



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

DETECCIÓN DE EMOCIONES Y RECONOCIMIENTO FACIAL
UTILIZANDO APRENDIZAJE PROFUNDO

JARAMILLO ROMERO SEGUNDO EZEQUIEL
INGENIERO DE SISTEMAS

MACHALA
2020



UTMACH

FACULTAD DE INGENIERÍA CIVIL
CARRERA DE INGENIERÍA DE SISTEMAS

DETECCIÓN DE EMOCIONES Y RECONOCIMIENTO FACIAL
UTILIZANDO APRENDIZAJE PROFUNDO

JARAMILLO ROMERO SEGUNDO EZEQUIEL
INGENIERO DE SISTEMAS

MACHALA
2020



UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE INGENIERÍA DE SISTEMAS

EXAMEN COMPLEXIVO

DETECCIÓN DE EMOCIONES Y RECONOCIMIENTO FACIAL UTILIZANDO
APRENDIZAJE PROFUNDO

JARAMILLO ROMERO SEGUNDO EZEQUIEL
INGENIERO DE SISTEMAS

RIVAS ASANZA WILMER BRAULIO

MACHALA, 19 DE FEBRERO DE 2020

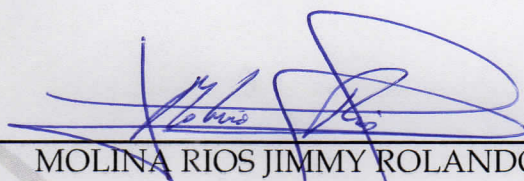
MACHALA
19 de febrero de 2020

Nota de aceptación:

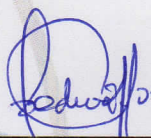
Quienes suscriben, en nuestra condición de evaluadores del trabajo de titulación denominado DETECCIÓN DE EMOCIONES Y RECONOCIMIENTO FACIAL UTILIZANDO APRENDIZAJE PROFUNDO, hacemos constar que luego de haber revisado el manuscrito del precitado trabajo, consideramos que reúne las condiciones académicas para continuar con la fase de evaluación correspondiente.



NOVILLO VICUÑA JOHNNY PAUL
0702947409
ESPECIALISTA SUPLENTE



MOLINA RÍOS JIMMY ROLANDO
0703691980
ESPECIALISTA 2



REDROVAN CASTILLO FAUSTO FABIAN
0702739228
ESPECIALISTA 3

Fecha de impresión: jueves 20 de febrero de 2020 - 08:07

Sistema de detección de emociones y reconocimiento facial a través del aprendizaje profundo

por Ezequiel Jaramillo

Fecha de entrega: 10-feb-2020 08:31p.m. (UTC-0500)

Identificador de la entrega: 1255133175

Nombre del archivo: JARAMILLO_ROMERO_SEGUNDO_EZEQUIEL_PT-011119.docx (4.14M)

Total de palabras: 4184

Total de caracteres: 22869

CLÁUSULA DE CESIÓN DE DERECHO DE PUBLICACIÓN EN EL REPOSITORIO DIGITAL INSTITUCIONAL

El que suscribe, JARAMILLO ROMERO SEGUNDO EZEQUIEL, en calidad de autor del siguiente trabajo escrito titulado DETECCIÓN DE EMOCIONES Y RECONOCIMIENTO FACIAL UTILIZANDO APRENDIZAJE PROFUNDO, otorga a la Universidad Técnica de Machala, de forma gratuita y no exclusiva, los derechos de reproducción, distribución y comunicación pública de la obra, que constituye un trabajo de autoría propia, sobre la cual tiene potestad para otorgar los derechos contenidos en esta licencia.

El autor declara que el contenido que se publicará es de carácter académico y se enmarca en las disposiciones definidas por la Universidad Técnica de Machala.

Se autoriza a transformar la obra, únicamente cuando sea necesario, y a realizar las adaptaciones pertinentes para permitir su preservación, distribución y publicación en el Repositorio Digital Institucional de la Universidad Técnica de Machala.

El autor como garante de la autoría de la obra y en relación a la misma, declara que la universidad se encuentra libre de todo tipo de responsabilidad sobre el contenido de la obra y que asume la responsabilidad frente a cualquier reclamo o demanda por parte de terceros de manera exclusiva.

Aceptando esta licencia, se cede a la Universidad Técnica de Machala el derecho exclusivo de archivar, reproducir, convertir, comunicar y/o distribuir la obra mundialmente en formato electrónico y digital a través de su Repositorio Digital Institucional, siempre y cuando no se lo haga para obtener beneficio económico.

Machala, 19 de febrero de 2020



JARAMILLO ROMERO SEGUNDO EZEQUIEL
0705642379

Sistema de detección de emociones y reconocimiento facial a través del aprendizaje profundo

INFORME DE ORIGINALIDAD

6%

INDICE DE SIMILITUD

2%

FUENTES DE INTERNET

0%

PUBLICACIONES

5%

TRABAJOS DEL ESTUDIANTE

FUENTES PRIMARIAS

1

Submitted to Universidad Carlos III de Madrid

Trabajo del estudiante

1%

2

reunir.unir.net

Fuente de Internet

1%

3

Submitted to EP NBS S.A.C.

Trabajo del estudiante

1%

4

Submitted to Universidad Estatal a Distancia

Trabajo del estudiante

1%

5

Submitted to Universidad Tecnologica del Peru

Trabajo del estudiante

<1%

6

Submitted to Universidad San Ignacio de Loyola

Trabajo del estudiante

<1%

7

documents.mx

Fuente de Internet

<1%

8

www.scn.org

Fuente de Internet

<1%

9	Submitted to UNILIBRE Trabajo del estudiante	<1 %
10	www.camisea.com.pe Fuente de Internet	<1 %
11	Submitted to Universidad Nacional de Educación a Distancia Trabajo del estudiante	<1 %
12	Submitted to Unviersidad de Granada Trabajo del estudiante	<1 %
13	Jose Daniel Britos. "", IEEE Latin America Transactions, 9/2007 Publicación	<1 %
14	Submitted to Pontificia Universidad Catolica del Peru Trabajo del estudiante	<1 %
15	www.slideshare.net Fuente de Internet	<1 %
16	Submitted to Universidad Cesar Vallejo Trabajo del estudiante	<1 %
17	Submitted to Universidad de Almeria Trabajo del estudiante	<1 %
18	www.scielo.org.mx Fuente de Internet	<1 %
19	www.fcen.uba.ar	

Fuente de Internet

<1%

20

Submitted to Consorcio CIXUG

Trabajo del estudiante

<1%

Excluir citas

Apagado

Excluir coincidencias

Apagado

Excluir bibliografía

Apagado

DEDICATORIA

Este trabajo se lo dedico a mis padres, José Ufredo Jaramillo y María del Carmen Romero, por su apoyo en cada una de las etapas de mi vida, por sus consejos y el cariño que me brindan.

Sr. Segundo Ezequiel Jaramillo Romero

AGRADECIMIENTO

A Dios por las bendiciones que derraman sobre mí, por haberme dado una familia, salud, amigos, y por las muestras de su amor hasta en los más pequeños detalles.

A mis padres, José Ufredo Jaramillo y María del Carmen Romero, quienes han estado incondicionalmente apoyándome y motivándome en esta etapa de formación profesional.

A mis familiares que han estado al pendiente, y sobre todo a mis tíos, Franco Jaramillo y Lupe Delgado, por el apoyo recibido durante todo este tiempo de preparación.

A los docentes de la Carrera de Ingeniería de Sistemas, de la Universidad Técnica de Machala, quienes han facilitado sus conocimientos y experiencias.

A mis compañeros de la Carrera de Ingeniería de Sistemas, con quienes he compartido tareas, proyectos, anécdotas y diferentes actividades hasta alcanzar la culminación de mis estudios de educación superior.

A mi tutor, Ing. Wilmer Rivas, por ser un guía en el transcurso de este caso práctico.

Sr. Segundo Ezequiel Jaramillo Romero

RESUMEN

Con los avances tecnológicos en el área de inteligencia artificial, y en especial gracias las técnicas de aprendizaje profundo, resulta posible la clasificación de imágenes y el reconocimiento facial. En este proyecto se ha creado una arquitectura de aprendizaje profundo con cuatro capas para la detección de emociones, utilizando cinco clases (enojado, feliz, neutral, triste y sorprendido) del *dataset* de *Kaggle* denominado "Learn facial expressions from an image" (FER2013), empleando un total de 15,000 imágenes para el entrenamiento de la red neuronal convolucional en la plataforma *Kaggle Notebooks*, las herramientas utilizadas fueron el lenguaje *Python*, la biblioteca *Keras*, *Tensorflow*, *OpenCV* y el algoritmo pre-entrenado "Frontal Face" de Haar Cascade, obteniendo una precisión del 67.13%. Mientras que para el reconocimiento facial se consideró la biblioteca *Face Recognition*, la misma que posee una precisión del 99.38%. Por último, se desarrolló un servicio web basado en *Django* para responder a las solicitudes que se realizan desde una aplicación web, implementada en *Laravel*, para la comunicación de los dos *frameworks* se utilizó la librería de *Axios*, el acceso a la aplicación se lo realiza a través del reconocimiento del rostro, una vez iniciada la sesión es posible realizar pruebas de detección de emociones y de reconocimiento facial.

PALABRAS CLAVES: Clasificación de imágenes, Detección de emociones, Haar Cascade, *OpenCV*, Reconocimiento facial, Redes neuronales convoluciones.

ABSTRACT

With technological advances in the area of artificial intelligence, and especially thanks to deep learning techniques, image classification and facial recognition are possible. This project has created a deep learning architecture with four layers for the detection of emotions, using five classes (angry, happy, neutral, sad and surprised) of the *Kaggle* dataset called "Learn facial expressions from an image" (FER2013) , using a total of 15,000 images for the training of the convolutional neural network in the *Kaggle Notebooks* platform, the tools used were the *Python* language, the *Keras* library, *Tensorflow*, *OpenCV* and the pre-trained algorithm "Front Face" of Haar Cascade, obtaining an accuracy of 67.13%. While the *Face Recognition* library was considered for facial recognition, it has an accuracy of 99.38%. Finally, a web service based on *Django* was developed to respond to requests made from a web application, implemented in *Laravel*, for the communication of the two frameworks the *Axios* library was used, access to the application is made through face recognition, once the session is started, it is possible to perform tests for the detection of emotions and facial recognition.

KEYWORDS: Image classification, Emotion detection, Haar Cascade, *OpenCV*, Facial recognition, Neural networks convolutions.

CONTENIDO

	Pág.
DEDICATORIA.....	1
AGRADECIMIENTO.....	2
RESUMEN	3
CONTENIDO.....	5
LISTA DE FIGURAS	6
LISTA DE TABLAS.....	7
INTRODUCCIÓN	8
Marco Contextual.....	9
Problema	9
Objetivo General.....	9
1. DESARROLLO.....	10
1.1 Marco Teórico	10
1.2 Solución del Problema	13
1.3 Resultados	17
2. CONCLUSIONES	19
BIBLIOGRAFÍA	20

LISTA DE FIGURAS

	Pág.
Figura 1. Pasos generales para predecir una imagen	10
Figura 2. Capa convolucional	10
Figura 3. Función de activación <i>relu</i>	11
Figura 4. Agrupación o submuestreo.....	11
Figura 5. Capa totalmente conectada.....	11
Figura 6. Filtros denominados características de Haar	12
Figura 7. Arquitectura del sistema	14
Figura 8. Arquitectura de la red neuronal convolucional	15
Figura 9. Pasos para el reconocimiento facial	16
Figura 10. Estructura de fotografías por persona	17
Figura 11. Resultados del entrenamiento	17
Figura 12. Inicio de sesión mediante reconocimiento facial	18
Figura 13. <i>Dataset</i> de <i>Kaggle</i> FER2013	22
Figura 14. Representación resumida del modelo	23
Figura 15. Expresión neutral	24
Figura 16. Expresión de una persona feliz	24
Figura 17. Expresión de una persona sorprendida	25
Figura 18. Expresión de una persona enojada	25
Figura 19. Expresión de una persona triste	26
Figura 20. Lista de personas registradas.....	27
Figura 21. Registro de una nueva persona.....	27
Figura 22. Edición de la información de una determinada persona	28
Figura 23. Componentes de <i>visual studio</i>	29
Figura 24. Instalación de <i>cmake</i> y <i>face recognition</i>	29
Figura 25. Toma de fotografías por persona.....	30
Figura 26. Generar un archivo con características únicas por cada cara.....	30
Figura 27. Test de reconocimiento facial con una persona.....	31
Figura 28. Test de reconocimiento facial con dos personas	31
Figura 29. Test de reconocimiento facial con tres personas	32
Figura 30. Test de reconocimiento facial con personas no registradas.....	32

LISTA DE TABLAS

	Pág.
Tabla 1. Herramientas para la detección de emociones y reconocimiento facial	14
Tabla 2. Herramientas adicionales para la detección de emociones	14
Tabla 3. Cantidad de imágenes por clase del <i>dataset</i> FER2013	15
Tabla 4. Herramientas adicionales para el reconocimiento facial	16
Tabla 5. Matriz de confusión	18

INTRODUCCIÓN

Las redes neuronales convolucionales (CNN) han mostrado un aumento del rendimiento de diversas tareas de procesamiento de imágenes, como la segmentación, eliminación de ruido, reconocimiento de objetos, reconocimiento de rostros y también en el reconocimiento de emociones faciales [1].

El reconocimiento facial es un proceso a través del cual se identifican a las personas analizando las características faciales. Esta tecnología es ampliamente aplicada en información de seguridad, acceso a áreas controladas, tarjetas inteligentes y tecnología de vigilancia. [2]

En este caso práctico se crea una red neuronal convolucional para la detección de emociones, por lo cual se ha seleccionado una base de datos de imágenes, específicamente de emociones para posteriormente hacer la respectiva clasificación, y para el reconocimiento facial se utiliza la librería *face recognition*.

El entrenamiento de la red se lo realizó en un núcleo de *Kaggle*, brindando potencia de cómputo, acceso a GPU y a la mayoría de bibliotecas necesarias para el aprendizaje profundo, una vez finalizado el entrenamiento se cargó el modelo y pesos al proyecto de *Django*.

El objetivo de esta investigación consiste en implementar un sistema de detección de emociones y reconocimiento facial a través del aprendizaje profundo, específicamente utilizando la técnica de redes neuronales convolucionales y la librería *face recognition*. Para esto se utilizaron los *frameworks* de *Django* y *Laravel*, el primero se lo utilizó porque se basa en el lenguaje de programación *Python*, lo que posibilita el uso de las herramientas de *OpenCV*, el algoritmo pre-entrenado "Frontal Face" de Haar Cascade y los pesos generados en el entrenamiento de la red neuronal, mientras que *Laravel* se lo empleó por su simplicidad para construir el aplicativo, efectuando la comunicación a través de la librería *Axios*.

El presente documento se estructura de tres capítulos, en el primer capítulo se presenta la introducción, contextualización, problemática y los objetivos de este caso práctico. En el capítulo 2, se presentan varias definiciones acordes a la investigación, así mismo se adjunta información acerca de la solución propuesta y un análisis de los resultados alcanzados. Por último, en el capítulo 3, se muestran las conclusiones que se han obtenido al finalizar el proceso investigativo y práctico.

Marco Contextual

El aprendizaje profundo es un enfoque que puede automatizar el proceso de extracción de características y es efectivo en la tarea de clasificación de imágenes, el cual ha llegado a tener una precisión superior al rendimiento de reconocimiento humano [3]. Esta técnica es muy útil en varios dominios, entre los que están la detección de emociones y el reconocimiento facial.

La detección de emociones tiene aplicabilidad en varios dominios, como el diagnóstico de enfermedades mentales y la detección de interacción social humana [4]. Así mismo el reconocimiento facial se utiliza para realizar pagos, en el desbloqueo de dispositivos, hasta en aspectos de seguridad.

Problema

Hoy en día, se considera un factor crítico la detección de emociones, ya que se encuentran relacionadas con la salud, y tienen repercusión tanto en la vida personal como profesional, es por ello que existe la necesidad de utilizar técnicas de aprendizaje profundo para la detección de emociones.

Por otra parte, los sistemas web están expuestos a varias amenazas, como los ataques de fuerza bruta en un intento por romper los *hashes* de contraseñas, esto se puede evitar haciendo uso del reconocimiento facial en el inicio de sesión de las aplicaciones.

Objetivo General

Implementar un sistema web para la detección de emociones y reconocimiento facial utilizando técnicas de aprendizaje profundo.

Objetivos Específicos

- Crear una red neuronal convolucional para la detección de emociones utilizando la base de datos de imágenes FER2013 de la plataforma *Kaggle*.
- Utilizar la librería *Face Recognition* para el reconocimiento facial.
- Crear un servicio web en *Django* para responder a las peticiones de detección de emociones y reconocimiento del rostro.
- Desarrollar una aplicación web en el *framework Laravel* para hacer uso del servicio web.

1. DESARROLLO

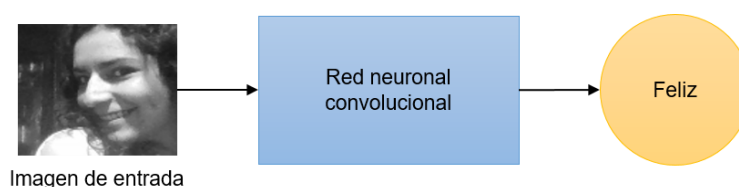
1.1 Marco Teórico

1.1.1 Redes Neuronales Convolucionales

Redes neuronales convolucionales (CNN) es una clase de red neuronal, que corresponde a una técnica del aprendizaje profundo, que presentan un alto rendimiento en la clasificación de imágenes [5] [6].

Es muy fácil comprender los pasos que realiza una CNN, ver figura 1, de manera muy general se resume en: colocar una imagen de entrada en la red, esta imagen atraviesa la parte convolucional de la red y, por último, la red predice el objeto en la imagen.

Figura 1. Pasos generales para predecir una imagen

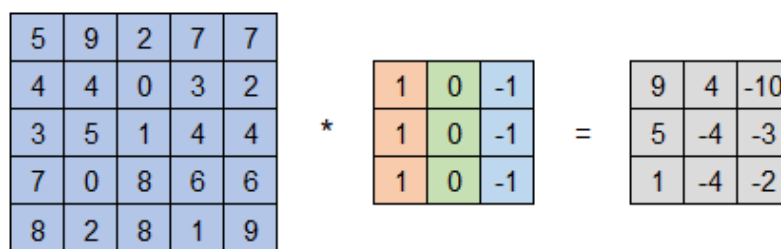


Fuente: Elaboración propia

Existen cuatro componentes básicos de CNN que son: capa convolucional, no linealidad (*relu*), agrupación o submuestreo y las capas completamente conectadas.

En la capa convolucional, se extraen las características del objeto en la imagen y se reduce el tamaño, obteniendo cálculos más rápidos. Es aquí donde la red, aprende patrones específicos dentro de la imagen. La figura 2 demuestra el proceso de extracción de características.

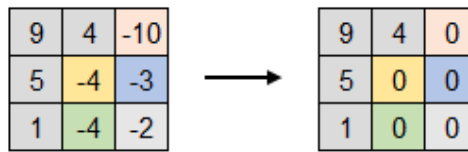
Figura 2. Capa convolucional



Fuente: Elaboración propia

La función de activación *relu* se usa para permitir la no linealidad, en este componente se reemplazan los valores negativos de los píxeles por cero, tal y como se muestra en la figura 3. La función *relu* se representa como $f(x) = \max(0, x) = \begin{cases} 0 & \text{para } x < 0 \\ x & \text{para } x \geq 0 \end{cases}$

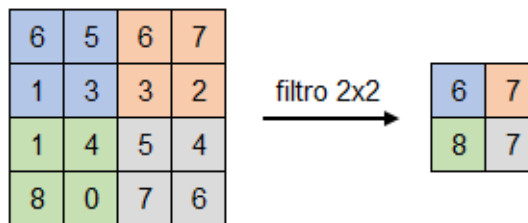
Figura 3. Función de activación *relu*



Fuente: Elaboración propia

La agrupación consiste en reducir la dimensionalidad de la imagen de entrada, por lo general, en este componente se toma el valor máximo de una determinada matriz, ver figura 4.

Figura 4. Agrupación o submuestreo

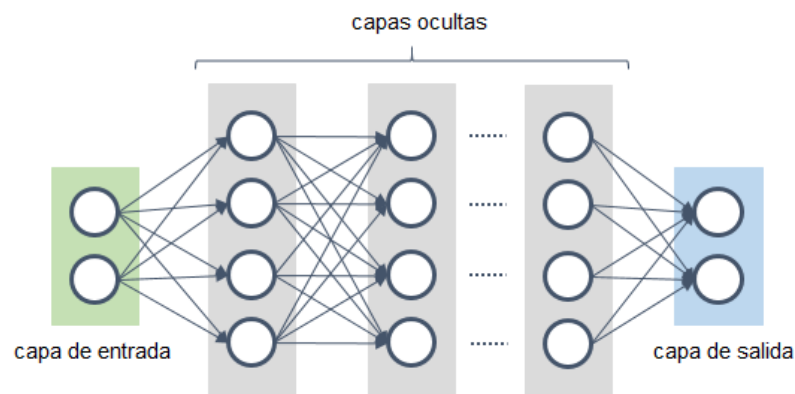


Fuente: Elaboración propia

En la capa totalmente conectada las neuronas de la capa anterior se conectan con la siguiente capa, utilizando la función de activación *softmax* para clasificar el número en la imagen de entrada. Esta función se utiliza cuando existen varias clases y permite obtener una representación de las probabilidades, donde la sumatoria de las salidas es igual a uno. En la figura 5 se muestra una representación de esta capa.

La función *softmax* se representa como $f(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}}$.

Figura 5. Capa totalmente conectada



Fuente: Elaboración propia

1.1.2 Clasificación de imágenes

La clasificación de imágenes consiste en encontrar la categoría a la que pertenece un objeto en una imagen de entre las categorías predefinidas [3], y con los avances de la inteligencia artificial y aprendizaje profundo es una tarea cada vez más fácil.

1.1.3 Herramientas

1.1.3.1 Kaggle Notebooks

Es una plataforma gratuita que permite explorar y ejecutar código de aprendizaje automático y profundo desde el navegador, donde la potencia de procesamiento proviene de servidores en la nube, facilitando un análisis colaborativo [13].

1.1.3.2 Python

Es un lenguaje de programación potente, de alto nivel, orientado a objetos y de propósito general [14]. Posee soporte de diccionario, paquetes de gráficos, es compacto y legible, con la posibilidad de empaquetar múltiples funciones por módulo [8].

1.1.3.3 OpenCV

Es una biblioteca de software de aprendizaje por computadora y visión por computadora de código abierto [7], fácil de usar para aquellas personas con poca experiencia en programación [8]. Esta librería posee alrededor de 2500 algoritmos que pueden ser rápidamente implementados en los lenguajes de programación de *Python*, *Java*, *MATLAB*, *C++*.

1.1.3.4 Haar Cascade

Un humano puede fácilmente detectar objetos, pero una computadora necesita instrucciones y restricciones precisas [9]. Se basa en el aprendizaje automático donde una función en cascada se entrena a partir de varias imágenes positivas y negativas, es decir, imágenes de caras e imágenes sin caras respectivamente [10] [11].

Figura 6. Filtros denominados características de Haar



Fuente: Elaboración propia

La idea consiste en pasar estos filtros (ver figura 6) sobre las imágenes, y en cada ventana se suman todas las intensidades de los píxeles de las partes blanca y negra.

Finalmente, el valor obtenido al restar estas dos sumas es el valor de la característica extraída. Cuanto mayor sea el resultado, de la diferencia entre las sumas de las partes blanca y negra, mayor será la probabilidad de que esa ventana sea una característica relevante [12].

1.1.3.5 *Tensorflow*

Es un marco de código abierto desarrollado para ejecutar el aprendizaje automático, el aprendizaje profundo y para el análisis estadístico y predictivo [15] [16]. *Tensorflow* está basado en diagramas de flujo de datos [17], y se representa como tensores, que son matrices multidimensionales.

1.1.3.6 *Keras*

Es una biblioteca para el aprendizaje profundo de código abierto escrita en *Python*, muy fácil de usar, que se ejecuta sobre *Tensorflow* [18]. *Keras* es un API de alto nivel, que se usa para crear prototipos, que admite redes convolucionales y redes recurrentes, capaz de ejecutarse en CPU y GPU [19].

1.1.3.7 *Django*

Es un *framework* web de código abierto gratuito y de alto nivel, escrito en *Python*, es eficiente y sistemático, se basa en la arquitectura modelo vista template (MVT), pudiéndose usar como cliente y servidor al mismo tiempo, que fomenta el desarrollo rápido y diseño limpio [20] [21].

1.1.3.8 *Laravel*

Es una *framework* web con una sintaxis expresiva y elegante [22], que comprende la arquitectura modelo vista controlador (MVC). *Laravel* permite la agregación modular y dispone de un extenso sistema de paquetes con los que la funcionalidad se amplía de manera fácil, robusta y segura [23].

1.1.3.9 *Axios*

Es una librería de *JavaScript* para efectuar solicitudes HTTP, basada en promesas, que se puede utilizar en una aplicación *frontend*, así como también en el *backend*.

1.2 Solución del Problema

Se crea un servicio en *Django* tanto para la detección de emociones como para el reconocimiento facial, este servicio recibe imágenes en *base64* tanto para almacenarlas como para emitir una respuesta con la mejor coincidencia. La información en *base64* es

enviada desde un proyecto creado en el *framework* de *Laravel*, por medio de la librería de *Axios*. En la figura 7 se observa la arquitectura del sistema implementado.

Figura 7. Arquitectura del sistema



Fuente: Elaboración propia

En la tabla 1, se detallan los materiales utilizados para la detección de emociones y el reconocimiento facial.

Tabla 1. Herramientas para la detección de emociones y reconocimiento facial

Tipos	Herramientas
Frameworks	<i>Django</i> 2.2.4, <i>Laravel</i> 6.9.0, <i>Vue</i> 2.6.11
Aplicaciones	<i>django-cors-headers</i> 3.1.0
Algoritmos	“Frontal Face” Haar Cascade
Lenguaje de programación	<i>Python</i> 3.7, <i>PHP</i> 7.2
Librerías	<i>OpenCV</i> 4.1.2, <i>Numpy</i> 1.18.1, <i>Axios</i> 0.19.0

Fuente: Elaboración propia

Para la detección de emociones, se utilizó el *dataset* de *Kaggle* denominado “Learn facial expressions from an image” (FER2013), ver anexo A, y el entrenamiento de la red convolucional se lo realizó en *Kaggle Notebooks* con las bibliotecas *Keras*, *Tensorflow*, *OpenCV* y el algoritmo pre-entrenado “Frontal Face” Haar Cascade. Con la librería de *Matplotlib* se genera una gráfica donde se evalúa el modelo, identificando los valores de pérdida y precisión tanto de la fase de entrenamiento y validación. La matriz de confusión se generó con la librería *Sklearn*, en la cual se evalúa la precisión de la clasificación, ver tabla 2 para obtener más detalles.

Tabla 2. Herramientas adicionales para la detección de emociones

Materiales	Descripción
Librerías	<i>Keras</i> 2.3.1, <i>Tensorflow</i> 2.1.0, <i>Matplotlib</i> 3.0.3, <i>Sklearn</i> 0.21.3
Dataset	Learn facial expressions from an image FER2013
Plataforma	<i>Kaggle Notebooks</i>

Fuente: Elaboración propia

El *dataset* FER2013 tiene 7 clases: enojado, disgustado, temeroso, feliz, triste, sorprendido y neutral, con un total de 28,709 imágenes, en la tabla 3 se indica la cantidad de imágenes que posee cada categoría.

Tabla 3. Cantidad de imágenes por clase del *dataset* FER2013

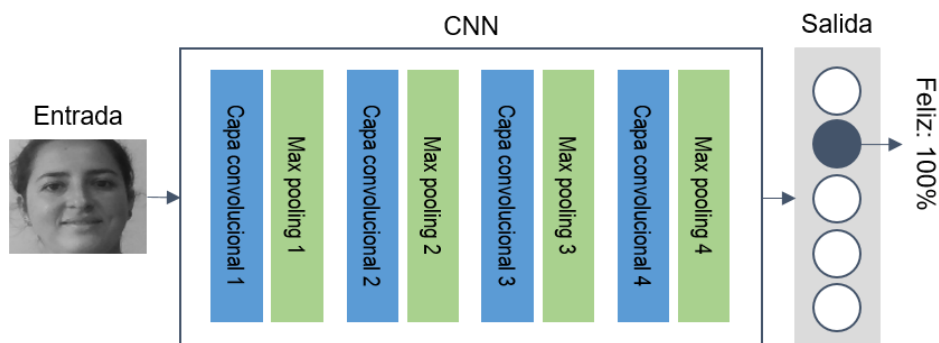
Clases	Enojado	Disgustado	Temeroso	Feliz	Triste	Sorprendido	Neutral
Cantidad	3995	436	4097	7215	4830	3171	4965
Total							28709

Fuente: Elaboración propia

Una vez descargada la base de datos de imágenes, se realizó la selección de las clases e imágenes a utilizar en este trabajo. Las clases que se emplearon son: enojado, feliz, triste, sorprendido y neutral, con un total de 3,000 imágenes por cada clase, resultando 15,000 imágenes para el entrenamiento de la red.

La figura 8 muestra la arquitectura de la red convolucional, la misma que tiene cuatro capas, cada una de las cuales tiene una convolución con una función de activación *relu*, y adicional una capa *maxpooling*. Al final de éstas se agregaron las capas *flatten*, *dropout* y una capa de salida con una función de activación *softmax*, ver el Anexo B para obtener más información acerca de los tipos de capas, formas de salida y parámetros.

Figura 8. Arquitectura de la red neuronal convolucional



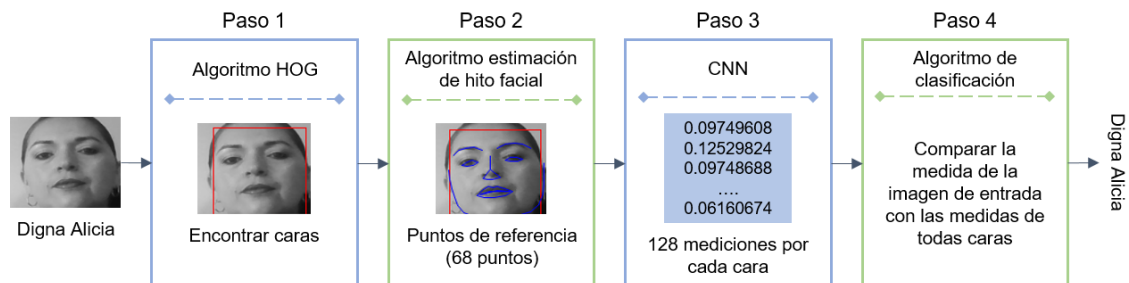
Fuente: Elaboración propia

En las configuraciones para el entrenamiento de la red se tomaron en cuenta 250 épocas, una detención temprana con una paciencia de 25, es decir, el entrenamiento se detiene cuando la precisión obtenida en la validación no aumenta durante 25 épocas, guardando sólo los mejores pesos del entrenamiento.

El reconocimiento facial es el conjunto de varios problemas relacionados, ver figura 9, el primer paso consiste en detectar todas las caras de una imagen aplicando el algoritmo de histogramas de gradientes orientados (HOG), el siguiente paso determina si una cara

con una dirección extraña sigue perteneciendo a la misma persona, para lo que se emplea el algoritmo de estimación de hito facial permitiendo identificar 68 puntos de referencia por cada cara.

Figura 9. Pasos para el reconocimiento facial



Fuente: Elaboración propia

En el tercer paso se crean 128 mediciones por cada rostro mediante aprendizaje profundo, que corresponden a características únicas de las caras, finalmente con las características únicas de cada cara se determina el nombre de las personas.

Se emplea la librería *face recognition* para el reconocimiento facial, además se registra información de personas en la base de datos MySQL, ver Anexo D, como el nombre, apellido, cédula, siendo este último utilizado como el nombre del directorio que contiene las imágenes correspondientes a cada persona. En la tabla 4 se detallan las demás herramientas utilizadas para el reconocimiento facial.

Tabla 4. Herramientas adicionales para el reconocimiento facial

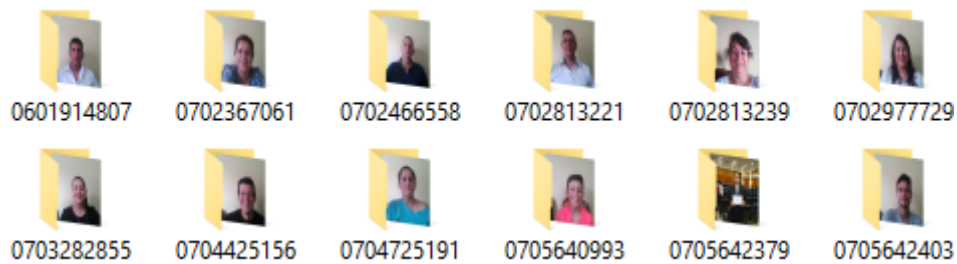
Materiales	Descripción
Servidor de base de datos	MySQL 10.4.8
Servidor web	Apache 2.4.41
Lenguaje del servidor	PHP 7.3.11
Librería	Face Recognition 1.2.3

Fuente: Elaboración propia

Previo a la instalación del paquete *Face Recognition* se instalaron componentes de *Visual Studio*, la librería de *cmake*, mientras que la biblioteca *dlib* se instala con *Face Recognition*, ver Anexo E para obtener más detalles.

Una vez registrada la información se procede a tomar fotografías por cada persona, como se aprecia en el Anexo F. En total se registraron a 20 personas. Las imágenes de cada persona se estructuran en una carpeta, la cual contiene como nombre el número de cédula de ciudadanía, como se observa en la figura 10.

Figura 10. Estructura de fotografías por persona

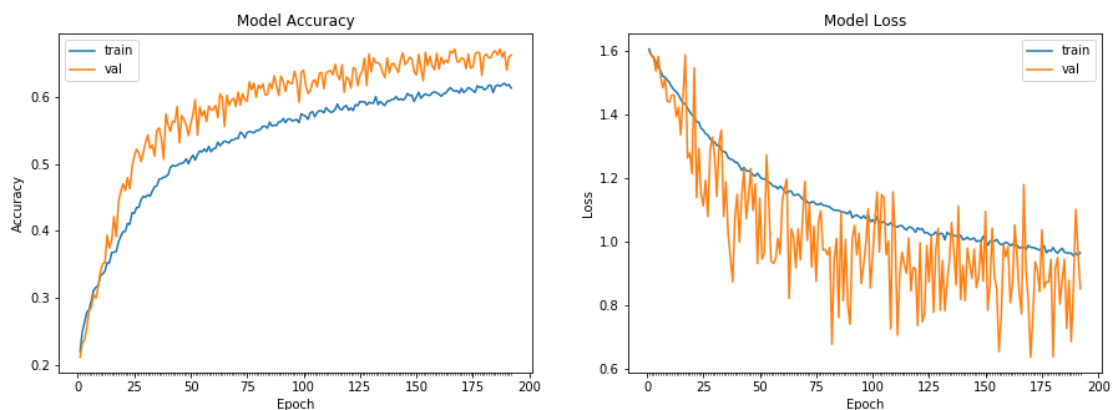


Fuente: Elaboración propia

1.3 Resultados

El entrenamiento de la red neuronal convolucional, haciendo referencia a la detección de emociones, terminó en la época 167, debido a la configuración de detención temprana que se realizó, alcanzando hasta esa época una precisión del 67,13%. En la figura 11 se observan los resultados del proceso de entrenamiento y validación.

Figura 11. Resultados del entrenamiento



Fuente: Elaboración propia

Los archivos de los pesos y el modelo generados al finalizar el entrenamiento de la red neuronal dentro la plataforma *Kaggle Notebooks* se descargan y agregan al proyecto de *Django*, para hacer la respectiva clasificación. Dentro del proyecto de *Laravel*, específicamente en la sección de expresiones, se pueden tomar fotografías con la cámara o se puede cargar una imagen para enviarla mediante *Axios* a *Django*, en donde se procesa para emitir una respuesta acerca de la emoción detectada.

La matriz de confusión, ver tabla 5, generada durante el proceso de validación permite identificar que la clase denominada feliz tiene una mayor cantidad de aciertos, siendo esta emoción la que más se diferencia de entre las otras categorías. Luego se determina que la siguiente clase con más aciertos es la denominada neutral.

Tabla 5. Matriz de confusión

	Enojado	Feliz	Neutral	Triste	Sorprendido
Enojado	182	247	219	148	162
Feliz	343	445	438	231	317
Neutral	210	309	312	168	234
Triste	243	306	297	168	233
Sorprendido	163	207	207	95	159

Fuente: Elaboración propia

En el Anexo C se presentan los resultados obtenidos en la detección de emociones corresponden a los porcentajes de coincidencia por cada clase, los cuales se muestran en gráficos estadísticos. Al realizar las pruebas se identifica que existe mayor precisión en la clasificación de las imágenes de acuerdo a la categoría de felicidad.

El inicio de sesión del aplicativo se puede realizar de manera tradicional ingresando las credenciales o mediante el reconocimiento del rostro, ver figura 12. Los resultados obtenidos, concerniente al reconocimiento facial, tienen una gran exactitud, y no es de extrañar, puesto que la librería utilizada posee una precisión del 99.38%.

Figura 12. Inicio de sesión mediante reconocimiento facial



Fuente: Elaboración propia

Para el test del reconocimiento facial, se consideraron a veinte personas, obteniendo una identificación correcta para cada una de estas, por otro lado, al realizarse pruebas con personas no registradas en algunos casos se identifican como desconocidas y en otros se determina la mejor coincidencia de acuerdo a sus facciones. En el Anexo G se adjuntan los resultados de este test.

2. CONCLUSIONES

En el entrenamiento de la red neuronal se ha alcanzado una precisión de validación del 67.13%, un porcentaje muy bueno si lo comparamos con la tabla de clasificación presentado en la plataforma de *Kaggle*, puesto que el mejor modelo obtuvo una precisión del 71.16%. Clasificar una imagen de una cara en emociones es una tarea difícil incluso para el ser humano, y esto se ve reflejado en el *dataset* FER2013, pues una misma imagen podría tener más de una categoría, lo que notablemente afecta la exactitud de una red neuronal.

En el reconocimiento facial se utilizó la librería *Face Recognition* y para evaluar este algoritmo se realizaron pruebas con veinte personas, a las mismas que se identificó de manera correcta. Es importante mencionar que al efectuar pruebas con personas no registradas se obtiene la mejor coincidencia de acuerdo al *dataset* registrado o en su defecto se identifican como desconocidas.

Se creó un servicio en *Django*, en el cual se empleó la librería *Face Recognition* para realizar el reconocimiento facial, mientras que para la detección de emociones se emplean dos archivos generados durante el entrenamiento de la red neuronal, el uno corresponde al modelo y el otro a los pesos, con los que se hace la predicción.

Se desarrolló una aplicación en *Laravel* donde los usuarios pueden interactuar y probar el funcionamiento de la detección de emociones y reconocimiento facial. Para registrar los usuarios se hace uso del *framework* *Vue*, y para la comunicación entre *Laravel* y *Django* se utiliza *Axios*. Adicional se logró realizar el inicio de sesión mediante el reconocimiento facial.

BIBLIOGRAFÍA

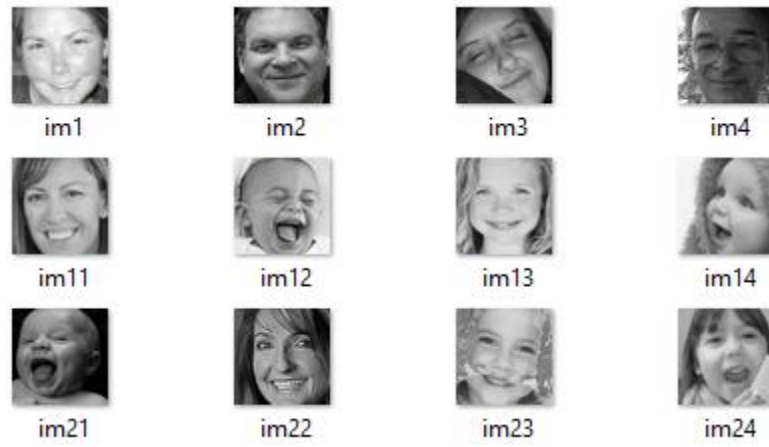
- [1] R. K. Pandey, S. Karmakar, A. G. Ramakrishnan, y N. Saha, «Improving Facial Emotion Recognition Systems Using Gradient and Laplacian Images», pp. 1-5, 2019.
- [2] S. Almabdy y L. Elrefaei, «Deep convolutional neural network-based approaches for face recognition», *Appl. Sci.*, vol. 9, n.º 20, 2019, doi: 10.3390/app9204397.
- [3] H. Fujiyoshi, T. Hirakawa, y T. Yamashita, «Deep learning-based image recognition for autonomous driving», *IATSS Res.*, 2019, doi: 10.1016/j.iatssr.2019.11.008.
- [4] N. Samadiani *et al.*, «A review on automatic facial expression recognition systems assisted by multimodal sensor data», *Sensors (Switzerland)*, vol. 19, n.º 8, pp. 1-27, 2019, doi: 10.3390/s19081863.
- [5] K. Munir, H. Elahi, A. Ayub, F. Frezza, y A. Rizzi, «Cancer diagnosis using deep learning: A bibliographic review», *Cancers (Basel)*, vol. 11, n.º 9, pp. 1-36, 2019, doi: 10.3390/cancers11091235.
- [6] K. Yasaka, H. Akai, A. Kunimatsu, S. Kiryu, y O. Abe, «Deep learning with convolutional neural network in radiology», *Jpn. J. Radiol.*, vol. 36, n.º 4, pp. 257-272, 2018, doi: 10.1007/s11604-018-0726-3.
- [7] «Acerca de». [En línea]. Disponible en: <https://opencv.org/about/>. [Accedido: 12-ene-2020].
- [8] B. Gupta, A. Chaube, A. Negi, y U. Goel, «Study on Object Detection using Open CV - Python», *Int. J. Comput. Appl.*, vol. 162, n.º 8, pp. 17-21, 2017, doi: 10.5120/ijca2017913391.
- [9] P. Pasumarti y P. P. Sekhar, «Classroom Attendance Using Face Detection and Raspberry-Pi», pp. 167-171, 2018.
- [10] E. R. Davies, «Face detection and recognition», *Comput. Vis.*, n.º December, pp. 631-662, 2018, doi: 10.1016/b978-0-12-809284-2.00021-6.
- [11] J. S. N. Valenzuela, «Python Aplicaciones prácticas». .
- [12] «Reconocimiento facial con OpenCV: Haar Cascade - DataSeries - Medium». [En línea]. Disponible en: <https://medium.com/dataseries/face-recognition-with-opencv-haar-cascade-a289b6ff042a>. [Accedido: 12-ene-2020].

- [13] «Kaggle: Your Home for Data Science». [En línea]. Disponible en: <https://www.kaggle.com/>. [Accedido: 17-ene-2020].
- [14] O. Rahmati *et al.*, «An automated Python language-based tool for creating absence samples in groundwater potential mapping», *Remote Sens.*, vol. 11, n.º 11, pp. 1-22, 2019, doi: 10.3390/rs11111375.
- [15] V. Leon, S. Mouselinos, K. Koliogeorgi, y S. Xydis, «A TensorFlow Extension Framework for Optimized Generation of Hardware CNN Inference Engines», 2020, doi: 10.3390/technologies8010006.
- [16] T. Caramihale, D. Popescu, y L. Ichim, «Emotion classification using a tensorflow generative adversarial network implementation», *Symmetry (Basel)*., vol. 10, n.º 9, 2018, doi: 10.3390/sym10090414.
- [17] L. Yang, Y. Li, J. Wang, y Z. Tang, «Post text processing of chinese speech recognition based on bidirectional LSTM networks and CRF», *Electron.*, vol. 8, n.º 11, p. 1249, 2019, doi: 10.3390/electronics8111248.
- [18] J. Villalba-Diez, D. Schmidt, R. Gevers, J. Ordieres-Meré, M. Buchwitz, y W. Wellbrock, «Deep learning for industrial computer vision quality control in the printing industry 4.0», *Sensors (Switzerland)*, vol. 19, n.º 18, pp. 1-23, 2019, doi: 10.3390/s19183987.
- [19] «Inicio - Documentación de Keras». [En línea]. Disponible en: <https://keras.io/>. [Accedido: 16-ene-2020].
- [20] «El marco web para perfeccionistas con plazos | Django». [En línea]. Disponible en: <https://www.djangoproject.com/>. [Accedido: 16-ene-2020].
- [21] H. J. Song *et al.*, «A web service for evaluating the level of speech in Korean», *Appl. Sci.*, vol. 9, n.º 3, 2019, doi: 10.3390/app9030594.
- [22] «Laravel - The PHP Framework For Web Artisans». [En línea]. Disponible en: <https://laravel.com/>. [Accedido: 16-ene-2020].
- [23] V. Villarreal, M. Nielsen, y M. Samudio, «Sensing and storing the blood pressure measure by patients through a platform and mobile devices», *Sensors (Switzerland)*, vol. 18, n.º 6, 2018, doi: 10.3390/s18061805.

ANEXO A

Dataset de Kaggle denominado "Learn facial expressions from an image"

Figura 13. *Dataset de Kaggle* FER2013



Fuente: Elaboración propia

ANEXO B

Información de la red neuronal convolucional

Figura 14. Representación resumida del modelo

```
Model: "sequential_2"
-----
```

Layer (type)	Output Shape	Param #
conv2d_5 (Conv2D)	(None, 46, 46, 32)	320
max_pooling2d_5 (MaxPooling2D)	(None, 23, 23, 32)	0
conv2d_6 (Conv2D)	(None, 21, 21, 64)	18496
max_pooling2d_6 (MaxPooling2D)	(None, 10, 10, 64)	0
conv2d_7 (Conv2D)	(None, 8, 8, 128)	73856
max_pooling2d_7 (MaxPooling2D)	(None, 4, 4, 128)	0
conv2d_8 (Conv2D)	(None, 2, 2, 128)	147584
max_pooling2d_8 (MaxPooling2D)	(None, 1, 1, 128)	0
flatten_2 (Flatten)	(None, 128)	0
dropout_2 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 512)	66048
dense_4 (Dense)	(None, 5)	2565

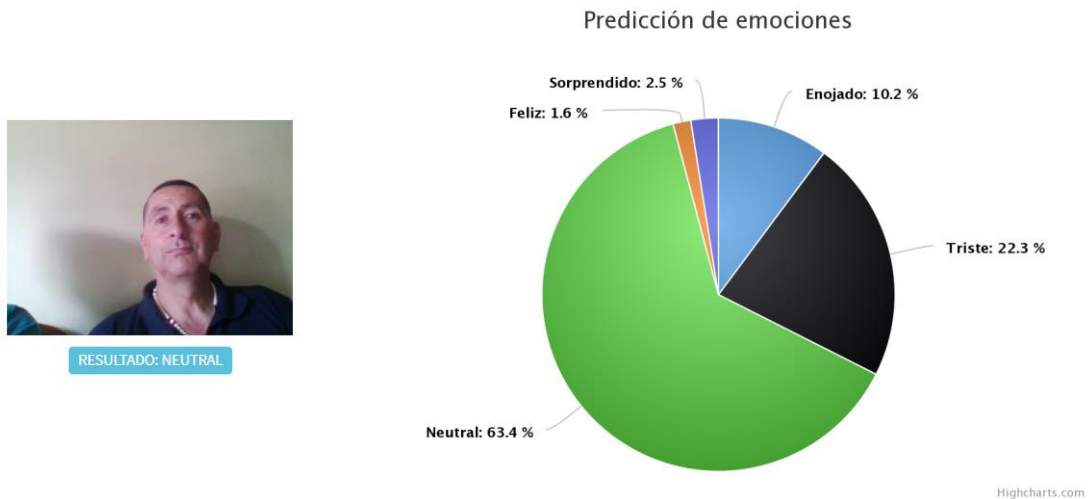
```
-----
Total params: 308,869
Trainable params: 308,869
Non-trainable params: 0
```

Fuente: Elaboración propia

ANEXO C

Detección de emociones

Figura 15. Expresión neutral



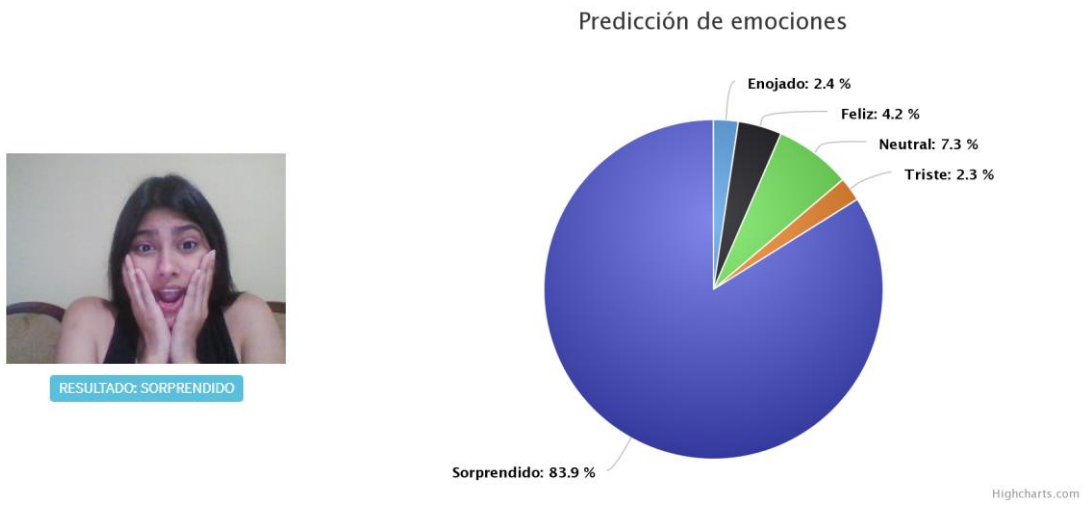
Fuente: Elaboración propia

Figura 16. Expresión de una persona feliz



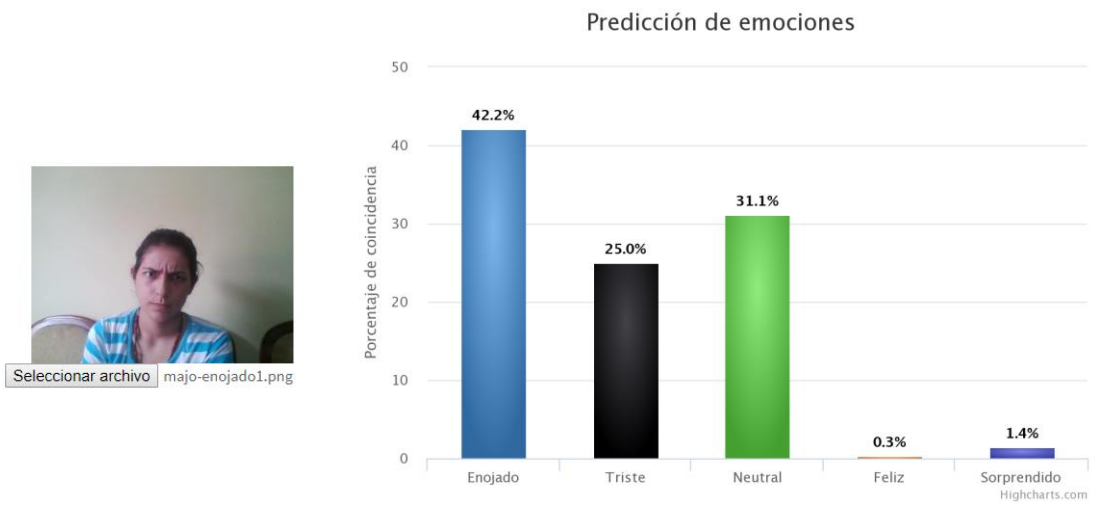
Fuente: Elaboración propia

Figura 17. Expresión de una persona sorprendida



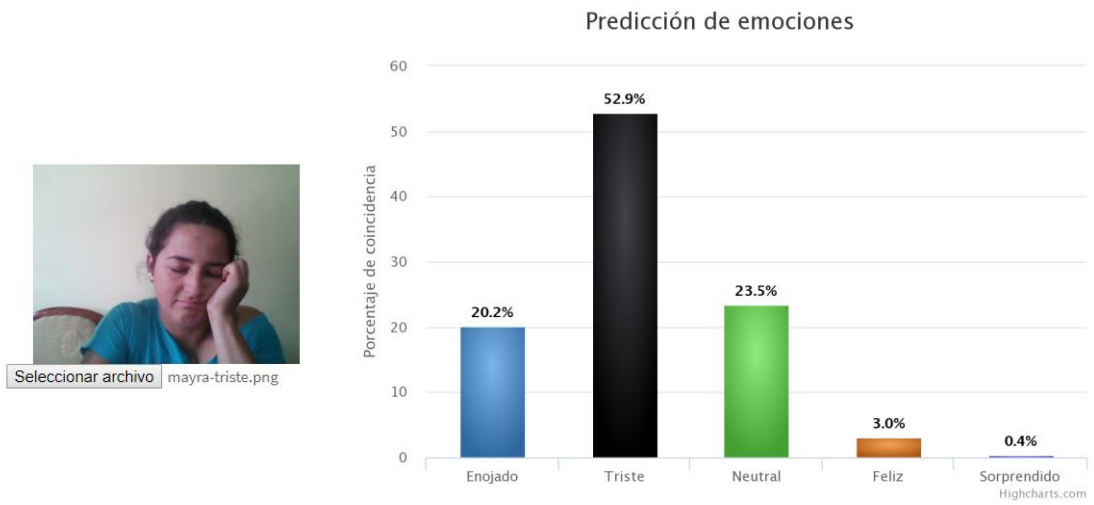
Fuente: Elaboración propia

Figura 18. Expresión de una persona enojada



Fuente: Elaboración propia

Figura 19. Expresión de una persona triste



Fuente: Elaboración propia

ANEXO D

Registro, edición y listado de personas

Figura 20. Lista de personas registradas

UTMACH

Personas

Reconocimiento facial

Expresiones

Lista de personas

Nuevo Entrenar

Mostrar 8 entradas

Buscar...

#	Nombre	Apellido	DNI	Email	Opciones
1	Ezequiel	Jaramillo	0705642379	sejr1996@gmail.com	EDITAR TOMAR FOTOS
2	María José	Jaramillo Romero	0705642411	majo@gmail.com	EDITAR TOMAR FOTOS
3	José Ufredo	Jaramillo Romero	0702466558	jose@gmail.com	EDITAR TOMAR FOTOS
4	Jose Ricardo	Jaramillo Romero	0705642403	ricardo@hotmail.com	EDITAR TOMAR FOTOS
5	Mayra Claribel	Toro Ruíz	0704725191	mayratoro@gmail.com	EDITAR TOMAR FOTOS
6	María del Carmen	Romero Freire	0702813239	mariadelcarmen@gmail.com	EDITAR TOMAR FOTOS
7	Yeni Maria	Romero Freire	0701786980	yeni@gmail.com	EDITAR TOMAR FOTOS
8	Anderson Eduardo	Freire Toro	0705640993	andersonfreire@gmail.com	EDITAR TOMAR FOTOS

Anterior 1 2 3 Siguiete

Fuente: Elaboración propia

Figura 21. Registro de una nueva persona

UTMACH

Personas

Reconocimiento facial

Expresiones

Nueva persona

Nombre: Ingrese el nombre

Apellido: Ingrese el apellido

Email: Ingrese el email

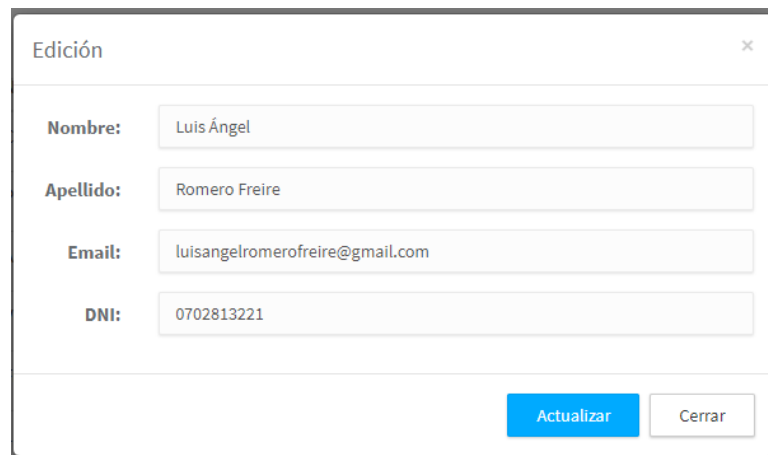
DNI: Ingrese el dni

Guardar Cancelar

© 2020 | UTMACH

Fuente: Elaboración propia

Figura 22. Edición de la información de una determinada persona



Edición ×

Nombre:

Apellido:

Email:

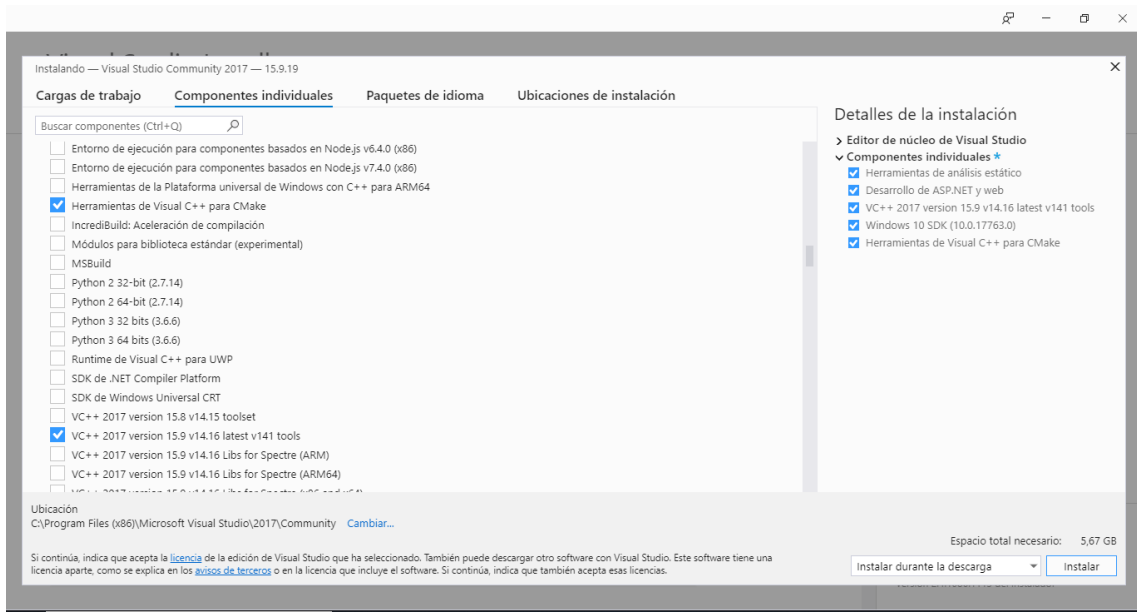
DNI:

Fuente: Elaboración propia

ANEXO E

Componentes requeridos por la librería *face recognition*

Figura 23. Componentes de *visual studio*



Fuente: Elaboración propia

Figura 24. Instalación de *cmake* y *face recognition*

```
Administrador: Símbolo del sistema
Microsoft Windows [Versión 10.0.18362.592]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\WINDOWS\system32>pip install cmake
Requirement already satisfied: cmake in c:\users\ziqui\appdata\local\programs\python\python37\lib\site-packages (3.13.3)

C:\WINDOWS\system32>pip install face_recognition
Collecting face_recognition
  Using cached face_recognition-1.2.3-py2.py3-none-any.whl (21 kB)
Collecting dlib==19.7
  Using cached dlib-19.19.0.tar.gz (3.2 MB)
Requirement already satisfied: Click>=6.0 in c:\users\ziqui\appdata\local\programs\python\python37\lib\site-packages (from face_recognition)
Requirement already satisfied: Pillow in c:\users\ziqui\appdata\local\programs\python\python37\lib\site-packages (from face_recognition)
Requirement already satisfied: face-recognition-models>=0.3.0 in c:\users\ziqui\appdata\local\programs\python\python37\lib\site-packages (from face_recognition)
Requirement already satisfied: numpy in c:\users\ziqui\appdata\local\programs\python\python37\lib\site-packages (from face_recognition)
Installing collected packages: dlib, face-recognition
  Running setup.py install for dlib ... done
Successfully installed dlib-19.19.0 face-recognition-1.2.3

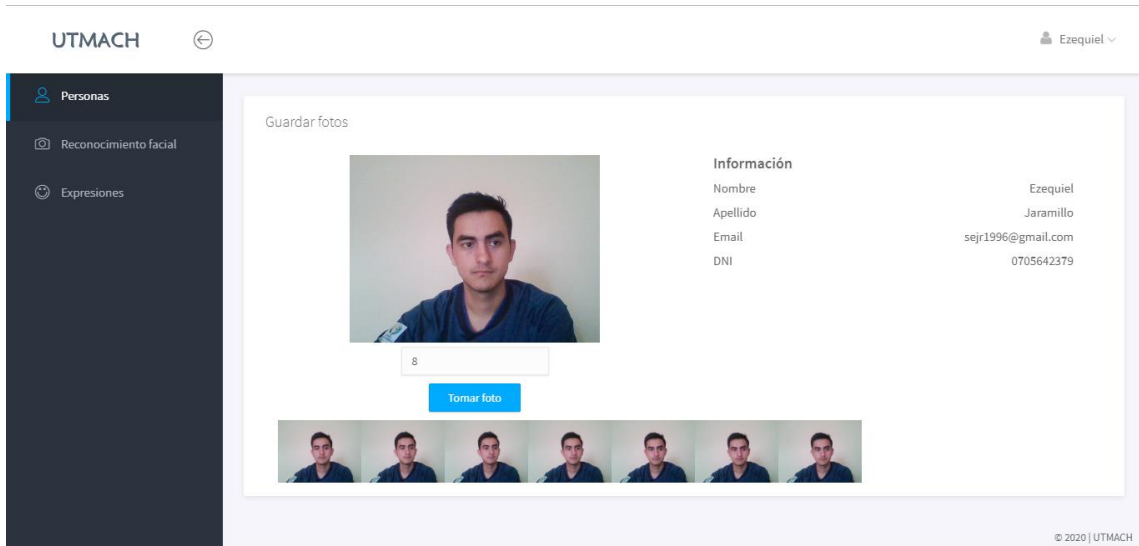
C:\WINDOWS\system32>
C:\WINDOWS\system32>
```

Fuente: Elaboración propia

ANEXO F

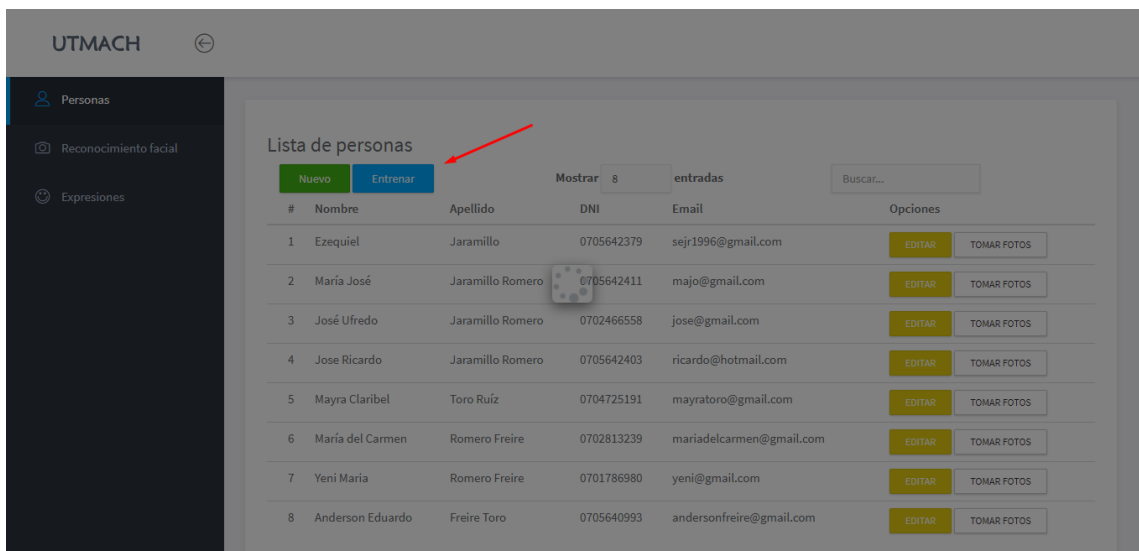
Almacenamiento y entrenamiento de las imágenes

Figura 25. Toma de fotografías por persona



Fuente: Elaboración propia

Figura 26. Generar un archivo con características únicas por cada cara



Fuente: Elaboración propia

ANEXO G

Pruebas de reconocimiento facial

Figura 27. Test de reconocimiento facial con una persona

UTMACH ⊖ ANÁLISIS FINALIZADO

Personas
Reconocimiento facial
Expresiones

Reconocimiento facial

Iniciar reconocimiento

DNI	Nombre	Apellido	Email
0705643112	Claudia	Romero Gomez	claudiag@gmail.com

Fuente: Elaboración propia

Figura 28. Test de reconocimiento facial con dos personas

UTMACH ⊖ ANÁLISIS FINALIZADO

Personas
Reconocimiento facial
Expresiones

Reconocimiento facial

Iniciar reconocimiento

DNI	Nombre	Apellido	Email
0705642403	Jose Ricardo	Jaramillo Romero	ricardo@hotmail.com
0705642411	María José	Jaramillo Romero	majo@gmail.com

Fuente: Elaboración propia

Figura 29. Test de reconocimiento facial con tres personas

UTMACH

Personas

Reconocimiento facial

Expresiones

Reconocimiento facial

Iniciar reconocimiento

DNI	Nombre	Apellido	Email
0702466558	José Ufredo	Jaramillo Romero	jose@gmail.com
0705642379	Ezequiel	Jaramillo	sejr1996@gmail.com
0702813239	María del Carmen	Romero Freire	mariadelcarmen@gmail.com

Fuente: Elaboración propia

Figura 30. Test de reconocimiento facial con personas no registradas

UTMACH

Personas

Reconocimiento facial

Expresiones

Reconocimiento facial

Iniciar reconocimiento

DNI	Nombre	Apellido	Email
0751606922	Ivana	Toledo Gomez	ivanatg@gmail.com
0702367061	Lupe	Delgado Romero	lupe@gmila.com
0702977729	Eugenia del Carmen	Jaramillo Romero	eugenia@gmail.com

Fuente: Elaboración propia