se puede ver, ambos presentan movimiento, sin embargo, en *long exposure* no necesariamente hay un desenfoque completo, pues las teclas del piano están perfectamente enfocadas.

Por otra parte, *rule of thirds* también presentó confusión con todas las clases. Este estilo refiere a la posición en la que se coloca el sujeto principal de una

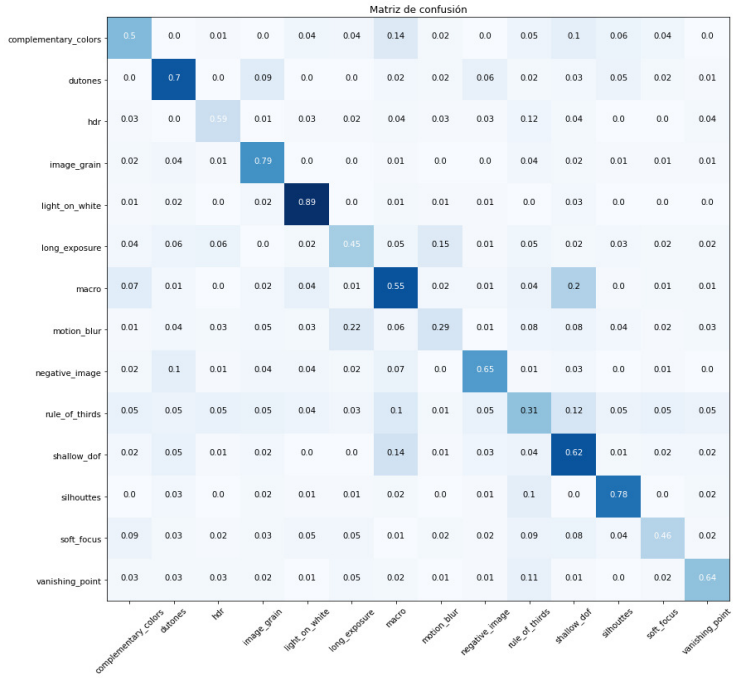

Fig. 6. Matriz de confusión del modelo *DenseNet201 + dropout + transfer learning + data augmentation* completo.

Fig. 7. a) Imagen izquierda, estilo *motion blur*. b) Imagen derecha, estilo *long exposure*

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 [27]. Este estilo puede confundir al modelo, 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 8 se muestra una imagen

Fig. 8. Imagen del estilo *shallow DOF* donde también se cumple el estilo *rule of thirds*.

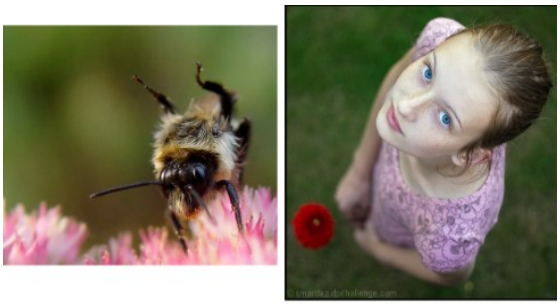

Fig. 9. Ejemplo del desenfoque del fondo en los estilos *macro* y *shallow DOF*.

que corresponde al estilo de *shallow DOF*, sin embargo, se observa que *rule of thirds* también se cumple.

Otro par de estilos en los que se puede apreciar confusión entre sí, son *macro* y *shallow DOF*, pues *macro* puede implicar una reducción de la profundidad de campo en algunos casos [28]. En la Figura 9 se aprecia cómo ambas imágenes tienen poca profundidad de campo, esto es, el efecto de desenfoque en el fondo [1].

Para verificar el impacto del 'ruido' que estos estilos podrían introducir al modelo, éste se entrenó nuevamente sin considerar los estilos *rule of thirds*, *motion blur*, y *shallow DOF*. En la Tabla 5 se muestra el *mAP* de los resultados obtenidos.

Como se puede observar, el modelo fue mejorando su desempeño conforme se fueron quitando cada uno de estos tres estilos, hasta alcanzar un *mAP* de 73.61 %. En la Figura 10 se muestra la matriz de confusión resultante al quitar los estilos *rule of thirds*, *motion blur* y *shallow DOF*.

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 validación cruzada denominado *stratified k-folds* [29]. Se utilizaron 10 folds

Tabla 5. Resultados de *DenseNet201 + dropout + transfer learning + data augmentation* completo sin los estilos *Rule of thirds, motion blur* y *shallow DOF*.

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**

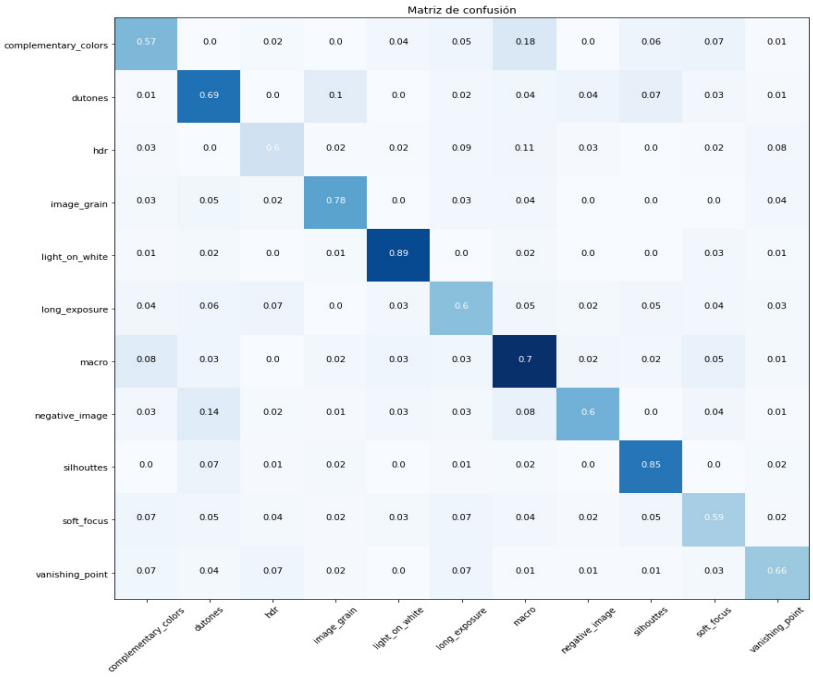

Fig. 10. Matriz de confusión del modelo *DenseNet201 + dropout + transfer learning + data augmentation* completo con 11 clases.

sobre el conjunto de datos aumentado completamente, los resultados se muestran en la Tabla 6.

4.2. Discusiones

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

Tabla 6. Resultados del modelo *DenseNet201 + dropout + transfer learning + data augmentation* completo utilizando 11 clases y validación cruzada de 10 folds.

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

Por otra parte, en cuanto a las técnicas utilizadas para reducir el *overfitting*, *dropout* tuvo mayor impacto positivo en el desempeño del modelo que al utilizar *data augmentation*. Lo anterior puede ser debido al dominio de aplicación, pues al aumentar los datos se pudo haber modificado el estilo de la imagen, impactando negativamente en el desempeño del modelo.

El desempeño alcanzado por el modelo pudo deberse a que algunos de los estilos no 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 quitar los estilos que introducían ruido, el modelo mejoró su desempeño, pasando de un mAP de 60.69 % a un 73.61 %.

5. Conclusiones y trabajos futuros

En este artículo 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.

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 implementar diferentes arquitecturas de CNNs para abordar clasificación multietiqueta en casos en los que una misma imagen pudiera pertenecer a más de un estilo fotográfico.

Referencias

1. M. Freeman, The photographer's mind, Lewes, UK: Elsevier, 2011.
2. A. P. Martínez Lanz Durán, Memorias: fotgrafía pictórica, tesis, Cholula, Puebla: Universidad de las Américas de Publa, 2003.
3. M. Freeman, El estilo en fotografía, Madrid, España: H. Blume, 1986.
4. Y. LeCun, Y. Bengio y Geoffrey Hinton, Deep learning, de Nature 521, 2015.
5. Roldán, A. K. M., Sánchez-Solís, J. P., Jiménez, V. G., Juárez, R. F., & Zárate, G. R. (2020). Convolutional Neural Network in a Pseudo-Distributed Environment for Classification of Chest X-Ray Images of Patients with Pneumonia. Res. Comput. Sci., 149(5), 101-110.
6. Song, G., Wang, Z., Han, F., Ding, S., & Iqbal, M. A. (2018). Music auto-tagging using deep recurrent neural networks. Neurocomputing, 292, 104-110.
7. Srivastava, S., Divekar, A. V., Anilkumar, C., Naik, I., Kulkarni, V., & Pattabiraman, V. (2021). Comparative analysis of deep learning image detection algorithms. Journal of Big Data, 8(1), 1-27.
8. Xu, Y., Wang, Y., Yuan, J., Cheng, Q., Wang, X., & Carson, P. L. (2019). Medical breast ultrasound image segmentation by machine learning. Ultrasonics, 91, 1-9.
9. 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.
10. S. Karayev, M. Trentacoste, H. Han, A. Agarwala, T. Darrell, A. Hertzmann y H. Winnemoeller, Recognizing Image Style, de British Machine Vision Conference, 2014.
11. Celona, L., Leonardi, M., Napoletano, P., & Rozza, A. (2022). Composition and style attributes guided image aesthetic assessment. IEEE Transactions on Image Processing, 31, 5009-5024.
12. 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.
13. Keras, ≪Image data preprocessing,≫ Keras, [En línea]. Available: https://keras.io/api/preprocessing/image/. [Último acceso: 10 Marzo 2021].
14. 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].

15. 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].
16. S. Karen y Z. Andrew, ≪Very deep convolutional networks for large-scale image recognition,≫ de International Conference on Learning Representations, Toulon, France, 2015.
17. 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.
18. 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.
19. Keras, ≪Keras Applications,≫ Keras, [En línea]. Available: https://keras.io/api/applications/. [Último acceso: 10 Enero 2021].
20. 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 2021].
21. 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].
22. D. Rodvold, A software development proccess model for artificial neural networks in critical applications, de International Joint Conference on Neural Networks, Washington, DC, 2002.
23. OpenCV, OpenCV modules, OpenCV, [En línea]. Available: https://docs.opencv.org/master/. [Último acceso: 10 Abril 2021].
24. Keras, Dropout layer, Keras, [En línea]. Available: https://keras.io/api/layers/regularization_layers/dropout/. [Último acceso: 14 Enero 2021].
25. Rondón, C. V. N., Carvajal, D. A. C., Casadiego, S. A. C., Delgado, B. M., & Ibarra, D. G. Dataset para la detección de elementos de bioseguridad facial mediante técnicas de aprendizaje computacional.
26. TensorFlow, Aumento de datos, TensorFlow, 19 Marzo 2021. [En línea]. Available: https://www.tensorflow.org/tutorials/images/data_augmentation. [Último acceso: 10 Abril 2021].
27. 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.
28. R. Sheppard, Macro Photography From Snapshots To Great Shots, Peachpit Press, 2015.
29. 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].