Universitat
de les Illes Balears

TRABAJO DE FIN DE MÁSTER

RECONOCIMIENTO AUTOMÁTICO DE EMOCIONES EN CONDICIONES REALES A PARTIR DE IMÁGENES Y AUDIO

Caterina E. Muntaner González

Máster Universitario en Sistemas Inteligentes (MUSI)

Especialidad: Inteligencia Artificial

Centro de Estudios de Posgrado

Año Académico 2020-21

RECONOCIMIENTO AUTOMÁTICO DE EMOCIONES EN CONDICIONES REALES A PARTIR DE IMÁGENES Y AUDIO

Caterina E. Muntaner González

Trabajo de Fin de Máster
Centro de Estudios de Posgrado
Universidad de las Illes Balears

Año Académico 2020-21

Palabras clave del trabajo:
Emotion Recognition in the Wild, Facial Emotion Recognition, Speech Emotion Recognition ...

Nombre Tutor/Tutora del Trabajo: Xavier Varona Gómez

Reconocimiento automático de emociones en condiciones reales a partir de imágenes y audio

Caterina E. Muntaner González
Tutor: Xavier Varona
Trabajo de fin de Máster Universitario en Sistemas Inteligentes (MUSI)
Universitat de les Illes Balears
07122 Palma, Illes Balears, Espanya
caterinamuntaner1@estudiant.uib.es

***Resumen*—En este trabajo de fin de máster se han estudiado y analizado las dificultades de los sistemas de reconocimiento automático de emociones en condiciones reales y se han comparado con el reconocimiento de emociones en condiciones controladas. Se ha usado como dataset de imágenes y audio en condiciones reales el dataset *AFEW* y se han implementado y evaluado modelos unimodales basados en redes convolucionales, estudiando cada tipo de entrada de manera independiente. Además, se han explorado las posibilidades del *transfer learning* para este tipo de sistemas. Finalmente, con el objetivo de estudiar si la incorporación del audio a un sistema basado en imágenes o vídeo comportaba una mejora de los resultados, se han analizado e implementado distintos modelos de combinación. El sistema combinado propuesto ha obtenido una accuracy del 45 %, mejorando los resultados de los modelos unimodales y demostrando que la incorporación de la señal de audio al modelo resulta positiva.**

ABSTRACT

In this Master's Thesis, the challenges of automatic emotion recognition systems in the wild have been studied and analyzed and they have been compared to automatic emotion recognition in controlled conditions. The *AFEW* dataset has been used as an image and audio in the wild dataset. Unimodal models based on convolutional neural networks have been implemented and evaluated studying each kind of input independently. Furthermore, the possibilities of *transfer learning* for this kind of system have been explored. Finally, with the aim of studying whether the incorporation of audio signal into an image or video based system entailed an improvement in the results, different combination models have been analyzed and implemented. The proposed combined model obtained an accuracy of 45 % proving that the incorporation of the audio signal to the model is positive.

***Index Terms*—Emotion Recognition in the Wild, Facial Emotion Recognition, Speech Emotion Recognition, Convolutional Neural Networks**

I. INTRODUCCIÓN

I-A. Motivación

Las emociones son un componente esencial de la vida y la comunicación humana. Una parte muy importante del mensaje que comunicamos se corresponde con factores emocionales como la expresión, los gestos, el tono o la forma de articular el mensaje. Todo esto hace que la identificación de las emociones sea fundamental para la interacción y la comprensión de la comunicación interpersonal.

Durante los últimos años, con el auge de la inteligencia artificial se han desarrollado multitud de algoritmos y técnicas para el reconocimiento de emociones. Este tipo de algoritmos abre la puerta a una interacción más natural persona-máquina y tiene multitud de aplicaciones, especialmente en el campo de la interacción persona ordenador.

Identificar emociones es una tarea ardua que a veces resulta difícil incluso para usuarios humanos. Durante los últimos años se han tomado diferentes enfoques para lograrlo. Se pueden distinguir distintos grupos de algoritmos en función del tipo de entrada que utilizan para identificar las emociones (imágenes faciales, audio,texto, señales físicas como electrocardiogramas...) y las técnicas utilizadas para procesar los datos y implementar el clasificador. En este trabajo nos centraremos en la detección de emociones a partir de imagen y audio.

Los avances en el campo del *deep learning* producidos en los últimos años han motivado que se desarrollen multitud de algoritmos de detección de emociones basados en redes neuronales que han demostrado mejorar los resultados de otros enfoques más clásicos.

En este TFM se estudiarán e implementarán distintos modelos de reconocimiento automático de emociones *in the wild* basados en *deep learning* tanto para imágenes, como para audio y se analizará si la combinación de ambas entradas puede contribuir a una mejora de los resultados.

I-B. Datasets

La identificación de emociones a partir de imágenes y audio es un problema complejo incluso para los humanos. Es por ello que, históricamente, los sistemas y modelos implementados trabajaban con datos generados en condiciones de laboratorio. Este tipo de modelos ha contribuido enormemente al desarrollo y mejora de la detección automática de emociones pero suponen una simplificación de la realidad.

Durante los últimos años, con el aumento de la cantidad de datos audiovisuales disponibles y la necesidad trabajar en condiciones más próximas a la realidad, se han creado una serie de datasets en condiciones reales o datasets *in the wild*.

(a) Angry (b) disgust (c) happy

(d) sad (e) surprise (f) fear

Figura 1: Ejemplo de imágenes etiquetadas del dataset SFEW

En este tipo de datasets de entorno no controlado se trabaja con unas condiciones más complejas y cercanas a las que se dan en las interacciones reales. Este campo de estudio se encuentra en pleno desarrollo y ha evolucionado sustancialmente durante los últimos años.

Para este trabajo se ha elegido el dataset AFEW2018/AFEW6.0 [8]. El AFEW 2018 es una versión del dataset AFEW [6] formado por clips de películas y series de televisión de entre 300 y 5400 ms [8] clasificados en 7 clases distintas, las 6 emociones básicas y neutralidad (enfado, asco, miedo, felicidad, tristeza, sorpresa y neutralidad). Está dividido en 3 particiones: entrenamiento, validación y test que contienen 773 muestras, 383 muestras y 636 muestras respectivamente. Debido a que el dataset usado forma parte de una competición la partición de test no viene etiquetada y por tanto no se ha usado para este trabajo.

El AFEW es un dataset que fue creado originalmente en 2012 con el objetivo de ofrecer una base de datos que se acercara más a la realidad que las que existían en su momento, que eran en su mayoría datasets creados en condiciones controladas de laboratorio. Como se explica en el artículo de Dhall et al. [6] la base de datos fue creada a partir de un proceso semiautomático dividido en dos pasos. Primero, un proceso automático seleccionaba clips de películas conocidas en base a una búsqueda de palabras clave relacionadas con las emociones en los subtítulos (reír, llorar,contento, triste...) y después estos clips seleccionados y etiquetados eran revisados (y reetiquetados en caso de ser necesario) por un conjunto de anotadores humanos.

A partir del AFEW también se generó otro dataset, que también se utilizará en este trabajo, formado por imágenes estáticas obtenidas a partir de *frames* del AFEW conocido como SFEW (Static Facial Expressions In The Wild). La descripción del proceso de creación del SFEW puede encontrarse en el artículo de Dhall et al. [7]. En la figura 1 se pueden ver ejemplos de imágenes del SFEW.

El AFEW y el SFEW han sido utilizados durante los últimos años en la competición EmotiW (Emotion Recognition In The Wild Challenge and Workshop). La EmotiW es una competición creada en 2013 con el objetivo de ofrecer una plataforma donde se pudieran evaluar distintos métodos de reconocimiento de emociones en condiciones similares a las del mundo real [5]. Para ello, se seleccionó la base de datos AFEW puesto que, a pesar de no ser una base de datos 100 % de condiciones reales porque contiene emociones actuadas, sí que se puede considerar una base de datos *in the wild* o de condiciones similares a las del mundo real puesto que emula situaciones reales sin las limitaciones que tienen los datasets generados en condiciones controladas de laboratorio.

La EmotiW se ha celebrado cada año ininterrumpidamente desde 2013. Con la evolución de las ediciones se han introducido nuevos retos como la predicción de la emoción grupal o del *student engagement*, además del reto inicial de reconocimiento de emociones a partir de imagen y audio que se ha mantenido durante todas las ediciones. Para cada edición de la competición se publica un *baseline* del reto que sirve como punto de partida o "barra de medir"para los participantes. Este *baseline* ha ido cambiando y evolucionando con los años.

En la primera edición, en 2013 los resultados de referencia (baseline) de la competición se obtuvieron a partir de un vector de características LBT-TOP que se pasaban a una SVM no lineal que aprendía a clasificar las emociones para el vídeo y con características extraídas con openSmile y una SVM lineal para el audio. Finalmente, también se presentaba un sistema que fusionaba ambos sistemas concatenando los dos tipos de características y pasándoselas a una SVM no lineal [5]. El clasificador de vídeo logró una *accuracy* del 27.2 % con el set de validación y de un 22.7 % con el set de test, el clasificador de audio logró un 19.5 % de *accuracy* con el conjunto de validación y un 22.2 % con el de test y, finalmente el sistema combinado logró un 22.2 % con el set de validación y un 27.2 % con el conjunto de test.

En la edición de 2018, la edición de la que se ha obtenido el dataset que se usará para este trabajo, el *baseline* presentado se obtuvo con un método similar que también usaba LBT-TOP features clasificados con una SVM con un *Chi-square Kernel*. En esta edición los resultados base del clasificador de vídeo fueron de un 38.81 % para el test de validación y del 41.07 % para el set de test [8]. Estos resultados con precisiones tan bajas pueden dar una idea de la complejidad del problema y del dataset.

En este trabajo se trabajará tanto con el SFEW como con el AFEW y se estudiarán tanto las imágenes como la señal de audio. El TFM está estructurado de la siguiente forma: en la sección II se explica el marco teórico en el que se basan los modelos desarrollados en este trabajo; en la sección III, se explican los modelos desarrollados para imagen, tanto para las imágenes estáticas como para el vídeo; en la sección IV se exponen los modelos implementados para el audio; en la sección V se presenta el modelo combinado y, finalmente, en la sección VI se exponen las conclusiones del trabajo. El código y los resultados de este TFM pueden consultarse en el repositorio https://github.com/Caterina1996/SFEW_dataset.git.

II. Redes convolucionales

El *deep learning* es un subcampo del *machine learning* basado en utilizar conjuntos de capas sucesivas que permiten aprender diferentes representaciones o características significativas de los datos en cada capa. El término *deep* hace referencia a la profundidad del modelo: cuantas más capas tiene un modelo más profundo es y más capacidad tiene de aprender representaciones complejas de los datos.

La mayoría de estos modelos se engloban en lo que conocemos como redes neuronales. Las redes neuronales están formadas por capas sucesivas de nodos llamadas neuronas que implementan un tipo de función determinada, llamada función de activación, que aplican a la entrada que reciben para producir una salida.

El objetivo de la red es encontrar una combinación de valores de los parámetros de la red tal que permita que el algoritmo desempeñe con éxito su tarea. La red debe ser capaz de aprender cuáles son estos valores óptimos de los parámetros a partir de ejemplos de las salidas que debería obtener para los datos de entrada de entrenamiento, hasta ser capaz de predecir correctamente la salida para entradas que no haya visto nunca.

Este proceso de aprendizaje de los pesos por exposición a ejemplos es lo que se conoce como entrenamiento de la red. La red parte de unos valores iniciales de los pesos, frecuentemente aleatorios, y los va corrigiendo hasta llegar a encontrar los mejores pesos o unos suficientemente buenos. Para lograrlo, es necesario que la red sea capaz de medir su desempeño y saber cómo de bien funciona una determinada combinación de pesos. Para ello, se utiliza la función de error o función de pérdida. Dicha función calcula una medida de distancia entre la predicción de la red y el *groundtruth* (la salida deseada para esa entrada), es decir, calcula cómo de bien o mal la red ha predicho la salida para una entrada conocida en concreto.

Este error se utiliza para ajustar el valor de los pesos de la red en la dirección que minimice el error del ejemplo actual. De este ajuste de los valores de los pesos se encarga el optimizador que implementa un proceso llamado *backpropagation* que es la esencia del aprendizaje de las redes neuronales.

El proceso de *backpropagation* o propagación hacia atrás, funciona de la siguiente forma: se calcula el vector gradiente a partir de las derivadas parciales según cada uno de los parámetros que nos indica la dirección en la que se encuentra el mínimo global. Se corrige el valor de los pesos dando un paso en la dirección del gradiente de tamaño α, donde α es el ratio de aprendizaje.

Este proceso de evaluación y corrección del valor de los parámetros se conoce como bucle de aprendizaje y se repite un número de iteraciones n, suficientemente grande hasta que los pesos adquieran unos valores tales que minimicen la función de pérdida o que permitan un desempeño lo suficientemente bueno.

Existen multitud de arquitecturas de redes y de tipos de capas. Uno de los tipos de redes más típicas, debido a su gran utilidad y buenos resultados en el campo de la visión por computador, son las redes convolucionales. Dado que en este trabajo se trabajará con imágenes este es uno de los tipos de red neuronal que se utilizará.

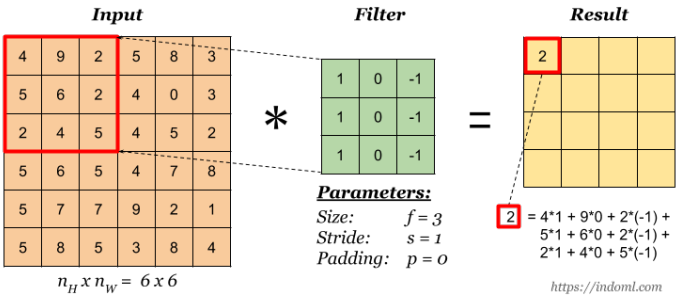


Figura 2: Operación de convolución

Las redes convolucionales reciben su nombre de su capa más característica, la capa convolucional que implementa la función de convolución. Las capas convolucionales son muy útiles para encontrar patrones dentro de las imágenes y permiten implementar filtros de distintos tipos. Normalmente, los filtros de las primeras capas implementan filtros básicos como detección de líneas o bordes y a medida que nos adentramos en la red la complejidad de las características que detectan los filtros aumenta.

Las capas convolucionales están formadas por conjuntos de filtros de pequeñas dimensiones normalmente 3x3 o 5x5. Entre el filtro y la imagen de entrada, que se suele representar en forma de matriz (tridimensional en el caso de imágenes en color o bidimensional para imágenes en blanco y negro), se aplica la función de convolución. Esta operación puede entenderse como una especie de producto escalar entre el filtro y una región de la imagen a la que se aplica dicho filtro que se mueve a través de la imagen. En las capas convolucionales el filtro se va moviendo a través de la imagen aplicando la convolución y de esta forma se va generando la nueva imagen. El número de posiciones que avanza el filtro sobre la imagen en cada desplazamiento es lo que se conoce como *stride*.

En ocasiones, para que los píxeles de los laterales no tengan menos representación que los centrales, se agranda la imagen de entrada añadiendo un número determinado de filas y columnas de ceros. La cantidad de filas y columnas que se añaden a la imagen original recibe el nombre de *padding*.

En la figura 2 puede verse un ejemplo de convolución. La imagen resultante de la convolución resalta las regiones de la imagen original en que se ha encontrado la característica que quería resaltar el filtro en cuestión.

Normalmente, a medida que se avanza dentro de la red convolucional, el número de filtros de las capas aumenta y en consecuencia la profundidad de las imágenes resultantes. Dado que en las redes neuronales frecuentemente interesa ir reduciendo el tamaño del input para que el número de parámetros no se dispare, en las redes neuronales convolucionales se usa normalmente otro tipo de capa llamada la capa de *pooling*.

La capa de *pooling* divide la imagen de entrada en diferentes subregiones y aplica una función a cada una de estas regiones para obtener de ellas un único valor que las represente y de esta forma reducir el tamaño de la imagen. Los dos tipos de *pooling* más típicos son el *MaxPooling* y el *AveragePooling*. El *Maxpooling* se queda con el máximo de cada región y el *AveragePooling* con la media. En la figura 3 se puede ver un

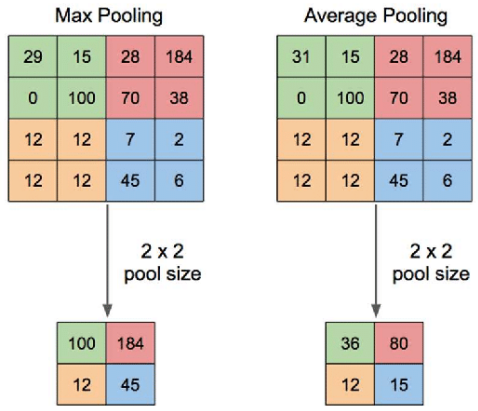


Figura 3: Ejemplo de MaxPooling y AveragePooling

ejemplo de ambos tipos de *pooling*.

Finalmente, en las redes convolucionales también se utiliza una capa llamada capa de *Flatten* o aplanamiento, que convierte la matriz bidimensional o tridimensional que recibe en un vector unidimensional. Esta capa es necesaria debido a que las últimas capas de una red neuronal convolucional suelen ser capas densas (capas estándar simples) que como entrada esperan vectores unidimensionales.

Cabe destacar que además de estos 3 tipos de capas básicos las redes convolucionales pueden contener también otro tipo de capas como las de *batchnormalization* o las de *dropout* entre muchas otras.

Existen algunas arquitecturas de redes convolucionales muy populares como *LeNet*, *AlexNet* o *VGG16* que se usan en multitud de problemas de visión distintos, especialmente para aplicar *transfer learning*.

De acuerdo con la definición de Torrey y Shavlik, el *transfer learning* es "la mejora del aprendizaje en una nueva tarea mediante la transferencia de conocimientos de una tarea relacionada que ya se ha aprendido"[22].

En el contexto del *deep learning* el *transfer learning* es una técnica muy extendida y eficaz a la hora de afrontar problemas para los que se tienen pocos datos. El *transfer learning* permite utilizar modelos preentrenados con datasets grandes y utilizarlos para problemas distintos para los que han sido entrenados. Las redes preentrenadas han sido típicamente entrenadas con datasets de gran tamaño, así que las características visuales genéricas que han aprendido pueden ser útiles para problemas de visión distintos.

Según Chollet [2], existen dos formas de utilizar redes preentrenadas:

- feature extraction: Esta técnica consiste en utilizar la red preentrenada excepto las últimas capas densas (base convolucional) para obtener vectores de características que se usan como entrada de un nuevo clasificador que es entrenado a partir de la salida del modelo preentrenado. El modelo preentrenado se congela y sólo se entrenan los pesos del nuevo clasificador añadido al final del modelo preentrenado congelado.
- fine-tuning: El fine-tuning consiste en reentrenar algunas de las últimas capas de la base convolucional del modelo preentrenado además de las capas del nuevo clasificador añadidas.

El *transfer learning* resulta muy útil para datasets pequeños como el dataset objeto de este trabajo. Es por ello que entre las arquitecturas utilizadas se encuentra la VGG16 [20] preentrenada con *Imagenet* que se ha utilizado para aplicar *transfer learning* tanto con las imágenes como con el audio.

Para implementar estas arquitecturas se han utilizado dos de las librerías más extendidas para la construcción de redes neuronales y modelos de *machine learning* como son *keras* y *tensorflow*.

Además, para realizar los entrenamientos se ha usado la plataforma *google colaboratory*, un entorno de programación de google en formato de cuadernos de *Ipython* que permite usar CPUs, GPUs y TPUs de google y guardar y ejecutar los cuadernos en la nube. Se ha seleccionado esta herramienta precisamente porque el acceso a los recursos de las GPUs de google, aunque limitado, permite realizar los entrenamientos de forma más rápida y eficiente.

III. Reconocimiento de emociones a partir de imágenes

Como se ha mencionado anteriormente el reconocimiento de emociones es un campo en auge con infinidad de aplicaciones, especialmente en el campo de la interacción persona-ordenador. La identificación de emociones a partir de imágenes es uno de los campos más explorados ya que la información visual es el principal canal de información a la hora de identificar emociones en las interacciones humanas.

A pesar de que se ha avanzado mucho en los últimos años, el reconocimiento de emociones a partir de imagen sigue siendo un campo en pleno desarrollo pues los algoritmos desarrollados hasta ahora aún quedan lejos de lograr un buen desempeño en entornos reales o próximos a lo que podría ser una interacción humana natural. Es por ello que durante los últimos años se han recopilado nuevos datasets *in the wild* con el objetivo de mejorar el reconocimiento de emociones en condiciones reales.

Tradicionalmente los algoritmos de reconocimiento de emociones a partir de imagen (imágenes estáticas o vídeo) se basaban en la pre-extracción de características predeterminadas (LBP, Action Units, Gabor filters...) [21]. En general, este tipo de algoritmos se centraban en la clasificación de imágenes faciales y seguían la estructura siguiente:

1. Detección de componentes faciales: Dada una imagen de entrada se detecta la región correspondiente a la cara y sus componentes(ojos, boca, nariz...) o puntos de referencia. Los puntos de referencia son puntos característicos que destacan en la cara como el final de las cejas o de la nariz.
2. Extracción de características: Se extraen las características que utilice el algoritmo concreto. Estas características suelen ser de apariencia o geométricas. En el caso de las geométricas el vector de características se construye a partir de la relación de los componentes faciales. Las características de apariencia, por otro lado, se obtienen de la región global de la cara o de distintas regiones

que contienen diferente tipo de información y tienen distintos grados de importancia. Cabe destacar que también existen características híbridas que combinan los dos tipos anteriores de características.

3. Clasificación de la expresión facial: Utilizando un clasificador previamente entrenado se clasifica la expresión facial. Para ello se utilizan clasificadores como máquinas de vectores de soporte (SVM), AdaBoost o random forest.

Este tipo de modelos fueron los predominantes durante las últimas décadas pero, con la creación de nuevos datasets y el avance de las redes neuronales, los modelos basados en *deep learning* han demostrado obtener mejores resultados y se han convertido en el paradigma dominante.

Los modelos basados en *deep learning* se fundamentan en que la red aprenda sus propias representaciones de los datos de entrada y por tanto, no dependen tanto de tener un conjunto de características predeterminado adecuado o de los modelos físicos de la cara como en el caso de los modelos anteriores.

Principalmente se utilizan 2 tipos de redes: las convolucionales y las recurrentes. Para la detección de imágenes estáticas las redes convolucionales son las más extendidas debido a su buen desempeño para la clasificación de imágenes. En el caso de las secuencias de vídeo, a parte de las redes convolucionales, se utilizan frecuentemente redes neuronales recurrentes para tratar con la componente temporal de los vídeos. Es muy frecuente también utilizar combinaciones de estos 2 tipos de redes como se puede ver en los trabajos de Kahou et al. [11] o utilizar Redes convolucionales 3D como en el trabajo de Fan et al. [12].

Para este trabajo nos centraremos en las redes convolucionales tanto para el estudio de las imágenes estáticas como para el del vídeo.

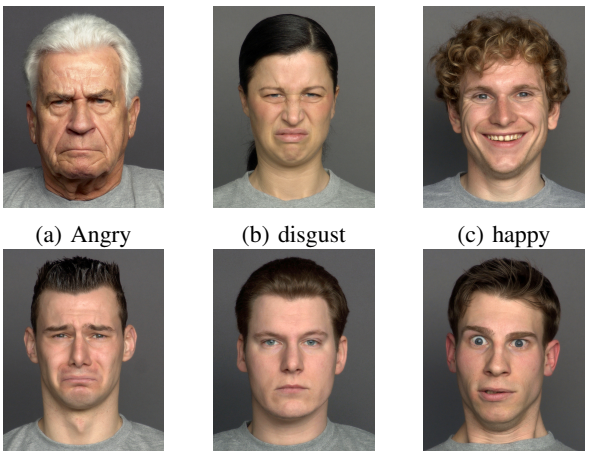
Para afrontar el problema de la clasificación de emociones a partir de imagen se ha estudiado el problema desde dos puntos de vista distintos. Primero se han implementado varios modelos para la clasificación de emociones con imágenes estáticas y luego se ha utilizado este modelo de base para la clasificación de emociones a partir de vídeo.

III-A. Modelo estático

Para la clasificación de emociones a partir de *frames* se han implementado 2 modelos distintos. En primer lugar, un modelo base que servirá de *baseline* que se evaluará con un dataset de condiciones controladas llamado FACES y posteriormente con el SFEW. En segundo lugar, se ha implementado un modelo basado en *transfer learning* con el objetivo de mejorar los resultados de este primer modelo.

Como se ha mencionado anteriormente, el dataset objeto de este trabajo (*EMOTIW2018*), cuenta con un conjunto de vídeos y uno de imágenes estáticas extraídas de los *frames* de dichos vídeos. El dataset de imágenes estáticas es el SFEW (Static facial emotion in the wild) y es el que se ha utilizado para evaluar este modelo estático.

Antes de realizar las primeras pruebas con SFEW, se decidió realizar la implementación de los modelos de *deep learning* con un dataset más sencillo obtenido en condiciones controladas de laboratorio como es la base de datos FACES con el objetivo de observar los comportamientos de los modelos usados con diferentes datasets y comprender con más detalle la complejidad del problema.



(a) Angry (b) disgust (c) happy

(d) sad (e) neutral (f) fear

Figura 4: Ejemplo de imágenes de cada emoción del dataset FACES

FACES es un dataset formado por imágenes de rostros de personas jóvenes, de mediana edad y mayores. Cada rostro está representado con dos conjuntos de seis expresiones faciales (neutralidad, tristeza, asco, miedo, ira y felicidad). El dataset contiene 171 imágenes para cada emoción, tanto para el conjunto de test como para el de entrenamiento, resultando en un total de 2.052 imágenes individuales [10]. En la figura 4 se puede ver un ejemplo de imágenes del dataset FACES de cada una de las emociones que contiene.

Así pues, como modelo de partida se implementó una variación de la arquitectura descrita en el artículo de Alizadeh y Fazel [19] y que también fue usada en el trabajo de Comparini [3]. La arquitectura utilizada puede verse en la figura 6 y consta de las siguientes capas:

- Una capa convolucional con 32 filtros de tamaño 3x3 y función de activación ReLu
- Una capa de MaxPooling con un filtro de tamaño 2x2
- Una capa de BatchNormalization
- Una capa de Dropout con un porcentaje del 25 %
- Una capa convolucional con 64 filtros de tamaño 3x3 y función de activación ReLu
- Una capa de BatchNormalization
- Una capa de Dropout con un porcentaje del 50 %
- Una capa de MaxPooling con un filtro de tamaño 2x2
- Una capa de Flatten
- Una capa densa de 512 neuronas y activación ReLu
- Una capa densa con 6 neuronas y activación softmax

Las imágenes originales se redimensionaron a un tamaño de 128 x 128 y se convirtieron a blanco y negro puesto que el color no debería aportar ninguna información útil a la hora de identificar las emociones.

Después de probar con diferentes valores para los hiperparámetros la combinación final utilizada fue la siguiente: se entrenó la red durante 50 epochs con un *learning rate* de

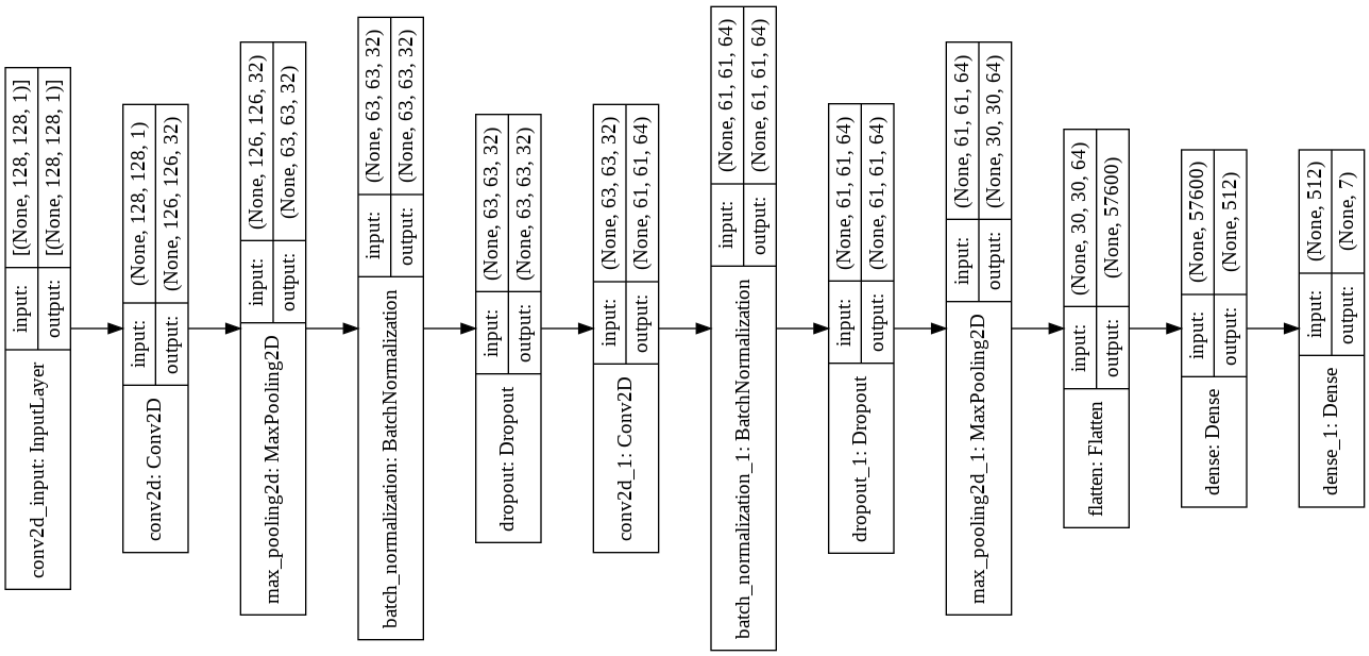


Figura 5: Modelo estático 1

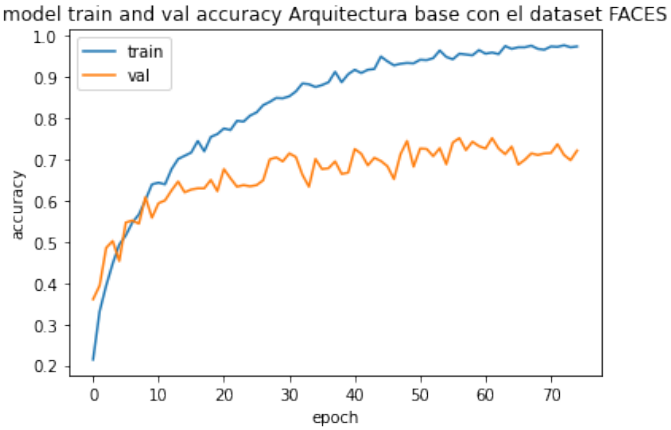


Figura 6: Train y validation *accuracy* obtenida por el modelo entrenado con el FACES

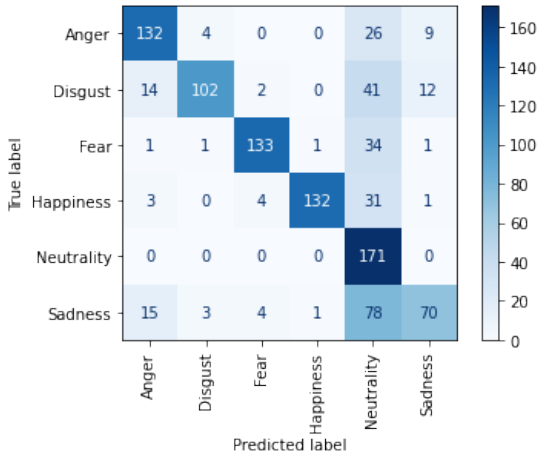


Figura 7: Resultados de aplicar la arquitectura 1 entrenada con el FACES al set de test del FACES

0.00001 y un *batch size* de 1.

Para preparar los datos se creó el script *tidyFaces.py* que genera el archivo *tidy.bat* que se ejecuta para ordenar las imágenes en 2 carpetas (test y train) que contienen 6 carpetas cada una correspondientes con las 6 emociones del FACES. Los datos se ordenaron de esta forma dado que se ha utilizado la función *ImageDatagenerator* de keras para cargar los datasets con el objetivo de facilitar la posibilidad de realizar *data augmentation* en caso de desearlo.

Con esta arquitectura se obtuvo una *accuracy* de alrededor del 70 %, un resultado similar al obtenido en el trabajo de Comparini [3]. En la figura 6 se puede ver la evolución de la *accuracy* obtenida con el conjunto de entrenamiento y el de test durante el entrenamiento.

Se observó que, como puede verse en la matriz de confusión de la figura 7, el modelo obtenido predice mayormente bien la mayoría de las emociones, no tiene problemas especiales con ninguna emoción aunque sí que presenta alguna dificultad con la tristeza. También se observó como la clase más escogida es la neutral, probablemente debido la ausencia de expresividad del sujeto en estos casos.

Una vez probado el primer modelo implementado con el dataset FACES, se evaluó el modelo con el dataset *in the wild* SFEW con el objetivo de determinar qué desempeño puede tener un modelo entrenado con un dataset en condiciones preparadas cuando es expuesto a imágenes de un dataset *in the wild*.

Se obtuvieron los resultados de la matriz de confusión de la figura 8 donde puede observarse que el modelo, que tenía

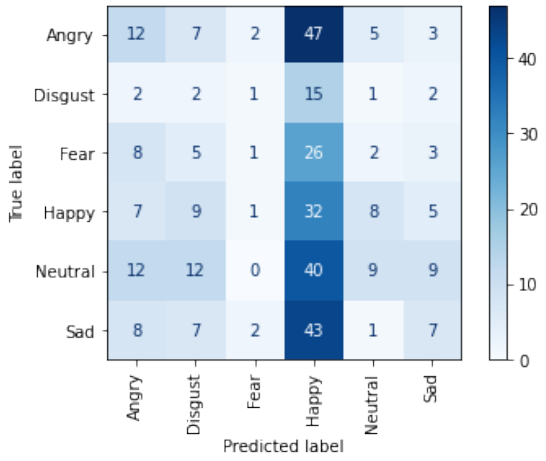


Figura 8: Resultados de aplicar la arquitectura 1 al set de test del SFEW

un buen desempeño con el FACES, clasifica la mayoría de imágenes del SFEW como felices.

El hecho de que este primer resultado no sea correcto nos da una idea de que el SFEW no parece ser un dataset sencillo y de que hay una clara diferencia con el FACES. Esta clara diferencia radica en que el FACES es un dataset obtenido en condiciones controladas y homogéneas (las caras están siempre perfectamente alineadas, la iluminación es siempre la misma para todas las imágenes, el fondo es invariante...) en cambio el SFEW se ha construido a partir de imágenes extraídas de segmentos de películas por lo que que las condiciones son mucho más complejas.

Se decidió reentrenar el modelo con el SFEW y se probaron diferentes valores de *learning rate*, *batch size* y también distintos optimizadores. Aun así, con ninguna combinación de parámetros se logró superar un *accuracy* de alrededor del 25 %, lo que es un resultado que permite un amplio grado de mejora.

Dado que los parámetros aprendidos por la red con el entrenamiento del FACES no parecían contribuir al buen desempeño del modelo con el SFEW, se decidió entrenar la arquitectura de cero con el SFEW para comprobar si, de esta forma, se lograban mejorar los resultados. Se obtuvo una *accuracy* del 25 % con el set de test, similar a la obtenida en la prueba anterior. La evolución de la *accuracy* de entrenamiento y test durante el entrenamiento puede verse en la figura 9 y la matriz de confusión obtenida en la figura 10.

De las pruebas anteriores, se observaron una serie de problemas o dificultades que presenta el dataset en condiciones reales. La primera y más obvia es que se trata de un dataset *in the wild* y por tanto, las imágenes del SFEW contienen imagenes que no están alineadas y que presentan diversidad de posturas de la cara, iluminación y tipo de fondo. Además existen algunas muestras que no se han detectado bien y no se corresponden con caras. Las imágenes de este tipo que se detectaron durante el entrenamiento fueron eliminadas manualmente puesto que no aportaban ninguna información útil y no ayudaban al aprendizaje del modelo.

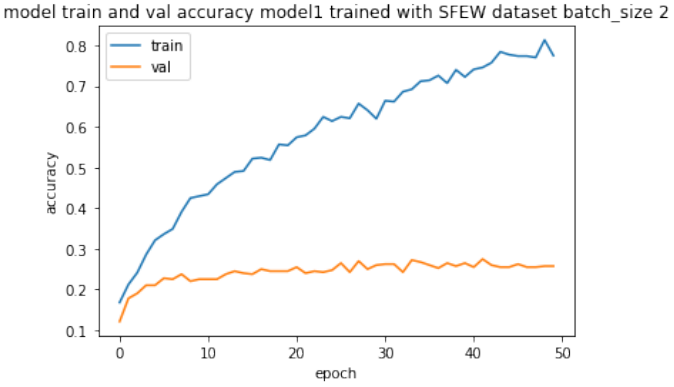


Figura 9: Resultados de entrenar el modelo con el SFEW

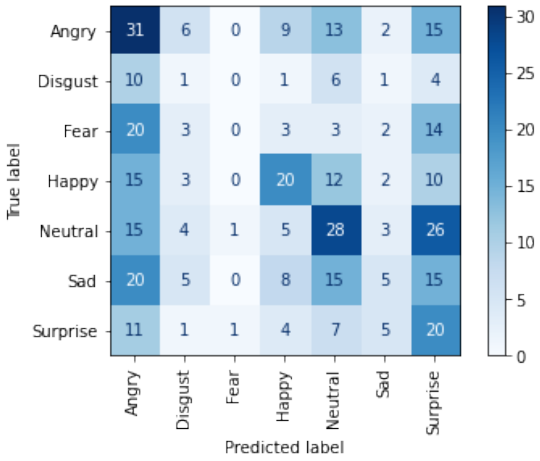


Figura 10: Matriz de confusión obtenida con el modelo estático reentrenando la red con el dataset SFEW

La segunda dificultad es que, como puede verse en la figura 11, el dataset está desbalanceado. Esto significa que la base de datos no tiene el mismo número de imágenes para cada emoción lo que puede dificultar el aprendizaje y podría provocar que la red tienda a clasificar con mayor frecuencia las imágenes como imágenes de la clase más representada y menos veces como la clase menos representada.

Por último, el tamaño del dataset también es una dificultad puesto que es un dataset muy pequeño (859 imágenes de entrenamiento y 405 imágenes de test después de eliminar las imágenes erróneas).

Para intentar afrontar este último problema, se decidió aplicar *data augmentation* de manera que se pudiera ampliar el tamaño del dataset. La aumentación de datos o *data augmentation* consiste en generar imágenes similares a las del dataset original con tal de multiplicar el número de muestras de entrenamiento y evitar en cierto grado el *overfitting*. Para ello, se usó el módulo *ImageGenerator* de *keras*. Para la aumentación de datos se aplicaron cambios en el brillo de las imágenes y en la orientación, en el zoom y giros; en concreto se usaron los siguientes parámetros: *brightness range*=[0.5,1.0], *shear range*=0.2, *zoom range*=0.1 y *horizontal flip*=*True*.

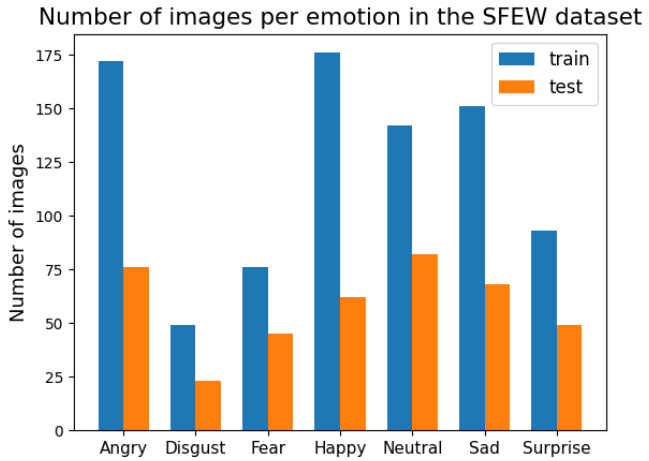


Figura 11: Número de imágenes por clase del SFEW

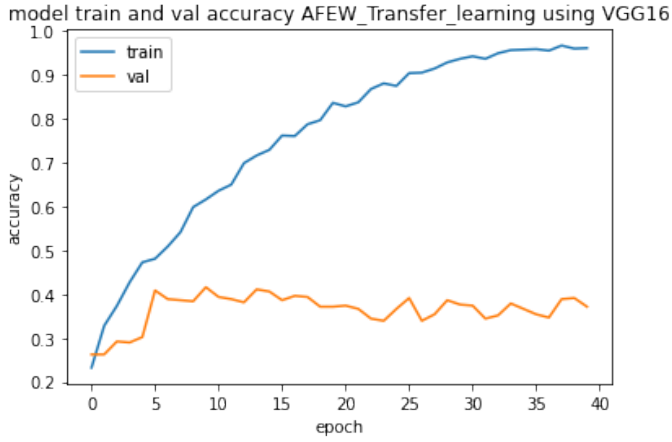


Figura 12: Resultados de utilizar *transfer learning* utilizando la VGG-16 preentrenada con *Imagenet*

Se repitieron las pruebas anteriores, esta vez utilizando *data augmentation*, pero no se mejoraron los resultados de manera significativa en ninguno de los dos casos.

Finalmente, dado que la arquitectura base utilizada era una arquitectura muy sencilla que tenía muy buenos resultados con el FACES pero no con el AFEW se decidió probar con otra arquitectura. Siguiendo la idea de trabajos similares como el de Bargal et al. [1] se decidió utilizar *transfer learning*.

Después de estudiar varias opciones, se eligió la red VGG-16 preentrenada con *Imagenet*. Se seleccionó esta red porque se ha usado en múltiples trabajos para tareas parecidas como el de Yi et al. [13] y porque, además, es uno de los modelos disponibles de keras.

Se congeló el núcleo convolucional de la red y se añadió un pequeño clasificador al final formado por una capa de *flatten*, para aplanar la salida del bloque convolucional, una capa densa de 1024 neuronas y activación *reLu* y una capa densa de salida formada por 7 neuronas y activación *softmax*. Se aplicó *fine-tuning* con los dos últimos bloques convolucionales de la red.

Para esta última prueba los mejores resultados se obtuvieron congelando los pesos de la red excepto los 2 últimos bloques convolucionales que se reentrenaron con el SFEW usando *data augmentation* con los mismos parámetros que en los casos anteriores. La curva de entrenamiento obtenida puede verse en la figura 12.

Se obtuvo una *accuracy* del 39 % lo que supone una mejora respecto a los casos anteriores pero sigue estando lejos de ser un buen resultado. Cabe destacar que este resultado no se aleja en exceso del *baseline* de la competición EMOTIW en su edición de 2018 [9].

III-B. Vídeo

Una vez implementados y estudiados los modelos con imágenes estáticas, se ha implementado también la clasificación de emociones a partir de vídeo. Para ello, se ha partido de los modelos anteriores, en concreto del modelo de la VGG16 que era el que daba mejores resultados. Se ha utilizado *late fusion*, es decir, las predicciones de un vídeo se han obtenido haciendo una media de las predicciones de las imágenes pertenecientes a ese vídeo.

Para obtener las predicciones de los vídeos del AFEW se ha ultilizado el dataset de imágenes estáticas SFEW con el que se ha trabajado en la sección anterior. Como se ha comentado anteriormente, las imágenes del SFEW han sido extraídas de los vídeos del AFEW así que en el identificador de cada una de ellas contienen la información del vídeo del AFEW al que pertenecen.

Cabe destacar que no todos los vídeos del AFEW tienen imágenes correspondientes del SFEW y que las etiquetas de las imágenes individuales no siempre se corresponden con la etiqueta del vídeo. Para seguir un único criterio se ha decidido utilizar como etiqueta válida la de los vídeos del AFEW. Además de esto, debe tenerse en cuenta que el número de imágenes por vídeo es bastante bajo. Es por eso, que también se testeó el modelo descrito a continuación con el subconjunto de frames de caras que forma parte del propio AFEW pero contiene muchas muestras mal seleccionadas que no se corresponden con imágenes faciales y que añaden mucho ruido al conjunto. Los resultados con el SFEW eran considerablemente mejores que los obtenidos con este segundo conjunto de datos así que finalmente se descartó utilizar este segundo conjunto de imágenes.

Para calcular las predicciones de cada uno de los vídeos, primero se han calculado los vectores de predicciones para cada una de las imágenes del sfew pertenecientes a ese vídeo utilizando el modelo de la VGG16 descrito en el apartado anterior. A continuación, se ha calculado la media de dichos vectores para obtener el vector de predicciones resultado del vídeo, de manera que cada valor del vector de resultados representa la media por emociones de las predicciones de cada uno de los frames del vídeo. Finalmente, la emoción predicha para el vídeo se ha obtenido como la emoción con mayor valor del vector de predicciones media. Este proceso se describe en la ecuación 1 donde $x \in R^k$, n es el número de imágenes del vídeo y k el número de emociones.

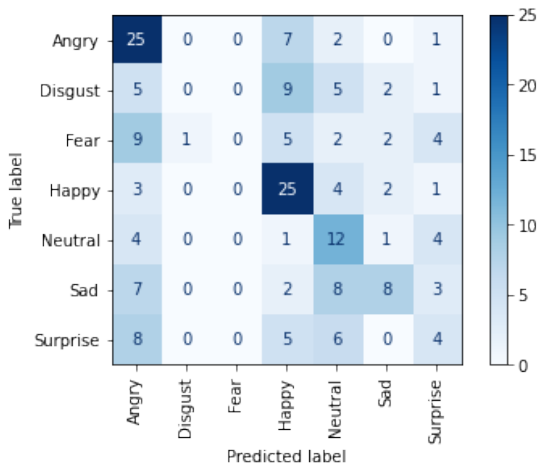


Figura 13: Resultados para la predicción de vídeo utilizando la VGG16 reentrenada con el SFEW

$$video\ prediction = \underset{k}{argmax}(\sum_{i=1}^{n} \frac{x_i}{n}) \quad (1)$$

Con este modelo,VGG16 reentrenada con *late fusion*, se obtiene una *accuracy* del 39.36 % y la matriz de confusión que puede verse en la figura 13

IV. Detección de emociones a partir del audio

Como se ha mencionado anteriormente, uno de los objetivos principales de este trabajo es estudiar la detección de emociones a partir de imágenes y audio y su combinación con tal de determinar qué método es mejor y si el audio aporta información relevante y puede contribuir a una mejora en los resultados. En este apartado se presentan distintos modelos para la clasificación de emociones a partir de audio que serán la base del modelo combinado presentado en la siguiente sección.

La clasificación de emociones a partir de audio es uno de los principales campos de estudio dentro del reconocimiento automático de emociones. La voz es la forma natural que tenemos los humanos de comunicar nuestro mensaje y codifica parte de la carga emocional de este.

El primer reto de todo algoritmo de reconocimiento de emociones por voz es determinar las mejores características que permitan distinguir entre las diferentes emociones. Dentro de las características más usadas en este tipo de métodos podemos distinguir dos grandes grupos. Por un lado, las características prosódicas que son aquellas que tienen que ver con las características del sonido que transmitimos(el volumen, el tono, las pausas...). Por otro lado, las características espectrales que son las relacionadas con las componentes frecuenciales del mensaje.

A pesar de lograr encontrar un buen set de características, la clasificación de emociones a partir de la información paralingüística del habla resulta compleja pues el audio solo codifica una parte del mensaje emocional que transmitimos. El resto, se transmite a través del mensaje y las expresiones faciales principalmente. Es por ello, que en muchos casos el audio es tratado como soporte añadido a la información visual más que como una señal de entrada a analizar de manera independiente.

De hecho, para el EMOTIW la mayoría de equipos se centran en la información visual, ya que es la que contiene la mayor parte de la información emocional [24]. En consecuencia, para el audio simplemente extraen vectores de características de distinto tipo (espectrales, funcionales, coeficientes delta...) con extractores como *OpenSmile* o *OpenEar* que luego pasan a una SVM [13], [11]. Este es el caso del baseline de la EMOTIW de 2013 que para el clasificador de audio utiliza una SVM lineal que aprende a partir de un vector de características obtenido con *OpenSmile* y obtiene un 19.5 % de *accuracy* [5].

Aunque este *baseline* ha sido ampliamente superado con los años [15], especialmente por aquellos equipos que han propuesto modelos de *deep learning* para el audio, nos da una idea de la dificultad de la tarea de construir un clasificador a partir de la señal de audio de los vídeos del AFEW.

Es precisamente por esta dificultad que la inmensa mayoría de trabajos se centran o bien en los modelos de imagen o bien utilizan el audio como complemento a un modelo de imagen.

Aun así, la mayoría de trabajos cuentan con un clasificador para la parte del audio, ya sea con SVM o con algún modelo de *deep learning*, y lo utilizan para combinarlo con el clasificador de imagen para obtener una predicción final. Esto se debe a que la contribución del audio suele ayudar a mejorar en cierto grado las predicciones del modelo únicamente de imágenes.

Dado que los modelos que utilizan *deep learning* son los que mejores resultados han obtenido con el audio en solitario se ha optado por implementar varios modelos convolucionales entrenados a partir de espectrogramas como se sugiere en el trabajo de Satt et al. [18] y comprobar cuál obtenía mejores resultados para el audio individualmente para luego utilizarlo como base en el modelo combinado.

IV-A. *Preprocesamiento de la señal*

Durante los últimos años la clasificación de señales de audio ha ido evolucionando y ha pasado de basarse en modelos que requerían un gran procesado y partían de vectores de características de audio típicas predefinidas a explorar las posibilidades que ofrece el *deep learning* para construir modelos de clasificación a partir de la señal de audio sin procesar [16].

En este trabajo se ha optado por un modelo intermedio y muy popular que consiste en preprocesar el audio para obtener *log-mel-spectrograms* y utilizar esta señal preprocesada para entrenar una red neuronal convolucional [14], [25].

Para la detección de emociones a partir de audio se han utilizado los audios extraídos de los vídeos de películas de los que está compuesta la base de datos AFEW. Estos audios tienen una duración de entre 1 y 6 segundos.

Primero, se ha extraído la señal de audio de los vídeos usando la librería *ffmpeg*. A continuación, se ha estandarizado la longitud de los audios a la longitud del audio más largo del dataset aplicando *zero-padding* al final del audio. Una

vez estandarizada la longitud, se ha aplicado un filtro de *pre-emphasis* a la señal para amplificar las altas frecuencias y balancear el espectro:

$$y(t) = x(t) - \alpha x(t-1) \quad (2)$$

Una vez aplicado el filtro de *pre-emphasis*, se ha dividido cada uno de los audios en *frames* y para cada *frame* se ha calculado el *Log-Mel-Spectrogram*. De esta forma, juntando las características obtenidas para cada *frame* se obtiene un vector de N *frames* donde cada frame se representa como 40 *filter-banks* en escala Mel. Se ha utilizado ventana de hamming, un tamaño de *frame* de 25 ms y un *stride* de 10 ms. Para la extracción de los *Log-Mel-spectrograms* se ha usado la libería *librosa*.

Finalmente, cada uno de los espectrogramas obtenidos se ha normalizado individualmente, con su media y su desviación estándar, y globalmente con la media y desviación estándar del set de entrenamiento.

IV-B. Modelo

Para la clasificación del audio se han utilizado redes convolucionales con dos configuraciones distintas. En la primera, se ha usado el espectrograma obtenido como resultado del proceso de preprocesado directamente como entrada de la red, mientras que en la segunda, se han convertido los espectrogramas obtenidos en imágenes y se ha entrenado la red con las imágenes de los espectrogramas. Esta segunda opción permite utilizar *transfer learning* con redes neuronales típicas que han sido entrenadas con imágenes como la VGG16.

Para cada una de las configuraciones se han entrenado dos arquitecturas distintas. Para la primera arquitectura se ha seleccionado un modelo sencillo que pudiera servir de base que había obtenido buenos resultados en la competición de kaggle de UrbandSound.[1] Este modelo se testeó como base con el propio dataset del Urbandsound (clasificación de sonidos ambientes de una ciudad) y se comprobó que se alcanzaba una *accuracy* de alrededor del 70 %.

Dados los buenos resultados se decidió probar qué desempeño podía tener esta arquitectura para la clasificación de emociones *in the wild*.

La arquitectura utilizada puede verse en la figura 14. Consta de 3 bloques convolucionales, una capa de *Flatten*, una capa densa de 512 neuronas y activación *relu*, una capa de *Dropout* y una capa de salida de 7 neuronas y activación *softmax*. Cada uno de los bloques convolucionales está formado por una capa convolucional de 32, 64 y 128 filtros de tamaño 3x3 respectivamente, *padding same* y activación *reLu*, una capa de *Maxpooling* de *pool size* 2x2 y una capa de *Dropout*.

Para la segunda configuración se ha implementado una red convolucional basada en el modelo propuesto por Fayek, Lech y Cavedon [14] utilizado para un problema de clasificación de emociones a partir de un dataset distinto.

El modelo está formada por 2 capas convolucionales con 16 y 32 filtros respectivamente de tamaño 10x10, seguidas de 2 capas densas de 716 con activación *ReLu*, una capa densa de 716 neuronas y activación *softmax* y una capa de salida con 7 neuronas y activación *softmax*. Después de cada una de las capas convolucionales se ha utilizado regularización l2=0.01 y *batchnormalization*. Después de las capas densas excepto la de salida se ha utilizado *batchnormalization* y *dropout*=0.5. El modelo puede verse en la figura 15.

Para la primera configuración se han utilizado los dos modelos anteriores entrenados a partir de los *Log-Mel-Spectrograms* obtenidos de procesar el audio siguiendo el proceso descrito en el apartado anterior. Se han probado diferentes combinaciones de hiperparámetros y diferentes grados de preprocesamiento del audio (con o sin filtro de pre-emphasis, normalizando los espectrogramas o sin normalizar...) pero ninguna combinación ha conseguido superar un *accuracy* de validación superior a valores de alrededor del 25 % para ninguno de los 2 modelos. Además, ambos modelos parecen tener un gran *overfiting* y la evolución del desempeño con el conjunto de validación no es bueno.

Para esta primera configuración, que consiste en entrenar los modelos a partir de los datos del audio procesados convertidos en Mel-Spectrogram, se ha probado también a entrenar una SVM con *kernel* lineal y RBF a partir de los espectrogramas pero los resultados han sido incluso peores que con los clasificadores anteriores puesto que las SVM clasificaban todas las muestras como la emoción mayoritaria.

Dados los malos resultados obtenidos para ambos modelos con la primera configuración, se ha decidido optar por otro enfoque popular a la hora de utilizar redes convolucionales para clasificación de audio que es la conversión de los espectrogramas en imágenes y tratar el problema como si fuera clasificación de imágenes (segunda configuración). Para ello se han entrenado las arquitecturas descritas anteriormente con los espectrogramas convertidos en imágenes pero los resultados no mejoraron demasiado. Un ejemplo de la incapacidad de la red para generalizar puede verse en la figura 16.

Finalmente, dados los malos resultados de los modelos anteriores y teniendo en cuenta que el hecho de utilizar el audio convertido en imagen como input nos permite utilizar cualquier red pre-entrenada con imágenes, se ha decidido testear la VGG16 que también habíamos probado para la clasificación de las imágenes para la clasificación de los espectrogramas. Con la VGG16 se obtienen resultados mejores que con los modelos anteriores, lo que una vez más confirma la potencia de los modelos pre-entrenados y el *transfer learning*. Aun así, estos resultados siguen sin ser demasiado satisfactorios.

Con esta configuración se logra una *accuracy* del 27 %, como puede observarse en la figura 18, un resultado ligeramente mejor que el *baseline* de la clasificación de emociones para el AFEW únicamente con audio para la EMOTIW2014 que era de un 26.78 % [4]. La matriz de confusión obtenida resultante puede verse en la figura 17.

De hecho, como se ha comentado anteriormente, en general puede observarse que muy pocos trabajos abordan la clasificación de emociones únicamente con audio para un dataset *in the wild* como el AFEW dada la dificultad de la tarea. La mayoría de los trabajos utilizan el audio como elemento complementario a las imágenes o el vídeo que suele ser la

[1] https://medium.com/gradientcrescent/urban-sound-classification-using-convolutional-neural-networks-with-keras-theory-and-486e92785df4

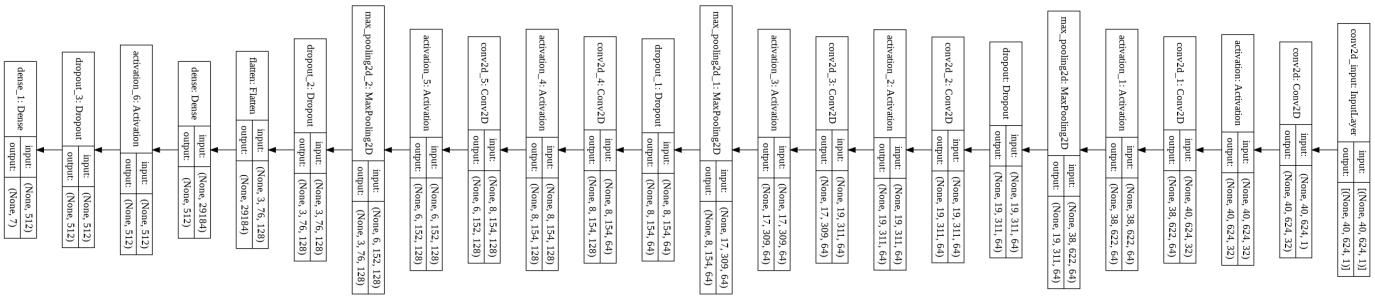


Figura 14: Modelo del audio 0 basado en el modelo del UrbandSound

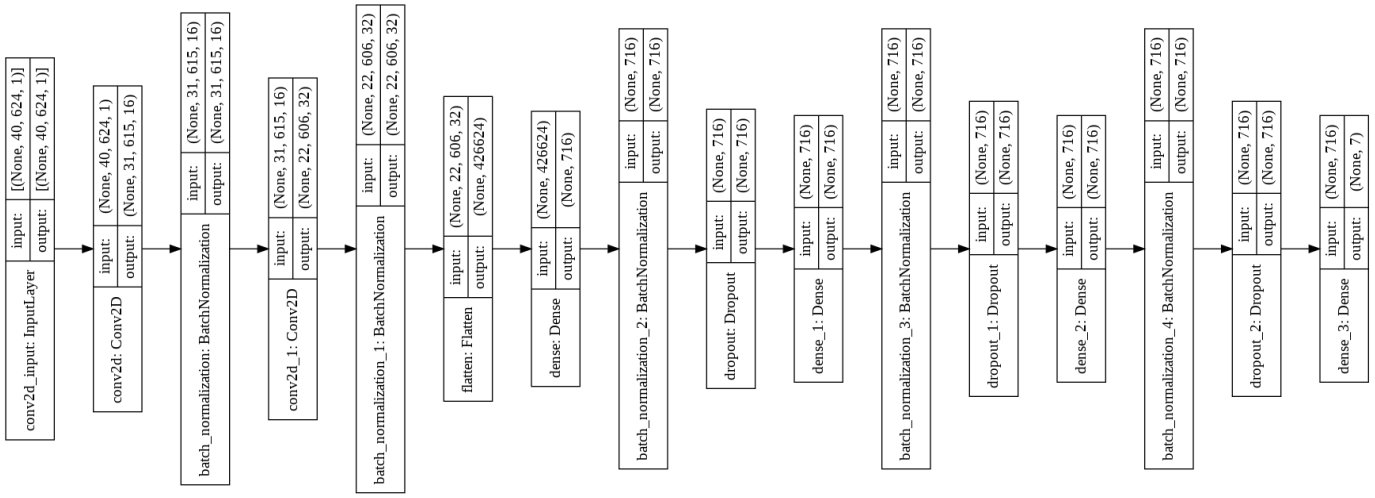


Figura 15: Modelo 1 del audio inspirado en el trabajo de Fayek, Lech y Cavedon [14]

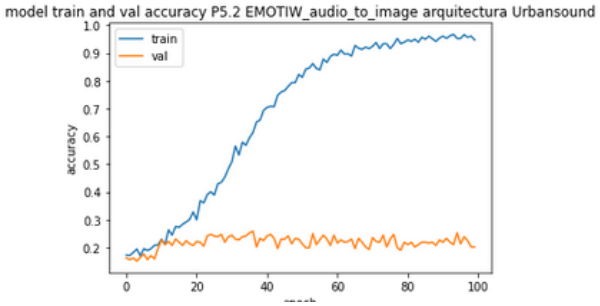


Figura 16: Resultados de utilizar la arquitectura 2 con espectrogramas convertidos en imágenes

Tabla I: Resultados modelos a partir del audio

Modelo	Configuración	Accuracy
Modelo 0 (UrbandSound)	1	22 %
Modelo 1 (Fayek)	1	25 %
SVM	1	15.4 %
Modelo 0 (UrbandSound)	2	21.2 %
Modelo 1 (Fayek)	2	21.7 %
VGG-16	2	27.6 %

fuente central de información.

En la tabla I puede verse un resumen de los resultados obtenidos con los distintos modelos para el audio.

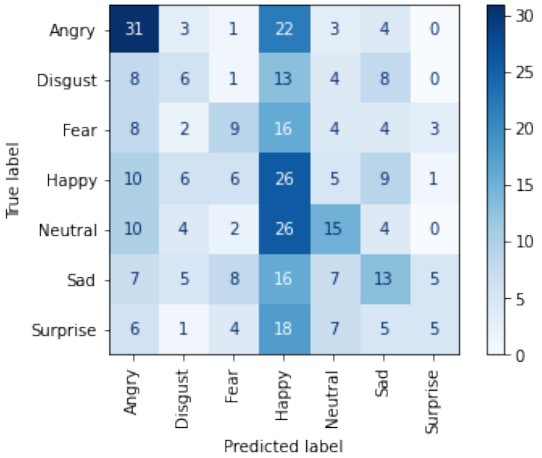


Figura 17: Matriz de confusión obtenida con la VGG16 entrenada con las imágenes de los espectrogramas aplicando fine-tuning a los dos últimos bloques convolucionales

V. Combinación de imágenes y audio

Una vez analizados los dos tipos de entradas por separado, en esta última fase del trabajo se han combinado las imágenes y el audio para comprobar si el audio aporta alguna mejora a la clasificación de emociones respecto al modelo con sólo imágenes.

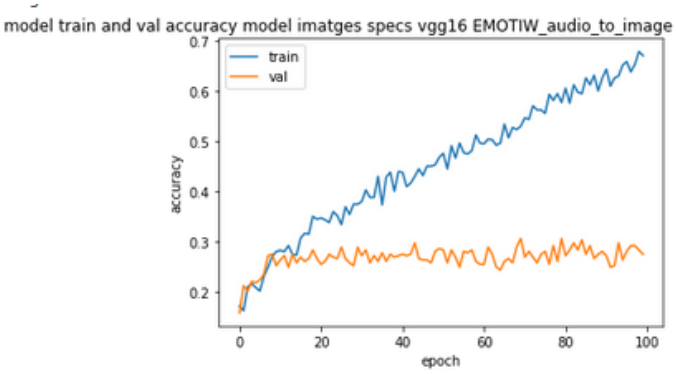


Figura 18: Resultados de utilizar la VGG16 con espectrogramas convertidos en imágenes

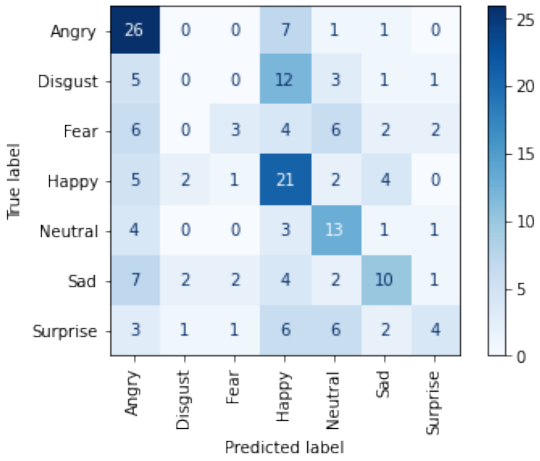


Figura 19: Resultados de promediar las predicciones del modelo del audio y el de vídeo

La combinación de modelos es una técnica extendida entre los trabajos que estudian el AFEW. La mayoría proponen modelos que terminan combinando la señal de audio y la de vídeo pues, normalmente la combinación suele contribuir a una mejoría de los resultados de los modelos unimodales.

Una propuesta frecuente, consiste en extraer vectores de características para cada uno de los modelos unimodales que se han desarrollado y, o bien hacer una media o una media ponderada de los distintos modelos [13], [23], [17], o bien concatenar los vectores de características y usarlos para entrenar algún tipo de clasificador, comúnmente una SVM o algún modelo secuencial [26]. De acuerdo con esto, en esta sección se presentan y comparan 3 modelos distintos de combinación basados en la combinación de los vectores de predicciones del audio y del vídeo.

Para combinar el audio y las imágenes es necesario emparejar las imágenes de un vídeo con el audio de dicho vídeo. Para ello, primero se han construido dos clasificadores sencillos, uno de audio y uno de vídeo, para obtener las predicciones para cada señal para cada uno de los vídeos.

Para el modelo del vídeo se ha utilizado el modelo descrito en la sección III-B que utiliza *late fusion* a partir de las imágenes del SFEW correspondientes a un mismo vídeo. Se ha utilizado este modelo entrenado con el conjunto de entrenamiento para obtener los vectores de predicciones del conjunto de entrenamiento y el conjunto de test.

Para las predicciones del audio se ha utilizado el modelo descrito en el apartado IV-B que utiliza la arquitectura VGG16 reentrenada con las imágenes de los *Log-Mel-espectrograms* obtenidos a partir de los audios de cada uno de los vídeos y se ha guardado también en un diccionario de predicciones.

Una vez obtenidos los vectores de predicciones del modelo del vídeo y del del audio se han probado tres modelos distintos para combinarlos.

Por un lado, se ha implementado un modelo combinatorio sencillo basado en hacer el promedio entre las predicciones del vídeo y las del audio. Éste es un enfoque muy sencillo y común que nos servirá para tener un punto de partida. El resultado de hacer el promedio de ambas predicciones resulta en una *accuracy* del 41 %, bastante mejor que la del audio individualmente y ligeramente superior a la del vídeo por sí solo. En la matriz de confusión, que puede verse en la figura 19, puede observarse que el modelo tiende claramente a las clases mayoritarias (enfadado, contento y neutral).

Por otro lado, una vez establecido un punto de referencia para el modelo combinado, se han testeado dos posibilidades para combinar las predicciones de los modelos de audio y vídeo. En la primera, para cada vídeo se han encadenado las predicciones obtenidas con el modelo del audio y el del vídeo para el conjunto de entrenamiento para formar vectores de características que se han pasado a una SVM. Para la segunda, estos vectores de características se han utilizado como input de un pequeño modelo secuencial.

Para la combinación utilizando SVM, se ha aplicado grid search para encontrar los mejores parámetros y los mejores resultados se han obtenido con una SVM con *kernel*='rbf', *gamma*=0.1, *C*=1. Para esta combinación de parámetros se ha obtenido una *accuracy* del 39.8 % muy similar a la del vídeo en solitario y una matriz de confusión con un reparto entre emociones algo mejor que la del vídeo como puede verse en la figura 22.

Para el modelo secuencial los mejores resultados se han obtenido con un modelo con una capa densa con 28 nodos y activación *reLu* y una capa densa con 7 neuronas y activación softmax. Se ha utilizado un 0.25 de *dropout* y *RMSprop* como optimizador. La evolución de la *accuracy* de entrenamiento y test puede verse en la figura 21.

Con este modelo se obtiene una *accuracy* del 45 %. Esto supone una mejora respecto al 39 % que obtenía el vídeo sin la contribución del audio. Además, si se observa la matriz de confusión de la figura 21 puede observarse que el modelo combinado parece contribuir a una mejor distribución entre las emociones y parece ayudar a que el modelo clasifique un poco mejor las emociones con las que el vídeo por sí solo tenía más dificultades como el asco.

En la tabla III puede verse un resumen de los resultados tanto para los modelos individuales como para las distintas configuraciones de combinación. Se puede apreciar una mejora de los resultados con la combinación de los modelos respecto al mejor modelo de base, que también se ve reflejada en las

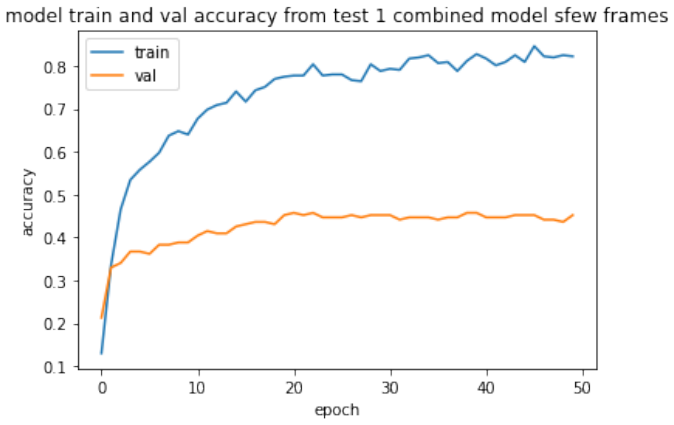


Figura 20: Train y test *accuracy* para el modelo secuencial de la combinación

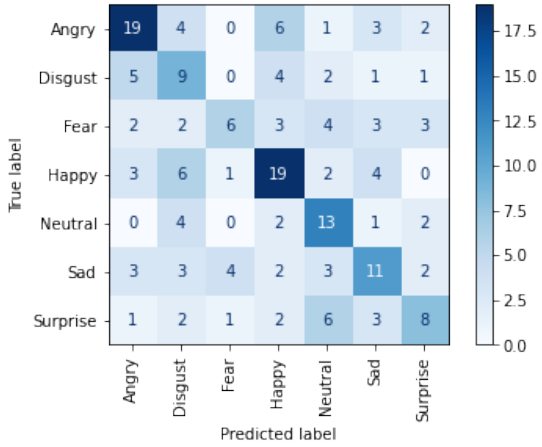


Figura 21: Resultados de combinar el audio y el vídeo con el modelo secuencial

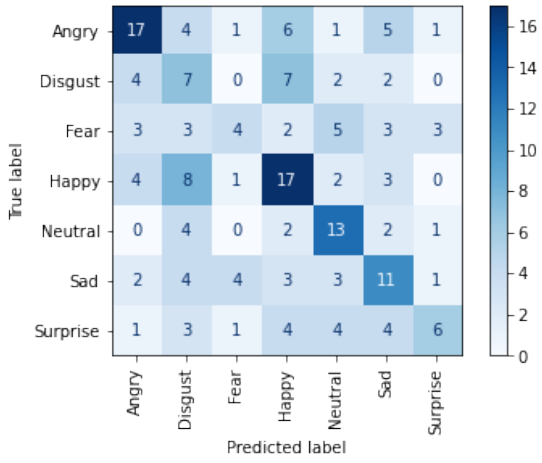


Figura 22: Resultados de combinar el audio y el vídeo utilizando SVM

Tabla II: Resumen de resultados de los modelos individuales y combinado

Modelo	Accuracy	f1-score
Audio	27.5 %	26 %
vídeo	39.4 %	32 %
Promedio	40.9 %	36 %
SVM	39.9 %	39.4 %
Modelo secuencial	45.2 %	44 %

Tabla III: Comparación de los resultados con el estado del arte

Modelo	Accuracy
Mejor modelo TFM	45.2 %
Baseline EMOTIW 2013	27.56 %
Baseline EMOTIW 2018	41 %
Ganadores EMOTIW 2018	60.34 %

matrices de confusión obtenidas. Además, puede observarse que la combinación de modelos en ambos casos parece contribuir a un mejor repartimiento en las predicciones de las clases en la matriz de confusión, contribuyendo a mejorar el desempeño del sistema a la hora de clasificar las emociones menos representadas en el dataset.

El modelo secuencial combinado ha demostrado mejorar los resultados de los modelos individuales significativamente. Finalmente, en la tabla III se muestra una comparativa de los resultados alcanzados en este trabajo respecto las dos ediciones de EMOTIW.

VI. Conclusión

En este trabajo se han estudiado e implementado diversos sistemas de reconocimiento de emociones *in the wild* a partir de imágenes y audio. Se han estudiado las dificultades de este tipo de datasets comparándolos con datasets en condiciones controladas y se han desarrollado una serie de modelos de *deep learning* para clasificar emociones a partir de audio y vídeo. Además, se han usado estos modelos unimodales como base para implementar un modelo combinado con el objetivo de estudiar si la combinación mejoraba los resultados.

A partir de este trabajo y los resultados obtenidos se ha podido constatar que, como ya avanzaba la bibliografía, el problema de la clasificación de emociones *in the wild* resulta complejo puesto que ni con las imágenes o el vídeo ni con el audio o el modelo combinado se han obtenido resultados suficientemente satisfactorios para un clasificador de emociones real. Además, se ha podido comprobar que modelos que obtenían buenos resultados para datasets sencillos o creados en condiciones de laboratorio disminuyen significativamente su rendimiento a la hora de aplicarlos a los datos *in the wild*.

De los modelos desarrollados se ha observado que de los dos canales, la imagen es el que contiene la mayor información emocional puesto que con los modelos de imagen se han conseguido unos valores de *accuracy* de alrededor del 39 % mientras que con el audio individualmente ninguno de los modelos ha logrado superar el 30 %. El audio parece contribuir en el modelo combinado a mejorar la capacidad del sistema de reconocer correctamente las emociones.

Por otro lado, de la comparativa entre modelos tanto con el vídeo como con el audio, se ha demostrado el gran potencial del *transfer learning* puesto que en ambos casos los mejores resultados se han obtenido re-entrenando la VGG-16 a partir de los pesos de *Imagenet*. Además la VGG-16 ha demostrado ser una red muy versátil y con grandes posibilidades puesto que con esta red se han logrado resultados similares a los primeros *baselines* de la competición sin aplicar ningún tipo de preprocesado a las imágenes.

Finalmente, aunque los resultados no quedan muy lejos de los *baselines* de la competición, presentan un amplio margen de mejora y no pueden considerarse satisfactorios ni suficientes para un clasificador destinado a una aplicación real. Estos resultados se deben a diversas causas. El dataset resulta muy pequeño (menos de 1000 imágenes de entrenamiento) y además está bastante desbalanceado lo que complica bastante aprender correctamente las emociones minoritarias. Además, al tratarse de un dataset *in the wild* las imágenes presentan diferencias de orientación de la cara, de iluminación, de zoom etc. De hecho, algunas imágenes del vídeo no corresponden a caras.

Todos estos factores explican que los modelos no hayan obtenido mejores métricas y ponen en valor los resultados obtenidos que pueden servir de *baseline* para trabajo futuro que desee explorar el AFEW más a fondo aplicando modelos más complejos En trabajos futuros, se podría investigar si preprocesando las imágenes aplicando detectores de caras o entrenando las redes a partir de vectores de características faciales se pueden mejorar estos resultados.

Referencias

[1] S. A. Bargal, E. Barsoum, C. C. Ferrer, and C. Zhang. Emotion recognition in the wild from videos using images. *ICMI 2016 - Proceedings of the 18th ACM International Conference on Multimodal Interaction*, pages 433–436, 2016.

[2] F. Chollet. *Deep learning with Python*. Simon and Schuster, 2017.

[3] N. Comparini. Análisis de expresiones faciales mediante técnicas de Deep Learning. *dspace.uib.es*, 2020.

[4] A. Dhall, R. Goecke, J. Joshi, K. Sikka, and T. Gedeon. Emotion recognition in the wild challenge 2014: Baseline, data and protocol. In *Proceedings of the 16th international conference on multimodal interaction*, pages 461–466, 2014.

[5] A. Dhall, R. Goecke, J. Joshi, M. Wagner, and T. Gedeon. Emotion recognition in the wild challenge 2013. In *Proceedings of the 15th ACM on International conference on multimodal interaction*, pages 509–516, 2013.

[6] A. Dhall, R. Goecke, S. Lucey, and T. Gedeon. Acted facial expressions in the wild database. *Australian National University, Canberra, Australia, Technical Report TR-CS-11*, 2:1, 2011.

[7] A. Dhall, R. Goecke, S. Lucey, and T. Gedeon. Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 2106–2112. IEEE, 2011.

[8] A. Dhall, A. Kaur, R. Goecke, and T. Gedeon. Emotiw 2018: Audio-video, student engagement and group-level affect prediction. In *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, pages 653–656, 2018.

[9] A. Dhall, O. Ramana Murthy, R. Goecke, J. Joshi, and T. Gedeon. Video and image based emotion recognition challenges in the wild: Emotiw 2015. In *Proceedings of the 2015 ACM on international conference on multimodal interaction*, pages 423–426, 2015.

[10] N. C. Ebner, M. Riediger, and U. Lindenberger. Faces—a database of facial expressions in young, middle-aged, and older women and men: Development and validation. *Behavior research methods*, 42(1):351–362, 2010.

[11] S. Ebrahimi Kahou, V. Michalski, K. Konda, R. Memisevic, and C. Pal. Recurrent neural networks for emotion recognition in video. In *Proceedings of the 2015 ACM on international conference on multimodal interaction*, pages 467–474, 2015.

[12] Y. Fan, X. Lu, D. Li, and Y. Liu. Video-based emotion recognition using cnn-rnn and c3d hybrid networks. In *Proceedings of the 18th ACM international conference on multimodal interaction*, pages 445–450, 2016.

[13] Y. Fan, X. Lu, D. Li, and Y. Liu. Video-Based emotion recognition using CNN-RNN and C3D hybrid networks. *ICMI 2016 - Proceedings of the 18th ACM International Conference on Multimodal Interaction*, (October 2017):445–450, 2016.

[14] H. M. Fayek, M. Lech, and L. Cavedon. Evaluating deep learning architectures for speech emotion recognition. *Neural Networks*, 92:60–68, 2017.

[15] S. E. Kahou, X. Bouthillier, P. Lamblin, C. Gulcehre, V. Michalski, K. Konda, S. Jean, P. Froumenty, Y. Dauphin, N. Boulanger-Lewandowski, et al. Emonets: Multimodal deep learning approaches for emotion recognition in video. *Journal on Multimodal User Interfaces*, 10(2):99–111, 2016.

[16] J. Lee, T. Kim, J. Park, and J. Nam. Raw waveform-based audio classification using sample-level cnn architectures. *arXiv preprint arXiv:1712.00866*, 2017.

[17] C. Liu, T. Tang, K. Lv, and M. Wang. Multi-feature based emotion recognition for video clips. In *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, pages 630–634, 2018.

[18] A. Satt, S. Rozenberg, and R. Hoory. Efficient emotion recognition from speech using deep learning on spectrograms. In *Interspeech*, pages 1089–1093, 2017.

[19] A. F. Shima Alizadeh. Convolutional neural networks for facial expression recognition. *Cognition and Emotion*, 4.3.

[20] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[21] K. Slimani, M. Kas, Y. El Merabet, R. Messoussi, and Y. Ruichek. Facial emotion recognition: A comparative analysis using 22 lbp variants. In *Proceedings of the 2nd Mediterranean Conference on Pattern Recognition and Artificial Intelligence*, pages 88–94, 2018.

[22] L. Torrey and J. Shavlik. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI global, 2010.

[23] V. Vielzeuf, C. Kervadec, S. Pateux, A. Lechervy, and F. Jurie. An occam's razor view on learning audiovisual emotion recognition with small training sets. In *Proceedings of the 20th ACM International Conference on Multimodal Interaction*, pages 589–593, 2018.

[24] A. Yao, J. Shao, N. Ma, and Y. Chen. Capturing au-aware facial features and their latent relations for emotion recognition in the wild. In *Proceedings of the 2015 acm on international conference on multimodal interaction*, pages 451–458, 2015.

[25] H. Zhou, D. Meng, Y. Zhang, X. Peng, J. Du, K. Wang, and Y. Qiao. Exploring emotion features and fusion strategies for audio-video emotion recognition. In *2019 International Conference on Multimodal Interaction*, pages 562–566, 2019.

[26] H. Zhou, D. Meng, Y. Zhang, X. Peng, J. Du, K. Wang, and Y. Qiao. Exploring emotion features and fusion strategies for audio-video emotion recognition. In *2019 International Conference on Multimodal Interaction*, pages 562–566, 2019.