

**DISEÑO E IMPLEMENTACIÓN DE UN DISPOSITIVO PARA RECOPILAR
INFORMACIÓN DEMOGRÁFICA COMO
EDAD Y GÉNERO MEDIANTE EL USO DE VISIÓN POR COMPUTADORA**

ANDRES FELIPE PINEDA TORRES
MICHEL STEFANNY IBAÑEZ LOZANO

REALIZADO CON LA ASESORÍA DE:
EMILIO ALEJANDRO SANCHEZ PINZÓN

UNIVERSIDAD EL BOSQUE
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA
MAYO, 2023

UNIVERSIDAD EL BOSQUE
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA ELECTRÓNICA

ÁREA DE ÉNFASIS: BÁSICA - PROGRAMACIÓN

**DISEÑO E IMPLEMENTACIÓN DE UN DISPOSITIVO PARA RECOPILAR
INFORMACIÓN DEMOGRÁFICA COMO
EDAD Y GÉNERO MEDIANTE EL USO DE VISIÓN POR COMPUTADORA**

ANDRÉS FELIPE PINEDA TORRES

MICHEL STEFANNY IBAÑEZ LOZANO

REALIZADO CON LA ASESORÍA DE:

EMILIO ALEJANDRO SANCHEZ PINZÓN

Página de Aprobación. Inclusión de Acta de grado.

NOTA DE SALVEDAD

Según el artículo 37 del 14 de diciembre de 1989 del acuerdo 017, “La Universidad El Bosque, no se hace responsable de los conceptos emitidos por los investigadores en su trabajo, solo velará por el rigor científico, metodológico y ético del mismo en aras de la búsqueda de la verdad y la justicia”.

AGRADECIMIENTOS

Queremos expresar nuestros más sinceros agradecimientos a todas las personas que nos apoyaron en la realización de este proyecto de grado de ingeniería electrónica. En primer lugar, a nuestro director, Emilio Sánchez, quien nos brindó su invaluable orientación y experiencia en cada etapa del proyecto, desde la formulación de la idea inicial hasta la presentación final.

Agradecemos especialmente a la Universidad El Bosque, por permitirnos el acceso a sus instalaciones, lo que hizo posible el éxito del proyecto.

También queremos agradecer a nuestros familiares y amigos por el apoyo incondicional y motivación constante.

Les agradecemos inmensamente a todos ustedes, porque el día de hoy podemos presentar este trabajo con orgullo y satisfacción. Esperamos que este proyecto contribuya de alguna manera al desarrollo de la ingeniería electrónica y a la sociedad en general.

RESUMEN

En este documento de proyecto de grado se presentó el diseño e implementación de un prototipo de dispositivo adaptable a las carteleras digitales conformado por una cámara que monitorea un punto en el cual se concentran las personas que visualizan la cartelera digital ofrecida por la empresa MA Electronika SAS en la ciudad de Bogotá. La información que se toma con la cámara será almacenada y procesada con un ordenador el cual indicará la medición audiencia con software que opera en conjunto con el dispositivo.

Durante el desarrollo de este proyecto de grado se usó la metodología CDIO, la cual consiste en generar una unión entre el conocimiento adquirido con la práctica y se divide en las siguientes fases: Conceptualización del proyecto, donde se realizó la investigación pertinente para iniciar con el proyecto, declaración de objetivos, se acordaron las limitaciones y el alcance que tiene el proyecto. Para la fase de diseño se realizó el análisis de requerimientos con respecto a las necesidades del usuario final, diseño del programa, análisis de los dispositivos a usar y sus diseños físicos, para la etapa de implementar, se ejecutaron los modelos, se realizaron modificaciones de estos y se llevan a cabo las pruebas del funcionamiento. El prototipo de dispositivo electrónico desarrollado e implementado permite al usuario final obtener información demográfica de edad y género, se basó en lograr obtener un dispositivo funcional y de fácil uso esto con el fin de evidenciar que efectivamente hiciera la detección de edad y género, para finalizar se ejecuta la etapa de operación, en la que se realizan pruebas de reconocimiento de edad, pruebas de reconocimiento de género, pruebas de inserción en base de datos de la información, el resultado final del proyecto fue un dispositivo electrónico con su respectiva interfaz gráfica que permite recopilar información demográfica de edad y género de forma más eficiente, la presentación y entrega final del mismo. Al realizar el diseño e implementación de este proyecto, se detecta edad y género con un modelo que maneja una sola entrada y 2 salidas, en las cuales se evidencia una precisión para género del 90% y un puntaje de R2 para la edad de 0.42, al generar la captura de esta información por cada uno de los individuos la muestra en un dashboard separando edad y género.

Palabras Clave: Reconocimiento Facial, CDIO, CNN, Redes Neuronales, Visión por computadora, Regresión.

ABSTRACT

This degree project document presented the design and implementation of a prototype of a device adaptable to digital billboards consisting of a camera that monitors a point where people who view the digital signage offered by the company MA Electronika SAS in the city of Bogota are concentrated. The information taken with the camera will be stored and processed with a computer which will indicate the audience measurement with software that operates in conjunction with the device.

During the development of this degree project the CDIO methodology was used, which consists of generating a union between the knowledge acquired with the practice and is divided into the following phases: Conceptualization of the project, where the relevant research was conducted to start with the project, statement of objectives, limitations and scope of the project were agreed. For the design phase, the analysis of requirements with respect to the needs of the end user, program design, analysis of the devices to be used and their physical designs, for the implementation stage, the models were executed, modifications were made and the operation tests were carried out. The prototype of the electronic device developed and implemented allows the end user to obtain demographic information of age and gender, it was based on obtaining a functional and easy to use device in order to demonstrate that it effectively made the detection of age and gender, and finally the operation stage is executed, The final result of the project was an electronic device with its respective graphic interface that allows the collection of age and gender demographic information in a more efficient way, the presentation and final delivery of the device. In the design and implementation of this project, age and gender are detected with a model that handles a single input and 2 outputs, which shows an accuracy for gender of 90% and an R2 score for age of 0.42, to generate the capture of this information for each of the individuals shown on a dashboard separating age and gender.

Keywords: Facial recognition, CDIO, CNN, Neural Networks, Computer Vision, Regression.

GLOSARIO DE TÉRMINOS

- Data set:** Colección de datos interrelacionados.[1]
- Inteligencia artificial:** Capacidad de un instrumento mecánico de efectuar operaciones normalmente propias de la inteligencia humana.[2]
- Predicción:** Anunciar por revelación, conocimiento fundado, intuición o conjetura algo que ha de suceder.[3]
- Software:** Programas, procedimientos, reglas y cualquier documentación y datos asociados concernientes a la operación de un sistema de tratamiento de la información. [4]
- Hardware:** Componentes eléctricos, magnéticos, electrónicos, electromecánicos y mecánicos [4]
- Dashboard:** Interfaz gráfica de usuario que actúa como un menú central [4]

TABLA DE CONTENIDOS

Pág.

AGRADECIMIENTOS	ix
RESUMEN	x
ABSTRACT	xi
GLOSARIO DE TÉRMINOS	xii
LISTA DE TABLAS	xvi
LISTA DE FIGURAS.....	xvii
1. INTRODUCCIÓN	22
2. PLANTEAMIENTO DEL PROBLEMA	23
2.1. Antecedentes y Estado del Arte	23
2.1.1. Nivel Internacional	23
2.1.2. Nivel Nacional	24
2.2. Descripción y Formulación del Problema	25
2.3. Justificación	26
2.4. Objetivos.....	27
2.4.1. Objetivo General.....	27
2.4.2. Objetivos Específicos	27
2.5. Alcance y Limitaciones del Proyecto	27
3. MARCO DE REFERENCIA.....	28
3.1. Marco Teórico o Conceptual	28
3.1.1. Inteligencia artificial	28
3.1.2. Visión por computador	29
3.1.3. Reconocimiento Facial	30
3.1.4. Redes neuronales Artificiales	31
3.1.5. Red neuronal Convolucional	32
3.1.6. Red Neuronal Recurrente.....	33
3.1.7. Capas.....	33

3.1.7.1.	Conv2D	34
3.1.7.2.	Activation ReLu.....	35
3.1.7.3.	BatchNormalization	35
3.1.7.4.	MaxPooling2D.....	36
3.1.7.5.	Dropout	37
3.1.7.6.	Flatten	37
3.1.7.7.	Dense – Full Connected.....	38
3.1.8.	Modelos de IA	38
3.1.9.	Entrenamiento de un modelo	39
3.1.10.	Modelos Preentrenados.....	39
3.1.11.	Aprendizaje Profundo	40
3.1.12.	Aprendizaje Automático	40
3.1.13.	Open CV	41
3.1.14.	Parámetros de medición en un modelo	41
3.1.14.1.	Función de pérdidas	41
3.1.14.2.	Precisión.....	41
3.1.14.3.	Regresión	42
3.1.14.4.	R2Score.....	42
3.1.15.	Cámara	43
3.1.16.	Interfaz Serie para Cámaras (CSI)	43
3.2.	Marco Legal o Normativo.....	46
4.	DESARROLLO DEL PROYECTO DE GRADO.....	46
4.1.	Requerimientos	46
4.2.	Metodología del Diseño.....	47
4.3.	Descripción Técnica del Producto	50
4.3.1.	Levantamiento de información	50
4.3.2.	Hardware.....	51
4.3.2.1.	Subsistema de edad y género	52
4.3.2.2.	Subsistema control de fragancias.....	56
4.3.3.	Software	56
4.3.3.1.	Diagrama de flujo del funcionamiento general del software	57

4.3.3.2.	Funcionamiento Algoritmo detección facial.....	58
4.3.3.3.	Estimación de edad y género	61
4.3.3.4.	Estructura de los 5 Modelo Diseñados	68
4.3.3.5.	Funcionamiento Algoritmo Detección de edad y genero	75
4.3.3.6.	Base de datos	76
4.3.3.7.	Diseño de aplicación web	77
5.	RESULTADOS Y ANÁLISIS DE RESULTADOS	78
5.1.	Resultados de implementación	78
5.1.1.	Algoritmo de Reconocimiento Facial	78
5.1.2.	Algoritmo de Edad y genero	79
5.1.3.	Modelos para Edad	83
5.2.	Análisis de cumplimiento requerimientos	86
5.3.	Pruebas de Funcionamiento	87
6.	CONCLUSIONES	92
7.	RECOMENDACIONES.....	92
8.	REFERENCIAS BIBLIOGRÁFICAS	93
9.	ANEXOS	101
	ANEXO 1: CÓDIGO FUENTE DEL ULTIMO MODELO ENTRENADO.....	101
	ANEXO 2: VIDEO PRUEBA PROTOTIPO 1	115

LISTA DE TABLAS

Tabla	Pág.
Tabla 1. Requerimientos Funcionales.....	46
Tabla 2. Requerimientos De Calidad	47
Tabla 3. Requerimientos De Gerencia	47
Tabla 4. Requerimientos Restrictivos	47
Tabla 5. Descripción características dispositivos investigados.	54
Tabla 6. Descripción características Cámara Raspberry Pi Camera 1.3.....	55
Tabla 7. Distribución de edad	65
Tabla 8. Distribución género data set	66
Tabla 9. Pruebas de predicción de edad y género	81
Tabla 10.Resultados obtenidos de género con prueba de 2304 imágenes del data set	85
Tabla 11.Tabla Cumplimiento Requerimientos Funcionales	86
Tabla 12. Prueba de una imagen con 3 rellenos (10,20 y 30px), la edad real es de 21 años.....	87
Tabla 13 Predicción de Edad con rostro en diferentes posiciones	88
Tabla 14.Pruebas con cámara USB.....	89
Tabla 15. Pruebas con cámara Raspberry Camera V1.3.....	91

LISTA DE FIGURAS

Figura	Pág.
figura 1. Red neuronal artificial tomada de C. B. Carrión, “REDES CONVOLUCIONALES,” Universidad de Sevilla, 2020.....	32
figura 2. Estructura de una red neuronal convolucional Tomada de Red neuronal artificial tomada de C. B. Carrión, “REDES CONVOLUCIONALES,” Universidad de Sevilla, 2020....	33
figura 3. Proceso de convolución tomada de [30]	34
figura 4. Capa Conv2D usada en los modelos (tomada directamente del código)	34
figura 5. Función de activación ReLU tomada de [32].....	35
figura 6. Capa Activation usada en los modelos (tomada directamente del código).....	35
figura 7. Batch normalization función tomada de [34]	35
figura 8. Capa BatchNormalization usada en los modelos (tomada directamente del código).....	36
figura 9. Función maxpooling tomada de [36]	36
figura 10. Capa MaxPooling2D usada en los modelos (tomada directamente del código).....	36
figura 11. Imagen Función dropout tomada de [38]	37
figura 12. Capa Dropout usada en los modelos (tomada directamente del código)	37
figura 13. Imagen función flatten tomada de [40]	37
figura 14. Capa Flatten usada en los modelos (tomada directamente del código).....	37
figura 15. Arquitectura de Capa dense tomada de [42]	38

figura 16. Capa Dense usada en los modelos (tomada directamente del código).....	38
figura 17. Matriz de Confusión tomada de J. M. Heras, “Precision, Recall, F1, Accuracy en clasificación,” 2020.	42
figura 18. Fórmula de Precisión tomada de tomada de J. M. Heras, “Precision, Recall, F1, Accuracy en clasificación,” 2020.	42
figura 19. Regresión Lineal tomada de L. Fernando, “¿Qué es y para qué sirve una regresión lineal en machine learning?,” 2022.....	42
figura 20. Ejemplo de posición de cámara y posiciones a tener en cuenta imagen tomada de https://forum.huawei.com/enterprise/es/requisitos-de-instalaci%C3%B3n-para-la-c%C3%A1mara-de-reconocimiento-facial/thread/499177-100259	43
figura 21. Figura de como organización de pines de la cámara y a que corresponde cada uno tomada de [59]	44
figura 22. Cámara de raspberry pi imagen tomada de [60].....	45
figura 23. Estructura de la Metodología	48
figura 24. Esquema de Equipo de fragancias.....	51
figura 25. Diagrama de entradas y salidas de la implementación general	52
figura 26. Diagrama subsistema edad y genero	52
figura 27. Módulo de Procesamiento de imágenes	53
figura 28. sensor OV5647 tipo CMOS de 5 megapíxeles imagen tomada de[63].....	56
figura 29. Diagrama de Bloques Software General	57
figura 30. Diagrama de flujo del funcionamiento general del dispositivo.....	57
figura 31. Flujo detección de rostros	58

figura 32 - Características Haar . tomada de M. J. J. PAUL VIOLA, “Robust Real-Time Face Detection,” Sep. 10, 2001. Accessed: May 04, 2023.	59
figura 33. Representación de la superposición que se usa para detección de rostro en el algoritmo de Viola Jones tomada de [64]	59
figura 34. Importación de librerías de preprocesamiento de imágenes para modelo de detección de rostros	60
figura 35. Preprocesamiento de imágenes para modelo de detección de rostros implementado.....	60
figura 36. Ilustración de la arquitectura usada tomada de [65].....	61
figura 37. gráfica de porcentaje cantidad de imágenes por rango de edad	62
figura 38. Prueba con sujeto 1 y entrenamiento data set Adience	62
figura 39. Prueba con sujeto 2 y entrenamiento data set Adience	63
figura 40. Prueba con sujeto 3 y entrenamiento data set Adience	63
figura 41. Prueba en empresa con entrenamiento data set Adience.....	63
figura 42. Imagen de la maquina autoservicio donde se implementó el primer prototipo.....	64
figura 43. gráfica de porcentaje cantidad de imágenes por rango de edad	65
figura 44. Pirámide de población Colombia 2020 tomada de DANE, “Panorama sociodemográfico de la juventud en Colombia,” Sep. 20AD. Accessed: May 03, 2023. [Online].....	67
figura 45. Gráfica de división de data set para pruebas, validación y entrenamiento.....	67
figura 46. Modelo 3 con 5 capas convolucionales.....	68

figura 47. Modelo 1 diseñado con dos capas convolucionales	69
figura 48. Gráfica modelo 2 capas y con 10 épocas	69
figura 49. Modelo 2 diseñado con cuatro capas convolucionales.....	70
figura 50. Gráfica modelo 4 capas y con 100 épocas	71
figura 51. Modelo 3 aumentando capa para edad	71
figura 52. Gráfica Modelo 5 capas y con 1000 épocas	72
figura 53. Modelo 4 se agrega una capa convolucional adicional a la anterior para edad.....	73
figura 54. Gráfica Modelo 6 capas y con 100 épocas	73
figura 55. Gráfica Modelo con 7 capas y con 2000 épocas	74
figura 56. Modelo 5 con una capa convolucional adicional para la edad	74
figura 57. Código Función detección de rostros.	75
figura 58. Función para predecir edad y género	76
figura 59. Estructura de tabla en Base de datos	76
figura 60. Datos insertados en la base de datos	77
figura 61. Dashboard Diseñado con información consultada de la Base de datos	77
figura 62. Proceso realizado por algoritmo de Reconocimiento Facial	78
figura 63. Prueba preprocesamiento de imagen para algoritmo de reconcomiendo facial.....	78
figura 64. Normalización de imágenes al realizar detección de rostros.	79

figura 65. Prueba detección de rostros	79
figura 66. Prueba sujeto 1 acertada en datos de predicción	80
figura 67. Prueba sujeto 2 no acertado en datos de predicción	80
figura 68. Evidencia FPS dispositivo 8 núcleos	82
figura 69. Evidencia FPS dispositivo de 8 núcleos.....	83
figura 70. Grafica modelo 3 comparación de edad real y estimada.....	84
figura 71. Grafica modelo 5 comparación de edad real y estimada.....	84
figura 72. Modelo 3 porcentajes para predicción real y estimada de género.....	84
figura 73. Modelo 5 porcentajes para predicción real y estimada de género.....	85
figura 74. Imágenes DataSet Prueba edad real vs edad esperada tomada de [68]	86
figura 75. Prueba de predicción con dos sujetos al tiempo.....	89

1. INTRODUCCIÓN

En este documento se muestra el proceso realizado para generar el diseño e implementación de un dispositivo electrónico con su respectiva interfaz para recopilar información demográfica como género y edad, mediante la adquisición de imágenes.

Para algunos de los clientes de la empresa MA Electronika SAS que adquieren quioscos interactivos resulta crucial recopilar datos demográficos como género y edad de las personas que se acercan a ver la publicidad proyectada, adicional contabilizar el tiempo que tardan las personas en visualizar el contenido del quiosco, el género y edad del sujeto. Esto debido a que no tienen una manera para verificar si la publicidad que están utilizando es la adecuada y a qué tipo de población están llegando. Recopilar dicha información les permitiría establecer indicadores que determinen si es necesario realizarle mejoras a la campaña o redirigirla de acuerdo con el tipo de población.

Este proceso empieza con el adquisidor de datos, que se realizó mediante una cámara, la cual captura imágenes para entregarlas a un equipo que realiza el procesamiento de imagen mediante un modelo de redes convolucionales. Posteriormente, se extrae la información generada por estos modelos y se almacena en una base de datos.

El reconocimiento facial se viene trabajando desde hace un tiempo para diferentes áreas de la tecnología. En este caso, se busca implementar para la medición de audiencias, con el principal objetivo de realizar el reconocimiento de género y de edad de los rostros de los clientes. Para complementar eso se dispondrá del tiempo que dure la persona siendo reconocida para así llevar un barrido de la información a una base de datos y así obtener el tiempo que mira la cartelería digital donde se busca implementar.

La estructura del documento se organiza de la siguiente manera: en la segunda sección se presenta una revisión con respecto a la literatura relacionada con reconocimiento facial, sus aplicaciones con respecto al marketing y otros campos, la detección de edad y género, algunos proyectos con finalidades similares, con el fin de establecer un marco teórico sólido y una buena contextualización. Adicional se evidencia la descripción y formulación del problema, la justificación, los objetivos, el alcance y algunas limitaciones que se hallaron. A continuación, en la tercera sección se establece el marco teórico en el que se describen algunos conceptos importantes para establecer el contexto en el cual se desarrolla el proyecto,

también para tener la base teórica necesaria para entender la metodología e interpretar los resultados.

En la cuarta sección, se presenta el desarrollo del proyecto en el cual se describe detalladamente la metodología utilizada, incluyendo requerimientos, estructura y metodología de lo realizado, la recopilación de datos, descripción técnica del producto y diseño de este. Seguido, en la quinta sección se presenta y analizan los resultados obtenidos, seguidos de una explicación de todo lo conseguido, donde se encuentra también las pruebas realizadas. Por último, se presentarán las conclusiones generales del proyecto, junto con las recomendaciones y posibles direcciones futuras de investigación.

2. PLANTEAMIENTO DEL PROBLEMA

2.1. Antecedentes y Estado del Arte

2.1.1. Nivel Internacional

Según el artículo de la revista virtual MarketingDirecto.com, es importante para el marketing y para las empresas de publicidad hallar un sistema de medición eficiente y capaz de cumplir las diversas demandas de las empresas publicitarias. Por lo anterior, es necesario actualizar el método de recolección de datos, capaz de medir a las diversas audiencias. Este sistema de medición incorporaría novedades tecnológicas que puedan reconocer al cliente y sus necesidades, sin comprometer la precisión y la transparencia en la medición.[5]

Existen algunos sistemas de información que realizan este proceso como Audience Analytics de la empresa Beabloom [6] y Reconocimiento Facial de BOARDS [7]; el primero es chileno y el segundo es español. Además de estos a nivel mundial, hay muchos algoritmos y desarrollos que implementan este tipo de tecnología [8], [9]

En el artículo “People movement tracking based on estimation of ages and genders” los autores presentan un Sistema de personas en un sandbox. Esto incluye detectar una persona, estimando el género y la edad, y rastreando la ubicación en el área seleccionada. El sistema puede localizar instantáneamente a las personas basándose en su posición actual, calculando el sexo y la edad, mostrando su movimiento visualmente. Este sistema es útil para hacer estadísticas de su detención en ciertas partes del edificio.[10]

Para el proyecto final de la carrera de Ingeniería Informática los estudiantes Universidad Nacional del Litoral en Argentina realizaron un sistema de reconocimiento facial, el cual consiste en realizar el diseño y la implementación de un sistema integral para identificación de personas mediante el reconocimiento del rostro. Ellos realizaron una interfaz de usuario e instalaron una cámara web para adquirir un conjunto de imágenes y realizar una clasificación manual. Para implementar este sistema, los estudiantes desarrollaron un sistema con tres etapas que consisten, primero en la localización del rostro; segundo, la extracción de características y finalmente la clasificación de los datos.[11]

Para una tesis de grado de Ingeniería del Software un estudiante de la Universidad de Málaga en España realizó la clasificación de edad y género basada en las características faciales. Según su propuesta, el sistema no es invariante a la pose de la cara del sujeto a clasificar, para la edad clasificaciones desde menores de 15 años a mayores de 60.[12]

En la Universidad técnica de Ambato en Ecuador, para el programa de ingeniería electrónica y comunicaciones, uno de sus estudiantes realiza su trabajo de graduación con un prototipo de un sistema de control de acceso al personal de una lavadora de jeans, mediante reconocimiento facial, en el cual generan una base de datos con las imágenes de los empleados las cuales serán comparadas y permitirá o no el ingreso.[13]

Adicional se encuentran algunos algoritmos como lo son Open VINO que tiene un margen de error de 6.99 años y una precisión del 95.80% en estimación de género [14] y Caffe Net que tiene una precisión de edad: 45-73%, Género: 75-93% [15]. Con este proyecto se busca tener los mejores resultados para la población colombiana más específicamente, la población de la ciudad de Bogotá, por lo cual se van a tener en cuenta las posibles mejoras que se puedan hacer a los algoritmos existentes, optimizándolos para la población bogotana.

2.1.2. Nivel Nacional

Respecto al ámbito nacional, en Colombia se evidencia información sobre la efectividad de estos algoritmos que son desarrollados en otros continentes, y solo se puede nombrar a la empresa Claro [16] que tiene una solución similar en Colombia. Esta hace medición de audiencia con internet de las cosas, en los cuales incluyen datos en tiempo real, indicadores, seguimiento y métricas, esto con respecto directamente a la medición de audiencias.

En la Universidad Autónoma de Occidente, estudiantes de pregrado de Ingeniería Mecatrónica realizaron su proyecto de grado realizando reconocimiento de objetos en imágenes usando aprendizaje profundo. Para ello, desarrollaron una aplicación de aprendizaje profundo (*Deep Learning*) la cual identifica objetos en el laboratorio de automática de la universidad, haciendo uso de la técnica *transfer learning*. [17]

En la Universidad Católica de Colombia realizaron un trabajo de grado para obtener el título de ingeniero electrónico y de telecomunicaciones, realizando un sistema de reconocimiento facial para control de acceso a viviendas. Para el desarrollo de este sistema se realizó la recopilación de información, generaron análisis de requerimientos, tanto para el sistema como para los dispositivos usados, realizando también pruebas de funcionamiento y eficacia del sistema. Esto con el fin de crear una solución que disminuya los porcentajes de hurto utilizando la identificación por medio de biometría facial. [18]

El proyecto realizado por un estudiante la Universidad del Valle, de la facultad de ingeniería electrónica propone evaluar la viabilidad de generar la comercialización de un sistema de reconocimiento facial para el control de acceso a un conjunto residencial, manejan una base de datos de 50 imágenes inicialmente de los residentes.[19]

Otro estudiante de la carrera Ingeniería electrónica de la Universidad de los Andes realiza su proyecto de grado un sistema de recomendación de contenido por medio de reconocimiento facial, en el cual va a crear una base de datos que almacene las imágenes e información de los usuarios para si poder entrenar al modelo. [20]

Dado que en Colombia el desarrollo del reconocimiento facial está más enfocado a seguridad, en este proyecto se busca enfocar hacia el área de la publicidad y marketing.

2.2. Descripción y Formulación del Problema

La empresa MA Electronika SAS vende equipos de audio y video tales como pantallas LED, quioscos interactivos, turnos digitales, entre otros. Para uno de los clientes, que adquirió un quiosco interactivo, resulta crucial recopilar datos demográficos como género y edad de las personas que se acercan a ver la publicidad proyectada en el quiosco. Además de lo anterior, para la empresa es importante poder contabilizar el tiempo empleado en visualizar el contenido del quiosco. Estos datos son fundamentales para verificar que la publicidad que están utilizando es la adecuada, de acuerdo con la población que quieren impactar. Recopilar

género, edad y tiempo que se demora el cliente viendo el comercial, les permitiría establecer indicadores que determinen si es necesario realizarle mejoras a la campaña o redirigirla de acuerdo con el tipo de población.

Si bien lo anterior surge de la necesidad específica de un cliente, la empresa Electronika SAS considera que este prototipo se debe replicar para ofrecer un mejor servicio a otros clientes potenciales. De este modo, se busca que el prototipo a desarrollar cumpla con los requerimientos del cliente, que sea compacto y que sea compatible con los demás equipos que ya venden actualmente.

Teniendo en cuenta lo anterior, se busca diseñar e implementar un prototipo de dispositivo adaptable a los quioscos, conformado por una cámara, la cual estará monitoreando un punto en el cual se concentran las personas que visualizan el quiosco con información del cliente de la empresa MA Electronika SAS en la ciudad de Bogotá. La información que se adquiere con la cámara será procesada empleando un ordenador que en este caso será una Raspberry Pi, la cual indicará la medición de audiencia con un software diseñado e implementado que opera en conjunto con el dispositivo.

Adicional, se implementará un algoritmo de reconocimiento facial para que se pueda cumplir con la solicitud del cliente de la empresa. La información que se adquiriera no incluirá las imágenes adquiridas para el reconocimiento facial, razón por lo cual no es necesario solicitar algún tipo de permiso o consentimiento informado por parte de las personas que se acercan a consultar la cartelera digital. Únicamente se guardarán datos como la edad, el género y la cantidad de tiempo de visualización.

2.3. Justificación

La empresa MA Electronika SAS se enfoca en el área de servicios de quioscos interactivos o cartelera digital, de modo que recopilar información sobre datos demográficos como género y edad de las personas que visualizan sus carteleras digitales es de vital importancia para que sus clientes puedan hacer una publicidad correcta. Además, para los clientes de la empresa MA Electronika SAS es importante poder tener el género y el rango de edades que están visualizando las carteleras, adicional contabilizar el tiempo que el cliente toma en visualizar el anuncio publicado en la cartelera digital. En la medida en que ellos no tienen una manera para verificar si la publicidad que están utilizando es la adecuada ni a qué tipo de población

están llegando, desean recopilar dicha información para establecer indicadores que determinen si es necesario que sus clientes realicen mejoras a la campaña o redirigirlas de acuerdo con la población.

2.4. Objetivos

2.4.1. Objetivo General

Diseñar e implementar un dispositivo electrónico con su respectiva interfaz para recopilar información demográfica como género y edad.

2.4.2. Objetivos Específicos

1. Implementar un algoritmo de reconocimiento facial en los equipos electrónicos adecuados.
2. Diseñar e implementar un dispositivo electrónico, el cual se encargará de adquirir los datos demográficos de género y edad empleando visión por computadora.
3. Realizar pruebas de funcionamiento y analizar los resultados obtenidos.

2.5. Alcance y Limitaciones del Proyecto

El fin de este proyecto es llegar al diseño e implementación de un prototipo de dispositivo electrónico que reconozca factores demográficos como género y edad aproximada. Este dispositivo electrónico se va a dotar con una cámara y un ordenador para poder generar el reconocimiento, mostrar la clasificación de los datos. Además, se va a generar una interfaz para la recopilación y visualización de información.

Originalmente, este proyecto estaba destinado para un cliente en específico de la empresa Electronika MA SAS y se iba a implementar el dispositivo en un espacio cerrado. No obstante, la empresa consideró que este dispositivo se le podía ofrecer a otros clientes, de modo que se debe diseñar para que se pueda implementar en otros equipos de su portafolio. Esta variedad en los tipos de espacio puede generar algunas limitaciones inicialmente en la adquisición de la imagen con respecto al entorno. Cabe nombrar factores que podrían dificultar el reconocimiento facial tales como: la iluminación, contraste, brillo, contraluz,

niebla u otra. Además, el clima de Bogotá, que es considerado como impredecible, puede afectar también la adquisición de la imagen.

Para realizar el reconocimiento facial también puede haber limitación con respecto al vestuario el cual dificulta el reconocimiento facial, tales como gorras, gafas de sol, bufandas, caretas y/o tapabocas. Adicional, es importante tener en cuenta la distancia a la cual estará la persona de la cámara en este será aproximadamente de 0.90 m y para adecuar la implementación al equipo de la empresa se va a situar la cámara a una altura de 1.65 m con una inclinación de 7° hacia abajo. Esta configuración ha sido seleccionada considerando las características del equipo. Teniendo en cuenta estos datos, se espera que el reconocimiento sea más estable, y si no se tienen en cuenta estas limitaciones los resultados pueden no ser óptimos.

3. MARCO DE REFERENCIA

3.1. Marco Teórico o Conceptual

El proyecto de grado se basa en el diseño y la implementación de un dispositivo electrónico que reconozca factores demográficos como género y edad aproximada, para lo cual se va a utilizar el reconocimiento facial. Es importante tener en cuenta y dar definición a algunos conceptos que resultan importantes para la aplicabilidad del proyecto.

3.1.1. Inteligencia artificial

La inteligencia artificial o también llamada por sus siglas IA se ha conocido desde hace varios años, inicialmente con Allan Turing que indicó un enigma en el cual se preguntaba si las máquinas podrían pensar. Para esto generó una prueba en la que si una persona interactúa con una máquina y no distinguía si era ser humano, este se cumpliría. Unos años después y con el avance de la tecnología se da una definición más clara de este concepto en la cual varios científicos lo definen como la ciencia e ingenio de hacer máquinas inteligentes, especialmente programas de cálculo inteligente.

El concepto y la aplicación de IA ha tenido como modelo natural las funcionalidades e inteligencia del hombre. La principal motivación fue intentar construir máquinas que pudieran pensar como el ser humano. Adicional, es una de las disciplinas en que la mayoría

de los científicos de otras disciplinas les gustaría trabajar. El estudio de la inteligencia artificial es una de las disciplinas más antiguas. Desde hace 2000 años los filósofos se han esforzado por comprender cómo se ve, aprende, recuerda y razona, así como la manera en que esas actividades deberían realizarse. [21]

Con el objetivo de replicar los procesos de inteligencia humana utilizando inteligencia artificial, se requieren sistemas computacionales, gestión de datos y algoritmos de IA avanzados que sean creados y aplicados en un entorno dinámico de computación.[22]

Una limitación que se puede ver en la inteligencia artificial es su dependencia del aprendizaje a partir de datos, ya que no hay otra manera de incorporar datos, esto quiere decir que la precisión o exactitud de los datos se evidenciarán en los resultados. Del mismo modo, los sistemas de autoaprendizaje no son autónomos. Sin embargo, las computadoras que pueden analizar datos complejos para aprender y mejorar tareas específicas son más habituales.[23]

La inteligencia artificial tiene muchas aplicaciones, una de ellas es la detección facial en la que se pueden realizar la detección de cara en imágenes, comparación de rostros, búsqueda de caras similares, atributos en un rostro y la identificación de personas., para realizar esto se requiere del entrenamiento de un modelo de inteligencia artificial, indicándole la manera para aprender por medio de un algoritmo o varios según la finalidad que tenga el reconocimiento facial, sin embargo, es importante que para este tipo de aplicaciones se debe tener imágenes de las personas que se quieren detectar, compara, buscar o identificar.[24]

3.1.2. Visión por computador

La visión por computador se puede considerar un subcampo de la inteligencia artificial, su finalidad es que las computadoras simulen la visión humana, pero no solo con el hecho de ver, sino también de reconocer, clasificar etc. Este proceso se da gracias a un aprendizaje y unos respectivos modelos, debido a que a las computadoras se les debe generar una enseñanza, deben recopilar información y entenderla. Por esto, se deben realizar muchas pruebas para que su aprendizaje sea correcto y puedan llegar a la finalidad correspondiente. Adicional consiste principalmente en deducir de forma automática, la estructura y características de un entorno tridimensional posiblemente en movimiento, esto a partir de una o más imágenes bidimensionales del mismo entorno. Esta disciplina integra algunos

conceptos de física de color, óptica, electrónica, geometría, algoritmos, sistemas informáticos, entre otros. [25]

En este caso el aprendizaje y la importancia de la visión por computadora requiere la captura de una imagen y para su correcto procesamiento depende de varios factores. Básicamente consiste en generar una señal electromagnética en forma de luz que se transmite, refracta y refleja en la escena hasta que alcanza el área del sensor de la cámara. [25]

Es importante tener en cuenta varios elementos para la correcta detección de la imagen algunos de estos elementos son:

- **Iluminación:** Es importante controlar la iluminación ya que si esto no se tiene en cuenta se pueden obtener imágenes con bajos contrastes, sombras o destellos. Esto con el fin de mejorar la información necesaria para la detección y extracción las características de interés.
- **Ópticas:** Las ópticas permiten enfocar luz procedente de un punto en un lado de la óptica en otro punto detrás de la misma. Así permiten capturar una imagen nítida de un objeto situado a una distancia específica y la enfoca en el sensor que también está a una distancia fija al otro lado. Adicional pueden estar formadas por una única lente o varias lentes.[25]

3.1.3. Reconocimiento Facial

El reconocimiento facial se puede definir como un modo de identificar una persona por medio del rostro utiliza, algunas medidas especialmente de la cara y así también poder identificar algunos aspectos de la persona, los cuales puede identificar en tiempo real desde una cámara, en imágenes o en videos, adicional se puede utilizar para realizar diferentes aplicaciones con software de este tipo de implementación. Los softwares en los cuales se implementa el reconocimiento facial tienen una buena precisión, por lo cual son seguros y efectivos en cuanto identificación de personas. [26]

Para realizar este proceso se deben tener en cuenta algunos pasos [4] que en la materia funciona de la misma manera:

- **Reconocimiento Facial:** se detecta el rostro en la imagen.

- **Análisis Facial:** se captura y se analiza la imagen, teniendo en cuenta el software que se utilice se identificara los factores primordiales del rostro con esto tener los puntos de referencia faciales.
- **Conversión de la imagen a datos:** Se transforma la información capturada anteriormente en datos digitales.
- **Búsqueda de una coincidencia:** En este caso se realiza la comparación y verificación de la imagen digital.

3.1.4. Redes neuronales Artificiales

En el año 1943 dos investigadores dan una propuesta con respecto al primer modelo simple de una red neuronal y en 1958 se atribuye el desarrollo del perceptrón y se introducen los pesos en la ecuación. [1]

El concepto de una red neuronal artificial viene basado del proceso biológico que tienen los seres humanos y la capacidad de aprendizaje, lo que busca es emularlo a través de algoritmos de *machine learning*. Por esto, se basa en información que proviene de datos experimentales, se toman datos de entrada y se generan salidas según corresponda, esto no quiere decir que a la primera entrada se tendrá la salida esperada o será perfecta, debido a que va a tener un aprendizaje continuo hasta que se pueda llegar a un resultado esperado.

Algunas de las ventajas de las redes neuronales es que pueden simplificar algoritmos para que realicen un aprendizaje continuo, no es necesario tener un conocimiento respecto al procedimiento matemático que está realizando la red desde que se asocien los datos de entrada con los datos de salida generara aprendizaje. Si hay problemas no lineales y más complejos con este tipo de procesamiento de información se puede realizar de una forma más rápida y algo muy importante es que si en algún punto falla y no da el resultado deseado, va a seguir intentando y continúa trabajando hasta llegar al resultado final esperado. [1]

La estructura base de una red neuronal artificial se compone de neuronas de entrada, neuronas ocultas y neuronas de salidas. Estas neuronas se sitúan en capas, cada una de las neuronas realizan transformación respecto a las capas de entrada que reciben la información proveniente de afuera de la red, capas ocultas son internas y pueden tener varios niveles y capas de salida que transmiten la información. Adicional, se manejan los pesos con esto se refiere a que la neurona que tenga mayor peso será dominante en la siguiente neurona, para

pasar la información y así sucesivamente con el resto de la red, pero cuando las neuronas tienen menos peso la información de estas no se tomará en cuenta. Los pesos son asignados de manera aleatoria y se irán calibrando a medida que se vaya entrenando la red neuronal. [27]

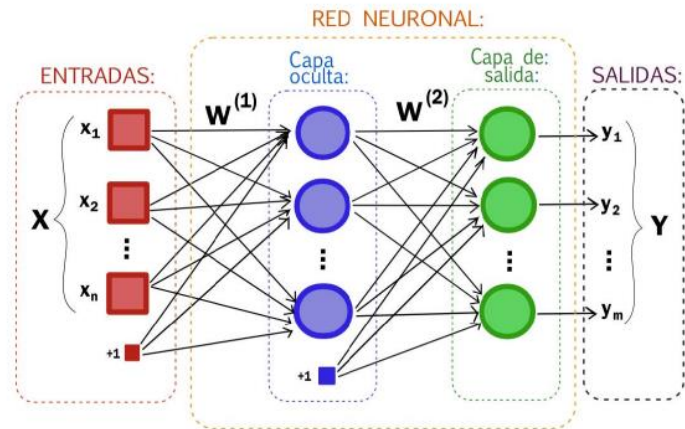


figura 1. Red neuronal artificial tomada de C. B. Carrión, "REDES CONVOLUCIONALES," Universidad de Sevilla, 2020.

3.1.5. Red neuronal Convolutional

Es uno de los modelos que utiliza de igual manera redes neuronales, sin embargo, presentan múltiples capas, para tener obtener una salida de un grupo de datos. Este modelo da inicio en los años 50 cuando dos investigadores Hubel y Wiesel modelaron la corteza visual animal. En su mayoría se utiliza este tipo de red para visión por computadora y procesamiento de imágenes y busca imitar la visión y reconocimiento de la corteza visual de un animal. La red neuronal tiene dispuestas tres dimensiones que son la anchura, altura y la profundidad agrupándolas en una capa oculta y todo esto se conecta en una capa interior. Se tiene la imagen de entrada que es la que va a procesar; esta pasa a la primera convolución, en la cual hace los filtros correspondientes se realiza un mapa de características del rostro como si está compuesto por líneas o curvas.

Si se realizan más convoluciones, este mapa de características podrá ir detectando formas un poco más complejas calculando también los pesos. Seguido de esto, el dispositivo realiza una segunda convolución con respecto a la anterior aplicando nuevos filtros y con una mayor cantidad de características; en este punto se realiza una reducción de las dimensiones de ancho y alto, para así generar una clasificación más exacta. Después de realizar las dos

convoluciones la imagen pasa a la red neuronal multicapa para ser analizada y posteriormente mostrar la capa de salida que este caso sería la clasificación final. [27]

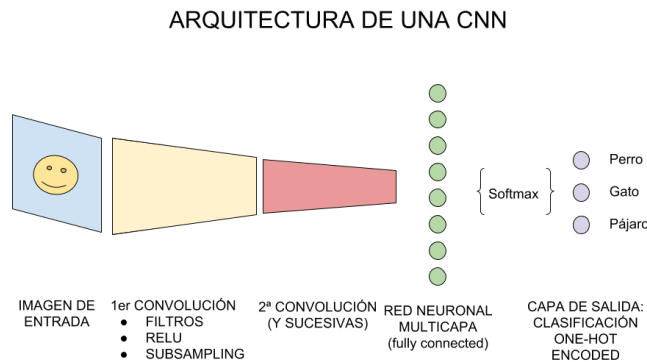


figura 2. Estructura de una red neuronal convolucional Tomada de Red neuronal artificial tomada de C. B. Carrión, "REDES CONVOLUCIONALES," Universidad de Sevilla, 2020.

3.1.6. Red Neuronal Recurrente

Este es un tipo de red neuronal que a diferencia de la red convolucional maneja conexiones de retroalimentación, lo que quiere decir que maneja un proceso de evolución para así hacer un proceso de evolución hacia un estado estable en el cual no haya cambios en el estado de la activación de neuronas. [21]

Con esta estructura le permite tomar en cuenta la información de forma temporal en los datos de entrada así permite utilizar dimensión de tiempo. Este tipo de redes son muy útiles para procesar secuencias de datos, como texto, audio y video, y se han utilizado con éxito en aplicaciones como la traducción automática, la generación de texto y el reconocimiento de voz.[28]

3.1.7. Capas

Keras proporciona una implementación de cada una de las capas utilizadas en los modelos. Para poder generar su implementación correcta se deben especificar ciertos parámetros; en este caso espera de cierta forma las imágenes de entrada en términos de filas (altura), columnas (ancho) y canales (profundidad). Adicional a esto, el filtro contiene los pesos que deben aprenderse durante el entrenamiento de la capa y la característica principal del mismo es que detectará la intensidad de la activación. [29]

Teniendo en cuenta los conceptos anteriores, cada red neuronal debe tener sus correspondientes capas, algunas de las que se usaron son las siguientes:

3.1.7.1. Conv2D

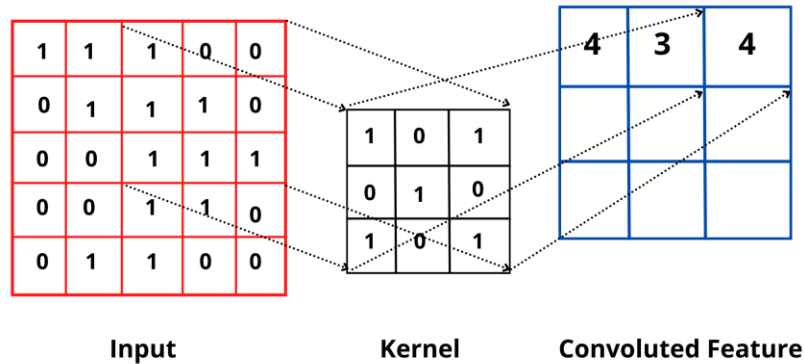


figura 3. Proceso de convolución tomada de [30]

```
x = Conv2D(16, (3, 3), padding="same")(inputs)
```

figura 4. Capa Conv2D usada en los modelos (tomada directamente del código)

La definición de esta capa según la documentación de TensorFlow [31] es utilizada para convolución espacial sobre imágenes, sin embargo, el fin de esta capa es crear un núcleo que se convoluciona con la entrada de la capa para producir un tensor de salidas. Si los argumentos “use_bias ” es igual a verdadero, se crea una vector y se agrega a las salidas y también se aplican salidas cuando el argumento “activation” no es None.

La figura 4 es una de las capas usadas en los modelos. Los argumentos y sus descripciones son las siguientes:

- **filters:** Entero, la dimensionalidad del espacio de salida (es decir, el número de filtros de salida en la convolución).
- **kernel_size:** Un entero o tupla/lista de 2 enteros, que especifica la altura y la anchura de la ventana de convolución 2D. Puede ser un solo entero para especificar el mismo valor para todas las dimensiones espaciales
- **padding:** Uno de "válido" o "igual" (insensible a mayúsculas y minúsculas). "válido" significa que no hay relleno. "same" da como resultado un relleno con ceros

uniformes a la izquierda/derecha o arriba/abajo de la entrada. Cuando `padding="same"` y `strides=1`, la salida tiene el mismo tamaño que la entrada.

3.1.7.2. Activation ReLu

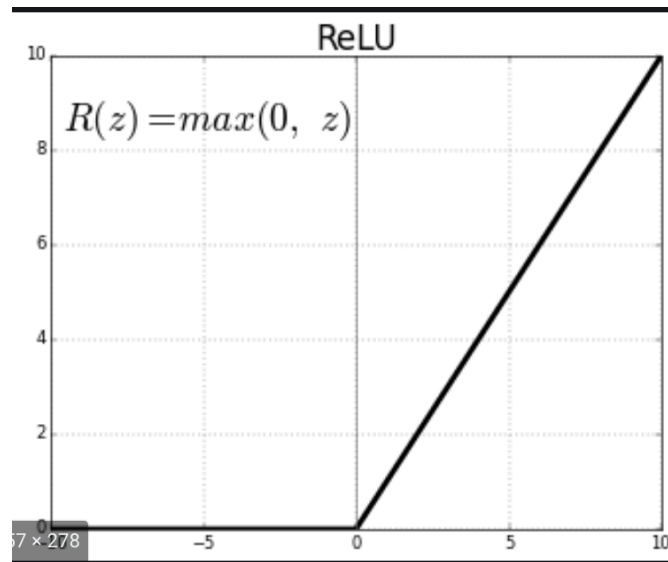


figura 5. Función de activación ReLU tomada de [32]

```
x = Activation("relu")(x)
```

figura 6. Capa Activation usada en los modelos (tomada directamente del código)

Para esta capa en específico se genera una activación a una salida y esta es aplicada directamente a una función, en la cual el nombre de cadena de la función es "relu" como se muestra en la figura anterior, así es utilizada en el modelo según la documentación [33]

3.1.7.3. BatchNormalization

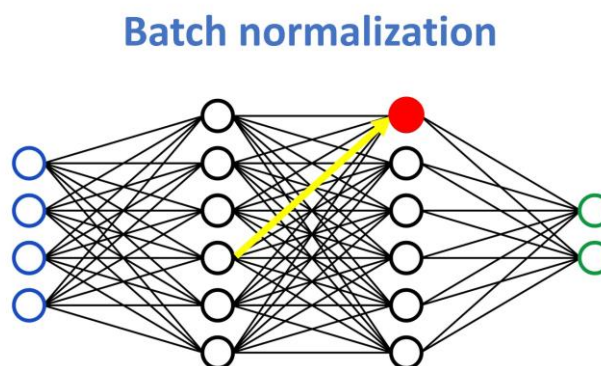


figura 7. Batch normalization función tomada de [34]

```
x = BatchNormalization(axis=-1)(x)
```

figura 8. Capa BatchNormalization usada en los modelos (tomada directamente del código)

El proceso que realiza esta capa es la normalización por lotes. La capa aplica una transformación que mantiene la salida media cercana a 0 y la desviación estándar de salida cercana a 1. Es importante tener en cuenta que esta normalización funciona de manera diferente al hacer el entrenamiento del modelo que cuando ya está entrenado [35].

El argumento que se utiliza en esta capa es:

- Axis: Es un entero el eje que debe normalizarse y generalmente es el eje en el que se encuentran las características.

3.1.7.4. MaxPooling2D

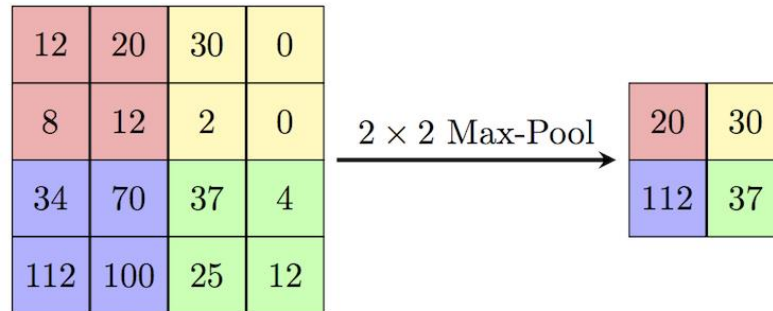


figura 9. Función maxpooling tomada de [36]

```
x = MaxPooling2D(pool_size=(3, 3))(x)
```

figura 10. Capa MaxPooling2D usada en los modelos (tomada directamente del código)

La capa de MaxPooling2D realiza una reducción de la muestra con respecto a la entrada y a lo largo de sus dimensiones espaciales (alto y ancho), toma el valor máximo sobre una ventana de entrada este tamaño es definido por el argumento. [37]

El argumento que se utiliza en esta capa es:

- Pool_size: Es un entero o tupla de 2 enteros, tamaño de la ventana sobre la que tomar el máximo. Tomará el valor máximo en una ventana de agrupación de 2x2. Si solo se especifica un número entero, se utilizará la misma longitud de ventana para ambas dimensiones (2,2) [37].

3.1.7.5. Dropout

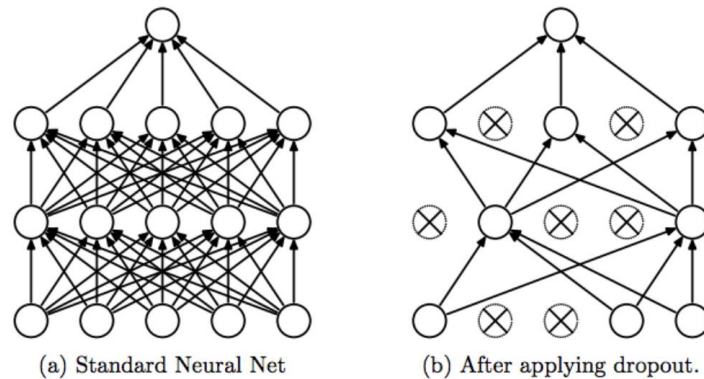


figura 11. Imagen Función dropout tomada de [38]

$$x = \text{Dropout}(0.25)(x)$$

figura 12. Capa Dropout usada en los modelos (tomada directamente del código)

La capa Dropout establece aleatoriamente las unidades de entrada en 0 con una frecuencia de rate (el cual es el argumento para el modelo). Esto sucede durante el entrenamiento del modelo, lo que ayuda a evitar el sobreajuste [39].

El argumento que se utiliza en esta capa es:

- Rate : Es un valor flotante entre 0 y 1. Fracción de las unidades de entrada a eliminar [39].

3.1.7.6. Flatten

Flatten Layer in Keras

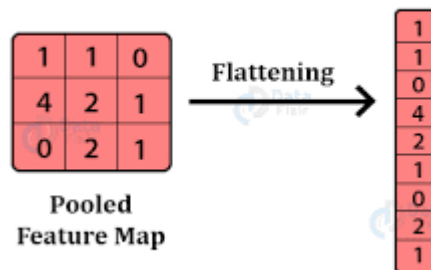


figura 13. Imagen función flatten tomada de [40]

$$x = \text{Flatten}()(x)$$

figura 14. Capa Flatten usada en los modelos (tomada directamente del código)

Esta capa aplanar la entrada que llegue, sin embargo, no afecta el tamaño del lote que llegue, maneja solo un argumento:

- `Data_format`: Se maneja con una cadena, es importante el orden de las dimensiones de las entradas, corresponde a entradas con forma y es importante tener en cuenta que el valor predeterminado es el que se encuentra en el archivo de configuración de Keras y si este nunca fue configurado entonces será `"channels_last"`[41].

3.1.7.7. Dense – Full Connected

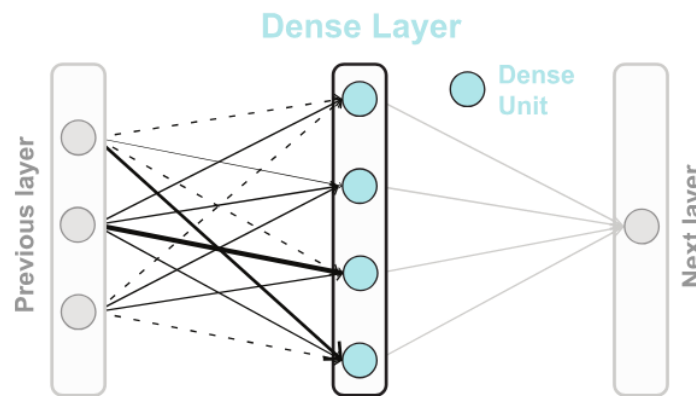


figura 15. Arquitectura de Capa dense tomada de [42]

```
x = Dense(128)(x)
```

figura 16. Capa Dense usada en los modelos (tomada directamente del código)

Esta capa implementa la operación: $\text{output} = \text{activation}(\text{dot}(\text{input}, \text{kernel}) + \text{bias})$ donde `activation` se pasa la función de activación por elementos como `activation` argumento. Sin embargo todos estos son atributos de la capa, es importante tener en cuenta que los atributos no se pueden modificar después que la capa ya se haya llamado una vez, el único que puede generar cambio es `"trainable"` [43].

El argumento que tiene esta capa es:

- `Units`: Es un número entero positivo, dimensionalidad del espacio de salida.

3.1.8. Modelos de IA

Con los modelos de IA se puede representar la creación e implementación de algoritmos de aprendizaje automático. Esto con el fin de emular la lógica de las decisiones y generar análisis avanzados, lo que quiere decir que estos modelados son la creación de un proceso que puede

tomar decisiones por sí mismo. Para generar estos modelos, se maneja cierto procedimiento donde primero se genera un modelado en el cual se describen las capas de algoritmos que interpretaría los datos y así se pueda generar la toma de decisiones, seguido de esto se realiza el entrenamiento del modelo creado. En el proceso de entrenamiento se deben generar varias pruebas con muchos datos para así tener una mayor exactitud en los resultados y su comportamiento sea correspondiente. Por último, la inferencia que es cuando el modelo creado, entrenado y probado se despliega, para que cumpla la función correspondiente. [44]

Para realizar un modelo de IA, su entrenamiento y pruebas se debe tener una CPU de alto rendimiento, alta capacidad de almacenamiento y redes de buen ancho de banda, ya que realizando estos procesos se utiliza gran cantidad de datos para generar los entrenamientos y entre más rápido y más eficaz el modelo puede ser implementado en menor tiempo. [44]

3.1.9. Entrenamiento de un modelo

El entrenamiento de modelo es el proceso mediante el cual se definen los parámetros de un algoritmo, para así el error pueda llegar a ser mínimo [45]. Para realizar el entrenamiento de un modelo inicialmente se le da un conjunto de datos con las etiquetas correctas, conocidos como datos de entrenamiento con los cuales se puede generar un aprendizaje a través de repeticiones.

Se hace el seguimiento correspondiente al modelo generado por el conjunto de datos anterior y así se puede detectar patrones y una agrupación en función de lo más similar. Cuando el modelo aprende, crea un modelo de inferencia que utilizará para resolver y / o clasificar el problema. Así, se forman estructuras de modelos bastantes completos sin necesidad de tener una abundante cantidad de datos. [46]

3.1.10. Modelos Preentrenados

Estos modelos de aprendizaje pre entrenados no requieren de volúmenes altos de datos ni de recursos para generar el entrenamiento del mismo. Esto se debe a que la finalidad es la utilización de modelos ya entrenados y se busca realizar cambios generalmente en las últimas capas [47] u obtener la información que ya se tiene para hacer modificaciones directamente en los algoritmos. Así, se espera generar una extracción de las características para la

transferencia de conocimiento del modelo principal y con esto se crea un nuevo modelo ajustado, modificado y afinado a la funcionalidad que se requiere. [48]

3.1.11. Aprendizaje Profundo

El aprendizaje profundo es básicamente acceden a una mayor cantidad de datos, para que así el modelo tenga más experiencia y así mejorar su rendimiento, la finalidad de este tipo de aprendizaje es que al llegar a la suficiente experiencia se pueda utilizar para una tarea específica, por ejemplo, conducir un auto. Sin embargo, el funcionamiento del mismo es debido a que se manejan estructuras más complejas. Esto quiere decir que hay varias capas que tienen más capas ocultas con un sinnúmero de neuronas artificiales conectadas, las cuales tienen un peso. Cuando este caso es positivo, incentivará un nodo a otro, pero si es negativo un nodo suprime al otro, los nodos que tengan mayor peso tienen más valor. [49] Se busca recopilar analizar e interpretar grandes cantidades de datos y esto hace que el proceso sea rápido y sencillo.

3.1.12. Aprendizaje Automático

El aprendizaje automático se refiere al desarrollo de algoritmos con la capacidad de identificar patrones complejos en grandes cantidades de datos, con los cuales puede generar predicciones de lo que puede llegar a pasar. Esto quiere decir que él mismo generará el aprendizaje basándose en esos ejemplos de información de entrada, teniendo en cuenta la finalidad que se requiere.[50]

Este tipo de aprendizaje consta de un algoritmo situado para la toma de las determinaciones. Adicional, tiene variables y funciones que van a tener la información de la decisión y el conocimiento inicial para así saber a qué respuesta debe llegar con la finalidad de que aprenda y se entrene. Teniendo en cuenta lo anterior, el aprendizaje se puede indicar que se centra en el análisis y la interpretación de cierta estructuras y patrones de entrada para que pueda generar una toma de decisión y llegue a la salida correcta. [51]

3.1.13. Open CV

Esta es una librería de Python, una de las más grandes en la cual el código es abierto y se especializa en visión por computadora, aprendizaje automático y procesamiento de imágenes. Esto es importante con respecto a la operación en tiempo real debido a que se pueden identificar objetos, rostros o también escritura. Se busca realizar una identificación de los patrones y sus diversas características en imágenes o videos. [52]

Con esta librería se puede realizar la reconstrucción y comprensión de una escena en 3D a partir de imágenes en 2D. Con la visión por computadora se trata de replicar la visión humana por lo cual se centra en procesamiento de imágenes, reconocimiento de patrones en las cuales explica estos y las clasificaciones para los mismos. La fotogrametría nos explica cómo tener las medidas precisas a partir de las imágenes. Con estos factores, la librería se complementa, para así obtener mejores resultados al momento de ejecutar el algoritmo. [53]

3.1.14. Parámetros de medición en un modelo

3.1.14.1. Función de pérdidas

Esta función permite tener claro cuántas veces hay equivocaciones en las predicciones. Adicional, evalúa las desviaciones entre los valores de predicción en la red neuronal y los valores reales en la salida. Si el resultado de esta función es de un valor mínimo, quiere decir que la red neuronal es eficiente y el modelo aprendió de manera correcta con el mínimo error. Sin embargo, si el valor es alto se debería ajustar el valor de los pesos en la red neuronal para que no se vea reflejada la pérdida. [54]

3.1.14.2. Precisión

Midiendo la precisión se puede obtener la calidad que tiene el modelo para así determinar qué tan buena es su clasificación, es importante tener en cuenta que se genera una matriz de confusión en la cual se evalúan el desempeño con respecto a las predicciones de los datos procesados, Se define las siglas como True Negative [TN], True Positive [TP], False Positive [FP], False Negative [FN]. [55]

		predicción	
		0	1
realidad	0	TN	FP
	1	FN	TP

figura 17. Matriz de Confusión tomada de J. M. Heras, "Precision, Recall, F1, Accuracy en clasificación," 2020.

$$precision = \frac{TP}{TP + FP}$$

figura 18. Fórmula de Precisión tomada de J. M. Heras, "Precision, Recall, F1, Accuracy en clasificación," 2020.

3.1.14.3. Regresión

El propósito de la regresión es establecer una relación de ciertas características y una variable. El resultado de la relación es un valor numérico dentro de un conjunto infinito de posibles resultados. [56]

Con la regresión se pueden generar predicciones con una variable dependiente que en este caso sería 'y' y variables independientes denominadas como 'x' un ejemplo de cómo se podría ver una gráfica de regresión puede ser la siguiente imagen [57].

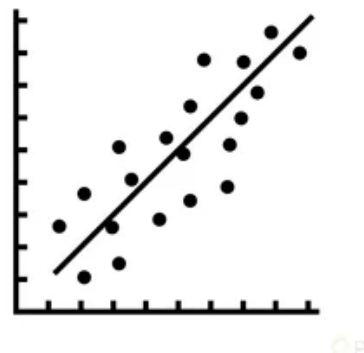


figura 19. Regresión Lineal tomada de L. Fernando, "¿Qué es y para qué sirve una regresión lineal en machine learning?," 2022.

3.1.14.4. R2Score

Es una manera de medir y evaluar el rendimiento de un modelo que se encuentra basado en regresión. Funciona midiendo la cantidad de variación en las predicciones explicadas por el

conjunto de datos por lo cual significa que es la diferencia entre las muestras en el conjunto de datos y las predicciones hechas por el modelo.[28]

3.1.15. Cámara

El principio de funcionamiento de una cámara inicialmente comienza al obtener las imágenes periódicamente de aquello que se está enfocando, pueden procesar aproximadamente más de 15 fotogramas por segundo, por lo cual se puede transmitir en video o en tiempo real. Según el uso es importante tener en cuenta el ángulo en el que se debe ponerla cámara para tener en cuenta los rostros que se van a reconocer, la iluminación u objetos que puedan interferir en la detección. [58]

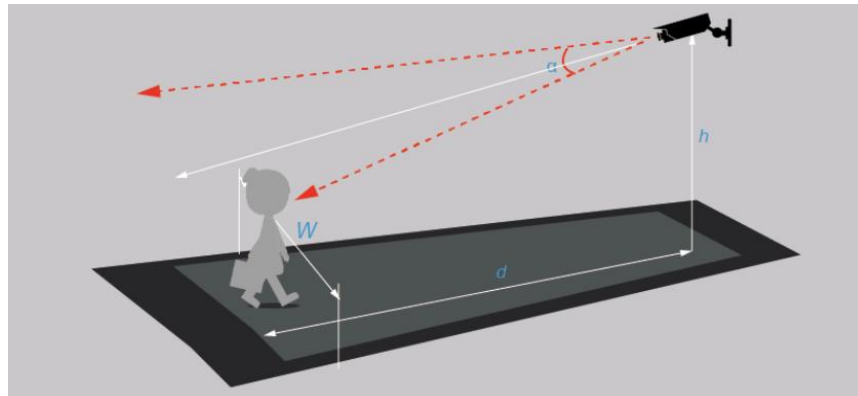


figura 20. Ejemplo de posición de cámara y posiciones a tener en cuenta imagen tomada de <https://forum.huawei.com/enterprise/es/requisitos-de-instalaci%C3%B3n-para-la-c%C3%A1mara-de-reconocimiento-facial/thread/499177-100259>

La Cámara FIT0701 es USB y se utiliza frecuentemente para Raspberry Pi, la cual no utiliza controladores y maneja 0.3 Megapíxeles y una resolución de 640×480. La cámara es compatible con las placas Raspberry Pi.[58]

3.1.16. Interfaz Serie para Cámaras (CSI)

El conector CSI es una interfaz que permite la conexión de cámaras a dispositivos electrónicos, y se utiliza para transmitir datos de imagen y comandos de control. Proporciona una conexión de alta velocidad y baja potencia, lo que lo hace adecuado para aplicaciones donde se requiere la captura y procesamiento de imágenes en tiempo real.

Los pines más comunes en un conector CSI son:

Pines de alimentación: Estos pines proporcionan la alimentación necesaria para la cámara. Generalmente, hay pines para la alimentación de 3.3V y 1.8V.

Pines de datos diferenciales: El conector CSI utiliza una transmisión de datos diferencial para garantizar una alta velocidad y una buena integridad de señal. Estos pines se utilizan para transmitir los datos de imagen desde la cámara al dispositivo anfitrión.

Pines de control: Estos pines se utilizan para la comunicación de control entre la cámara y el dispositivo anfitrión. Pueden incluir señales como el disparador de captura, señales de sincronización y otros comandos de control.

Es importante destacar que el conector CSI puede admitir diferentes configuraciones y versiones, como CSI-1, CSI-2 y CSI-3. Cada versión tiene características y capacidades específicas en términos de ancho de banda, resolución de imagen y funciones adicionales.

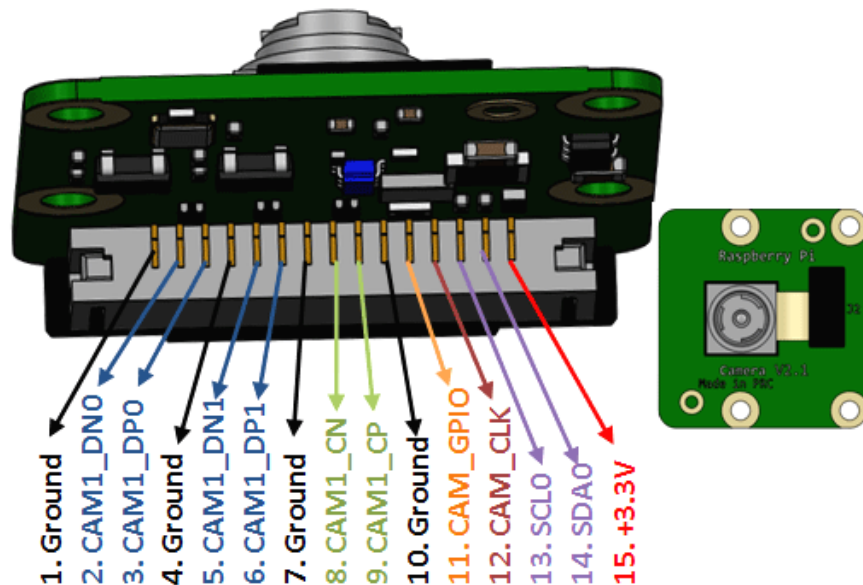


figura 21. Figura de como organización de pines de la cámara y a que corresponde cada uno tomada de [59]

CSI-2

Lanzada en 2005, la primera versión de MIPI CSI-2 vino con un protocolo dividido en capas, como:

Capa física (PHY): La capa física es responsable de la transmisión física de los datos a través de los pines del conector CSI-2. Define aspectos como la topología de la conexión, la señalización eléctrica, el tipo de conector y la velocidad de transmisión. La capa física asegura que los datos se transmitan de manera confiable y eficiente.

Capa de enlace de datos (Data Link Layer): La capa de enlace de datos se encarga de organizar los datos de imagen en paquetes y agregar metadatos y señales de control. Define la estructura de los paquetes y los protocolos para el envío y recepción de los datos. También se ocupa de la detección de errores y la corrección de errores durante la transmisión.

Capa de control (Control Layer): La capa de control se encarga de la comunicación de control entre la cámara y el dispositivo anfitrión. Define los comandos y señales necesarios para configurar y controlar la cámara, como la configuración de la resolución de imagen, el control del obturador y la configuración de los parámetros de captura.

Capa de aplicación (Application Layer): La capa de aplicación define los formatos y protocolos específicos utilizados para transmitir datos de imagen y metadatos. Puede incluir formatos de compresión de imagen, como JPEG o RAW, y protocolos de aplicación específicos para diferentes aplicaciones y requisitos.[60]

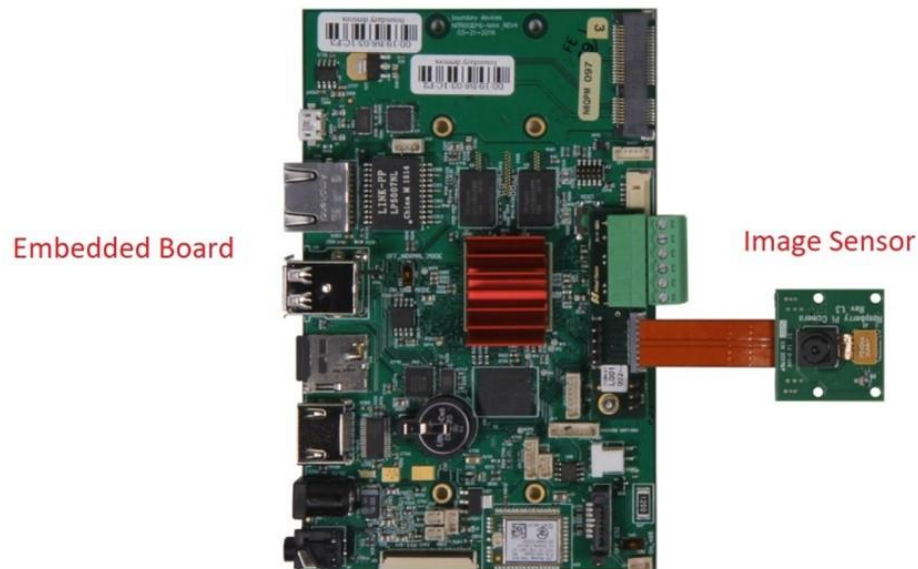


figura 22. Cámara de raspberry pi imagen tomada de [60]

3.2. Marco Legal o Normativo

Para Colombia hay una ley que protege el tratamiento de datos personales: esta es la Ley 1581 de 2012, en la cual su objeto en el artículo 1° indica que reconoce y protege el derecho que tienen todas las personas a conocer, actualizar y rectificar las informaciones que se hayan recogido sobre ellas en bases de datos o archivos[61], adicional se incluirán todos aquellos documentos que puedan ser procesados por organismos de naturaleza pública o privada.[62] Teniendo en cuenta lo anterior, no se puede generar la publicación de imágenes tomadas sin consentimiento.

Para la finalidad de este proyecto, se tiene en cuenta la ley indicada anteriormente, por lo mismo no se recolectan imágenes de las personas ni datos personales. Solo se generará la identificación de un aproximado de edad y el género, no se obtiene información verídica ya que no se puede constatar la edad y el género real.

4. DESARROLLO DEL PROYECTO DE GRADO

4.1. Requerimientos

Tabla 1. Requerimientos Funcionales

Código	Descripción
RFU01	Se puede realizar la instalación del algoritmo en sistemas operativos Windows y Linux
RFU02	Se debe generar una prueba funcional, para comprobar que el dispositivo realiza la medición de edad y de género y recopila la información correctamente
RFU03	Se debe detectar una edad aproximada del individuo que se está detectando en rango de error entre -8 y +8 años
RFU04	Se deben almacenar edad, género, tiempo de permanencia y fecha en una base de datos, que se encuentra en un servidor web.
RFU05	Se debe mostrar los datos registrados en un dashboard.
RFU06	Se debe detectar el género del individuo que se está detectando, esta estimación debe ser al menos un <u>90%</u> acertada.
RFU07	Se debe agregar un contador para tomar el tiempo de permanencia de los sujetos.
RFU08	El dispositivo debe tener la posibilidad de implementar el sistema de fragancias ya manejado por la empresa.
RFU09	El dispositivo se debe poder adaptar fácilmente con otros desarrollos que pueda solicitar el cliente

Tabla 2. Requerimientos De Calidad

Código	Descripción
RC01	El lugar dónde se va a instalar debe tener mínimo un espacio de 9m ²

Tabla 3. Requerimientos De Gerencia

Código	Descripción
RGE01	Se debe realizar el desarrollo del proyecto en un año y medio
RGE02	Los costos son asumidos por la empresa que financia el proyecto
RGE03	La empresa entrega los dispositivos necesarios para la realización del proyecto (Raspberry pi, cámara) y remunera por la implementación de este.
RGE04	Se realizarán las comunicaciones vía online únicamente hasta la entrega del proyecto que se realizará presencial.
RGE05	Se realizará una capacitación funcional y técnica, tanto para el usuario final como para los técnicos de la empresa.
RGE06	Se realiza compra de cámara adicional para hacer pruebas fuera del presupuesto como adquisición personal.

Tabla 4. Requerimientos Restrictivos

Código	Descripción
RRES01	El dispositivo no detecta una persona con tapabocas
RRES02	La persona debe estar máximo a una distancia de 2 metros o no será detectada
RRES03	El rango de error de la predicción cuando la persona tiene gafas de sol, bufanda, gorras, se encuentra a contraluz o tiene el rostro cubierto es de 0% a 50%
RRES04	Se toma un data set ya creado, no se crea un data set propio.

4.2. Metodología del Diseño

4.2.1. Fases del Proyecto

En la figura 23 se observa un diagrama general de las diferentes etapas para la realización del proyecto, esto es con base a una estructura diseñada cronológicamente especificando las actividades para cada etapa.

Estructura de la metodología

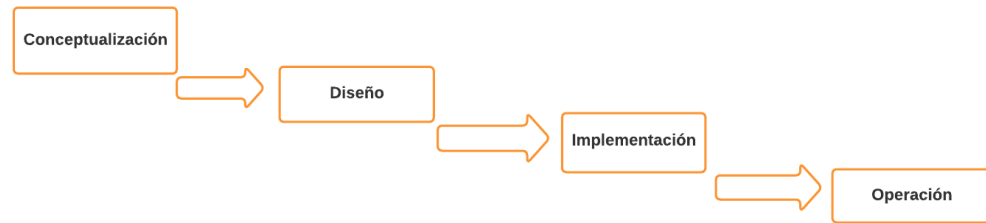


figura 23. Estructura de la Metodología

4.2.1.1. Etapa 1 Conceptualización

Para la primera etapa que es de conceptualización inicialmente se planteó un problema para iniciar un proyecto, seguido de esto se generó un Anteproyecto en el cual se planteó un Objetivo general y específicos para el proyecto, se incluye una descripción, algunos antecedentes y justificación de lo que se realizó, adicional el alcance inicial del proyecto, las limitaciones que se pueden llegar a tener, los recursos necesarios para la realización y un cronograma inicial para el desarrollo del proyecto. El anteproyecto fue entregado para una revisión preliminar y para aprobación para así empezar a realizar todo lo que se requiere para el proyecto.

Seguido de lo anterior se realizó una revisión literaria, a pesar de que ya se levantó información para el anteproyecto es importante tener en cuenta toda la documentación para verificar ciertos aspectos del desarrollo e implementación, también investigación avanzada con respecto a trabajos ya realizados y sus aplicaciones, adicional revisar documentación con respecto a los equipos necesarios, generar comparaciones y verificar resultado para así elegir las opciones más viables.

Teniendo las revisiones correspondientes y las comparaciones, se compran de los equipos y adquirir los recursos necesarios para así iniciar el diseño e implementación.

4.2.1.2. Etapa 2 Diseño

Para la etapa de diseño inicialmente se definen los requerimientos necesarios de diseño para generar la implementación del dispositivo.

- Requerimientos Funcionales
- Requerimientos No Funcionales
- Requerimientos de gerencia
- Requerimientos de Calidad
- Requerimientos Restrictivos

Seguido de esto se realizó un diseño del programa para generar la adquisición de imágenes y de datos, esto con el fin de verificar si es viable generar un data set propio y se debe generar investigación sobre data set existentes que sean de código abierto y que sean óptimos para la aplicación requerida.

Se revisan algunos modelos existentes para comprobar si se pueden adaptar a las necesidades para cumplir el objetivo de generar la detección de edad y género. Se realizó la investigación de ciertos modelos, en este caso el de Open VINO y Caffe Net los cuales no se consideraron como aptos debido a que para open vino solo se podrían instalar en dispositivos Intel, por eso fue descartado principalmente y para el de Caffe Net no generaba la mejor precisión a pesar de que la documentación si lo indica, adicional se buscó realizar el entrenamiento propio y el desarrollo del modelo.

4.2.1.3. Etapa 3 Implementación

En la etapa de implementación como su nombre lo indica ya se empieza a generar la implementación de los diseños realizados en la anterior etapa, ya con el data set escogido y el modelo de diseño evaluado, se empieza a realizar modificaciones al modelo para generar una implementación final y seguir con pruebas.

Se realizaron pruebas e implementación de 2 prototipos, uno de ellos realiza la clasificación de edad por rango de edades, como, por ejemplo: el sujeto 1 tiene un rango de edad entre 25 - 32 años y el segundo prototipo genera la estimación de edad en un solo valor aproximado a la edad del sujeto.

Seguido de esto se implementaron y se preparan los equipos con los algoritmos para poder generar las pruebas correspondientes. Se realiza la implementación de 5 modelos diferentes, con diferentes épocas de los cuales se realizan pruebas para verificar cual es el que se adapta mejor a la predicción de edad y género y presenten el menor porcentaje de error.

4.2.1.4. Etapa 4 Operación

Adicional a lo anterior en esta etapa teniendo el diseño y la implementación del equipo se inicia con el documento final para la última presentación del proyecto. En este caso, se avanza en el marco de referencia y la introducción que se va a generar en el documento.

Se debe generar pruebas de comparación con los modelos anteriores y el modelo que se va a utilizar para verificar el funcionamiento y verificar las características que diferencian cada uno con ciertas épocas y haciendo ciertas modificaciones para verificar sus variaciones.

Para esta etapa se realiza la implementación final del dispositivo, se implementó en uno de los equipos dispensador de fragancias de la empresa, se realizaron algunas pruebas en una máquina de autogestión. Con el fin de probar el modelo modificado y adaptado para la detección de rostro donde se pueda predecir edad y género, este último modelo ya queda implementado en el dispositivo físico de la empresa.

Después de realizar ciertas pruebas se debe avanzar en el documento, ingresando esas pruebas y resultados haciendo un análisis de los mismos, seguido de tener esta información se debe terminar el documento con la forma indicada y con todos los capítulos completos, para entregar el documento y tener una revisión total del mismo. Finalmente, se sustentará todo el diseño y la implementación del dispositivo electrónico para recolectar información de edad y género.

4.3. Descripción Técnica del Producto

El producto propuesto es un dispositivo basado en visión por computadora diseñado para recopilar información demográfica, como la edad y el género de las personas. El objetivo principal del dispositivo es proporcionar una solución automatizada y precisa para la recopilación de estos datos, evitando así la necesidad de métodos tradicionales que requieren interacción humana. Teniendo en cuenta que este dispositivo se diseñó e implementó en un equipo que ya tiene la empresa, se realizó el diseño en base a los requerimientos del cliente y la información levantada.

4.3.1. Levantamiento de información

Este proyecto implemento en una máquina de la empresa, de tal manera que fue necesario realizar una visita para obtener la información sobre esta máquina. Se trata de una maquina autoservicio en la cual las personas se acercan a probar perfumes.

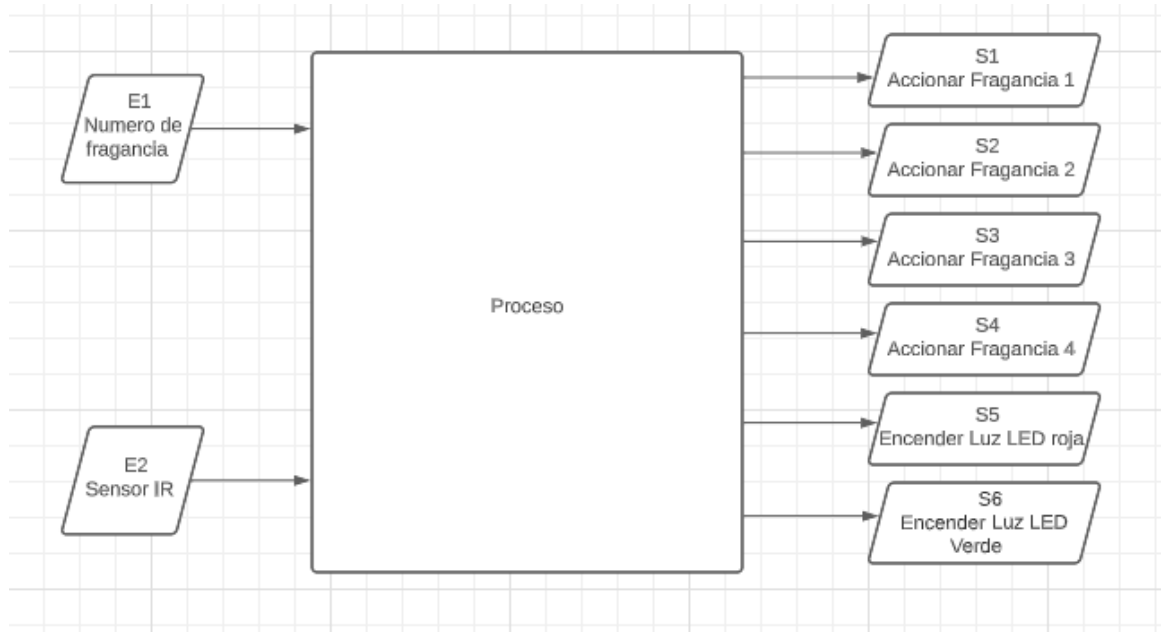


figura 24. Esquema de Equipo de fragancias

En la figura 24 se observa un esquema básico de cómo es el funcionamiento del equipo de fragancias que maneja la empresa, inicialmente se maneja una entrada, en esta caso la recibe un monitor táctil, en el monitor aparece una interfaz en la cual se podrá seleccionar que tipo de fragancia desea probar.

Este proceso es confidencial debido a que la empresa no autoriza a mostrar esta información.

4.3.2. Hardware

Una vez que se entendió todo el sistema se planteó una solución, en la cual se propuso diseñar 2 subsistemas, debido a que surgió la necesidad por parte del cliente de que uno de los dispositivos se pueda implementar fácilmente sin tener la necesidad de hacer cambios a estructuras ya existentes. Adicional se generó el diseño e implementación de un dispositivo de bajo costo que integre la opción de fragancias y el dispositivo para generar reconocimiento de datos demográficos como edad y género.

A continuación, se muestra un diagrama general de las entrada y salidas de los sistemas unificados en uno solo. Adicional en la figura 25 se observa un esquema donde se evidencia como queda la unificación de los dos subsistemas.

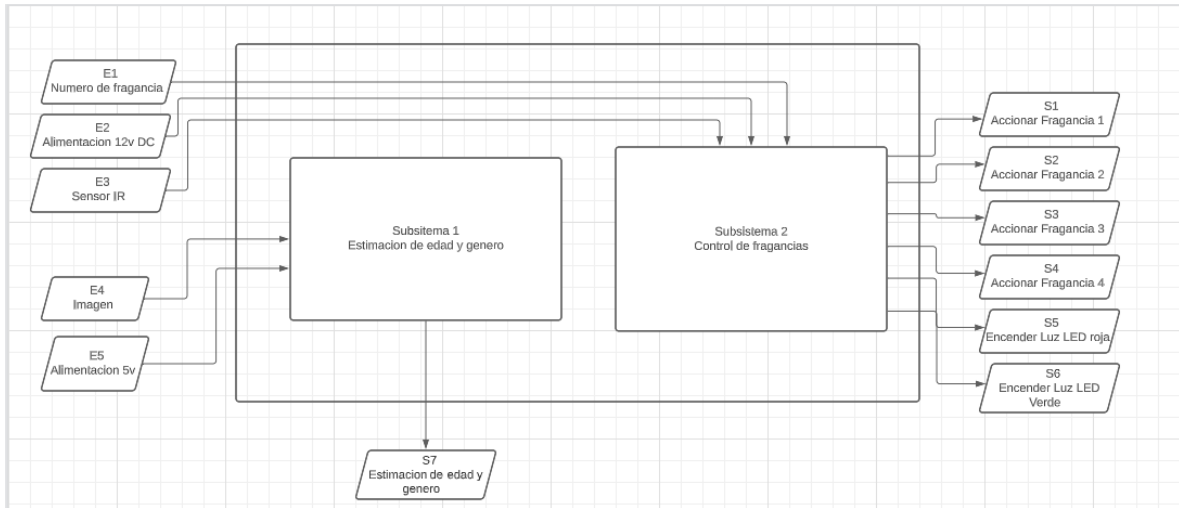


figura 25. Diagrama de entradas y salidas de la implementación general

Este proceso es confidencial debido a que la empresa no autoriza a mostrar esta información.

4.3.2.1. Subsistema de edad y género

En el subsistema de edad y género es el encargado de realizar las estimaciones de edad y género, este subsistema tiene como entrada una imagen, seguido de tener la entrada pasa por un algoritmo de reconocimiento de caras y luego cada cara pasa por un algoritmo para estimación de edad y género. Por último, este subsistema entrega una salida en forma de datos, edad y género como se muestra en la siguiente figura 26.

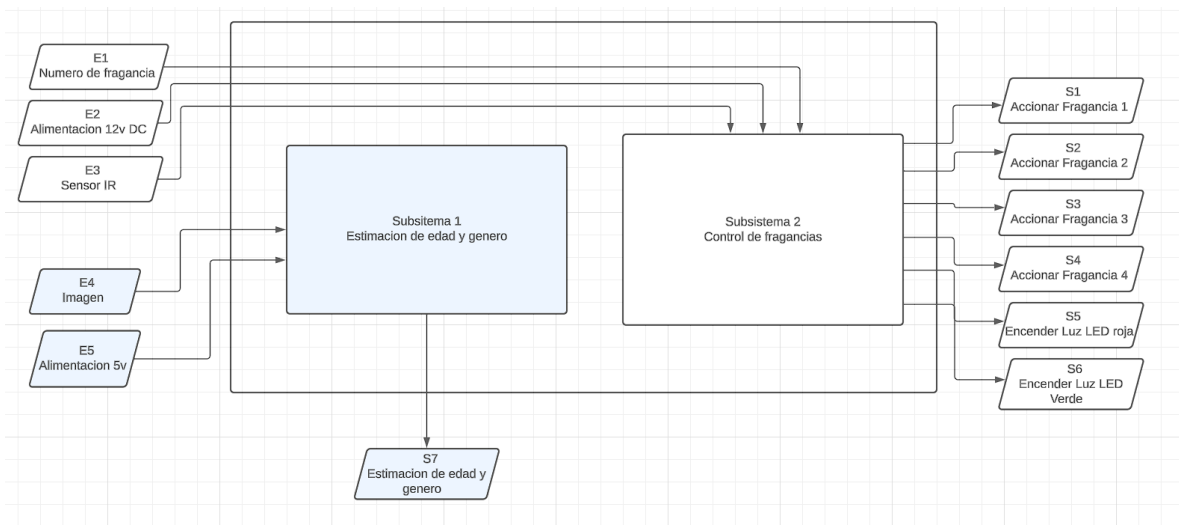


figura 26. Diagrama subsistema edad y genero

Esta propuesta de diseño se genera a partir de la necesidad de implementar el sistema de detección de edad y género en una maquina ya existente sin necesidad de realizar cambios, para este diseño es importante mencionar que si bien ya se tiene un diseño funcional que integra el sistema de fragancias y el sistema de edad y género, este diseño mencionado solo sería conveniente implementarlo en máquinas nuevas, pero al implementarlo en una máquina que ya esté en sitio se incurren en muchos gastos. Por lo cual se le propuso una solución rápida y funcional en la cual se implemente el sistema de edad y genero la maquina ya existente.

Por lo anterior se observa en la figura 27 que se encuentra el sensor de imagen que es la cámara con la cual se va a generar la adquisición de la imagen, el procesador y la fuente de alimentación hacia el procesador

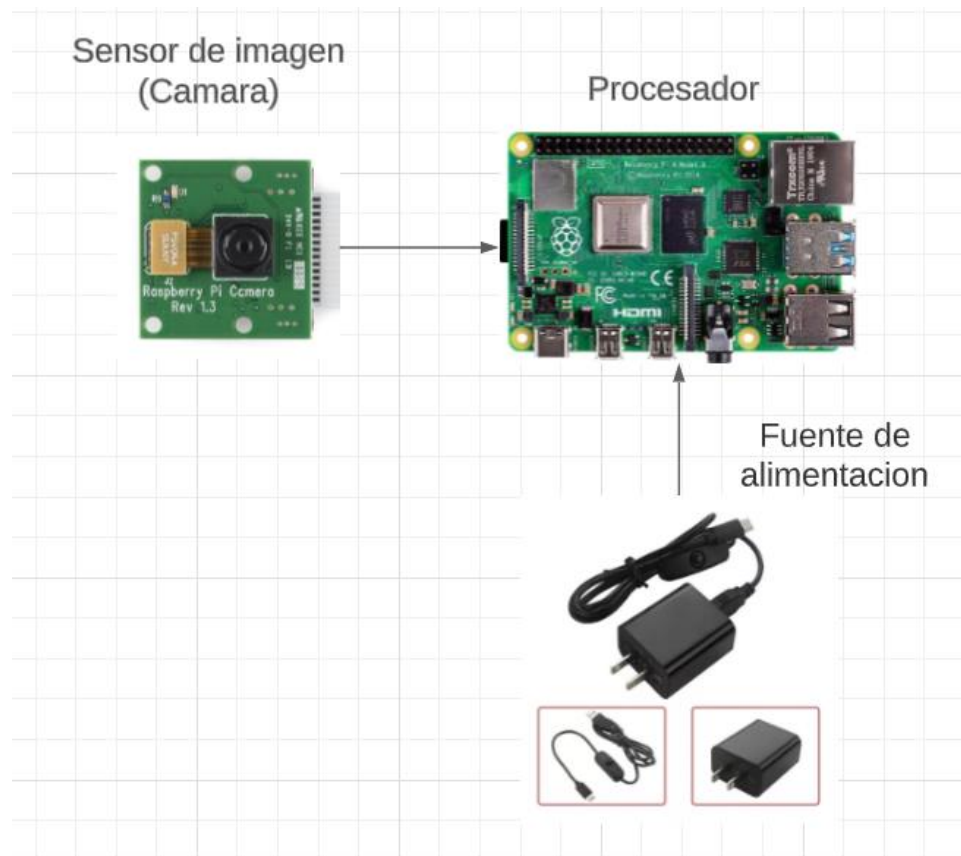





figura 27. Módulo de Procesamiento de imágenes

Los equipos que se tuvieron en cuenta fueron los siguientes:

Tabla 5. Descripción características dispositivos investigados.

Imagen			
Nombre	Raspberry Pi 4	Nvidia Jetson Nano	Asus tinkin board
Procesador	64-bit 4-core Cortex-A72 (ARM v8) @ 1.8GHz	64-bit 4-core ARM A57 @ 1.43GHz	32 bits 4-core Cortex-A17 @ 1.8Ghz
GPU	Videocore VI	128-core NVIDIA Maxwell @ 921MHz	4-core Mali™-T764 @ 600Mhz
Memoria RAM	1GB/2GB/4GB/8GB LPDDR4	4GB LPDDR4	2GB of LPDDR3
Puertos USB	2xUSB 3.0 2xUSB 2.0	4x USB 3.0	4xUSB 2.0
Almacenamiento	Ranura Micro-SD	Ranura Micro-SD	Ranura Micro-SD
Alimentación	5V DC conexión USB-C	5V DC conexión micro USB o Jack de barril	5V DC conexión micro USB
Precio -Cop (Amazon Mayo del 2023)	\$470.000-\$910.000	\$728.000 - \$1.100.000	\$570.000-\$770.000

Se realizó una comparación entre las tres placas donde la Raspberry Pi 4 que fue el equipo seleccionado para la implementación, se identificó lo siguiente:


- Con respecto a costos las 3 placas manejan precios muy similares.
- En la parte de procesamiento las 3 placas cuentan con procesadores de arquitectura ARM, con 4 núcleos. Sin embargo, se evidencia una mejora de la Raspberry pi 4 y la Jason Nano con respecto a las Asus Tinker, ya que las Asus Tinker maneja una arquitectura de 32 bits.
- Con respecto al GPU la Jetson Nano es superior a las otras placas, ya que cuenta con una unidad de procesamiento grafico de 128 núcleos
- Con respecto a memoria RAM, la ASUS tinkin maneja una tecnología anterior con respecto a la Jetson Nano y la Raspberry pi4, donde la placa Asus maneja LPDDR3 mientras que las otras 2 placas manejan LPDDR4, por otro la placa que puede tener una configuración de mayor RAM es la Raspberry pi 4 con 8GB

Con esta comparación se evidencia que la Asus Tinker Board, es la opción que menos favorece en este proyecto debido a que tiene una arquitectura de 32 bits y maneja tecnología LPDDR3 lo cual la deja en desventaja en comparación a la Raspberry pi 4 y La Nvidia Jetson Nano.

Para esta propuesta de diseño se eligió la **Raspberry pi 4** ya que cuenta con una configuración que proporciona el doble de memoria RAM a comparación de la Nvidia Jetson Nano.

La cámara que se usó para este diseño fue la Raspberry Pi Camera 1.3.

Tabla 6. Descripción características Cámara Raspberry Pi Camera 1.3

Imagen de la cámara	
Nombre	Pi camera rev1.3
Megapíxeles	5MP
Resolución	2592 × 1944
Numero de colores	40569
Brillo	24.05%
Saturación	10.74%
Matriz de color	162°
Sensor	CMOS - OmniVision OV5647
Conexión	interfaz MIPI

Esta cámara cuenta con un sensor CMOS de la marca OmniVision que tiene referencia OV5647 de 5 megapíxeles, que se caracteriza por su bajo consumo y alto rendimiento, el sensor OV5647 puede proporcionar una salida de video de 2592x1944. [63]

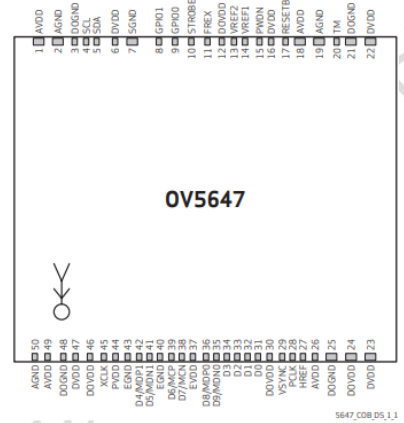


figura 28. sensor OV5647 tipo CMOS de 5 megapíxeles imagen tomada de[63]

Implementación subsistema de edad y genero

Se realizo la implementación en sitio del sistema de estimación de edad y género. Donde se tiene las siguientes características de este dispositivo

- Sistema operativo Raspbian OS (basado en Linux, código abierto)
- Cámara de 5MP
- Conexión de la cámara CSI
- Voltaje de alimentación 120v AC
- Voltaje de la placa 5v DC - USB tipo C
- Consumo en ralentí ~4w
- Consumo realizando estimación de edad y genero ~7w

4.3.2.2. Subsistema control de fragancias

Este proceso es confidencial debido a que la empresa no autoriza a mostrar esta información.

4.3.3. Software

Partiendo el concepto de visión por computador la cual es una tecnología que se aplica en este proyecto y nos permite recopilar información demográfica como edad y género. Se maneja un proceso el cual tiene una imagen como entrada, en este proceso hace uso de un algoritmo de detección de rostros, luego otro algoritmo va a procesar estos rostros, seguido de esto está el algoritmo que va a generar la estimación de edad y género.

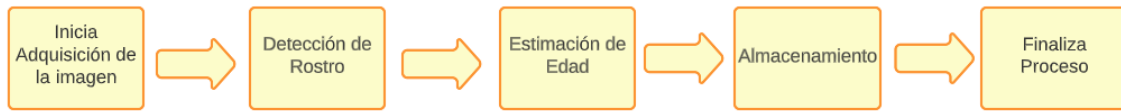


figura 29. Diagrama de Bloques Software General

4.3.3.1. Diagrama de flujo del funcionamiento general del software

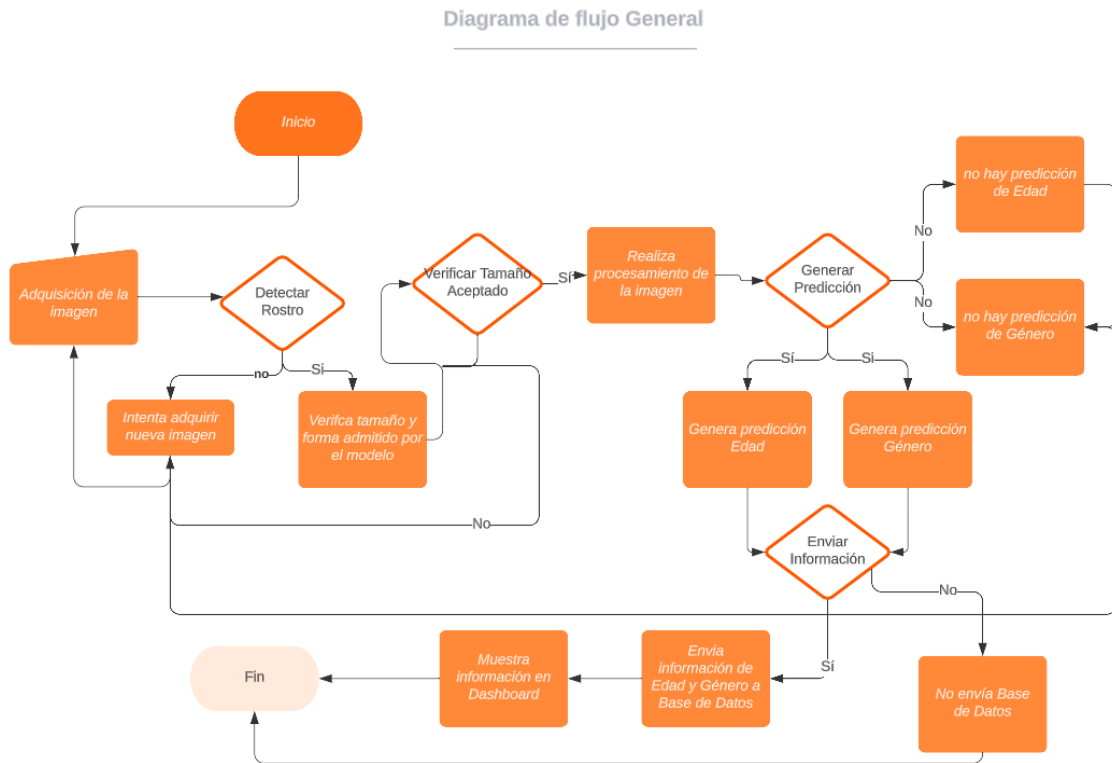


figura 30. Diagrama de flujo del funcionamiento general del dispositivo

Teniendo en cuenta el diagrama de bloques y diagrama de flujo mostrado en las figuras anteriores, iniciamos con la descripción del algoritmo para generar detección de rostros.

4.3.3.2. Funcionamiento Algoritmo detección facial

La detección facial es un tema muy usado en el ámbito de la visión por computadora por lo cual podemos encontrar fácilmente modelos entrenados en internet con muy buena precisión.

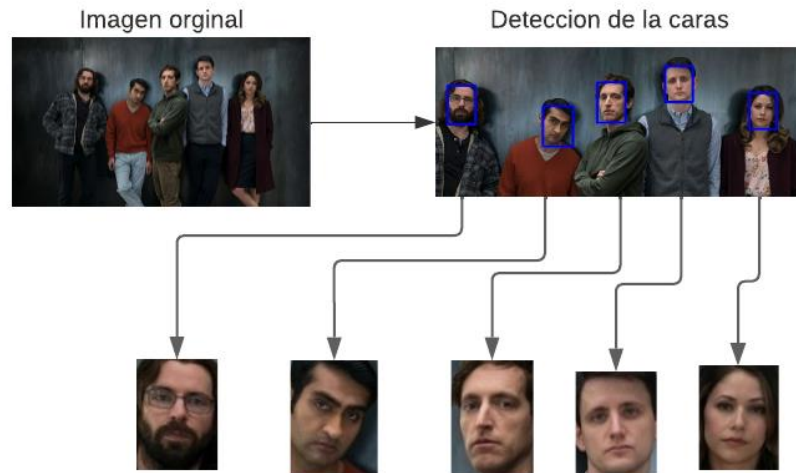


figura 31. Flujo detección de rostros imagen original tomada de [64]

El algoritmo más utilizado fue presentado por primera vez por Viola y Jones. El algoritmo divide la imagen de entrada en secciones rectangulares, y no píxel por píxel como se hacía antes, de tal manera que se mejora el tiempo de procesamiento. En este método se incluyeron nuevos términos como el de imagen integral, la cual se calcula a partir de la imagen original usando pocas operaciones por píxel.

El valor de una imagen integral en una posición (x,y) es igual que la suma de todos los píxeles por arriba y hacia la derecha de esa posición [65] como se evidencia en la siguiente ecuación (1).

$$ii(x, y) = \sum_{x' \leq x, y' \leq y}^i i(x', y') \quad (1)$$

Donde $ii(x,y)$ es la imagen integral y $i(x,y)$ es la imagen original.[65]

La imagen integral, en forma recursiva, es representada por:

$$s(x, y) = s(x, y - 1) + i(x, y) \quad (2)$$

$$ii(x, y) = ii(x - 1, y) + s(x, y) \quad (3)$$

Donde $S(x, y)$ es la suma acumulada en fila, $S(x, -1) = 0$ y $II(-1, y) = 0$. [65]

En el algoritmo Viola-Jones se usan características Haar, las cuales permiten obtener información de una zona en particular. En este algoritmo se usan al menos 3 características Harr.

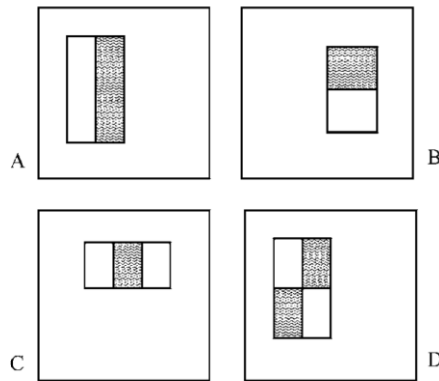


figura 32 - Características Haar . tomada de M. J. J. PAUL VIOLA, "Robust Real-Time Face Detection," Sep. 10, 2001. Accessed: May 04, 2023.

En el algoritmo de Viola-Jones hace un mapeo por las regiones importantes de la cara. Comenzando desde la parte superior de la cara buscando primero si hay pelo, luego busca si hay frente y así hasta llegar al mentón, cabe recalcar que este es un proceso en cascada, lo que quiere decir que si no se cuenta alguna característica no va a buscar la siguiente característica en esa área de la imagen

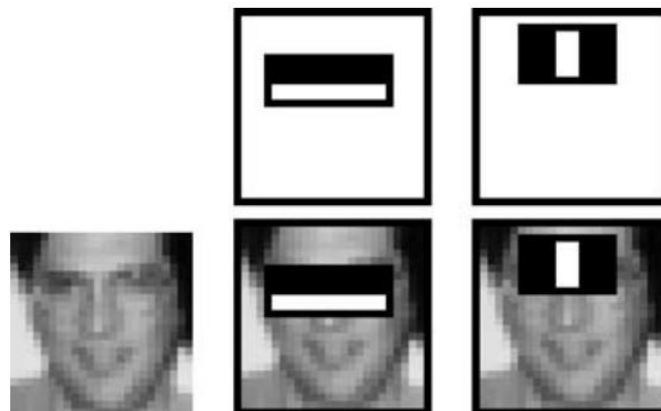


figura 33. Representación de la superposición que se usa para detección de rostro en el algoritmo de Viola Jones tomada de [65]

Teniendo en cuenta lo anterior y cómo funciona el algoritmo para detección de rostros de Viola y Jones, se realiza la adquisición de la imagen por medio una cámara. Una vez se obtiene la imagen con la cámara, esta imagen se envía al modelo de detección de caras, esperando obtener la cara de la imagen. Cuando se obtiene la cara, se procesa aplicando filtros; sin embargo, se le realiza un preprocesamiento a la imagen, se genera un recorte de la imagen para tener más control sobre el funcionamiento del algoritmo de detección de rostros y que en efecto tome la imagen correctamente y que el comportamiento del algoritmo sea más exacto. A continuación, se muestra el código con el que se realiza el preprocesamiento de la imagen:

Inicialmente se importan las librerías como se muestra en la siguiente figura:

```
import cv2
from datetime import date,datetime
```

figura 34. Importación de librerías de preprocesamiento de imágenes para modelo de detección de rostros

Seguido de esto se crea una función para generar un recuadro alrededor del rostro, posterior de hacer la detección del rostro

```
#Declaracion de objeto donde donde hace el seguimiento de las caras
tracker = CentroidTracker(maxDisappeared=50)
# Se inicializa el cursos de la base de datos
cur = conn.cursor()
#Funcion para detectar los rostros, poner los rectangulos en las caras y obtener tamaño de las caras en pixeles
def DeteccionDeCaras(frame):

    frameOpencvDnn=frame.copy()
    #se realiza una transformación a escala de grises de la imagen o frame de entrada
    gray = cv2.cvtColor(frameOpencvDnn, cv2.COLOR_BGR2GRAY)

    faces = faceCascade.detectMultiScale(
        gray,
        scaleFactor=1.1,
        minNeighbors=5,
        #tamaño mínimo en pixeles para hacer una detección
        minSize=(30, 30),
        flags = cv2.CASCADE_SCALE_IMAGE
    )
    faceBoxes=[]
    for (x, y, w, h) in faces:
        #según la cantidad de caras hacce los rectangulos
        cv2.rectangle(frameOpencvDnn, (x, y), (x + w, y + h), (0, 255, 0), 2)
        faceBoxes.append([x,y,x + w,y + h])

    return frameOpencvDnn,faceBoxes
#esta variable maneja la ruta en donde se encuentra el modelo de detección de rostros
cascPath = r"C:\Users\stefa\Downloads\haarcascade_frontalface_alt.xml"
faceCascade = cv2.CascadeClassifier(cascPath)
```

figura 35. Preprocesamiento de imágenes para modelo de detección de rostros implementado

En la figura 35 se observa el código de la función de detección de rostros, adicional se pueden observar los comentarios de cada parte del código relevante para poder traer el modelo y ejecutarlo, así mismo que se genere la detección de rostros y que por cada rostro que detecte me genere un rectángulo.

Una vez que se realiza el proceso anterior la imagen de entrada, se pasa por el modelo y se obtiene la edad y el género, a continuación, se evidencia como se realiza el proceso para el modelo de predicción de género y edad.

4.3.3.3. Estimación de edad y género

Para realizar el proceso de estimación de edad y género, se pueden usar varias técnicas de inteligencia artificial, en este proyecto se usó el **aprendizaje automático**.

El aprendizaje automático se divide en diferentes tipos, en este proyecto se usó aprendizaje supervisado, en el cual el modelo necesita que se le dé el dato de entrada y la respuesta correcta.

Teniendo esto en cuenta se definieron 2 puntos importantes, primero el **data set** que se va a usar y segundo la **estructura de la red** que se va a usar para el entrenamiento.

Prototipo 1

Para el prototipo 1 se parte de la red neuronal trabajada en [66] donde se usa el data set Adience, y una red neuronal de tres capas convolucionales y dos capas completamente conectadas, para realizar la clasificación de edad y género.

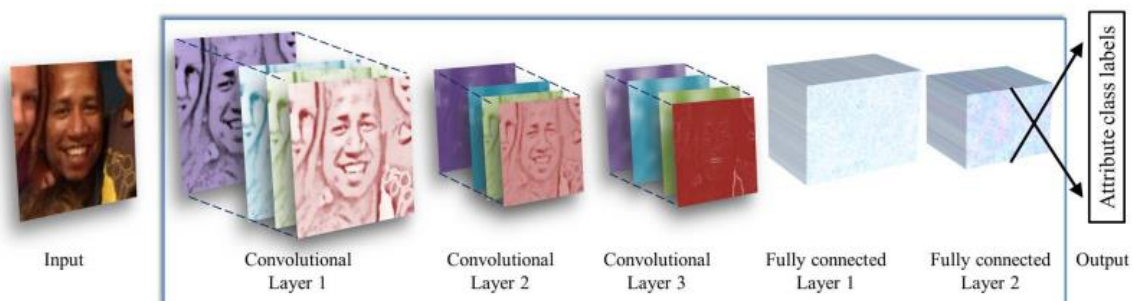


figura 36. Ilustración de la arquitectura usada tomada de [66]

Data set Adience

El data set “Adience”, cuenta con 18.591 imágenes[67], con etiquetas de edad que están agrupadas por 8 rangos de edad (0-2, 4-6, 8-13, 15-20, 25-32, 38-43, 48-53, 60+), el rango

que mayor cantidad de imágenes mayor es de 25 a 32 con una cantidad de 5.296 para tener un total del 98%, como se muestra en la figura 37 para el género femenino se manejan un total de 9372 imágenes y para el género masculino 8120, sin embargo, algunas de las imágenes tienen el género como indefinido en este caso son un aproximado de 1099[68].

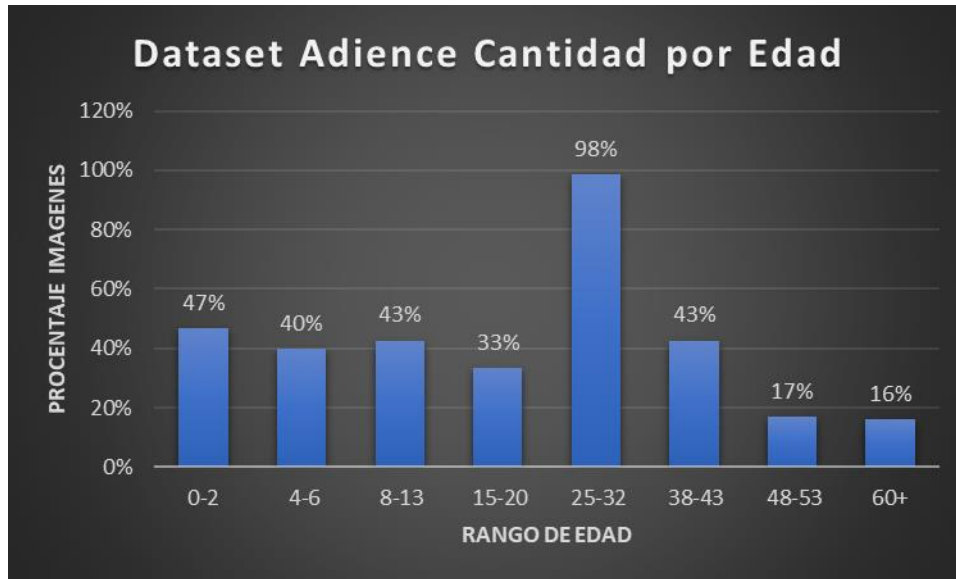


figura 37. gráfica de porcentaje cantidad de imágenes por rango de edad

Con esta red neuronal se realizó un script, donde se le asigna un ID a cada persona y se contabiliza el tiempo de esta persona, a continuación, se muestran algunas pruebas con este script, adicional en anexos del video de prueba del script con esta red neuronal.

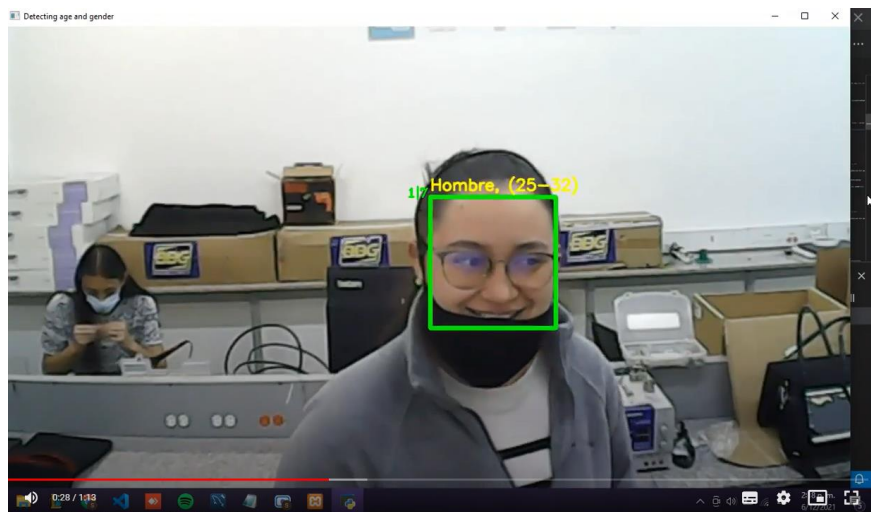


figura 38. Prueba con sujeto 1 y entrenamiento data set Adience

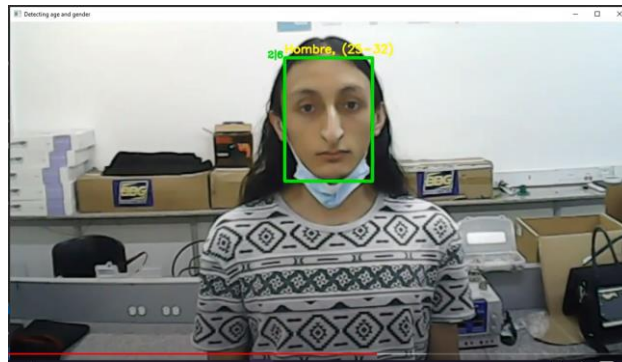


figura 39. Prueba con sujeto 2 y entrenamiento data set Adience

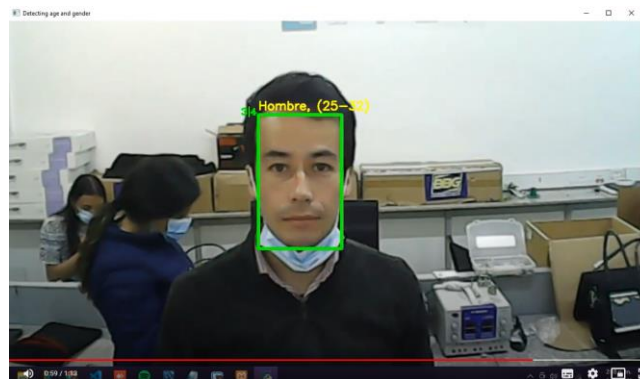


figura 40. Prueba con sujeto 3 y entrenamiento data set Adience

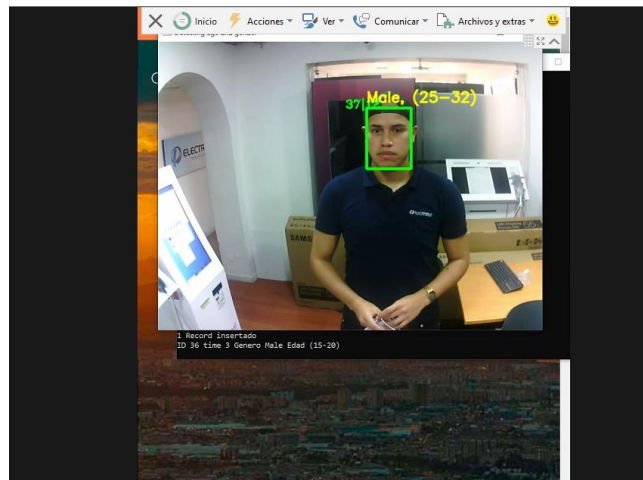


figura 41. Prueba en empresa con entrenamiento data set Adience

En la figura 41 se muestra las pruebas realizadas la empresa Electronika. Adicional Este algoritmo se implementó en una máquina de autogestión, la cual cuenta con una cámara integrada, como se muestra en la figura 42, es importante tener en cuenta que se muestra activo el recuadro donde se pueden ver las personas que han sido detectadas sin embargo esto se activó únicamente para las pruebas en su momento, debido a que en los equipos siempre

se muestra la publicidad o cartelera digitales según corresponda, adicional a lo anterior la precisión del algoritmo no es totalmente certera debido a que las personas tienen tapabocas y son algunas limitaciones indicadas para los algoritmos.

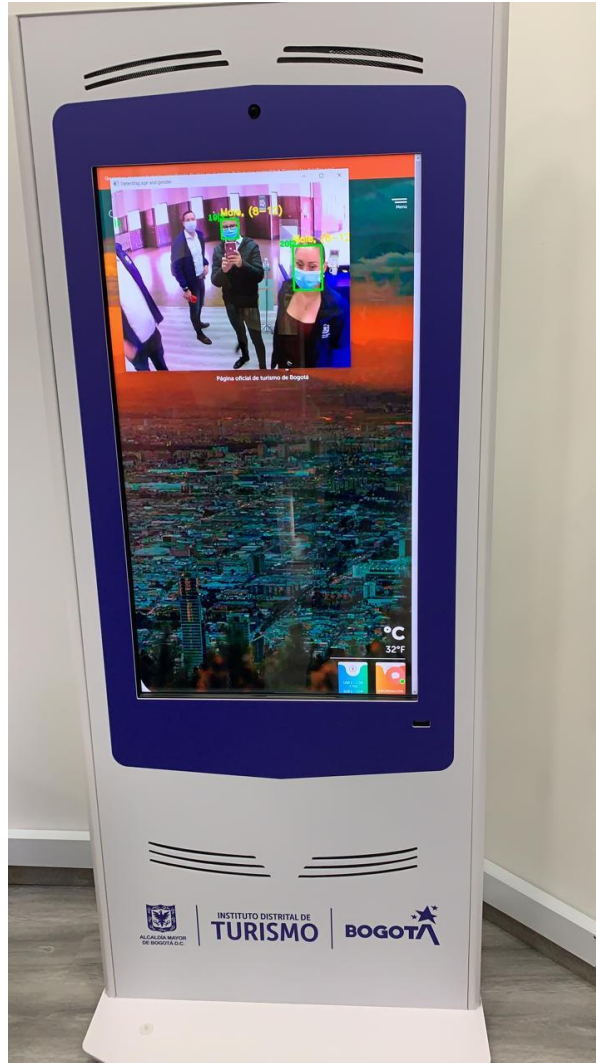


figura 42. Imagen de la maquina autoservicio donde se implementó el primer prototipo.

Prototipo 2

En este proyecto se planteó un segundo prototipo donde se planteó una red neuronal realizando un cambio en la salida en comparación con la red neuronal del prototipo 1, en este caso no se clasifica a la persona en un rango de edades, sino que realizara la estimación de un valor de edad.

Para este algoritmo se usó el data set UTKFaces [69] que es un data set que tiene aproximadamente 23.700 imágenes con sus etiquetas de Edad, Género, Raza y fecha. la

etiqueta de **edad** es un entero de 0 a 116, la etiqueta de **género** es un entero 0 o 1(0-hombre, 1-mujer). La etiqueta de **raza** es un entero de 0 a 4, denotando blanco, negro, asiático, indio y otros (como hispano, latino, del Medio Oriente).

En la siguiente tabla (Tabla 7) se puede observar la distribución de edad que tiene este data set, donde se evidencia que el mayor porcentaje de edades que tiene el data set es de 20 a 30 años.

Tabla 7. Distribución de edad

Rango de edad (Años)	Cantidad de imágenes	Porcentaje
0-10	3216	13.6
10-20	1652	7
20-30	7780	32.8
30-40	4336	18.3
40-60	4307	18.2
60-80	1854	7.83
80+	537	2.27

En la figura 43 podemos ver un poco más claro esta distribución, donde se nota que el mayor número de imágenes está en el rango de edades de 20 a 30 años.

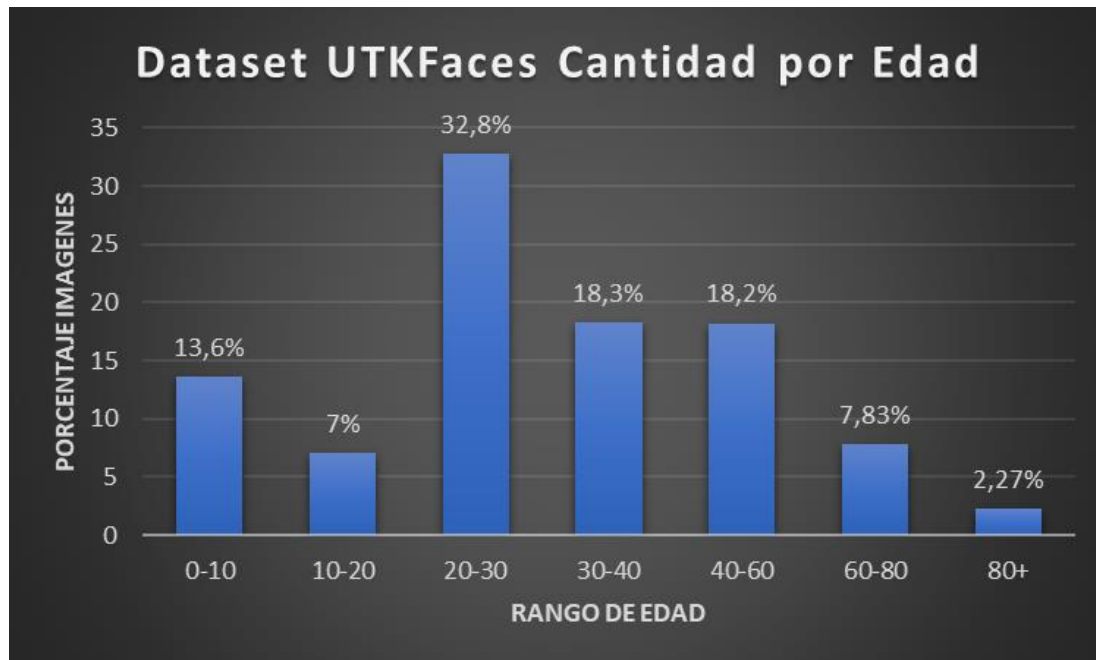


figura 43. gráfica de porcentaje cantidad de imágenes por rango de edad

Distribución de género del dataset UTKFaces

En este data set tenemos un poco más de 12000 imágenes de personas de género masculino, sin embargo, se tiene un muy buen porcentaje de cada género para poder entrenar el modelo y que pueda tener la una precisión muy similar para cada uno de los géneros esto se puede observar en la Tabla 8.

Tabla 8. Distribución género data set

Género	Cantidad de imágenes	Porcentaje
Femenino	11310	47.7
Masculino	12377	52.3

Distribución de Raza dataset UTKFaces

En la distribución de raza tenemos un desbalance amplio, ya que la mayoría de las imágenes corresponden a la categoría ‘blancos’, que se podría decir que son personas de Europa o Norte América. En menor porcentaje está en la categoría otros, donde están los hispanos, latinos y personas del Medio Oriente. Hay que tener en cuenta que es una desventaja, ya que la raza con la que más va a trabajar corresponde a la latina. Sin embargo, es importante tener en cuenta que a pesar de que este es el data set utilizado, no se va a manejar la distribución de raza, únicamente edad y género, esto debido a los objetivos del proyecto y su finalidad. Para este datase se evidencia que hay mayor cantidad de imágenes para el rango de 20 a 30 años de edad, esto se debe a que las imágenes ya vienen etiquetadas de esta manera.

Debido a que la mayor cantidad de imágenes del data set se encuentra en el rango de 20 – 30 años y según un informe del panorama sociodemográfico de la juventud en Colombia emitido por el DANE, se evidencia que para el año 2020 los jóvenes entre los 14 y 28 años representan el 25% de la población total [70] como se logra observar en figura 44. Es importante tener en cuenta esta información, ya que por esta misma razón los data set manejan mayor cantidad de imágenes para estas edades, pues de esta manera facilita el entrenamiento.

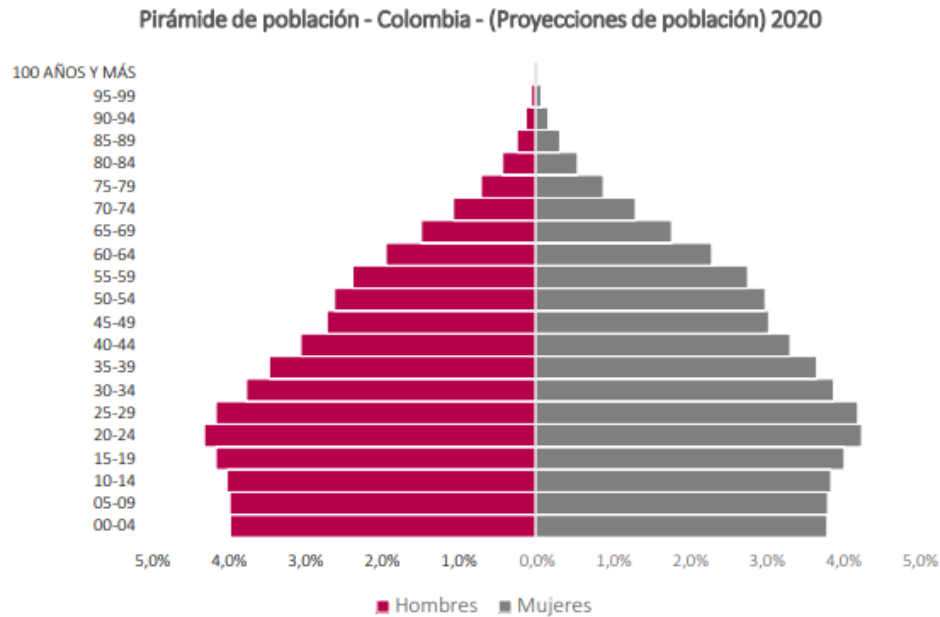


figura 44. Pirámide de población Colombia 2020 tomada de DANE, “Panorama sociodemográfico de la juventud en Colombia,” Sep. 20AD. Accessed: May 03, 2023. [Online].

Para realizar el entrenamiento del modelo de aprendizaje automático, se dividió 3 partes el data set UTKface, una parte para entrenamiento, otra para validación y otra parte para pruebas, donde se usaron 19200 imágenes para entrenamiento, 2134 imágenes para validación y por último 2371 imágenes para pruebas.

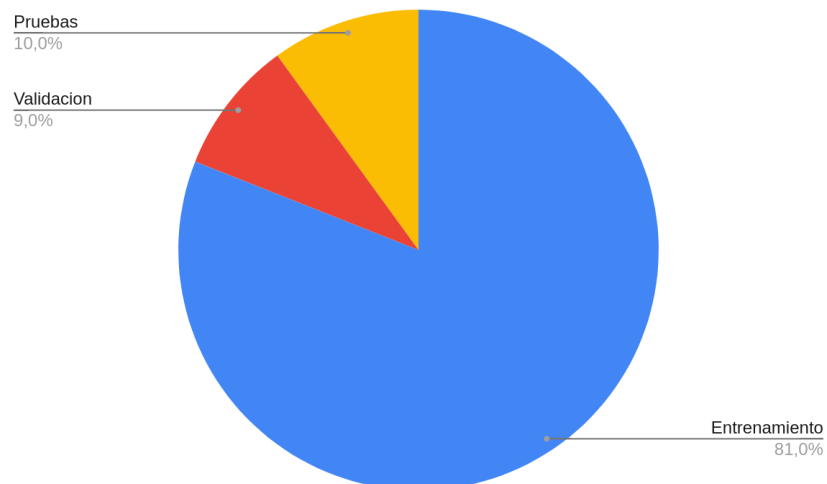


figura 45. Gráfica de división de data set para pruebas, validación y entrenamiento

4.3.3.4. Estructura de los 5 Modelo Diseñados

En este prototipo se diseñó y entreno a un modelo de 5 capas convolucionales y 2 capas densas, pero para llegar a este modelo, se entrenaron un total de 5 modelos con diferentes cantidades de capas, cada uno maneja capas convolucionales tanto para edad como para género, lo que hacen estas capas básicamente es manejar una imagen de entrada, y esa imagen la divide en bloques y se le aplican ciertos filtros los cuales pueden aprender a detectar patrones visuales simples, como bordes, texturas o colores, y también características más complejas.

A continuación, se muestra el modelo implementado con 5 capas convolucionales:

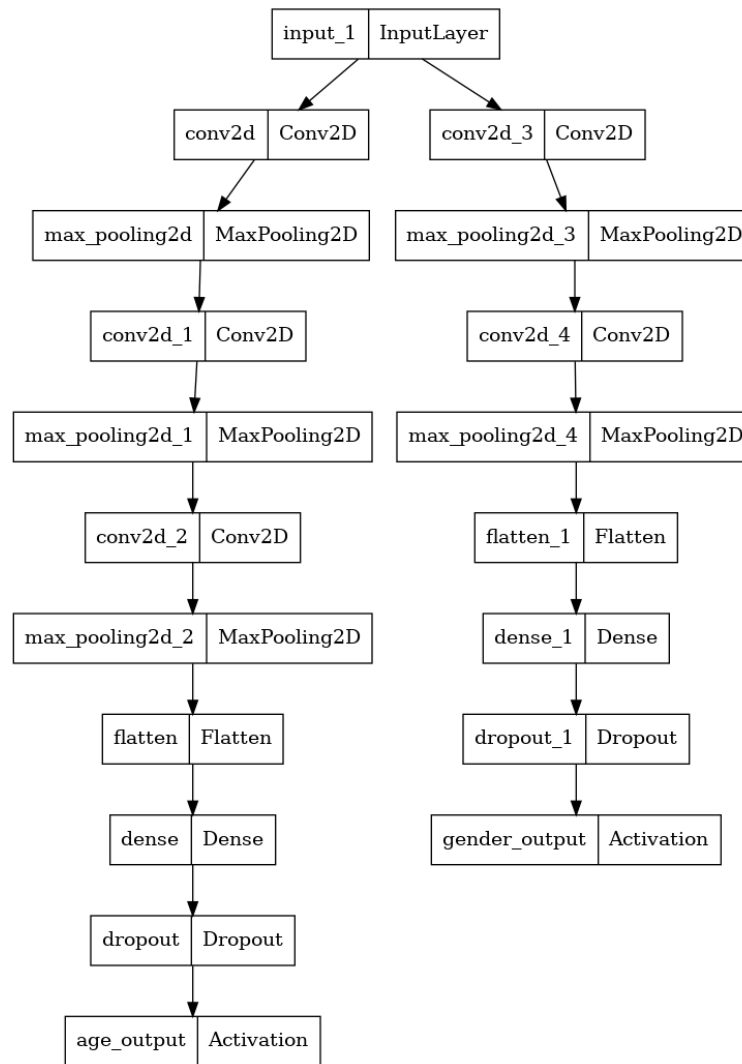


figura 46. Modelo 3 con 5 capas convolucionales

Entrenamiento de modelo con 2 capas convolucionales y 10 Épocas

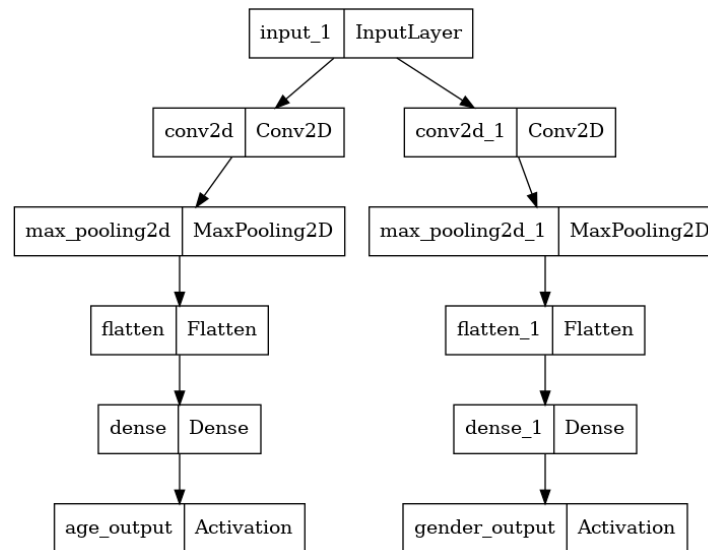


figura 47. Modelo 1 diseñado con dos capas convolucionales

En la figura 47 se evidencia el primer modelo para predicción edad y género, con dos capas convolucionales una para cada salida, teniendo en cuenta que tiene como una única entrada una imagen.

Para el primer entrenamiento con este modelo cabe resaltar que solo se entrenó en 10 épocas lo cual llevó a que el modelo no tuviera el aprendizaje adecuado, ya que puede que este modelo no haya encontrado una tendencia adecuada en el conjunto de datos. Con este modelo se obtuvo una precisión en clasificación de genero de 86% y un puntaje R2 de -2.21

Mean Absolute Error for age feature

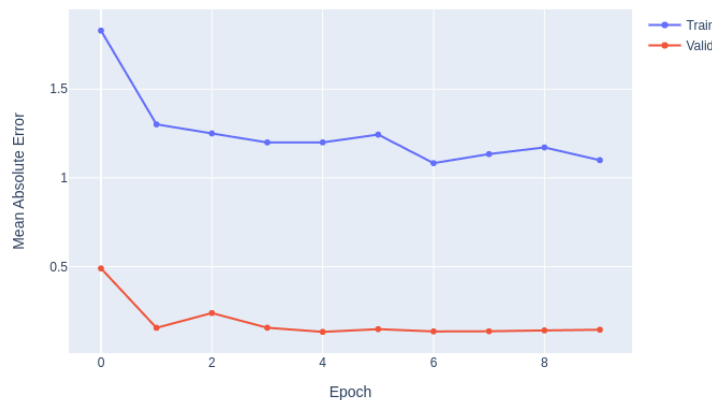


figura 48. Gráfica modelo 2 capas y con 10 épocas

Entrenamiento modelo con 4 capas convolucionales y 100 épocas

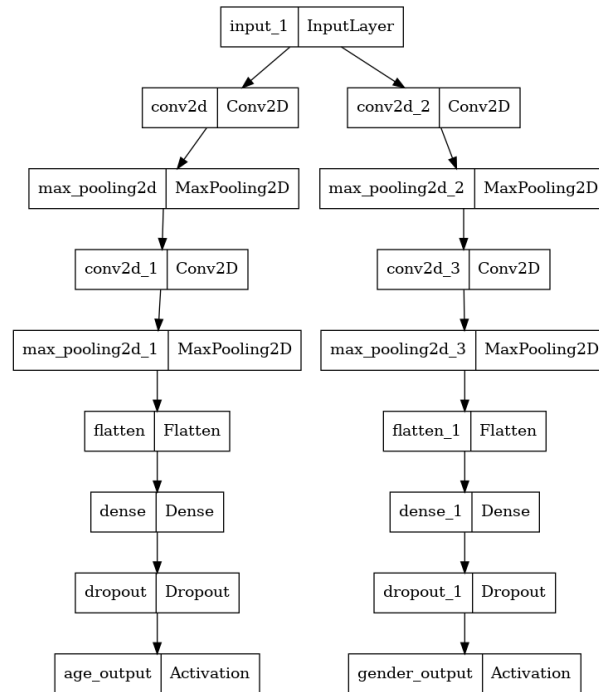


figura 49. Modelo 2 diseñado con cuatro capas convolucionales

En la figura 49, se agregan **2 capas convolucionales** para cada una de las salidas, adicional a las iniciales esto con el fin de que se puedan capturar más características para mejorar el entrenamiento del modelo, adicional se agregan **2 capas Dropout** para **evitar el sobreajuste**, teniendo en cuenta que el **sobreajuste** se da cuando el modelo buscar características muy exactas en las imágenes de entrada (data set) y esto hará que el modelo no tenga la facilidad de reconocer nuevas caras.

En el modelo con 4 capas convolucionales, 2 para edad y 2 para género con 100 épocas se observa que el aprendizaje automático tuvo muchas variaciones, en algunas épocas estaba generando mayor cantidad errores, como se puede evidenciar en el rango de la época 0 a la 40. Con este modelo se obtuvo una precisión de genero de 86% y un puntaje R2 de -2.21

Mean Absolute Error for age feature

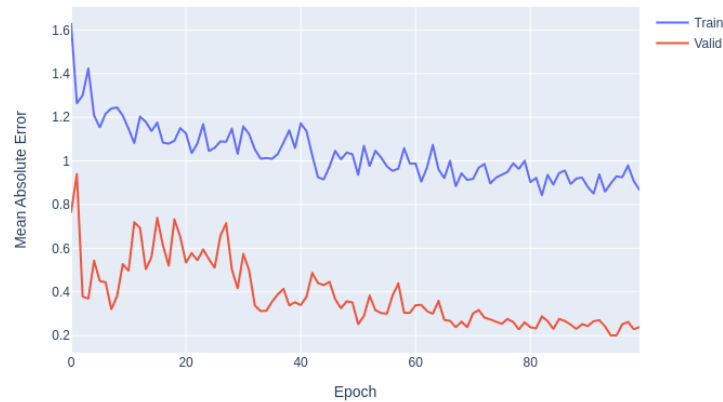


figura 50. Gráfica modelo 4 capas y con 100 épocas

Entrenamiento modelo con 5 capas convolucionales y 1000 épocas

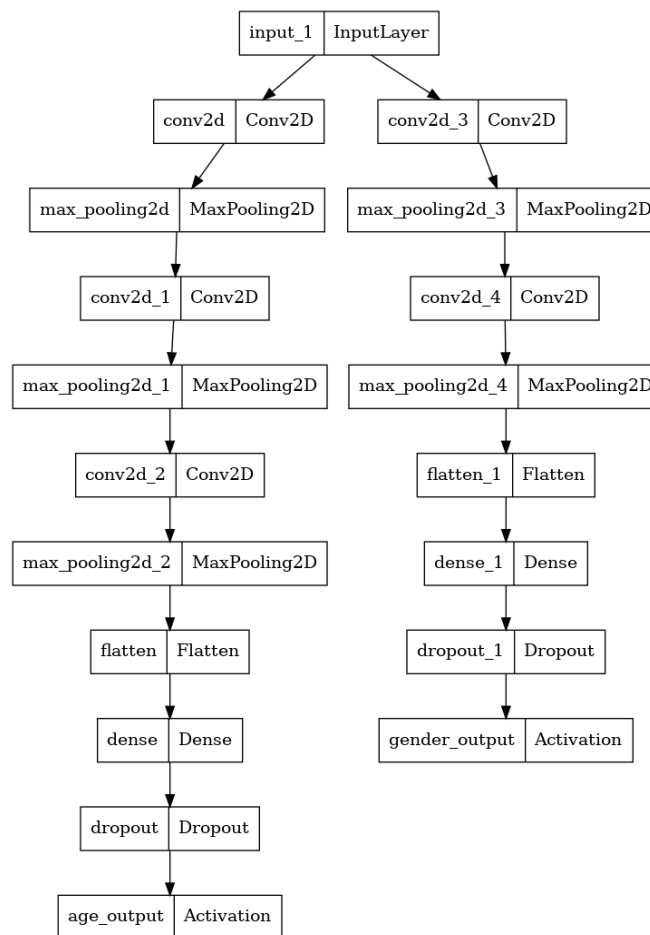


figura 51. Modelo 3 aumentando capa para edad

En la anterior figura 51, se muestra el modelo en el cual se agrega una capa convolucional adicional a las anteriores, pero únicamente para edad, esto es debido a que para el entrenamiento de edad es más complejo con respecto al género y requiere de las capas profundas para generar un aprendizaje más robusto con respecto a la edad ya que con el género agregando la segunda capa convolucional se evidencio que ya funcionaba correctamente.

En este entrenamiento se manejan 5 capas convolucionales, 3 para edad y 2 para género, además, podemos observar que al principio (época 0 a las 200) del aprendizaje, estaba teniendo bastantes errores en el entrenamiento, sin embargo, a partir de la época 200 tuvo menos errores en el entrenamiento. Con este modelo se obtuvo una precisión de género en clasificación de 87% y un puntaje R2 de -0.65

Mean Absolute Error for age feature

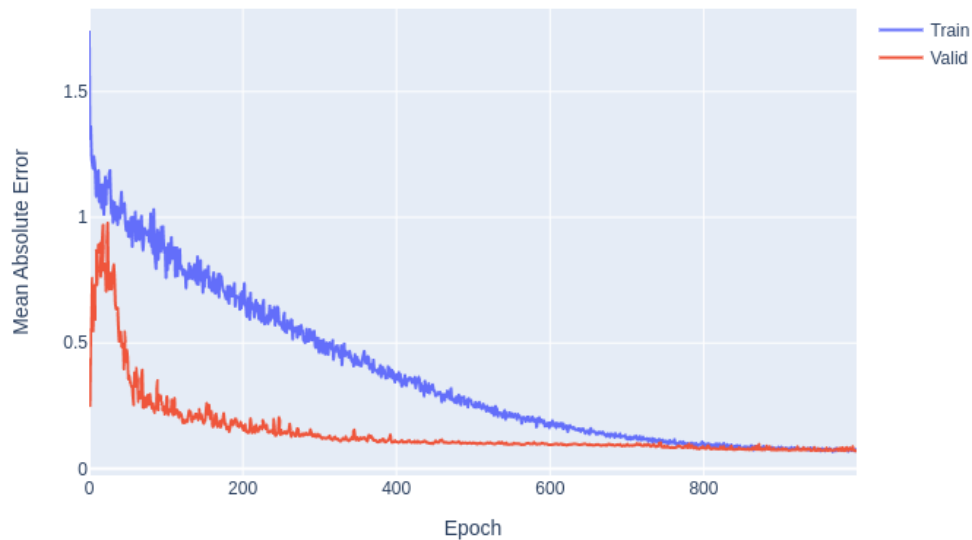


figura 52. Gráfica Modelo 5 capas y con 1000 épocas

Modelo entrenado con 6 capas convolucionales y 100 épocas

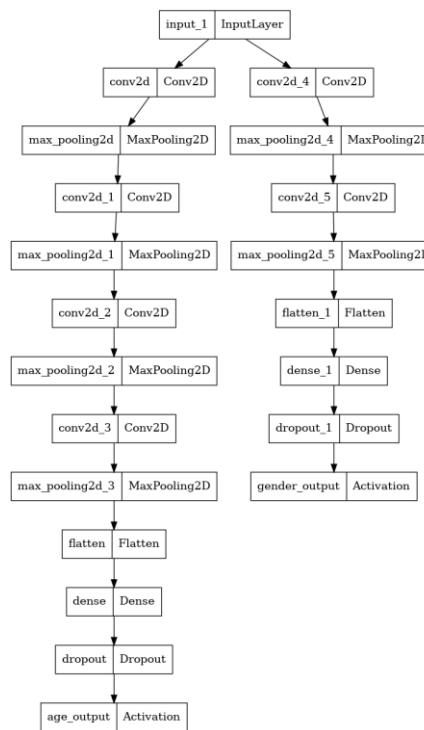


figura 53. Modelo 4 se agrega una capa convolucional adicional a la anterior para edad

Con este modelo se hizo una variación a la estructura del modelo en el que se agregaron 4 capas convolucionales para la edad y 2 capas para el género, obteniendo una red neuronal más robusta. En este entrenamiento se evidencio que al modelo le cuesta más entrenarse ya que tiene más capas. En la gráfica podemos ver que el error de este modelo fue muy oscilante y nunca se estabilizo. Con este modelo se obtuvo una precisión en clasificación de genero de 84% y un puntaje R2 de -213.15



figura 54. Gráfica Modelo 6 capas y con 100 épocas

Entrenamiento modelo con 7 capas convolucionales y 2000 épocas

Por último, se entrenó un modelo con 2000 épocas ya que en el modelo anterior con 7 capas convolucionales las cuales 5 capas son para edad y 2 son para género y 1000 épocas se obtenía un puntaje R2 de -2.72 , lo cual generaba muchos errores en la predicción de edad. El último modelo tuvo una precisión en clasificación de género de 90% y un puntaje R2 de 0.44. En la gráfica de entrenamiento se evidencio que, aunque pareciera que las pérdidas se estabilizaran aproximadamente en la época 600, las perdidas comenzaron a aumentar un poco en la época 800 aproximadamente, esto se puede suceder por un sobre entrenamiento en el modelo, sin embargo, se corrigió a partir de la época 1500 aproximadamente.

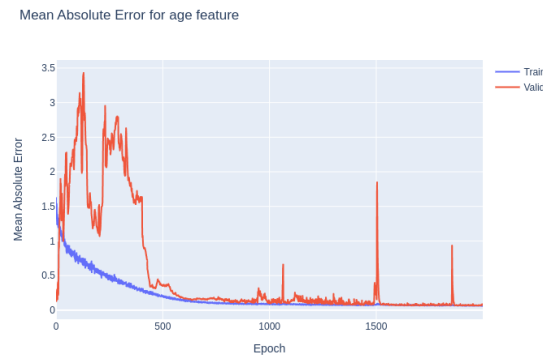


figura 55. Gráfica Modelo con 7 capas y con 2000 épocas

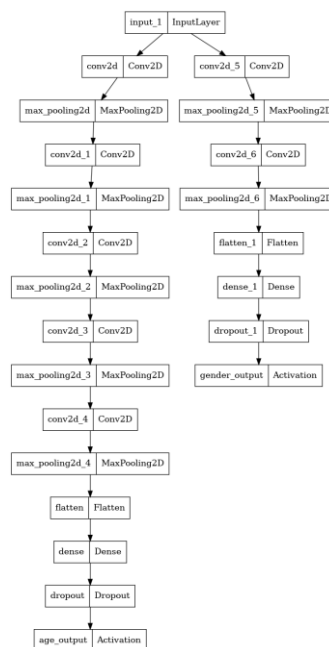


figura 56. Modelo 5 con una capa convolucional adicional para la edad

4.3.3.5. Funcionamiento Algoritmo Detección de edad y genero

Con un programa en Python que se hace uso de los algoritmos de detección de rostros y estimación edad y género.

El programa en Python hace el seguimiento de las caras, asignándoles un ID, y contabilizando el tiempo de permanencia. También realizar el procesamiento de la imagen y envía la información a la base de datos y se genera una breve explicación de sus funciones a continuación:

DeteccionDeCaras: esta función es la encargada de hacer uso del modelo de detección de caras.

Entradas: imagen (Arreglo de la imagen)

Salida: imagenOpencvDnn (Imagen con los rectángulos de las caras), cajaCaras (Arreglo con las posiciones de las caras en pixeles)

```
#Función para detectar los rostros, poner los rectangulos en las caras y obtener tamaño de las caras en pixeles
def DeteccionDeCaras(frame):

    frameOpencvDnn=frame.copy()
    gray = cv2.cvtColor(frameOpencvDnn, cv2.COLOR_BGR2GRAY)

    faces = faceCascade.detectMultiScale(
        gray,
        scaleFactor=1.1,
        minNeighbors=5,
        minSize=(30, 30),
        flags = cv2.CASCADE_SCALE_IMAGE
    )

    faceBoxes=[]
    for (x, y, w, h) in faces:
        cv2.rectangle(frameOpencvDnn, (x, y), (x + w, y + h), (0, 255, 0), 2)
        faceBoxes.append([x,y,x + w,y + h])

    return frameOpencvDnn,faceBoxes

cascPath = r"C:\Users\stefa\Downloads\haarcascade_frontalface_alt.xml"
faceCascade = cv2.CascadeClassifier(cascPath)
```

figura 57.Código Función detección de rostros.

En la figura 57 se observa el código de la función de detección de rostros, en la cual se implementa el algoritmo de detección de rostros de viola y jones, adicional se observa el arreglo que se genera para los rectángulos que bordean el rostro y los ponen de cierto color, adicional recibe la imagen, al recibirla la transforma a escala de grises.

- **prededir_edad_género:** Esta función recibe la imagen y hace uso del modelo de predicción para devolver la edad y género.

```

# Crea una lista con los géneros posibles
generos = ['Hombre', 'Mujer']
# Carga del modelo entrenado en formato h5 (Edad y genero)
modelo = keras.models.load_model(r'C:\Users\stefa\Documents\U\2023-1\Caras2\model7.h5')
# Función hacer la predicción
def predecir_edad_genero(imagen):
    """
    Entrada: imagen
    Salida: Edad (Entero), Genero (Texto)
    """
    resultado = modelo.predict(imagen, verbose=0)
    edad = float(resultado[0][0])
    edad = int(edad * 116)
    genero = generos[np.argmax(resultado[1])]
    return edad, genero

```

figura 58. Función para predecir edad y género

En la figura 58 se observa la función para generar la predicción de edad y género, inicialmente se crea un array con los géneros que se van a predecir, en este caso Hombre o Mujer, seguido de esto se carga el modelo ya entrenado e implementado, la función recibe como entrada una imagen, se obtiene la edad guardada en la variable con el mismo nombre, se declara como un entero para multiplicarlo por 116 que es el límite que maneja de edades el data set, seguido de esto se tiene genero el cual según el valor me indicara que valor del arreglo es, por último la función retorna edad y genero para ser mostrada e insertada en base datos.

4.3.3.6. Base de datos

Para este proyecto se utilizó PostgreSQL como gestor de bases de datos, en el cual implementamos una base de datos de 1 sola tabla con 5 columnas.

Column Name	#	Data Type	Default	A
ABC Edad	5	varchar(50)		
ABC Genero	4	varchar(50)		
123 Tiempo	3	int		
🕒 Hora	2	timestamp	CURRENT_TIMESTAMP	
123 ID	1	int		

reco
123 ID
🕒 Hora
123 Tiempo
ABC Genero
ABC Edad

figura 59. Estructura de tabla en Base de datos

	ID	Hora	Tiempo	Genero	Edad
1	57	2023-05-03 17:35:30	50	Male	25
2	58	2023-05-03 17:35:45	70	Male	28
3	59	2023-05-03 17:35:57	10	Male	32
4	60	2023-05-03 17:36:11	90	Female	22

figura 60. Datos insertados en la base de datos

4.3.3.7. Diseño de aplicación web

Se diseñó una plataforma web, en la cual se puede visualizar las tablas de las personas que han sido detectadas. Esta plataforma web tiene la estructura de un dashboard, donde se tiene 4 cajas con información relevante como lo es total de personas, total de mujeres, total hombres y rango de edades. Este dashboard también tiene la tabla donde se ven los últimos 10 registros de hombres y los últimos 10 registros de mujeres. Por último, se tiene una gráfica con los porcentajes de hombres y mujeres.

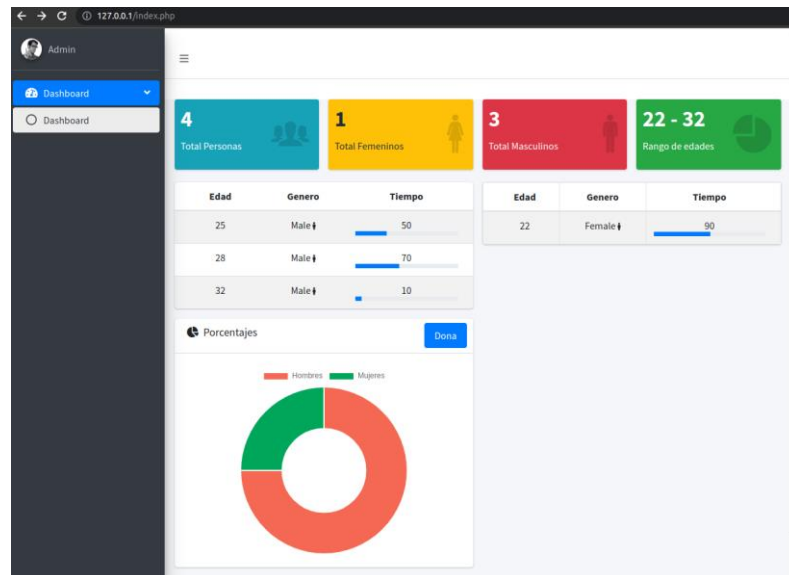


figura 61. Dashboard Diseñado con información consultada de la Base de datos

Con la finalidad de hacer pruebas en tiempo real, la base de datos se encuentra en un servidor propio, adicional se puede validar la información del dashboard ingresada hasta la fecha por medio del siguiente enlace <http://dashboard.pitechnology.site/>.

5. RESULTADOS Y ANÁLISIS DE RESULTADOS

5.1. Resultados de implementación

5.1.1. Algoritmo de Reconocimiento Facial

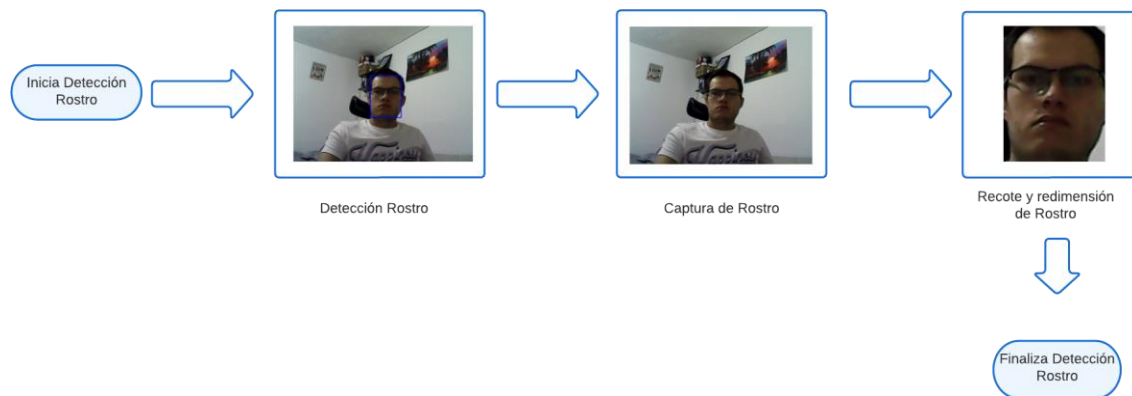


figura 62. Proceso realizado por algoritmo de Reconocimiento Facial

Teniendo en cuenta que, al implementar el algoritmo de detección de rostros, realizamos el preprocesamiento de la imagen para verificar que se está tomando correctamente el rostro de la imagen que está capturando, se realizan algunas pruebas para corroborar el correcto funcionamiento de este proceso adicional como se muestra en la figura 63.

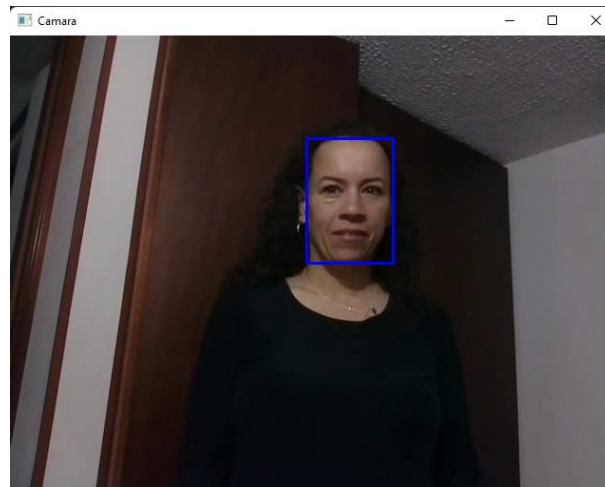


figura 63. Prueba preprocesamiento de imagen para algoritmo de reconociendo facial

En la figura 63 se evidencia como al detectar el rostro genera un recuadro azul indicando que efectivamente se encontró el rostro.

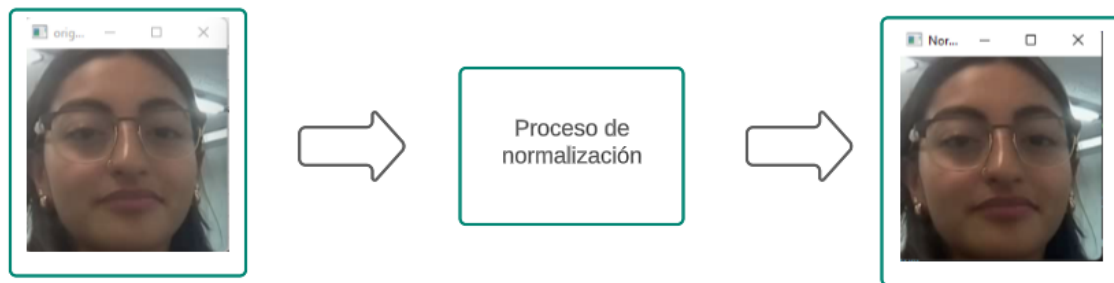


figura 64. Normalización de imágenes al realizar detección de rostros.

En la figura 64 realiza el proceso de normalización de una imagen, con eso se busca que al realizar el reconocimiento de rostros sea más efectivo.

Adicionalmente se muestran otras pruebas con respecto a la detección de rostros y el preprocesamiento de la imagen, se evidencian en la figura 65.

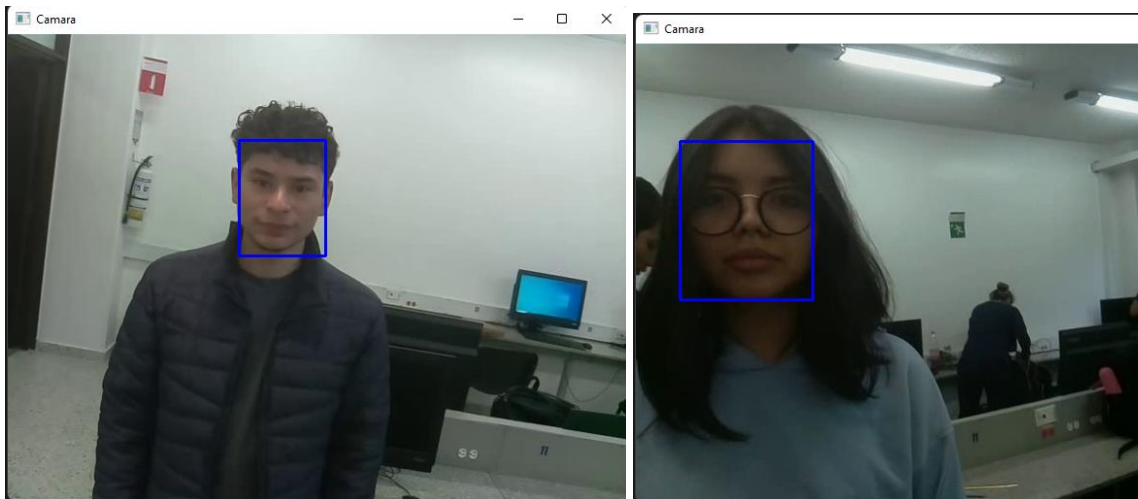


figura 65. Prueba detección de rostros

5.1.2. Algoritmo de Edad y genero

Se diseñó un modelo de inteligencia artificial de una entrada y 2 salidas. A partir de este modelo se hicieron diferentes pruebas realizando variaciones en la estructura, donde se evidenció que uno de los factores más significativos para tener mejores resultados es la cantidad de capas. Además, se evidenció que, en el entrenamiento, al aumentar la cantidad de épocas, se obtenían mejores resultados.

Implementación del modelo diseñado y entrenado

Se implementó el modelo 3 en una Raspberry pi 4 dotada con una cámara Raspberry v1.3, haciendo uso del programa Python que se diseñó, donde se evidencio que, al implementar el modelo en tiempo real, se obtienen valores que varían mucho, esto dependiendo de posición de la cara, la iluminación del entorno, la altura de la cámara y la expresión de la persona.

A continuación, se muestra dos de las pruebas que se dieron con las características nombradas anteriormente, en las cuales se obtiene en una de ellas (figura 66) resultados bastante acertados ya que el sujeto de prueba tiene como edad real 24 y su género es masculino y la predicción fue la misma. Esto con el fin de demostrar que en algunos casos cuando se maneja iluminación, no estando en contra luz y la posición del rostro son favorables.



figura 66. Prueba sujeto 1 acertada en datos de predicción

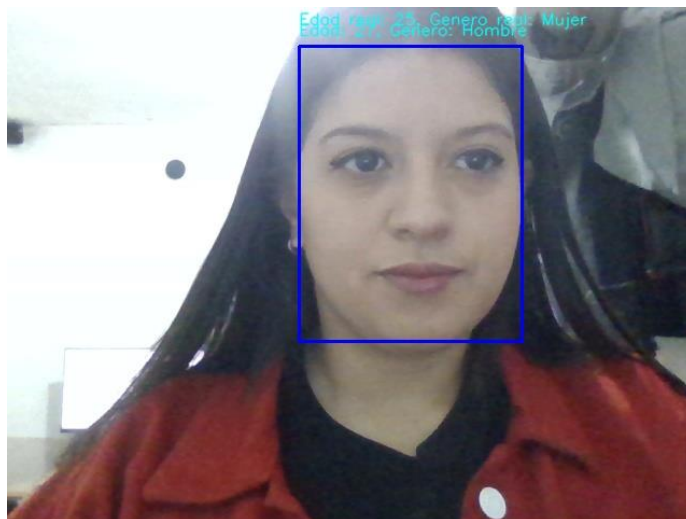
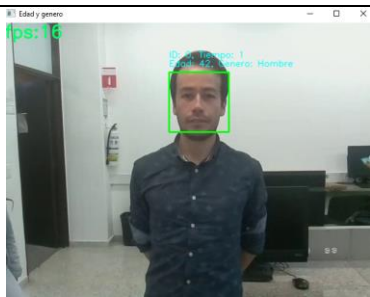
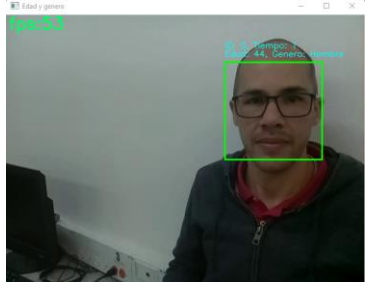



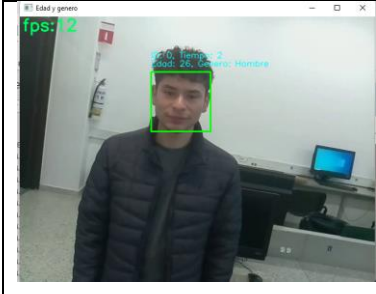
figura 67. Prueba sujeto 2 no acertado en datos de predicción

Para la figura 67, se evidencia la prueba con condiciones similares al sujeto 1, sin embargo se evidencia en la imagen contra luz, para el sujeto 2 se evidencia que su edad es de 25 años y la predicción se generó con una edad de 27 años, con respecto al género si tiene error en esta predicción ya que indica que es hombre, sin embargo su género es femenino, esto se puede dar debido a que al hacer el entrenamiento del modelo con el data set utilizado, presentan más imágenes con género masculino que femenino.

Adicional a lo anterior se realizan pruebas con otros sujetos, como se muestra en las siguientes figuras:

Tabla 9. Pruebas de predicción de edad y género

<i>Imagen Tomada</i>	<i>Edad Real</i>	<i>Genero Real</i>	<i>Edad Estimada</i>	<i>Genero Estimado</i>
	34	Hombre	42	Hombre
	46	Hombre	44	Hombre
	21	Mujer	22	Mujer

	22	Hombre	26	Hombre
---	----	--------	----	--------

En la Tabla 9 se evidencian algunas pruebas realizadas con sujetos diferentes, en este caso para el primer sujeto con una edad estimada de 42 años y con una edad real de 34 años se evidencia un desfase, sin embargo, cumple con los requerimientos solicitados por el cliente. Para el segundo sujeto con edad estimada 44 y para el cuarto sujeto con edad estimada de 26 se encuentra un desfase de solo 2 años y por último para el sujeto 3 en la cual la edad estimada es de 22 hay un desfase de solo 1 año con respecto a la edad real. Adicional para la estimación de genero se observa correctamente la predicción para los cuatro casos.

Al procesar el algoritmo se tiene en cuenta los FPS (fotogramas por segundos) se mantiene en un rango de 1 a 11, si se implementa en otro dispositivo con 8 núcleos en el procesador mejora a un rango de 11 a 18 FPS, como se muestra en la siguiente figura, sin embargo, se mantiene estable con la Pi 4.

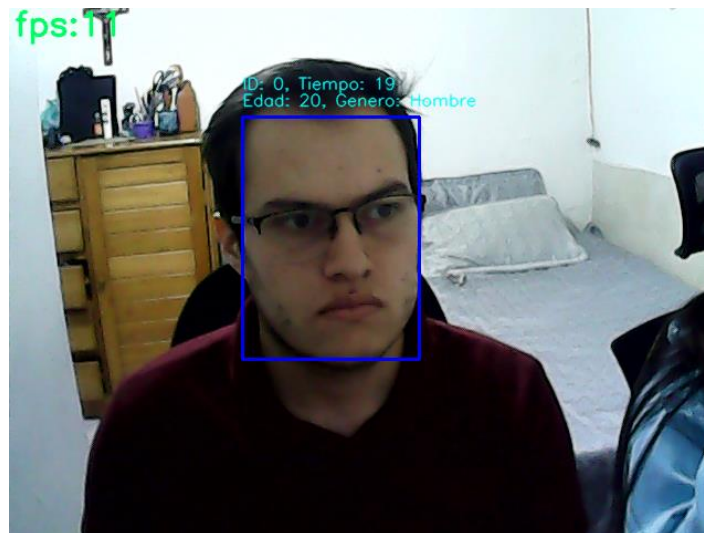


figura 68. Evidencia FPS dispositivo 8 núcleos

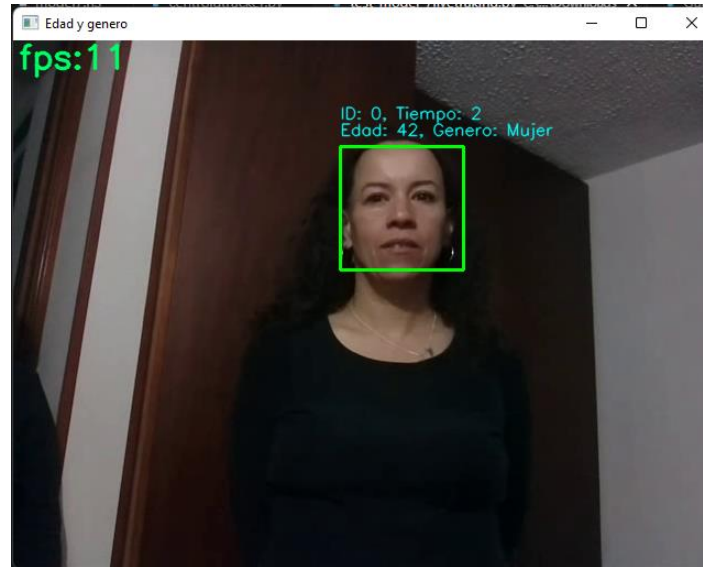


figura 69. Evidencia FPS dispositivo de 8 núcleos.

5.1.3. Modelos para Edad

Se realizaron pruebas con 100 sujetos de la población bogotana, con los modelos 3 y 5 ya que estos modelos fueron los que tuvieron mejores resultados realizando las pruebas de entrenamiento con el data set, para el modelo 3 maneja un Promedio de error 2,83% y para el modelo de 5 un promedio de error 5,17% con respecto a la edad y género, estos valores se obtuvieron de la siguiente manera:

Teniendo en cuenta la edad real y la edad estimada por el algoritmo se toman estos dos valores y se aplica la siguiente fórmula para cada uno de los sujetos:

$$\text{Porcentaje de error} = \frac{\text{Valor Real} - \text{Valor esperado}}{\text{Valor Esperado}} * 100$$

Ecuación 1. Porcentaje de error

Teniendo el resultado de todos los porcentajes de error pro cada sujeto, se prosigue a sacar un promedio del error total para los 100 sujetos realizando este proceso ya tenemos el porcentaje total para cada uno de los modelos.

A continuación, se observa las 2 grafica para los modelos 3 y 5 donde se evidencia en color azul la edad real y en color rojo la edad estimada. (figura 70, figura 71)

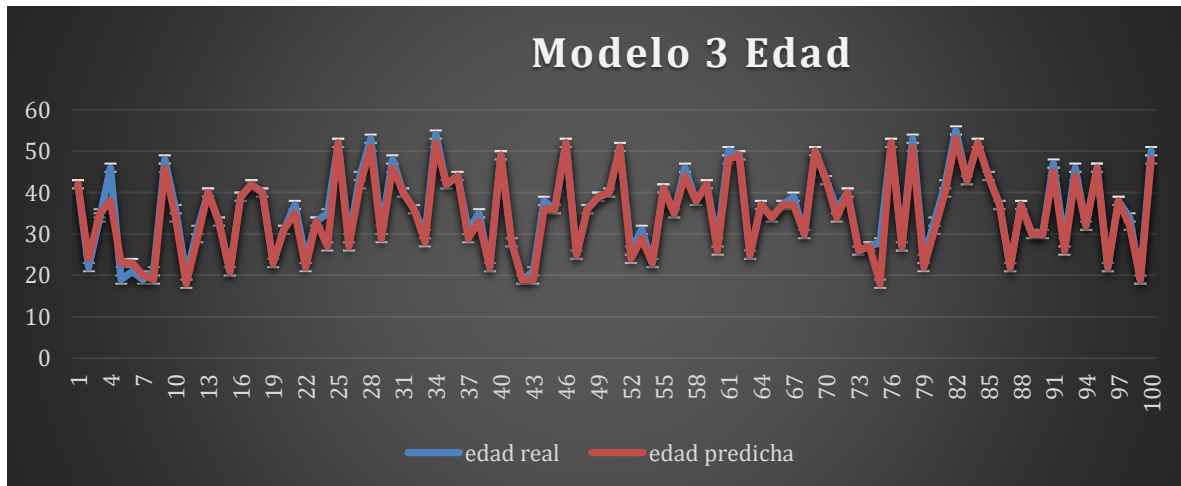


figura 70. Grafica modelo 3 comparación de edad real y estimada

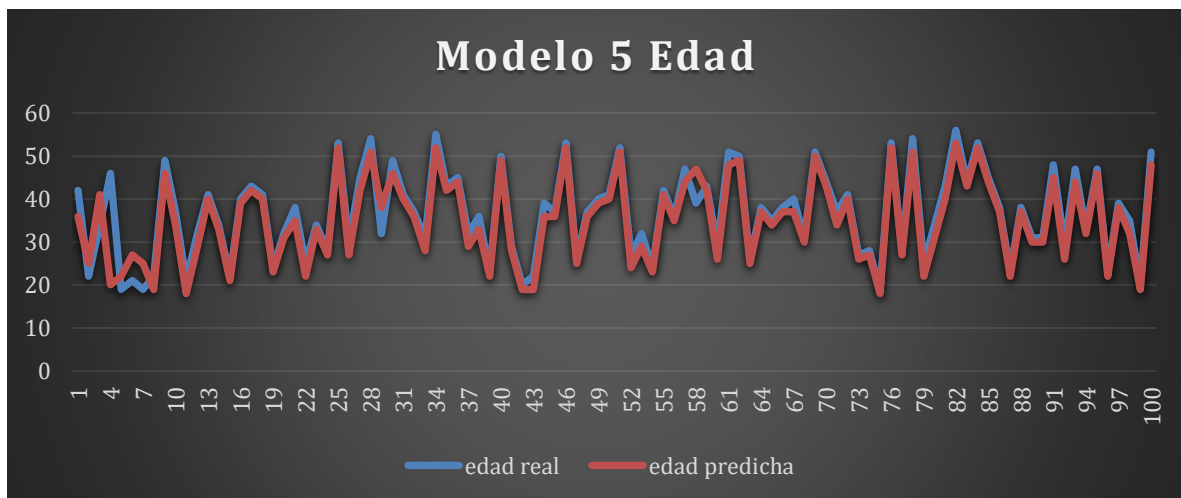


figura 71. Grafica modelo 5 comparación de edad real y estimada

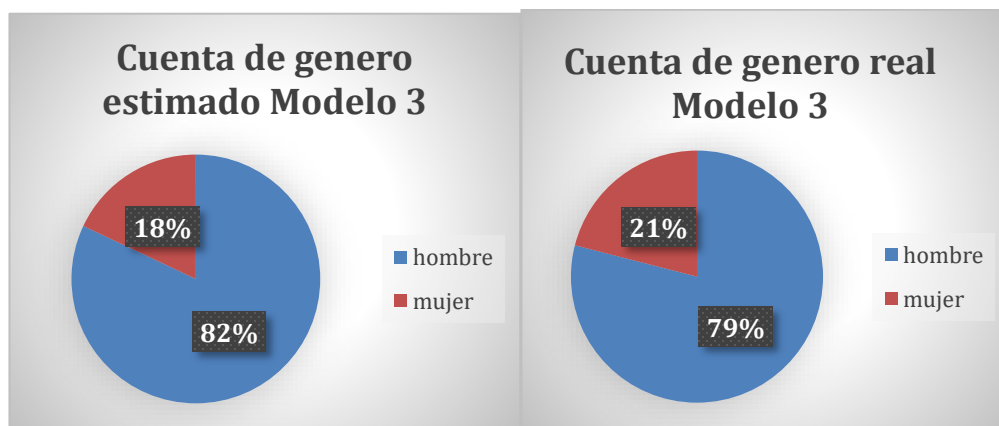


figura 72. Modelo 3 porcentajes para predicción real y estimada de género

Para el modelo 3 tenemos la estimación de género en la cual se observa que hay un porcentaje de error del 3%, como se muestra en la figura 72.

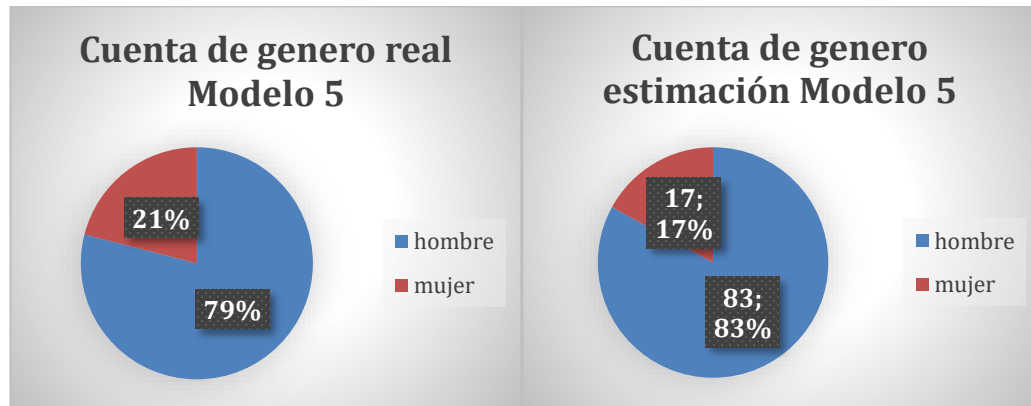


figura 73. Modelo 5 porcentajes para predicción real y estimada de género

Para el modelo 5 tenemos la estimación de género en la cual se observa que hay un porcentaje de error del 4%, como se muestra en la figura 73.

Pruebas con imágenes del data set

Se realizaron pruebas con 2304 imagen de la data set, las cuales se seleccionaron antes de realizar el entrenamiento, y representan el 10% del data set. Estas pruebas se dividieron por género, donde se obtuvo que el modelo tiene una precisión del 94% detectando género de hombres, y una 93% para mujeres. Se obtuvo un valor R2 de 0,54.

Tabla 10.Resultados obtenidos de género con prueba de 2304 imágenes del data set

Género	Precisión	Cantidad de imágenes
Masculino	0,94	1224
Femenino	0,92	1080

De esta prueba, se obtuvo una imagen, en la cual hay 16 rostros con su edad real y la predicción generada por el modelo. En color verde están los resultados donde el género fue correcto y la edad no tiene una diferencia mayor a 8 en relación con los datos reales.

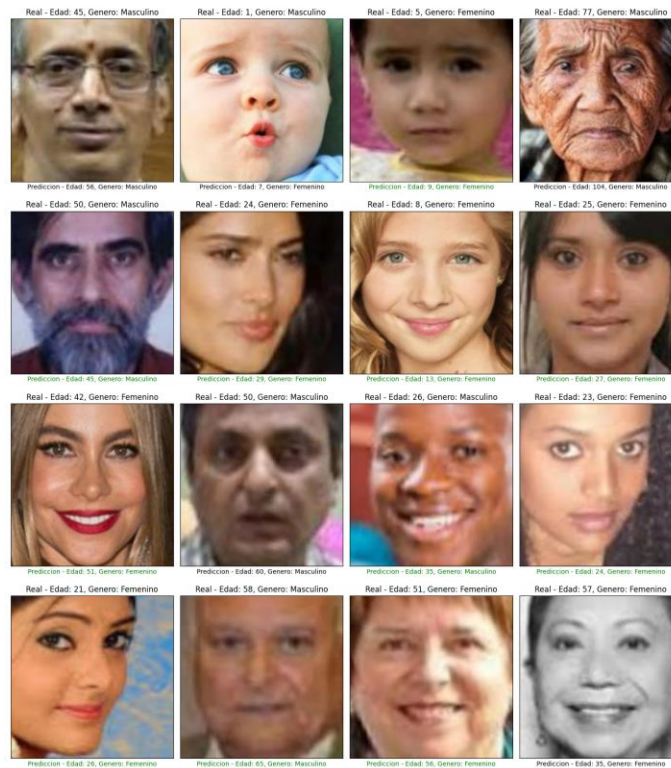


figura 74. Imágenes DataSet Prueba edad real vs edad esperada tomada de [69]

5.2. Análisis de cumplimiento requerimientos

Requerimientos Funcionales:

Tabla 11. Tabla Cumplimiento Requerimientos Funcionales

Código	Descripción	Cumplimiento
RFU01	Se puede realizar la instalación del algoritmo en sistemas operativos Windows y Linux	Se cumple, debido a que se maneja un sistema para Linux que en este caso es la raspberry y una implementación de un subsistema en Windows como se solicitó.
RFU02	Se debe generar una prueba funcional, para comprobar que el dispositivo realiza la medición de edad y de género y recopila la información correctamente.	Se cumple, se muestra en las pruebas que realiza la medición de edad y genero correctamente y con una o más persona.
RFU03	Se debe detectar una edad aproximada del individuo que se está detectando en rango de error entre -8 y $+8$ años	Se Cumple, ya que al generar la estimación de edad lo hace por estos rangos, no los excede
RFU04	Se deben almacenar edad, género, tiempo de permanencia y fecha en una base de datos, que se encuentra en un servidor web y borra información cada que el usuario lo necesite.	Se cumple, todo queda almacenado en una base de datos.
RFU05	Se debe mostrar los datos registrados en un dashboard.	Se cumple, debido a que en el dashboard se muestra genero los rangos de edades y el tiempo.


RFU06	Se debe detectar el género del individuo que se está detectando, este debe ser de un 90% acertado	Se cumple, debido a que se detecta el género del individuo, es muy acertado y cumple con el porcentaje establecido, incluso en las pruebas se evidencia un 100%
RFU07	Se debe agregar un contador para tomar el tiempo de permanencia de los sujetos.	Se cumple, se agrega un contador pro medio de código para que indique el tiempo de cada persona viendo la cartelería digital
RFU08	El dispositivo se debe poder adaptar fácilmente con otros desarrollos que pueda solicitar el cliente	Se cumple, debido a que es posible adaptar el dispositivo a otros equipos y con respecto a otros desarrollos también.
RFU09	El dispositivo se debe poder adaptar fácilmente con otros desarrollos que pueda solicitar el cliente	No se logró, esto debido a que antes del momento de la entrega del dispositivo no se generaron desarrollos nuevos por parte de la empresa para comprobar la aplicación, sin embargo en los existentes sí.



5.3. Pruebas de Funcionamiento

Pruebas imagen con diferentes rellenos

Se realizó una prueba donde se usaron 3 valores de relleno para la misma imagen de una cara cuya persona tiene 21 años, en esta prueba se usó rellenos de 10, 20 y 30px. De esta manera, se obtuvieron 3 imágenes que se pueden ver en la Tabla 12 cada una de estas imágenes se realizó la predicción en los 5 modelos. En la tabla se muestran los diferentes valores obtenidos en cada uno de los modelos, se pudo evidenciar que el resultado más erróneo fue obtenido en el modelo 4, el cual y fue entrenado en 100 épocas, estos resultados tienen mucha coherencia, si tenemos en cuenta la Tabla 12 donde podemos ver que el modelo 4 fue el que tuvo el peor puntaje r^2 con un valor de -213.15. Por otro lado, vemos que el modelo 5, tuvo un resultado con buena precisión. Cabe resaltar que el parámetro de relleno que le demos a la cara puede llegar a hacer que la predicción tenga un error del 6,3% si se valida con los resultados obtenidos en el modelo de 5.

Tabla 12. Prueba de una imagen con 3 rellenos (10,20 y 30px), la edad real es de 21 años.





Imagen	Modelo 1	Modelo 2	Modelo 3	Modelo 4	Modelo 5
	34	1	44	-295	21

	33	12	49	71	28
	38	50	48	56	30

Pruebas con diferentes posiciones de la cara

Se realizó una prueba con una misma persona, pero con diferentes posiciones de la cara, evidenciando que, según la posición de la cara, se pueden tener resultados muy diferentes.

Tabla 13 Predicción de Edad con rostro en diferentes posiciones

Predicción Edad					
<i>Imagen</i>	<i>Edad real</i>	<i>Predicción de edad</i>	<i>Error en predicción de edad</i>	<i>Predicción de género</i>	<i>Error en predicción de género</i>
	24	26	8,333333333	Hombre	0
	24	43	79,16%	Hombre	0
	24	53	120,83%	Mujer	100%
	24	60	150%	Hombre	0

Pruebas con varias personas

En esta prueba se realiza la predicción con dos sujetos al mismo tiempo, detectando, edad, género y contabilizando tiempo de permanencia.

Se obtiene que para el sujeto uno se predice una edad de 27 años y como género hombre y con un tiempo de 118 segundos, para el sujeto 2 predice una edad de 20 años como género femenino y con un tiempo de 27 segundos. Para este caso se logra evidenciar que el desfase de edades no es muy grande y que los géneros son predichos de manera correcta.

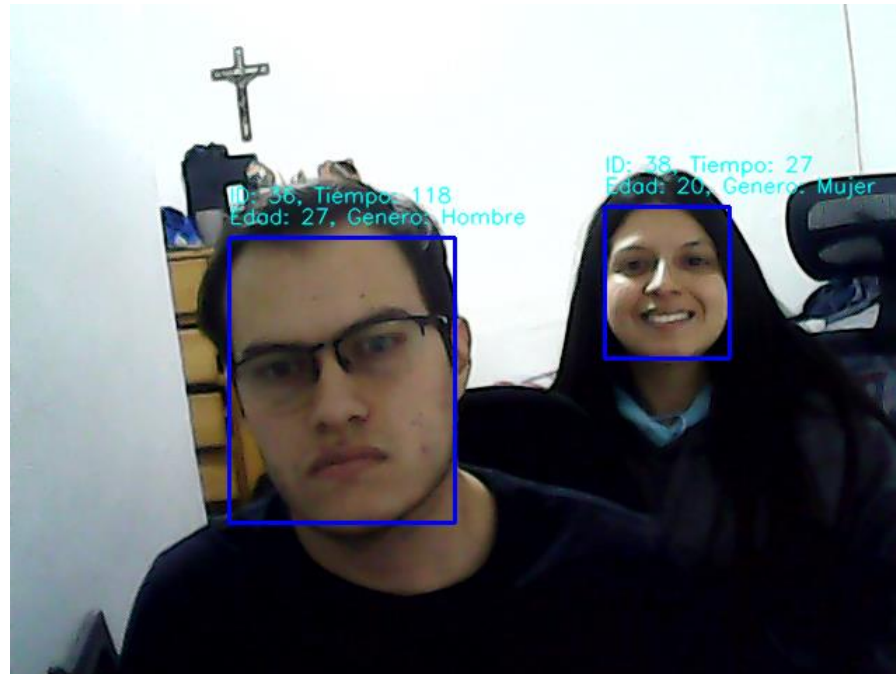

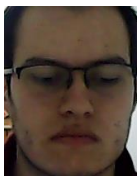








figura 75. Prueba de predicción con dos sujetos al tiempo

Pruebas con cámara USB

A continuación, se muestra en la Tabla 14 las pruebas realizadas con la cámara USB FIT0701, se realizan las pruebas con diferentes individuos, indicando la edad real que maneja el sujeto, la predicción que se realiza, el error aproximado teniendo en cuenta la edad que se predijo con respecto a la edad real, adicional se muestra el género real del sujeto y la predicción del mismo.

Tabla 14. Pruebas con cámara USB

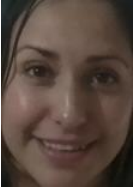
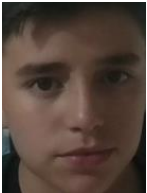

<i>Imagen</i>	<i>Edad real</i>	<i>Predicción de edad</i>	<i>Error en predicción de edad</i>	<i>Género Real</i>	<i>Predicción de género</i>
	12	11	8,33%	Hombre	Hombre
	24	21	12,5%	Hombre	Hombre

	25	21	16%	Mujer	Mujer
	30	26	13,33%	Mujer	Mujer
	30	30	0	Hombre	Hombre
	9	13	44,44%	Mujer	Mujer
	39	39	0	Hombre	Hombre
	42	37	11,90%	Hombre	Hombre

Pruebas con cámara Raspberry

En la Tabla 15 se evidencias las pruebas realizadas con la cámara Raspberry Camera V1.3, de igual manera se muestra la edad real del sujeto, la predicción de la edad, un porcentaje de error, el género real y la predicción del género.

Tabla 15. Pruebas con cámara Raspberry Camera V1.3

<i>Imagen</i>	<i>Edad real</i>	<i>Predicción de edad</i>	<i>Error en predicción de edad</i>	<i>Genero Real</i>	<i>Predicción de genero</i>
	36	32	11,11%	Mujer	Mujer
	12	14	16,66%	Hombre	Hombre
	39	39	0	Hombre	Hombre

6. CONCLUSIONES

Con el presente proyecto, se cumplió el objetivo de diseñar e implementar un dispositivo electrónico y su correspondiente interfaz para recopilar información demográfica, en cuanto a edad y género, utilizando un algoritmo de reconocimiento facial. En este caso la metodología aplicada resultó satisfactoria debido a que se implementaron dos algoritmos: uno para detección espacial de rostros y el otro cuya finalidad tenía estimar la edad y la clasificación binaria del sujeto como hombre o mujer.

Durante la etapa de desarrollo e implementación del proyecto, se pudo observar que la elección del entorno, en la que se realiza la adquisición de la imagen, tiene una influencia significativa con la precisión de los resultados, debido a la influencia de factores no controlables como la iluminación, la posición del rostro, las expresiones faciales y el tamaño de la imagen.

Para llevar a cabo el proyecto, se decidió utilizar un ordenador Raspberry Pi 4, el cual se manejó como uno de los componentes centrales con una considerable flexibilidad en las decisiones de diseño tanto de software como de hardware. Además, facilitó la identificación de los otros subsistemas necesarios para alcanzar una solución que cumpliera con los requisitos planteados. Adicional en las pruebas de rendimiento funcionó a una tasa de entre 1 y 11 fotogramas por segundo. Estas pruebas se realizaron con un máximo de 4 personas, ya que era el número de personas que podían ser detectadas dentro del campo visual de la cámara. Se considera que la tarjeta de desarrollo funcionó adecuadamente para las necesidades del proyecto, ya que tiene la capacidad de realizar la detección de rostros, clasificación de género y predicción de edad al menos una vez por segundo con un máximo de 4 personas simultáneamente.

7. RECOMENDACIONES

Durante la implementación del proyecto, se obtuvieron resultados favorables tanto en la clasificación de género como en la predicción de edad. Sin embargo, se sugiere utilizar un detector de puntos faciales para lograr resultados más precisos con respecto a la estimación de edad y clasificación de género. Esto permitirá filtrar los rostros en posición frontal, ya que

el modelo utilizado puede detectar rostros incluso en posición lateral. Cabe mencionar que el modelo de predicción de género y edad fue entrenado únicamente con rostros en posición frontal, es decir, cuando el sujeto se encuentra mirando directamente hacia la cámara.

Es importante destacar que este proyecto se centró en la población de Bogotá, Colombia, con edades comprendidas entre los 12 y 42 años. Se recomienda extender esta investigación a otros sectores geográficos para determinar si los resultados obtenidos son consistentes.

Si se busca un mayor rendimiento en el tiempo de procesamiento, se recomienda utilizar una placa de desarrollo como la Orange Pi 5. Esta placa cuenta con el doble de núcleos (Cores) y ofrece una ranura M.2 que permite la conexión de un disco de estado sólido. Estas características pueden mejorar significativamente el tiempo de procesamiento del modelo. Adicional tiene un costo similar al de la Raspberry Pi4.

8. REFERENCIAS BIBLIOGRÁFICAS

- [1] Glosario Terminología Informática, “data set.” <http://www.tugurium.com/gti/termino.php?Tr=data%20set> (accessed May 07, 2023).
- [2] Tesauro de la UNESCO, “Inteligencia artificial.” <https://vocabularies.unesco.org/browser/thesaurus/es/page/concept3052> (accessed May 07, 2023).
- [3] Real Academia Española, “predecir.” <https://dle.rae.es/predecir> (accessed May 07, 2023).
- [4] Glosario Terminología Informática, “software.” <http://www.tugurium.com/gti/termino.php?Tr=software> (accessed May 07, 2023).
- [5] MarketingDirecto.com, “Los retos en la medición de audiencias digitales: hacia el consenso de la industria,” Feb. 01, 2021. <https://www.marketingdirecto.com/digital-general/digital/retos-medicion-audiencias-digitales-consenso-industria> (accessed May 01, 2023).

- [6] beabloo, “Business analytics,” 2018. <https://www.beabloo.com/analisis-datos/> (accessed May 01, 2023).
- [7] BOARDS, “Reconocimiento Facial,” 2018.
- [8] Cao Hong Nga; Khai-Thinh Nguyen; Nghi C. Tran; Jia-Ching Wang, “Transfer Learning for Gender and Age Prediction,” 2020. <https://ieeexplore.ieee.org/document/9258347/authors#authors> (accessed Jun. 07, 2023).
- [9] Difei Gao; Lili Pan; Risheng Liu; Rui Chen; Mei Xie, “Correlated warped Gaussian processes for gender-specific age estimation,” 2015. <https://ieeexplore.ieee.org/document/7350774/authors#authors> (accessed Jun. 07, 2023).
- [10] Nemanja Krajčinović; Nataša Avramović; Marko Adžić; Živko Bojović, “People movement tracking based on estimation of ages and genders,” Belgrade, Serbia, Nov. 25, 2020.
- [11] G.án M.ías Scarel, “sistema de reconocimiento facial,” 2010. Accessed: May 05, 2023. [Online]. Available: https://drive.google.com/drive/u/1/folders/1579hbhy7tCNclFC19nNpJgJpgTpbX_SK
- [12] Sergio Domínguez Fernández, “CLASIFICACIÓN FACIAL EN GÉNERO Y EDAD,” UNIVERSIDAD DE MÁLAGA, MALAGA, 2014. Accessed: May 05, 2023. [Online]. Available: https://drive.google.com/drive/u/1/folders/1579hbhy7tCNclFC19nNpJgJpgTpbX_SK
- [13] Romel Daniel Castro Arias, “SISTEMA DE CONTROL DE ACCESO AL PERSONAL DE LA LAVADORA DE JEANS FASHION MEDIANTE RECONOCIMIENTO FACIAL,” UNIVERSIDAD TÉCNICA DE AMBATO, Ecuador, 2016. Accessed: May 26, 2023. [Online]. Available: https://repositorio.uta.edu.ec/bitstream/123456789/20347/1/Tesis_t1107ec.pdf
- [14] OpenVino, “age-gender-recognition-retail-0013.”
- [15] S. B. C. A. C. D. N. and N. B. P. R. C. Koushik, *Determination of Age, Gender, Dress Color and Type of a Person by Convolutional Neural Network (CNN)*. 2020.

- [16] Claro, “Medición de Audiencias,” 2020. <https://www.claro.com.co/empresas/sectores/retail/medicion-audiencias/>. (accessed May 01, 2023).
- [17] G. A. S. G. NICOLAS DIAZ SALAZAR, “RECONOCIMIENTO DE OBJETOS EN IMÁGENES USANDO APRENDIZAJE PROFUNDO,” UNIVERSIDAD AUTÓNOMA DE OCCIDENTE, Santiago de cali, 2018. Accessed: May 05, 2023. [Online]. Available: https://drive.google.com/drive/u/1/folders/1579hbhy7tCNclFC19nNpJgJpgTpbX_SK
- [18] DAVID LEONARDO CASTAÑO SAAVEDRA-JUAN DAVID ALONSO SIERRA, “SISTEMA DE RECONOCIMIENTO FACIAL PARA CONTROL DE ACCESO A VIVIENDAS,” Universidad Católica, Bogotá, 2019. Accessed: May 05, 2023. [Online]. Available: <https://repository.ucatolica.edu.co/server/api/core/bitstreams/c627cd7f-698d-4dc8-b552-736c318cae4c/content>
- [19] L. D. C. Selene Ximena Andrade, “Plan de Empresa dedicada al Reconocimiento de Rostros por Visión Artificial,” Jul. 2020, Accessed: May 26, 2023. [Online]. Available: <https://bibliotecadigital.univalle.edu.co/bitstream/handle/10893/18250/3744%20A248?sequence=1>
- [20] Nicolas Garzon Rodriguez, “Sistema de recomendación de contenido por medio de reconocimiento facial,” LA UNIVERSIDAD DE LOS ANDES, 2019. Accessed: May 26, 2023. [Online]. Available: <https://repositorio.uniandes.edu.co/bitstream/handle/1992/44546/u830679.pdf?sequence=1&isAllowed=y>
- [21] P. P. Cruz, *Books on Google Play Inteligencia artificial: con aplicaciones a la ingeniería*. 2010.
- [22] NetApp, “¿Qué es la inteligencia artificial?” <https://www.netapp.com/es/artificial-intelligence/what-is-artificial-intelligence/> (accessed May 27, 2023).

- [23] SAS, “Inteligencia Artificial Qué es IA y Por Qué Importa.” https://www.sas.com/es_cl/insights/analytics/what-is-artificial-intelligence.html#world (accessed May 27, 2023).
- [24] Fernando de la Fuente, “Inteligencia Artificial: La Detección facial y sus aplicaciones,” Jun. 2019. https://www.amurait.com/actualidad/ai_deteccion_facial_amurait/ (accessed May 27, 2023).
- [25] Varios autores, “Conceptos y métodos en Visión por Computador,” p. 372, 2016, [Online]. Available: <https://intranet.ceautomatica.es/sites/default/files/upload/8/files/ConceptosyMetodosenVxC.pdf>
- [26] L. Institute, “Reconocimiento facial: Descubre cómo funciona y quién (y para qué) lo utiliza.” <https://www.lisainstitute.com/blogs/blog/reconocimiento-facial-como-funciona-quien-utiliza>
- [27] C. B. Carrión, “REDES CONVOLUCIONALES,” Universidad de Sevilla, 2020.
- [28] Jordi Torres inteligencia artificial, “Redes Neuronales Recurrentes,” Sep. 22, 2019. <https://torres.ai/redes-neuronales-recurrentes/> (accessed May 03, 2023).
- [29] Sitiobigdata.com, “Convolutional neural network: Intro a Padding y Stride.” <https://sitiobigdata.com/2019/05/01/convolutional-neural-network-padding-stride/#>
- [30] opengenius, “Operación Conv2D en TensorFlow.” <https://iq.opengenus.org/conv2d-in-tf/> (accessed Jun. 07, 2023).
- [31] T. V2.11.0, “2D convolution layer,” 2022. https://www.tensorflow.org/api_docs/python/tf/keras/layers/Conv2D#used-in-the-notebooks
- [32] Precious Chima, “Activation Functions: ReLU & Softmax,” 2020. <https://medium.com/@preshchima/activation-functions-relu-softmax-87145bf39288> (accessed Jun. 07, 2023).
- [33] T. V2.11.0, “tf.keras.layers.Activation,” 2022. https://www.tensorflow.org/api_docs/python/tf/keras/layers/Activation

- [34] Strahinja Zivkovic, “How to apply Batch Normalization in PyTorch,” 2021. <https://datahacker.rs/017-pytorch-how-to-apply-batch-normalization-in-pytorch/> (accessed Jun. 07, 2023).
- [35] T. V2.11.0, “tf.keras.layers.BatchNormalization.” https://www.tensorflow.org/api_docs/python/tf/keras/layers/BatchNormalization
- [36] paperswithcode, “Max Pooling.” <https://paperswithcode.com/method/max-pooling> (accessed Jun. 07, 2023).
- [37] T. V2.11.0, “tf.keras.layers.MaxPool2D,” 2022.
- [38] Amar Budhiraja, “Dropout in (Deep) Machine learning,” 2016. <https://medium.com/@amarbudhiraja/https-medium-com-amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machine-learning-74334da4bfc5> (accessed Jun. 07, 2023).
- [39] T. V2.11.0, “tf.keras.layers.Dropout.” https://www.tensorflow.org/api_docs/python/tf/keras/layers/Dropout
- [40] data-flair, “Keras Convolution Neural Network Layers and Working.” <https://data-flair.training/blogs/keras-convolution-neural-network/> (accessed Jun. 07, 2023).
- [41] T. V2.11.0, “tf.keras.layers.Flatten.” https://www.tensorflow.org/api_docs/python/tf/keras/layers/Flatten
- [42] Florian Malard and Stéphanie Olivier-Van Stichelen., “Fully Connected (Dense),” 2021. <https://epynn.net/Dense.html> (accessed Jun. 07, 2023).
- [43] T. V2.11.0, “tf.keras.layers.Dense,” 2022. [tensorflow.org/api_docs/python/tf/keras/layers/Dense](https://www.tensorflow.org/api_docs/python/tf/keras/layers/Dense)
- [44] Intel, “Modelado de IA: impulso de la inteligencia en los análisis.” [https://www.intel.es/content/www/es/es/analytics/data-modeling.html#:~:text=Los modelos de IA \(aprendizaje,decisiones en la inteligencia empresarial.](https://www.intel.es/content/www/es/es/analytics/data-modeling.html#:~:text=Los modelos de IA (aprendizaje,decisiones en la inteligencia empresarial.)
- [45] P. Económicas, “Cómo entrenar modelos de Machine Learning.” https://www.percepcioneseconomicas.cl/data-science/como-entrenar-modelos-de-machine-learning/#¿Que_significa_entrenar_modelos_de_machine_learning
- [46] Neuroons, “¿QUÉ SON ‘ENTRENAMIENTO’ E ‘INFERENCIA’ EN INTELIGENCIA ARTIFICIAL?,” 2021. <https://neuroons.com/es/que-son->

- entrenamiento-e-inferencia-en-inteligencia-artificial/#:~:text=¿Qué es el entrenamiento de,de manera veraz o falsa
- [47] WOLFRAM, “Uso de modelos preentrenados para visualizar características.” <https://www.wolfram.com/language/12/neural-network-framework/use-pre-trained-models-to-visualize-features.html.es>
- [48] Datos.gob, “Transfer Learning: cómo entrenar modelos de Deep Learning de manera asequible,” 2022. <https://datos.gob.es/es/blog/transfer-learning-como-entrenar-modelos-de-deep-learning-de-manera-asequible>
- [49] AWS, “¿Qué es una red neuronal?” <https://aws.amazon.com/es/what-is/neural-network/#:~:text=Arquitectura de una red neuronal profunda,entre un nodo y otro.>
- [50] TATIC, “La diferencia entre inteligencia artificial, aprendizaje automático y aprendizaje profundo.” <https://tatic.net/es/blog/la-diferencia-entre-inteligencia-artificial-aprendizaje-automatico-y-aprendizaje-profundo/>
- [51] NetApp, “¿Qué es el aprendizaje automático?” <https://www.netapp.com/es/artificial-intelligence/what-is-machine-learning/#:~:text=Un subconjunto de la inteligencia,de decisiones sin interacción humana>
- [52] Geeksforgeeks, “OpenCV – Overview,” Oct. 2022.
- [53] Tutorialspoint, “tutorialspoint.” https://www.tutorialspoint.com/opencv/opencv_overview.htm
- [54] A. VANNIEUWENHUYZE, “Inteligencia artificial fácil - Machine Learning y Deep Learning prácticos,” in *Inteligencia artificial fácil*, 2019.
- [55] J. M. Heras, “Precision, Recall, F1, Accuracy en clasificación,” 2020. <https://www.iartificial.net/precision-recall-f1-accuracy-en-clasificacion/>
- [56] J. M. Heras, “¿Clasificación o Regresión?,” 2020. <https://www.iartificial.net/clasificacion-o-regresion/>
- [57] L. Fernando, “¿Qué es y para qué sirve una regresión lineal en machine learning?,” 2022. <https://platzi.com/blog/que-es-regresion-lineal/>
- [58] “¿Cómo funciona una webcam?,” Jun. 01, 2006. <https://www.muyinteresante.es/curiosidades/12725.html#:~:text=Una%20webcam%20es%20una%20peque%C3%B1a,usuarios%20o%20a%20una%20p%C3%A1gina%20web> (accessed Apr. 10, 2023).

- [59] COMPONENTS 101, “Pi Camera Module – 5MP,” 2018. <https://components101.com/misc/pi-camera-module> (accessed Jun. 12, 2023).
- [60] Prabu Kumar, “What is a MIPI Camera? How does MIPI Camera Work?,” 2021. <https://www.e-consystems.com/blog/camera/technology/what-is-a-mipi-camera-how-does-mipi-camera-work/> (accessed May 31, 2023).
- [61] EL CONGRESO DE COLOMBIA, *LEY ESTATUTARIA 1581 DE 2012*. Colombia, 2012. Accessed: May 14, 2023. [Online]. Available: <https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=49981>
- [62] Ministerio de Ambiente y Desarrollo Sostenible, “Protección de Datos Personales,” 2012. <https://www.minambiente.gov.co/politica-de-proteccion-de-datos-personales/#:~:text=Ley%20de%20Protecci%C3%B3n%20de%20Datos,de%20naturaleza%20p%C3%ABlica%20o%20privada.https://www.minambiente.gov.co/politica-de-proteccion-de-datos-personales/#:~:text=Ley%20de%20Protecci%C3%B3n%20de%20Datos,de%20naturaleza%20p%C3%ABlica%20o%20privada.> (accessed May 04, 2023).
- [63] OVNIVISION, “datasheet OV56447.”
- [64] Series HBO, “Silicon Valley, en octubre la sexta y última temporada,” 2019. <https://netflixyseries.com/silicon-valley-en-octubre-la-sexta-y-ultima-temporada.html> (accessed Jun. 12, 2023).
- [65] M. J. J. PAUL VIOLA, “Robust Real-Time Face Detection,” Sep. 10, 2001. Accessed: May 04, 2023. [Online]. Available: <https://www.face-rec.org/algorithms/boosting-ensemble/16981346.pdf>
- [66] T. O. U. of I. Department of Mathematics and Computer Science, “Age and gender classification using convolutional neural networks,” 2015. <https://ieeexplore.ieee.org/document/7301352/authors#authors> (accessed Jun. 07, 2023).
- [67] University of Israel and Adience, “Unfiltered faces for gender and age classification,” 2014.
- [68] ALIA ALRAWI, “age_gender_CNN,” 2021. <https://www.kaggle.com/code/aliaalrawi/age-gender-cnn> (accessed May 01, 2023).

- [69] UTKFace, “UTKFace,” 2020. <https://www.kaggle.com/datasets/jangedoo/utkface-new> (accessed May 01, 2023).
- [70] DANE, “Panorama sociodemográfico de la juventud en Colombia,” Sep. 20AD. Accessed: May 03, 2023. [Online]. Available: <https://www.dane.gov.co/files/investigaciones/genero/informes/informe-panorama-sociodemografico-juventud-en-colombia.pdf>

9. ANEXOS

ANEXO 1: CÓDIGO FUENTE DEL ULTIMO MODELO ENTRENADO

```

""" Importación de las librerías """
import numpy as np
import pandas as pd
import os
import glob
import pandas as pd
# importing cv2
import matplotlib.pyplot as plt
#import seaborn as sns
import plotly.graph_objects as go
from tensorflow.keras.utils import to_categorical
from PIL import Image
import plotly.express as px

def plot_distribution(pd_series):
    labels = pd_series.value_counts().index.tolist()
    counts = pd_series.value_counts().values.tolist()

    pie_plot = go.Pie(labels=labels, values=counts, hole=.3)
    fig = go.Figure(data=[pie_plot])
    fig.update_layout(title_text='Distribucion de %s' % pd_series.name)

    fig.show()

""" Se guarda en una variable el DataSet """
dataset_folder_name =
'/home/consultant1/Documents/Personal/ProyectoDeGrado/UTKface_inthewild/archive/utk
face_aligned_cropped/UTKFace'

```

```
TRAIN_TEST_SPLIT = 0.7
```

```
IM_WIDTH = IM_HEIGHT = 198
```

```
""" Se genera un diccionario para genero """
```

```
""" se omitira el diccionario para raza de dataset """
```

```
dataset_dict = {
```

```
    'gender_id': {
```

```
        0: 'Masculino',
```

```
        1: 'Femenino'
```

```
    }
```

```
}
```

```
""" Con el items devuelve cada uno de los elemento en un diccionario como una lista, recorre con un
```

```
arreglo el diccionario ya dado tanto para edad como para raza """
```

```
dataset_dict['gender_alias'] = dict((g, i) for i, g in dataset_dict['gender_id'].items())
```

```
""" Se genera una función para empezar a extraer los datos del data set
```

```
retorna un pandas con la información de edad, genero y raza """
```

```
def analizar_dataset( dataset_path , ext = 'jpg' ) :
```

```
    def analizar_info_archivo(path):
```

```
        """ se realiza analisis de un solo archivo """
```

```
        try:
```

```
            """ dividie el normbre de la ruta en el primer archivo """
```

```
            archivo = os.path.split(path)[1]
```

```
            """ divide el nombre del archivo separando la extensión """
```

```
            archivo = os.path.splitext(archivo)[0]
```

```
            """ reemplaza en loss archivos el _ y crea un arreglo """
```

```
            edad, genero, raza, _ = archivo.split('_')
```

```

        return int(edad), dataset_dict['gender_id'][int(genero)]
    except Exception as ex:
        return None, None

files = glob.glob(os.path.join(dataset_path, "*.%s" % ext))

records = []
for file in files:
    info = analizar_info_archivo(file)
    #print("info",info)
    records.append(info)

df = pd.DataFrame(records)
df['archivo'] = files
df.columns = ['edad', 'genero', 'archivo']
df = df.dropna()

return df

df = analizar_dataset(dataset_folder_name)
df.head()
#print(df)

def plot_data():
    plot_distribution(df['gender'])

fig = px.histogram(df, x="age", nbins=20)
fig.update_layout(title_text='Age distribution')
fig.show()
bins = [0, 10, 20, 30, 40, 60, 80, np.inf]
names = ['<10', '10-20', '20-30', '30-40', '40-60', '60-80', '80+']

```

```
age_binned = pd.cut(df['age'], bins, labels=names)
plot_distribution(age_binned)
```

```
class UtkFaceDataGenerator():
```

```
    """
```

Data generator for the UTKFace dataset. This class should be used when training our Keras multi-output model.

```
    """
```

```
    def __init__(self, df):
```

```
        self.df = df
```

```
    def generate_split_indexes(self):
```

```
        p = np.random.permutation(len(self.df))
```

```
        #print("p: ",p)
```

```
        train_up_to = int(len(self.df) * TRAIN_TEST_SPLIT)
```

```
        #print("train_up_to: ",train_up_to)
```

```
        train_idx = p[:train_up_to]
```

```
        #print("train_idx: ",train_idx)
```

```
        test_idx = p[train_up_to:]
```

```
        #print("test_idx: ",test_idx)
```

```
        train_up_to = int(train_up_to * TRAIN_TEST_SPLIT)
```

```
        #print("train_up_to: ",train_up_to)
```

```
        train_idx, valid_idx = train_idx[:train_up_to], train_idx[train_up_to:]
```

```
        #print("train_idx: ",train_idx)
```

```
        #print("valid_idx: ",valid_idx)
```

```
        # converts alias to id
```

```
        self.df['gender_id'] = self.df['genero'].map(lambda gender:
dataset_dict['gender_alias'][gender])
```

```

self.max_age = self.df['edad'].max()
print("EDAD maxima -----")
print(self.df['edad'].max())
#print("Final")
#print("train_idx: ",train_idx)
#print("valid_idx: ",valid_idx)
#print("test_idx: ",test_idx)

```

```

return train_idx, valid_idx, test_idx

```

```

def preprocess_image(self, img_path):

```

```

    """

```

Se procesa la imagen para que quede de la forma [192,192,3] ya que esta es la entrada de la red

```

    """

```

```

    im = Image.open(img_path)
    im = im.resize((IM_WIDTH, IM_HEIGHT))
    im = np.array(im) / 255.0

```

```

    return im

```

```

def generate_images(self, image_idx, is_training, batch_size=16):

```

```

    """

```

Used to generate a batch with images when training/testing/validating our Keras model.

```

    """

```

```

    # arrays to store our batched data

```

```

    images, ages, genders = [], [], []

```

```

    while True:

```

```

        for idx in image_idx:

```

```

            person = self.df.iloc[idx]

```

```

age = person['edad']
gender = person['gender_id']
file = person['archivo']

im = self.preprocess_image(file)
ages.append(age / self.max_age)
if age / self.max_age > 1:
    print("-----age / self.max_age:",age / self.max_age)
genders.append(to_categorical(gender, len(dataset_dict['gender_id'])))
if len(im.shape)>2:
    images.append(im)

# yielding condition

if len(images) >= batch_size:
    #print(np.shape(images))
    yield np.array(images), [np.array(ages), np.array(genders)]
    images, ages, genders = [], [], []

if not is_training:
    break

data_generator = UtkFaceDataGenerator(df)
print("Data generator ",data_generator)
train_idx, valid_idx, test_idx = data_generator.generate_split_indexes()

from tensorflow.keras.models import Model
from tensorflow.keras.layers import BatchNormalization
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.layers import MaxPooling2D

```

```

from tensorflow.keras.layers import Activation
from tensorflow.keras.layers import Dropout
from tensorflow.keras.layers import Lambda
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Flatten
from tensorflow.keras.layers import Input
import tensorflow as tf

```

```

class UtkMultiOutputModel():

```

```

    """

```

Used to generate our multi-output model. This CNN contains two branches, one for age and other for

sex . Each branch contains a sequence of Convolutional Layers that is defined on the make_default_hidden_layers method.

```

    """

```

```

    def make_default_hidden_layers(self, inputs):

```

```

        """

```

Used to generate a default set of hidden layers. The structure used in this network is defined as:

Conv2D -> BatchNormalization -> Pooling -> Dropout

```

        """

```

```

        x = Conv2D(16, (3, 3), padding="same")(inputs)

```

```

        x = Activation("relu")(x)

```

```

        x = BatchNormalization(axis=-1)(x)

```

```

        x = MaxPooling2D(pool_size=(3, 3))(x)

```

```

        x = Dropout(0.25)(x)

```

```

        x = Conv2D(32, (3, 3), padding="same")(x)

```

```

        x = Activation("relu")(x)

```

```

        x = BatchNormalization(axis=-1)(x)

```

```
x = MaxPooling2D(pool_size=(2, 2))(x)
x = Dropout(0.25)(x)
```

```
x = Conv2D(64, (3, 3), padding="same")(x)
x = Conv2D(64, (3, 3), padding="same")(x)
x = Activation("relu")(x)
x = BatchNormalization(axis=-1)(x)
x = MaxPooling2D(pool_size=(2, 2))(x)
x = Dropout(0.25)(x)
```

```
x = Conv2D(128, (3, 3), padding="same")(x)
x = Conv2D(128, (3, 3), padding="same")(x)
x = Activation("relu")(x)
x = BatchNormalization(axis=-1)(x)
x = MaxPooling2D(pool_size=(2, 2))(x)
x = Dropout(0.25)(x)
#print("---x",x)
return x
```

```
def build_gender_branch(self, inputs, num_genders=2):
```

```
    """
```

```
    Used to build the gender branch of our face recognition network.
```

```
    This branch is composed of three Conv -> BN -> Pool -> Dropout blocks,
    followed by the Dense output layer.
```

```
    """
```

```
x = Lambda(lambda c: tf.image.rgb_to_grayscale(c))(inputs)
```

```
x = self.make_default_hidden_layers(inputs)
```

```
x = Flatten()(x)
```

```
x = Dense(128)(x)
```



```

x = Activation("relu")(x)
x = BatchNormalization()(x)
x = Dropout(0.5)(x)
x = Dense(num_genders)(x)
x = Activation("sigmoid", name="gender_output")(x)

return x

def build_age_branch(self, inputs):
    """
    Used to build the age branch of our face recognition network.
    This branch is composed of three Conv -> BN -> Pool -> Dropout blocks,
    followed by the Dense output layer.

    """
    x = self.make_default_hidden_layers(inputs)

    x = Flatten()(x)
    x = Dense(128)(x)
    x = Activation("relu")(x)
    x = BatchNormalization()(x)
    x = Dropout(0.5)(x)
    x = Dense(1)(x)
    x = Activation("linear", name="age_output")(x)

    return x

def assemble_full_model(self, width, height, ):
    """
    Used to assemble our multi-output model CNN.

    """

```

```

input_shape = (height, width, 3)
#print("--input_shape",input_shape)

inputs = Input(shape=input_shape)
#print("--intpus",inputs)

age_branch = self.build_age_branch(inputs)
gender_branch = self.build_gender_branch(inputs)

model = Model(inputs=inputs,outputs = [age_branch,
gender_branch],name="face_net")
#print("-----Model ",model)
return model

model = UtkMultiOutputModel().assemble_full_model(IM_WIDTH, IM_HEIGHT)

from tensorflow.keras.optimizers import Adam

init_lr = 1e-4
epochs = 2000

opt = Adam(learning_rate=init_lr, decay=init_lr / epochs)

model.compile(optimizer=opt,
              loss={
                  'age_output': 'mse',
                  'gender_output': 'binary_crossentropy'},
              loss_weights={
                  'age_output': 4.,
                  'gender_output': 0.1},

```

```

        metrics={
            'age_output': 'mae',
            'gender_output': 'accuracy'},run_eagerly=True)

from tensorflow.keras.callbacks import ModelCheckpoint

batch_size = 32
valid_batch_size = 32
train_gen      =      data_generator.generate_images(train_idx,      is_training=True,
batch_size=batch_size)
valid_gen      =      data_generator.generate_images(valid_idx,      is_training=True,
batch_size=valid_batch_size)
#print("-----train_gen",train_gen )
#print("-----valid_gen",valid_gen)

callbacks = [
    ModelCheckpoint("./model_checkpoint", monitor='val_loss')
]

history = model.fit(train_gen,
                    epochs=epochs,
                    steps_per_epoch= 10,
                    callbacks=callbacks,
                    validation_data=valid_gen,
                    validation_steps=len(valid_idx)//valid_batch_size)

#Presicion de genero
plt.clf()
fig = go.Figure()

```

```
fig.add_trace(go.Scatter(  
    y=history.history['gender_output_accuracy'],  
    name='Train'))  
fig.add_trace(go.Scatter(  
    y=history.history['val_gender_output_accuracy'],  
    name='Valid'))  
fig.update_layout(height=500,  
    width=700,  
    title='Accuracy for gender feature',  
    xaxis_title='Epoch',  
    yaxis_title='Accuracy')  
fig.show()
```

#Erro absoluto de edad

```
plt.clf()
```

```
fig = go.Figure()  
fig.add_trace(go.Scattergl(  
    y=history.history['age_output_mae'],  
    name='Train'))  
fig.add_trace(go.Scattergl(  
    y=history.history['val_age_output_mae'],  
    name='Valid'))  
fig.update_layout(height=500,  
    width=700,  
    title='Mean Absolute Error for age feature',  
    xaxis_title='Epoch',  
    yaxis_title='Mean Absolute Error')  
fig.show()
```

```
fig = go.Figure()
```

```

fig.add_trace(go.Scattergl(
    y=history.history['loss'],
    name='Train'))
fig.add_trace(go.Scattergl(
    y=history.history['val_loss'],
    name='Valid'))
fig.update_layout(height=500,
    width=700,
    title='Overall loss',
    xaxis_title='Epoch',
    yaxis_title='Loss')
fig.show()

model.summary()
model.save('model6.h5')

from keras.utils import plot_model
import matplotlib.image as mpimg
import matplotlib.pyplot as plt

plot_model(model, to_file='model6.png')
img = mpimg.imread('model6.png')

plt.figure(figsize=(40, 30))
plt.imshow(img)

test_batch_size = 128
test_generator = data_generator.generate_images(test_idx, is_training=False,
batch_size=test_batch_size)
age_pred, gender_pred = model.predict(test_generator,
                                     steps=len(test_idx)//test_batch_size,

```

```
verbose= 2)
```

```
test_generator      =      data_generator.generate_images(test_idx,      is_training=False,
batch_size=test_batch_size)
samples = 0
images, age_true, gender_true = [], [], []
for test_batch in test_generator:
    image = test_batch[0]
    labels = test_batch[1]

    images.extend(image)
    age_true.extend(labels[0])
    gender_true.extend(labels[1])

age_true = np.array(age_true)
gender_true = np.array(gender_true)
gender_true = np.argmax(gender_true,axis=1)
gender_pred = np.argmax(gender_pred,axis=1)

age_true = age_true * data_generator.max_age
age_pred = age_pred * data_generator.max_age
#if age_pred > data_generator.max_age:

from sklearn.metrics import classification_report

cr_gender      =      classification_report(gender_true,      gender_pred,
target_names=dataset_dict['gender_alias'].keys())
print(cr_gender)
```

```
from sklearn.metrics import r2_score

print('R2 score for age: ', r2_score(age_true, age_pred))
```

ANEXO 2: VIDEO PRUEBA PROTOTIPO 1

<https://drive.google.com/file/d/1aossCqPiBeStxjYRvgylv5AF4z8Jq1nk/view?usp=sharing>