



Desarrollo de sistema de identificación de personas basado en aprendizaje automático y gafas inteligentes para aplicaciones de seguridad

Lascano Endara, Neptaly Alexander y Pico Benitez, Pedro Andres

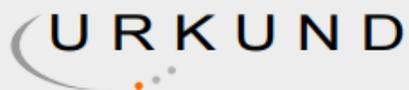
Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Ingeniería en Electrónica, Automatización Y Control

Trabajo de titulación, previo a la obtención del título de Ingeniero en Electrónica,
Automatización y Control

Ph.D. Aguilar Castillo, Wilbert Geovanny

25 de febrero del 2021



Urkund Analysis Result

Analysed Document: Tesis - Sistema de reconocimiento facial y gafas espía - Alexander Lascano Pedro Pico.pdf (D95335130)
Submitted: 2/12/2021 12:07:00 AM
Submitted By: papico@espe.edu.ec
Significance: 1 %

Sources included in the report:

TESIS_CHAUCA_BRYAN_urkund.pdf (D78121896)
Tarea12_Calvache_Garcia_Nunez_Tufino_Sani_Grupo3.pdf (D84397140)
Proyecto Juan Pareño.pdf (D62489524)
<https://bibdigital.epn.edu.ec/bitstream/15000/20098/1/CD-9539.pdf>

Instances where selected sources appear:

6

.....
Ph.D. Aguilar Castillo, Wilbert Geovanny

Director de Tesis



**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN Y
CONTROL**

CERTIFICACIÓN

Certifico que el trabajo de titulación, “Desarrollo de sistema de identificación de personas basado en aprendizaje automático y gafas inteligentes para aplicaciones de seguridad” fue realizado por los señores **Lascano Endara, Neptaly Alexander** y **Pico Benitez, Pedro Andres**, el cual ha sido revisado y analizado en su totalidad por la herramienta de verificación de similitud de contenido; por lo tanto cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, razón por la cual me permito acreditar y autorizar para que lo sustente públicamente.

Sangolquí, 26 de febrero de 2021

.....
Ph.D. Aguilar Castillo, Wilbert Geovanny
C. C. 0703844696



**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN
Y CONTROL**

RESPONSABILIDAD DE AUTORÍA

Nosotros, **Lascano Endara, Neptaly Alexander** y **Pico Benitez, Pedro Andres**, con cédulas de ciudadanía n°1718379173 y n° 1718187345, declaramos que el contenido, ideas y criterios del trabajo de titulación: **Desarrollo de sistema de identificación de personas basado en aprendizaje automático y gafas inteligentes para aplicaciones de seguridad**, es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Sangolquí, 25 febrero de 2021

Lascano Endara Neptaly Alexander

C.C.: 1718379173

Pico Benitez Pedro Andres

C.C.: 1718187345



**DEPARTAMENTO DE ELÉCTRICA, ELECTRÓNICA Y
TELECOMUNICACIONES**

**CARRERA DE INGENIERÍA EN ELECTRÓNICA, AUTOMATIZACIÓN
Y CONTROL**

AUTORIZACIÓN DE PUBLICACIÓN

Nosotros, **Lascano Endara, Neptaly Alexander y Pico Benitez, Pedro Andres**, con cédulas de ciudadanía n°1718379173 y n° 1718187345, autorizamos a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación: **Desarrollo de sistema de identificación de personas basado en aprendizaje automático y gafas inteligentes para aplicaciones de seguridad**, en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

Sangolquí, 25 febrero de 2021

Lascano Endara Neptaly Alexander

C.C.: 1718379173

Pico Benitez Pedro Andres

C.C.: 1718187345

Dedicatoria

Este trabajo es dedicado a mis padres, quienes me han acompañado durante todo el trayecto, y me apoyan día a día a seguir adelante, cumplir mis metas y mis sueños. Gracias por ser el motor que me impulsó cada día a no decaer y saber que todo en la vida con paciencia y perseverancia al final se consigue.

Neptaly Alexander Lascano Endara

Este proyecto va dedicado a mis abuelitos, que esperaron mucho por este momento. Gracias por todo el apoyo incondicional que me dieron, por nunca dejarme solo, por confiar en mí y ser mi voz de aliento.

Abuelito, este proyecto desde que se inició tenía tu nombre grabado en él. Te amo.

Abuelita, sé que estas festejando esto junto a nosotros, juntos cumplimos esto. Te amo.

Pedro Andres Pico Benitez

Agradecimiento

Agradezco primeramente a las personas más importantes en mi vida, mis padres, sin ellos, su amor incondicional y su gran apoyo hacia mí, hoy no podría estar donde me encuentro en la vida, y este mérito es mas de ustedes que mío, a pesar de todas las dificultades que hemos pasado, siempre supieron anteponerme y mis estudios y les estoy eternamente agradecido por todo lo que han hecho por mí. A mis hermanos que a pesar de la distancia siempre me apoyaban y querían ver que triunfara. A mis amigos, a aquellas personas de antaño que aún siguen a mi lado y aquellas que desde el primer día de universidad se convirtieron en una familia. A mi compañero Pedro, eres un gran amigo y un excelente compañero, sin ti no habiéramos logrado esto. Y finalmente a mi querida Michelle, que siempre me apoyó aun cuando parecía que no progresaba, y siempre estuvo a mi lado todos estos años para animarme a seguir adelante, en todo ámbito, en verdad gracias.

Neptaly Alexander Lascano Endara

Agradezco a mis padres, sin su amor, consejos, apoyo, jamás hubiera terminado esta etapa. A mi hermana, por todos esos momentos en los que te necesité y siempre estuviste ahí. A mis tíos, primos, abuelos, que me enseñaron a pensar en grande y nunca agachar cabeza. A mis amigos, que cuando los necesite me ayudaron siempre con una sonrisa. muchas palabras de aliento, muchísimas explicaciones y si era necesario, con una cerveza helada. A mi hermano Jorge, que desde el primer día que pise la universidad se convirtió en un gran amigo y hermano, gracias por siempre estar ahí. A mi compañero Alexander, sin ti esto no se hubiera logrado. Y finalmente a mi querida Mayrita, que me apoyo de principio a fin en este proyecto, que nunca me dejó solo y siempre me llevo de la mano a seguir adelante.

A todos ustedes gracias.

Pedro Andres Pico Benitez

Índice de Contenidos

Urkund.....	2
Certificación	3
Responsabilidad de Autoría.....	4
Autorización de publicación	5
Dedicatoria.....	6
Agradecimiento	8
Resumen	19
Abstract.....	20
Capítulo I.....	21
Introducción	21
Antecedentes	21
Justificación e Importancia	25
Alcance del Proyecto	27
Objetivos	29
<i>Objetivo General</i>	29
<i>Objetivos Específicos</i>	29
Capítulo II.....	31
Fundamentación Teórica.....	31
Hardware de Gafas Espía.	31
<i>ESP32-CAM</i>	31
<i>Módulo FTDI FT232RS</i>	33

<i>Cámara OV2640</i>	34
<i>Micromotor Vibrador</i>	35
Algoritmo de Reconocimiento Facial	36
<i>Inteligencia Artificial (IA)</i>	36
<i>Reconocimiento Facial</i>	37
<i>Redes Neuronales</i>	38
<i>Redes Neuronales Convolucionales (CNN)</i>	40
<i>AdaBoost</i>	42
<i>Máquinas de Vectores de Soporte (SVM)</i>	44
<i>JavaScript y TensorFlow</i>	45
<i>Faceapi</i>	46
<i>Base de Datos: MongoDB</i>	52
<i>Firebase</i>	54
<i>React</i>	55
Capítulo III	57
Desarrollo de Las Gafas Espía	57
Diseño de Hardware	57
<i>Construcción del Sistema</i>	60
Gasto Energético	65
Configuración de Enlace Entre Dispositivos	65
Especificaciones del Servidor	67

Capítulo IV	69
Desarrollo del Algoritmo	69
Diseño e Integración de la Base de Datos con el Algoritmo	71
Desarrollo de Algoritmo Para Clasificación de Datos y Reducción de Costo Energético.	73
Integración del Algoritmo con el Envío de Datos.	76
Desarrollo de Interfaz del Usuario.	77
Integración Total del Sistema.	79
<i>Costo del Prototipo</i>	81
Capítulo V	83
Validación, Pruebas y Resultados	83
Protocolo Previo a Pruebas de Experimento	84
Pruebas de Reconocimiento de Personas	86
<i>Evaluaciones en Escenario 1 – ESPE CICTE (Sala de Reunión Horas de la Mañana – Día Soleado)</i>	86
<i>Evaluaciones en Escenario 2 – Conocoto (Sala de una Casa – Horas de la Tarde - Ambiente Nublado y Lluvioso)</i>	93
<i>Evaluaciones de Detección de Personas en Escenario 3 – Sangolquí (Habitación de Estudio – Anochecer)</i>	100
<i>Comparación de Resultados Entre los 3 Escenarios de Reconocimiento</i>	102
Pruebas de Sistema Háptico	106

Pruebas de Carga Energética.....	109
Duración de Pruebas en Base a la Velocidad de Internet.....	111
Capítulo VIII	115
Conclusiones y Recomendaciones	115
Conclusiones	115
Recomendaciones	118
Referencias	120

Índice de Tablas

Tabla 1 Áreas y aplicaciones de los sistemas de reconocimiento facial	22
Tabla 2 Especificaciones técnicas ESP32-CAM	32
Tabla 3 Especificaciones de la cámara OV2640	35
Tabla 4 Especificaciones técnicas del Micro Motor Vibrador Tipo Moneda 102.....	36
Tabla 5 Especificaciones técnicas de la batería Lipo	65
Tabla 6 Especificaciones Servidor.....	67
Tabla 7 Costo elementos del prototipo	81
Tabla 8 Escenario 1 - Porcentaje de acierto de reconocimiento de personas sin luna oscura sobre la cámara sin iluminación externa	87
Tabla 9 Escenario 1 - Porcentaje de acierto de reconocimiento de personas sin luna oscura sobre la cámara con iluminación externa.....	88
Tabla 10 Escenario 1 - Porcentaje de acierto de reconocimiento de personas con luna oscura sobre la cámara sin iluminación externa	89
Tabla 11 Escenario 1 - Porcentaje de acierto de reconocimiento de personas con luna oscura sobre la cámara con iluminación externa.....	90
Tabla 12 Escenario 1 – Diferencia de resultados entre las diferentes pruebas con y sin luna de gafas oscuras sobre la lente de la cámara, en diferentes ángulos de visión ...	91
Tabla 13 Escenario 1 - Cantidad de falsos positivos en el escenario 1.....	93
Tabla 14 Escenario 1 - Promedio de resultados de pruebas en escenario 1	93
Tabla 15 Escenario 2 - Porcentaje de acierto de reconocimiento de personas sin luna oscura sobre la cámara sin iluminación externa	94
Tabla 16 Escenario 2 - Porcentaje de acierto de reconocimiento de personas sin luna oscura sobre la cámara con iluminación externa.....	96

Tabla 17 Escenario 2 - Porcentaje de acierto de reconocimiento de personas con luna oscura sobre la cámara sin iluminación externa	97
Tabla 18 Escenario 2 - Porcentaje de acierto de reconocimiento de personas con luna oscura sobre la cámara con iluminación externa	98
Tabla 19 Escenario 2 – Diferencia de resultados entre las diferentes pruebas con y sin luna de gafas oscuras sobre la lente de la cámara, en diferentes ángulos de visión ...	99
Tabla 20 Escenario 2 - Promedio de resultados de pruebas en escenario 2	100
Tabla 21 Escenario 3 - Porcentaje de acierto de reconocimiento de personas sin luna oscura sobre la cámara con iluminación externa	101
Tabla 22 Porcentaje promedio final de reconocimiento de personas	105
Tabla 23 Pruebas de sistema háptico	107
Tabla 24 Carga energética en todas las pruebas realizadas	109

Índice de Figuras

Figura 1 Índice Delictivos Nacionales	26
Figura 2 Diagrama de bloques de las gafas espía	28
Figura 3 Diagrama de flujo del algoritmo	28
Figura 4 Puertos de entrada y salida del módulo ESP32-CAM.....	31
Figura 5 Conexiones para programar un ESP32-CAM	33
Figura 6 Placa FT232RS	34
Figura 7 Ejemplo de red neuronal de 3 capas	39
Figura 8 Izquierda: Red neuronal biológica; Derecha: Representación del modelo matemático de una neurona	40
Figura 9 Esquema de una Red Convolutiva	41
Figura 10 Esquema de entrenamiento de AdaBoost	43
Figura 11 Hiperplano del SVM	44
Figura 12 Distribución de los puntos de referencia del rostro	47
Figura 13 Capas del algoritmo SSD	48
Figura 14 Izquierda: Convolución típica, Centro: Convolución profunda, Derecha: Convolución puntuada o 1x1	50
Figura 15 Arquitectura MobileNet v1	50
Figura 16 Diagrama de bloques de ResNet	51
Figura 17 Arquitectura funcional de Mongo DB	53
Figura 18 Logo React.....	55
Figura 19 Peso del diseño final de las gafas.....	57
Figura 20 Bus de alimentación de las gafas espía.....	58
Figura 21 Diagrama esquemático de la alimentación del sistema	58
Figura 22 Bus de datos de las gafas espía	59

Figura 23 Diagrama esquemático del bus de datos	60
Figura 24 Canales internos en las varillas de las gafas	61
Figura 25 Batería e interruptor colocados en la varilla externa del lado izquierdo de las gafas	62
Figura 26 Vista frontal de las gafas sin el lente derecho.....	63
Figura 27 Vista frontal de las gafas colocado el lente derecho	63
Figura 28 Vista superior de las gafas.....	64
Figura 29 Ejemplo de envío de datos al Servidor.....	66
Figura 30 Diagrama de convolución separable en profundidad MobileNetV1	69
Figura 31 Arquitectura MobileNetV1	70
Figura 32 Diagrama de convolución de MobileNetV1	71
Figura 33 Mensaje del terminal con la carga completa de modelos y conexión con la base de datos	72
Figura 34 Flujo de datos entre imagen front-end	77
Figura 35 Interfaz de usuario	78
Figura 36 Comunicación del sistema total.....	79
Figura 37 Diagrama de bloques del sistema.....	80
Figura 38 Protocolo bioseguridad Covid-19 para realización de pruebas experimentales	85
Figura 39 Escenario 1 - Reconocimiento facial acertado del sistema acompañado con un Falso positivo	92
Figura 40 Gráfica de resultados Escenario 1	103
Figura 41 Gráfica de resultados Escenario 2	103
Figura 42 Gráfica de resultados Escenario 3	104
Figura 43 Gráfica del porcentaje promedio final del reconocimiento de personas.....	105
Figura 44 Resultados de pruebas del sistema háptico.....	108

Figura 45 Resultados de pruebas en base a velocidad de internet en Escenario 1 ..111

Figura 46 Resultados de pruebas en base a velocidad de internet en Escenario 2 ..112

Figura 47 Resultados de pruebas en base a velocidad de internet en Escenario 3 ..113

Resumen

El presente trabajo es parte del proyecto de investigación "Optimización de recursos y efectivos militares a través de una plataforma web y dispositivos móviles que permita determinar su ubicación, monitoreo y visualización", y consiste en el uso de inteligencia artificial (IA) para la implementación de un sistema de detección e identificación de personas. Su funcionamiento es a través del uso del módulo WI-FI ESP32-CAM, en el que se encuentra situada una cámara que permite el envío de imágenes, además de la detección de coordenadas que determina la posición de la persona que ha sido detectada. El procesamiento se lo realiza por medio de un servidor local, que hace uso de la librería *face api*, una red neuronal previamente entrenada que detecta y diferencia rasgos faciales. El sistema se encuentra instalado ergonómicamente en unas gafas espía, que disimula toda la electrónica utilizada simulando unas gafas de sol normales. El sistema háptico que centra el objetivo funciona a través de micro motores que alertan al usuario de la detección de una persona y su ubicación respecto al ángulo de visión de las gafas. La interfaz en donde el usuario puede observar los resultados es una aplicación web basada en REACT, en la que se visualiza lo que la cámara está enviando, con una tasa de actualización de imágenes lo más cercano posible al tiempo real, y que a su vez entrega la información de la persona que se ha detectado en conjunto con sus datos previamente almacenados en una base de datos.

PALABRAS CLAVE:

- **INTELIGENCIA ARTIFICIAL**
- **RECONOCIMIENTO DE ROSTROS**
- **GAFAS ESPÍA**

Abstract

This research project consists of the use of artificial intelligence (AI) for the implementation of a system of detection and identification of people. Its operation is using the WI-FI ESP32-CAM module, which have a camera that allows the sending of images, in addition to the coordinate detection that determines the position of the person who has been detected. The processing is done by means of a local server, which makes use of the face api library, a previously trained neural network that detects and differentiates facial features. The system is ergonomically installed in spyglasses, which disguises all the electronics used simulating normal sunglasses. The haptic system that centers the lens works through micro motors that alert the user to the detection of a person and their location regarding the viewing angle of the glasses. The interface in which the user can observe the results is a REACT-based web application, in which what the camera is sending is displayed, with an image update rate as close as possible to real time, and which in turn delivers the information of the person that has been detected in conjunction with their data previously stored in a database.

KEYWORDS:

- **ARTIFICIAL INTELLIGENCE**
- **FACE RECOGNITION**
- **SPY GLASSES**

Capítulo I

Introducción

Antecedentes

En la actualidad, los sistemas de reconocimiento facial se han convertido en una herramienta que permite mejorar los protocolos de seguridad de distintas industrias y plataformas en todo el mundo. Los inicios de estos sistemas se remontan a 1960, donde Woodrow Bledsoe fue uno de los pioneros en el ámbito de reconocimiento facial. Bledsoe desarrolló una técnica llamada “Reconocimiento facial Hombre-Máquina” (NEC, 2020), la cual podía clasificar fotos de rostros a mano por medio de una tarjeta RAND, un dispositivo que por medio del ingreso de coordenadas horizontales y verticales en una cuadrícula, emitía pulsos electromagnéticos utilizando un lápiz óptico. En este sentido, se almacenaban en una base de datos puntos característicos del rostro (Galindo, Aguilar, & Reyes Ch, 2019), tales como: los ojos, nariz, boca, etc. y cuando se ingresaba una nueva fotografía, el sistema determinaba la fotografía de la persona con el mayor nivel de parentesco.

En los años 70, Goldstein, Harmor y Lesk (Bruce, 1999) mejoraron el sistema de reconocimiento facial manual de Bledsoe, dotándolo de una mayor precisión, para lo cual utilizaron 21 marcadores subjetivos específicos, aquellos podían definir características aún más específicas como el grosor de los labios o el color del cabello. Sin embargo, al igual que la técnica de Bledsoe el proceso era manual.

Para 1987, Sirovich y Kirby (Sirovich & Kirby, 1987) utilizaban álgebra lineal para resolver el problema de reconocimiento facial, con la técnica de análisis de componentes principales (PCA), esto se conoció como el enfoque *Eigenface*. Este estudio demostró

que el análisis de las características de una colección de imágenes del rostro podría formar un conjunto de características básicas, así como también, se comprobó que para codificar una imagen del rostro con precisión se necesitaban no más de cien valores. En 1991, Turk y Pentland (Turk & Pentland, 1991) extendieron el enfoque *Eigenface*, detectando rostros dentro de imágenes. Este estudio se constituye como el primer paso para la automatización de los sistemas de reconocimiento facial.

El mejoramiento de los sistemas de reconocimiento facial ha ido mejorando en dependencia de las tecnologías disponibles de cada época. Su uso se extiende no solo en empresas, sino, también en los dispositivos de uso personal, tales como celulares, tablets y computadoras. Las principales áreas de aplicación de estos sistemas se pueden observar en la Tabla 1.

Tabla 1

Áreas y aplicaciones de los sistemas de reconocimiento facial (Zhao, Chellappa, Phillips, & Rosenfeld, 2003).

Área	Aplicación
Entretenimiento	Videojuegos (Amaguaña, Collaguazo, Tituaña, & Aguilar, 2018), realidad virtual y aumentada (Grijalva & Aguilar, Landmark-Based Virtual Path Estimation for Assisted UAV FPV Tele-Operation with Augmented Reality, 2019) (Orbea, y otros, Math Model of UAV Multi Rotor Prototype with Fixed Wing Aerodynamic

	Structure for a Flight Simulator, 2017) (Basantes, y otros, 2018), interacción Humano-Robot y Humano- Computador.
Tarjetas Inteligentes	Licencias de conducir, ID's, pasaportes, inmigración.
Seguridad	Inicio de sesión o desbloqueo de dispositivos personales, encriptamiento de archivos, base de datos.
Cumplimiento de la ley y vigilancia	Video-vigilancia avanzada (Pardo, Aguilar, & Toulkeridis, 2017) (Tenezaca, Canchignia, Aguilar, & Mendoza, 2019), control de circuitos cerrados (Andrea, Byron, Jorge, Inti, & Aguilar, 2018), control de accesos.

Nota: En la tabla se muestra las diferentes aplicaciones que se le puede dar a un sistema de reconocimiento facial.

Actualmente, se han desarrollado varias aplicaciones para reconocimiento facial con un enfoque diferente dependiendo del área a la cual va dirigido. Por ejemplo, FACEFIRST (FACEFIRST, 2020) es una herramienta muy popular utilizada en la policía y ejército chinos, la cual relaciona una foto que ha sido tomada en ese instante, con una base de datos de personas conocidas, esto con la finalidad de obtener una identidad en el menor tiempo posible. Convirtiéndose en una herramienta de gran utilidad para las

fuerzas del orden público, ya que permite la detección de criminales, la disminución de falsos arrestos y la inscripción de nuevos delincuentes en bases de datos oficiales. De manera que influye directamente en la armonía y seguridad de las comunidades.

Por otro lado, FACE2GENE (Face2Gene, 2020) es una aplicación dirigida al ámbito de la salud, su función es ayudar a los médicos a determinar ciertos trastornos y variantes en sus pacientes. El sistema se ejecuta con una base de datos que posee fotos de individuos con distintas enfermedades, que debido a su condición poseen una morfología especial. Finalmente, el médico puede diagnosticar síndromes genéticos y/o enfermedades de manera eficiente y eficaz.

Otra aplicación destinada a la seguridad es FACE PHI (FacePhi, 2020), la cual es utilizada por entidades bancarias para proteger a sus clientes y prevenir algún tipo de fraude. El sistema funciona en teléfonos móviles inteligentes, en donde el usuario sube una foto de sí mismo y esta sirve como medio de identificación que permite o no el acceso a la banca móvil.

Como ya se mencionó, los aplicativos de reconocimiento facial no solo están enfocados a la seguridad de las empresas, sino también a las personas particulares, ya que, sus dispositivos móviles, tablets o computadoras, poseen sistemas como OASIS FACE, VISIDON APPLOCK PLUS o FACE LOCK PRO (FACEFIRST, 2020), que permiten el acceso al dispositivo o aplicaciones.

Los sistemas anteriormente mencionados, son algunos ejemplos de una amplia gama disponible en el mercado, su uso depende del enfoque para el que fueron creados. Estos algoritmos de reconocimiento facial pueden ser incorporados en diferentes dispositivos. Las gafas o lentes se constituyen como elementos innovadores en los que se puede asociar sistemas de reconocimiento facial. Se puede observar cómo en

Emiratos Árabes (Chavez, 2019) o China (BBC, 2018) se hace uso de dichos elementos para el desarrollado de prototipos que pueden servir como una herramienta para que el personal del orden público pueda detectar algún delincuente. Las gafas que se han desarrollado funcionan por medio de la toma de fotografías, estas son enviadas a una base de datos y retornan la información de la persona, donde se visualiza si ha cometido o no un crimen.

Considerando los antecedentes señalados y con el objeto de superar varias dificultades suscitadas en el diseño de una arquitectura totalmente funcional en un sistema de identificación de personas, se implementará en el presente proyecto de investigación, un prototipo de gafas o lentes con las funciones necesarias para recibir alertas o señales y el envío de datos, como respuesta. De esta manera permitir al usuario, obtener la información sobre la identidad de una persona.

Justificación e Importancia

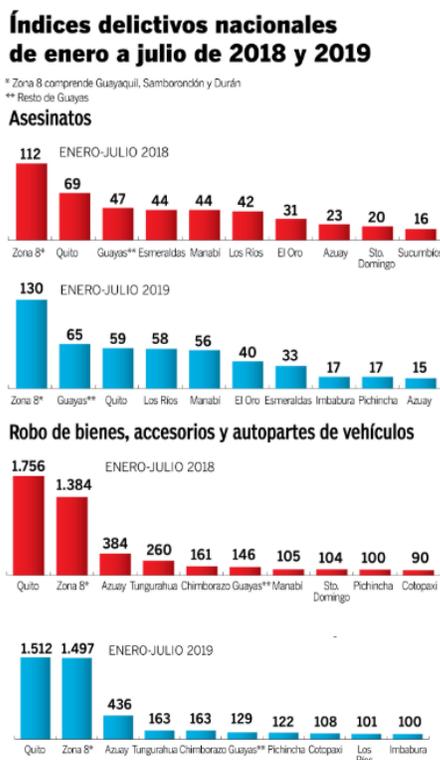
La identificación rápida y eficiente de personas puede solventar un gran número de problemáticas sociales en el país. El crimen organizado es uno de los problemas más graves que afronta el Ecuador. El grado de criminalidad incrementa de manera desmesurada, especialmente, en los últimos años. Una de las razones de este fenómeno, es la ineficiente identificación y captura de los sospechosos, principalmente en lugares que carecen de sistemas modernos de monitoreo. En la Figura 1, se observa el índice delictivo que se presenta en el país en distintas modalidades.

Tradicionalmente, los operativos delincuenciales en el Ecuador consisten en realizar tomas fotográficas de sospechosos, las mismas que son consideradas como evidencia y llevadas a un centro de detención para ser comparadas con una base de datos de individuos con orden de captura, y en caso de que la persona se encuentre dentro de esta lista, se procede a regresar al lugar de su avistamiento, para ejecutar el

procedimiento correspondiente, pese a la posibilidad de que ya no se encuentre a la persona buscada.

Figura 1

Índice Delictivos Nacionales (Diario El Universo, 2019)



Nota: En la figura se muestra información revelada por el gobierno del Ecuador, acerca del índice delictivo que existió en el país durante los años 2018 y 2019.

La suplantación de identidad es otro de los crímenes que se presenta de manera recurrente (Diario El Universo, 2019), esta problemática ha ido generando juicios por deudas no adquiridas o ingreso a establecimientos sin autorización, actualmente no se tiene la tecnología suficiente para la incorporación de un registro de personas autorizadas para realizar trámites o con permisos para el ingreso a zonas restringidas.

El sistema planteado busca aumentar la eficiencia en la identificación de personas, mediante unas gafas espía que a través de una cámara que envíe información del entorno hacia un dispositivo móvil en donde se realiza la discretización de rostros y comparación con una base de datos. Las gafas poseen un sistema de posicionamiento para indicar al usuario la dirección en la cual se encuentra la persona identificada, este sistema es de tecnología háptica y trabaja mediante vibraciones.

Alcance del Proyecto

El presente proyecto de investigación consta de dos etapas, dentro de la primera se detalla el diseño de las gafas espía, en el cual se ensambló un sistema que incluyó los siguientes elementos:

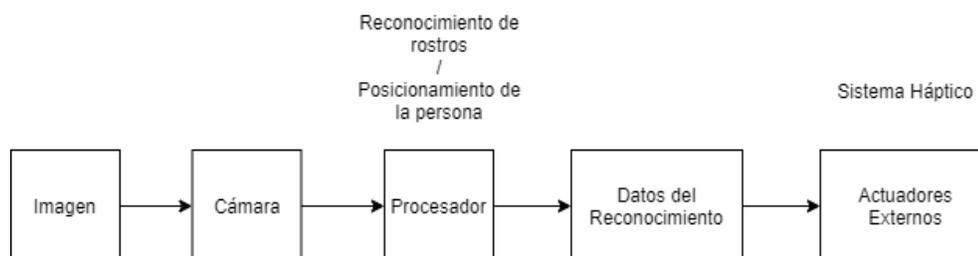
- Elemento háptico: un sistema de posicionamiento mediante vibraciones laterales que direcciona a la persona que ha sido identificada.
- Cámara: elemento de captura de imágenes del entorno que serán posteriormente procesadas.
- Procesador: módulo que gestiona el envío de imágenes y reconocimiento de coordenadas.

En la Figura 2, se observa el diagrama de bloques de la arquitectura. Se integran los elementos que permitieron la implementación de un hardware discreto, ergonómico y de bajo costo.

La segunda etapa comprende el algoritmo de reconocimiento, la construcción de una base de datos y el desarrollo de una página web que pueda ser visualizada en un dispositivo móvil. La Figura 3 comprende todas las etapas del algoritmo.

Figura 2

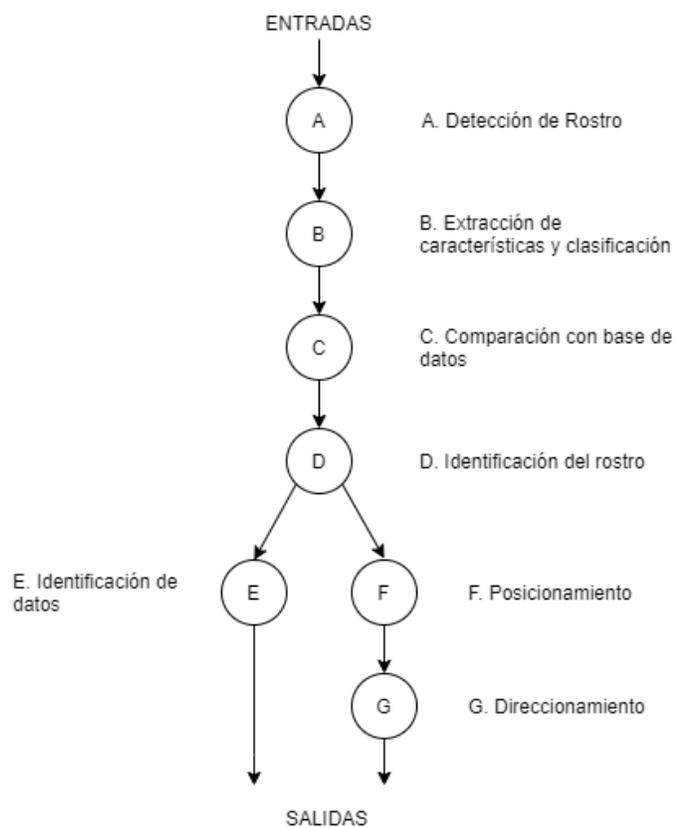
Diagrama de bloques de las gafas espía



Nota: En la figura se muestra un diagrama de bloques inicial de cómo funciona las gafas espía, expresado de forma general.

Figura 3

Diagrama de flujo del algoritmo



Nota: En la figura se muestra el esquema que sigue el algoritmo para realizar la identificación de personas y posterior entrega de resultados para las gafas espía.

La base de datos se constituyó por dieciséis voluntarios que se presentaron para participar del proyecto. El proceso de identificación se lo realizó en la Universidad de las Fuerzas Armadas y en varias localidades a lo largo de Sangolquí y Conocoto, parroquia de la ciudad de Quito. Se ejecutó la identificación de una sola persona a la vez.

Para garantizar que la identificación se realizó con éxito, se desarrolló un aplicativo web en donde se manifestó la foto de la persona que ha sido identificada con todos sus datos.

Finalmente, las pruebas del sistema de reconocimiento se realizaron en un rango aproximado a 1.5 metros. Esto se ejecutó en un ambiente intervenido, donde se controló la cantidad de luz, velocidad del internet y el número de personas frente a la cámara. El sistema no trabajó en tiempo real. Con respecto al funcionamiento energético de las gafas, se realizaron pruebas con respecto a la duración de la batería.

Objetivos

Objetivo General

Desarrollar un sistema de reconocimiento facial mediante el uso de gafas espías y algoritmos de aprendizaje automático, para aplicaciones de seguridad tales como identificación de delincuentes o de reconocimiento de personal en tiempo cercano al real.

Objetivos Específicos

- Analizar el estado del arte relativo a sistemas de gafas espía, algoritmos de detección de rostros y reconocimiento de personas por medio de rasgos faciales.
- Diseñar un hardware electrónico para gafas espía que incluya una respuesta háptica, permitiendo centrar el rostro de la persona que se desea identificar.

- Implementar un algoritmo de reconocimiento facial que sea robusto y que genere una respuesta cercana al tiempo real en el hardware diseñado.
- Realizar la evaluación de desempeño del sistema en diferentes entornos.

Capítulo II

Fundamentación Teórica

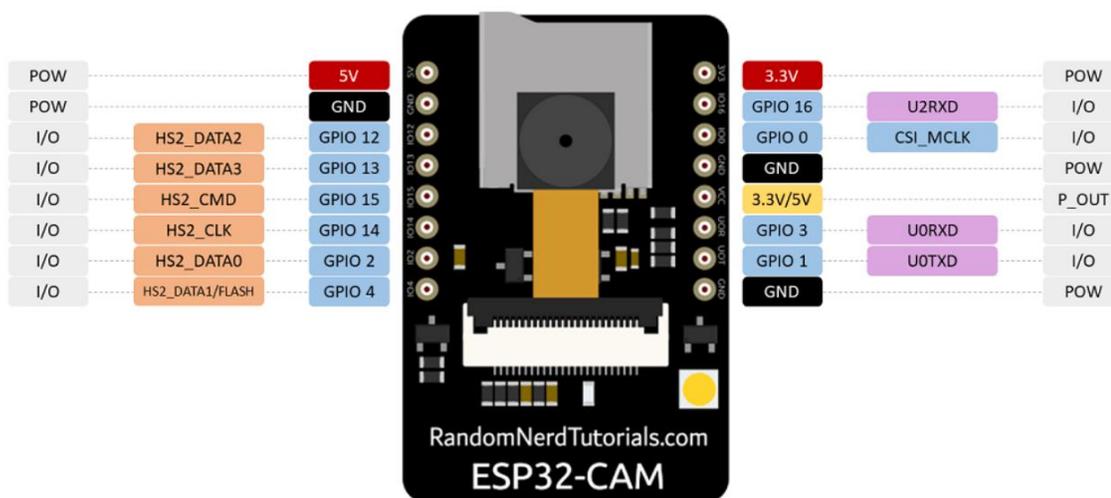
Hardware de Gafas Espía.

ESP32-CAM

El módulo ESP32-CAM (Technology Co, 2017) posee dimensiones de 27*40.5*4.5 mm, necesarias para aplicaciones que demandan su instalación en dispositivos pequeños. Este módulo es utilizado para aplicaciones como cámaras de seguridad CCTV, visión artificial embebida y visión remota para robots móviles.

Figura 4

Puertos de entrada y salida del módulo ESP32-CAM



Nota: En la figura se puede observar el diagrama de pines que presenta el módulo de la esp32-cam para su uso e intercambio de datos.

Las principales características de este módulo se visualizan en la Tabla 2, mientras que, en la Figura 4 se visualizan los nombres y ubicaciones de cada uno de los puertos.

Tabla 2

Especificaciones técnicas ESP32-CAM. (Mechatronics, 2020)

SPI Flash	32 Mbit
RAM	520KB SRAM +4M PSRAM
Bluetooth	Bluetooth 4.2 BR/ estándar EDR y BLE
Wi-Fi	802.11 b/g/n
Antena	PCB, dentro del módulo, 2dBi
UART Baudrate	Por default 115200 bps
Formato de Imagen de salida	JPEG (solo OV2640), BMP, escala de grises
Alimentación	5 V
GPIO	9
Temperatura de operación	-20°C a 85°C
Seguridad	WPA/WPA2/WPA2-Enterprise/WPS
Consumo de energía	Modo sleep liviano: mínimo 6.7 mA Modo sleep moderado: mínimo 20 mA Modo sleep total: mínimo 6 mA Cámara activada con flash desactivado: 180mA

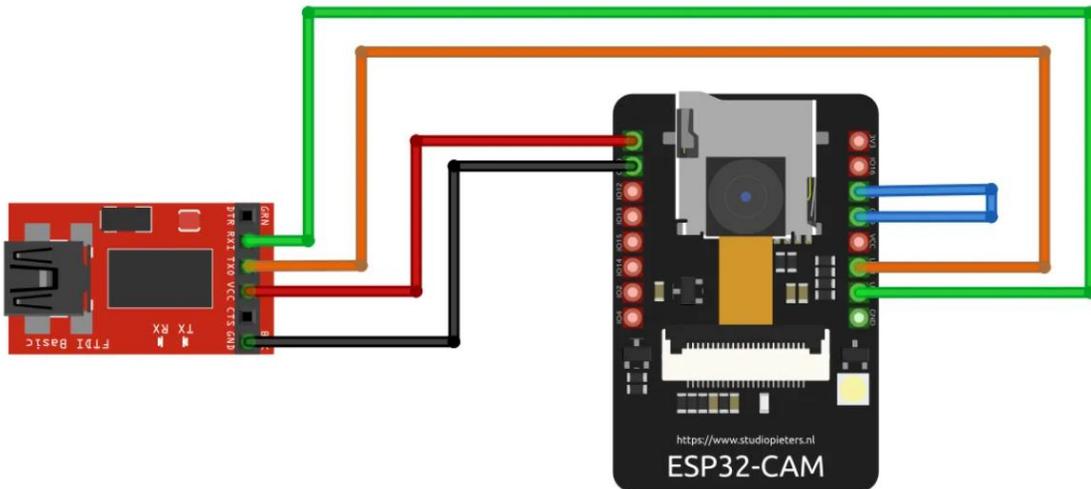
Módulo funcionando con máximas capacidades (flash activado y brillo al máximo): 310 mA

Nota: En la tabla se muestra las características disponibles en el módulo de la ESP32-cam.

Debido a que el módulo no posee una interfaz adaptable para un conector USB que permita conectarlo a una computadora para programarlo, es necesario conectar un módulo FTDI tal como se observa en la Figura 5.

Figura 5

Conexiones para programar un ESP32-CAM (Profesor Tolocka, 2020)



Nota: En la figura se muestran las conexiones básicas que deben ser usadas junto con el módulo para poder entrar en modo programación y cargar código en él.

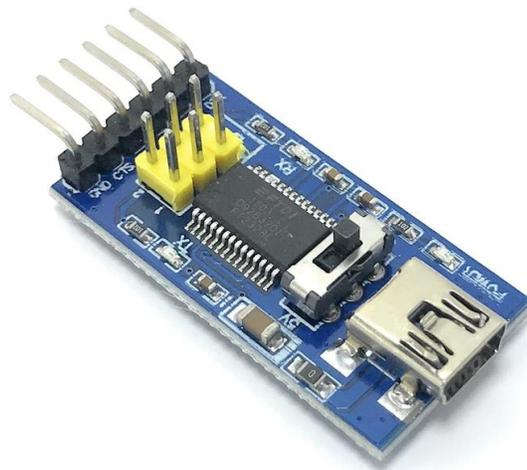
Módulo FTDI FT232RS

Es un convertidor USB-Serial (FTDI, 2012), ideal cuando el software no consta de una interfaz de este tipo implementado. Este dispositivo cuenta con un socket para

configurar el nivel de TTL de 3.3 o 5 voltios. Posee además indicadores led que permiten al usuario visualizar si se está transmitiendo y/o recibiendo la información desde esta placa hasta la tarjeta que se desea programar. Los controladores de este dispositivo están disponibles para PC, Mac y Linux.

Figura 6

Placa FT232RS



Nota: En la figura se muestra el módulo necesario para poder programar el esp32-cam.

Cámara OV2640

Esta cámara pertenece a la empresa OmniVision (OmniVision Technologies, 2006), utilizada para aplicaciones como juguetes, celulares, computadores, entre otros. Las características principales de este dispositivo se visualizan en la Tabla 3.

Tabla 3*Especificaciones de la cámara OV2640*

Fuente de alimentación	Núcleo: 1.2 VDC Analógico: 2.5 a 3 VDC Entradas/Salidas: 1.7 a 3.3 V
Tamaño de la matriz	UXGA 1600x1200
Corriente de alimentación	600 Ua
Rango de temperatura	-30 a 70 C
Transferencia de imagen	UXGA/SXGA:15fps SVGA:30fps CIF:60 fps
Sensibilidad	0.6 V/Lux-seg

Nota: En la tabla se muestran las características que tiene la cámara que se conecta directamente con el módulo de la ESP32-cam.

Micromotor Vibrador

Uno de los sistemas hápticos (Cabras, Rosell, Pérez, Aguilar, & Rosell, 2011) más pequeños que se encuentran actualmente en el mercado son los de tipo moneda, normalmente utilizados en celulares. Las características de este tipo de elementos se encuentran en la siguiente tabla.

Tabla 4

Especificaciones técnicas del Micro Motor Vibrador Tipo Moneda 1027 (SANDOROBOTICS, 2019)

Fuente de alimentación	1.5V a 3V
Diámetro del motor	2.7mm
Longitud de línea	11mm
Tiempo de vida	10000 horas
Corriente	0.05A a 0.1A
Fabricante	Haitronic

Nota: En la tabla se pueden ver las especificaciones técnicas del motor usado para el sistema háptico.

Algoritmo de Reconocimiento Facial

Inteligencia Artificial (IA)

La definición de IA difiere dependiendo del autor y del enfoque en el cual se lo tome, pero en términos generales, se puede definir a la inteligencia artificial como una disciplina científica que desarrolla programas con habilidades propias de los seres humanos como el razonamiento lógico y el aprendizaje (Russell & Norvig, 2004) (Nilsson, 1980) (Aguilar, Casaliglla, & Pólit, Obstacle Avoidance Based-Visual Navigation for Micro Aerial Vehicles, 2017).

Para diferenciar entre un sistema integrado con inteligencia artificial del resto de algoritmos automáticos, se creó una prueba propuesta por Alan Turing (Ponce Gallegos, y otros, 2014), que dicta que, en lugar de proporcionar una larga lista con cualidades controvertidas que definan una IA, la prueba este basada en la incapacidad de diferenciar

entre entidades inteligentes indiscutibles y seres humanos. El computador o sistema supera la prueba si el evaluador humano no es capaz de distinguir si las respuestas provienen de un ser humano o de una máquina. Para superar esta evaluación es necesario que el procesamiento del sistema sea en lenguaje natural, que pueda representar su conocimiento en un razonamiento y aprendizaje automático.

Para este proyecto la inteligencia artificial realizó el proceso de detección y reconocimiento de rostros humanos de manera automática (Aguilar W. G., y otros, 2017). La información se sumerge en una red neuronal que interpreta los datos de entrada y proporciona una salida lógica correspondiente a los requerimientos necesarios. Para dicho fin, es necesaria la visión artificial, que es una disciplina derivada de las IAs, la misma que consiste en un sistema integrado de hardware y software, que a partir del procesamiento de imágenes, se puede extraer información como objetos, animales, etc. (Rouhiainen, 2018)

Reconocimiento Facial

El reconocimiento facial forma parte de la visión artificial. En los últimos años se ha incrementado el estudio de este tipo de sistemas, debido a la necesidad de desarrollar aplicaciones de seguridad y vigilancia para diferentes ámbitos (Aguilar, Álvarez, Grijalva, & Rojas, 2019) (Jara-Olmedo A. , y otros, 2018) (Jara-Olmedo A. , Medina-Pazmiño, Tozer, Aguilar, & Pardo, 2018) (Zurita, Aguilar, & Enríquez, 2019) (Orbea, y otros, Vertical take off and landing with fixed rotor, 2017). El rostro humano posee una gran cantidad de información, que permite una diferenciación entre una persona y otra (Hernández, 2010).

Las etapas para realizar el reconocimiento facial se dividen en (Cadena Moreano, Montaluís Pulloquina, Flores Lagla, Chancúsig Chisag, & Guaypatín Pico, 2017):

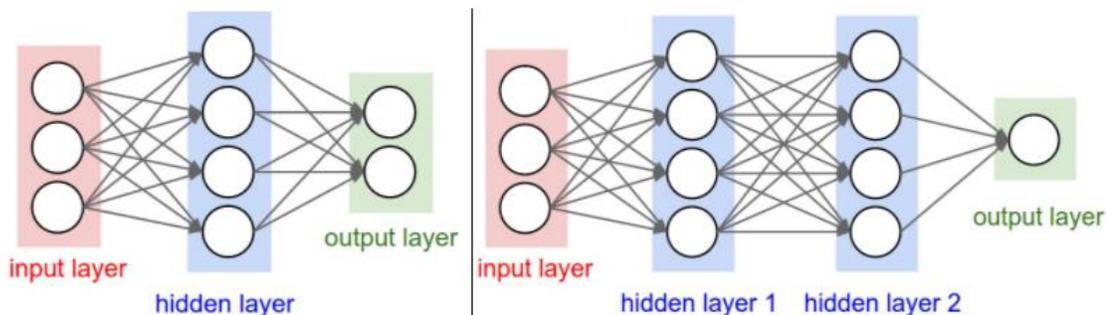
- Detección: realiza por medio de un algoritmo, localiza la región facial y la segmentación (Aguilar & Morales, 3D Environment Mapping Using the Kinect V2 and Path Planning Based on RRT Algorithms, 2016) de esta en toda la escena.
- Acondicionamiento: localiza los componentes y la escala a la que se encuentra el rostro por medio de transformaciones geométricas (Salcedo, Aguilar, Cobeña, Pardo, & Zahira, 2018).
- Normalización: normaliza las imágenes en etapas de pre procesamiento, de forma que se atenúen los efectos de la iluminación (Aguilar, Manosalvas, Guillén, & Collaguazo, 2018).
- Determinación de características: extrae la información necesaria para diferenciar el rostro principal con los que se va a procesar (Aguilar, y otros, Real-Time 3D Modeling with a RGB-D Camera and On-Board Processing, 2017).
- Reconocimiento del patrón facial: depende de las características del rostro y lo que se obtenga de la base de datos.

Redes Neuronales

Las redes neuronales son la base del Deep Learning (Aggarwal, 2018) (Calderón, Aguilar, & Merizalde, 2020). Estos modelos intentan emular el funcionamiento del sistema nervioso central por medio de unidades de procesamiento interconectado, siendo versiones abstractas de las neuronas. Están formadas por tres partes, como se observa en la Figura 7. La primera es la capa de entrada, en la que se encuentra la información que ingresa en el sistema, subsecuentemente está una o varias capas ocultas, que realizan el procesamiento del algoritmo, y al último está la capa de salida, que representa el campo de destino en donde se propaga la información. La conexión entre capas se produce a través de las unidades básicas (neuronas) y su arquitectura depende del modelo que se desea emplear.

Figura 7

Ejemplo de red neuronal de 3 capas (Pizarro Justel, 2017)

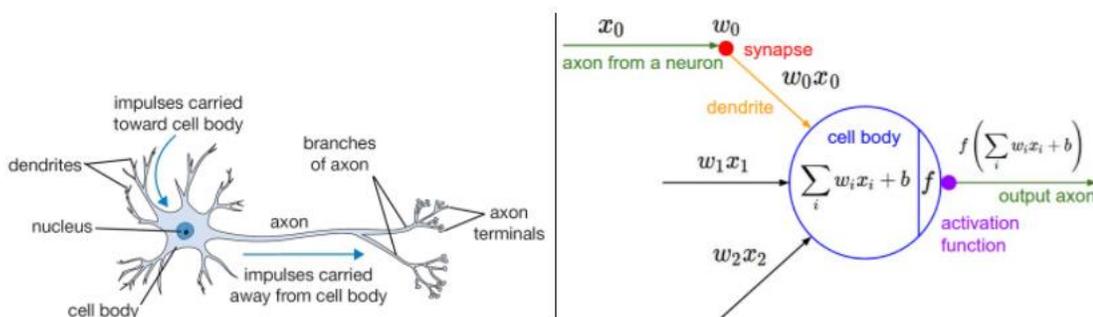


Nota: En la figura se puede ver cómo funcionan las capas de las redes neuronales en general.

El cerebro humano posee millones de neuronas que se encuentran interconectadas mediante sinapsis. Cada neurona recibe señales de entrada de sus dendritas y produce señales de salida por medio de su axón. Haciendo una analogía y con referencia a la Figura 8, en un modelo computacional de las neuronas, las señales se transmiten por medio de axones y mediante sinapsis a las dendritas de la siguiente neurona siendo la fuerza de conexión entre estas (Çiço & Nishani, 2017). Esta fuerza es conocida como pesos o parámetros, los cuales son aprendibles y controlan la fuerza de influencia o inhibición entre dos neuronas. Las dendritas son las que llevan la información hacia el cuerpo del sistema en donde se unen todas. Si la suma supera a un cierto valor umbral, la neurona propaga la información. Este umbral, viene definido por lo que se denomina función de propagación.

Figura 8

Izquierda: Red neuronal biológica; Derecha: Representación del modelo matemático de una neurona (Pizarro Justel, 2017)



Nota: En la figura se puede ver la representación de las partes de una neurona biológica y la representación matemática del mismo.

Por lo tanto, en cada una de las capas se posee pesos o parámetros, que contienen todo el conocimiento que adquirió la red. La red tiene como objetivo hallar parámetros que satisfagan en la salida un sistema, con resultados conformes a su entrada, por lo que es necesario tener un conjunto de datos robusto.

Mientras mayor complejidad tenga una red neuronal, mayor será su capacidad de aprendizaje, lo que implica que pueda resolver problemas muy complejos. Por lo contrario, mientras más simple sea la red, más limitantes presentará. Es importante tomar en cuenta que una de las características principales de estas redes es el aprendizaje adaptativo, que consiste en un entrenamiento de la red por medio de patrones que son tomados como ejemplos para aprender sobre un proceso, con el fin de reajustar a la red a los requerimientos que este implique y sean necesarios para emularlo.

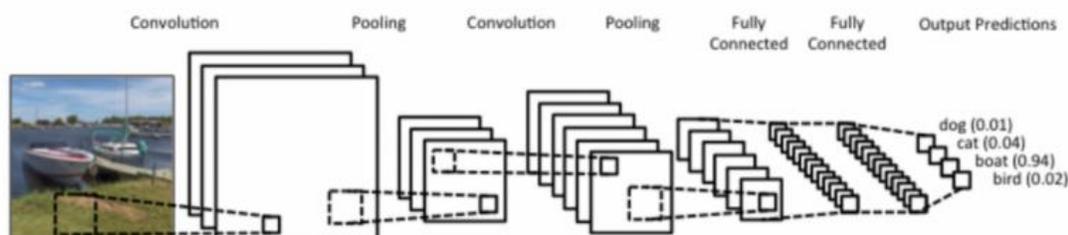
Redes Neuronales Convolucionales (CNN)

Este es un algoritmo de Deep Learning que obtiene una imagen de entrada y se enfoca en ciertas características dentro de esta (Lawrence, Lee Giles, Chung Tsoi, &

Back, 1997) (Blunsom, Grefenstette, & Kalchbrenner, 2014) (Aguilar, et al., 2018) (Aguilar, Quisaguano, Alvarez, Pardo, & Zahira, 2018). Esta arquitectura es similar a la conexión neuronal del cerebro humano, está inspirado por la corteza visual. En la Figura 9 se encuentra un ejemplo de arquitectura ConvNet que clasifica una imagen de entrada en cuatro categorías. Dependiendo de la predicción de salida, el algoritmo es calificado como correcto o no.

Figura 9

Esquema de una Red Convolutiva (Britz, 2015)



Nota: En la figura se muestra el esquema que tiene una red convolutiva para para realizar la identificación desde una imagen hasta la respuesta entregada.

Una CNN puede capturar dependencias espaciotemporales en una imagen por medio de filtros (Mosquera Victoria & Dussan Narvaez, 2020) (Togashi, Gian Do, Nishio, & Yamashita, 2018). Esta arquitectura se ajusta mejor a un dataset con imágenes, ya que disminuye los parámetros involucrados y los pesos de las neuronas reusables. Una neurona convolutiva puede dividirse en operaciones como; convolución, ReLU, Agrupación (pooling) y clasificación. Además, las CNN presentan una arquitectura multicapa, en donde cada una posee un número determinado de convoluciones con una función de activación lineal.

El procesamiento de este tipo de redes se basa en una convolución, en donde cada pixel de salida es una combinación lineal de los pixeles de entrada. La convolución es una operación en tiempo continuo, la cual está definida como la integral del producto de las funciones después de que una de ellas es desplazada en una distancia t . Esta operación es aplicable a matrices y es utilizada normalmente como filtros para crear máscaras, contornos y en general busca ubicar segmentos de interés en una imagen (Saldarriaga Quintero, 2020)

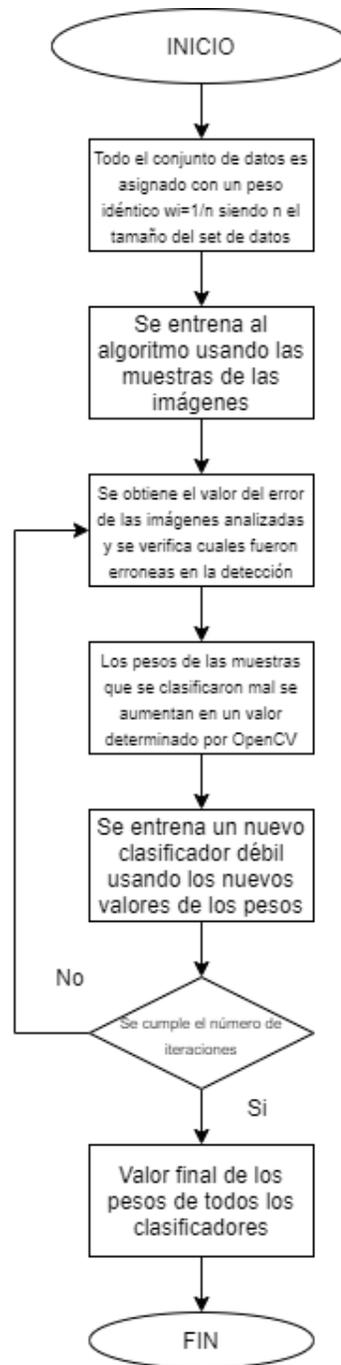
AdaBoost

Es un clasificador débil de tipo Boosting, desarrollado por Y. Freund y R.E. Schapire (Freund & Schapire, 1996) (Shen & Bai, 2004) (Aguilar W. G., y otros, Pedestrian Detection for UAVs Using Cascade Classifiers and Saliency Maps, 2017) (Aguilar W. G., y otros, Pedestrian detection for UAVs using cascade classifiers with meanshift, 2017). Este algoritmo utiliza dos conjuntos de imágenes, en el primer grupo se encuentra el objeto a detectar, mientras que en el otro no se encuentra dicho objeto, siendo las primeras muestras positivas y las otras muestras negativas.

Este algoritmo es iterativo, por lo que tiende a repetir un mismo procedimiento varias veces hasta lograr la meta deseada, lo que implica que se requiere un procesamiento muy pesado y en ciertos casos toma mucho tiempo (Wang, 2012). Normalmente este tipo de procedimientos requiere de la supervisión de algoritmos más robustos para cerciorarse de que el funcionamiento sea correcto. Una de las ventajas que posee este tipo de clasificadores es que es capaz de corregir los errores de sus iteraciones anteriores, lo que significa una mejor clasificación. En la Figura 10 se puede visualizar el funcionamiento del algoritmo de AdaBoost.

Figura 10

Esquema de entrenamiento de AdaBoost (Soto Espinoza, 2018)



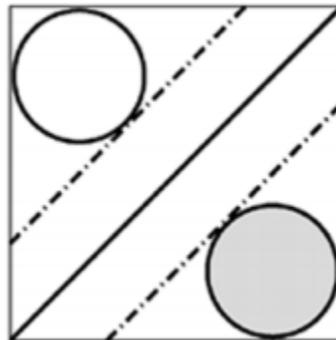
Nota: En la figura se puede observar cómo se va desarrollando el entrenamiento del algoritmo.

Máquinas de Vectores de Soporte (SVM)

También denominada Support Vector Machines. Es un clasificador que utiliza los descriptores de características del histograma de gradientes orientados para realizar el entrenamiento (Murillo Mejía & Jiménez Arboleda, 2019). El algoritmo trabaja en el reconocimiento de patrones y realiza una detección binaria. Separa los datos como positivos y negativos por medio de una línea divisoria, así como se observa en la Figura 11. El primer plano se encuentra bajo esta línea y sus datos son considerados positivos, caso contrario aquellos que se encuentran sobre la línea, los cuales son los datos negativos.

Figura 11

Hiperplano del SVM (Waring & Liu, 2005)



Nota: Se observan un hiperplano con dos regiones y una línea que las separa.

Los vectores de entrada serán considerados en un espacio n-dimensional (Vishwanathan & Murty, 2002). Subsecuentemente se construye un hiperplano en donde se separan las dos regiones de datos y se traza una separación. Los vectores de soporte son los que determinan la distancia entre los planos positivo y negativo. Mientras mayor sea la distancia de separación mejor será la detección.

JavaScript y TensorFlow

JavaScript es un lenguaje de programación de secuencias de comandos de eventos, clases y acciones para el desarrollo de aplicaciones en internet de cliente y servidor (Ayoze Castillo, 2017) (Flanagan, 2013). Permite trabajar con elementos dinámicos, lo que significa que se obtiene una experiencia interactiva y muy práctica, que permite la gestión y el manejo de datos.

Este lenguaje fue creado por Netscape y actúa directamente en páginas HTML (Goodman, 2002), permitiendo la creación de aplicaciones familiares a los CGI. Originalmente se lo planteo solo para navegadores web y que los usuarios pudieran interactuar con los comandos de ejecución.

JavaScript posee propiedades importantes (Ribes Alba, 2011) que lo diferencian de los otros tipos de lenguajes como:

- El ordenador interpreta lo que recibe del programa, este no compila el algoritmo.
- Es una programación orientada a objetos, siendo el código de estos predefinido y expandible, además, no usa clases ni herencias.
- El código es incluido en el HTML y trabaja con sus elementos.
- No requiere declarar ningún tipo de variable.
- Su ejecución es dinámica, los programas y funciones no se revisan hasta que se ejecutan.
- Los programas se ejecutan cuando ocurre algún evento.

TensorFlow es una librería de código abierto que fue desarrollada por Google para construir y entrenar redes neuronales. Su arquitectura es flexible y puede ser ejecutada por múltiples plataformas como GPUs, CPUs o TPUs, en clústeres de servidores, ordenadores y dispositivos móviles. Esta librería también puede aprovechar los recursos de las tarjetas gráficas como Nvidia para acelerar cualquier proceso (Conde Ortiz, 2018).

Los tensores utilizados por esta librería son objetos geométricos que describen relaciones entre vectores geométricos, escalares y otros tensores, es decir, estos objetos son los que opera la red para obtener valores (Nian, Zhang, Zhao, Qu, & Yuan, 2017). Así, cada nodo de grafo tomará como entrada un 0, 1 o más tensores y producirá un tensor de salida que ira transmitiéndose a otros nodos. Para ingresar la información a la red será necesario el uso de variables, las cuales serán siempre fijas o parámetros de sustitución, y se podrán ir modificando cada vez que se usa la red. En TensorFlow también, se permite el uso de librerías numéricas avanzadas para representaciones de valores como son NumPy y Pandas (Shukla & Fricklas, 2018).

Así también, TensorFlow posee todo lo necesario para crear redes neuronales de calidad, siendo una de las librerías más utilizadas a nivel mundial. Es usado por empresas como AMD, Nvidia, Intel, Airbnb, Twitter, Qualcomm, EBay, Xiaomi, etc. (Google, 2020)

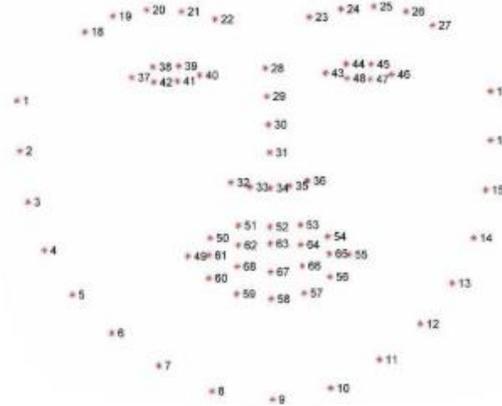
Faceapi

Es un módulo de JavaScript que es construido sobre TensorFlow, este funciona en base a CNNs optimizado para servidores web y para dispositivos móviles (Carleto, 2018) (Mühler, 2020).

El algoritmo necesita de 68 hitos faciales para el reconocimiento del rostro, para lo cual utiliza dos modelos ligeros y rápidos, el `face_landmark_68_model`, el mismo que requiere solo de 350kb y el `face_landmark_68_tiny_model` que solo usa 80kb. Los dos modelos emplean las ideas de convoluciones separables en profundidad, así como bloques densamente conectados. Estos dos modelos han sido entrenados por medio de un conjunto de 35 mil imágenes. Los puntos de referencia tomados en este modelo para la detección del rostro son de las cejas, ojos, nariz, boca y mandíbula, como se observa en la Figura 12.

Figura 12

Distribución de los puntos de referencia del rostro (Rosebrock, 2017).



Nota: En la figura se puede observar cómo se expresan los puntos de referencia para el funcionamiento del algoritmo.

Dependiendo de la aplicación a la que va dirigido Faceapi, puede hacer uso de dos modelos más, AgeGenderNet.ts, que puede estimar la edad y el género, mientras que FaceExpressionNet.ts reconoce 7 expresiones faciales, entre las que están felicidad, tristeza, enojo, miedo, disgusto, sorpresa y neutralidad (Nalini & Karan, 2017).

Para el reconocimiento facial se utiliza una arquitectura similar a la ResNet-34 que permite el cálculo de un descriptor facial a partir de una imagen ingresada. Este modelo no se limita al conjunto de rostros que han sido usado para el entrenamiento, por lo que se puede realizar la detección de cualquier persona. Se determina la similitud de rostros entre el modelo ingresado con la imagen a detectar por medio de los descriptores de la fisonomía de su cara.

Por otro lado, Faceapi.js está compuesto por tres etapas (Saldarriaga Quintero, 2020): la primera es la detección del rostro utilizando SSD (Single Shot Multibox Detector), el mismo que está basado en la red MobileNetV1, añadiendo un algoritmo que

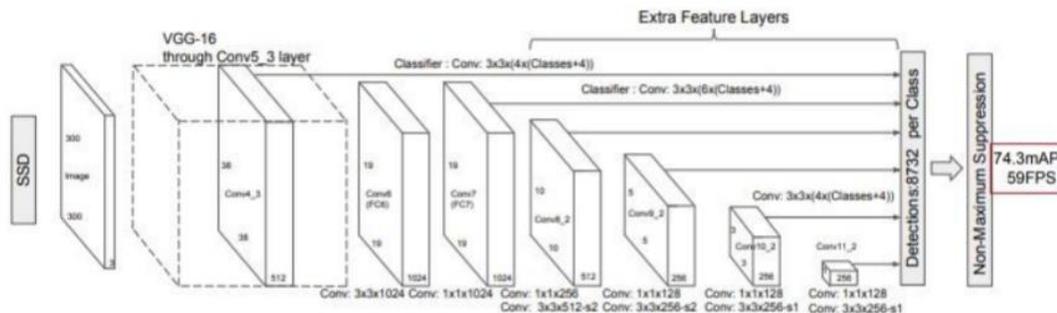
por medio de capas apiladas en la parte superior de la red predice la ubicación del rostro en la imagen que se ingresó. La segunda parte consiste en la ubicación de los puntos de referencia en el rostro a través del modelo antes mencionado. Finalmente, se procede al reconocimiento facial por medio de la arquitectura ResNet-34. Se usa la distancia euclidiana entre los descriptores de los dos rostros que se comparan para determinar si es o no el mismo sujeto.

Single Shot Multibox Detector (SSD)

Este algoritmo tiene como característica principal entregar en un solo paso la imagen de entrada a la red. El SSD añade varias capas de características al final de la base de la red neuronal, estas tienen como objetivo predecir los valores que cada objeto detectado entrega por defecto. En la Figura 13 se observa la arquitectura de esta red, con una secuencia de entrada de 300 x 300 (59 fps) y una precisión de 74.3% (Liu, y otros, 2016) (Yi, Wu, & Metaxas, 2019).

Figura 13

Capas del algoritmo SSD (Pérez Nasser, 2019)



Nota: En la figura se puede ver la estructura que tiene el algoritmo SSD.

Este detector tiene un buen equilibrio entre velocidad y precisión al momento de ser utilizado (Konishi, Hanzawa, Kawade, & Hashimoto, 2016). En primer lugar, ejecuta una red convolucional sobre la imagen de entrada y realiza un mapeo de características,

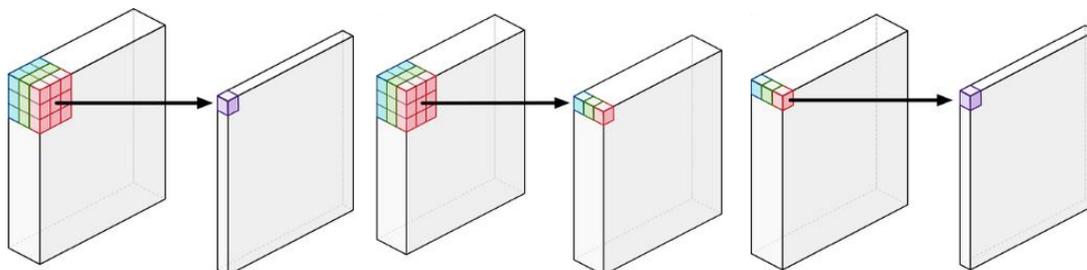
prosigue con la ejecución de un pequeño núcleo convolucional en el mapa para lograr predecir los cuadros delimitadores y la probabilidad de clasificación. El SSD obtiene una predicción por medio de múltiples capas convolucionales (Parreño Escobar, 2020)

MobileNetV1

Estos modelos están dirigidos a sistemas embebidos y dispositivos móviles. Su concepto se basa en convoluciones separables en profundidad (Caiza, Aguilar, Albán, & Fernández, 2020). Si la convolución regular combina todos los canales dentro de una sola entrada, la convolución en profundidad procede a operar cada canal por separado, lo que implica que cada canal obtiene su respectivo conjunto de pesos (Sheng, y otros, 2018). Al filtrar cada canal se puede obtener filtros de color, detectores de bordes y otras características. Se puede visualizar este concepto en la Figura 14, en donde a la izquierda se observa la convolución típica, la cual a la salida de los tres canales tiene un solo punto. En medio de la ilustración está la convolución separada en profundidad, donde cada convolución tiene su propio canal de salida, en este caso cada salida tiene su propia operación. Finalmente, a la izquierda se encuentra una convolución en un solo punto, con una profundidad igual a los canales de la imagen, la salida se centra en un solo punto como la convolución típica. La principal ventaja de utilizar este tipo de convolución es su eficacia computacional, ya que se requiere menor cantidad de matemática que las convoluciones en 2D (Yao, Zhang, Wu, & Lin, 2020).

Figura 14

Izquierda: Convolución típica, Centro: Convolución profunda, Derecha: Convolución puntuada o 1x1 (Mosquera Victoria & Dussan Narvaez, 2020)

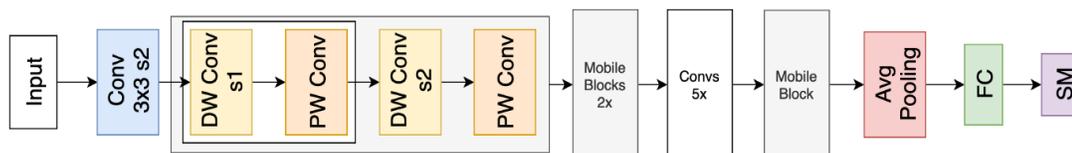


Nota: En esta figura se puede ver la estructura que toma una convolución típica.

Con respecto a la arquitectura, esta consta de una capa convolucional, 3 bloques de Mobile, que se dividen en una capa de convolución separable profunda (DW), una capa de convolución de punto (PW), otra capa DW y una nueva capa PW. Después se encuentran 5 bloques más de convolución, conectados a una capa Mobile, para finalizar se encuentra con capas de agrupamiento y clasificación. Esta división se aprecia en la Figura 15.

Figura 15

Arquitectura MobileNet v1 (Mosquera Victoria & Dussan Narvaez, 2020)



Nota: En la figura se puede ver la arquitectura que maneja MobileNet desde la entrada de datos hasta la respuesta entregada.

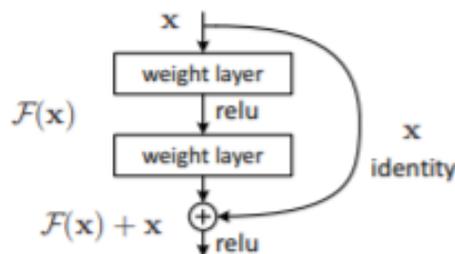
Estas operaciones por medio de convoluciones profundas separables permiten una precisión del 70.6% en operaciones y además son capaces de ejecutar a 20 frames por segundo en un teléfono inteligente (Howard A. G., y otros, 2017).

Redes Residuales (ResNet)

Las redes residuales son un tipo de modelo utilizado para varias tareas de visión por computadora, como por ejemplo extracción de características (Targ, Almeida, & Lyman, 2016) (Wu, Shen, & van den Hengel, 2019). Fue introducida por Microsoft en el año 2015, siendo el aumento del número de capas, el principal objetivo de la arquitectura, de manera que introduce una conexión residual, como se observa en la Figura 16, esto permitió entrenar redes muy profundas (de más de 100 capas), controlando con éxito el problema de desvanecimiento de gradiente, que a la vez permite mejorar el proceso de aprendizaje.

Figura 16

Diagrama de bloques de ResNet (He, Zhang, Ren, & Sun, 2016)



Nota: En la figura se puede observar cómo está representado el funcionamiento de ResNet en un diagrama de bloques.

Este tipo de red obtiene una mejor respuesta a una mayor profundidad, dado que mientras se obtenga mayor cantidad de capas (más características de la imagen), se procesan más datos y se incrementan los escenarios que analiza la red neuronal (Jiang & Lu, 2018). También se disminuyen los errores de entrenamiento y de las validaciones

en los resultados de las pruebas del sistema. Su base matemática se puede visualizar en Kaiming He (He, Zhang, Ren, & Sun, 2016) o Joaquín Justel (Pizarro Justel, 2017).

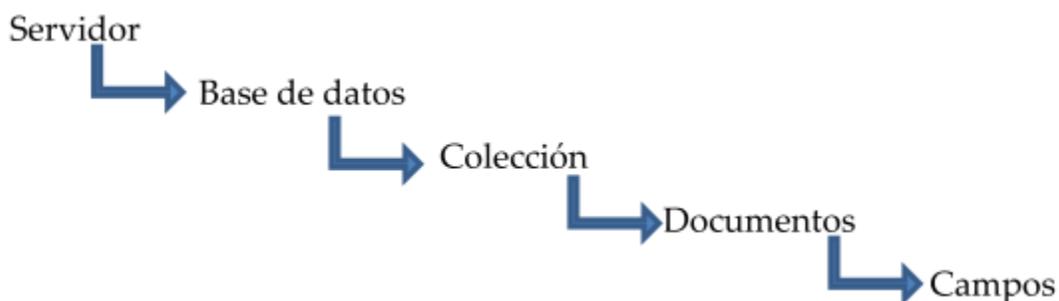
Base de Datos: MongoDB

MongoDB es una base de datos de tipo no relacional (NoSQL,) multiplataforma de licencia libre (MongoDB, 2020). Se encuentra orientada a la recopilación de información de esquema libre, lo que implica que cada registro puede tener un esquema de datos diferente. Es una solución encaminada a la mejora de escalabilidad horizontal de la capa de datos, con un desarrollo sencillo y con una capacidad de almacenamiento mayor. Cuando se requiere que muchas maquinas escalen, el enfoque no relacional es el más eficiente. El modelo de datos (JSON/BSON) brinda facilidades de manejo y alto rendimiento mediante la agrupación de datos relevantes entre sí.

En MongoDB, la información que ha entrado a la base de datos, se la denomina documentos, los mismos que pueden ser agrupados en colecciones, que son equivalentes a las tablas de las bases de datos relacionales, pero que no son sometidos a un esquema fijo. En la Figura 17 se visualiza la jerarquía que posee MongoDB, en donde un conjunto de campos formará los documentos, y si estos se asocian formaran una colección y sucesivamente. La diferencia con otro tipo de sistemas gestores de base de datos relacionales, es que no se utiliza ni tablas, ni filas, ni columnas.

Figura 17

Arquitectura funcional de Mongo DB (Graterol, 2014)



Nota: En la figura se observa cómo funciona la base de datos de Mongo DB.

La información que ingresa a este tipo de base de datos también se puede relacionar entre sí pese a no utilizar un esquema estándar como en SQL. La mayoría de las veces se trabaja con documentos estructurados que no necesariamente seguirán un mismo esquema, sino que cada uno podrá tener uno propio y que resulta apropiado.

Para el intercambio de datos en el almacenamiento y la transferencia de documentos se utiliza el formato BSON (Binary JavaScript Object Notation), el cual es una representación binaria de estructuras de datos y mapas, diseñado para ser más ligero y eficiente que JSON (JavaScript Object Notation).

MongoDB cuenta con herramientas útiles para trabajar desde diferentes perspectivas o propósitos, entre las que se menciona (Bellido, 2011):

- Mongod: Servidor de bases de datos de MongoDB.
- Mongo: Cliente para la interacción con la base de datos MongoDB.
- Mongofiles: Herramienta para trabajar con ficheros directamente sobre la base de datos MongoDB.

Cabe mencionar que las bases de datos no relacionales se destacan por su alta escalabilidad horizontal, que supone una adaptabilidad para cualquier sistema que

requiera una gran cantidad de información para funcionar (Ioana Agapin, Radulescu, & Boicea, 2012). MongoDB presenta una arquitectura adecuada para la ejecución de esta tarea, además de presentar un método para dividir los datos entre múltiples servidores que pudieran presentar una solución escalada: el Sharding. La principal característica de este método es su ejecución de forma automática, lo que permite una ágil distribución de datos entre todos los equipos que trabajan en el sistema (Gargouri & Abbas, 2016).

Firestore

Es una plataforma de desarrollo de aplicaciones web. Desarrollada por Google en 2014, está integrada con “Google Cloud platform”, que sirve para la creación y sincronización de proyectos de alta calidad. (Cardona, 2016)

Gracias a su facilidad de implementación y rapidez se evitó la creación de una infraestructura compleja, aparte de la integración para APIs como es el caso de JavaScript. Entre una de las características más importantes a manejar es el “Realtime Database” (Moroney, 2017).

Esta plataforma tiene una base de datos que funciona en tiempo real, y que se encuentra organizada en formato de árbol JSON, lo cual permite su ejecución en una API que admite el intercambio de información de manera rápida y eficiente. Los datos ingresan de forma sincronizada y son almacenados en una nube de Firestore. El sistema tiene integraciones para varios sistemas de JavaScript como son:

- AngularJS
- React
- Ember.js
- Backbone.js

La sincronización en tiempo real permite la interacción del usuario desde cualquier dispositivo a la información dentro de la base de datos. En el caso de que un usuario varíe los datos, los cambios se visualizaran simultáneamente en los demás dispositivos conectados al servidor y que tengan el respectivo acceso (Huang, Tung, Lin, Yen, & Li, 2018).

React

React es una librería de JavaScript que está centrada en el desarrollo de interfaces de usuario o front-end (Alvarez, 2019). Esta librería es idónea cuando se trata de realizar todo tipo de aplicaciones web o móviles, debido a que React se compone de un completo ecosistema de módulos y componentes que ayudan al desarrollador a cubrir objetivos avanzados con poco esfuerzo.

Figura 18

Logo React. (Boduch, 2017)



Nota: En la figura se muestra el logo perteneciente al framework de javascript React.

React es creado con el objetivo de desarrollar aplicaciones más ordenadas y con la menor cantidad de código posible. Si el enfoque de la aplicación lo requiere, se puede hacer uso de JavaScript centrado en manipulación del DOM, para permitir que todo se encuentre asociado, de manera que, si cambian los datos, la vista del usuario también se modifica (Danielsson, 2016). A diferencia de otras librerías, React genera el cambio de la página con respecto a los datos, esto de forma automática y sin tener que generar un

nuevo código. La librería se maneja a través de “Virtual DOM”, donde, si se actualiza una vista, se actualiza este último y el intercambio de datos se consigue con mucha más rapidez que con el DOM propio del navegador, por lo que el entorno de trabajo y visualización se vuelve más potente, lo que significa mayor rendimiento (Boduch, 2017).

Una de las características particulares de React es que es isomórfico, por lo que trata solo con un mismo código, renderiza el HTML tanto en el servidor como en el cliente, y rebaja la carga significativamente de trabajo. Con esto se puede realizar aplicaciones web amigables, desacoplando el desarrollo del lado del servidor y el del cliente (Eisenman, 2016).

Capítulo III

Desarrollo de Las Gafas Espía

Diseño de Hardware.

Todos los dispositivos del sistema fueron escogidos para caber en unas gafas de tamaño promedio, es decir, los elementos eran considerablemente pequeños. El peso del dispositivo también debía ser ligero, con la finalidad de conferir comodidad al usuario, que podría utilizar las gafas en prolongados lapsos de tiempo. Este dispositivo contó con: un sistema háptico compuesto por dos motores para celular, un módulo ESP32-CAM, una batería Lipo, un interruptor pequeño y un conjunto de cables que funcionaban como buses de datos y alimentación. Todos los elementos en conjunto tuvieron un peso de 80g, este valor fue obtenido mediante el uso de una balanza analógica, como se observa en la Figura 19.

Figura 19

Peso del diseño final de las gafas



Nota: En la figura se observa el peso final que se midió por medio de una balanza de cocina usada para medir cantidades exactas de preparaciones culinarias.

El hardware del sistema poseía dos tipos de buses, el primero es el de alimentación, que permitió la conexión entre el módulo ESP32-CAM con la batería, tal como se observa en la Figura 20. También se puede observar que tenía implementado un interruptor que permitía o no el paso de corriente. Cabe recalcar que la batería no se encontraba soldada al interruptor, por lo que puede ser removida para ser cargada.

Figura 20

Bus de alimentación de las gafas espía

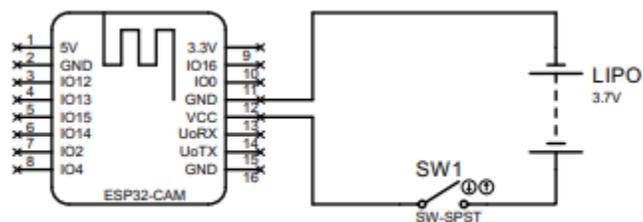


Nota: En la figura se observa una representación del modelo final de las gafas espía.

En la Figura 21, se denotan todos los pines que fueron utilizados para conectar el módulo con la batería.

Figura 21

Diagrama esquemático de la alimentación del sistema. (Mechatronics, 2020)



Nota: En la figura se muestra un diagrama de conexión del sistema para el encendido y apagado de las gafas.

El bus de datos constó de dos partes, en la Figura 22 se observa que los cables rosado, azul claro, amarillo, celeste, verde oscuro y morado que van hacia la varilla derecha de las gafas, se encuentran desconectados, esto es debido a que son los cables de programación. El cable celeste tenía la función de transmisión de datos, mientras que el verde oscuro la recepción, por otro lado, el cable azul claro era la alimentación a 5 voltios y el rosado era la conexión a GND, todos los cables anteriormente mencionados iban conectados directamente al módulo FTDI. Finalmente, el cable amarillo habilitaba o no la programación del dispositivo, para esto era necesario que sea conectado a GND directamente, por lo que se colocó un jumper que realice el cortocircuito entre los dos cables.

Los cables rojo y azul estaban conectados a la varilla derecha, mientras que los cafés y verdes se conectaron a la izquierda, habilitando o no el sistema háptico en ambas varillas.

Figura 22

Bus de datos de las gafas espía

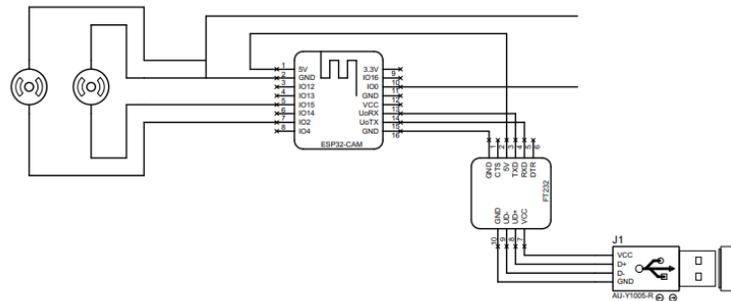


Nota: En la figura se muestra una representación de la conexión usada para el sistema háptico montado en las gafas espía.

En la Figura 23, se puede observar cómo eran las conexiones para la programación del módulo ESP32-CAM y que pines se debía colocar cada motor del sistema háptico. Es notable que, en el diagrama, el PIN 11 se encuentra desconectado, esto se debe a que tiene que hacer cortocircuito con tierra para habilitar la programación, este procedimiento fue realizado manualmente por medio de un jumper. El GND del PIN 2 fue dividido para la conexión del sistema háptico y la habilitación de la programación.

Figura 23

Diagrama esquemático del bus de datos



Nota: En la figura se muestra el diagrama esquemático correspondiente a las conexiones para el modo de programación conjunto con el programador y la conexión eléctrica para el sistema háptico.

Construcción del Sistema

En primer lugar, se utilizaron unas gafas de plástico de color negro, tal que, en caso de ser necesario, se podía adaptar su diseño debido a la flexibilidad de su material. De acuerdo con los requerimientos del sistema, era necesario que el cableado y elementos pasen desapercibidos una vez se integren a las gafas, por lo que se procedió

a manipular una lente y las varillas internas y externas. En las varillas internas se utilizó un cautín a altas temperaturas, lo que permitió fundir el plástico y crear canales en donde pase todo el cableado como se ve en la Figura 24. También se perforó la varilla externa del lado izquierdo para que pueda ingresar el interruptor que permitía o no el paso de corriente desde la batería. Este último fue colocado cerca de la batería para evitar un mayor cableado como se observa en la Figura 25.

Figura 24

Canales internos en las varillas de las gafas



Nota: En la figura se puede ver una foto de los canales que se realizaron al interior de las gafas para mejorar la eficiencia del paso de los cables a través de las gafas.

Figura 25

Batería e interruptor colocados en la varilla externa del lado izquierdo de las gafas

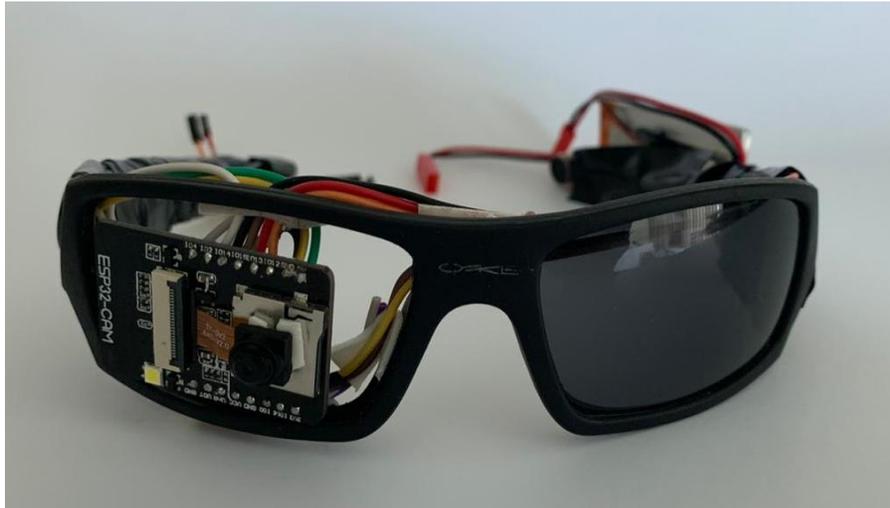


Nota: En la figura se muestra la implementación del cableado montado en el prototipo.

Una vez que se rediseño todo el cuerpo de las gafas, se procedió a desoldar los pines de conexión del ESP32-CAM. Esto permitió colocar al módulo en cualquier lugar del armazón evitando molestias en el usuario. En este trabajo se escogió colocarlo en lugar del lente derecho, como se ve en la Figura 26, debido a que el bus de datos de la cámara OV2640 no era de tamaño grande (un centímetro aproximadamente). Pese a que se perdió gran parte de la visibilidad del ojo derecho, el sistema continuaba brindando comodidad y permitía el paso de cables con mayor facilidad.

Figura 26

Vista frontal de las gafas sin el lente derecho



Nota: En la figura se muestra una vista frontal de las gafas espía sin ningún lente que oculte el módulo y la visualización de la cámara.

Figura 27

Vista frontal de las gafas colocado el lente derecho



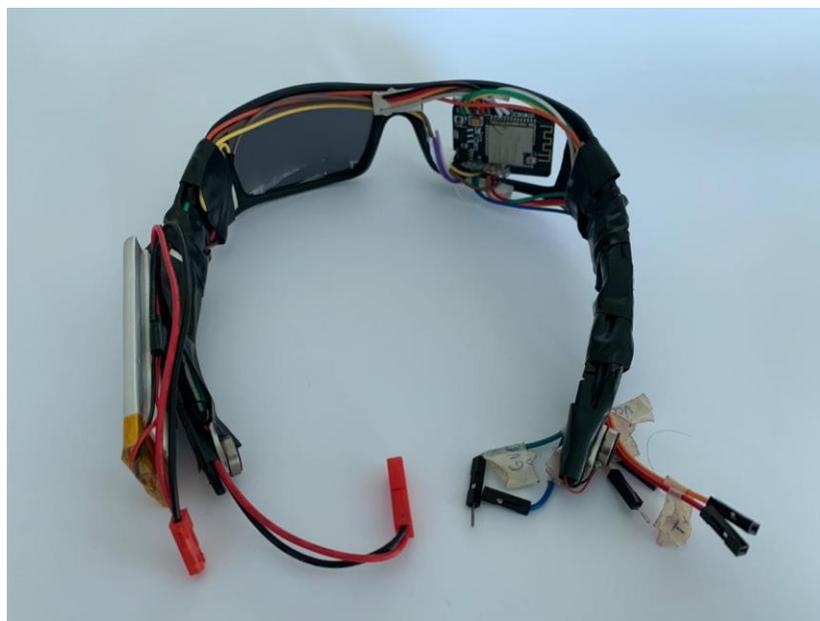
Nota: En la figura se puede observar una vista frontal de las gafas con su diseño final, con la lente montada encima lo que oculta a la vista la cámara y lo hace pasar en su mayoría desapercibido.

Como se observó en la Figura 22 y Figura 20, 8 de 12 conexiones se dirigieron a la varilla derecha de las gafas, con esto se evitó un mayor cableado. Las únicas 4 conexiones en el lado izquierdo son las dos de alimentación y dos del sistema háptico.

Todo el cableado pasó de un lado a otro a través del armazón, específicamente por la parte superior, y en el caso de los cables de alimentación, también fue necesario recorrerlos por la parte interior de los soportes nasales, para evitar molestias al usuario. El cableado se observa en la Figura 28.

Figura 28

Vista superior de las gafas



Nota: En la figura se muestra una vista desde la parte superior de las gafas en su prototipo final.

Finalmente se colocó la batería y el sistema háptico. El sistema háptico se instaló en cada terminal de las gafas, mientras que la batería en la varilla izquierda.

Gasto Energético

Como se puede observar en las siguientes tablas, el consumo energético de todo el sistema en su máxima capacidad era alrededor de 315 mA. Tomando en cuenta cualquier eventualidad que se pudiera dar con respecto al arranque de los motores del sistema háptico, se escogió una batería de polímero de litio de 3.7 V y de 900 mAh (Battery, 2006). Esta batería consta de las características especificadas en la Tabla 5.

Tabla 5

Especificaciones técnicas de la batería Lipo. (Battery, 2006)

Modelo	402770042770
Capacidad	900mAh
Tensión nominal	3.7 V
Voltaje de Carga	4.2 V
Voltaje de protección	Sobrecarga: 4.2 ± 0.05 Sobredescarga: 2.7 ± 0.1
Temperatura de Funcionamiento	-20°C a 60 °C

Nota: en la tabla se muestra las características de funcionamiento que cuenta la batería.

Configuración de Enlace Entre Dispositivos.

La comunicación de enlace entre los dispositivos se realizó por medio de Wi-Fi a 2.4 ghz, mediante el módulo implementado en la ESP32 CAM, donde el envío de datos era esencial en el prototipo. Cabe recalcar que el módulo de esta cámara es un módulo

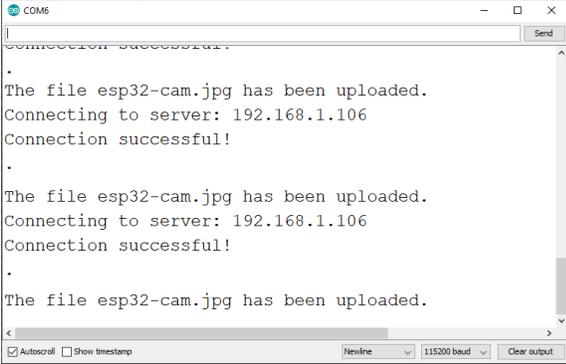
programable, que se pudo configurar mediante la IDE de Arduino y que usa una programación en el mismo lenguaje, C y C++.

Después de iniciar con la configuración de la tarjeta, se ajustó el hardware para que pueda programar, es decir que colocó el jumper entre el pin de programación y GND. Se procedió a realizar pruebas con el ejemplo preprogramado de cámara web server, ya existente en Arduino, que hace uso de librerías como Wifi.h y soc/soc.h que permiten el intercambio de datos con un servidor y el establecimiento a una conexión con el internet.

Para lograr la comunicación externa se hizo uso de “HTTP POST/GET” método usado para envío y recepción de datos en páginas web. Debido a que el sistema funcionaba en un servidor local, se establecieron puertos de comunicación independientes para el envío de imágenes, así como para el procesamiento y análisis de la información obtenida de estas, teniendo una comunicación con el front-end para la interfaz de usuario y con el servidor para el reconocimiento e identificación de personas. Debido a la naturaleza en la que trabaja este módulo.

Figura 29

Ejemplo de envió de datos al Servidor



```
COM6
Connection successful.
.
The file esp32-cam.jpg has been uploaded.
Connecting to server: 192.168.1.106
Connection successful!
.
The file esp32-cam.jpg has been uploaded.
Connecting to server: 192.168.1.106
Connection successful!
.
The file esp32-cam.jpg has been uploaded.
```

Autoscroll Show timestamp Newline 115200 baud Clear output

Nota: En esta figura se muestra un ejemplo del monitor serie del IDE de Arduino mostrando que la conexión con el internet y el servidor fue correcta y que la imagen fue enviada por la cámara satisfactoriamente.

El envío de imágenes se lo realizó a un tamaño de SVGA, el cual era suficiente para la aplicación propuesta, además, al tratarse de un sistema de reconocimiento, se propuso que trabajara lo más cercano a tiempo real posible, por lo que se lo configuró a un intervalo de envío de 2 frames por segundo, y también se habilitaron algunas salidas digitales, donde, a través de los datos de coordenadas recibidas por el servidor, se ajustó el sistema háptico para poder centrar el objetivo a la cámara de las gafas espía y obtener la mejor calidad de reconocimiento.

Especificaciones del Servidor.

El servidor en donde se receptaron las imágenes que fueron captadas por la cámara, para su posterior análisis y predicción, tenía una velocidad de respuesta dependiendo de las especificaciones del hardware donde esté montado. Ya que se necesitaba robustez en el procesador, se implementó el proyecto en una laptop “Msi GE62VR 7RF Apache Pro”, perteneciente a una gama media alta de computadores portátiles, debido a que el sistema está hecho para ser versátil y portátil no se puede montar el servidor en un computador de sobremesa. Las especificaciones que tiene el sistema se muestran en la siguiente tabla.

Tabla 6

Especificaciones Servidor. (MSI, 2020)

GE62VR 7RF Apache Pro	
SO	Windows 10 Home
PANTALLA	15.6" FHD (1920x1080), IPS-Level 15.6" UHD (3840x2160), IPS-Level
PROCESADOR	Intel i7-7700HQ 2,8 Ghz(Turbobost 3,8 Ghz)

GRAFICOS	GeForce® GTX 1060 with 6GB GDDR5
MEMORIA	DDR4-2400, 2 Slots, Max 32GB,
ALMACENAMIENTO	128 GB x M.2 SSD Combo (NVMe PCIe Gen3 / SATA) SSD 1 TB x 2.5" SATA HDD
UNIDAD OPTICA	DVD Super Multi
WEBCAM	HD type (30fps@720p)
COMMUNICATION	Killer Gb LAN 802.11 ac Wi-Fi + Bluetooth v4.2
I/O PORTS	1x Type-C USB3.1 Gen2
	1x Type-A USB2.0
	1x RJ45
	1x SD (XC/HC)
	1x (4K @ 30Hz) HDMI
	1x Mini-DisplayPort
BATERIA	51 Whr
CARGADOR	180W adapter

Nota: En la tabla se muestran las especificaciones de la computadora donde se montó el servidor para que el reconocimiento sea lo suficientemente rápido y no entregue ningún problema por parte de procesamiento.

Capítulo IV

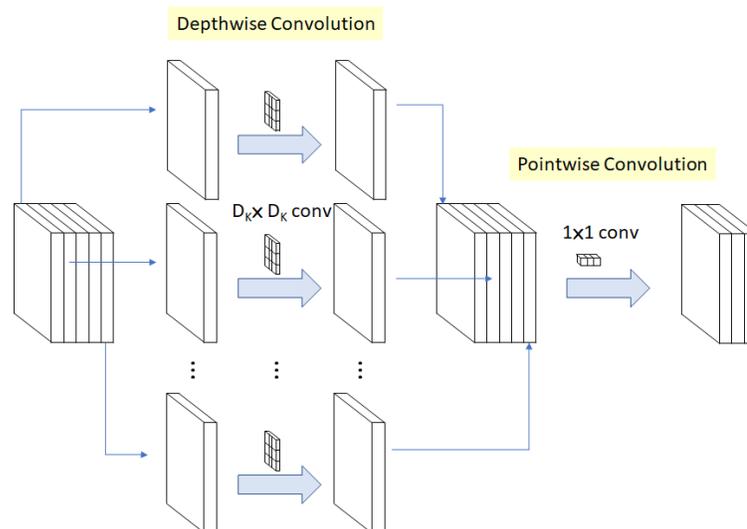
Desarrollo del Algoritmo

El algoritmo base que se usó lleva una arquitectura de MobileNetV1, este modelo usa la convolución separable en profundidad, utilizado para reducir el tamaño y complejidad del modelo a ingresar en la red neuronal. (Caiza, Aguilar, Albán, & Fernández, 2020)

Esta convolución es una convolución profunda seguida de una convolución puntual como se muestra en la figura.

Figura 30

Diagrama de convolución separable en profundidad MobileNetv1 (Howard A. , y otros, 2017)



Nota: En la figura se muestra el proceso que el algoritmo de convolución lleva a cabo para entregar una respuesta, dentro de MobileNetV1.

Para este sistema se tiene la convolución en profundidad en $D_k \times D_k$ por canal, y la puntual es la salida del sistema con una convolución de 1×1 , como este es un sistema en particular útil para aplicaciones de visión integradas y móviles. Teniendo en cuenta

esto se tiene que el costo de operación para un ejemplo de 5 canales, con 5 convoluciones espaciales $D_K \times D_K$, se muestra la ecuación:

$$D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F \quad (1)$$

Donde M es el número de canales de entrada, N el número de canales de salida, D_K es el tamaño del núcleo y D_F el tamaño del mapa de características (Howard A. G., y otros, 2017). La reducción del cálculo se reduce a:

$$\frac{D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F}{D_K \cdot D_K \cdot M \cdot D_F \cdot D_F} = \frac{1}{N} + \frac{1}{D_K^2} \quad (2)$$

La arquitectura de la red completa se muestra en la siguiente figura.

Figura 31

Arquitectura MobileNetV1. (Howard A. G., y otros, 2017)

Table 1. MobileNet Body Architecture

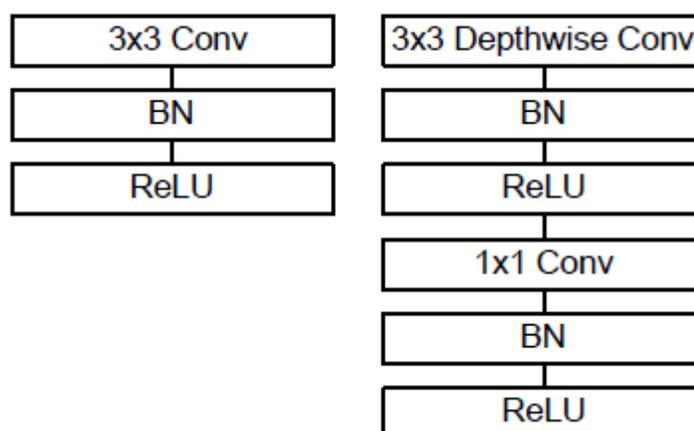
Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5 \times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Nota: En la figura se muestra la arquitectura que lleva a cabo las diferentes convoluciones de MobileNetV1.

Por último, se visualiza que la normalización por lotes (BN) y ReLU se aplican después de cada convolución como se muestra en la siguiente figura.

Figura 32

Diagrama de convolución de MobileNetV1. (Howard A. G., y otros, 2017)



Nota: En la figura se puede observar el diagrama de convolución de MobileNetV1, donde se tiene la convolución estándar en la parte izquierda y la convolución separable en profundidad con BN y ReLU en la parte derecha.

Diseño e Integración de la Base de Datos con el Algoritmo

Teniendo en cuenta que el algoritmo de reconocimiento poseía su propia red neuronal, y que no era necesario un dataset para el entrenamiento, la base de datos solo constó con una foto de cada una de las personas que se reconocieron. La información que ingresaba a la base de datos no requería un orden específico (Aguilar, Cobeña, Rodríguez, Salcedo, & Collaguazo, 2018), lo que implicó una mayor flexibilidad en su uso, por lo que se escogió la base de datos MongoDB, de carácter no relacional.

Otro parámetro que se tomó en cuenta es que se necesitaba una base de datos con una buena capacidad de almacenamiento, ya que se requería procesar una considerable cantidad de imágenes.

Cuando se ingresaba una nueva imagen dentro de la base de datos, el algoritmo extraía características que eran guardadas para una futura comparación. Los datos que se almacenaba eran: el nombre de la persona a la que pertenece la foto, los descriptores de la cara y finalmente la propia foto. Todo este código se encuentra dentro de los modelos del controlador del sistema.

En el back-end del sistema se definió que se trabajaría en un servidor local, asignándole el puerto 3001 a la base de datos. La configuración se encontraba en el programa principal y si la conexión era exitosa, enviaría a un terminal el mensaje de “Database Ready” como se observa en la Figura 33, en el caso de existir problemas entre la conexión de la base de datos y el servidor, el programa no iniciaba.

Figura 33

Mensaje del terminal con la carga completa de modelos y conexión con la base de datos

```
> node app.js
node-pre-gyp info This Node instance does not support builds for N-API version 4
node-pre-gyp info This Node instance does not support builds for N-API version 5
node-pre-gyp info This Node instance does not support builds for N-API version 4
node-pre-gyp info This Node instance does not support builds for N-API version 5
2021-01-03 21:29:35.861540: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your TensorFlow binary was not compiled to use: AVX2
Example app listening at http://localhost:3001
Database Ready
Loaded Models
```

Nota: En la figura se observa la respuesta del servidor cuando está listo para usar el prototipo.

Desarrollo de Algoritmo Para Clasificación de Datos y Reducción de Costo Energético.

Para el adecuado desarrollo del algoritmo fue necesario investigar todas las características que ofrecía face-api, entre las que se encontró diferentes modelos:

- `tiny_face_detector`, que determinaba la posición del rostro de una persona.
- `ssd_mobiletv1` y `mtcnn` las cuales eran parte de la red convolucional con la que trabaja esta librería
- `face_recognition` que permitía el reconocimiento de rostros.
- `face_landmark_68` que extrae los descriptores que caracterizan la fisionomía que ha sido detectada en la imagen.
- `age_gender` y `face_expression` que detectaron la edad, género y la expresión facial de la persona.

El proyecto propuesto no requería todos los modelos que se encontraban dentro de face-api, ya que, el objetivo no era reconocer características de las personas ajenas a su rostro y sus rasgos. Por lo tanto, los modelos que identificaban, expresiones, género y edad no formaron parte de este trabajo.

En una carpeta consiguiente se agrupó a todos los módulos y dependencias que han sido utilizadas. La principal dependencia era la que contenía todo lo relacionado a tensorflow, la versión 1.7.0, que permitía el procesamiento del sistema en el núcleo del computador. En el `package.json` del sistema también se encontraban todos estos módulos, los cuales, si se requería migrar el código, se utilizaba la función “`npm install`” que podía descargarlos desde otro computador por medio de una terminal.

El programa principal tenía como nombre `app.js`, en esta se definió todas las librerías que se utilizaban a lo largo del programa, entre estas estaban:

- *CANVAS*, esta permitió al desarrollador trabajar con código HTML, lo que implicaba la formación de una interfaz en un navegador web.
- *EXPRESS*, era una librería que facilitó el uso de múltiples frameworks web de Node. Entre sus características estaban la integración de los motores de renderización, manejo de peticiones en HTTP por medio de diferentes URLs, establecimiento de ajustes de aplicaciones web por medio de puertos para conectar, entre otras.
- *FIREBASE*, era una base de datos que nos permitía integrar el hardware y software del sistema por medio de la recepción de imágenes desde la cámara.
- *MONGOOSE*, habilitaba la base de datos, en donde se guardaron todas las imágenes de las personas que se deseaba identificar.
- *FACE-API*, era la librería que contenía todos los modelos y funciones que permitieron el reconocimiento de rostros en una imagen y la extracción de sus características.

Una vez se identificaron todas las características de las librerías, se procedió a definir los puertos por donde se levantó la base de datos y se procesó la información, en este caso fueron los puertos 3001 y 3000 respectivamente. Por lo tanto, se definió que el servidor iba a ser local y era levando por medio de una terminal. Cabe mencionar que se configuro como un servidor público.

Finalmente, se configuraron cada una de las librerías y se cargaron los modelos en su respectiva URL. Una vez todo el proceso se completó, la consola enviaba dos mensajes, el primero era "Loaded Models" que significa que los modelos se cargaron satisfactoriamente y el segundo fue "Database Ready", que indicaba que la base de datos se ha levantado satisfactoriamente.

Para poder programar ordenadamente, se dividió el código en controladores, modelos, rutas e interfaces (Aguilar, Salcedo, Sandoval, & Cobeña, 2017) (Aguilar, Angulo, & Costa-Castello, Autonomous Navigation Control for Quadrotors in Trajectories Tracking, 2017).

En la carpeta de interfaces se programó "function.js", que tenía como labor ir configurando los parámetros del código HTML, tales como, el ancho y el largo de las imágenes que ingresaban y el valor de string que poseía cada una de las fotos. En la carpeta de rutas se encontraba el código "faceRecognition.js" en donde se definieron las posiciones en donde se tiene que enviar o recibir datos. En primer lugar, se determinó la ruta del método post, por donde se envió los descriptores de las imágenes para una futura comparación, además de guardar el nombre de la persona a la que pertenece dicha imagen. La próxima ruta a configurar fue la de las imágenes que se iban a comparar con las ya guardadas en la base de datos, por lo que, se guardó solo una imagen enviada a una ruta especificada. En el caso de presentarse algún error en cualquiera de los métodos post, se enviaba un mensaje que se visualizó en la consola e indicaba que el proceso tuvo una falla. Estas rutas pueden ser comprobadas por medio del programa PostMan.

En la carpeta perteneciente a modelos, estaba la configuración correspondiente a la base de datos MongoDB, en el archivo de nombre "faceDescriptor.js", donde se guardaban las imágenes de las personas que se deseaban comparar, su arreglo de descriptores y el nombre de la persona a la que pertenecía dicha imagen. Finalmente, en la carpeta controladores estaba el programa "faceRecognitionController.js", donde se definió todas las acciones que requería el sistema para funcionar correctamente. En primer lugar, la imagen que se ingresaba en el HTML y que iba a ser guardada en la base de datos, tenía que ser comprimida, esto para disminuir el peso de la imagen. Una vez realizada esta operación, se procedió a extraer los descriptores de la imagen por medio de los modelos de la librería face.api. Se realizó un proceso similar con las imágenes

obtenidas de la cámara montada en el módulo ESP32-CAM, y una vez extraído los descriptores, se procedió a calcular la distancia euclidiana entre la imagen que fue guardada previamente y la que se había enviado ese momento. Si el valor de la comparación era pequeño, significaba que la persona que a la que se reconocía en ese momento, en frente de la cámara, era la misma que aquella almacenada en la base de datos. Es importante resaltar que además de los descriptores en la imagen proveniente del módulo en las gafas, también se extrajo las coordenadas de donde se encontraba el rostro, esta funcionalidad era importante para el sistema háptico. En la sección final del código, se obtuvieron de datos del módulo ESP32-CAM por medio de la base de datos FireBase, en donde, se envió dos frames cada segundo, y en el caso de existir un rostro en dicho frame, se obtenía un descriptor. Si se enviaba alguna información a la base de datos, se visualizaba en consola que un mensaje fue enviado, caso contrario se mostraba un mensaje de error.

Integración del Algoritmo con el Envío de Datos.

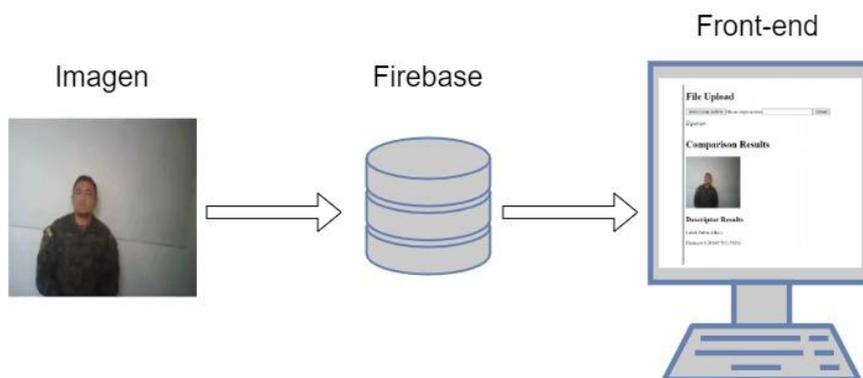
El envío de imágenes a una interfaz de usuario era una necesidad planteada en el prototipo. Se programó un aplicativo web que permitió visualizar todas las imágenes que enviaba la cámara desde las gafas, también se mostró la información del reconocimiento facial. Se escogió este tipo de interfaz debido a su costo y versatilidad.

Para ello se usó como intermediario la base de datos “Firebase”, debido a su eficacia y gran rapidez de respuesta dado que trabaja a tiempo real. Esta base de datos pertenece a Google, y permitió la conexión con el Front-end, además de actualizarlo al instante en caso de alguna modificación en la base de datos. Debido a que este es un servicio pagado, se decidió sobrescribir los datos que llegaban a la base cada vez que una nueva imagen era enviada, esto para evitar salir del límite impuesto por Google en su opción de uso gratuito.

La interacción se muestra en la Figura 34, donde se ve como la imagen era guardada en la base de datos de Firebase e inmediatamente escuchada por el Front-end quien exponía la imagen recibida por la cámara.

Figura 34

Flujo de datos entre imagen front-end



Nota: En esta Figura se puede observar cómo se desarrolla la interacción entre la base de datos y el front end cuando al mismo llega una nueva imagen y como se muestra en pantalla lo que la cámara está viendo.

Desarrollo de Interfaz del Usuario.

La interfaz de usuario era una de las partes más importantes del proyecto, ya que permitió la comunicación entre la máquina y el proceso con el usuario. Debido a que se usó Firebase como parte de la comunicación se decidió usar REACT como framework para el desarrollo del front-end. Las características que destacaban de este framework era su versatilidad y facilidad de comunicación, tanto con JavaScript como con Firebase, por lo que se diseñó una interfaz intuitiva y fácil de usar, donde se podía visualizar todo lo que recibía la cámara en las gafas espía. A su vez, debajo de la imagen, la interfaz mostraba el nombre de la persona que se identificó, y de ser el caso, también la distancia

euclidiana, es decir, el porcentaje de seguridad de la identificación con respecto a la imagen analizada.

Figura 35

Interfaz de usuario



The screenshot shows a web interface with the following elements:

- File Upload**: A section with a button labeled "Seleccionar archivo", a text input field containing "No se eligió archivo", and an "Upload" button.
- preview**: A small icon with the text "preview" below it.
- Comparison Results**: A section with a large image of a person in a military-style uniform.
- Descriptor Results**: A section with the text "Label Darwin Merizalde" and "Distance 0.33685212949436033".

Nota: En la Figura se muestra como está diseñada la interfaz de usuario para que se pueda observar todo lo que la cámara ve, a través de la persona que lo está usando y a su vez, como se muestra el resultado de una identificación.

La interfaz tenía como opción, el subir imágenes de personas que se quieran registrar en la base de datos para una futura identificación, donde se escogía la foto de la persona y se colocaba su nombre en el espacio, para después presionar el botón "UPLOAD" y si los datos se guardaban correctamente, la interfaz entregaba un mensaje de confirmación.

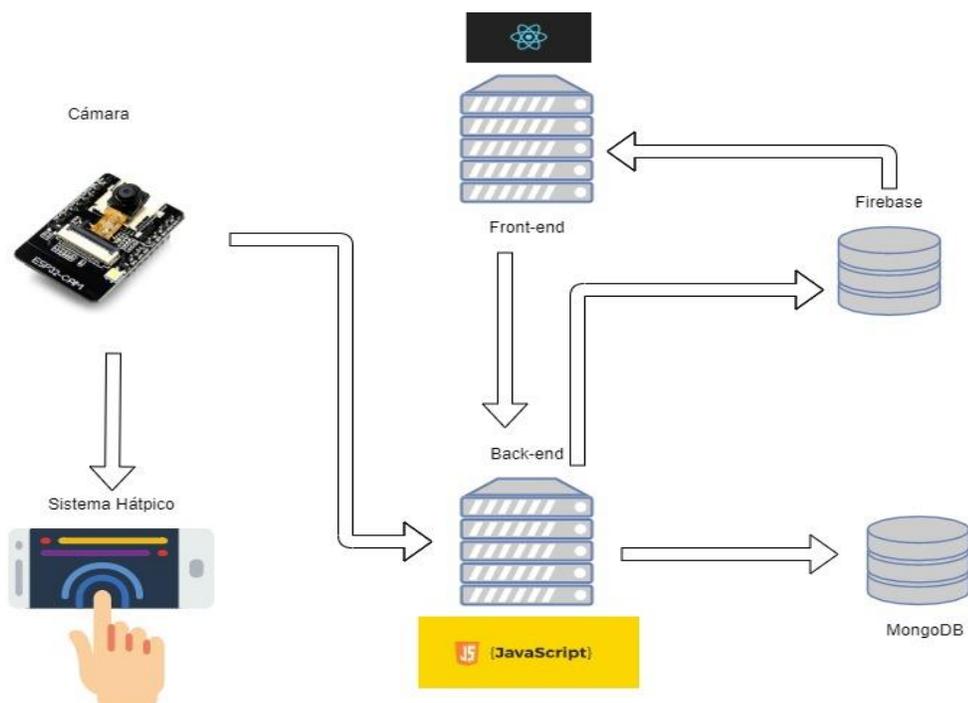
La interfaz estaba programada en lenguaje JavaScript y usó un formato para el HTML, que hace uso de métodos HTTP POST/GET para obtener respuestas del servidor, el mismo se encuentra conectado directamente con Firebase. La interfaz estaba programada para estar siempre escuchando los cambios que sucedan dentro de la base de datos, asegurándose de que las imágenes lleguen en el menor tiempo posible al usuario, y que este proceso se dé lo más cercano a comunicación en tiempo real.

Integración Total del Sistema.

La integración total del Hardware con el Software, y la forma en la que se comunicaban entre sí, se muestra en la siguiente Figura 36.

Figura 36

Comunicación del sistema total

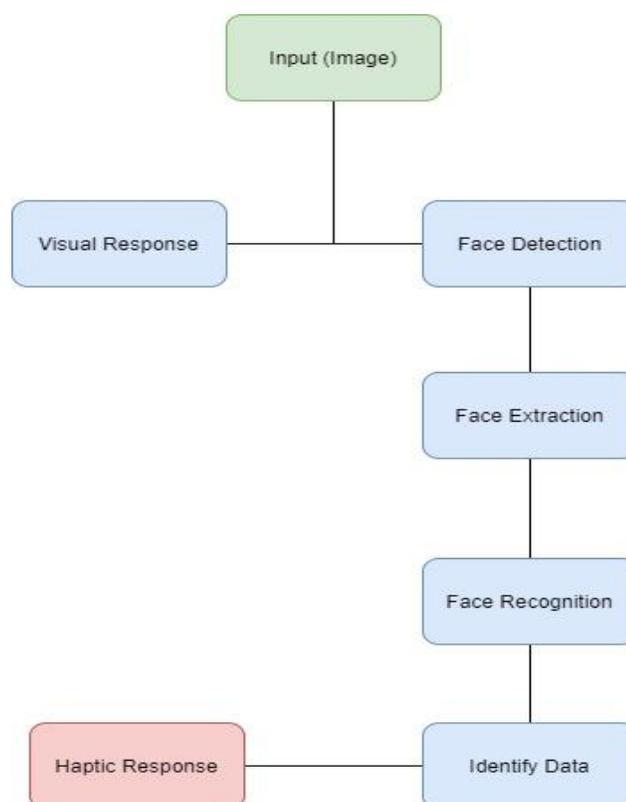


Nota: En la Figura se muestra cómo se realiza la comunicación del sistema desde la cámara a los diferentes servidores y como envía respuesta al sistema háptico implementado.

El funcionamiento completo de las gafas espía requirió que múltiples sistemas trabajen a la vez. La comunicación no mostraba ningún retraso entre ellos, por lo que, no se generó problema alguno en este aspecto, siendo así que todos los sistemas se encontraban interconectados como se muestra en la Figura 36. El sistema interactuaba con cada uno de ellos dando como resultado final el objetivo propuesto. En el siguiente diagrama de bloques se puede observar cómo funciona en concreto la identificación del sistema completo.

Figura 37

Diagrama de bloques del sistema



Nota: En la figura se puede observar en un diagrama de bloques el funcionamiento general del sistema de gafas espía implementado.

Costo del Prototipo

Otra ventaja importante que se encontró en el prototipo, radicaba en el bajo costo del sistema. Se seleccionaron elementos que cumplieran con ciertas características, como son la disponibilidad del mercado y que no tengan un costo excesivo. El precio de los elementos que fueron utilizados se muestran en la siguiente tabla.

Tabla 7

Costo elementos del prototipo

Elemento	Costo
Armazón de Gafas	30
Esp32-cam	15
Programador FTDI	5
Cable conexión	1,5
Soldadura de pines	5
Interruptor	0,3
Módulo cargador de batería	5
Cables conexión para batería	3,6
Micro motor vibrador	4
Batería 3,7 V 900 mA	10
Cables	5
TOTAL	84,4

Nota: En la tabla se puede observar el costo de cada uno de los elementos que se usaron para armar el prototipo final de las gafas espía.

Considerando que en el mercado actual no existe una gran variedad de gafas inteligentes, los costos de estas por lo general son elevados, como es el caso de las gafas inteligentes Vuzix Blade (Vincent, 2019), que constan de un sistema de reconocimiento facial pero que tienen un costo aproximado de 1000\$. Aunque las Vuzix disponen de muchas otras funciones, para este proyecto estaban sobrecalificadas, ya que solo se necesitó de la detección y reconocimiento facial, disminuyendo así el costo a un 10% a comparación de las gafas Vuzix, aunque el precio puede variar si se utilizan elementos

más modernos, como una mejor cámara o un modelo de armazón de gafas en impresión

3d. El prototipo actual se lo puede considerar bastante económico.

Capítulo V

Validación, Pruebas y Resultados

En este capítulo se abordan los resultados del análisis de los datos obtenidos en las dieciséis pruebas realizadas para verificar el comportamiento del sistema propuesto. Estos resultados se concentraron en la respuesta del sistema a una cantidad de experimentos realizados con voluntarios que fueron ingresados en la base de datos, bajo ambientes controlados y variaciones previstas. Todo esto corresponde a información necesaria que valida o no el prototipo, además, se realizó una medición del gasto energético, para calcular el consumo de todo el sistema. Se analizó el porcentaje de acierto que tiene el sistema ante diferentes escenarios, en los cuales interfieren el nivel de luz y la incidencia de lunas de las gafas, así como la cantidad de energía que se está consumiendo durante los ensayos. Los objetivos a determinar fueron: si la arquitectura propuesta es eficiente, cumple su objetivo de reconocimiento facial y entrega una respuesta háptica adecuada para centrar al objetivo.

Para el análisis de los resultados se siguió el siguiente procedimiento:

- Realizar un estudio del porcentaje de acierto en el reconocimiento de personas que tiene el sistema en diferentes ambientes: Luminosidad, diferentes ángulos de vista a la cámara, eficiencia del algoritmo de reconocimiento frente a los diferentes escenarios.
- Comprobar la cantidad de gasto energético que el prototipo genera, y su eficiencia.
- Comprobar el funcionamiento de la respuesta háptica integrada para el rastreo y centrado del objetivo.

Inicialmente, para todas las pruebas se llevó a cabo un protocolo de bioseguridad, debido a la situación mundial al momento de realizar este trabajo (pandemia del Covid-

19). Pese a la problemática, se llevó satisfactoriamente a cabo todos los ensayos planteados, variando lugares, condiciones climáticas y horarios. Los resultados fueron clasificados en base a estas variaciones, ya que era necesario tomar en cuenta factores como la luminosidad que existe en el día y que tiene un alto grado de afectación en el envío de imágenes. Los datos entregaron una respuesta real del comportamiento del dispositivo.

Protocolo Previo a Pruebas de Experimento

Debido a la pandemia del Covid-19 por la que el mundo se encuentra atravesando, son muchos los países que han adaptado normas o protocolos de bioseguridad en las rutinas diarias de las personas. Por la naturaleza en que las pruebas deben efectuarse, se requirió de un ambiente en su mayoría controlado, por lo general los escenarios fueron dentro de una habitación o casa. Para la preservación de la salud de los voluntarios que participaron del experimento y de los tesisistas, se procedió a tomar la temperatura de todos los que participaron y verificar que ninguno presente indicios de fiebre, después de ello, se procedió a desinfectar el asiento para que cada voluntario tome asiento y así se de inicio a las pruebas.

Figura 38

Protocolo bioseguridad Covid-19 para realización de pruebas experimentales



Nota: En la figura se puede ver a uno de los tesistas realizando las pruebas de protocolo de seguridad seguido a una de las personas que participaron en las pruebas.

Es importante indicar el valor que toman los protocolos de bioseguridad, estos no podían ser ignorados puesto que la situación actual obliga a tomar medidas por la seguridad de los voluntarios y de los investigadores, ya que las pruebas del experimento debían realizarse con el sujeto sin mascarilla, o algún otro objeto sobre su rostro, ya que la identificación está basada en los rasgos faciales de la persona y ello podría afectar la veracidad de las pruebas.

Pruebas de Reconocimiento de Personas

Evaluaciones en Escenario 1 – ESPE CICTE (Sala de Reunión Horas de la Mañana – Día Soleado)

Estas son las primeras pruebas que fueron ejecutadas, estas tuvieron lugar en el centro de reuniones de las instalaciones del CICTE en la Universidad de las Fuerzas Armadas ESPE del campus Sangolquí. Para este caso de estudio y siendo un ambiente controlado, la luminosidad que existía en la habitación era lo suficientemente clara, el clima del día era soleado y la sala de reunión tenía grandes ventanas que permitían la entrada de la luz, la variación de las condiciones de luminosidad existentes estaba dada por el encendido y apagado de luminarias de la habitación. Las personas que realizaron las pruebas se encontraban a 1.5 metros de distancia de la cámara que detectaba los rostros, los cuales se encontraban montados en las gafas espía, y que, a su vez, uno de los investigadores la tenía puesta, para mostrar que los datos del reconocimiento son en un uso real.

Pruebas de Detección de Personas a Diferentes Cantidades de Luz sin Lunas de Gafas Oscuras Sobre la Cámara

La primera prueba se enfocó en el análisis de la eficiencia al momento de identificar a los voluntarios por medio de las gafas espía en diferentes condiciones de luminosidad y ángulos de visión.

1era Condición De Luz - Natural

Tabla 8

Escenario 1 - Porcentaje de acierto de reconocimiento de personas sin luna oscura sobre la cámara sin iluminación externa

Prueba	Iluminación (lux)	Porcentaje de Acierto en diferentes ángulos de visión (%)				
		De frente	45°	-45°	Perfil derecho	Perfil izquierdo
1	346	100	20	40	0	0
2	342	100	60	40	0	0
3	352	100	100	100	0	0
4	314	100	100	60	0	0
5	345	100	100	40	0	0
Promedio total	339,8	100	76	56	0	0

Nota. Esta tabla muestra los porcentajes de acierto al reconocer personas a diferentes ángulos de visión del rostro a la cámara sin la intervención de luz externa y sin usar una luna de gafas oscuras sobre la lente de la cámara.

De los resultados obtenidos en la Tabla 8, se pudo observar cómo en el ambiente controlado, la luminosidad promedio era de 339.8 lux, por lo que podemos decir que es un buen nivel de iluminación, además, en ese momento del día, el clima se encontraba soleado, y se pudo determinar que el sistema sin ninguna intervención sobre el lente de la cámara, tuvo un reconocimiento del 100% cuando el sujeto se encontraba de frente hacia la cámara de las gafas espía. A medida que la persona iba cambiando el ángulo de su rostro con respecto a la cámara, el sistema iba perdiendo efectividad, se determinó un porcentaje de acierto del 76 % a 56 %. Finalmente, cuando el sujeto se encontraba de perfil, el sistema no era capaz de reconocerlo.

2da Condición De Luz - Externa (Lámparas)

Tabla 9

Escenario 1 - Porcentaje de acierto de reconocimiento de personas sin luna oscura sobre la cámara con iluminación externa

Prueba	Iluminación (lux)	Porcentaje de Acierto en diferentes ángulos de visión				
		De frente	45°	-45°	Perfil derecho	Perfil izquierdo
1	371	100	40	0	0	0
2	370	100	80	20	0	0
3	371	100	100	100	0	0
4	341	100	60	80	0	0
5	371	100	80	0	0	0
Promedio total	364,8	100	72	40	0	0

Nota. Esta tabla muestra los porcentajes de acierto al reconocer personas a diferentes ángulos de visión del rostro a la cámara con la intervención de luz externa y sin usar una luna de gafas oscuras sobre la lente de la cámara.

De los resultados mostrados en la Tabla 9 se obtuvo un promedio de luz de 364.8 lux, donde se tuvo un escenario con mucha iluminación, parecida la que se obtendría si se usará el sistema en un lugar descubierto, a luz del día , y sin ninguna intervención sobre la cámara, tuvo un reconocimiento de 100% de acierto cuando el sujeto se encontraba con su vista de frente a las gafas espía. Entre $\pm 45^\circ$ de visión, el sistema va perdiendo efectividad entre 72 % a 40 %. Cuando el sujeto se encuentra de perfil el sistema no es capaz de reconocerlo.

Pruebas de Detección de Personas a Diferentes Cantidades de Luz con Lunas de Gafas Oscuras Sobre la Cámara.

La primera prueba que se realizó con una interferencia sobre la cámara, en la cual se colocó la luna de unas gafas oscuras normales encima de la lente de la cámara de las

gafas espía, en este experimento se analizó la precisión de la identificación de personas en diferentes condiciones de luminosidad en los diferentes ángulos de visión establecidos.

1era Condición De Luz - Natural

Tabla 10

Escenario 1 - Porcentaje de acierto de reconocimiento de personas con luna oscura sobre la cámara sin iluminación externa

Prueba	Iluminación (lux)	Porcentaje de Acierto en diferentes ángulos de visión				
		De frente	45°	-45°	Perfil derecho	Perfil izquierdo
1	346	80	80	0	0	0
2	342	100	80	60	0	0
3	352	100	100	40	0	0
4	314	100	60	0	0	0
5	345	100	60	0	0	0
Promedio total	339,8	96	76	20	0	0

Nota. Esta tabla muestra los porcentajes de acierto al reconocer personas a diferentes ángulos de visión del rostro a la cámara sin la intervención de luz externa y con el uso de una luna de gafas oscuras sobre la lente de la cámara.

De los resultados mostrados en la Tabla 10 el nivel de luz que se obtuvo decayó en gran manera, donde también la luminosidad decayó con el uso de la luna oscura, se tuvo un promedio de luz de 339.8 lux. Se determinó que el sistema tiene un reconocimiento de 96 % de acierto cuando el sujeto estaba de frente hacia la cámara de las gafas espía, perdiendo un poco de rendimiento con respecto a pruebas anteriores. A medida que iba cambiando de ángulo, el sistema iba perdiendo efectividad entre 76 % a 20 %. Cuando el sujeto se encontraba de perfil el sistema no era capaz de reconocerlo.

2da Condición De Luz - Externa (Lámparas)

Tabla 11

Escenario 1 - Porcentaje de acierto de reconocimiento de personas con luna oscura sobre la cámara con iluminación externa

Prueba	Iluminación (lux)	Porcentaje de Acierto en diferentes ángulos de visión				
		De frente	45°	-45°	Perfil derecho	Perfil izquierdo
1	371	100	100	0	0	0
2	370	100	100	40	0	0
3	371	100	100	100	0	0
4	341	100	80	20	0	0
5	371	100	60	0	0	0
Promedio total	364.8	100	88	32	0	0

Nota. Esta tabla muestra los porcentajes de acierto al reconocer personas a diferentes ángulos de visión del rostro a la cámara con la intervención de luz externa y con el uso de una luna de gafas oscuras sobre la lente de la cámara.

De los resultados mostrados en la Tabla 11, la obstrucción de la luna de gafas oscuras que estaba sobre la cámara, generó mayor obstrucción de la luz que logró captar, en esta prueba obtuvimos un promedio de luz de 364,8 lux. El sistema obtuvo un reconocimiento de 100 % de acierto cuando el sujeto se encontraba de frente hacia la cámara de las gafas espía, Con variación de ángulo de su rostro hacia la cámara, el sistema iba perdiendo efectividad entre 88 % a 32 %. Cuando el sujeto se posicionaba de perfil el sistema no es capaz de reconocerlo.

Comparativa de Resultados Entre Todas las Pruebas Dadas en el Primer

Escenario

El primer escenario muestro una distintiva entre las diferentes pruebas que se realizaron y son mostradas a continuación:

Tabla 12

Escenario 1 – Diferencia de resultados entre las diferentes pruebas con y sin luna de gafas oscuras sobre la lente de la cámara, en diferentes ángulos de visión.

Prueba	Porcentaje de diferencia entre distintas pruebas (%)					Perfil izquierdo
	Iluminación (lux)	De frente	45°	-45°	Perfil derecho	
Sin Luna	25	0	4	16	0	0
Con Luna	25	4	12	12	0	0

Nota. Esta tabla muestra los porcentajes de diferencia entre las diferentes pruebas realizadas a diferentes ángulos de visión del rostro a la cámara con la intervención una luna de gafas oscuras sobre la lente de la cámara y sin la intervención de ella.

En la tabla 12 se puede observar la diferencia entre el porcentaje de reconocimiento de personas que existió entre las pruebas con luz natural y con luz externa, y como se pudo comprobar, con la cámara sin ningún obstáculo sobre su lente, presentó una gran diferencia del reconocimiento en los ángulos de $\pm 45^\circ$, generando mayor diferencia en el ángulo de -45° , esto dado por la posición de la persona con respecto a la luminosidad que tenía la habitación de prueba, y la ubicación del lugar para el experimento (Merizalde, Aguilar, & Calderón, 2020) (Aguilar, Manosalvas, Guillén, & Collaguazo, 2018) (Galarza, Pérez, Serrano, Tapia, & G, 2018). Debido que para ese ángulo el rebote de la luz se encontraba más hacia la pared y con menor incidencia sobre el rostro del sujeto en las pruebas.

También se pudo observar que existe una mayor diferencia entre las pruebas con la luna de gafas oscura puesta sobre la lente de la cámara, ya que la incidencia de la luz en este caso afecta de mayor manera, dado por la categoría de en la que la luna este clasificada (Trabajo, 2019). Por lo que se observó que entre mayor cantidad de luz exista

en el lugar de prueba del sistema, mejor será la respuesta que este mida, sobre todo con la luna sobre la cámara. Finalmente hay que recalcar que no se obtuvo resultado alguno con el reconocimiento en ángulo de visión de perfil de los sujetos de prueba, un resultado esperado debido a que el algoritmo de reconocimiento facial utilizando funcionaba a través de las facciones faciales de las personas dentro de la base de datos, y cuando la misma se encuentra de perfil, la cámara no lograba captarlas completamente, también gracias a ello, existió un caso donde el reconocimiento se efectuó con éxito pero a su vez entregó falsos positivos, dando como respuesta dos sujetos reconocidos tal y como se puede observar en la siguiente figura.

Figura 39

Escenario 1 - Reconocimiento facial acertado del sistema acompañado con un Falso positivo



Nota. Esta figura indica una de las muestras tomadas donde se dio el problema del falso positivo.

Tabla 13

Escenario 1 - Cantidad de falsos positivos en el escenario 1

Cantidad de Falsos positivos Escenario 1	
Falsos Positivos	4

Nota. Esta tabla muestra la cantidad de falsos positivos que se dieron en total durante las pruebas en este escenario.

Durante 4 de las muestras tomadas para la toma de datos se mostró los falsos positivos como se podían observar en la figura pasada, recalcando que el sistema realizó un reconocimiento correcto, pero a su vez entregó también una respuesta errónea.

Promedio De Resultados De Pruebas Realizados En Escenario 1

Tabla 14

Escenario 1 - Promedio de resultados de pruebas en escenario 1

Promedio total de reconocimiento Escenario 1 (%)				
De frente	45°	-45°	Perfil derecho	Perfil izquierdo
99	78	37	0	0

Nota. Esta tabla muestra el promedio de los porcentajes de reconocimiento facial correcto dados, en total en el primer escenario.

Evaluaciones en Escenario 2 – Conocoto (Sala de una Casa – Horas de la Tarde - Ambiente Nublado y Lluvioso)

Estas pruebas presentaban el segundo escenario de experimentación, el cual fue ejecutado en diferentes lugares de Conocoto, parroquia de la ciudad de Quito. Debido a la situación de la pandemia para este caso de experimentos, se modificó el escenario en

donde se encontraban las personas que se ofrecieron de voluntarias para las pruebas, se visitó uno por uno a sus respectivos hogares, siguiendo de la misma manera los protocolos de bioseguridad establecidos. Se efectuó la variación de luz establecidas por el encendido y apagado de lámparas de la habitación, y las personas que realizaron las pruebas se encontraban a 1.5 metros de distancia de la cámara que detectaba sus rostros, para mostrar que los datos del reconocimiento eran en un uso real.

Pruebas de Detección de Personas a diferentes Cantidades de Luz Sin Lunas de Gafas Oscuras Sobre la Cámara

La prueba consistió en analizar la precisión de la identificación de las gafas espía en diferentes condiciones de luminosidad, en diferentes ángulos de visión que tiene la cámara, con muchos cambios de luminosidad que entrega un día nublado o lluvioso, por lo que la luz natural entregada para las diferentes pruebas y casas varió dependiendo de cómo estaba construida su sala y cuanta entrada de luz tuvo.

1era Condición De Luz - Natural

Tabla 15

Escenario 2 - Porcentaje de acierto de reconocimiento de personas sin luna oscura sobre la cámara sin iluminación externa

Prueba	Iluminación (lux)	Porcentaje de Acierto en diferentes ángulos de visión (%)				
		De frente	45°	-45°	Perfil derecho	Perfil izquierdo
1	297	100	100	100	0	20
2	96	100	100	100	0	0
3	15	0	0	0	0	0
4	150	100	100	100	0	0
5	176	100	100	100	100	80
6	345	100	100	100	0	0
Promedio total	179,83333	83,33333	83,33333	83,33333	16,6666	16,6666

Nota. Esta tabla muestra los porcentajes de acierto al reconocer personas a diferentes ángulos de visión del rostro a la cámara sin la intervención de luz externa y sin el uso de una luna de gafas oscuras sobre la lente de la cámara en diferentes salas de casas del sector Quito, Conocoto.

De los resultados mostrados en la Tabla 15, debido a las condiciones de cada una de las pruebas, se observó que la luz es un factor que afecta de manera directamente proporcional al reconocimiento que se puede dar en este algoritmo, se denota que el resultado que se obtuvo sobre todo en ángulo de visión de frente es de 83.3 %, lo cual era un buen resultado, sin embargo, se pudo observar que en 15 lux, una cantidad de luz muy baja, no se pudo captar prácticamente nada y por ello el reconocimiento fue igual a 0 %, sin tomar en cuenta la prueba con este factor de luz podríamos considerar que el reconocimiento fue del 100%, con respecto a los ángulos se observó que el reconocimiento es de 100% en $\pm 45^\circ$, además, el sistema fue capaz de captar 16.67 % cuando la persona se encontraba de perfil sin entregar falsos positivos. Hay que destacar que la prueba de 354 lux se hizo con luz del día directa sobre la persona y la cámara, lo que significa que la prueba se efectuó a techo descubierto.

2da Condición De Luz - Externa (Lámparas)

Tabla 16

Escenario 2 - Porcentaje de acierto de reconocimiento de personas sin luna oscura sobre la cámara con iluminación externa

Prueba	Iluminación (lux)	Porcentaje de Acierto en diferentes ángulos de visión				
		De frente	45°	-45°	Perfil derecho	Perfil izquierdo
1	319	100	100	100	100	0
2	178	100	100	100	0	0
3	35	100	100	100	0	20
4	234	100	100	100	0	20
5	208	100	100	100	100	20
Promedio total	194,8	100	100	100	40	12

Nota. Esta tabla muestra los porcentajes de acierto al reconocer personas a diferentes ángulos de visión del rostro a la cámara con la intervención de luz externa y sin el uso de una luna de gafas oscuras sobre la lente de la cámara en diferentes salas de casas del sector Quito, Conocoto.

De los resultados mostrados en la Tabla 16, se pudo observar que en ángulo de visión de frente el reconocimiento que se tuvo es del 100 % de acierto, lo cual se consideró como un excelente desempeño. En los ángulos de $\pm 45^\circ$, nuevamente el sistema volvió a entregar resultados favorables, sin embargo, inesperadamente cuando se realizó la prueba del ángulo de perfil de la persona, el sistema entregó entre 40 % a 12 % de acierto, lo que demostró que dependiendo del ángulo, si la persona no se colocaba completamente de perfil, existía una pequeña posibilidad de que el sistema lo reconociera, y también se observó que algunas personas tenían rasgos característicos que hacía que su reconocimiento fuese más eficaz. Por último, destacar que esta prueba constó solo de 5 personas en vez de 6, debido a que la anterior se la realizó bajo la luz

del día y por ello no se pudo variar la cantidad lumínica, ya que el ambiente de pruebas no debía ser cambiado y por ello no se tomó en cuenta para este caso.

Pruebas de Detección De Personas A Diferentes Cantidades De Luz Con Lunas De Gafas Oscuras Sobre La Cámara

Esta es la segunda prueba que se realizó con una interferencia sobre la cámara, en esta serie de experimentos en diferentes locaciones como se había especificado, la cantidad de luz que la luna disminuye a la cámara afecta en mayor manera en ciertos escenarios provocando que no sea posible reconocer nada.

1era Condición De Luz - Natural

Tabla 17

Escenario 2 - Porcentaje de acierto de reconocimiento de personas con luna oscura sobre la cámara sin iluminación externa

Prueba	Iluminación (lux)	Porcentaje de Acierto en diferentes ángulos de visión				
		De frente	45°	-45°	Perfil derecho	Perfil izquierdo
1	297	100	100	100	0	0
2	96	100	60	0	0	0
3	15	0	0	0	0	0
4	150	80	60	40	0	0
5	176	100	0	0	0	0
6	345	100	100	100	0	0
Promedio total	179,8333	80	53,3333	40	0	0

Nota. Esta tabla muestra los porcentajes de acierto al reconocer personas a diferentes ángulos de visión del rostro a la cámara sin la intervención de luz externa y con el uso de una luna de gafas oscuras sobre la lente de la cámara en diferentes salas de casas del sector Quito, Conocoto.

De los resultados mostrados en la Tabla 17, al añadir la luna de gafas oscuras sobre la lente de la cámara se denotó que los resultados varían en gran medida sobre el reconocimiento, más aún cuando las pruebas no se realizaron en el mismo lugar. Con el ángulo de frente a la cámara se obtuvo un 80 % de acierto, bastante fiable. Incluyendo el problema antes mencionado en el análisis de la tabla 15 con la baja cantidad de luz, sin embargo, a partir de ello, con el cambio de ángulos de visión el reconocimiento redujo su efectividad entre 53.33 % a 40 %. Observando estos resultados, como era de esperarse, de perfil el sistema no logró reconocer nada.

2da Condición de luz - externa (lámparas)

Tabla 18

Escenario 2 - Porcentaje de acierto de reconocimiento de personas con luna oscura sobre la cámara con iluminación externa

Prueba	Iluminación (lux)	Porcentaje de Acierto en diferentes ángulos de visión				
		De frente	45°	-45°	Perfil derecho	Perfil izquierdo
1	319	100	100	100	100	0
2	178	100	100	0	0	0
3	35	0	0	0	0	0
4	234	100	100	100	0	40
5	208	100	100	100	60	0
Promedio total	194,8	80	80	60	32	8

Nota. Esta tabla muestra los porcentajes de acierto al reconocer personas a diferentes ángulos de visión del rostro a la cámara con la intervención de luz externa y con el uso de una luna de gafas oscuras sobre la lente de la cámara en diferentes salas de casas del sector Quito, Conocoto.

De los resultados mostrados en la Tabla 18, se pudo observar que el porcentaje de reconocimiento mejoró con respecto a la luz natural de la habitación donde se

realizaron las pruebas, sin embargo, en el ángulo de frente se tiene el 80% de acierto en el reconocimiento, esto es debido a que en 35 lux con la luna sobre la cámara, la luz que captó el sensor de la cámara es muy baja y por ello no logró reconocer nada, pero a distintas luminosidades el sistema se desempeñó bien logrando reconocimiento de ángulo de perfil entre 32 % a 8 % sin entregar falsos positivos.

Comparativa de resultados entre todas las pruebas dadas en el segundo escenario

El segundo escenario mostró una distintiva aun mayor entre las diferentes pruebas que se realizaron, debido a las condiciones de las que se tomaron y son mostradas a continuación:

Tabla 19

Escenario 2 – Diferencia de resultados entre las diferentes pruebas con y sin luna de gafas oscuras sobre la lente de la cámara, en diferentes ángulos de visión.

Prueba	Porcentaje de diferencia entre distintas pruebas (%)					
	Iluminación (lux)	De frente	45°	-45°	Perfil derecho	Perfil izquierdo
Sin Lente	14,96667	16,666	16,666	16,666	23,333	4,666667
Con Lente	14,96667	0	26,666	20	32	8

Nota. Esta tabla muestra los porcentajes de diferencia entre las diferentes pruebas realizadas a diferentes ángulos de visión del rostro a la cámara con la intervención una luna de gafas oscuras sobre la lente de la cámara y sin la intervención de ella.

En la Tabla 19 se puede observar la diferencia entre el porcentaje de reconocimiento de personas que existió entre las pruebas con luz natural y con luz externa. Se presentó una gran diferencia del reconocimiento entre todas las pruebas, sin embargo, como se mencionó en los análisis individuales la baja cantidad de luz afecta el

reconocimiento, por lo que el porcentaje de reconocimiento entre ambas situaciones, con respecto a los 45° se notó una mejora, lo que probó que el sistema funcionaba en diferentes ambientes mientras existan buenas condiciones de iluminación. Cuando se habla del análisis del ángulo de perfil de la persona se mostró como la diferencia que existe entre pruebas es aún mayor, esto debido a que el sistema fue capaz de reconocer en algunas ocasiones al sujeto al frente de la cámara, y las condiciones de luminosidad permitían que el reconocimiento trabaje de mejor manera gracias a ello. Por último, en este escenario de pruebas no existieron falsos positivos por lo que el sistema funcionó de la manera esperada.

Promedio De Resultados De Pruebas Realizados En Escenario 2

Tabla 20

Escenario 2 - Promedio de resultados de pruebas en escenario 2

Promedio total de reconocimiento Escenario 1 (%)				
De frente	45°	-45°	Perfil derecho	Perfil izquierdo
85,83333	79,16667	70,83333	22,16667	9,166667

Nota. Esta tabla muestra el promedio de los porcentajes de reconocimiento facial correcto dados, en total en este escenario.

Evaluaciones de Detección de Personas en Escenario 3 – Sangolquí (Habitación de Estudio – Anochecer)

Estas pruebas presentan el tercer escenario de experimentación, que fue ejecutado en Sangolquí en la habitación de estudio de una casa. Para esta serie de eventos se consideró verificar cual era la respuesta que tenía el sistema en un escenario nocturno o por anochecer, sabiendo que la luminosidad iba a afectar altamente. Se realizó unas pruebas previas para determinar si ameritaba realizar el experimento en diferentes

condiciones de luz o si era necesario implementar la luna de gafas oscuras sobre la cámara, como es de conocimiento general este tipo de lunas es usada para gafas de sol, por ello las mismas no están diseñadas para ser usadas en la noche, por lo tanto, efectivamente la cantidad de luz era muy baja. Las pruebas para este ambiente, se realizaron con las gafas sin ninguna obstrucción sobre la lente de la cámara y con una luz externa.

Pruebas De Detección De Personas Con Luz Externa Sin Lunas De Gafas Oscuras Sobre La Cámara

Esta es la tercera prueba que se realizó sin ninguna interferencia sobre la cámara, en la serie de experimentos se midió el porcentaje de reconocimiento que tenía el sistema cuando se encontraba bajo una condición de luz natural como es el anochecer, pero tenía una luz externa como es la de una habitación normal de estudio de una casa para mejorar la iluminación.

Tabla 21

Escenario 3 - Porcentaje de acierto de reconocimiento de personas sin luna oscura sobre la cámara con iluminación externa

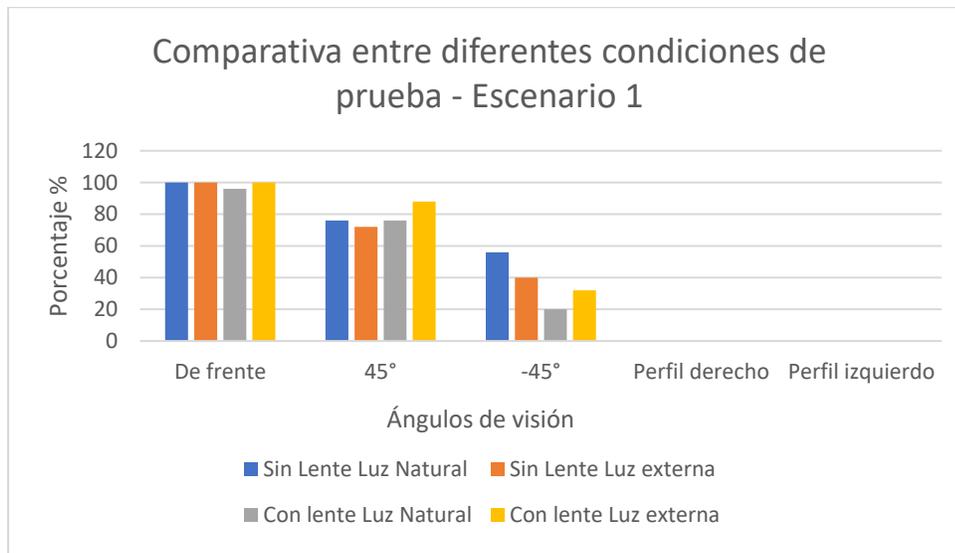
Prueba	Iluminación (lux)	Porcentaje de Acierto en diferentes ángulos de visión (%)				
		De frente	45°	-45°	Perfil derecho	Perfil izquierdo
1	141	100	100	100	0	0
2	292	100	80	100	20	0
3	279	100	100	100	0	0
4	273	100	100	100	0	0
5	220	100	100	80	0	0
Promedio total	241	100	96	96	4	0

Nota. Esta tabla muestra los porcentajes de acierto al reconocer personas a diferentes ángulos de visión del rostro a la cámara con la intervención de luz externa y sin el uso de una luna de gafas oscuras sobre la lente de la cámara en la habitación de estudio de una casa al anochecer del día.

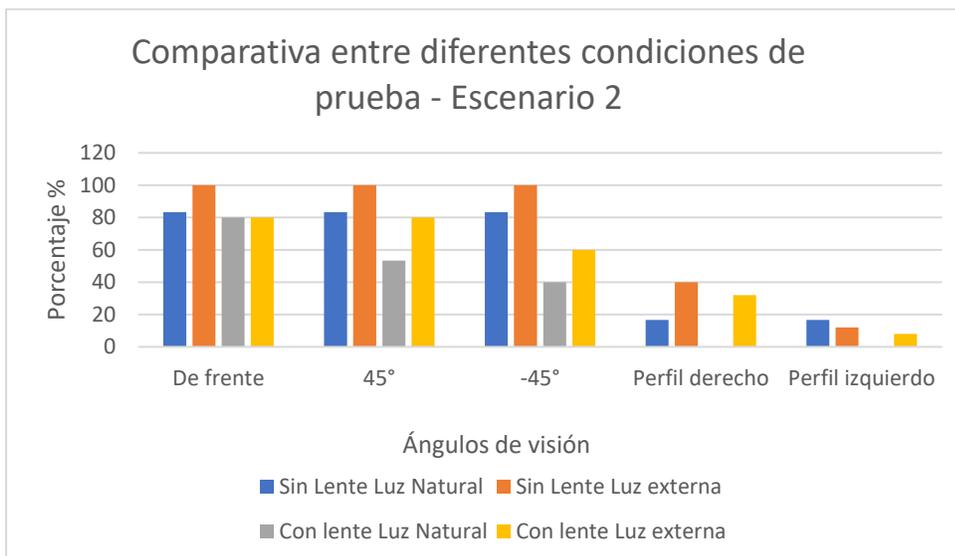
En los resultados mostrados en la Tabla 21, se obtuvieron porcentajes de aciertos bastante altos en todas las pruebas realizadas, siendo así, que se tuvo alrededor de 100% de reconocimiento acertado en 3 de los ángulos de visión más importantes. Dentro del ángulo de visión de perfil, el sistema mostró un desempeño menor al de anteriores pruebas, logrando reconocer tan solo en 1 de los casos al objetivo. Este comportamiento del sistema se lo atribuyó a la hora del día en el que se realizó las pruebas, debido a que la estabilidad del internet y la cantidad de luz fueron las adecuadas.

Comparación de Resultados Entre los 3 Escenarios de Reconocimiento

Una vez realizada las pruebas de reconocimiento en los escenarios planteados, se observó cómo realmente se comportó el sistema en cada uno de los ambientes, y después se realizó una comparación general que permitió determinar si el sistema era viable. Para ello se presentará primero los gráficos del promedio de reconocimiento de los 3 escenarios.

Figura 40*Gráfica de resultados Escenario 1*

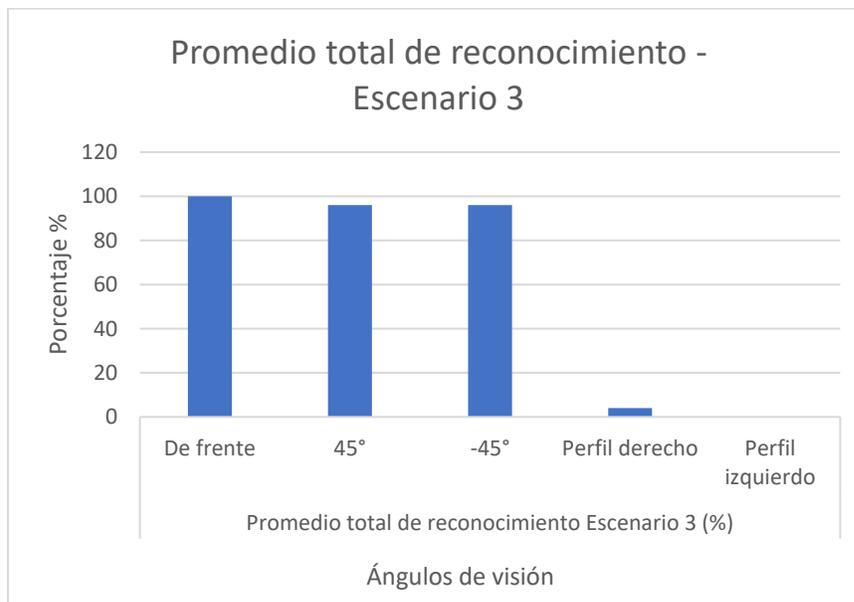
Nota. En esta Figura se muestran de forma gráfica la diferencia que existe del reconocimiento entre los diferentes casos de prueba en el escenario 1.

Figura 41*Gráfica de resultados Escenario 2*

Nota. En esta Figura se muestran de forma gráfica la diferencia que existe del reconocimiento entre los diferentes casos de prueba en el escenario 2.

Figura 42

Gráfica de resultados Escenario 3



Nota. En esta Figura se muestran de forma gráfica la diferencia que existe del reconocimiento entre los diferentes casos de prueba en el escenario 3.

A través de las Figura 41 y Figura 42 presentadas podemos ver como la incidencia de una luz externa en los experimentos mejora el porcentaje de reconocimiento de personas que el sistema puede medir, sobre todo cuando este se encontraba con la luna de gafas oscura sobre la cámara, situación donde el sistema de reconocimiento se sobreponía. Con el objetivo de tener una estética de gafas espía que pase desapercibido ante las personas, tal como gafas de sol normales, el prototipo debe funcionar bajo estas condiciones, aunque depende del usuario que la vaya a usar determinar de qué manera quiere aplicarlo. Además de ello se pudo observar que el sistema a menor condición de

luz, tendía a variar más su porcentaje de reconocimiento entre luminosidad natural y teniendo una luz externa de ayuda. En las siguientes tablas se muestra el resultado final del reconocimiento de personas que entrega el sistema.

Tabla 22

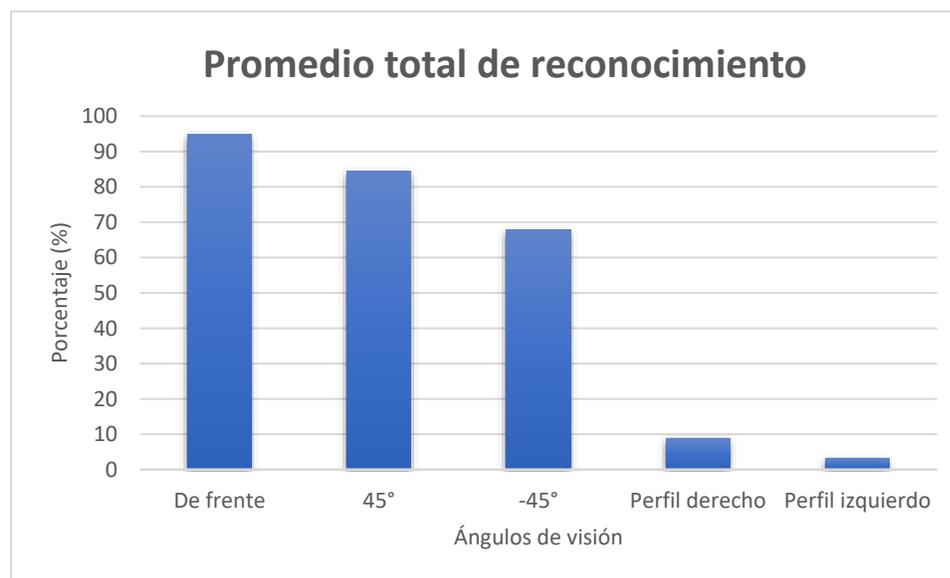
Porcentaje promedio final de reconocimiento de personas

Promedio total de reconocimiento (%)				
De frente	45°	-45°	Perfil derecho	Perfil izquierdo
94,9444433	84,38889	67,9444433	8,72222333	3,05555567

Nota. En esta tabla se muestran los porcentajes de reconocimiento totales de todas las pruebas que se realizaron.

Figura 43

Gráfica del porcentaje promedio final del reconocimiento de personas



Nota. En esta gráfica se muestran los porcentajes de reconocimiento totales de todas las pruebas que se realizaron.

De los resultados del reconocimiento de personas, se pudo observar que el sistema tenía un porcentaje de reconocimiento de 94.94 % de acierto cuando el sujeto se ubicaba de frente a la cámara, mostrando un excelente desempeño, pero dependiendo de la aplicación el resultado puede no ser el más óptimo. Se mostró que de igual forma, se tuvo una excelente respuesta del sistema a 45° con 84.38 % de éxito en el reconocimiento, sin embargo, se vió una reducción en el ángulo de -45° a 67.94 %, todo esto atribuido a la cantidad de luz que llegaba hacia los sujetos de prueba en ese ángulo en las primeras pruebas. Cuando el voluntario se colocaba de perfil, era de esperarse que no exista acierto en estos ángulos, debido al algoritmo de reconocimiento que se uso está basado en rasgos faciales, sin embargo, hemos encontrado que el sistema pudo reconocer en un 8.72 % y en un 3.05 % a las personas correctamente, por lo que se dedujo que mientras el sujeto de prueba no se encuentre totalmente de perfil, y este con un ligero ángulo de inclinación hacia la cámara, existen probabilidades de sea reconocido.

Para finalizar, sabemos que la cámara requirió de cierto nivel de luz para poder reconocer, y en base a las pruebas realizadas, se ha encontrado que a partir de 30 lux el sistema reconoce a las personas en un 0 %. Con la luna de gafas oscuras montada sobre el prototipo muestra que a partir de los 35 lux reconoce un 0 %, por lo que el nivel de luminosidad en el lugar donde se vaya a usar debe tener un mínimo para funcionar, y mientras más luz haya será mejor.

Pruebas de Sistema Háptico

Dentro del prototipo que se ha desarrollado, existen algunas funciones de ayuda al usuario, aparte del reconocimiento facial se puede direccionar a la persona hacia el objetivo que se requiere por medio de un sistema háptico, que funciona a la misma velocidad del envío de imágenes al servidor, y su función es centrar el sujeto al que la cámara reconoció. Para estas pruebas se hizo uso de 3 posiciones del rostro de la persona identificada, variando con respecto al eje cartesiano que presenta la cámara,

siendo así el centro de la imagen el punto cero, la derecha nuestro eje x positivo y la izquierda nuestro eje x negativo. Se ha agregado la prueba de cuando no existe nadie frente a la cámara para evitar falsos positivos. Se realizó una serie de muestras del funcionamiento girando la cabeza del usuario de las gafas espía de izquierda a derecha para verificar que el sistema entregue respuestas correctamente.

Tabla 23

Pruebas de sistema háptico

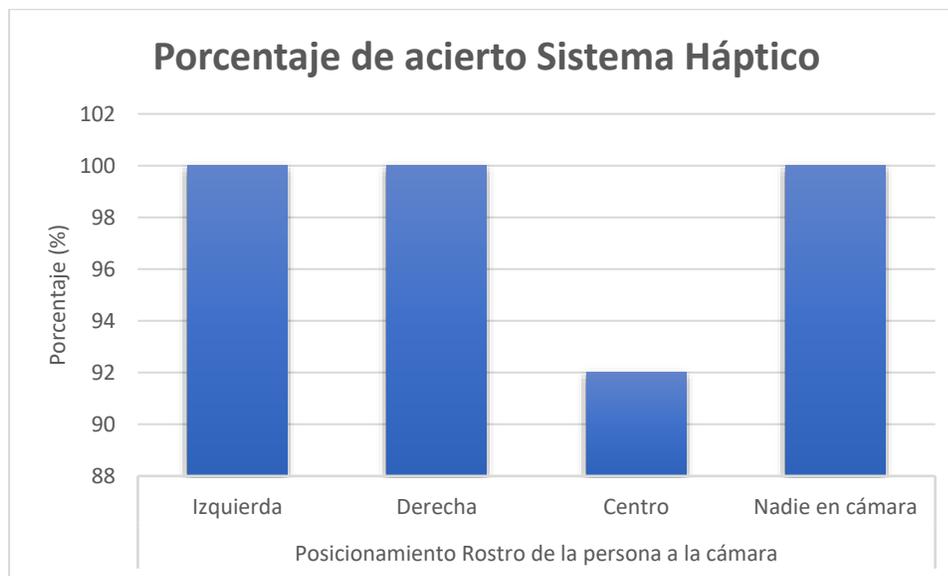
Porcentaje de acierto Sistema Háptico (%)				
Prueba	Posicionamiento Rostro de la persona a la cámara			
	Izquierda	Derecha	Centro	Nadie en cámara
1	100	100	100	100
2	100	100	100	100
3	100	100	100	100
4	100	100	60	100
5	100	100	100	100
Promedio TOTAL (%)	100	100	92	100

Nota: En la siguiente tabla se puede ver el porcentaje de reconocimiento acertado del sistema háptico frente a las pruebas realizadas.

Como se mostró en la Tabla 23 la respuesta que entregó el sistema háptico funciona con casi ningún margen de error, mostrando casi un 100 % de acierto en todas las pruebas realizadas, por lo que se dedujo que el centrado funcionaba bien y no sobrecargó a la batería con el encendido de los motores que tenía el sistema.

Figura 44

Resultados de pruebas del sistema háptico



Nota: En la siguiente figura se puede ver unas gráficas equivalentes a el porcentaje de reconocimiento acertado del sistema háptico frente a las pruebas realizadas.

El resultado de reconocimiento total de las pruebas fue del 98 % de acierto con respecto a las perspectivas dadas, a pesar de la buena respuesta que se ha encontrado con el experimento, se esperaba que el resultado del reconocimiento cuando el voluntario este de frente a la cámara sea del 100 % y como se observó en la Tabla 23 uno de los casos presenta solo un 60 % de reconocimiento exitoso, esto después de otras pruebas se concluyó que es debido a la velocidad de respuesta que tenía el sistema, ya que esta enviaba imágenes al servidor por aproximadamente 1 imagen por segundo, el sistema háptico tardaba la misma cantidad de tiempo en mostrar al usuario una respuesta, por ello, la movilidad a tener con las gafas espía debe ser lenta, porque a su vez la cámara por no poseía autoenfoco y ser una cámara de 2 MPX, el movimiento brusco no permitió un buen enfoque (Aguilar & Angulo, Real-Time Model-Based Video Stabilization for Microaerial Vehicles, 2016), (Aguilar & Angulo, Real-time video stabilization without

phantom movements for micro aerial vehicles, 2014), (Aguilar, Angulo, & Pardo, Motion intention optimization for multirotor robust video stabilization, 2017), y el centrado de objetivo debía ser a una velocidad de rotación del usuario baja.

Pruebas de Carga Energética

Para el prototipo se implementó con una batería de 3.7 V y 900 mA, que fuera lo suficientemente discreta pero que a su vez sea capaz de conllevar la carga del módulo de la cámara y el sistema háptico. Dentro de las pruebas también se midió cual es la eficiencia energética que tenía el prototipo para verificar por cuánto tiempo se va a poder usar antes de requerir cargar de nuevo su batería.

Tabla 24

Carga energética en todas las pruebas realizadas

Prueba	Tiempo de prueba (min)	Voltaje Inicio (V)	Voltaje Final (V)	Consumo energético (V)
1 - 5	0:51	3,78	3,66	0,12
6	0:17	3,66	3,64	0,02
7	0:23	3,64	3,62	0,02
8	0:14	3,62	3,62	0
9	0:12	3,62	3,62	0
10	0:10	3,62	3,62	0
11	0:33	3,62	3,576	0,044
12	0:11	3,57	3,55	0,02
13	0:04	3,55	3,545	0,005
14	0:04	3,545	3,53	0,015
15	0:05	3,538	3,536	0,002
16	0:05	3,536	3,533	0,003
17	0:06	3,533	3,513	0,02
18	0:07	3,513	3,508	0,005
19	0:01	3,508	3,493	0,015
20	0:04	3,493	3,491	0,002
21	0:05	3,491	3,488	0,003
Total	3:32			0,294

Nota: En la tabla se puede ver la cantidad de tiempo de duración de cada una de las pruebas realizadas, junto con el desgaste energético que se tuvo en la batería durante cada prueba.

Como se observó en la Tabla 24, la mayoría de las pruebas muestreaban un desgaste energético bastante bajo, en esta tabla también se tomó las primeras 5 muestras como 1 sola fila, debido a que los experimentos realizados en el primer escenario se hicieron seguidas una de la otra, y las gafas espía se encontraban encendidas todo el tiempo hasta finalizar en ese escenario. Las demás pruebas se hicieron de manera individual y pausada. Se tomaron en consideración las pruebas del sistema háptico, razón por la cual el número de muestras totales que se muestran en este apartado es de 21, por ello, tenemos que el experimento inició con la batería totalmente cargada, con 3.78 V y termino sin recargarla con 3.488 V. Con un tiempo total de experimentación de 3 horas con 32 minutos. Se tiene un promedio de gasto energético de:

$$\frac{\text{Gasto energético}}{\text{minuto}} = 0.00138679 \text{ V}$$

Sabiendo esto, debemos tomar en cuenta que por el tipo de batería que se tenía y por el sistema que se usó, el suministro energético no debía bajar de 3.3 V, por lo que, haciendo una estimación, el tiempo de uso de las gafas espía con carga completa, y usándolos antes de volver a cargar es de aproximadamente.

$$\text{Tiempo de uso} = 346.12 \text{ min} \cong 5.46 \text{ horas}$$

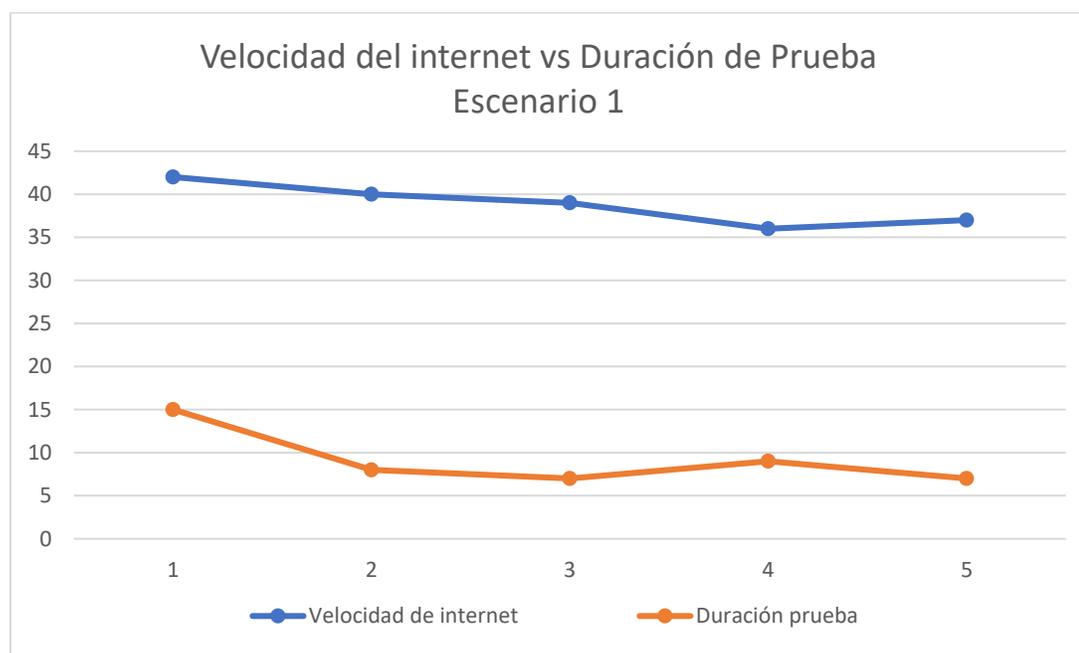
Con aproximadamente 6 horas de suficiencia energética, consideramos que el prototipo cumple con las expectativas programadas.

Duración de Pruebas en Base a la Velocidad de Internet

Un riesgo durante la implementación del sistema, fue su dependencia a la velocidad del internet que llegaba al módulo wifi, para la velocidad de las pruebas. Al tratar de representar lo más cercano a tiempo real, cualquier interrupción de comunicación o envío de datos perjudicará la experiencia del usuario con el uso de las gafas espía, por ello, se decidió medir la duración de las pruebas que se tuvieron en base a la velocidad de internet que se tenía en su momento y realizar una comparación.

Figura 45

Resultados de pruebas en base a velocidad de internet en Escenario 1



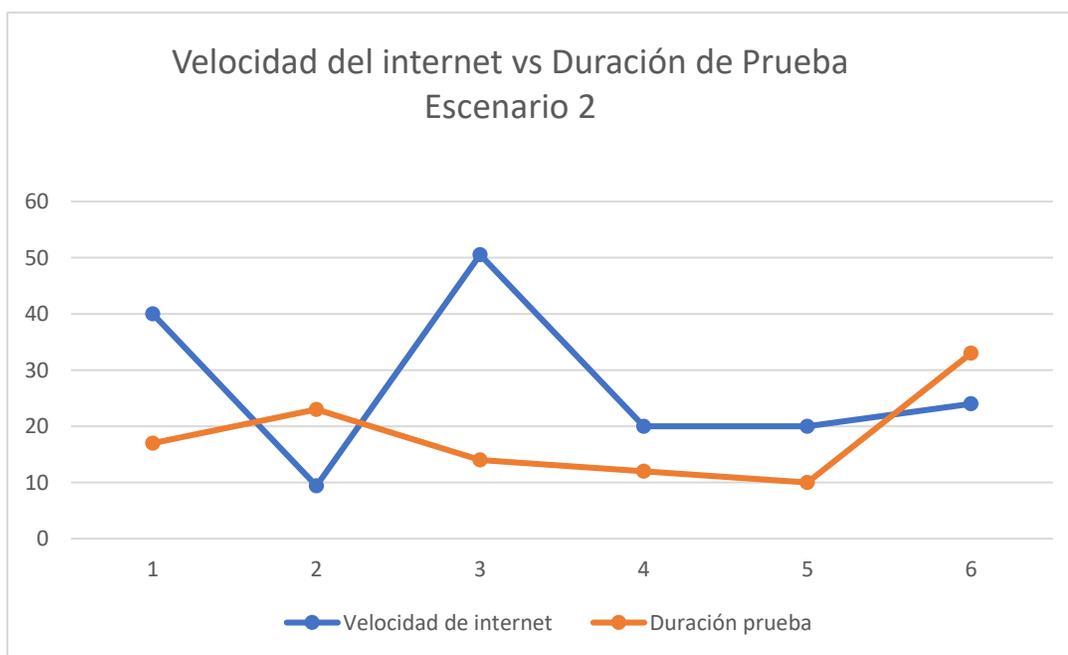
Nota: En esta gráfica se puede observar la variación que tiene el tiempo de las pruebas en base a la fluctuación de la velocidad del internet en el escenario 1.

Para el primer escenario se observó, que la velocidad de internet se mantenía constante y por ello las pruebas no variaban en gran medida, aunque, en cualquier escenario, el sistema se demoraba en conectarse a la red y esto provocó que la primera

prueba dure más. En este escenario en específico surgió un problema de conexión donde el módulo no era capaz de enviar datos, y después de cambiarlo a una red wifi generada por una computadora portátil logró enviar con éxito, pero existió muchas fluctuaciones iniciales que hacían que las imágenes que enviaba la cámara, fueran de aproximadamente 1 cada 5 segundos, situación que fue mejorando después de un tiempo.

Figura 46

Resultados de pruebas en base a velocidad de internet en Escenario 2



Nota: En esta gráfica se puede observar la variación que tiene el tiempo de las pruebas en base a la fluctuación de la velocidad del internet en el escenario 2.

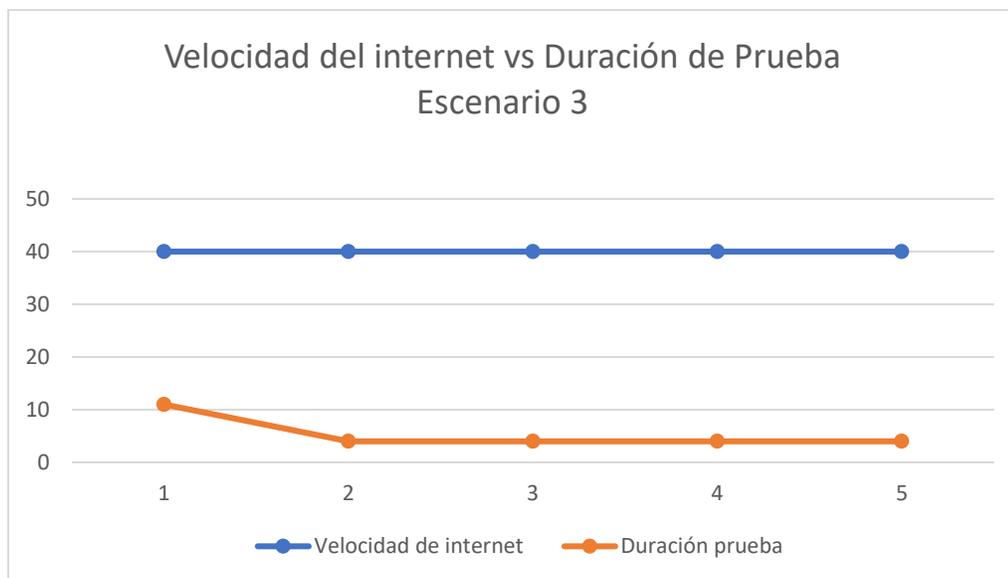
Para el escenario 2 pudimos observar que, cuando la velocidad del internet se redujo en comparación a las otras, él envió de imágenes era más lento, por ende, la duración de la prueba fue mayor como es en la segunda toma de este escenario, sin embargo, hay un caso en particular en este dónde a pesar de tener 24 Mbps de velocidad

el prototipo no era capaz de enviar imágenes. Después de varios intentos, se logró el envío de datos al servidor, pero el tiempo de respuesta era de entre 5 a 10 segundos por cada envío de imagen, aparte de ello en una prueba que se intentó realizar después en un sitio cercano, no se logró captar ningún envío de datos por lo que el experimento en ese lugar fue cancelado.

Para asociar estos casos los lugares de pruebas fueron cercanos al primer escenario, donde pudimos deducir que el sector tiene ciertas interferencias en el servicio de internet y puede que esto provocó que el sistema no funcione de manera óptima, por lo que habrá sectores donde no se pueda utilizar el sistema debido al servicio que se tenga de internet.

Figura 47

Resultados de pruebas en base a velocidad de internet en Escenario 3



Nota: En esta gráfica se puede observar la variación que tiene el tiempo de las pruebas en base a la fluctuación de la velocidad del internet en el escenario 3.

Para el último escenario de pruebas no se tuvo ningún inconveniente y vemos cómo el sistema funcionaba de manera óptima. Se dedujo que se necesitan al menos 10 Mbps de velocidad para que el sistema funcione correctamente, pero que está limitado por el servidor de internet al que esté conectado y las interferencias del mismo. Pero que mientras mayor sea la velocidad como es el caso de 40 Mbps va a trabajar de manera óptima.

Capítulo VIII

Conclusiones y Recomendaciones

Conclusiones

Tras llevar a cabo una serie de investigaciones sobre técnicas y estrategias de inteligencia artificial enfocadas al área de computer vision, junto con el uso de un módulo wifi esp32-cam y su cámara incorporada, se desarrolló e implementó un sistema de reconocimiento facial portátil para la detección y reconocimiento de personas en entornos distintos, y diferentes ángulos de visión. De manera que ofrece al usuario un método ergonómico y discreto de identificación de personas, por medio de una respuesta visual y física, con un sistema basado en detección de facciones del rostro, para un aplicativo de seguridad.

El sistema de reconocimiento facial por medio de la librería Face-api ofrece resultados óptimos acorde a nuestro aplicativo, posee un alto grado de aciertos dependiendo del ángulo de visibilidad que disponga hacia el rostro del objetivo a identificar, puesto que el algoritmo basa su identificación en puntos de referencia tales como la boca, nariz, ojos y cejas. Se muestra un porcentaje de acierto de 94,94 % cuando la persona se encuentra con su rostro totalmente de frente hacia la cámara, a medida que el sujeto gira su rostro entre $\pm 45^\circ$ el porcentaje de acierto decrementa a 84,38 % y 67,94 % respectivamente, lo que asegura una buena identificación en estos casos, pero el porcentaje disminuye drásticamente si la persona se encuentra de perfil debido a la dificultad de reconocer sus rasgos faciales en esa posición, encontrando un 8,72 % a 3,05 % de acierto, determinando que el sistema no entregará buenos resultados mientras más se acerque el ángulo de visión del rostro de la persona con respecto a la cámara, hacia los $\pm 90^\circ$.

El reconocimiento entregado por el prototipo es de gran confiabilidad, entregando entre las 260 pruebas realizadas un porcentaje de reconocimiento del 94.94 % en ángulo frontal, donde solo en 3 de las pruebas existió la incidencia de falsos positivos, seguidos de la respuesta correcta, dando un total de 4 muestras de falsos positivos en las pruebas, por lo que se infiere que el sistema funciona correctamente y tiene una probabilidad de 1,15% de fallo o de entregar falsos positivos como respuesta de identificación.

El sistema de centrado de imagen se lo implementó por medio de un sistema háptico, el cual fue colocado en cada lado de las gafas, y dirige al usuario hacia el rostro identificado. Este sistema tiene una efectividad del 98 % donde se determinó que el sistema no puede funcionar al 100 % de acierto debido a movimientos bruscos y que depende también de la velocidad del internet. El usuario debe ajustar lentamente y con suavidad el ángulo de visión que tiene para que el sistema pueda responder acorde a lo que el mismo se encuentra observando en ese momento.

Se implementó el algoritmo de reconocimiento dentro de un servidor local el cual tiene una conexión directa con el módulo ESP32-CAM, este último envía las imágenes capturadas a través de una cámara y que son procesadas para entregar la respuesta de la identificación a los diferentes sistemas implementados. El sistema se ha dispuesto para trabajar lo más cercano posible al tiempo real, sin embargo, esto se ve delimitado por las capacidades provistas en el módulo de la cámara que se usó, donde actualmente el promedio de envío de imágenes es 1 frame por segundo, debido a que al reducir el tiempo de envío de imágenes por debajo de 1000 ms, esta no cambia su tiempo de reacción, dando como conclusión que la mejor respuesta que puede entregar la cámara al sistema es de 1 imagen cada segundo.

Los costos totales del sistema se encuentran por debajo de los 100\$, y estos pueden variar dependiendo en su mayoría del armazón en el cual se implementará, por lo que hay que considerar que el prototipo que se ha desarrollado tiene un considerable

bajo costo en comparación a un producto ya existente como son las gafas inteligentes Vuzix Blade que proveen un sistema de reconocimiento facial, además de muchas otras características que no son necesarias para el proyecto, pero que tiene un costo de aproximadamente 1000\$, viendo una reducción en el costo del 90% sin considerar mejoras a futuro.

El sistema posee una estabilidad con respecto al tiempo de duración en el que se realizaron las pruebas en cada escenario. Se puede concluir que la primera conexión, será la que más tiempo tarda en entregar una respuesta con respecto a los datos subsecuentes. Sin embargo, en el segundo escenario se tiene una gran variación en la duración de las pruebas, esto debido a que dependiendo de la velocidad de conexión que llegue al módulo ESP32-cam y de la calidad de internet que tenga el sistema, el mismo podría bajar su velocidad de respuesta hasta los 5 s por cada imagen, e inclusive presentar pérdida de paquetes de datos al no lograr entregar ningún frame al servidor, por lo que se concluye que la velocidad mínima para trabajar debería ser de 10 Mbps y al menos de 40 Mbps para trabajar de manera óptima y que además de la velocidad pueden existir factores externos, como el servicio del proveedor de internet que puede dificultar el correcto trabajo del prototipo.

Se especuló que el sistema tenga una duración de funcionamiento próxima al tiempo de gafas con características similares, es decir, un promedio de 8 horas. Debido a que cada uno de los elementos posee un consumo de energía mínimo y en conjunto no supera los 500 mA, el gasto energético es de 0.00138679 V, por lo tanto, se puede trabajar aproximadamente 5.46 horas continuas, con la batería de 3.7 V y 900 mAh. Se puede aumentar o disminuir este tiempo dependiendo de las características que se implemente en el módulo, o a su vez de los elementos con los que se trabaje, para los requerimientos establecidos y debido a la difícil situación para conseguir cierto tipo de elementos, el sistema es totalmente adecuado en tiempos.

Recomendaciones

Se recomienda iniciar el sistema un tiempo antes de empezar a utilizarlo, esto debido a que los primeros datos a ser procesados demoran un poco más en enviarse que los subsecuentes.

El sistema háptico tiene como requerimiento un posicionamiento correcto de la lente de la cámara, en caso de querer recolocar la misma en una nueva posición es necesario realizar una recalibración de las condiciones para el sistema háptico para su correcto funcionamiento. Ya que esta es la que enviará las coordenadas al algoritmo y puede dirigir a la persona a un lugar erróneo por dicho problema.

Se recomienda realizar una medida constante del voltaje disponible en la batería, ya que el mínimo voltaje de funcionamiento de los elementos es de 3.3 voltios, y si la batería disminuye de este límite, puede existir percances en el funcionamiento de cada uno de los elementos. Otra opción viable es realizar la electrónica adecuada para que se regule la energía de entrada al voltaje que se desea y en el caso de disminuir de un valor umbral, que esta se apague.

Existen varias librerías que poseen mejores características de identificación que la de face-api, pero tienen un costo después de un cierto número de iteraciones, muchas de estas librerías poseen una mayor exactitud y si se tiene los recursos es recomendable tomar en cuenta estas opciones. Inclusive existen algunos dispositivos que realizan la identificación por medio de sensores que capaces de extraer características (Segarra, Caballeros, Aguilar, Samà, & Rodríguez-Martín, 2019) (Grijalva, Cueva, Ramírez, & G, 2019), en el caso de tener este recurso al alcance se puede apreciar como otra opción.

En cuanto a la ergonomía del sistema completo, se recomienda construir un armazón completo desde el principio, ya que se puede ir añadiendo buses que permitan la entrada de cables de alimentación y de datos. Esto puede aumentar la comodidad y

mejorar la experiencia del usuario. En general esto se recomienda diseñarlo y realizar una impresión 3d.

Referencias

- Aggarwal, C. (2018). *Neural Networks and Deep Learning*. New York: Springer.
- Aguilar, W. G., & Angulo, C. (2014). Real-time video stabilization without phantom movements for micro aerial vehicles. *EURASIP Journal on Image and Video Processing*, 1, 1-13.
- Aguilar, W. G., & Angulo, C. (2016). Real-Time Model-Based Video Stabilization for Microaerial Vehicles. *Neural Processing Letters*, 43(2), 459-477.
- Aguilar, W. G., & Morales, S. (2016). 3D Environment Mapping Using the Kinect V2 and Path Planning Based on RRT Algorithms. *Electronics*, 5(4), 70.
- Aguilar, W. G., Álvarez, L., Grijalva, S., & Rojas, I. (2019). Monocular Vision-Based Dynamic Moving Obstacles Detection and Avoidance. In *Lecture Notes in Computer Science* (pp. 386-398).
- Aguilar, W. G., Angulo, C., & Costa-Castello, R. (2017). Autonomous Navigation Control for Quadrotors in Trajectories Tracking. In *Lecture Notes in Computer Science* (pp. 287-297).
- Aguilar, W. G., Angulo, C., & Pardo, J. A. (2017). Motion intention optimization for multirotor robust video stabilization. *Chilean Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*. Pucón, Chile.
- Aguilar, W. G., Casaliglla, V. P., & Pólit, J. L. (2017). Obstacle Avoidance Based-Visual Navigation for Micro Aerial Vehicles. *Electronics*, 6(1), 10.
- Aguilar, W. G., Cobeña, B., Rodríguez, G., Salcedo, V. S., & Collaguazo, B. (2018). SVM and RGB-D Sensor Based Gesture Recognition for UAV Control. *International*

Conference on Augmented Reality, Virtual Reality and Computer Graphics (pp. 713-719). Springer.

Aguilar, W. G., Luna, M. A., Moya, J. F., Abad, V., Parra, H., & Ruiz, H. (2017). Pedestrian detection for UAVs using cascade classifiers with meanshift. *IEEE 11th international conference on semantic computing (ICSC)*.

Aguilar, W. G., Luna, M. A., Moya, J. F., Abad, V., Ruiz, H., Parra, H., & Angulo, C. (2017). Pedestrian Detection for UAVs Using Cascade Classifiers and Saliency Maps. In *Lecture Notes in Computer Science* (pp. 563-574).

Aguilar, W. G., Luna, M. A., Ruiz, H., Moya, J. F., Luna, M. P., Abad, V., & Parra, H. (2017). Statistical Abnormal Crowd Behavior Detection and Simulation for Real-Time Applications. In *Lecture Notes in Computer Science* (pp. 671-682).

Aguilar, W. G., Manosalvas, J. F., Guillén, J. A., & Collaguazo, B. (2018). Robust Motion Estimation Based on Multiple Monocular Camera for Indoor Autonomous Navigation of Micro Aerial Vehicle. *International Conference on Augmented Reality, Virtual Reality and Computer Graphics* (pp. 547-561). Springer.

Aguilar, W. G., Quisaguano, F. J., Alvarez, L. G., Pardo, J. A., & Zahira, P. (2018). Monocular depth perception on a micro-UAV using convolutional neuronal networks. In *Lecture Notes in Computer Science* (pp. 392-397).

Aguilar, W. G., Quisaguano, F. J., Rodríguez, G. A., Alvarez, L. G., Limaico, A., & S, S. D. (2018). Convolutional neuronal networks based monocular object detection and depth perception for micro UAVs. In *Lecture Notes in Computer Science* (pp. 401-410).

- Aguilar, W. G., Rodríguez, G. A., Álvarez, L., Sandoval, S., Quisaguano, F., & Limaico, A. (2017). On-Board Visual SLAM on a UGV Using a RGB-D Camera. In *Lecture Notes in Computer Science* (pp. 298-308).
- Aguilar, W. G., Rodríguez, G. A., Álvarez, L., Sandoval, S., Quisaguano, F., & Limaico, A. (2017). Real-Time 3D Modeling with a RGB-D Camera and On-Board Processing. In *Lecture Notes in Computer Science* (pp. 410-419).
- Aguilar, W. G., Rodríguez, G. A., Álvarez, L., Sandoval, S., Quisaguano, F., & Limaico, A. (2017). Visual SLAM with a RGB-D Camera on a Quadrotor UAV Using on-Board Processing. In *Lecture Notes in Computer Science* (pp. 596-606).
- Aguilar, W. G., Salcedo, V., Sandoval, D., & Cobeña, B. (2017). Developing of a Video-Based Model for UAV Autonomous Navigation. *Communications in Computer and Information Science*, 94-105.
- Alvarez, M. (2019, 02 25). *desarrolloweb*. Retrieved from <https://desarrolloweb.com/articulos/que-es-react-motivos-uso.html>
- Amaguaña, F., Collaguazo, B., Tituaña, J., & Aguilar, W. G. (2018). Simulation System Based on Augmented Reality for Optimization of Training Tactics on Military Operations. *International Conference on Augmented Reality, Virtual Reality and Computer Graphics* (pp. 394-403). Springer.
- Andrea, C. C., Byron, J. Q., Jorge, P. I., Inti, T. C., & Aguilar, W. G. (2018). Geolocation and Counting of People with Aerial Thermal Imaging for Rescue Purposes. *International Conference on Augmented Reality, Virtual Reality and Computer Graphics* (pp. 171-182). Springer.

- Ayoze Castillo, A. (2017). *Curso de Programación Web JavaScript, Ajax y jQuery*. IT Campus Academy.
- Basantes, J., Godoy, L., Carvajal, T., Castro, R., Toulkeridis, T., Fuentes, W., . . . Addison, A. (2018). Capture and processing of geospatial data with laser scanner system for 3D modeling and virtual reality of Amazonian Caves. *IEEE Ecuador Technical Chapters Meeting (ETCM)*. Samborondón, Ecuador.
- Battery, L.-p. (2006). *Type:063048 860mAh*.
- BBC, M. (2018). Retrieved from <https://www.bbc.com/mundo/noticias-43003860>
- Bellido, S. S. (2011). *Arquitectura Mongo DB*.
- Blunsom, P., Grefenstette, E., & Kalchbrenner, N. (2014). *A Convolutional Neural Network for Modelling Sentences*. Oxford: University of Oxford.
- Boduch, A. (2017). *React and React Native*. Mumbai: Packt.
- Britz, D. (2015). *Understanding Convolutional Neural Networks for NLP*. Retrieved from WildML: <http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/>
- Bruce, V. (1999). *Identification of human faces*. IEE Conference Publication.
- Cabras, P., Rosell, J., Pérez, A., Aguilar, W. G., & Rosell, A. (2011). Haptic-based navigation for the virtual bronchoscopy. *18th IFAC World Congress*. Milano, Italy.
- Cadena Moreano, J. A., Montaluisa Pulloquina, R. H., Flores Lagla, G. A., Chancúsig Chisag, J. C., & Guaypatín Pico, O. A. (2017). *Reconocimiento Facial Con Base En Imágenes*. Revista Boletín Redipe.

- Caiza, O., Aguilar, W. G., Albán, P., & Fernández, Y. (2020). Kinect and Manipulator-Based Sample Collection System for Military Robot. In *Smart Innovation, Systems and Technologies* (pp. 75-87).
- Calderón, M., Aguilar, W. G., & Merizalde, D. (2020). Visual-Based Real-Time Detection Using Neural Networks and Micro-UAVs for Military Operations. In *Smart Innovation, Systems and Technologies* (pp. 55-64).
- Cardona, M. (2016, 10 14). *IEBS*. Retrieved from <https://www.iebschool.com/blog/firebase-que-es-para-que-sirve-la-plataforma-desarrolladores-google-seo-sem/>
- Carleto, D. (2018). *Face-api.js: JavaScript Face Recognition Leveraging TensorFlow.js*. Retrieved from Github: <https://www.infoq.com/news/2018/11/faces-api-js/>
- Chavez, A. (2019). *Una compañía de los Emiratos Árabes están creando lentes capaces de reconocimiento facial*. Retrieved from <https://www.merca20.com/una-compania-de-los-emiratos-arabes-estan-creando-lentes-capaces-de-reconocimiento-facial/>
- Çiço, B., & Nishani, E. (2017). *Computer vision approaches based on deep learning and neural networks: Deep neural networks for video analysis of human pose estimation*. Bar: IEEE.
- Conde Ortiz, D. (2018). *Inteligencia artificial con TensorFlow para predicción de comportamientos*. Sevilla: Escuela Técnica Superior de Ingeniería - Universidad de Sevilla.
- Danielsson, W. (2016). *React Native application development*. Linköpings: Linköpings universitet.

Diario El Universo. (2019). Delitos en aumento en distintas zonas del país. Guayaquil: Diario El Universo.

Eisenman, B. (2016). *Learning React Native*. Gravenstein: O'Reilly.

EL UNIVERSO. (2018, Julio 16). Suplantación de identidad genera juicios por deudas no adquiridas en Ecuador. *EL UNIVERSO*, pp. 1-2.

El Universo. (2019). *Delitos en aumento en distintas zonas del país | Seguridad | Noticias*. Retrieved from El Universo: <https://www.eluniverso.com/noticias/2019/08/04/nota/7455459/delitos-aumento-distintas-zonas-pais>

Face2Gene. (2020). *Face2Gene is a suite of phenotyping applications that facilitate comprehensive and precise genetic evaluations*. Retrieved from Face2Gene: <https://www.face2gene.com/>

FACEFIRST. (2020). *About FACEFIRST*. Retrieved from <https://www.facefirst.com/>

FACEFIRST. (2020). *FACEFIRST*. Retrieved from <https://www.facefirst.com/blog/21-mejores-aplicaciones-de-reconocimiento-facial-para-2018/>

FacePhi. (2020). *FacePhi*. Retrieved from <https://www.facephi.com/en/content/about-facephi/>

Flanagan, D. (2013). *JavaScript O guia definitivo*. São Paulo: Bookman.

Freund, Y., & Schapire, R. (1996). *Experiments with a New Boosting Algorithm*. Murray Hill: Machine Learning: Proceedings of the Thirteenth International Conference.

FTDI. (2012). *Ft232R Interface Usb Uart Ic*. Technology.

- Galarza, J., Pérez, E., Serrano, E., Tapia, A., & G, A. W. (2018). Pose estimation based on monocular visual odometry and lane detection for intelligent vehicles. In *Lecture Notes in Computer Science* (pp. 562-566).
- Galindo, R., Aguilar, W. G., & Reyes Ch, R. P. (2019). Landmark based eye ratio estimation for driver fatigue detection. In *Lecture Notes in Computer Science* (pp. 565-576). Germany: Springer.
- Gargouri, F., & Abbes, H. (2016). *Big Data Integration: A MongoDB Database and Modular Ontologies based Approach*. Tunisia: ELSEVIER.
- Goodman, D. (2002). *Dynamic HTML*. Gravenstein Highway North: O'Reilly.
- Google. (2020). *TensorFlow*. Retrieved from <https://www.tensorflow.org/?hl=es-419>
- Graterol, Y. D. (2014). *MongoDB en Español*.
- Grijalva, S., & Aguilar, W. G. (2019). Landmark-Based Virtual Path Estimation for Assisted UAV FPV Tele-Operation with Augmented Reality. In *Lecture Notes in Computer Science* (pp. 688-700).
- Grijalva, S., Cueva, G., Ramírez, D., & G, A. W. (2019). HSVM-Based Human Activity Recognition Using Smartphones. In *Lecture Notes in Computer Science* (pp. 217-228).
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep residual learning for image recognition*. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition.
- Hernández, R. G. (2010). *Estudio De Técnicas De Reconocimiento Facial*. Informacion Tecnologica.

- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., . . . Adam, H. (2017). *MobileNets: Efficient convolutional neural networks for mobile vision applications*. arXiv.
- Howard, A., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., . . . Adam, H. (2017). *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision*.
- Huang, S.-M., Tung, S.-C., Lin, Y.-S., Yen, C., & Li, W.-J. (2018). *JustIoT Internet of Things based on the Firebase real-time database*. Yunlin: IEEE.
- Ioana Agapin, L., Radulescu, F., & Boicea, A. (2012). *MongoDB vs Oracle -- Database Comparison*. Bucharest: IEEE.
- Jara-Olmedo, A., Medina-Pazmiño, W., Mesías, R., Araujo-Villaroel, B., Aguilar, W. G., & Pardo, J. A. (2018). Interface of Optimal Electro-Optical/Infrared for Unmanned Aerial Vehicles. In *Smart Innovation, Systems and Technologies* (pp. 372-380).
- Jara-Olmedo, A., Medina-Pazmiño, W., Tozer, T., Aguilar, W. G., & Pardo, J. A. (2018). E-services from Emergency Communication Network: Aerial Platform Evaluation. *International Conference on eDemocracy & eGovernment (ICEDEG)* (pp. 251-256). IEEE.
- Jiang, X., & Lu, Z. (2018). *Deep Coupled ResNet for Low-Resolution Face Recognition*. IEEE.
- Konishi, Y., Hanzawa, Y., Kawade, M., & Hashimoto, M. (2016). *Fast 6D Pose Estimation Using Hierarchical Pose Trees*.
- Lawrence, S., Lee Giles, C., Chung Tsoi, A., & Back, A. (1997). *Face recognition: a convolutional neural-network approach*. IEEE.

- Liu, W., Anguelov, A., Erhan, D., Szegedy, C., Reed, S., Scott, C.-Y., & Berg, A. (2016). *SSD: Single Shot MultiBox Detector*. Springer.
- Loncomilla, P. (2017). *Deep learning : Redes convolucionales 1 Introducción*.
- Mechatronics, N. (2020). *ESP32-CAM con cámara OV2640 - ESP32 WiFi*. Retrieved from <https://www.naylampmechatronics.com/espressif-esp/700-esp32-cam-con-camara-ov2640-esp32-wifi.html>
- Merizalde, D., Aguilar, W. G., & Calderón, M. (2020). Autonomous Navigation Based on Proportional Controller with GPS Setpoint for UAV in External Environments. In *Smart Innovation, Systems and Technologies* (pp. 89-99).
- MongoDB. (2020). *La base de datos líder del mercado para aplicaciones modernas | MongoDB*. Retrieved from <https://www.mongodb.com/es>
- Moroney, L. (2017). *The Firebase Realtime Database*. Seattle: Springer.
- Mosquera Victoria, C. A., & Dussan Narvaez, G. A. (2020). *Detección y seguimiento de múltiples objetos en tiempo real para vehículos autónomos*. Retrieved from <https://red.uao.edu.co/bitstream/10614/12295/5/T09147.pdf>
- MSI. (2020). *GE62VR 7RF APACHE PRO*. Retrieved from MSI: <https://es.msi.com/Laptop/GE62VR-7RF-Apache-Pro>
- Mühler, V. (2020). *face-api.js*. Retrieved from <https://justadudewhohacks.github.io/face-api.js/docs/index.html>
- Mundo, B. (2018). *Así son las nuevas gafas con reconocimiento facial que usa la policía en China para capturar sospechosos*. Retrieved from <https://www.bbc.com/mundo/noticias-43003860>

- Murillo Mejía, J. H., & Jiménez Arboleda, H. D. (2019). *Aplicación de un algoritmo de optimización secuencial mínima*. Bogotá: Universidad cooperativa de Colombia.
- Nalini, N., & Karan, M. (2017). *Facial Recognition Enabled Smart Door Using Microsoft Face API*. Vellore: VIT University.
- NEC. (2020). *A brief history of Facial Recognition*. Retrieved from <https://www.nec.co.nz/market-leadership/publications-media/a-brief-history-of-facial-recognition/>
- Nian, Q., Zhang, H., Zhao, Y., Qu, Z., & Yuan, L. (2017). *A convolutional neural network based on TensorFlow for face recognition*. Chongqing: IEEE.
- Nilsson, N. (1980). *Principles of Artificial Intelligence*. Palo Alto: Morgan Kaufmann.
- Orbea, D., Moposita, J., Aguilar, W. G., Paredes, M., León, G., & Jara-Olmedo, A. (2017). Math Model of UAV Multi Rotor Prototype with Fixed Wing Aerodynamic Structure for a Flight Simulator. In *Lecture Notes in Computer Science* (pp. 199-211).
- Orbea, D., Moposita, J., Aguilar, W. G., Paredes, M., Reyes, R. P., & Montoya, L. (2017). Vertical take off and landing with fixed rotor. *Chilean Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*. Pucón, Chile.
- Pardo, J. A., Aguilar, W. G., & Toulkeridis, T. (2017). Wireless communication system for the transmission of thermal images from a UAV. *Chilean Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*. Pucón, Chile.
- Parreño Escobar, J. F. (2020). *Implementación de un prototipo para contar personas en video capturado en tiempo real utilizando técnicas de aprendizaje profundo*.

- Pérez Nasser, J. (2019). *Análisis de tráfico vehicular mediante vision artificial*.
- Pizarro Justel, J. T. (2017). *Detección y clasificación de diferentes formas eritrocitarias anómalas mediante redes neuronales profundas*.
- Ponce Gallegos, J. C., Torres Soto, A., Quezada Aguilera, F. S., Silva Sprock, A., Martínez Flor, E. U., Casali, A., . . . Ornelas Zapata, F. J. (2014). *Inteligencia Artificial*.
- Ribes Alba, P. (2011). *Manual de JavaScript - Formación para el empleo*.
- Rosebrock, A. (2017). *Facial landmarks with dlib, OpenCV, and Python - PyImageSearch*.
- Rouhiainen, L. (2018). *Inteligencia artificial 101*.
- Rusell, S., & Norvig, P. (2004). *Inteligencia Artificial. Un Enfoque Moderno*.
- Salcedo, V. S., Aguilar, W. G., Cobeña, B., Pardo, J. A., & Zahira, P. (2018). On-board target virtualization using image features for UAV autonomous tracking. In *Lecture Notes in Computer Science* (pp. 384-391).
- Saldarriaga Quintero, M. (2020). *Reconocimiento de sujetos y caracterización de sus emociones a partir de sus expresiones faciales usando algoritmos de reconocimiento facial*.
- SANDOROBOTICS. (2019). *Micro Motor Vibrador Tipo Moneda 1027 – 3V*.
- Segarra, D., Caballeros, J., Aguilar, W. G., Samà, A., & Rodríguez-Martín, D. (2019). Orientation Estimation Using Filter-Based Inertial Data Fusion for Posture Recognition. In *Lecture Notes in Computer Science* (pp. 220-233).
- Shen, L., & Bai, L. (2004). *AdaBoost Gabor Feature Selection for Classification* . Nottingham: University of Nottingham.

- Sheng, T., Feng, C., Zhuo, S., Zhang, X., Shen, L., & Aleksic, M. (2018). *A Quantization-Friendly Separable Convolution for MobileNets*. Williamsburg: IEEE.
- Shukla, N., & Fricklas, K. (2018). *Machine Learning with TensorFlow*. Manning.
- Sirovich, L., & Kirby, M. (1987). *Low-dimensional procedure for the characterization of human faces*.
- Soto Espinoza, J. J. (2018). *Desarrollo de un sistema embebido para detectar en tiempo real la presencia de personas en ambientes controlados*. Sangolquí: Universidad de las Fuerzas Armadas - ESPE.
- Targ, S., Almeida, D., & Lyman, K. (2016). *Resnet in Resnet: Generalizing Residual Architectures*. Nueva York: Cornell University.
- Technologies, O. (2006). *Datasheet - OV2640 Color CMOS UXGA (2.0 MegaPixel) CAMERACHIP with OmniPixel2 Technology*.
- Technology Co, S. A.-T. (2017). *ESP32-CAM Wi-Fi +BT SoC Module V1.0*.
- Tenezaca, D., Canchignia, C., Aguilar, W. G., & Mendoza, D. (2019). Implementation of Dubin Curves-Based RRT* Using an Aerial Image for the Determination of Obstacles and Path Planning to Avoid Them During Displacement of the Mobile Robot. In *Smart Innovation, Systems and Technologies* (pp. 205-215).
- Togashi, K., Gian Do, R. K., Nishio, M., & Yamashita, R. (2018). *Convolutional neural networks: an overview and application in radiology*. Springer.
- Tolocka, P. (2020). *Probando el streaming de video con ESP32-CAM y Arduino IDE*. Retrieved from <https://www.profetolocka.com.ar/2020/07/15/probando-el-streaming-de-video-con-esp32-cam-y-arduino-ide/>

- Trabajo, I. N. (2019). *Gafas de protección frente a la radiación solar*.
- Turk, M., & Pentland, A. (1991). *Eigedces for Recognition*.
- Vincent, J. (2019, 06 10). *THE VERGE*. Retrieved from <https://www.theverge.com/2019/6/10/18659660/facial-recognition-smart-glasses-sunglasses-surveillance-vuzix-nntc-uae>
- Vishwanathan, S., & Murty, N. (2002). *SSVM: a simple SVM algorithm*. Honolulu: IEEE.
- Wang, R. (2012). *AdaBoost for Feature Selection, Classification and Its Relation with SVM*. Yongchuan: International Conference on Solid State Devices and Materials Science.
- Waring, C., & Liu, X. (2005). *Face detection using spectral histograms and SVMs*. IEEE.
- Wu, Z., Shen, C., & van den Hengel, A. (2019). *Wider or Deeper: Revisiting the ResNet Model for Visual Recognition*. Adelaide: University of Adelaide.
- Yao, T., Zhang, Q., Wu, X., & Lin, X. (2020). *Image Recognition Method of Defective Button Battery Base on Improved MobileNetV1*. Springer.
- Yi, J., Wu, P., & Metaxas, D. (2019). *ASSD: Attentive single shot multibox detector*. Piscataway: ELSEVIER.
- Zhao, W., Chellappa, R., Phillips, P. J., & Rosenfeld, A. (2003). *Face recognition: A literature survey*.
- Zurita, M., Aguilar, W. G., & Enríquez, V. (2019). Toward the Development of Surveillance and Reconnaissance Capacity in Ecuador: Geolocation System for Ground Targets Based on an Electro-Optical Sensor. In *Smart Innovation, Systems and Technologies* (pp. 123-133).

