



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Máster en Ingeniería Informática



TFM del Máster en Ingeniería
Informática

Desarrollo de una Aplicación Web
para la Detección de Intrusos
empleando Técnicas de
Inteligencia Artificial



Presentado por Juan Manuel López Santana
en la Universidad de Burgos – 13 de junio de
2022

Tutor: José María Cámara Nebreda



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Máster en Ingeniería Informática



D. José María Cámara Nebreda, profesor del departamento de Ingeniería Electromecánica, área de Tecnología Electrónica.

Expone:

Que el alumno D. Juan Manuel López Santana, con DNI G34949879, ha realizado el Trabajo Final de Máster en Ingeniería Informática titulado: Desarrollo de una Aplicación Web para la Detección de Intrusos empleando Técnicas de Inteligencia Artificial.

Y que dicho trabajo ha sido realizado por el alumno bajo la dirección del que suscribe, en virtud de lo cual se autoriza su presentación y defensa.

En Burgos, 13 de junio de 2022

Vº. Bº. del Tutor:

D. José María Cámara Nebreda

Resumen

El mundo se encuentra conectado a Internet con mayor ahínco cada vez más. Hoy en día es muy común encontrar una gran variedad de dispositivos conectados a través de alguna red. No únicamente teléfonos inteligentes o computadores, sino cámaras de vigilancia, aparatos electrodomésticos y relojes, por mencionar algunos. Ahora, si bien el vivir en un mundo interconectado brinda diversas ventajas, es necesario observar la contraparte de esta situación. Y es que mayor interconectividad significa también una mayor posibilidad de que exista un robo de información.

Es aquí donde la Inteligencia Artificial (I.A.) entra en la ecuación, para auxiliar en la implementación de soluciones que erradiquen o mitiguen los incidentes que pongan en peligro la integridad de la información. El artículo científico titulado SSD: Single Shot MultiBox Detector describe una Red Neuronal Convolutiva (R.N.C.), que luego de completar todas las fases de su entrenamiento, permite identificar rostros humanos, al igual que otros objetos.

Finalizado el estudio de dicha RNC, se ha procedido a probarla en una aplicación web desarrollada en React que tiene como objetivo identificar intrusos. El sistema detecta los rasgos y las expresiones faciales de una persona que se encuentra frente a la cámara web del ordenador, así como su edad y género, para concluir si efectivamente se trata de un intruso o no. En un futuro, no se descarta la posibilidad de llevar esta implementación al campo de telefonía móvil o al de sistemas empotrados o ubicuos.

Descriptores

Red Neuronal Convolutiva, Reconocimiento de Rostro, Detección de Intrusos, Inteligencia Artificial, Desarrollo Web, React.

Abstract

The world is increasingly connected to the Internet. Nowadays it is very common to find a variety of devices connected through a network. Not just smartphones or computers, but surveillance cameras, home appliances and watches, to name a few. Now, while living in an interconnected world offers several advantages, it is necessary to observe the counterpart of this situation. Greater interconnectivity also means a greater possibility of information theft.

It is here that Artificial Intelligence (AI) enters the equation, to assist in the implementation of solutions that eradicate or mitigate incidents that endanger the integrity of information. The scientific paper entitled SSD: Single Shot MultiBox Detector, describes a Convolutional Neural Network (CNN), which after completing all phases of its training, allows to identify human faces, as well as other objects.

After the study of said CNN, it has proceeded to test it in a web application developed in React that aims to identify intruders. The system detects the facial features and expressions of a person who is in front of the computer webcam, as well as their age and gender, to conclude whether it is an intruder or not. In the future, the possibility of taking this implementation to the field of mobile telephony or embedded or ubiquitous systems is not ruled out.

Keywords

Convolutional Neural Network, Face Recognition, Intrusion Detection, Artificial Intelligence, Web Development, React.

Índice general

Apartado 1	10
1.1. Contexto	10
1.2. Motivación	11
1.3. Estructura del documento	11
Apartado 2	14
2.1. Objetivo general	14
2.2. Objetivos específicos	14
Apartado 3	16
3.1. Fundamentos básicos	16
3.2. Análisis del artículo científico SSD: Single Shot MultiBox Detector	20
Apartado 4	29
4.1. Descripción de técnicas y herramientas utilizadas	29
Apartado 5	32
5.1. Funcionamiento de la red neuronal convolucional	32
5.2. Desarrollo de la aplicación web en React	33
5.3. Comunicación entre la aplicación web y la red neuronal	33
Apartado 6	34
6.1. Face Recognition using Convolutional Neural Network and Simple Logistic Classifier	34
6.2. Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks	36
6.3. Face Recognition Based on Lightweight Convolutional Neural Networks	38
Apartado 7	40
7.1. Cumplimiento de objetivos	40
7.2. Competencias demostradas	41
7.3. Trabajos futuros	42

Apéndice A.....	43
A.1. Introducción.....	43
A.2. Planificación temporal	43
A.3. Viabilidad económica.....	47
A.4. Viabilidad legal.....	48
Apéndice B.....	50
B.1. Introducción.....	50
B.2. Objetivos generales	50
B.3. Especificación de requisitos.....	50
Apéndice C.....	52
C.1. Casos de uso.....	52
C.2. Diagrama de secuencia.....	55
C.3. Diagrama de despliegue.....	56
C.4. Diseño de interfaz.....	57
Apéndice D.....	60
D.1. Introducción.....	60
D.2. Estructura de directorios del proyecto.....	60
D.3. Manual de programador	61
D.4. Complicación, instalación y ejecución del proyecto	62
D.5. Pruebas del sistema	63
Apéndice E.....	70
E.1. Introducción	70
E.2. Requisitos de usuarios	70
E.3. Instalación	70
E.4. Manual de usuario.....	70
Referencias	74

Índice de figuras

Figura 3.1: Arquitectura de la red neuronal convolucional utilizada en el artículo científico analizado [1]	20
Figura 3.2: Convolución separable en profundidad [6]	21
Figura 3.3: Comparación de una convolución estándar y una convolución separable en profundidad [7].....	23
Figura 3.4: Representación gráfica del ejemplo en cuestión [8].....	24
Figura 3.5: Algoritmo SSD empleado en el modelo de la red neuronal [1]	25
Figura 3.6: Detección de objetos mediante el algoritmo SSD [1].....	25
Figura 3.7: Representación de la detección de los mapas de características de la imagen [1]	26
Figura 3.8: Cálculo de la distancia entre los píxeles originales de la imagen y los que se han detectado mediante el algoritmo [1].....	26
Figura 3.9: Resultados de la evaluación de la red neuronal [1]	27
Figura 3.10: Nivel de precisión arrojado por la red neuronal [1].....	27

Apartado 1

Introducción

A lo largo de este apartado se describe la motivación que se siguió para elegir y desarrollar este Trabajo Fin de Máster (T.F.M). De igual manera, se presenta un contexto acerca del avance que ha tenido la tecnología en los últimos años y cómo esta se encuentra presente en el día a día del ser humano.

1.1. Contexto

Es increíble la cantidad de avances tecnológicos que han surgido en las últimas décadas. Años atrás, hubiese sido imposible imaginar que eventualmente, un ser humano sería capaz de tener prácticamente todo su mundo en la palma de su mano.

En el Grado, tuve la oportunidad de adentrarme en forma al mundo de las Tecnologías de la Información y Comunicación (T.I.C) y ya desde aquel tiempo temas como la ciberseguridad, I.A., Internet de las cosas y sistemas empotrados eran relevantes. El haber tomado este Máster no hizo más que confirmarme que dichos conceptos son una completa realidad. El que compra en línea, por ejemplo, lo hace en diversas ocasiones gracias a las recomendaciones de productos que le aparecen en el sitio web. En la mayoría de casos, estas recomendaciones le aparecen al usuario gracias al empleo de un algoritmo de aprendizaje automático.

Los empleados de una compañía acceden a sus instalaciones a través de un sistema de reconocimiento, ya sea colocando su huella dactilar en un sensor o posicionando su pupila en otro. Los entornos donde el ser

humano vive se encuentran cada vez más repletos de sistemas, redes y telecomunicaciones, que le proporcionan la capacidad de realizar sus actividades profesionales, familiares y personales de maneras mucho más sencillas. Por otro lado, un mundo interconectado, representa contar con cantidades enormes de información que se desplazan de un lado hacia otro, al alcance no solamente de aquel a quien le pertenece, sino también de usuarios malintencionadas.

Contar pues, con mecanismos de protección de información que coloquen una barrera contra los usuarios no autorizados es de suma importancia, sin importar el ámbito y área de la cual se trate.

1.2. Motivación

Bajo la premisa, y más que nada, realidad, de vivir en un entorno interconectado y en constante cambio debido a los distintos avances tecnológicos, y aunado al hecho que existe en la importancia de proteger la integridad de la información (y más, la de nosotros mismos), me resulta de sumo interés llevar a la práctica los conocimientos adquiridos a lo largo de la realización de este Máster para desarrollar una aplicación web encargada de detectar intrusos, mediante el empleo de técnicas de Inteligencia Artificial.

A saber, se ha estudiado, probado y empleado la red neuronal convolucional que se describe en el artículo SSD: Single Shot MultiBox Detector [1] para crear la aplicación web que gracias al empleo de dicha red neuronal, detecta los rasgos y expresiones faciales de un usuario, indicando si este es o no intruso.

1.3. Estructura del documento

Enseguida se describe la estructura que tiene el presente documento, con una explicación clara y breve de cada uno de sus apartados.

Objetivos del proyecto

Apartado donde se detalla el objetivo principal del desarrollo del proyecto, así como los objetivos parciales del mismo.

Conceptos teóricos

Sección en la cual se introducen los fundamentos teóricos necesarios para la comprensión de este documento. Al mismo tiempo, se lleva a cabo un análisis del artículo científico estudiado para la realización del proyecto.

Técnicas y herramientas

Dentro de este apartado, se describen las diferentes herramientas empleadas para el desarrollar el sistema del proyecto. Además, se explica el tipo de metodología implementada en el proceso de desarrollo del mismo.

Aspectos relevantes del desarrollo del proyecto

En este apartado se redactan los aspectos más relevantes del proceso de desarrollo del sistema; se analiza el proceso de creación de la aplicación web, así como el de la comunicación entre esta y la red neuronal convolucional.

Trabajos relacionados

Dentro de esta sección se presentan algunos de los trabajos que se consideran destacados relacionados al uso de redes neuronales para la detección de rostros humanos.

Conclusiones y Líneas de trabajo futuras

Apartado donde se estudian los objetivos que se alcanzaron a lo largo del desarrollo del proyecto. Por otra parte, se presentan distintas directrices a las cuales se puede dirigir el trabajo planteado en este proyecto en un futuro.

Anexo A: Plan de Proyecto Software

Anexo que detalla la planificación de trabajo seguida para el desarrollo del Trabajo Fin de Máster, al igual que la viabilidad económica y legal del mismo.

Anexo B: Especificación de Requisitos

En este anexo se encuentran los requerimientos funcionales y no funcionales de la aplicación web desarrollada.

Anexo C: Especificación de Diseño

Sección donde se muestra el diseño del sistema, detallando los bocetos del desarrollo del mismo.

Anexo D: Documentación técnica de programación

Anexo que detalla la estructura de los directorios de la aplicación web. A su vez, contiene un manual técnico de usuario.

Anexo E: Documentación de usuario

Apartado donde se describe una serie de instrucciones para que un usuario sea capaz de ejecutar la aplicación web desarrollada.

Apartado 2

Objetivos del proyecto

A continuación se detallan el objetivo principal y los objetivos específicos que persigue la elaboración del proyecto.

2.1. Objetivo general

Analizar, estudiar y comprender el comportamiento y funcionamiento de la Red Neuronal Convolucional descrita en el artículo SSD: Single Shot MultiBox Detector, para poder emplearla en el desarrollo de una aplicación web encargada de detectar intrusos.

2.2. Objetivos específicos

Entre los objetivos específicos que tiene la elaboración de este T.F.M. se encuentran:

- Desarrollar un sistema capaz de detectar en tiempo real la presencia del rostro de una persona por medio del empleo de una cámara web, indicando el grado de similitud entre el rostro de la persona y un rostro humano real.
- Dicha aplicación debe de ser capaz de identificar una expresión facial en el rostro del usuario.

- La aplicación mostrará el grado de exactitud que presenta la expresión facial de la persona en relación a expresiones faciales determinadas, tales como felicidad, asombro y enojo.
- La aplicación web reconocerá la edad y el género de la persona ubicada frente a la cámara web.
- Con ello se pretende que el usuario que utilice la aplicación pueda observar de primera mano a la persona que se coloca frente a la cámara, sus expresiones y rasgos faciales, así como su edad y género, con el objetivo de que el sistema lo auxilie en la toma de la decisión respecto a si se trata o no de un intruso.

Apartado 3

Conceptos Teóricos

3.1. Fundamentos básicos

En el presente apartado, se presentan todos aquellos fundamentos teóricos básicos que son necesarios para entender el planteamiento, desarrollo y solución del proyecto elaborado.

Peso

Valor asociado a cada dato de entrada en una red neuronal que señala su relevancia o importancia en la generación de los datos de salida finales.

Neurona

También conocida como nodo, representa una función matemática que calcula el promedio de los pesos de múltiples datos de entrada en una red neuronal, y genera como salida un determinado valor.

Red Neuronal Artificial (R.N.A)

Conjunto de algoritmos que se encargan de buscar la existencia de relaciones dentro de un conjunto de datos. Su funcionamiento es una similitud al cerebro humano. Es un tipo de aprendizaje de máquinas del tipo aprendizaje profundo, que emplea neuronas que están interconectadas entre sí en una estructura formada por capas. [2]

De esta manera las computadoras pueden estar aprendiendo continuamente y mejorar su funcionamiento para evitar errores en el futuro, relativos al reconocimiento de patrones o de rostros, por ejemplo.

Función de activación

Función matemática, denominada en ocasiones rectificador, que se emplea en redes neuronales para obtener resultados con un menor margen de error, en contraste a otras funciones como la función logística.

Unidad Lineal Rectificada (ReLU)

Una ReLU (por sus siglas en inglés), es la unidad de medida que se utiliza en las funciones de activación.

Convolución

Operación que por lo general toma dos parámetros como elementos de entrada, los combina y genera un tercer elemento como parámetro de salida.

Convolución en profundidad

Convolución que se aplica únicamente en una sola dirección a la vez dentro de una capa de la red neuronal convolucional. Este tipo de convoluciones son particularmente notables para acelerar el cálculo de operaciones. [3]

Convolución punto a punto

Operación uno a uno o punto a punto que se aplica sobre los parámetros de la capa de convolución de la red neuronal convolucional. [3]

Capa de entrada

Capa en la cual residen los parámetros de entrada; llamada también capa visible.

Capa de salida

Capa donde se producen los valores de salida.

Capa oculta

Capa ubicada entre la capa de entrada y la capa de salida. Generalmente una capa oculta es una capa de convolución, de reducción o una capa completamente conectada; esto es, una capa donde tiene lugar el cómputo de cálculos.

Red Neuronal Convolutiva (R.N.C)

Corresponde a un tipo de redes neuronales artificiales que representa un algoritmo de aprendizaje profundo, el cual toma como parámetro de entrada cierto(s) dato(s), para asignarle(s) un peso o importancia específica y luego procesarlo(s), con el propósito de generar determinados datos como parámetros de salida. Esta clasificación de redes neuronales es de gran utilidad sobre todo para procesar señales de audio, voz e imágenes, debido a que presentan un rendimiento más eficaz que una red neuronal artificial estándar. [4]

Una R.N.C. se compone de tres capas principales [4]; la primera de ellas es una capa de convolución. Esta es el núcleo sobre el cual se construye la red, puesto que aquí es donde tiene lugar la mayor parte del procesamiento de datos. Necesita datos como parámetros de entrada, que luego analizará para identificar la presencia de ciertas características entre dichos datos. Como salida, la capa arrojará nuevos conjuntos de características. En cada operación que ejecuta, la red aplica una transformación ReLU a cada conjunto de características que ha encontrado.

La segunda es la capa de reducción, que representa una operación que se suscita entre dos capas de convolución. Recibe como entrada el conjunto de características arrojado como salida en una, y luego filtra los datos para obtener resultados más precisos y preservar sus características más relevantes.

Por su parte, la tercera es la capa completamente conectada; en esta capa cada salida de la capa anterior se encuentra conectada con un peso a cada neurona presente en la capa.

Aprendizaje de máquinas

Rama de la Inteligencia Artificial que permite desarrollar técnicas para que un sistema aprenda a través de analizar datos, en lugar de aprender a través de programación explícita.

Aprendizaje profundo

Tipo de aprendizaje de máquinas que imita la manera en cómo los seres humanos obtienen conocimiento. Modela una abstracción de datos utilizando una arquitectura de computadoras para realizar una transformación no lineal de datos.

Inteligencia Artificial (I.A.)

Ciencia e ingeniería que se encarga de la construcción de máquinas inteligentes, especialmente, de programas informáticos. Abarca la simulación de tareas por parte de sistemas o entes, que intentan replicar el comportamiento humano. [5]

Dentro de este campo de estudio, podemos encontrar tres paradigmas de aproximación; el primero de ellos es el Representacional o Simbólico, que intenta representar la realidad a través del uso de símbolos que pueden llegar a tener un significado determinado. El segundo es conocido como Conexionista, que señala que toda computación puede inspirarse en el funcionamiento del sistema nervioso del ser humano; se trata de una abstracción de las redes neuronales. [5]

El tercero se conoce como Situado o Reactivo, y postula que todo ente se comporta en base a percepciones y reacciones, las cuales interactúan con los sensores y efectores del mismo.

3.2. Análisis del artículo científico SSD: Single Shot MultiBox Detector

Para la realización de este T.F.M. se ha analizado el artículo indicado en [1], que describe el uso de una red neuronal convolucional para el reconocimiento de rostros humanos.

Arquitectura de la red neuronal convolucional

La red neuronal que se ha empleado en el artículo posee una arquitectura como la siguiente:

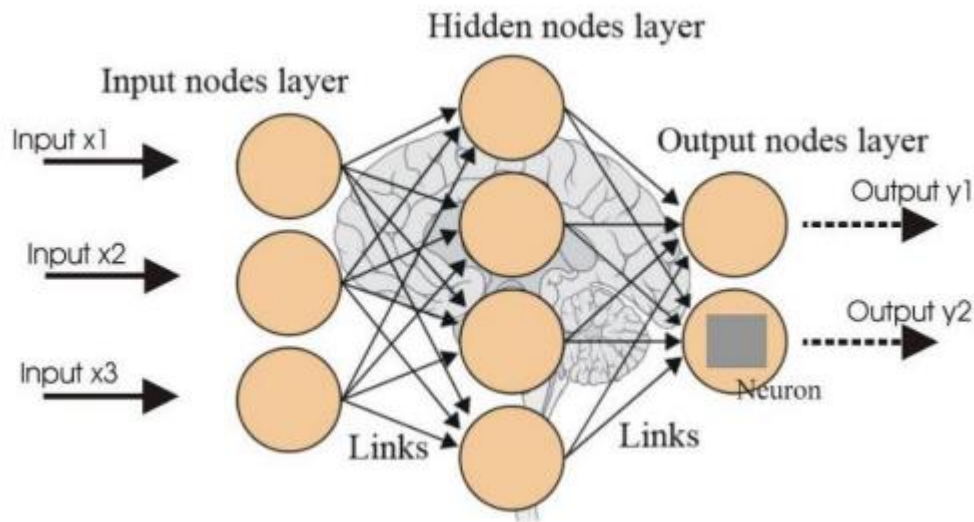


Figura 3.1: Arquitectura de la red neuronal convolucional utilizada en el artículo científico analizado [1]

Se puede observar que la red:

- Cuenta con neuronas de entrada.
- Posee una capa de neuronas de entrada.

- Tiene una capa de neuronas ocultas.
- Consta de una capa de neuronas de salida.
- Las neuronas de una capa están conectadas con las neuronas de la capa siguiente.

Los modelos

La red neuronal emplea dos modelos para detectar los rasgos y expresiones del ser humano.

Single Shot Multibox Detector (SSD) MobileNet v1

Modelo desarrollado por Google que funciona en base a una convolución separable en profundidad, y al algoritmo Single Shot Multibox Detector. [1]

Una convolución separable en profundidad es una combinación de otras dos; una convolución en profundidad y una convolución punto a punto.

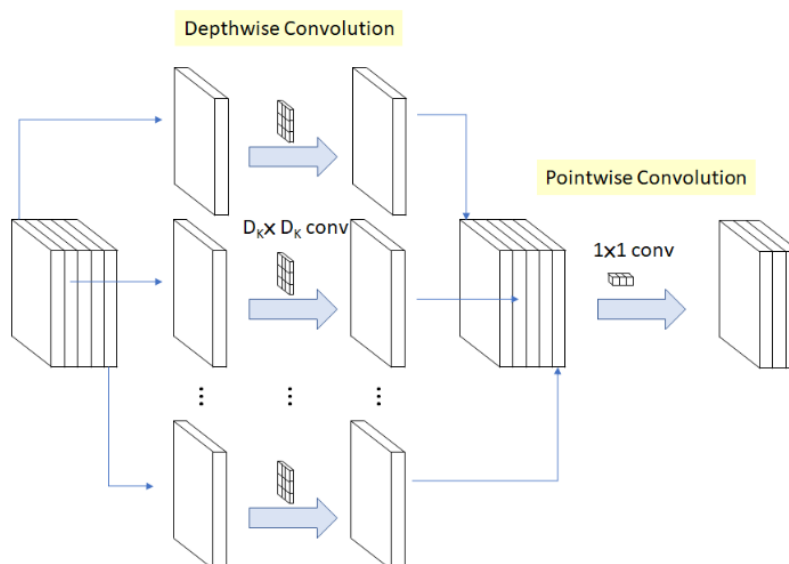


Figura 3.2: Convolución separable en profundidad [6]

La capa inicial toma como parámetro de entrada la imagen a procesar, que en esencia es una matriz de píxeles cuyas celdas tienen un valor que oscila entre 0 y 255, tal y como señala el sistema RGB (Red, Green, Blue).

En la capa de convolución en profundidad, se separa la imagen en tres canales diferentes: uno para el color rojo, verde y azul respectivamente, dado que se asume que la imagen es a color y no a blanco y negro; en este último caso el procedimiento se realizaría en un único canal, no en tres. El parámetro de salida de esta capa sigue siendo la misma imagen, pero separada en tres canales distintos.

Posteriormente, en la capa de convolución punto a punto, se efectúa el filtrado de píxeles que conforman cada imagen en cada canal. Dicho filtrado consiste en formar distintos grupos de píxeles cercanos entre sí para aplicar una operación de producto escalar entre ellos y una matriz de valores aleatorios denominada núcleo.

Así se obtiene un mapa de características, es decir, un grupo conformado por píxeles del mismo color para cada canal. Finalmente, los tres mapas de características se juntan para obtener como resultado una imagen final de un único canal.

Dada una determinada convolución clásica con un total de M canales de entrada, los cuales poseen una dimensión de $D_f \times D_f$, N canales de salida y una matriz núcleo de tamaño $D_k \times D_k$, el coste computacional para poder ejecutar la convolución es:

$$D_k \times D_k \times M \times N \times D_f \times D_f$$

En una convolución por profundidad separable, para realizar la primera parte de la convolución, es decir, la convolución en profundidad, se filtra cada uno de los canales de entrada, M en este caso, con una operación de filtro de costo $D_k \times D_k$. Así se puede obtener un total de M canales con un tamaño de $D_f \times D_f$ cada uno.

Luego, para ejecutar la convolución punto a punto, se realiza una combinación lineal de cada una de las entradas de los M canales que se obtuvieron en la fase anterior. Seguidamente, se hace una convolución con un total de M canales de entrada, matrices núcleo con tamaño 1 y N número de salidas con tamaño $D_f \times D_f$.

El poder de cómputo requerido para ejecutar una convolución separable por profundidad es:

$$D_k \times D_k \times M \times D_f \times D_f + M \times N \times D_f \times D$$

Una de las ventajas de una convolución por profundidad separable en comparación a una convolución clásica es que estas últimas aprenden un total de $M \times N$ filtros; esto es, para cada canal de salida, se van a generar un total de M filtros que son independientes los unos de los otros.

Las convoluciones separables por profundidad no emplean el concepto de filtros independientes, ya que su proceso de aprendizaje se basa en el empleo de un solo filtro para cada uno de los canales de entrada que lleguen a existir, o lo que es lo mismo, el total de parámetros que se necesita para ejecutar una convolución no depende directamente del número de canales de salida.

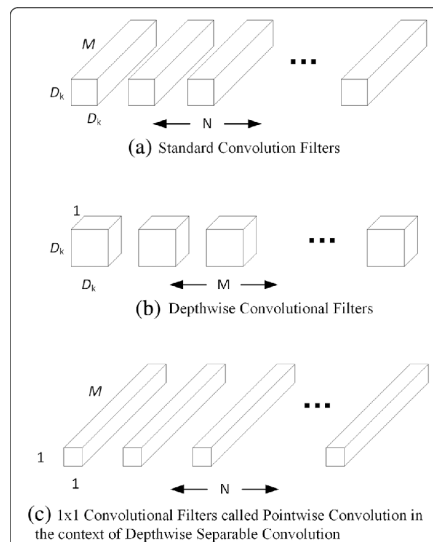


Figura 3.3: Comparación de una convolución estándar y una convolución separable en profundidad [7]

Supongamos, por ejemplo, que tenemos una imagen de W de largo y H de alto con $C = 4$ canales y una matriz núcleo de tamaño $K \times K$:

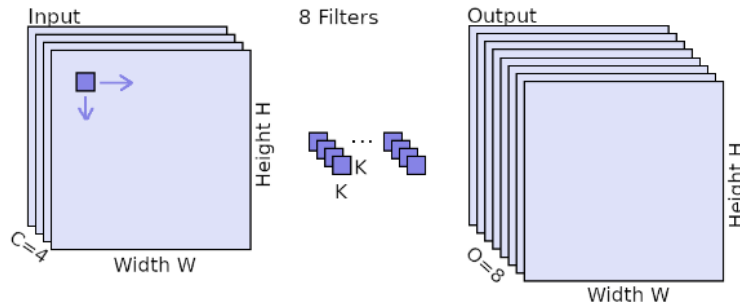


Figura 3.4: Representación gráfica del ejemplo en cuestión [8]

Aplicando una convolución en profundidad separable, se tiene que para realizar la convolución en profundidad (primera fase) sobre cada canal, es necesario computar $W \times H \times 1 \times K_w \times K_h$ operaciones. Debido a que se cuenta con un canal $C = 4$, entonces el costo computacional requerido es:

$$W \times H \times C \times K \times K$$

Ahora, la convolución punto a punto (segunda fase) utiliza una matriz núcleo de tamaño 1×1 . Como se tienen $O = 8$ canales de salida, el costo computacional requerido para ejecutar esta fase es:

$$W \times H \times C \times O$$

Sumando los costos de ambas fases, se obtiene:

$$W \times H \times C \times K \times K + W \times H \times C \times O$$

$$W \times H \times C \times (O + K \times K)$$

Como puede observarse, tenemos un factor de $(O + K \times K)$, mientras que una convolución clásica arrojaría un factor de $(O \times K \times K)$.

Asumiendo, por ejemplo, $K = 3$ y $O = 64$, se tiene que $(O + K \times K) = 73$ para una convolución por profundidad separable, y $(O \times K \times K) = 576$ para una convolución clásica.

Algoritmo Single Shot Detector (SSD)

Algoritmo que se encarga del reconocimiento de los rostros y su clasificación, realizando un recorrido hacia adelante a través de toda la red neuronal convolucional. [1]

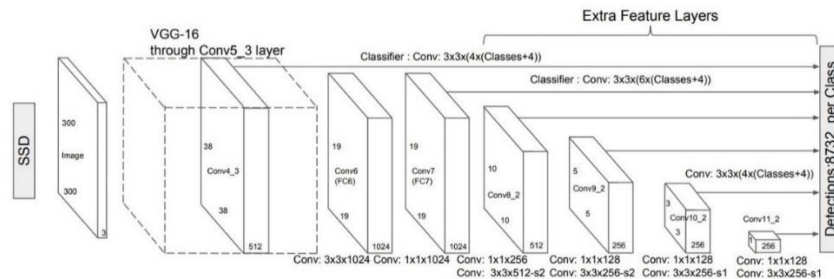


Figura 3.5: Algoritmo SSD empleado en el modelo de la red neuronal [1]

Este algoritmo tiene como base la red neuronal convolucional VGG16, uno de los modelos de visión por computadora con mayor rendimiento; pero a diferencia del VGG16, el modelo SSD elimina las capas completamente conectadas y añade capas de convolución auxiliares, para extraer los mapas de características de la imagen y disminuir su tamaño.

Tomando una imagen inicial como parámetro de entrada, el modelo se entrena para reconocer las zonas que se encuentran dentro de una caja alrededor de los objetos que aparecen en dicha imagen, hasta poder ser capaz de dibujarlas lo más correctamente posible.

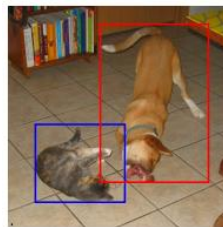


Figura 3.6: Detección de objetos mediante el algoritmo SSD [1]

Para cada zona que está dentro de una caja, se crea un mapa de características por medio de una operación de convolución:

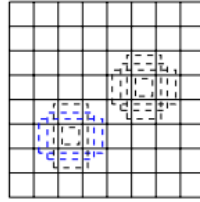


Figura 3.7: Representación de la detección de los mapas de características de la imagen [1]

Después, teniendo reconocidos todos los mapas de características de la imagen, se predice el nivel de similitud que tiene cada uno con un determinado objeto de una categoría $c_1, c_2, c_3, \dots, c_p$.

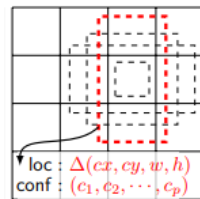


Figura 3.8: Cálculo de la distancia entre los píxeles originales de la imagen y los que se han detectado mediante el algoritmo [1]

Mediante el empleo de una función de pérdida L1, se calcula la diferencia de la distancia que existe entre los píxeles que marcan las zonas delimitadas por cajas y los píxeles originales de la imagen.

Entrenamiento

Para entrenar la red neuronal se empleó un par de conjuntos de datos. El primero de ellos, denominado ILSVR CLS-LOC, se compone de 150,000 imágenes y 1,000 tipos de categorías de objetos. El segundo fue el PASCAL VOC 2007, que tiene más de 20 clases de objetos, entre ellos sillas, botellas, bicicletas y también, rostros humanos.

Por lo que los parámetros de entrada fueron entonces diferentes imágenes que mostraban distintos rostros humanos y objetos (muebles, vehículos, entre otros). Mientras que los parámetros de salida eran las imágenes mismas con una caja alrededor de cada rostro y objeto, con un texto que indicaba la categoría a la que pertenecía. La tasa de aprendizaje fue de $10^{-3} = 0.001$.

Evaluación del sistema

Los resultados que se obtuvieron luego del entrenamiento de la red neuronal, utilizando el conjunto de datos VOC2007 fueron:

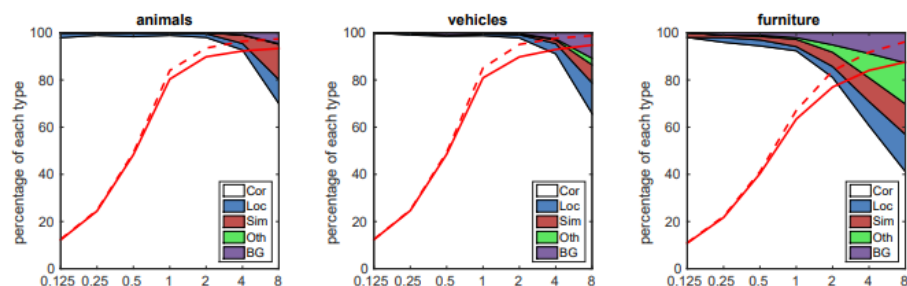


Figura 3.9: Resultados de la evaluación de la red neuronal [1]

El número de detecciones correctas (Cor) fue mucho mayor en cada caso en comparación a los falsos positivos (Loc), las confusiones entre categorías (Sim), las confusiones entre objetos la misma categoría (Oth) y las confusiones con fondos de imagen (BH).

El nivel de precisión con el cual cada objeto fue categorizado en cada fase de entrenamiento del modelo fue:

aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
74.5	78.3	69.2	53.2	36.6	77.3	78.2	82.0	40.7	72.7	67.9	79.6	79.2	73.0	69.0	30.1	65.4	70.2	75.8	65.8
77.0	78.1	69.3	59.4	38.3	81.6	78.6	86.7	42.8	78.8	68.9	84.7	82.0	76.6	69.9	31.8	70.1	74.8	80.4	70.4
70.0	80.6	70.1	57.3	49.9	78.2	80.4	82.0	52.2	75.3	67.2	80.3	79.8	75.0	76.3	39.1	68.3	67.3	81.1	67.6
76.5	79.0	70.9	65.5	52.1	83.1	84.7	86.4	52.0	81.9	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83.0	72.6
84.3	82.0	77.7	68.9	65.7	88.1	88.4	88.9	63.6	86.3	70.8	85.9	87.6	80.1	82.3	53.6	80.4	75.8	86.6	78.9
73.4	77.5	64.1	59.0	38.9	75.2	80.8	78.5	46.0	67.8	69.2	76.6	82.1	77.0	72.5	41.2	64.2	69.1	78.0	68.5
75.5	80.2	72.3	66.3	47.6	83.0	84.2	86.1	54.7	78.3	73.9	84.5	85.3	82.6	76.2	48.6	73.9	76.0	83.4	74.0
80.9	86.3	79.0	76.2	57.6	87.3	88.2	88.6	60.5	85.4	76.7	87.5	89.2	84.5	81.4	55.0	81.9	81.5	85.9	78.9
75.1	81.4	69.8	60.8	46.3	82.6	84.7	84.1	48.5	75.0	67.4	82.3	83.9	79.4	76.6	44.9	69.9	69.1	78.1	71.8
82.4	84.7	78.4	73.8	53.2	86.2	87.5	86.0	57.8	83.1	70.2	84.9	85.2	83.9	79.7	50.3	77.9	73.9	82.5	75.3
86.6	88.3	82.4	76.0	66.3	88.6	88.9	89.1	65.1	88.4	73.6	86.5	88.9	85.3	84.6	59.1	85.0	80.4	87.4	81.2

Figura 3.10: Nivel de precisión arrojado por la red neuronal [1]

Como se puede observar, cuando la imagen mostraba el rostro de una persona, el modelo nunca tuvo una precisión por debajo del 50% (véase la columna de nombre *person* en la tabla anterior).

Por lo que se pudo probar que el entrenamiento de la red era adecuado y el funcionamiento de la misma también era óptimo.

Conclusión sobre el estudio y análisis del artículo científico

Una vez analizados los resultados obtenidos, se determinó que el algoritmo SSD encargado de entrenar el modelo de red neuronal convolucional lo hizo de una manera eficaz y eficiente, pudiendo así detectar objetos y rostros de seres humanos con una precisión muy adecuada.

Este hecho se sustenta aún más en que los autores utilizaron dos de los conjuntos de datos más completos en cuanto al reconocimiento de patrones y objetos se refiere.

Apartado 4

Técnicas y herramientas

A lo largo de este capítulo se describen todas aquellas herramientas y tecnologías empleadas en el desarrollo de la aplicación web.

4.1. Descripción de técnicas y herramientas utilizadas

Se describen enseguida las herramientas y técnicas utilizadas en el desarrollo del proyecto

React

Biblioteca de JavaScript que permite construir interfaces de usuario utilizando programación reactiva.

React permite la creación de componentes (similar a las clases en Programación Orientada a Objetos). Los componentes se renderizan en el navegador y poseen un estado que permite alterar su funcionamiento para las necesidades del desarrollador.

Posee un Document Object Model (DOM) denominado DOM virtual, similar al DOM clásico de los navegadores y que permite obtener soluciones de software más eficaces.

Combina elementos HyperText Markup Language (HTML) con elementos Extensible Markup Language (XML) para crear una sintaxis denominada JSX (JavaScript XML). [9]

JavaScript

Lenguaje de programación interpretado que se utiliza generalmente para la construcción de páginas y sitios web. Cada componente que se ha generado con React, tiene código fuente escrito en JavaScript.

HTML

Se ha empleado HTML como lenguaje de marcado para definir las plantillas de las distintas pantallas que tiene la aplicación web.

Cascading Style Sheet (CSS)

Los estilos de los elementos que aparecen en la aplicación web, tales como botones, encabezados o pies de página fueron implementados utilizando hojas de estilo en cascada.

Bootstrap

Biblioteca para JavaScript, HTML y CSS que permite añadir diseños y personalización a elementos de una aplicación web, como lo puede ser un botón, encabezado o pie de página.

Visual Studio Code (VSC)

El desarrollo del código fuente de la aplicación se realizó en VSC, editor propiedad de Microsoft.

Microsoft Word Online

Se empleó Microsoft Word en su versión en línea para desarrollar el presente texto; de esta forma tanto el autor como su tutor pudieron colaborar en afinar detalles y realizar revisiones continuas.

Git

Sistema para el control de versiones. Se emplearon sus comandos para publicar las nuevas versiones de los archivos de código fuente que se iban trabajando.

GitHub

Plataforma para el desarrollo colaborativo de proyectos. Se utilizó para alojar el código fuente de la aplicación en un repositorio remoto.

Node.js

Entorno de ejecución en tiempo real para multiplataformas. Durante el proyecto se hizo uso de Node.js para crear el servidor local donde se probaba la aplicación web.

Node Package Manager (NPM)

Sistema de gestión de paquetes o dependencias para Node.js, que se ha utilizado para instalar las librerías necesarias para la generación y desarrollo del proyecto, tales como React, Bootstrap, entre otras.

Scrum

Se ha seguido el marco de trabajo conocido como Scrum. Este es una metodología ágil para la gestión de proyectos. Permite trabajar de manera iterativa y continua durante la elaboración de un proyecto, con el objetivo de que se vayan recibiendo entregables (funcionalidades de la aplicación web en este caso) cada cierto período breve de tiempo.

Entre las características de Scrum, se puede observar: [10]

- Iteración: Período de tiempo que abarca una pequeña cantidad de días, durante la cual se trabaja en el desarrollo del software.
- Sprint: Tiempo en días que tiene de duración cada iteración. En este caso se ha utilizado un total de siete días como duración de un sprint.
- Revisiones de sprint: Procesos para revisar y demostrar el avance que se realizó en el desarrollo de la última iteración.
- Retrospectiva de sprint: Evaluación y análisis de lo que ha y no funcionado durante un sprint. Se provee retroalimentación y se documentan las formas en cómo mejorar para la siguiente iteración.

Apartado 5

Aspectos relevantes del desarrollo del proyecto

El presente apartado describe los aspectos de mayor interés que tuvieron lugar durante el desarrollo del proyecto, desde las pruebas realizadas con la red neuronal convolucional hasta su integración con la aplicación web.

5.1. Funcionamiento de la red neuronal convolucional

Se han instalado las dependencias necesarias para poder hacer uso de ella a través de una Interfaz de Programación de Aplicaciones (API, por sus siglas en inglés). Tras finalizar el análisis del código fuente, se ha desarrollado la aplicación web haciendo uso de diferentes implementaciones en código para lograr reconocer el rostro de un ser humano, así como sus rasgos y expresiones faciales para determinar si se trata de un intruso o no.

5.2. Desarrollo de la aplicación web en React

La aplicación web ha sido desarrollada en React, que como se ha mencionado previamente, es una librería de JavaScript adecuada para crear aplicaciones web. Si bien el autor de este proyecto había trabajado ya con React, no es un experto en él, por lo que realizó diversos cursos en Internet y en plataformas de aprendizaje electrónico para reforzar sus conocimientos y adquirir nuevos a la vez, pues como cualquier otra tecnología, React presentaba diversos cambios en relación a las ocasiones en las cuales el autor lo había llegado a utilizar.

Los conocimientos del autor relativos a HTML, CSS y JavaScript presentaban mayor nivel de fortalecimiento en comparación a los relacionados con React, por lo que, sin ser un experto tampoco, hizo empleo de ellos para combinarlos con los de React y producir como resultado la aplicación web desarrollada.

A su vez, se ha consultado la documentación oficial de la API mencionada anteriormente para poder integrar la lógica de reconocimiento de patrones, rasgos y expresiones faciales dentro de la aplicación web.

Cuando iban apareciendo distintos problemas en el desarrollo de la aplicación, estos se fueron resolviendo consultando diferentes sitios web con información relacionada al tema en cuestión.

5.3. Comunicación entre la aplicación web y la red neuronal

Para conectar la aplicación web con la red neuronal fue necesario instalar la dependencia requerida a través del gestor de paquetes de Node. Una vez realizada esta acción, se importaron las librerías que se fueron necesitando en cada componente de React.

Dentro del código fuente de cada componente en React, es posible encontrar llamadas a métodos de la API que permiten integrar diferentes aspectos de la funcionalidad de la red neuronal dentro de la aplicación web, tales como el dibujo de rasgos y expresiones faciales o la obtención de valores de determinados parámetros de entrada y salida.

Apartado 6

Trabajos relacionados

Se muestran enseguida algunos de los trabajos considerados relevantes relacionados al empleo de redes neuronales en el reconocimiento de rostros e imágenes:

6.1. Face Recognition using Convolutional Neural Network and Simple Logistic Classifier

El artículo presenta un sistema híbrido formado por una red neuronal convolucional y un Clasificador de Regresión Logística (LRC, por sus siglas en inglés).

La red es entrenada para detectar y reconocer imágenes de rostros humanos; su arquitectura es la siguiente:

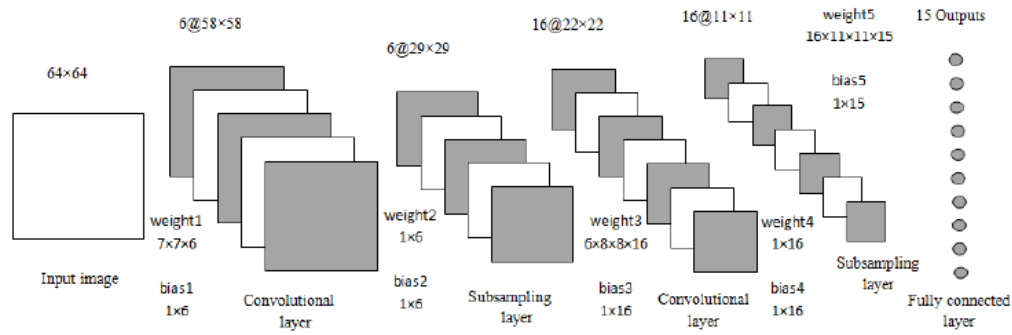


Figura 6.1: Arquitectura del modelo propuesto en [11]

El LRC se utiliza para clasificar las características de las imágenes (mapas de características) que tienen en común. En la siguiente imagen se aprecia su arquitectura:

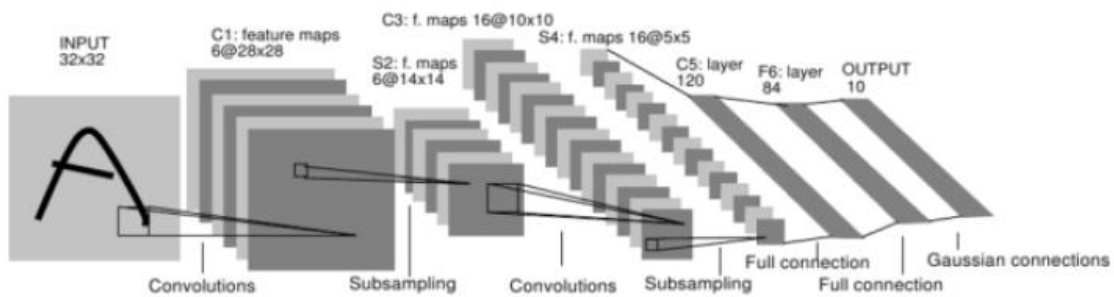


Figura 6.2: Integración del LRC con la red neuronal [11]

Como conjunto de datos se emplea una base de datos que contiene 165 imágenes de los rostros de 15 individuos diferentes, propiedad de la Universidad de Yale.

Los resultados mostraron que se obtuvo una tasa de exactitud del 86.06%, lo que permitió llegar a la conclusión de que el sistema híbrido construido era eficaz.

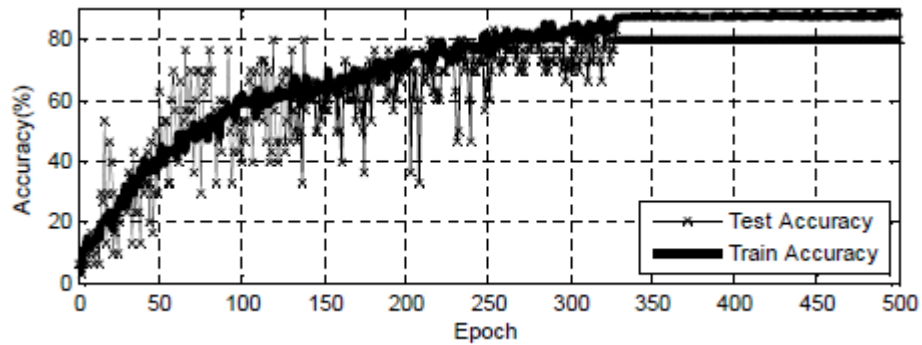


Figura 6.3: Gráfica con los resultados obtenidos; la tasa de precisión oscila entre 86.06%, lo que demuestra su excelente funcionamiento [11]

6.2. Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks

Este artículo describe la creación de un sistema encargado de la detección de rostros humanos mediante el empleo de una red neuronal convolucional en cascada multitarea.

Esta red es una composición de varias redes, y opera en un total de tres fases:

En la primera, una red neuronal completamente conectada etiqueta de candidatas a aquellas imágenes que ella cree, se asemejan a un rostro humano.

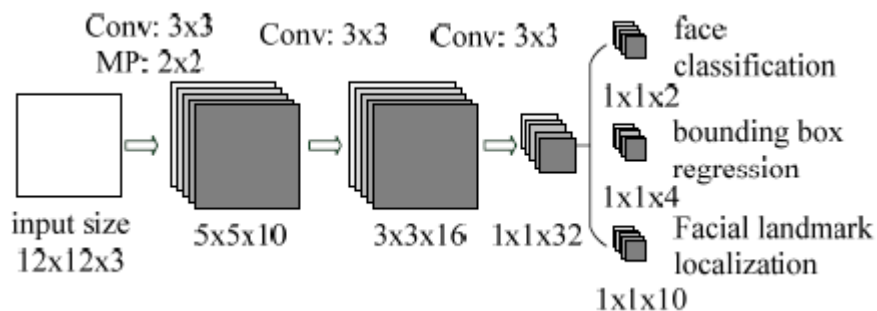


Figura 6.4: Primera fase de convolución del modelo [12]

Posteriormente, en la segunda etapa, una red neuronal convolucional (no una completamente conectada, como la utilizada en la primera etapa) descarta las imágenes que no tienen un mapa de características bien definido; esto quiere decir que muy probablemente dichas imágenes descartadas no son de un rostro humano.

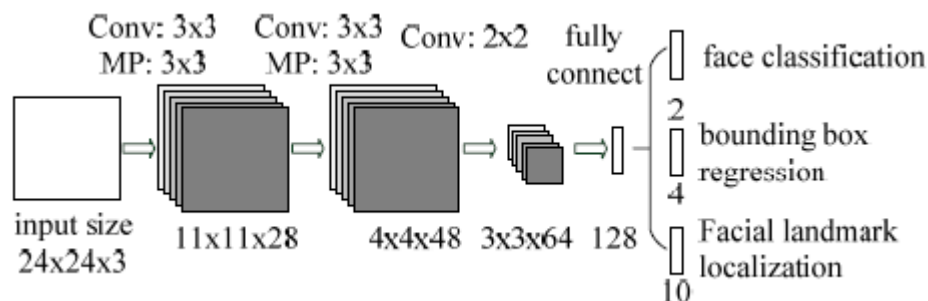


Figura 6.5: fase de convolución del modelo [12]

Por último, en la tercera etapa se lleva a cabo la clasificación de las imágenes.

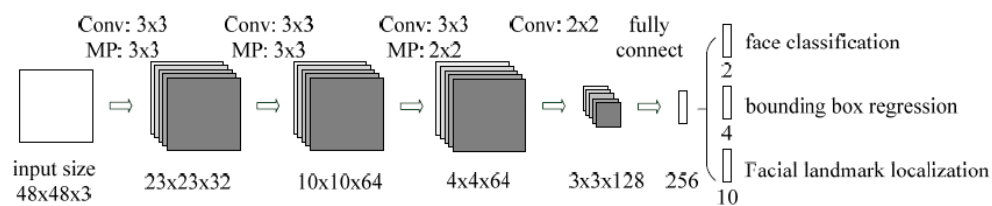


Figura 6.6: Tercera y última fase de convolución en el modelo [12]

Los resultados muestran que el modelo de red empleado tiene una mayor exactitud para clasificar rostros, en comparación a otras técnicas empleadas comúnmente en este campo:

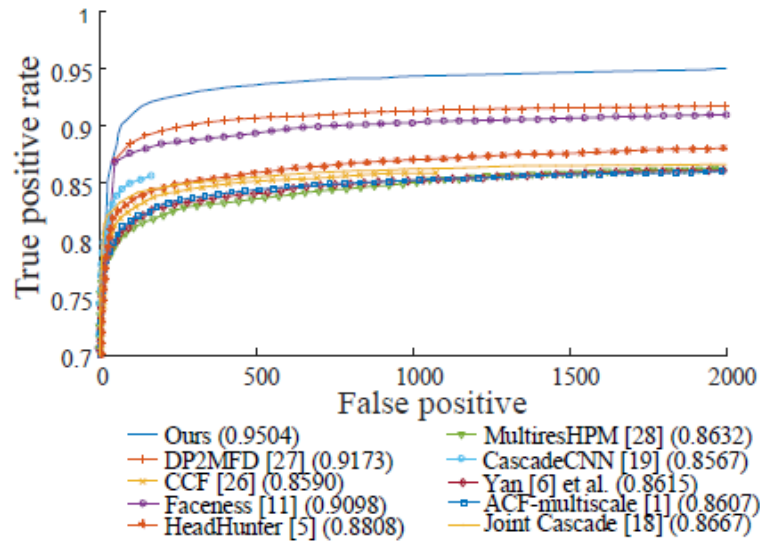


Figura 6.7: Resultados obtenidos por la red neuronal desarrollada [12]

6.3. Face Recognition Based on Lightweight Convolutional Neural Networks

En este artículo se emplea una red neuronal convolucional para el reconocimiento de rostros humanos que tiene la particularidad de contar con factores de peso muy pequeños para cada neurona que la conforma, lo que le provee un rendimiento más óptimo en comparación a otros modelos.

Su arquitectura es la siguiente:

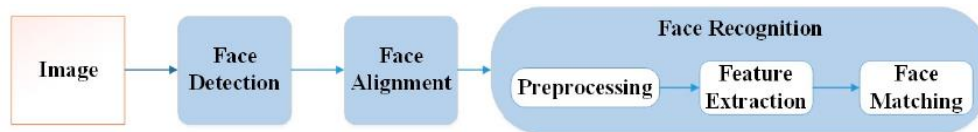


Figura 6.8: Arquitectura del modelo desarrollado [13]

Para el proceso de entrenamiento de la red, se emplearon diversos conjuntos de datos, entre los que destacan: CASIA-WebFace (0.5 millones de imágenes), MegaFace (un millón de imágenes), MS-Celeb-1M (10 millones de imágenes), YTF (3425 imágenes) y VGGFace (2.6 millones de imágenes).

Los resultados arrojaron que a pesar de que existía mucha invariancia al inicio de la fase de entrenamiento del modelo, esta disminuyó considerablemente conforme la red se entrenaba:

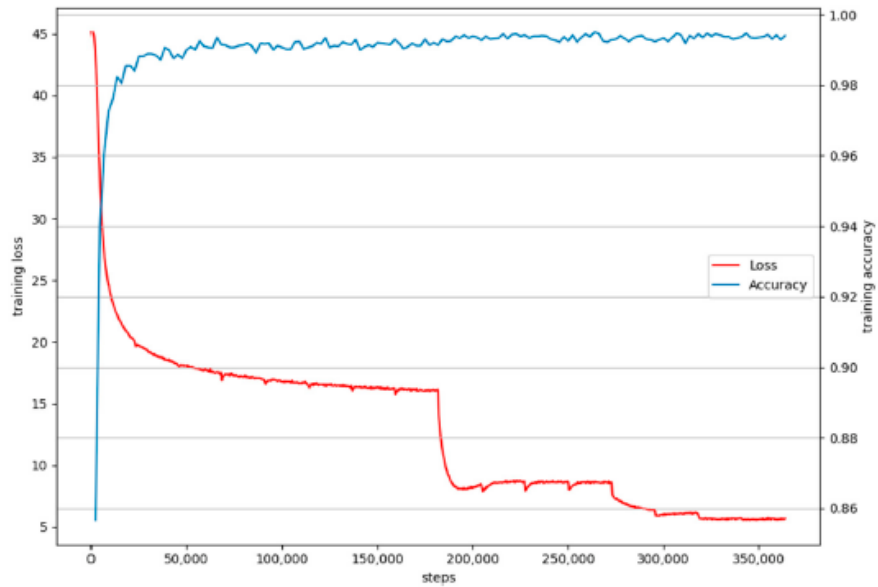


Figura 6.9: Los resultados obtenidos indican cómo disminuye la invariancia conforme el modelo iba entrenándose [13]

Apartado 7

Conclusiones y Líneas de trabajo futuras

El contenido del presente capítulo analiza los objetivos del proyecto que se han logrado cumplir teniendo de acuerdo a los que se definieron al inicio del mismo.

De igual forma, se explica un panorama a futuro para próximos trabajos adicionales relacionados al ya realizado en este T.F.M.

7.1. Cumplimiento de objetivos

El objetivo principal del presente proyecto era analizar, estudiar y comprender la red neuronal que se empleaba en el artículo SSD: Single Shot MultiBox Detector, para desarrollar a su vez una aplicación web para el reconocimiento de intrusos.

El artículo se ha estudiado y analizado, y gracias a él se ha podido hacer uso de la red neuronal descrita en el mismo para construir una aplicación web que utiliza dicha red, y es capaz de reconocer las expresiones y rasgos faciales de rostros humanos, así como la edad y el género que aparentan. Esto, aunado a la retroalimentación que se le brinda al usuario en tiempo real mientras utiliza la aplicación web, le es de mucha ayuda para considerar si la persona frente a la cámara web es o no un intruso.

Mientras que de los objetivos parciales, se puede decir:

Como se ha mencionado, se ha desarrollado una aplicación web que detecta la presencia de un rostro humano frente a una cámara, mostrando en tiempo real a través de la pantalla del ordenador el grado de exactitud.

La aplicación web es capaz de desplegar en la pantalla del computador y en tiempo real la expresión facial que posee la persona frente a la cámara, tal como enojo, sorpresa o felicidad.

El usuario puede observar en la pantalla del computador el grado de exactitud que refleja la expresión de la persona que se encuentra frente a la cámara, por medio de un texto colocado en torno al rostro de la persona.

Al mismo tiempo, es posible observar en tiempo real sobre la pantalla del ordenador, la edad y el género de la persona que está frente a la cámara.

Todos los datos que se visualizan en tiempo real (edad, expresión y rasgos faciales y género) permiten al usuario obtener una conclusión sobre si la persona ubicada frente a la cámara es un intruso o no. Además, se cuenta con información adicional que es desplegada en pantalla al mismo tiempo que se visualiza el rostro de la persona en la cámara web. Dicha información consiste en consejos del tipo *si la persona frente a la cámara tiene una expresión de felicidad y se detectan sus ojos, la persona probablemente no es un intruso, o, si la persona frente a la cámara tiene una expresión de asombro y no se detectan sus ojos, la persona probablemente es un intruso.*

7.2. Competencias demostradas

A lo largo del desarrollo del presente T.F.M se logró demostrar el buen uso de las competencias que el autor obtuvo durante su estancia escolar; algunas de ellas son:

- Resolución de problemas en entornos dentro de contextos más amplios y multidisciplinarios.

- Realizar el modelado, diseño, y gestión de aplicaciones web y contenidos informáticos.
- Aplicación de métodos matemáticos y de Inteligencia Artificial para construir sistemas inteligentes.
- Hacer el análisis, diseño, desarrollo y evaluación de sistemas informáticos combinando técnicas de Ingeniería del Software e Inteligencia Artificial.
- Obtener conocimientos avanzados acerca de las técnicas empleadas en la implementación de los aspectos de presentación e interacción con el usuario en entornos informáticos .

7.3. Trabajos futuros

A partir de los resultados que se obtuvieron luego del desarrollo de la aplicación web, se presenta una serie de diferentes aspectos que se planean mejorar en el futuro para dotar al sistema de mayor robustez:

- Permitir el guardado de los parámetros de salida que el usuario observa en pantalla en un fichero para almacenarlo; esto es, nombre de la expresión facial de la persona frente a la cámara, así como su género y edad.
- Actualizar las dependencias que utiliza la aplicación web para poder ejecutarse, tales como React DOM y Bootstrap, con la finalidad de que sea capaz de iniciarse en un tiempo menor al que posee actualmente.
- Brindarle al usuario la capacidad de detener la grabación en vídeo, para que con pulsar un botón en la pantalla, la aplicación web le genere un modelo tridimensional del rostro que está siendo captado por la cámara.
- Trasladar la funcionalidad de la aplicación web a un sistema empujado o ubicuo.

Apéndice A

Plan de Proyecto Software

A.1. Introducción

Dentro de este apéndice se presenta el plan de proyecto del desarrollo del software que se siguió para la construcción de la aplicación web.

A.2. Planificación temporal

En la presente sección se describe el trabajo que se realizó en diferentes semanas, desde el inicio del proceso de desarrollo del software hasta su finalización, pasando por el análisis del artículo científico que fue analizado.

Semana 1: Inicio

- Definir los requisitos funcionales de la aplicación web.
- Definir los requisitos no funcionales de la aplicación web.
- Definir las herramientas de desarrollo a utilizar, tales como lenguaje de programación, editores de código fuente y librerías.

Semana 2: Configuración de ambientes de trabajo

- Crear el repositorio en GitHub para albergar el código fuente.
- Crear documento del T.F.M.

- Una vez definidas las herramientas de desarrollo a utilizar, instalarlas o configurarlas adecuadamente en el ordenador.

Semana 3: Análisis de artículos científicos

- Iniciar la búsqueda de diferentes artículos científicos cuyo contenido sea acorde al problema a tratar en el T.F.M.
- Seleccionar uno, e iniciar su estudio y análisis.
- Iniciar la redacción del capítulo de fundamentos teóricos.

Semana 4: Estudio de la red neuronal convolucional

- Verificar el contenido y redacción del capítulo de conceptos teóricos.
- Comenzar con el apartado de la explicación de los temas tratados en el artículo científico que se sigue como base, dentro del capítulo de conceptos teóricos.
- Iniciar el estudio de la librería empleada para la implementación de la red neuronal convolucional.

Semana 5: Funcionamiento la red neuronal convolucional y definición de la estructura de la aplicación web

- Instalar las bibliotecas y dependencias necesarias para que la red neuronal convolucional funcione adecuadamente.
- Realizar diferentes pruebas para probar el funcionamiento de la red neuronal.
- Definir la estructura básica de la aplicación web.

Semana 6: Creación de la estructura básica de la aplicación web

- Crear la estructura básica de la arquitectura de la aplicación web.
- Crear los directorios y ficheros básicos para echar a andar la aplicación web en un servidor local.

Semana 7: Desarrollo del código fuente de la aplicación web

- Construcción de la sección del encabezado y pie de página.
- Crear la pantalla de inicio de sesión.
- Implementar la funcionalidad que le permita a los usuarios iniciar sesión dentro de la aplicación web.

Semana 8: Desarrollo del código fuente de la aplicación web, parte II

- Crear la página de Acerca De y Contacto.
- Crear la pantalla donde el usuario arribará luego de iniciar sesión (la pantalla donde se enciende la cámara web y se detecta al usuario en tiempo real).

Semana 9: Desarrollo del código fuente de la aplicación web, parte III

- Integrar la red neuronal convolucional con la aplicación web.
- Lograr detectar el rostro del usuario en tiempo real mediante la cámara web.
- Implementar la detección de expresiones faciales.

Semana 10: Desarrollo del código fuente de la aplicación web parte IV

- Implementar el reconocimiento de rasgos faciales.
- Implementar la detección de género.
- Implementar la predicción de edades.

Semana 11: Implementación de estilos

- Implementar estilos para cada sección de la aplicación web, con el objetivo de dotarla de un mayor grado de accesibilidad y fluidez.
- Comprobar el correcto funcionamiento de los estilos de la aplicación web en distintos navegadores.

Semana 12: Limpieza y optimización de código fuente

- Llevar a cabo las tareas pertinentes para limpiar cada uno de los archivos del código fuente de la aplicación; eliminar espacios en blanco, quitar comentarios no relevantes, entre otras.
- Optimizar código fuente reemplazando partes con otras que realicen las mismas tareas pero de una manera más óptima.

Semana 13: Realización de pruebas

- Realizar pruebas manuales relativas a los requerimientos de la aplicación web.
- Llevar a cabo pruebas de accesibilidad.

Semana 14: Lanzamiento de la aplicación en GitHub Pages

- Realizar la publicación de la aplicación web en un servidor de GitHub Pages para que cuente con una dirección al alcance del público.

Semana 15: Elaboración de documentación

- Escribir la parte de introducción del T.F.M.
- Elaborar el capítulo de los objetivos del proyecto.
- Revisar contenido, ortografía y semántica de dichos capítulos.

Semana 16: Elaboración de documentación, parte II

- Escribir el capítulo correspondiente a las técnicas y herramientas utilizadas durante el desarrollo del proyecto.
- Elaboración del capítulo de los aspectos relevantes del desarrollo del proyecto.
- Revisar contenido, ortografía y semántica de dichos capítulos.

Semana 17: Elaboración de documentación, parte III

- Escritura de la parte de los trabajos relacionados sobre el proyecto.
- Elaboración del capítulo de trabajos futuros.
- Revisar contenido, ortografía y semántica de dichos capítulos.

Semana 18: Elaboración de documentación, parte IV

- Creación de apéndices del documento del T.F.M.
- Creación de los diferentes índices del documento.
- Revisar contenido, ortografía y semántica de dichos capítulos.

Semana 19: Elaboración de documentación, parte V

- Escribir las diferentes fuentes bibliográficas utilizadas.
- Revisar ortografía y semántica del apartado donde aparece la bibliografía.
- Creación de portada y secciones restantes.
- Revisar ortografía y semántica de la página de portada y otras secciones restantes.
- Crear la presentación para el T.F.M.
- Elaborar el par de vídeos para la presentación del T.F.M.

A.3. Viabilidad económica

A continuación, se muestra el estudio de viabilidad económica del proyecto. Ya que no se planea lanzar el sistema a un ambiente de producción (ni tampoco su desarrollo, puesto que este precisamente ya se ha realizado), se plantea un escenario en donde se simulan estas acciones en una empresa pequeña o mediana.

A la par, se muestra la viabilidad legal bajo la cual se rige el sistema desarrollado.

En caso de que la aplicación web desarrollada fuese a lanzarse al mercado, se describen diversos aspectos que reflejan cuán viable sería esta operación.

Requisitos tecnológicos necesarios

Computadores de desarrollo con procesador Intel Core i5 o similares.

Si se deseara emplear un servidor web donde alojar la aplicación en producción, este tendría que contar con al menos (de acuerdo a lo visto hoy en día en la industria) un procesador Intel Xeon Silver o alguno similar.

Conforme el número de usuarios que utilice la aplicación vaya incrementando, se podría hacer uso de más de un servidor, pero para este fin se considerará únicamente uno solo.

Además, si no se llegase a contar con un Centro de Procesamiento de Datos (CPD), se debe de considerar optar por la ayuda de terceros para el mantenimiento y soporte de los servidores.

Por supuesto, se tiene que tomar en consideración contar con un personal con los conocimientos adecuados para llevar a cabo el desarrollo de la aplicación. Un desarrollador web con al menos dos años de experiencia bastaría.

Análisis de presupuesto

Por medio del análisis realizado en el punto anterior, es posible contar con una idea aproximada del total del presupuesto.

Concepto	Costo	Cantidad
Ordenador de desarrollo	715 €	1
Servidor	950 €	1
Mantenimiento y soporte de servidor mensuales	310 €	No aplica
Salario mensual de desarrollador web	1150 €	1

Figura A.1: Resultados del análisis de presupuesto

Como se puede observar, el costo total de la puesta en marcha del desarrollo de la aplicación web en cuestión, siempre limitándose a un escenario donde no existen grandes empresas involucradas, sería de aproximadamente 3125 €.

A.4. Viabilidad legal

El código fuente del proyecto cuenta con la licencia Massachusetts Institute of Technology (MIT), también conocida como X11. Permite que el código sea accesible para toda persona que se encuentre interesada en utilizarlo, además, no define restricciones para que el código pueda llegar a ser utilizado en otro software propietario. [14]

- Condiciones: La marca de derechos exclusivos del autor se debe de incluir en todas aquellas partes donde el software esté siendo utilizado y tenga un peso sustancial en la participación para alcanzar determinados objetivos. [14]

- Derechos: La licencia permite el uso, copia, modificación e integración del código fuente con otro software. A su vez, permite la venta de copias del software, así como la creación de publicaciones del mismo o de sublicencias. Las personas que tengan el software, serán capaces de realizar estas acciones sin ninguna clase de penalización. [14]

- Responsabilidades: El autor ni los titulares de los derechos de este son responsables de cualquier tipo de daño que pueda surgir relacionado al software. [14]

Apéndice B

Especificación de Requisitos

B.1. Introducción

Se muestran enseguida los objetivos a cumplir mediante la realización del presente T.F.M., además de los requisitos funcionales y no funcionales de la aplicación web que se prevén implementar.

B.2. Objetivos generales

Analizar, estudiar y comprender el comportamiento y funcionamiento de la Red Neuronal Convolucional descrita en el artículo SSD: Single Shot MultiBox Detector, para poder emplearla en el desarrollo de una aplicación web encargada de detectar intrusos.

B.3. Especificación de requisitos

Los requisitos funcionales y no funcionales que se desean cumplir a través del desarrollo de la aplicación web, son, respectivamente:

Requisitos funcionales

La aplicación web permitirá identificar las expresiones faciales de un usuario en tiempo real, por medio del empleo de la cámara web del ordenador; esto haciendo uso de la red neuronal convolucional. Por lo cual podrá reconocerse si el usuario se encuentra feliz, sorprendido o enojado.

La aplicación web identificará los rasgos faciales de un usuario en tiempo real, por medio del empleo de la cámara web del ordenador; esto haciendo uso de la red neuronal convolucional. Por lo cual podrán reconocerse tanto los ojos, la nariz, la boca y el contorno del rostro.

La aplicación web mostrará la edad y el género de la persona que se encuentra frente a la cámara en tiempo real, gracias al empleo de las funcionalidades provistas por la red neuronal convolucional.

Se mostrará en pantalla si el usuario frente a la cámara se trata de un intruso o no.

Requisitos no funcionales

La comunicación entre la aplicación web y la red neuronal deberá de ser totalmente transparente para los usuarios.

El renderizado de los rasgos faciales del usuario en la pantalla debe de producirse en menos de 10 segundos.

La aplicación web deberá de tener una sección donde se indique su propósito e información sobre su uso básico.

La aplicación web estará desarrollada en React.

Existirá un componente de React para cada sección dentro de la aplicación web.

Apéndice C

Especificación de diseño

A lo largo de este apéndice se describe el diseño del sistema; se muestran imágenes de los borradores utilizados para las pantallas de la interfaz gráfica de usuario y diagramas necesarios para entender su operación.

C.1. Casos de uso

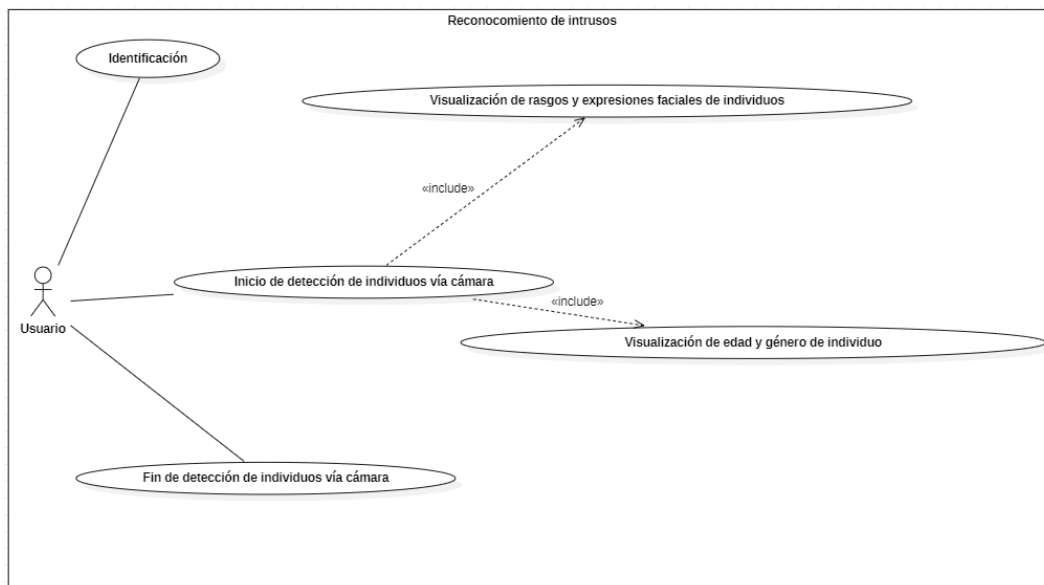


Figura C.1: Diagrama de casos de uso

Identificación de usuario	
Actor(es)	Usuario
Pasos previos	Dirigirse a la ruta del servidor que aloja la aplicación web
Pasos posteriores	Ingreso al tablero de usuario
Total de pasos	<ol style="list-style-type: none"> 1. Ingresar nombre de usuario 2. Ingresar contraseña de usuario
Condiciones	<ol style="list-style-type: none"> 1. El nombre de usuario debe de ser el correcto 2. La contraseña de usuario debe de ser la correcta

Figura C.2: Caso de uso para la identificación del usuario

Inicio de detección de individuos vía cámara	
Actor(es)	Usuario
Pasos previos	Ingresar al tablero de la aplicación web
Pasos posteriores	Encendido de la cámara web
Total de pasos	<ol style="list-style-type: none"> 1. Pulsar botón de encendido para activar la cámara web
Condiciones	<ol style="list-style-type: none"> 1. Ordenador debe de contar con una cámara web. 2. Conceder permiso de uso de cámara web al navegador utilizado

Figura C.3: Caso de uso para el inicio del proceso de detección de intrusos

Finalización de detección de individuos vía cámara	
Actor(es)	Usuario
Pasos previos	Encendido de la cámara web
Pasos posteriores	Permanencia en la sección del tablero de usuario dentro de la aplicación web
Total de pasos	1. Pulsar botón de apagado para desactivar la cámara web
Condiciones	1. Cámara web debe de estar encendida

Figura C.4: Caso de uso para el proceso de detenimiento de la detección de intrusos

Visualización de rasgos y expresiones faciales de usuario en pantalla	
Actor(es)	Usuario
Pasos previos	1. Usuario debe de estar en la sección del tablero de usuario
Pasos posteriores	1. Detener la detección de individuos
Total de pasos	1. Ingresar al tablero de usuario 2. Presionar botón de inicio de detección de individuos
Condiciones	1. Usuario debe de iniciar sesión

Figura C.5: Caso de uso para la visualización de rasgos y expresiones faciales de intruso en pantalla

Visualización de edad y género de usuario en pantalla	
Actor(es)	Usuario
Pasos previos	1. Usuario debe de estar en la sección del tablero de usuario
Pasos posteriores	1. Detener la detección de intrusos
Total de pasos	1. Ingresar al tablero de usuario 2. Presionar botón de inicio de detección de intrusos
Condiciones	1. Usuario debe de iniciar sesión

Figura C.6:: Caso de uso para la visualización del género y la edad del intruso en pantalla

C.2. Diagrama de secuencia

En la siguiente figura se puede apreciar el diagrama de secuencia de un caso de uso base de la aplicación web desarrollada.

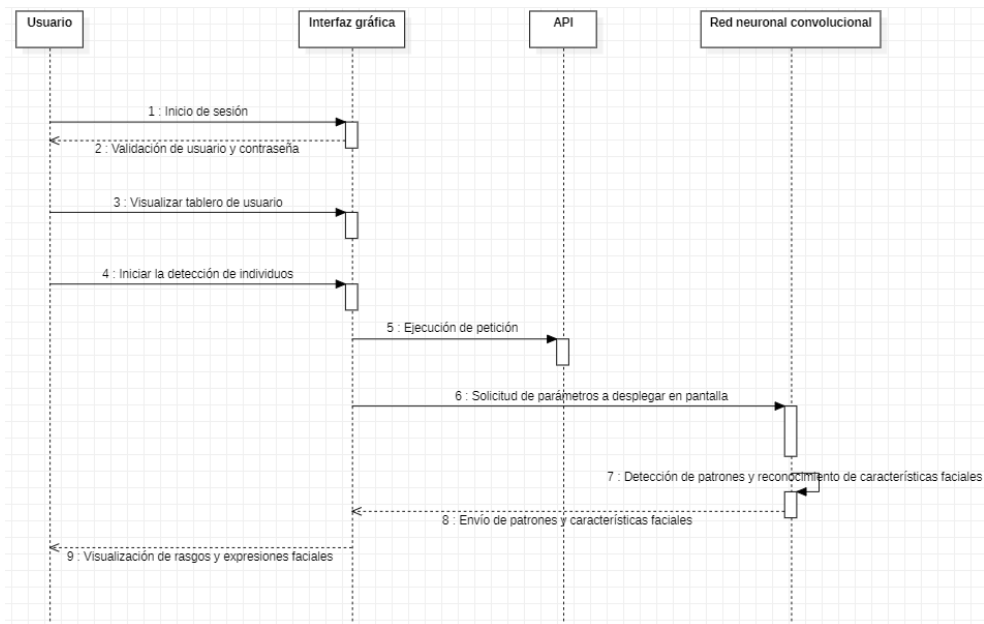


Figura C.7: Diagrama de secuencia de las acciones básicas necesarias para la operación del sistema

C.3. Diagrama de despliegue

El diagrama de despliegue que muestra los artefactos del software desarrollado junto con cada nodo involucrado se aprecia enseguida:

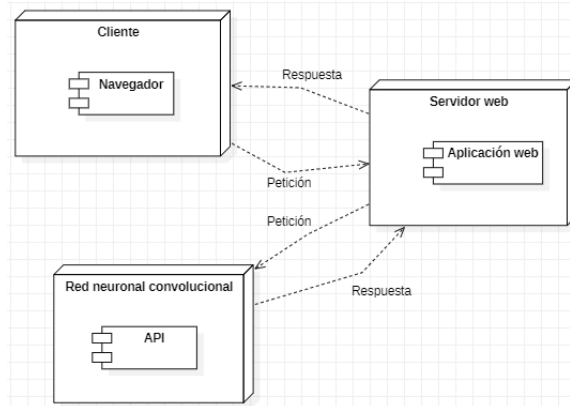


Figura C.8: Diagrama de despliegue de los diferentes nodos del sistema

C.4. Diseño de interfaz

Las siguientes imágenes muestran los borradores de las diferentes pantallas con las que cuenta la aplicación web.

The wireframe shows a login interface. At the top, the word "Bienvenido" is centered. Below it, the label "Usuario:" is followed by a text input field. Underneath that, the label "Contraseña:" is followed by another text input field. At the bottom, there is a button labeled "Iniciar sesión".

Figura C.9: Pantalla de inicio

The wireframe shows an 'About' page. The title "Acerca De" is centered at the top. Below the title is a gray rectangular box containing the text "Acerca de esta aplicación:" followed by a paragraph of Lorem Ipsum placeholder text.

Figura C.10: Pantalla de la página de Acerca De

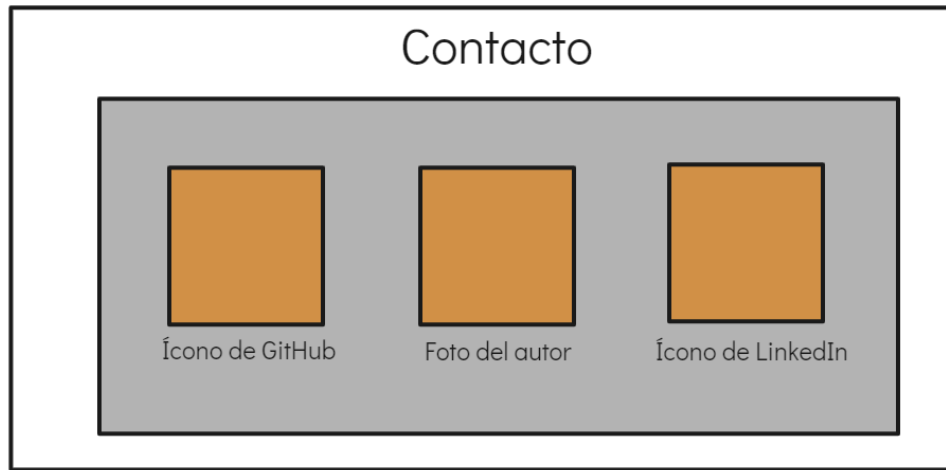


Figura C.11: Pantalla de la página de Contacto

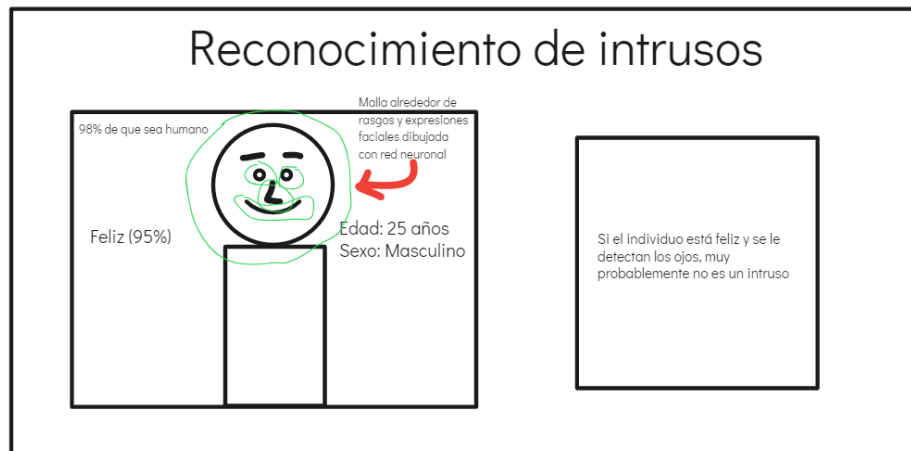


Figura C.12: Pantalla del tablero del usuario; aquí se comienza la detección de intrusos

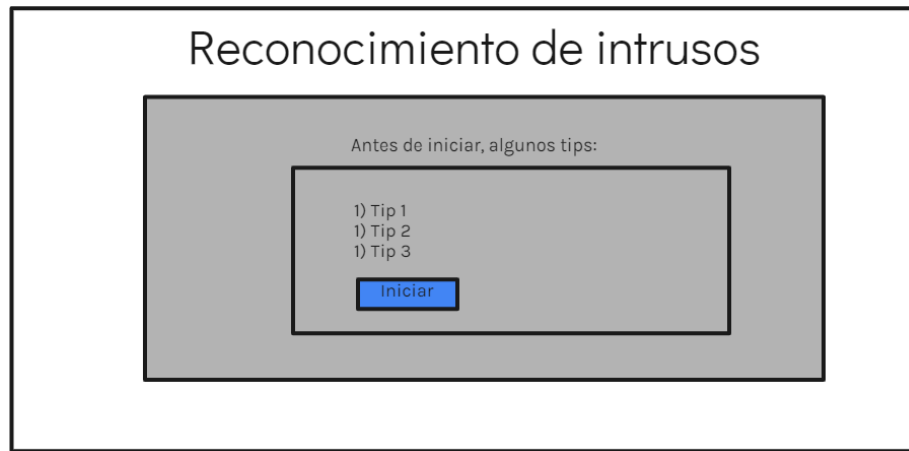


Figura C.13: Pantalla que muestra los consejos necesarios para el uso del sistema

Apéndice D

Documentación técnica de programación

D.1. Introducción

A lo largo de este apéndice se encuentra la documentación que describe las características técnicas de la aplicación.

D.2. Estructura de directorios del proyecto

A continuación se describe la estructura general de la aplicación web desarrollada:

- `.git`: Directorio creado por defecto cuando se inicializa un repositorio de GitHub.
- `node_modules`: Carpeta que contiene todas las dependencias necesarias para la ejecución del proyecto, tales como React, Bootstrap y React Dom.
- `public`: Directorio que almacena las imágenes y modelos empleados en la aplicación web.
- `src`: Carpeta que cuenta con todos los componentes y las pantallas que se crearon con React.
- `.gitignore`: Directorio para almacenar los ficheros que se omiten cuando se sube código fuente al repositorio de GitHub.
- `package.json`: Archivo que contiene una lista de los nombres de las dependencias almacenadas en `node_modules`.
- `package-lock.json`: Fichero que guarda el número de versión de cada dependencia definida en el archivo `package.json`.
- `README.md`: Archivo donde se almacenan las instrucciones a seguir necesarias para ejecutar el proyecto.

D.3. Manual de programador

La presente sección describe el contenido a detalle de los directorios con los que cuenta la aplicación web:

- `node_modules`: Esta carpeta, como se ha mencionado previamente, contiene las distintas dependencias que son necesarias para la ejecución del proyecto. Son un poco más de 800, por lo que sólo se muestran algunas de ellas en la siguiente imagen:

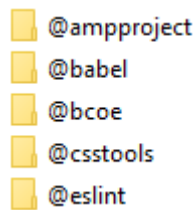


Figura D.1: Algunas de las dependencias empleadas en el desarrollo del sistema

- `public/index.html`: Fichero que toma el servidor web como base al momento de iniciarse.
- `public/images`: Imágenes que se despliegan en la interfaz gráfica de usuario.
- `public/models`: Modelos empleados para la detección de rasgos y expresiones faciales.
- `src/pages/about.js`: Página que despliega la sección de Acerca De dentro de la aplicación web.
- `src/pages/contact.js`: Página que despliega la sección de Contacto dentro de la aplicación web.
- `src/pages/footer.js`: Página que despliega la sección del pie de página dentro de la aplicación web.
- `src/pages/header.js`: Página que despliega la sección del encabezado dentro de la aplicación web.
- `src/pages/home.js`: Página que despliega la sección del inicio de sesión dentro de la aplicación web.
- `src/App.js`: Fichero que almacena la parte de la programación que se encarga de cargar los modelos, comenzar con la grabación de vídeo por medio del empleo de la cámara web, realizar los cálculos necesarios para desplegar los rasgos y las expresiones faciales en la pantalla del usuario y detener el vídeo cuando el usuario pulsa el botón correspondiente.

Nota: Se han omitido partes del código por la extensión del mismo; esto con la finalidad de ahorrar espacio. Se puede consultar el código completo en la carpeta que lo almacena.

D.4. Complicación, instalación y ejecución del proyecto

Este apartado describe las acciones necesarias para instalar, compilar y ejecutar el proyecto.

1. Instalar Node.js y npm a través de la página web oficial; al instalar Node.js desde el enlace que aparece en la página oficial, también se instala automáticamente npm.
2. Abrir una terminal del Símbolo del sistema en Windows, una terminal bash en MacOS o una terminal de Git en cualquier plataforma (para utilizar una terminal de Git es necesario instalarlo previamente). En este caso, se utiliza una terminal del Símbolo del sistema en Windows, pero los pasos siguientes son similares independientemente de cuál terminal se utilice. Una vez abierta la terminal, verificar la instalación de Node.js y npm, como se muestra a continuación, respectivamente:

```
C:\Users\Juan Manuel>node -v  
v16.15.0
```

Figura D.2: Comprobación de la versión instalada de Node.js

```
C:\Users\Juan Manuel>npm -v  
8.5.5
```

Figura D.3: Comprobación de la versión instalada de NPM

3. Dirigirse a la carpeta raíz del código fuente del proyecto, y ejecutar `npm install` para instalar las dependencias que necesita la aplicación web para poder ejecutarse:

```
C:\Users\Juan Manuel\Documents\Maestría\intruder-recognition-app>npm install
```

Figura D.4: Comando para instalar las librerías necesarias para la ejecución del proyecto

4. Una vez que se terminan de instalar todas las dependencias necesarias, ejecutar el comando `npm run start` para iniciar el servidor web:

```
C:\Users\Juan Manuel\Documents\Maestria\intruder-recognition-app>npm run start
```

Figura D.5: Comando para ejecutar el proyecto

5. Abrir un navegador web e ingresar la dirección `localhost:3000` para abrir la aplicación web en el servidor. Como aviso, este proceso es realizado automáticamente luego de ejecutar `npm run start`, por lo que no se tiene que hacer manualmente, a no ser que se quiera ir a la ruta mencionada en otra pestaña o en otro navegador.

6. Cada vez que se realiza un cambio en el código fuente de la aplicación web, el proyecto lo detecta y se compila de manera automática. No es necesaria ninguna otra acción por parte del usuario; esto se puede observar en la terminal del Símbolo del sistema que se abrió previamente.

```
> mdt-app@0.1.0 start
> react-scripts start

(node:14004) [DEP_WEBPACK_DEV_SERVER_ON_AFTER_SETUP_MIDDLEWARE]
deprecated. Please use the 'setupMiddlewares' option instead.
(Use `node --trace-deprecation ...` to show where the warning was
(node:14004) [DEP_WEBPACK_DEV_SERVER_ON_BEFORE_SETUP_MIDDLEWARE]
deprecated. Please use the 'setupMiddlewares' option instead.
Starting the development server...
```

Figura D.6: Proyecto ejecutándose sobre el servidor web

D.5. Pruebas del sistema

Se han llevado a cabo una serie de pruebas manuales para asegurar el correcto funcionamiento de la aplicación web.

Los resultados se muestran enseguida:

Iniciar sesión con usuario y/o contraseña erróneos.

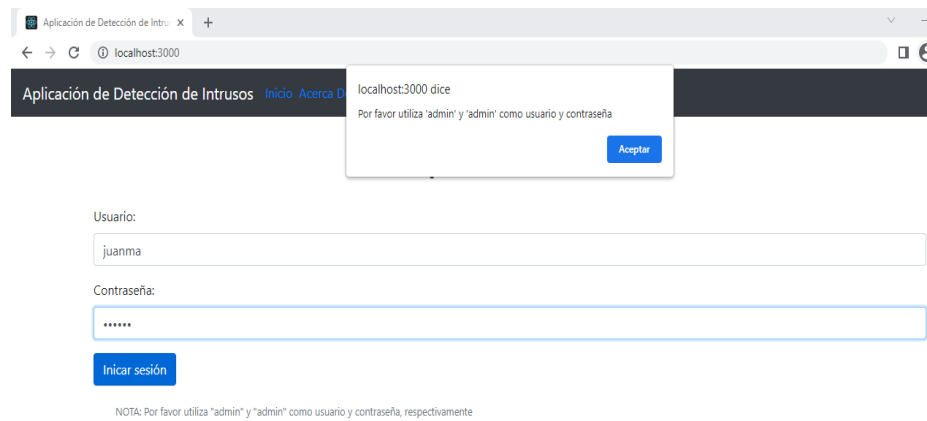


Figura D.7: Inicio de sesión con nombre de usuario y/o contraseña incorrectos

Visualización de la sección de Acerca De.

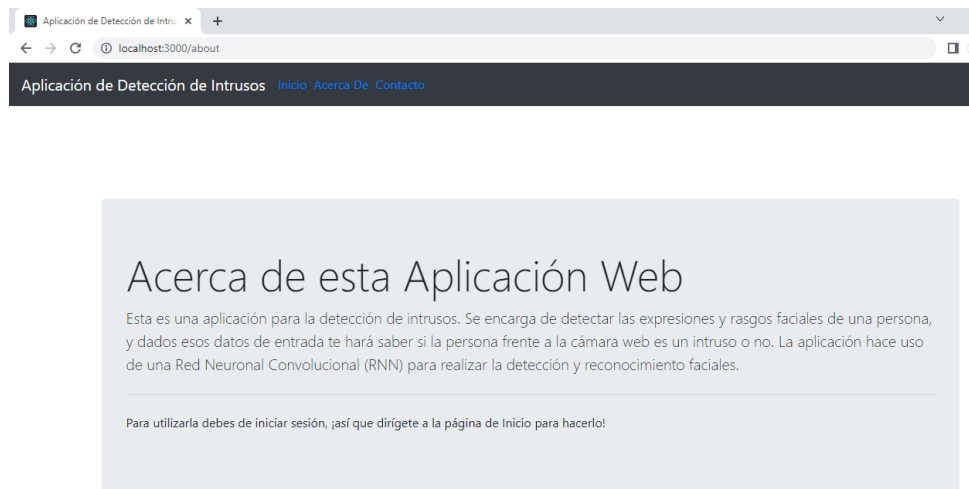


Figura D.8: Visualización de la página de Acerca De

Visualización de la sección de Contacto.

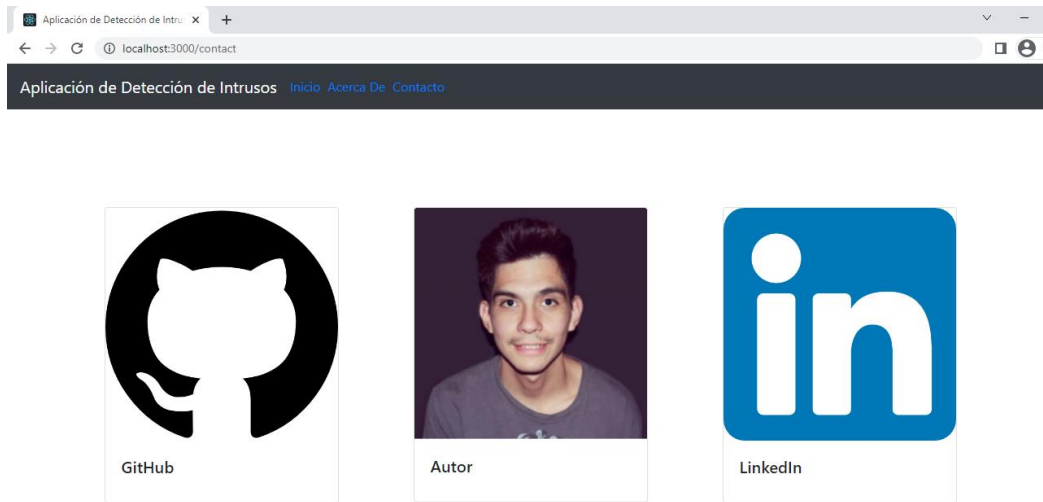


Figura D.9: Visualización de la página de Contacto

Al iniciar sesión utilizando el nombre de usuario y la contraseña correctos, se puede apreciar un mensaje con algunos consejos a tener en cuenta antes de comenzar con la detección de intrusos.

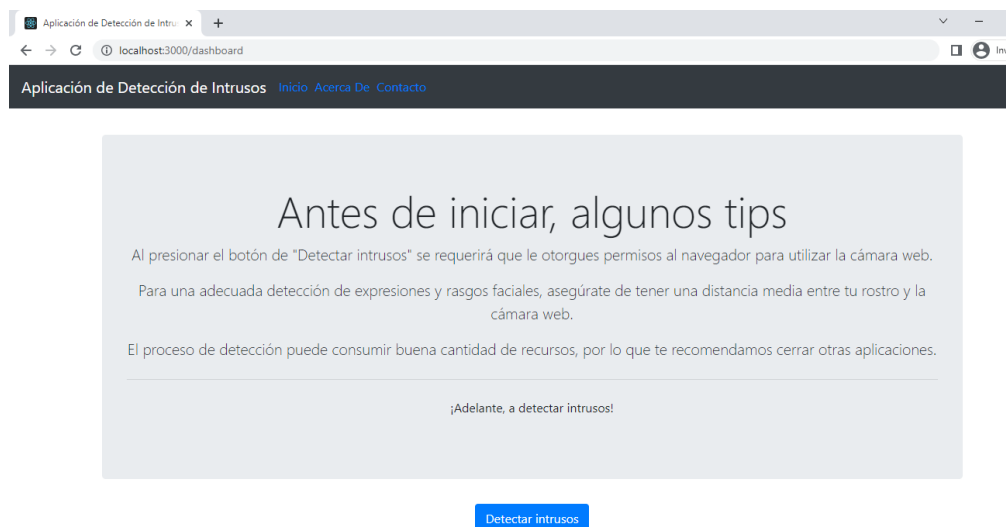


Figura D.10: Pantalla con consejos que aparece antes de iniciar el proceso de detección y reconocimiento de intrusos

Detección rasgos y expresiones faciales de un usuario; en este caso este está feliz y se puede observar también la edad que proyecta, así como su género. El número en la parte superior izquierda del cuadrado que se dibuja alrededor del rostro indica la probabilidad de que el rostro sea humano.

Al mismo tiempo, puede observarse una tabla de probabilidades en el costado derecho de la pantalla, donde se determina si la persona es o no un intruso.

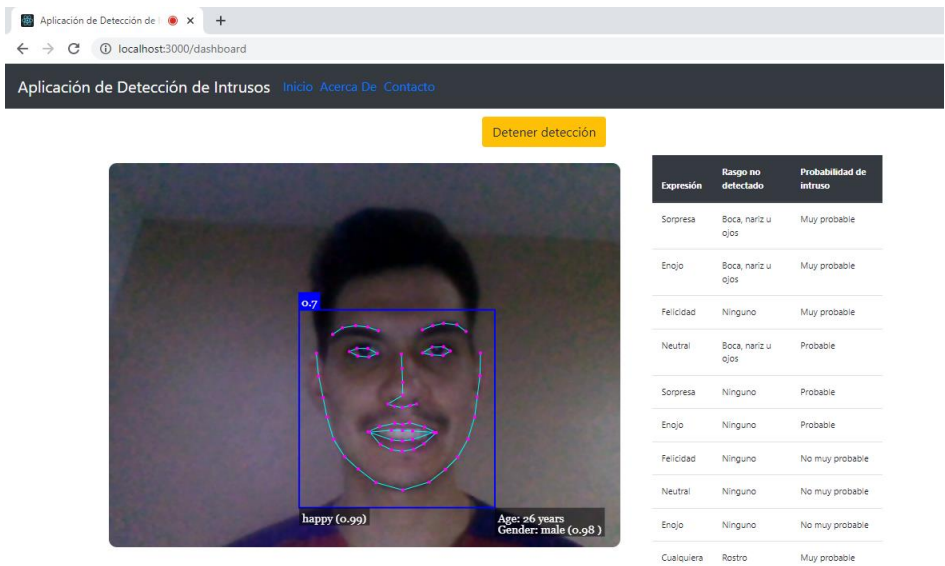


Figura D.11: Detección de rasgos faciales de una persona feliz y la probabilidad de que sea un intruso

Detección de rasgos faciales para una persona con una expresión de sorpresa en el rostro:

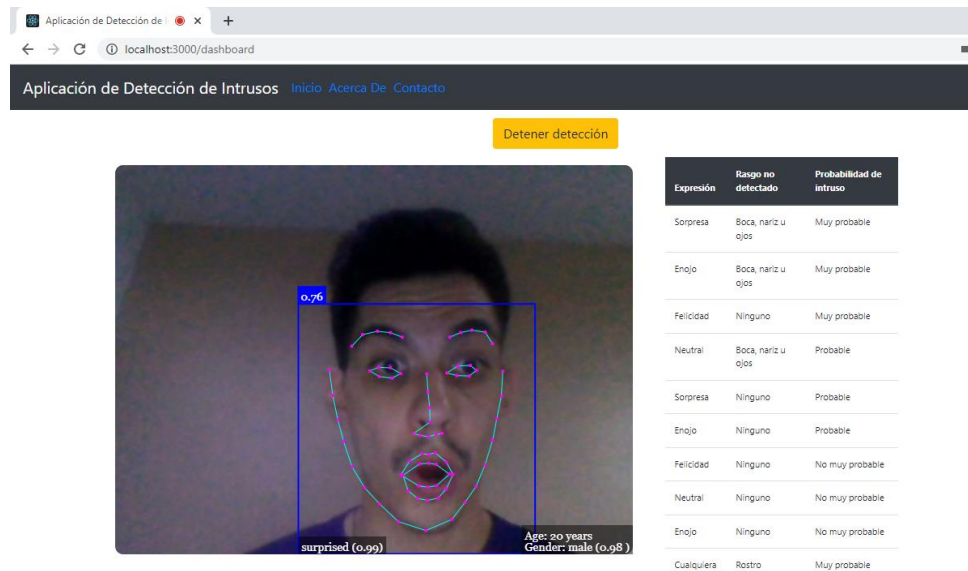


Figura D.12: Detección de rasgos faciales de una persona con expresión de asombro feliz y la probabilidad de que sea un intruso

Detección de rasgos faciales para una persona con una expresión de enojo en el rostro:

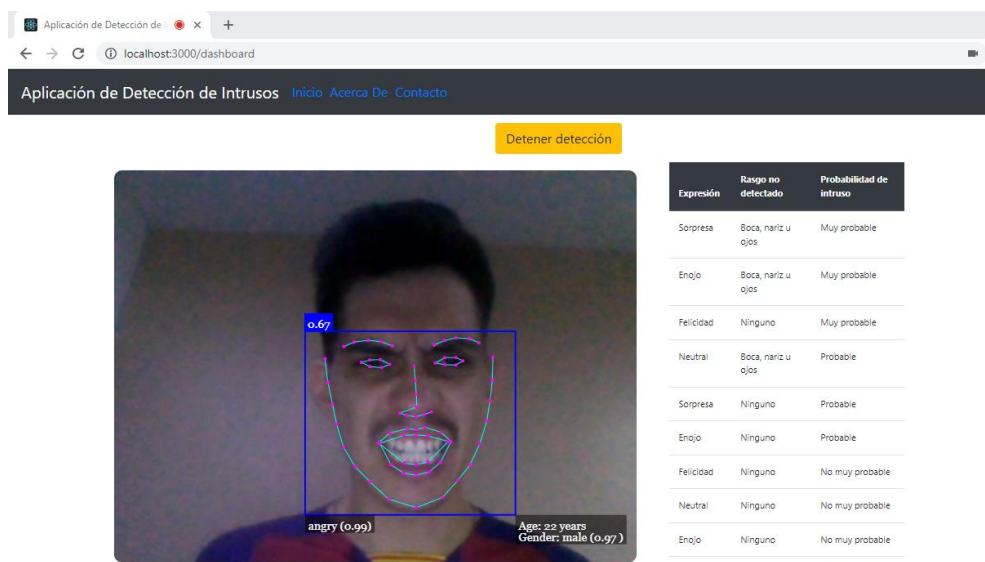


Figura D.13: Detección de rasgos faciales de una persona con expresión de enojo feliz y la probabilidad de que sea un intruso

Detección de rasgos faciales para una persona con una expresión de neutralidad en el rostro:

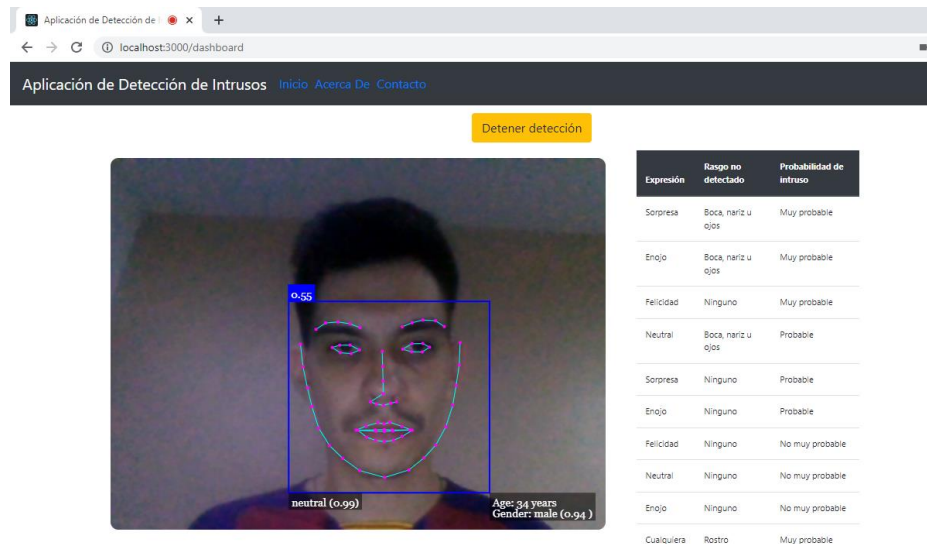


Figura D.14: Detección de rasgos faciales de una persona con expresión de neutralidad y la probabilidad de que sea un intruso

Persona que cubre parcialmente su rostro.

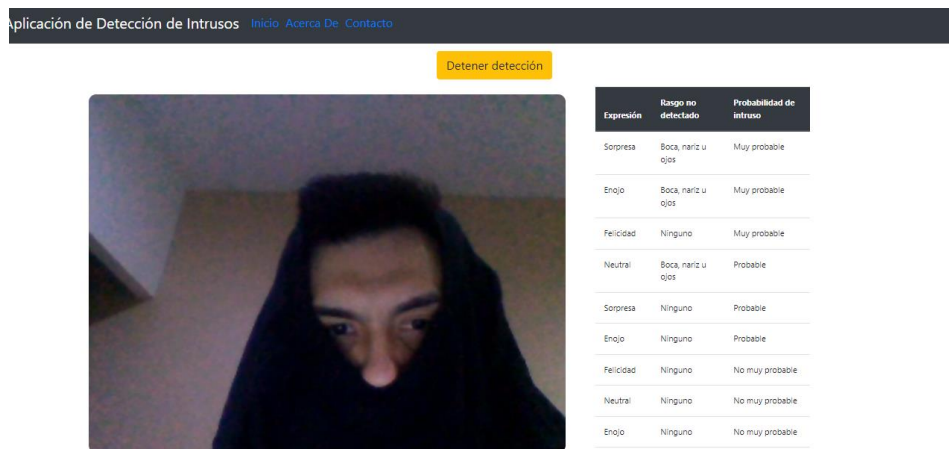



Figura D.15: Rasgos o expresiones faciales del usuario no detectados cuando oculta parte del rostro; posible intruso

Otro ejemplo similar al anterior

Aplicación de Detección de Intrusos [Inicio](#) [Acerca De](#) [Contacto](#)

Detener detección



Expresión	Rasgo no detectado	Probabilidad de intruso
Sorpresa	Boca, nariz u ojos	Muy probable
Enojo	Boca, nariz u ojos	Muy probable
Felicidad	Ninguno	Muy probable
Neutral	Boca, nariz u ojos	Probable
Sorpresa	Ninguno	Probable
Enojo	Ninguno	Probable
Felicidad	Ninguno	No muy probable
Neutral	Ninguno	No muy probable
Enojo	Ninguno	No muy probable

Figura D.16: Rasgos o expresiones faciales del usuario no detectados; posible intruso también

Apéndice E

Documentación de usuario

E.1. Introducción

Este apartado indica los requisitos necesarios para que un usuario pueda ejecutar la aplicación web.

E.2. Requisitos de usuarios

Para que un usuario sea capaz de utilizar la aplicación web, es necesario que cuente con:

- Un equipo de cómputo con cámara web instalada.
- Un navegador web instalado en dicho equipo de cómputo.

E.3. Instalación

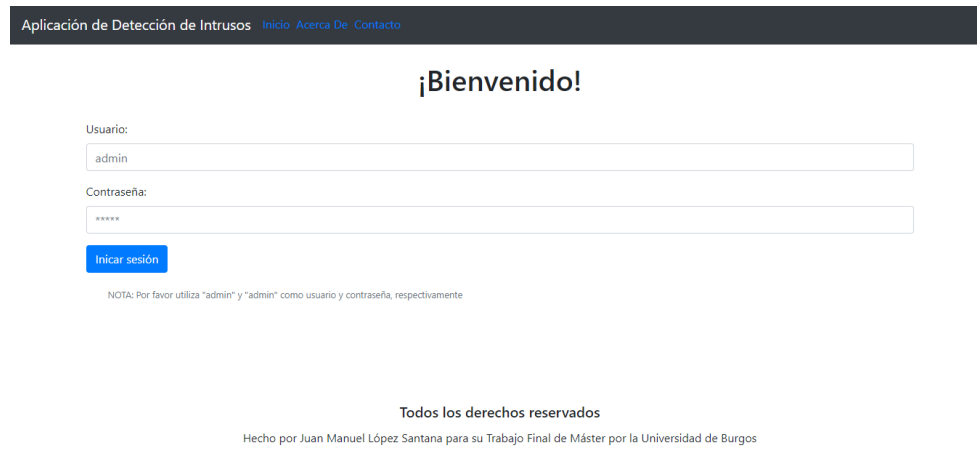
Ejecutar la aplicación web con el comando `npm run start` (para más detalles revisar la sección de instalación, compilación y ejecución del proyecto mencionada anteriormente).

E.4. Manual de usuario

Este apartado muestra el funcionamiento de la aplicación web en relación al uso que un usuario le puede proporcionar.

Iniciar sesión

La pantalla inicial de la aplicación web despliega un formulario donde el usuario debe de iniciar sesión para poder utilizar la funcionalidad de detección de intrusos.



The screenshot shows the login interface of the 'Aplicación de Detección de Intrusos'. At the top, a dark navigation bar contains the application name and links for 'Inicio', 'Acerca De', and 'Contacto'. The main heading is '¡Bienvenido!'. Below it, there are two input fields: 'Usuario:' with the text 'admin' and 'Contraseña:' with masked characters '*****'. A blue 'Iniciar sesión' button is positioned below the password field. A small note states: 'NOTA: Por favor utiliza "admin" y "admin" como usuario y contraseña, respectivamente'. At the bottom, it says 'Todos los derechos reservados' and 'Hecho por Juan Manuel López Santana para su Trabajo Final de Máster por la Universidad de Burgos'.

Figura E.1: Pantalla de inicio de sesión para el usuario

Pantalla de Acerca De

Dando clic en la opción de Acerca De en la barra de navegación de la aplicación web, se muestra una pantalla con información acerca del funcionamiento de la aplicación web.

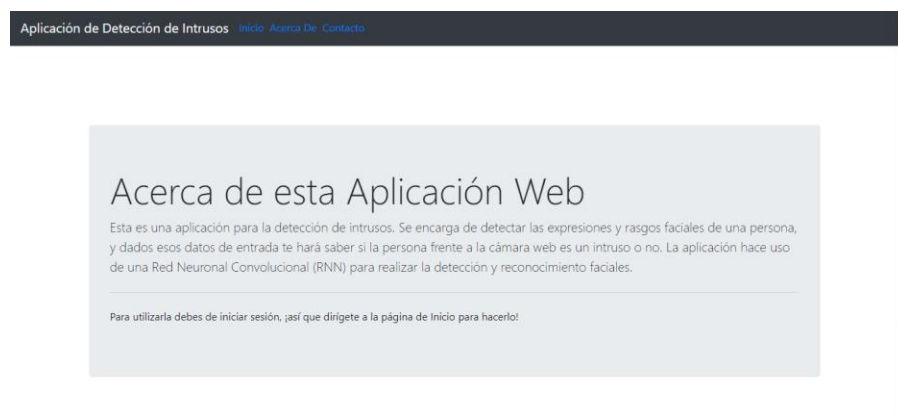


Figura E.2: Pantalla sobre la sección de Acerca De

Pantalla de Contacto

En esta página se muestra una fotografía del creador de la aplicación web, así como el enlace a su página de GitHub y LinkedIn (diseño final puede variar).

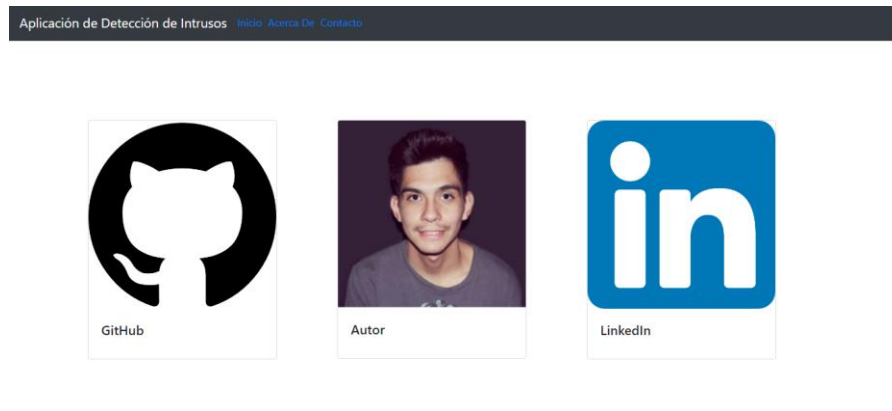


Figura E.3: Página de Contacto donde se visualiza al creador del sistema

Inicio de sesión exitoso

Al iniciar sesión exitosamente, se muestra una pantalla con algunos consejos útiles antes de iniciar con el proceso de detección de intrusos.

Para iniciar la detección de intrusos vía cámara web, es necesario pulsar el botón de Detectar intrusos.



Figura E.4: Pantalla que observar el usuario antes de iniciar a detectar posibles intrusos

Detección de intrusos

Al dar clic al botón de Detectar intrusos, se pasa a la página del tablero del usuario, donde se puede capturar el tiempo real la expresión y el conjunto de rasgos faciales del usuario. Además, se brinda una tabla con probabilidades para determinar si la persona frente a la cámara es o no un intruso.

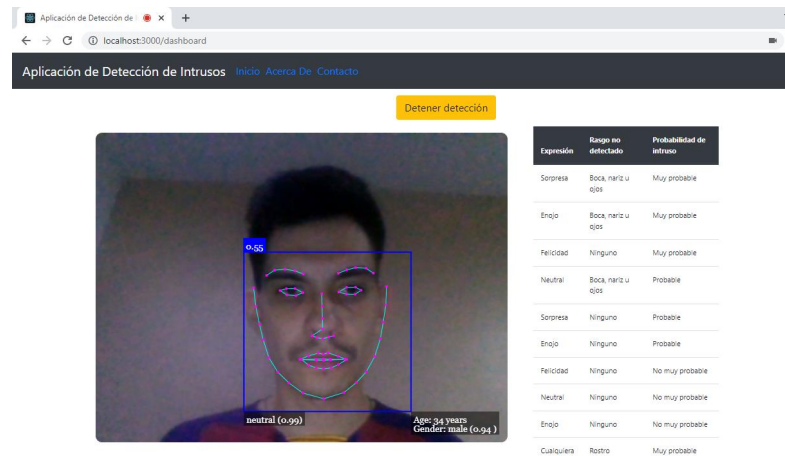


Figura E.5: Detección de un posible intruso

Detener la detección de intrusos

Para ello basta con hacer clic al botón de Detener detección; enseguida el usuario regresará a la pantalla previa:

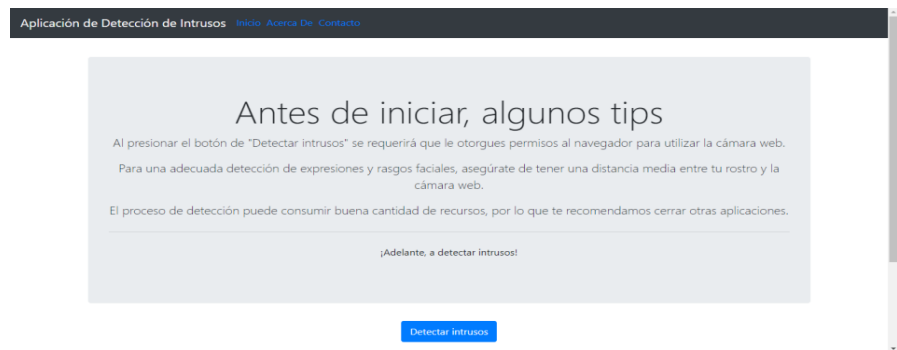


Figura E.6: Al detener la detección de intrusos, se regresa a la página anterior

Referencias

Bibliografía

- [1] Wei Liu, Dragomir Anguelov, Dumitru Erhan, cf. vv. aa. *SSD: Single Shot MultiBox Detector*, 2016.
- [2] IBM. El modelo de redes neuronales. <https://www.ibm.com/docs/es/spss-modeler/SaaS?topic=networks-neural-model>, 2019.
- [3] TowardScience. Understanding Depthwise Separable Convolutions and the efficiency of MobileNets. <https://towardsdatascience.com/understanding-depthwise-separable-convolutions-and-the-efficiency-of-mobilenets>, 2017.
- [4] Keepcoding. ¿Qué son las Redes Neuronales Convolucionales? <https://keepcoding.io/blog/redes-neuronales-convolucionales>, 2019.
- [5] IBM. Artificial Intelligence (AI). <https://www.ibm.com/cloud/learn/what-is-artificial-intelligence>, 2020.
- [6] Research gate. https://www.researchgate.net/figure/Depthwise-separable-convolution-block_fig1_343943234, 2018.
- [7] Journals - Plos - Plos-on-one. <https://journals.plos.org/plosone/article/figure?id=10.1371/journal.pone.0221720.g006>, 2017.
- [8] Paepper - Depthwise Separable Convolutions. <https://www.paepper.com/blog/posts/depthwise-separable-convolutions-in-pytorch>, 2021.
- [9] ReactJS - Getting Started. <https://es.reactjs.org/docs/getting-started.html>, 2022.
- [10] Atlassian - Qué es Scrum. <https://www.atlassian.com/es/agile/scrum>, 2019.

- [11] Hurieh Khalajzadeh, Mohammad Mansouri, Mohammad Teshnehlab. *Face Recognition Using Convolutional Neural Network and Simple Logistic Classifier*, 2015.
- [12] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, Yu Qiao. *Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks*, 2017.
- [13] Wenting Liu, Li Zhou, Jie Chen. *Face Recognition Based on Lightweight Convolutional Neural Networks*, 2019.
- [14] Choose A Licences – MIT. <https://choosealicense.com/licenses/mit>, 2017.