Sistema De Reconocimiento De Emociones Faciales

Adrià Martínez Moreno

Resumen. Actualmente existe la necesidad de ser capaces de clasificar las expresiones faciales de los humanos de manera automática para realizar algún tipo de análisis. El proyecto está dirigido al estudio e implementación de un sistema de reconocimiento de emociones faciales, utilizando redes neuronales (Deep Learning).
A lo largo de este trabajo se hace uso de seis datasets diferentes. El resultado demuestra que el modelo creado a partir de la unión de los seis datasets obtiene mejores resultados que cualquiera de ellos por solitario.

Palabras Clave. deep learning, red neuronal convolucional, aprendizaje automático, reconocimiento de emociones faciales, tensorflow, tflearn

Abstract Currently there is a need to be able to automatically classify the facial expressions of humans to perform some type of analysis. The project is aimed at the study and implementation of a program that performs the recognition of a system of recognition of facial emotions using neural networks (Deep Learning).
Throughout this work we use six different datasets. The result shows that the model created with the union of the six datasets in one only, obtains better results that any of them by solitary.

Index Terms. Deep learning, convolutional neural network, machine learning, recognition of facial emotions, tensorflow, emotions

1 Introducción

Los seres humanos estamos acostumbrados a reconocernos entre nosotros usando los rasgos faciales. Las emociones faciales han sido usadas en multitud de campos, por ejemplo, en la Psicología [1] o la Antropometría [2]. Esto ha dado paso a multitud de estudios, y la oportunidad de crear un sistema automático de reconocimiento de emociones faciales.

Las caras constituyen uno de los estímulos más importantes de la interacción social, ya que aportan información sobre la identidad de la persona y además de su estado emocional. Las emociones faciales son uno de los sistemas de señales que más importancia tienen en el momento de expresar a otras personas lo que nos sucede [3].

La automatización de esta tarea es compleja, debido a las variaciones entre las distintes imágenes de un mismo individuo, ya sea la posición de la cara, la iluminación, el ángulo, la distancia, el color, el maquillaje, el peinado, la presencia de sombras, gafas, etc.

El proyecto está dirigido al estudio e implementación de un programa que efectúe el reconocimiento de emociones faciales utilizando redes neuronales (Deep Learning) a partir de Python, utilizando OpenCV y TFLearn.

El sistema tiene que ser capaz de detectar siete emociones faciales: 1) Normal, 2) Feliz, 3) Miedo, 4) Enfado, 5) Triste, 6) Disgusto y 7) Sorpresa.

Acto seguido, se introducirá un emoticono con su correspondiente estado encima de la imagen analizada.

El Trabajo está estructurado en las siguientes secciones:

La **sección 2** presenta el contexto del proyecto a través del estado del arte. La **sección 3** muestra los objetivos generales y específicos del proyecto, y expone los requisitos funcionales y no funcionales. La **sección 4** muestra el caso de uso. La **sección 5** detalla conceptos a entender del proyecto. La **sección 6** presenta la metodología que se ha seguido. La **sección 7** enseña los experimentos y resultados obtenidos. La **sección 8** detalla las conclusiones y trabajos futuros.

- *E-mail de contacte: adriamartinezmo@e-campus.uab.cat*
- *Mención realitzada: Computació*
- *Treball tutoritzat per: Katerine Diaz (CVC)*
- *Curs 2016/17*

2 ESTADO DEL ARTE

El reconocimiento de las emociones faciales es utilizado en una gran variedad de campos diferentes.

Microsoft lanzó en 2016 Emotion Api [4], aplicación que considera una expresión facial de una imagen como una entrada y devuelve la emoción detectada para cada cara de la imagen. Capaz de clasificar entre 8 emociones diferentes, las mismas que clasifica nuestro sistema más la emoción de desprecio. La Api es totalmente gratuita pero aún sigue en versión preliminar.

La fundación Ave María presentó el proyecto AutonomMe, que reconoce el estado emocional de las personas con altos grados de dependencia y obtiene la validación clínica de los resultados.

Eyeris es la empresa líder del software de reconocimiento de emociones faciales basado en Deep Learning. Su proyecto estrella, EmoVu [5], ofrece la suite más completa de análisis de cara y se utiliza en multitud de aplicaciones comerciales hoy en día, como de automóviles, robótica y análisis de vídeo.

Otra de las aplicaciones más famosas tiene el nombre de Affectiva [6], desarrollada en el Massachussets Institute of Techonology (MIT). El software capta imágenes a través de una webcam y las clasifica en 7 emociones diferentes (las mismas que nuestro sistema menos el estado normal, en su caso clasifican la emoción desprecio), además de proveer de 20 métricas faciales. Sus algoritmos han sido entrenados con más de 5 millones de rostros de 75 paises diferentes. Es utilizada, al parecer, por marcas como Coca-cola, Unilever, Mars, Kellogg y CBS para evaluar sus anuncios y el impacto producido a los espectadores [7].

Affectiva nos ofrece un accuracy del 90%, y ha sido testeado en más de 3.2 millones de vídeos, representando el mundo real, con cambios en las condiciones como puede ser la luminosidad, diferentes movimientos de cabeza y en características del individuo como la raza, edad, género, color de pelo y gafas.

Tanto Affectiva como Emovu, son capaces de determinar la edad de los usuarios dentro de un rango, además de clasificar a partir del sexo.

3 OBJETIVOS

3.1 Objetivos Generales

El objetivo general del proyecto es desarrollar un sistema que detecte siete emociones faciales, y que este funcione en tiempo real a partir de modelos entrenados con redes neuronales y seis datasets diferentes, también se compararán resultados para comprobar cual clasifica mejor.

La información generada debe mostrarse en pantalla. Se utilizarán emoticonos para que el usuario pueda identificar que emoción ha detectado la aplicación.

El resultado se guardará en archivos de texto para así controlar su correcto funcionamiento. Además, se guardará una copia de la imagen con el resultado obtenido, así como el emoticono que la representa.

3.2 Objetivos Específicos

Los objetivos específicos del proyecto son:

- Validar el sistema generado.
- Proponer una arquitectura que clasifique 7 emociones faciales.
- Realizar Fine-Tuning a un modelo ya entrenado.
- Testear el modelo con mejores resultados con todos los demás datasets.
- Implementar un sistema para detectar el rostro de una persona tanto en imágenes como en vídeos a tiempo real.
- A partir del rostro de la persona, reconocer la emoción a la que corresponde.
- Diseñar un sistema capaz de introducir un emoticono en la imagen original correspondiente al estado de ánimo detectado.

3.3 Requisitos Funcionales

- El sistema ha de poder trabajar con imágenes en tiempo real.
- El sistema ha de tener un fácil uso y accesibilidad para usuarios no experimentados.
- El sistema guardará la información generada y el tipo de emoción detectada en un archivo.
- El sistema mostrará los resultados por pantalla.

Por otra parte, no se han especificado requisitos no funcionales para el proyecto.

4 CASOS DE USO

El usuario que utilice nuestra aplicación tendrá la capacidad de elegir si quiere un uso manual o a tiempo real (requiere de una webcam).

Para el uso manual, nuestro sistema trabajará con las imágenes del usuario que haya dispuesto en el directorio. El resultado se mostrará por pantalla y se guardará en formato jpg junto a un archivo de texto con el resultado obtenido, tal como presenta la figura 1.

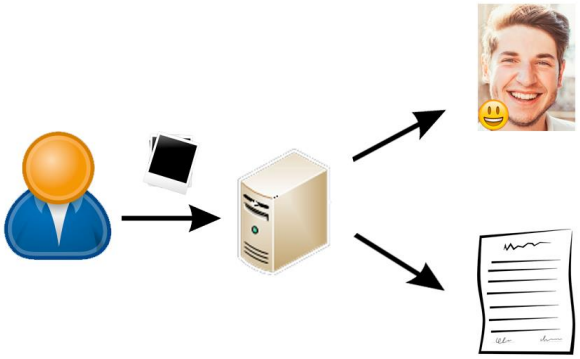

Figura 1: Caso de Uso Manual.

Para el uso a tiempo real, nuestro sistema trabajará con la imagen obtenida por webcam u otras aplicaciones de vídeo. El usuario podrá verse en la pantalla junto a los resultados, que irán cambiando según la expresión que el usuario ponga en ese momento, tal como presenta la Figura 2.

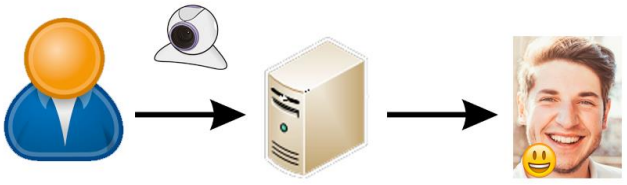

Figura 2: Caso de Uso a Tiempo Real.

En la tabla 1 podemos observar cuál es el emoticono correspondiente a cada emoción:

EMOCIÓN	EMOTICONO
Normal	😐
Feliz	😀
Miedo	😨
Enfado	😠
Triste	😢
Disgusto	😖
Sorpresa	😯

Tabla 1. Emociones y su emoticono correspondiente.

5 Conceptos

A continuación se hará un breve resumen sobre los diferentes conocimientos necesarios para entender completamente el funcionamiento de la aplicación.

5.1 Términos

- Epoch: Consiste en un ciclo de entrenamiento completo. Una vez que se haya visto cada muestra del conjunto, comienza de nuevo, marcando el inicio del siguiente epoch.
- Batch Size: Define el número de muestras que van a propagarse a través de la red.
- Accuracy: Precisión obtenida.
- Loss: Se calcula en el entrenamiento y en la validación y es una interpretación de cómo de correcto es el modelo para esos 2 sets.
- Overfitting: Efecto de sobreentrenar el algoritmo de aprendizaje, quedando ajustado a unas características muy específicas de los datos de entrenamiento.

5.2 Red Neuronal Convolucional

Es un sistema de aprendizaje automático inspirado en el sistema nervioso biológico. Este, es creado a partir de la conexión de diferentes elementos independientes llamados neuronas.

Sigue una estructura por capas (input, hidden y output layers) [8], con diferente número de neuronas en cada capa. Las conexiones entre neuronas están ponderadas por unos pesos (weights) que se van ajustando en la etapa de entrenamiento con el objetivo de minimizar el coste. En la Figura 3 se puede observar como los pesos (w) se multiplican por la salida de la neurona anterior (x) y luego se aplica un sumatorio de todas las entradas de la neurona para calcular una función de activación.

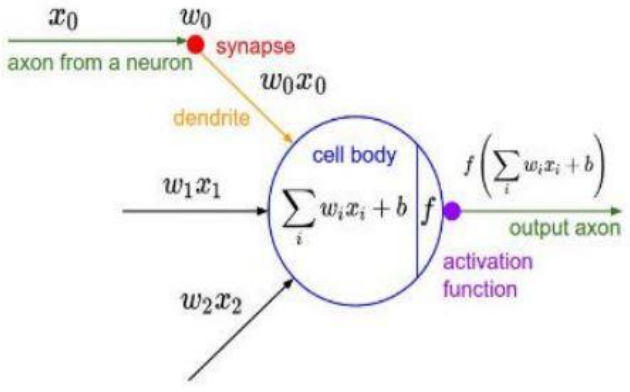

Figura 3: Neurona

La red neuronal convolucional está optimizada para trabajar con entradas de más de una dimensión, como es el caso de las imágenes. A diferencia de una red neuronal normal, los pesos son matrices n-dimensionales, también conocidas como filtros.

Su objetivo principal es extraer información de alto nivel de las imágenes para clasificarlas en unos determinados estados, en nuestro caso siete.

Para realizarlo, la red se encarga de procesar la imagen de entrada de manera que la salida de la última capa dé un resultado. Cada capa procesa la salida de la anterior, exceptuando la primera capa que directamente procesa la imagen de entrada.

En las redes convolucionales existen multitud de capas, pero solamente se explicarán las capas que han sido utilizadas para este proyecto.

- **Capa Convolucional:** Es la capa básica de las redes convolucionales. Los parámetros de esta capa son un conjunto de filtros (matrices de pesos) que hacen una transformación sobre la información que pasamos de entrada. Es decir, si tenemos N filtros de dimensiones MxMxZ y unos datos de entrada de dimensiones VxVxR tendremos (M-V+1) x (M-V+1) x N neuronas totales en nuestra red. La capa convolucional aumenta la profundidad de la salida. La función de actvación utilizada es la Rectified Linear Unit (ReLU) que permite normalizar la entrada calculando el máximo entre el valor de cada píxel de la entrada y 0. De esta manera, se eliminan los valores negativos que se hayan podido crear en anteriores operaciones.

- **Max Pooling:** Es la capa encargada de reducir en parámetros la información de su entrada. El procedimiento que sigue consiste en elegir el mayor valor de píxel como salida, eliminando el resto, ya que no es necesario. Esta acción permite reducir el tamaño de la imagen dependiendo del tamaño de la ventana que hayamos configurado previamente en la red neuronal.

- **Fully-Connected:** Se coloca en las últimas capas de la red neuronal y es la encargada de realizar el paso final, la clasificación. Es decir, devuelve el resultado obtenido durante el entrenamiento. Utiliza el dropout [9] para determinar la probabilidad de conexión entre neuronas. Así evitaremos el overfitting de la red, además de reducir el ajuste insuficiente. Su salida es el vector de características que procederá a ser conectado al clasificador, en nuestro caso un softmax.

5.3 Entrenamiento

El entrenamiento es realizado con el proceso backpropagation [10]. Para cada imagen de entrada se calcula el error producido durante la fase de clasificación y se cambian automáticamente los parámetros de los filtros de la red para tratar de minimizar este error y obtener un mejor resultado.

Para conseguir mejor resultado, se utilizan algoritmos de optimización para el cálculo de los pesos de los filtros.

En este caso se ha utilizado Momentum, ya que obtenemos estas ventajas:

- Evitar sobreajuste, ya que si alcanzamos un mínimo local este lo bordearía gracias al acumulador.

- Aumento de la velocidad, ya que no puede variar bruscamente la trayectoria y puede filtrar pequeñas fluctuaciones.

- Aumenta la convergencia al suavizar la trayectoria.

5.4 Clasificador

Después de haber extraído el vector de características de la capa Fully Connected se ha clasificado la entrada a una única clase. Para ello se ha utilizada la función softmax [11], con la cual pasamos de un vector de características a un vector de probabilidades asignadas entre 0 y 1 (dando como resultado 1 la suma de estas en total). Con ello podremos determinar su correcto funcionamiento.

5.5 Finetune

Existen muchos programadores e investigadores que han desarrollado redes neuronales para realizar el reconocimiento de personas, vehículos, colores, entre otros.

Se ha buscado la posibilidad de adaptar alguna de estas redes ya entrenadas a nuestro proyecto. Para adaptar la red ya entrenada a otro problema, en casos generales, se reentrena la red cambiando las últimas capas y utilizando un dataset definido para la aplicación. Este proceso es conocido como finetune o como transfer learning.

6 METODOLOGÍA

Esta sección da una visión global de las fases de desarrollo del proyecto. En la figura 4 podemos observar el diagrama de la metodología.

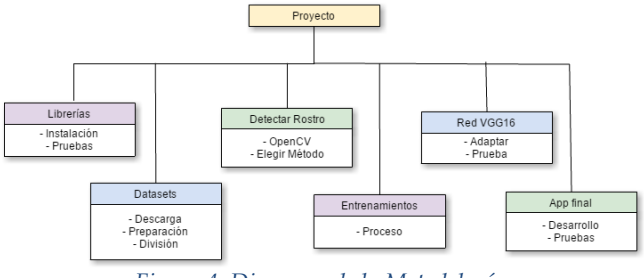

Figura 4: Diagrama de la Metodología.

6.1 Planificación

El cronograma de la planificación del proyecto es el mostrado en la tabla 2.

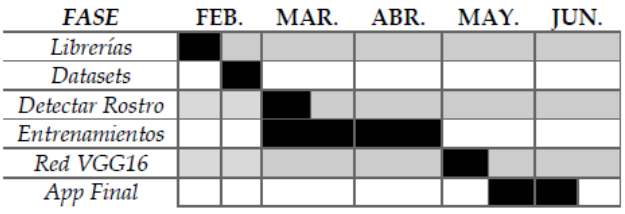

Tabla 2. Cronograma del proyecto.

6.2 Librerías

Entre las librerías que existen (Theano [12], Caffe [13]…) se escogió Tensorflow, ya que nos permite trabajar con Python, posibilidad de visualizar de los modelos, además de facilitar el trabajo con imágenes, en la tabla 3 podemos observar las principales diferencias entre las librerías más famosas:

	TensorFlow	Theano	Caffe
Python	SI	SI	SI
Uso GPU	SI	SI	SI
Extracción De Grafos	SI	SI	NO
Pre-Trained Models	Pocos	Pocos	Muchos
Rapidez	Lenta	Lenta	Rápida
Documentación	Mucha	Media	Media
Open Source	SI	SI	SI

Tabla 3. Comparación librerías.

La opción más lógica hubiese sido escoger Caffe, pero en primer momento se optó por Tensorflow, puesto que nos permitía usar la herramienta TFLearn, un framework diseñado a partir de Tensorflow que aporta las siguientes ventajas:

- Facil de usar y comprender: API de alto nivel para la implementación de redes neuronales profundas, con mucha cantidad de tutoriales y ejemplos.
- Prototipado rápido a través de capas de redes neuronales integradas y altamente modulares.
- Transparencia total sobre TensorFlow. Se puede usar independientemente de TFLearn.
- Sencilla visualización de gráficos, con detalles sobre los pesos, gradientes, activaciones…
- Uso de múltiples CPU/GPU.

6.3 Datasets

Se han utilizado 6 Datasets diferentes para entrenar los modelos, en la tabla 4 se resume el número de imágenes que contienen y el número de estados que engloban.

DATASET	NUM. IMG	ESTADOS
10K	10.168	3
CK	350	5
JAFFE	213	7
MUG	91.210	7
RAFD	7.035	7
YALE	45	3

Tabla 4. Cantidad imágenes y estados en los datasets.

- **10K:** Este dataset cuenta con 10.168 imágenes de personas entre 30 y 45 años, el 57.1% son hombres y el 42.9% son mujeres. Solamente tiene las emociones normal, feliz y enfado. [14].
- **CK:** Este dataset cuenta con 350 imágenes de 97 sujetos diferentes. Tiene las emociones normal, feliz, miedo, enfado y triste [15].
- **JAFFE:** Está formado por 213 imágenes de 7 modelos japonesas. Tiene todas las emociones [16].
- **MUG:** El dataset cuenta con 91.210 imágenes de personas de entre 20 y 35 años, 35 mujeres y 51 hombres. Tiene todas las emociones [17].
- **RAFD:** En total son 7.035 imágenes, aunque la mayoría son de perfil. Tiene todas las emociones. [18]
- **YALE:** Este dataset está formado por 15 personas diferentes. Solo tiene las emociones normal, feliz y triste [19].

A continuación, se ha utilizado un script de Matlab para detectar rostros automáticamente. Debido a que daba bastantes problemas, sobretodo con datasets como rafd por las imágenes en perfil, solamente se utilizó en los datasets con menos imágenes: 10K, CK, Jaffe y Yale.

Este script nos ha permitido obtener el rostro de la imagen para así evitar información no importante para el entreno, como puede ser el fondo.

Para el dataset Mug se ha utilizado un script de Python en OpenCV de detección de rostro (el mismo utilizado en la aplicación final). Además se ha automatizado la tarea, quedando en total 74.324 imágenes al obviar casi 20.000 imágenes en las que no se detectaba el rostro correctamente, asegurándonos así de que nuestra red hará un entreno correcto.

Al utilizar Rafd el resultado no ha sido bueno por lo comentado anteriormente, muchas de sus imágenes son en perfil, así que se ha optado por trabajar con el dataset sin modificar directamente.

Los datasets también se han equilibrado para tener el

mismo número de imágenes para cada estado, y así evitar que el modelo tenga preferencia por una determinada clase. Para la clasificación se han rotado imágenes y se han hecho transformaciones de tipo espejo hasta que ha quedado equilibrado, como podemos observar en la figura 5.

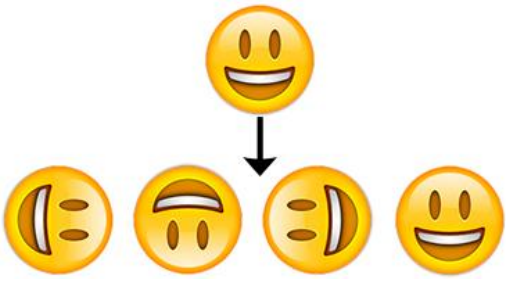

Figura 5: Generación de imágenes.

También se ha creado un nuevo dataset, formado por todos los demás, lo que nos permite poder realizar más pruebas y obtener mejores conclusiones. Los datos finales con los que contamos tras estos procedimientos están mostrados en la tabla 5:

Emoción	10K	CK	Jaffe	Mug	Rafd	Yale	All
1. Normal	1.752	99	32	12.761	1.005	15	15.664
2. Feliz	1.752	99	32	12.761	1.005	15	15.664
3. Miedo	0	99	32	12.761	1.005	0	15.664
4. Enfado	1.752	99	32	12.761	1.005	0	15.664
5. Triste	0	99	32	12.761	1.005	15	15.664
6. Disg.	0	0	32	12.761	1.005	0	15.664
7. Sorpr.	0	0	32	12.761	1.005	0	15.664

Tabla 5. Número de imágenes en los datasets.

El nuevo dataset consta de 15.664 imágenes de cada estado, teniendo 109.508 en total, obtenidos a partir del conjunto de los demás datasets. Cuando algún dataset no tenía imágenes de algún estado se ha optado por realizar el procedimiento anterior, haciendo transformaciones de los otros datasets hasta llegar al mismo valor.

Todas las imágenes se han convertido a blanco y negro para así trabajar solamente con un canal de color, y también se han redimensionado al tamaño 48x48 para evitar problemas de memoria de la GPU en el momento de hacer los entrenamientos de los modelos.

Para realizar el proceso de training, validation y test, se ha optado por dividir el dataset en un 50% para la fase de entrenamiento, un 30% para la fase de validación, y un 20% para testear el modelo entrenado, quedando los datasets divididos tal y como se muestra en las tabla 6:

Partes	10K	CK	Jaffe	Mug	Rafd	Yale	All
Training	2.628	248	112	44.663	3.517	23	51.191
Validation	1.576	149	67	26.798	2.110	13	30.713
Test	1.052	98	44	17.865	1.407	9	20.475

Tabla 6. Número de imágenes en cada división del dataset.

6.4 Detección de Rostro

Para esta opción se ha utilizado OpenCV, pues ofrece herramientas muy útiles y sencillas para lograr el cometido. Antes de empezar, se va a explicar el funcionamiento del reconocimiento facial de OpenCV.

OpenCV utiliza un conjunto de "bloques" mostrados en la figura 6 para reconocer formas llamadas Haar-like features:

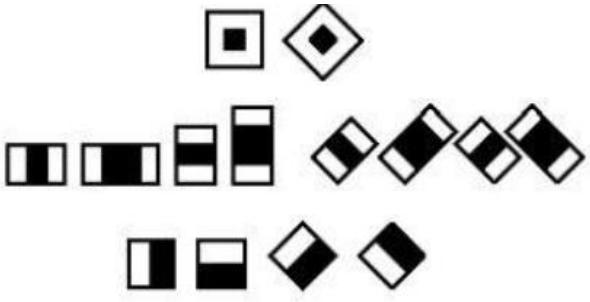

Figura 6: Edge, Line y Four-Rectangle Features.

El algoritmo busca en la imagen combinaciones de estos patrones. Por ejemplo, si queremos detectar un rostro, como es el caso del proyecto, el algoritmo buscará en la imagen la combinación de estos bloques que, si se juntan, se aproximan a un rostro. La figura 7 muestra el proceso de este algoritmo:

Figura 7: Detección de rostro

Este sistema tiene un porcentaje de aciertos bastante alto, y también es el que se ha utilizado para detectar el rostro de las imágenes del dataset Mug. Su éxito dependerá del tipo de cámara utilizada, la iluminación del lugar, distancia, etc.

6.5 Extracción de características

La arquitectura de la red diseñada para el desarrollo de la aplicación es la mostrada en la figura 8.

Consta de tres capas convolucionales de 2 dimensiones.

La primera capa convolucional utiliza un filtro de profundidad de 1, ya que la entrada de la red es una imagen de tamaño 48x48x1.

En las otras capas de convolución, esta profundidad viene establecida por la profundidad de la capa anterior. Por consiguiente, si la salida de la segunda capa de

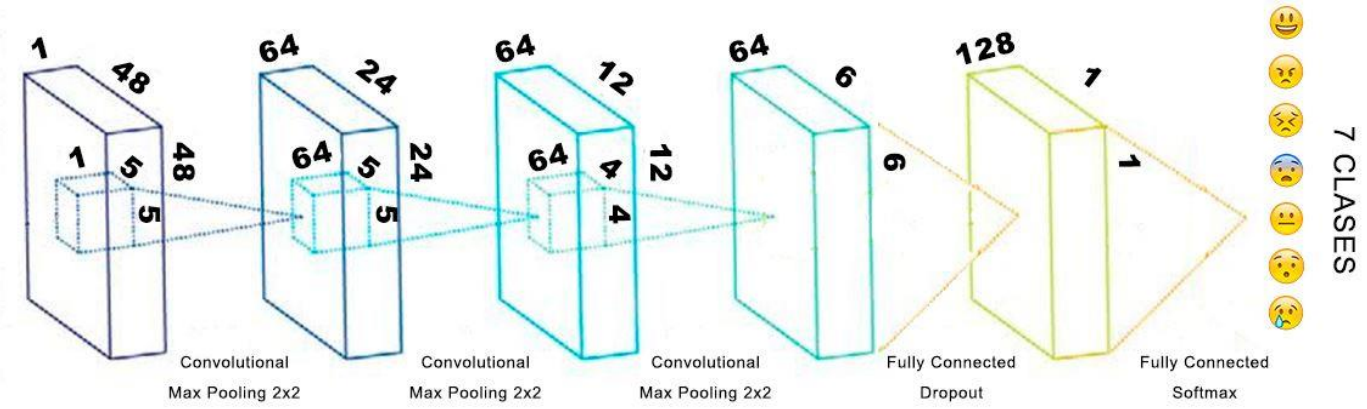

Figura 8: Arquitectura de la Red.

convolución tiene profundidad 64, en la siguiente capa cada uno de los filtros tendrá esa profundidad.

Después de cada capa de convolución, tenemos una capa de max pooling, que reduce la dimensión de la imagen a una de 6x6. Esto es debido al tamaño del max polling 2x2 y el stride de 2 píxeles. El tamaño de salida de la última capa max pooling es de 6x6x64.

La salida está conectada a dos capas fully-connected de 128 neuronas de dimensión. La salida de esta última capa es el vector de características, que será conectado al clasificador Softmax con 7 clases posibles.

Por otro lado, se ha utilizado también el modelo ya entrenado VGG16 [20]. Este, fue propuesto por K. Simonyan y A. Zisserman de la universidad de Oxford. El modelo cuenta con un 92.7% en el top-5 test accuracy de Imagenet, además de estar entrenado con más de 15 millones de imágenes y clasificar 1000 clases diferentes. En la figura 9 podemos observar la arquitectura del modelo:

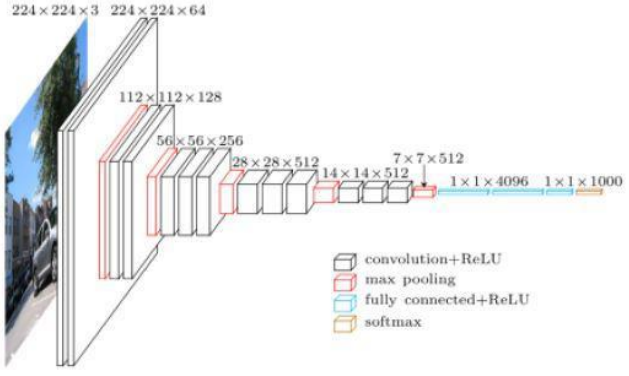

Figura 9: Arquitectura VGG16

La primera capa convolucional de la red VGG16 utiliza un filtro de profundidad 3, ya que la entrada de la red es una imagen 224x224x3 (contiene información en los tres canales de color).

VGG16 dispone de 12 capas convolucionales más, donde la profundidad viene establecida por la profundidad de la capa anterior, igual que en la red desarrollada para el proyecto.

También cuenta con 5 capas de max pooling que reducen la dimensión de la imagen a una de 7x7, puesto que se produce el mismo caso que en nuestra red. El tamaño de salida de la última capa max pooling es de 7x7x512.

La salida, como en nuestro caso, está conectada a dos capas fully-connected, pero esta vez con una dimensión de 4096 neuronas, que realizan la clasificación es realizada para 1.000 clases diferentes.

7 Experimentos y Resultados

En este apartado se van a comentar los experimentos realizados y los resultados obtenidos, utilizando matrices de confusión. De este modo, podremos observar que emociones son las mejor y peor clasificadas. El accuracy nos mostrará la precisión obtenida durante esta clasificación.

En la tabla 7 podemos observar el tiempo de entreno y los resultados obtenidos durante el entrenamiento. El accuracy obtenido es respecto al 20% del dataset que corresponde al conjunto de test.

	10K	CK	Jaffe	Mug	Rafd	Yale	All
Tiempo Entreno	1h 26m	56m	3h 11m	2h 30m	3h 39m	40m	3h 9m
Accuracy	0.95	0.43	0.738	0.9285	0.77	0.67	0.8871
Loss	0.145	1.28	1.28	0.162	0.628	1.137	0.1628

Tabla 7. Entrenamientos y resultados durante la fase de test de los datasets.

Los resultados obtenidos son bastante satisfactorios, destacan los modelos entrenados con una cantidad mayor de imágenes, entre ellos el formado por el conjunto de todos los demás datasets. Tenemos que tener en cuenta que los resultados obtenidos en fase de test no serán los mismos que con imágenes que nunca ha visto. Aunque el modelo no vea el conjunto de test durante el entrenamiento, este, pertenece al mismo dataset, por tanto las personas que aparecen como el nivel de sombras, cámara, lejanía, etc, son los mismos, así que el resultado obtenido es mejor comparado al obtenido con imágenes que no pertenecen al dataset.

Otros investigadores también han utilizado estos datasets en proyectos parecidos a este, con arquitecturas y técnicas totalmente diferentes; sus resultados se pueden observar en la tabla 8. Cabe destacar que el accuracy obtenido con Mug para ambos casos es casi el mismo. Algunos datasets se han obviado al no haber encontrado resultados de trabajos similares.

DATASET	PROYECTO	OTROS
CK	43%	81.8% [21]
JAFFE	73.8%	90.7% [22]
MUG	92.85%	97.8% [23]

Tabla 8. Comparación de accuracy de nuestros modelos con otros de diferentes proyectos similares.

A partir de este punto, se ha decidido investigar y realizar más pruebas con el modelo entrenado con todos los datasets para comprobar si la variedad de rostros afecta de forma positiva al resultado. En la tabla 9 podemos observar la matriz de confusión correspondiente a dicho modelo, los resultados que se muestran son los conseguidos en la fase de test con el 20% del dataset. Los valores numéricos de las emociones los podemos encontrar en el apartado A1 del apéndice.

Emoción Predicted		1	2	3	4	5	6	7
	1	0.87	0.08	0	0.03	0	0.02	0
	2	0.01	0.94	0.02	0	0.01	0.02	0
	3	0.02	0.05	0.89	0	0.01	0.02	0
	4	0.03	0.06	0	0.89	0	0.02	0
	5	0.01	0.05	0.01	0	0.87	0.03	0.03
	6	0.01	0.06	0.03	0.01	0	0.89	0
	7	0.01	0.06	0.03	0	0.02	0.02	0.86
	Emocion Real							
Accuracy: 88.71%								

Tabla 9. Matriz de Confusión del modelo All en la fase test.

Si observamos la tabla 9, vemos que en ningún caso lleva a cabo la clasificación con un 100%. Aunque las emociones que mejor clasifica son 1) normal, 2) feliz, 3) miedo, 4) enfado y 5) triste, obteniendo un accuracy final de 88.71%.

La Tabla 10 muestra la matriz de confusión del modelo All haciendo test con el dataset correspondiente a Jaffe.

Emoción Predicted		1	2	3	4	5	6	7
	1	0.72	0.12	0.06	0	0.09	0	0
	2	0.31	0.69	0	0	0	0	0
	3	0.12	0.03	0.69	0	0	0.16	0
	4	0.31	0	0	0.56	00	0.12	0
	5	0.09	0.06	0.16	0	0.53	0.16	0
	6	0.03	0.06	0.16	0	0.09	0.66	0
	7	0.09	0.06	0.22	0	0.31	0	0.31
	Emocion Real							
Accuracy: 59.42%								

Tabla 10. Matriz de Confusión del modelo All haciend test sobre el dataset Jaffe.

El accuracy obtenido al clasificar imágenes del dataset Jaffe no es demasiado elevado. Esto es debido a que la mayoría de las imágenes del modelo corresponden al dataset Mug y los rasgos en las expresiones faciales son bastante diferentes.

En el apartado A1 del apéndice se pueden encontrar los resultados de las diferentes pruebas que se han realizado probando el modelo All.

En la tabla 11 vemos un resumen del accuracy obtenido del modelo All probándolo con los demás datasets:

	10K	CK	Jaffe	Mug	Rafd	Yale
Accuracy	0.75	0.69	0.59	0.95	0.14	0.67

Tabla 11. Resultados del modelo All testeándolo con los demás datasets.

Al ser entrenado el modelo con la mayoría de imágenes del dataset Mug, este obtiene mejor accuracy de los demás, aunque los resultados son bastante satisfactorios con cada uno de los sets. Rafd es el único que no ha obtenido buenos resultados debido a que su conjunto de imágenes difiere demasiado con el resto. Esto es debido a la gran diferencia que tiene con las demás imágenes, como podemos ver en la figura 10.

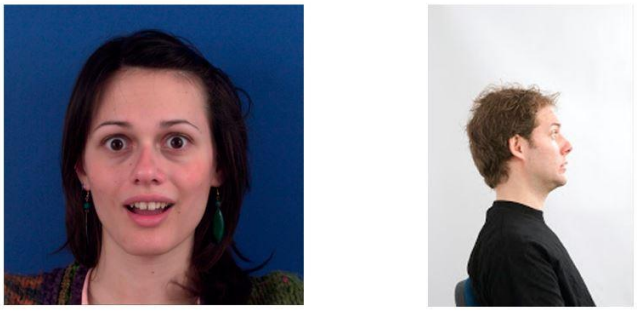

Figura 10: A la izquierda imagen del dataset Jaffe, a la derecha imagen del dataset Ralf

7.1 Pruebas en Tiempo Real

Se han realizado pruebas con todos los datasets, aunque al no trabajar en un espacio ideal (cámara profesional, focos, evitar sombras…), el resultado no es siempre el correcto; algunos estados se confunden entre ellos. Esto hace que no podamos garantizar cual funciona mejor.

En la figura 11 se muestra el funcionamiento de la herramienta utilizando el modelo All. En la esquina superior izquierda se dibuja el porcentaje de predicción para cada emoción y, justo debajo, el emoticono con la emoción predecida. Según vayamos cambiando la expresión de nuestra cara estos valores irán cambiando indicando lo predecido por el modelo.

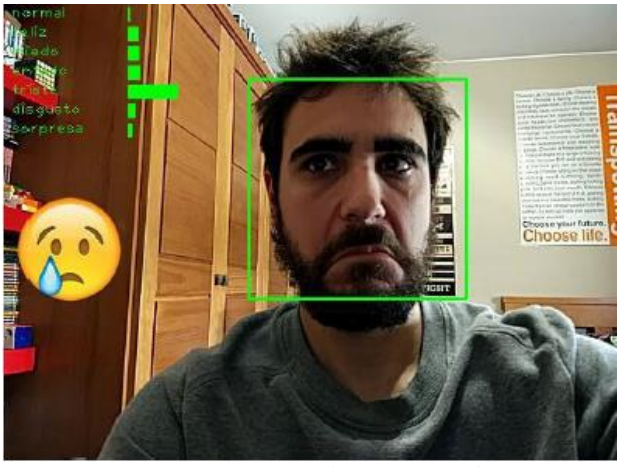

Figura 11: Prueba a tiempo real.

En el apartado A2 del apéndice se muestran más imágenes de las pruebas realizadas a tiempo real, con diferentes modelos y expesiones.

7.2 Pruebas Manuales

Las pruebas manuales se han realizado con fotografías de buena calidad, bien iluminadas y cuya expresión facial era bastante exagerada, así, se asegura que el error sea mínimo y que nuestro modelo clasifique correctamente la emoción.

En la figura 12 se puede observar que el funcionamiento es el mismo que en tiempo real, aunque en este caso la aplicación guarda una copia de la fotografía con los datos obtenidos, además de escribir un log con el accuracy y la emoción detectada. En esta prueba también se ha utilizado el modelo All.

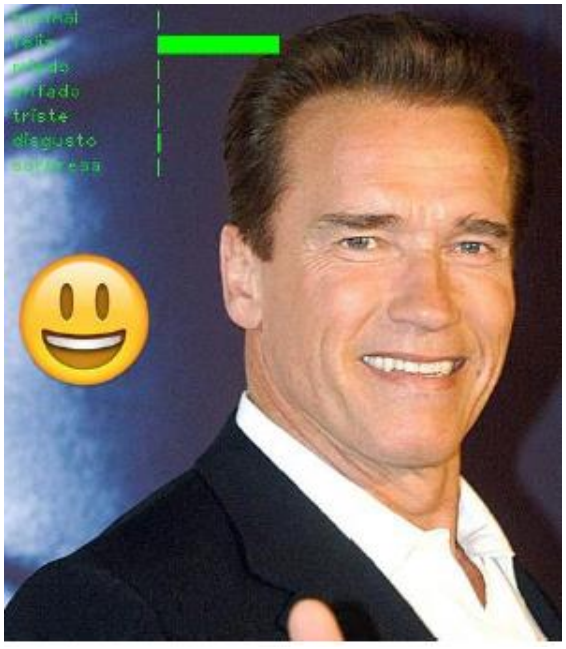

Figura 12: Prueba Manual.

La figura 12 se ha probado con todos los modelos. El accuracy obtenido por cada uno de ellos prediciendo el estado feliz se puede observar en la figura 13. En este caso todas han predicho bien menos Rafd.

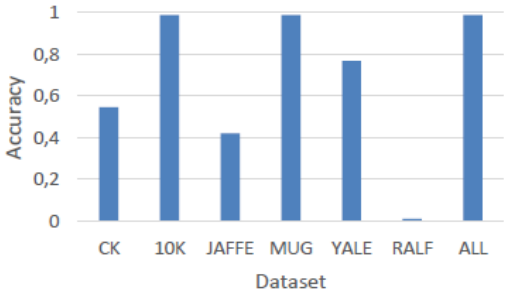

Figura 13: Prueba manual con emoción feliz.

En el apartado A3 del apéndice se muestran más pruebas realizadas con otras imágenes y diferentes modelos.

7.3 FineTune de VGG16

Se ha procedido a realizar FineTune de este modelo ya entrenado, pero no se han obtenido buenos resultados debido a limitaciones de hardware, entre otras, en el momento de su entrenamiento:

- Tamaño de las entradas 224x224x3: al tener que cambiar nuestras imágenes a ese tamaño se genera mucho ruido y pérdida.
- Tarjeta gráfica GeForce GTX970: no es capaz de iniciar el proceso de finetune y se ha tenido que usar la CPU.
- Debido al último punto, el tiempo de tratamiento de una imagen es 100 veces más lento.

En la tabla 12 se muestra que el entrenamiento no era viable. Esto se debe a que el modelo está entrenado con millones de imágenes diferentes (no solo rostros), y ajustar todo correctamente supone un gran trabajo computacional.

ACCURACY	LOSS	TIEMPO DE ENTRENO
16.6%	22.24	18h 48min

Tabla 12. Resultados VGG16.

8 Conclusiones

En este TFG se ha elaborado un sistema de reconocimiento de emociones a partir del uso de redes neuronales, permitiendo construir modelos que clasifiquen entre siete emociones faciales diferentes.

A partir de la experimentación realizada se concluye que el dataset con mejores resultados ha sido el que incluía todas las imágenes y, que la emoción clasificada con mayor facilidad (mayor tasa de aciertos) ha sido feliz, seguramente debido a que el rasgo facial presenta una gran diferencia con el resto. Hay que tener en cuenta que las redes neuronales más famosas están entrenadas durante largos períodos de tiempo y con un gran volumen

de datos, mientras que en nuestro caso los entrenos solamente han durado horas y como máximo se ha contado con 109.648 imágenes diferentes para realizar el training, la validación y el test.

Los resultados obtenidos en tiempo real no son del todo fiables, a no ser que se exagere mucho la expresión. También influye en gran medida la luminosidad del lugar, calidad de la cámara, las sombras… hecho que ha limitado bastante a la hora de realizar pruebas y asegurar su buen funcionamiento.

El resultado del uso manual depende de la imagen usada. Hay emociones que son clasificadas mejor que otras, también dependiendo del modelo empleado. Esto lleva a pensar que hubiese sido mejor centrarse en un par de datasets y probar diferentes arquitecturas de la red para comparar resultados.

Cabe destacar que no se ha podido obtener buen resultado con la red neuronal VGG16. Aunque se haya podido entender y realizar el trabajo de finetune, se necesitaban más horas de re-entrenamiento, ya que el modelo original fue entrenado con más de 15 millones de imágenes y clasifica en 1.000 clases. Es de suponer que con 18 horas de cómputo con la CPU no es suficiente, por lo que quedaría como trabajo futuro probar la aplicación con un modelo entrenado, además de intentar utilizar Caffe para esto último.

AGRADECIMIENTOS

Agredecer a mi tutora de proyecto Katerine Díaz, por la ayuda otorgada en este trabajo en todo momento, respondiendo mis dudas cuando se las planteaba.

También agradecer a mis compañeros y amigos Enoc Martínez y Oscar Prades por la ayuda que me han dado.

Por último a mi pareja Claudia Núñez por apoyarme y estar a mi lado durante el transcurso del proyecto.

BIBLIOGRAFÍA

[1] D. Matsumoto, H. S. Hwang. 2011. Reading facial expressions of emotion. Science Brief [Online] Available: http://www.apa.org/science/about/psa/2011/05/facial-expressions.aspx

[2] J. Jarkiewicz, R. Kocielnik, K. Marasek. "Anthropometric Facial Emotion Recognition". Polish-Japanese Institute of Information Technology.Pages 188-189 2009.

[3] P. Ekman. "Facial Expressions". University of California, San Francisco, CA, USA. Pages: 302-318. 2013.

[4] Microsoft. Azure Cognitive Services Documentation – Tutorials, API Reference, page 73. 2016.

[5] Modar (JR) Alaoui, Applying Al In A New Era Of Predictive Emotional Analytics. Eyeris. Pages 10-16. 2016.

[6] D. McDuff, A. Mahmoud, M. Amr, J. Turcot, M. Mavadati, R. Kaliouby. "AFFDEX SDK: A Cross-Platform Real-Time Multi-Face Expression Recognition Toolkit". Waltham, Ma, USA. Page 2. 2016.

[7] L. Entis. 2015. Ads can now tell how fast your heart is beating. The Guardian. [Online] Available: https://www.theguardian.com/lifeandstyle/2015/jul/31/biometric-data-apple-wimbledon-facebook-mindshare-affectiva-unilever-coca-cola-mars

[8] M. Lin, Q. Chen, S. Yan. "Network In Network". National University of Singapore, Singapore. Pages 2 – 3. 2014.

[9] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". University of Toronto. Pages 1928-1955. 2014.

[10] R. Rojas. "Neural Networks". Springer-Verlag. Pages 151 – 184. 1996.

[11] A. Ng. "CS229 Lecture Notes". Stanford. Pages 26 – 30. 2016.

[12] J. Bergstra, O. Breuleux, F. Bastian, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, Y. Bengio. "Theano: A CPU and GPU Math Compiler in Python". Python in science conf. Pages 1 – 7. 2010.

[13] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrel. "Caffe: Convolutional Architecture for Fast Feature Embedding". UC Berkeley EECS. Pages 1- 4. 2014.

[14] W. A. Bainsbridge, P. Isola, A. Oliva. "The intrinsic Memorabily of Face Photographs". Journal of Experimental Psychology: General. Pages 1323 – 1334. 2013.

[15] T. Kanade, J. F. Cohn, Y. Tian. "Comprehensive database for facial expression analysis". University of Pittsburgh. Pages 46 - 53. 2000.

[16] M. J. Lyons, J. Budynek, S. Akamatsu. "Automatic Classification of Single Facial Images". IEEE Computer Society Washington, DC, USA. Pages 1357-1362. 1999.

[17] D. Ghimire, J. Lee, Z. Li, S. Jeong, S. H. Park, H. S. Choi. "Recognition of Facial Expressions Based on Tracking and Selection of Discriminative Geometric Features". School of Computing Science, Burnaby, Canada. Pages 35 – 44. 2015.

[18] O. Langner, R. Dotsch, G. Biljstra, D. H. J. Wigboldus, S. T. Hawk, A. Knippenberg. "Presentation and validation of the Radboud Faces Database". Radboud University Nijmegen. Pages 1377 – 1385. 2010.

[19] A. Georghiades. 1997. Yale face database. Center for computational Vision and Control at Yale University. [Online] Available: http://vision.ucsd.edu/content/yale-face-database

[20] K. Simonyan, A. Zisserman. "Very Deep Convolutional Networks For Large-Scale Image Recognition". University of Oxford. Pages: 1-14. 2015.

[21] P. Khorrami, T. L. Paine, T. S. Huang." Do Deep Neural Networks Learn Facial Actions Unit When Doing Expression Recognition?" Beckman Institute for Advanced Science and Technology. Page 20-22. 2015.

[22] D. Hamester, P. Barros, S. Wermter. "Face Expression Recognition with a 2-channel Convolutional Neural Network", University of Hamburg. Page 5. 2015

[23] D. Ghimire, J. Lee, Z. Li, S. Heong, S Hyun Park, H. Sub Choi. Recognition of Facial Expressions Based on Tracking and Selection of Discriminative Geometric Features. International Journal of Multimedia and Ubiquitous Engineering. Pages 35 – 44. 2015.

APÉNDICE

A1. RESULTADOS FASE TEST

Las emociones corresponen a: 1 = Normal, 2 = Feliz, 3 = Miedo, 4 = Enfado, 5 = Triste, 6 = Disgusto, 7 = Sorpresa. Los resultados pueden variar.

En la tabla 13 se muestra la matriz de confusión del modelo creado con el dataset CK y testeado con CK, solamente clasifica en 5 emociones diferentes.

Emoción Predicted	1	2	3	4	5	6	7
1	0.29	0.21	0.14	0.21	0.14		
2	0.07	0.86	0.07	0	0		
3	0	0.21	0.5	0.14	0.14		
4	0.14	0	0.07	0.36	0.43		
5	0.29	0	0.29	0.29	0.14		
6							
7							
Emocion Real							
Accuracy: 43%							

Tabla 13. Matriz de Confusión del modelo CK en la fase test.

En la tabla 14 se muestra la matriz de confusión del modelo creado con el dataset 10K y testeado con 10K, solamente clasifica en 3 emociones diferentes.

Emoción Predicted	1	2	3	4	5	6	7
1	0.93	0		0.07			
2	0	0.93		0.07			
3							
4	0.01	0		0.99			
5							
6							
7							
Emocion Real							
Accuracy: 95%							

Tabla 14. Matriz de Confusión del modelo 10K en la fase test.

En la tabla 15 se muestra la matriz de confusión del modelo creado con el dataset Jaffe y testeado con Jaffe, clasifica todas las emociones.

Emoción Predicted	1	2	3	4	5	6	7
1	0.67	0.17	0	0	0.17	0	0
2	0.17	0.83	0	0	0	0	0
3	0.17	0.17	0.5	0	0	0.17	0
4	0	0	0	1	0	0	0
5	0	0	0.17	0	0.67	0.17	0
6	0	0	0	0	0	1	0
7	0	0.17	0.17	0	0.17	0	0.50
Emocion Real							
Accuracy: 73.85%							

Tabla 15. Matriz de Confusión del modelo Jaffe en la fase test.

En la tabla 16 se muestra la matriz de confusión del modelo creado con el dataset Mug y testeado con Mug, clasifica todas las emociones.

Emoción Predicted	1	2	3	4	5	6	7
1	0.86	0.02	0.01	0.04	0	0.06	0.01
2	0	0.95	0.01	0	0	0.03	0.01
3	0.01	0.02	0.91	0.02	0.01	0.02	0.02
4	0	0	0	0.97	0	0.02	0
5	0.01	0.01	0.01	0	0.93	0.02	0.02
6	0.01	0	0.01	0.01	0	0.96	0
7	0	0.02	0.02	0.01	0.02	0.01	0.92
Emocion Real							
Accuracy: 92.85%							

Tabla 16. Matriz de Confusión del modelo Mug en la fase test.

En la tabla 17 se muestra la matriz de confusión del modelo creado con el dataset Rafd y testeado con Rafd, clasifica todas las emociones.

Emoción Predicted	1	2	3	4	5	6	7
1	0.86	0	0.09	0	0.05	0	0
2	0	0.62	0	0	0	0.38	0
3	0	0	0.75	0	0.2	0	0.05
4	0.17	0	0	0.72	0.06	0.06	0
5	0.12	0	0	0.09	0.75	0.03	0
6	0	0	0.08	0	0	0.92	0
7	0	0	0.17	0	0	0	0.83
Emocion Real							
Accuracy: 77.85%							

Tabla 17. Matriz de Confusión del modelo Rafd en la fase test.

En la tabla 18 se muestra la matriz de confusión del modelo creado con el dataset Yale y testeado con Yale, solamente clasifica en 3 emociones diferentes.

Emoción Predicted	1	2	3	4	5	6	7
1	0.67	0.33			0		
2	0	0.67			0.33		
3							
4							
5	0	0.33			0.67		
6							
7							
Emocion Real							
Accuracy: 67%							

Tabla 18. Matriz de Confusión del modelo Yale en la fase test.

Los siguientes resultados muestran el modelo All testeado con las partes de test de todos los datasets.

En la tabla 19 se muestra la matriz de confusión obtenida haciendo la fase de test con el conjunto de CK.

Emoción Predicted	1	2	3	4	5	6	7
1	0.69	0.1	0.09	0.01	0.1		
2	0.04	0.89	0.06	0	0.01		
3	0.14	0.11	0.51	0.03	0.14		
4	0.09	0.04	0.04	0.67	0.16		
5	0.03	0.03	0.1	0.16	0.69		
6							
7							
Emocion Real							
Accuracy: 69%							

Tabla 19. Matriz de Confusión del modelo All testeando con los datos de CK.

En la tabla 20 se muestra la matriz de confusión obtenida haciendo la fase de test con el conjunto de 10K.

Emoción Predicted	1	2	3	4	5	6	7
1	0.55	0.27		0.19			
2	0.02	0.96		0.02			
3							
4	0.14	0.14		0.72			
5							
6							
7							
Emocion Real							
Accuracy: 75.33%							

Tabla 20. Matriz de Confusión del modelo All testeando con los datos de 10K.

En la tabla 21 se muestra la matriz de confusión obtenida haciendo la fase de test con el conjunto de Jaffe.

Emoción Predicted	1	2	3	4	5	6	7
1	0.72	0.12	0.06	0	0.09	0	0
2	0.31	0.69	0	0	0	0	0
3	0.12	0.03	0.69	0	0	0.16	0
4	0.31	0	0	0.56	00	0.12	0
5	0.09	0.06	0.16	0	0.53	0.16	0
6	0.03	0.06	0.16	0	0.09	0.66	0
7	0.09	0.06	0.22	0	0.31	0	0.31
Emocion Real							
Accuracy: 59.42%							

Tabla 21. Matriz de Confusión del modelo All testeando con los datos de Jaffe.

En la tabla 22 se muestra la matriz de confusión obtenida haciendo la fase de test con el conjunto de Mug.

Emoción Predicted	1	2	3	4	5	6	7
1	0.99	0	0.01	0	0	0	0
2	0.01	0.96	0.02	0	0	0.01	0
3	0.02	0	0.95	0	0.01	0.01	0
4	0	0	0	0.99	0	0.01	0
5	0.01	0	0.01	0	0.95	0.01	0
6	0.01	0	0.03	0	0.01	0.94	0
7	0.01	0.01	0.03	0	0.02	0.01	0.93
Emocion Real							
Accuracy: 95.85%							

Tabla 22. Matriz de Confusión del modelo All testeando con los datos de Mug.

En la tabla 23 se muestra la matriz de confusión obtenida haciendo la fase de test con el conjunto de Rafd.

Emoción Predicted	1	2	3	4	5	6	7
1	0.01	0.70	0	0.02	0.01	0.26	0.01
2	0	0.75	0	0.01	0.01	0.22	0.01
3	0.01	0.74	0	0.01	0.01	0.23	0
4	0.02	0.69	0	0.03	0.01	0.25	0
5	0.01	0.72	0	0.03	0.01	0.23	0
6	0	0.71	0	0.03	0.01	0.24	0
7	0.01	0.74	0	0.01	0	0.22	0
Emocion Real							
Accuracy: 14.85%							

Tabla 23. Matriz de Confusión del modelo All testeando con los datos de Mug.

En la tabla 24 se muestra la matriz de confusión obtenida haciendo la fase de test con el conjunto de Yale.

Emoción Predicted	1	2	3	4	5	6	7
1	0.67	0.33			0		
2	0.33	0.67			0		
3							
4							
5	0.67	0			0.33		
6							
7							
Emocion Real							
Accuracy: 67%							

Tabla 24. Matriz de Confusión del modelo All testeando con los datos de Yale.

A2. Pruebas en Tiempo Real

Las figuras 14 y 15 son una muestra del funcionamiento del modelo All a tiempo real.

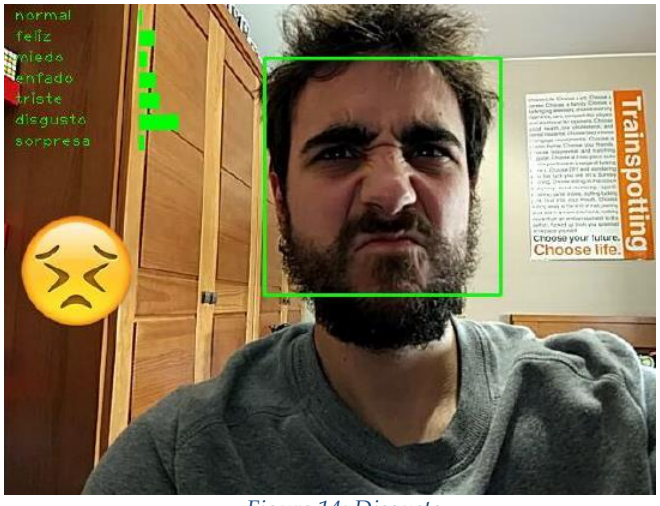

Figura 14: Disgusto

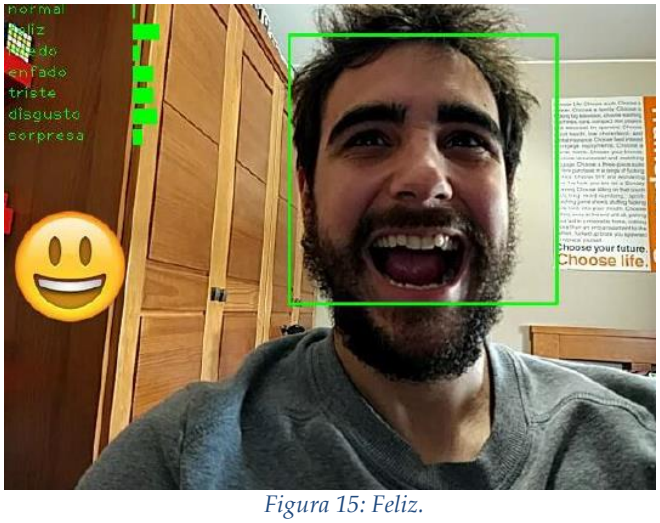

Figura 15: Feliz.

A3. Pruebas Manuales

La siguiente prueba ha sido con la figura 16 la cual se tendría que clasificar como triste.

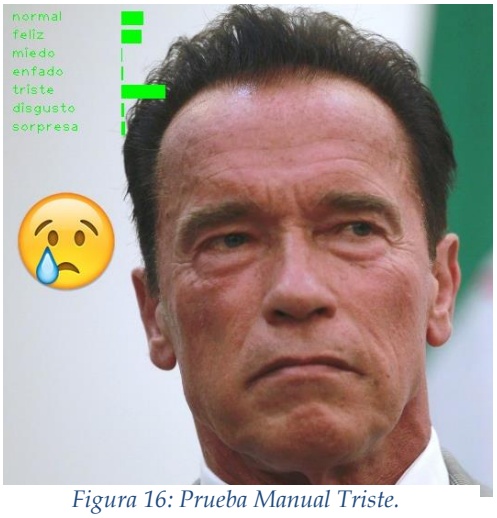

Figura 16: Prueba Manual Triste.

En esta prueba los modelos que mejor resultado dan es Mug, 10K y Yale indican que es miedo, viendo la figura 15 está claro que se podría confundir.

En la figura 16 se observa el accuracy de cada uno de ellos de la predicción del estado triste.

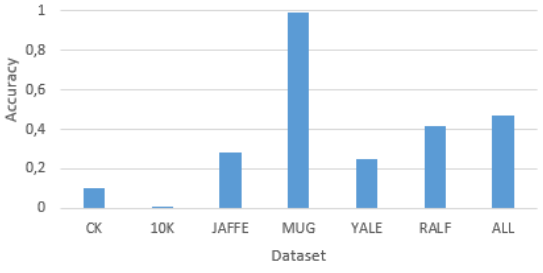

Figura 16: Prueba manual con emoción triste.