



**UNIVERSIDAD NACIONAL DE CHIMBORAZO
FACULTAD DE INGENIERÍA
CARRERA DE ELECTRÓNICA Y TELECOMUNICACIONES**

**Desarrollo de un sistema de detección de personas usando
inteligencia artificial para controlar el distanciamiento social
en la zona comercial de Pelileo**

**Trabajo de Titulación para optar al título de
Ingeniero en Electrónica y Telecomunicaciones**

Autor:

Silva Chipantiza, Christian Fabricio

Tutor:

PhD. Daniel Antonio Santillán Haro

Riobamba, Ecuador. 2023

DERECHOS DE AUTORÍA

Yo, **Christian Fabricio Silva Chipantiza**, con cédula de ciudadanía **1803878261**, autor del trabajo de investigación titulado:” **Desarrollo de un Sistema de Detección de Personas Usando Inteligencia Artificial para Controlar el Distanciamiento Social en la Zona comercial de Pelileo**, certifico que la producción, ideas, opiniones, criterios, contenidos y conclusiones expuestas son de mí exclusiva responsabilidad.

Asimismo, cedo a la Universidad Nacional de Chimborazo, en forma no exclusiva, los derechos para su uso, comunicación pública, distribución, divulgación y/o reproducción total o parcial, por medio físico o digital; en esta cesión se entiende que el cesionario no podrá obtener beneficios económicos. La posible reclamación de terceros respecto de los derechos de autor (a) de la obra referida, será de mi entera responsabilidad; librando a la Universidad Nacional de Chimborazo de posibles obligaciones.

En Riobamba, 13 de abril 2023.



Christian Fabricio Silva Chipantiza

C.I: 1803878261

DICTAMEN FAVORABLE DEL TUTOR Y MIEMBROS DE TRIBUNAL,

Quienes suscribimos, catedráticos designados Tutor y Miembros del Tribunal de Grado para la evaluación del trabajo de investigación “**Desarrollo de un Sistema de Detección de Personas Usando Inteligencia Artificial para Controlar el Distanciamiento Social en la Zona comercial de Pelileo**”, presentado por **Christian Fabricio Silva Chipantiza**, con cédula de identidad número 1803878261, certificamos que recomendamos la APROBACIÓN de este con fines de titulación. Previamente se ha asesorado durante el desarrollo, revisado y evaluado el trabajo de investigación escrito y escuchada la sustentación por parte de su autor; no teniendo más nada que observar.

De conformidad a la normativa aplicable firmamos, en **Riobamba 13 de abril de 2023**

Carlos Ramiro Peñafiel Ojeda, Dr.
PRESIDENTE DEL TRIBUNAL DE GRADO



Leonardo Fabian Rentería Bustamante, PhD.
MIEMBRO DEL TRIBUNAL DE GRADO



Manuel Antonio Meneses Freire, Dr.
MIEMBRO DEL TRIBUNAL DE GRADO



Daniel Antonio Santillán Haro, PhD
TUTOR





CERTIFICACIÓN

Que, **CHRISTIAN FABRICIO SILVA CHIPANTIZA** con CC: **1803878261**, estudiante de la Carrera **ELECTRONICA Y TELECOMUNICACIONES, NO VIGENTE**, Facultad de **INGENIERIA**; ha trabajado bajo mi tutoría el trabajo de investigación titulado " **DESARROLLO DE UN SISTEMA DE DETECCIÓN DE PERSONAS USANDO INTELIGENCIA ARTIFICIAL PARA CONTROLAR EL DISTANCIAMIENTO SOCIAL EN LA ZONA COMERCIAL DE PELILEO**", cumple con el 1%, de acuerdo al reporte del sistema Anti plagio **URKUND**, porcentaje aceptado de acuerdo a la reglamentación institucional, por consiguiente autorizo continuar con el proceso.

Riobamba, 21 de marzo de 2023



Firmado electrónicamente por:
**DANIEL ANTONIO
SANTILLAN HARO**

PhD. Daniel Santillán
TUTOR TRABAJO DE INVESTIGACIÓN

DEDICATORIA

La vida nos enseña que nunca debemos dejar de luchar y ser constantes, con el pasar del tiempo llega la tan anhelada recompensa.

El trabajo de grado está dedicado a Dios por darme salud y fuerza para finalizar con mi formación académica y ser un profesional.

A mi querido padre Luis Silva que me ha dado tanto sus consejos, la capacidad de superarme y desear lo mejor en cada paso por este camino difícil y arduo de la vida, gracias por ser como es papa, porque con su apoyo incondicional he construido y forjado la persona que soy hoy.

A mi hermano Marcelo quien es una parte fundamental en mi vida, no hay palabras para agradecerle todo lo que ha hecho por mí, tu apoyo nunca dejo de brillar siempre confiaste en mí.

A mis familiares quienes de la misma manera de igual forma colaboraron con un grano de arena, hoy es el día en que ustedes son testigos de la finalización de una etapa de mi vida.

Christian Fabricio Silva Chipantiza

AGRADECIMIENTO

Agradezco a mi madre Esther Chipantiza que me bendice desde el cielo, a mi Padre, Luis Silva por tener siempre su constante apoyo por medio de su trabajo, esfuerzo y poder llegar a este momento de éxito, mediante su confianza que deposito en mi para poder concluir mi estudio universitario.

Agradezco el apoyo moral y fraterno de mi hermano y su familia quienes fueron un motivo, que me impulsaban día tras día para poder culminar y no decaer con la formación académica mil gracias

A mi tutor el Dr. Daniel Santillán, por brindar su tiempo y conocimiento para poder realizar este proyecto de grado, quien estuvo en el proceso y culminación del mismo.

A la Universidad Nacional de Chimborazo, la cual fue la entidad donde forme mis ideales académicos, a mis revisores quienes de la misma manera me guiaron con su apoyo mediante su conocimiento y gracias a ellos puedo hoy concluir con mi preparación de tercer nivel.

Christian Fabricio Silva Chipantiza

ÍNDICE GENERAL

DERECHOS DE AUTORÍA	2
DICTAMEN FAVORABLE DEL TUTOR Y MIEMBROS DE TRIBUNAL	3
CERTIFICACIÓN	4
DEDICATORIA	5
AGRADECIMIENTO	6
ÍNDICE GENERAL	7
ÍNDICE TABLAS	9
ÍNDICE FIGURAS	10
RESUMEN	12
ABSTRACT	13
Capítulo 1.....	15
Introducción	15
1.1. Antecedentes.....	15
1.2. Planteamiento del problema.	16
1.3. Justificación.....	17
1.4. Objetivos	18
1.4.1. Objetivo General.....	18
1.4.2. Objetivos Específicos.	18
Capítulo 2.....	19
Fundamento teórico.	19
2.1. Estado del arte.....	19
2.2. Marco Teórico.....	20
2.2.1. Procesamiento digital de imágenes.	20
2.2.3. Técnicas de inteligencia artificial	23
2.2.4. Algoritmos para detección de personas	28
Capítulo 3.....	33
Metodología.....	33
3.1. Tipo de investigación.	33
3.2. Métodos de la investigación.....	33
3.2.1. Técnicas de recolección de datos primarios y secundarios.	33
3.3. Escenario de prueba.	34
3.4. Hipótesis y operacionalización de variables.	37
3.4.1. Hipótesis de investigación H.....	37
3.5. Sistema de adquisición de video	39
3.6. Diseño del algoritmo para detección de personas	42
3.7. Cálculo de distancia entre personas.....	52
3.8. Lógica de Programación del Sistema.....	56
3.8.1. Diagrama de Flujo Detección de Personas.....	57

3.8.2.	Diagrama de Flujo Medición Distancia	58
3.8.3.	Diagrama de Flujo Modo Offline	59
3.8.4.	Diagrama de Flujo Modo Online	60
3.9.	Interfaz de usuario para monitoreo de aglomeración	61
Capítulo 4.....		75
Resultados y discusión.		75
4.1.	Resultados Fiabilidad.	75
4.1.1.	Escenario de Prueba 1.....	76
4.1.2.	Escenario de Prueba	77
4.2.	Comprobación de Hipótesis.	78
4.2.2.	Escenario de Prueba 2.....	81
4.3.1.	Escenario de Prueba 1.....	83
4.3.2.	Escenario de Prueba 2.....	84
4.4.	Resultados en Campo.	86
Capítulo 5.....		90
Conclusiones y Recomendaciones		90
5.1.	Conclusiones.....	90
5.2.	Recomendaciones.....	91
Bibliografía		92
Anexos.....		96
A.1.	Datos generados por el Sistema	96
A.2.	Código del Programa	99
A.3.	Código de Pruebas Preliminares	111
A.4.	Capturas del Sistema en funcionamiento	112

ÍNDICE TABLAS

Tabla 1 Comparativo de algoritmos para detección de personas basado en MATLAB.....	32
Tabla 2 Variables del proyecto de investigación.....	38
Tabla 3 Características Técnicas de la cámara IP PTZ HOMKEEP	41
Tabla 4 Tabla comparativa de imágenes de 400x250px.....	46
Tabla 5 Tabla comparativa de imágenes de 800x520px.....	48
Tabla 6 Tabla comparativa de imágenes de 1080x720px.....	50
Tabla 7 Datos obtenidos en las pruebas de medición a 0.5, 1, 1.5 y 2 metros	54
Tabla 8. Prueba de estimación del algoritmo de píxeles a metros.....	55
Tabla 9. Datos Estadísticos del Error Obtenido en el Escenario de Prueba 1.	76
Tabla 10. Datos Estadísticos del Error Obtenido en el Escenario de Prueba 2.	77
Tabla 11. Datos Estadísticos de las pruebas realizadas para el Escenario de Prueba 1.....	79
Tabla 12. ANOVA para el Escenario de Prueba 1.	81
Tabla 13. ANOVA para el Escenario de Prueba 2.	82
Tabla 14. Análisis de grupos post hoc para el Escenario 1	83
Tabla 15. Análisis de grupos post hoc para el Escenario 2	84

ÍNDICE FIGURAS

Figura 1. Representación de Colores RGB en escala tridimensional [1]	22
Figura 2. Mapa de Color de una Imagen [2].....	22
Figura 3. Niveles de procesamiento de una imagen [2]	24
Figura 4. Niveles de procesamiento de una imagen [3]	25
Figura 5. Modelo de una red neuronal perceptrón multicapa [4]	27
Figura 6. Canales del Algoritmo ACF para detección de personas [5]	30
Figura 7. Ejemplo de filtro Gabor [6].....	30
Figura 8. Hiperplano para representación de SVM [7]	31
Figura 9. Esquema del Sistema de Detección de Distanciamiento Social.....	34
Figura 10. Distancia de la Avenida Confraternidad de Pelileo El Tambo.	35
Figura 11. Esquema del sistema de adquisición de video.	39
Figura 12. Cámara IP SMART ONLY	41
Figura 13. Cámara Seleccionada para la Obtención de Datos de Prueba.....	42
Figura 14. Ubicación Real de la Cámara de Pruebas	42
Figura 15. Diagrama de Bloques de comparativa y selección de algoritmos.....	43
Figura 16. Diagrama de Bloques de comparativa y selección de algoritmos.....	44
Figura 17. Imágenes de prueba de una resolución 800x533 px.....	45
Figura 18. Imágenes de prueba de una resolución 1080x720 px.....	45
Figura 19. Diagrama de Caja Falsos Positivos en Imágenes de 400px.	46
Figura 20. Diagrama de Caja Falsos Negativos en Imágenes de 400px.....	47
Figura 21. Diagrama de Caja % Detección en Imágenes de 400px.	47
Figura 22. Diagrama de Caja Falsos Positivos en Imágenes de 800px.	48
Figura 23. Diagrama de Caja Falsos Negativos en Imágenes de 800px.....	49
Figura 24. Diagrama de Caja %Detección en Imágenes de 800px.	49
Figura 25. Diagrama de Caja Falsos Positivos en Imágenes de 1080px.	50
Figura 26. Diagrama de Caja Falsos Negativos en Imágenes de 1080px.....	51
Figura 27. Diagrama de Caja %Detección en Imágenes de 1080px.	51
Figura 28. Calculo de la distancia entre objetivos en pixeles.....	53
Figura 29. Pruebas de algoritmo a distancia de 0.5, 1, 1.5, 2 metros.	53
Figura 30. Diagrama de Flujo Principal del Sistema.....	56
Figura 31. Diagrama de Flujo del Algoritmo de Detección de Personas.	57

Figura 32. Diagrama de Flujo del Algoritmo para la Medición de Distancia.	58
Figura 33. Diagrama de Flujo Modo Offline.....	59
Figura 34. Diagrama de Flujo Modo Online.	60
Figura 35. Componentes de Pestaña OFFLINE del sistema	61
Figura 36. Selección de Archivo de video Offline	62
Figura 37. Información del video e inicio de reproducción	63
Figura 38. Inicio de reproducción de video offline	64
Figura 39. Acumulación de Aglomeraciones para Precaución.....	65
Figura 40. Sistema detectando Peligro de Aglomeraciones modo Offline.....	66
Figura 41. Generación de Archivo txt en Modo Offline	66
Figura 42. Componentes de la pestaña del modo ONLINE del sistema	68
Figura 43. Establecer Conexión con Cámara IP	68
Figura 44. Sistema de Detección de distanciamiento en tiempo real inicializado	69
Figura 45. Precaución de Aglomeraciones en modo Online	70
Figura 46. Inicio de Grabación de Video modo Online	71
Figura 47. Detención y Generación de archivo de Video Grabado modo Online.....	72
Figura 48. Peligro de Aglomeraciones en modo Online	73
Figura 49. Generación de archivo txt en modo Online.....	74
Figura 50. Ubicación de las Cámaras a lo largo de la Zona Comercial de el Tambo.	75
Figura 51. Diagrama de Caja del Error en el Escenario de Prueba 1.	77
Figura 52. Diagrama de Caja del Error en el Escenario de Prueba 2.	78
Figura 53. Diagrama de Caja de los Datos del Escenario de Prueba 1.....	80
Figura 54. Diagrama de Caja de los Datos del Escenario de Prueba 2.....	82
Figura 55. Agrupación de medias.....	84
Figura 56. Agrupación de medias.....	85
Figura 57. Prueba 1 de Sistema Desplegado.	86
Figura 58. Prueba 2 de Sistema Desplegado.	87
Figura 59. Prueba 3 de Sistema Desplegado.	87
Figura 60. Prueba 4 de Sistema Desplegado.	88
Figura 61. Prueba 5 de Sistema Desplegado.	89

RESUMEN

El presente proyecto tiene como finalidad diseñar un sistema a bajo costo que permita la detección de personas con el uso de algoritmos de inteligencia artificial y medir la distancia entre estos objetos de estudio. Esto se desplegará en Pelileo específicamente en la zona comercial del Tambo para proporcionar un método de prevención de aglomeraciones.

El segundo capítulo contiene una descripción detallada de los componentes que conforman una imagen digitalizada, como el píxel, espacio RGB, mapa de color, etc.; así como los métodos de procesamiento de imágenes y algoritmos de inteligencia artificial que permiten detectar objetos, en este caso personas.

En el tercer capítulo se presenta la descripción de los diferentes componentes involucrados en el diseño e implementación del sistema. En primer lugar, se analizan las especificaciones mínimas para la selección de la cámara que se ajusta a los requerimientos del sistema. Se realiza un análisis de los algoritmos de inteligencia artificial que provee Matlab con el fin de seleccionar el más adecuado para el sistema, siendo ACF CALTECH el que mejores resultados presenta.

Se implementó el cálculo de la estimación de la distancia entre objetos utilizando patrones de prueba con distancias fijas para poder obtener datos y calcular la ecuación que describe dicha estimación. Se procede a dar una explicación gráfica mediante diagramas de flujo de la lógica de programación implementada y finalmente se presenta la interfaz gráfica diseñada con el funcionamiento y los diferentes modos que pueden usarse.

Por último, se exponen los resultados obtenidos en la evaluación de fidelidad del sistema, en donde se realizaron diferentes pruebas a distancias fijas de cámara y se realizó el cálculo de ANOVA para determinar si la distancia estimada difiere cuando existe un cambio de posición del objeto con respecto a la cámara. Se realizaron pruebas en donde se midió el error del sistema bajo diferentes distancias fijas obteniendo un porcentaje de error menor del 12 %. Finalmente, se desplegó el sistema en la zona comercial de Pelileo - El Tambo y se comprobó el funcionamiento del sistema dando resultados muy satisfactorios en casi todas las condiciones, teniendo un reconocimiento de personas cercano al 100 % de efectividad.

Palabras clave: COVID-19, aprendizaje profundo, imágenes, videos, inteligencia artificial, distanciamiento, píxeles.

ABSTRACT

This project aims to design a low-cost system that allows the detection of people with the use of artificial intelligence algorithms and measure the distance between these objects of study. This will be deployed in Pelileo specifically in the commercial area of Tambo to provide a method of crowd prevention.

The second chapter contains a detailed description of the components that make up a digitised image, such as pixel, RGB space, colour map, etc.; as well as the image processing methods and artificial intelligence algorithms that allow the detection of objects, in this case people.

The third chapter presents the description of the different components involved in the design and implementation of the system. First, the minimum specifications for the selection of the camera that meets the requirements of the system are analysed. An analysis of the artificial intelligence algorithms provided by Matlab is carried out in order to select the most suitable algorithm for the system, being ACF CALTECH the one with the best results.

The calculation of the estimation of the distance between objects was implemented using test patterns with fixed distances in order to obtain data and calculate the equation that describes this estimation. A graphical explanation of the programming logic implemented is given by means of flow diagrams and finally, the graphical interface designed with the operation and the different modes that can be used is presented.

Finally, the results obtained in the evaluation of the fidelity of the system are presented, where different tests were carried out at fixed camera distances and the ANOVA calculation was performed to determine whether the estimated distance differs when there is a change in the position of the object with respect to the camera. Tests were carried out to measure the error of the system under different fixed distances, obtaining an error rate of less than 12 %. Finally, the system was deployed in the commercial area of Pelileo - El Tambo and the operation of the system was checked, giving very satisfactory results in almost all conditions, having a recognition of people close to 100 % of effectiveness.

Keywords: COVID-19, deep learning, images, videos, artificial intelligence, distance, pixels.



firmado electrónicamente por:
SANDRA LILLIANA
ABARCA GARCIA

Reviewed by:

Lic. Sandra Abarca Mgs.

ENGLISH PROFESSOR

C.C. 0601921505

Capítulo 1

Introducción

1.1. Antecedentes

En la actualidad, los dispositivos móviles ya cuentan con cámaras de video en alta definición, que combinados con los procesadores de última generación ejecutan algoritmos basados en tiempo real, para mejorar el control y la interacción con aplicaciones móviles, en el trabajo realizado por [8], se menciona la interacción de los gestos y reconocimiento de los mismos para el manejo de un dron basándose en algoritmos de inteligencia artificial.

Un ejemplo de esto es el reconocimiento de gestos que incorporan los teléfonos inteligentes, emitiendo mensajes en la pantalla cuando detecta una persona con ojos cerrados o cuando en la imagen se detectan movimientos que distorsionan el cuadro [9].

El rendimiento y la eficiencia energética en los dispositivos móviles ha sido un tema que las grandes empresas de tecnología han logrado optimizar, de manera que en la actualidad los Smartphones ya integran aplicaciones móviles con mayor procesamiento y menor consumo de batería [9].

Para el caso del reconocimiento de personas, en los estudios realizados por [10], se determina que los sistemas de Inteligencia Artificial (IA), deben ser implementados en un computador distinto del microcontrolador encargado del control de vuelo, haciendo que el computador externo se encargue de ejecutar todos los algoritmos y procesamiento necesarios para la identificación de personas. Para el caso de este proyecto, al obtener las imágenes por medio de una cámara ip se requiere del mismo computador para obtener las capturas, y hacer el procesamiento usando las técnicas de Inteligencia Artificial.

En el estudio realizado por [8], se determina que para la identificación de gestos existen dos técnicas, la primera y más conocida es por medio de una máquina de vectores de soporte (SVM), mientras que el segundo es por medio del uso del algoritmo de esqueletización, y la red neuronal convolucional. En esta investigación se realizará un estudio teórico de las dos técnicas para determinar la que mejor se adapta al entorno de pruebas que se va a realizar.

Este proyecto de investigación nace como una propuesta para controlar el distanciamiento social usando herramientas y técnicas aprendidas durante los estudios en la carrera, ya que como se ha determinado por [11], una de las recomendaciones para evitar contagiarse del Covid-19 es mantener por lo menos 1m de distancia entre personas.

1.2. Planteamiento del problema.

El distanciamiento social consiste en mantener limitación en la interacción entre las personas, evitando el contacto físico, así como conservando cierta distancia. Esta medida se la implementó a partir del 2020 en todo el mundo a raíz del COVID 19. Sin embargo, actualmente existe un relajamiento en la población debido a la inmunización generada por las vacunas que actualmente se aplican [12].

El relajamiento de la sociedad, así como el afán por reactivar la economía ha provocado un aumento en la cantidad de contagios de la enfermedad, por esta razón las autoridades gubernamentales requieren un mayor control de las medidas de seguridad para contrarrestar las nuevas olas de contagio que están surgiendo.

El sistema de vigilancia ECU911 consta de una serie de cámaras que permiten visualizar en varios puntos del país acontecimientos relevantes que puedan perjudicar a la población, sin embargo, el monitoreo de las cámaras tiene que ser realizado por un operador el cual se encarga de alertar cualquier situación fuera de lo normal [13].

Este sistema puede permitir controlar de una mejor forma el distanciamiento de las personas, sin embargo, no existe ninguna herramienta que utilice inteligencia artificial para detectar y controlar el distanciamiento social. Existen algunos tipos de cámaras las cuales poseen la tecnología que permite el reconocimiento facial, así como la detección de personas, pero su alto costo únicamente ha permitido tenerlas en pocos puntos de la ciudad Capital.

En varios trabajos de investigación desarrollados en los últimos años, se ha demostrado que los detectores de personas [14], pueden localizar peatones en calles con ambientes complejos, sin embargo, es alto el número de falsos positivos debido a los obstáculos, cantidad de personas juntas o incluso por el movimiento de las mismas.

La necesidad de recuperar o mejorar la economía de las familias ecuatorianas, ha hecho que el COE Nacional ya no imponga medidas restrictivas de aforo en espacios abiertos a menos que se realicen actividades masivas en estadios en las que se debe controlar el 60 por ciento mínimo [15]. Por lo antes expuesto, las personas han vuelto a salir a las calles, tanto por motivos de trabajo o por distracción, volviendo a tener la presencia elevada de personas en la zona comercial de Pelileo (Tambo) [16].

De lo expuesto anteriormente, se plantea como alternativa al control o conteo de personas manual, realizar un sistema con herramientas tecnológicas de bajo costo que permita detectar personas y calcular la distancia entre ellas, de manera que se pueda generar una alerta informativa que se activaría en caso de que los transeúntes que recorran la zona se encuentren

a menos de un metro de distancia, la cual es la mínima requerida para evitar contagios mediante el uso de mascarilla [11].

1.3. Justificación

En la actualidad existen algunos sistemas o cámaras de video que reconocen rostros o cuerpos enteros de personas, la mayoría de estos son desarrollados a nivel industrial con costos elevados debido a los materiales, herramientas y los servicios posteriores que ofrecen [17]. El desarrollo de este proyecto pretende dar a conocer, que es posible implementar un sistema de detección y distanciamiento de personas para disminuir el contagio de Covid-19 empleando elementos de bajo costo y fáciles de conseguir, con la finalidad de que sirva como soporte al control de personas que actualmente se lo realiza de manera visual por medio de las cámaras del ECU911 en esta zona de estudio [13].

Pelileo produce el 70 por ciento de jeans en el país, y en el barrio el Tambo es donde se concentra la mayor cantidad de negocios que atraen a comerciantes o público en general para la compra y venta de sus productos [16], razón por la cual se ha tomado como estudio esta zona, debido a la cantidad de personas que transitan, principalmente los fines de semana.

El objetivo principal de este proyecto plantea el diseño e implementación de un sistema que permita controlar el distanciamiento social en una zona comercial del cantón Pelileo, utilizando técnicas o algoritmos de inteligencia artificial. Para ello se diseñará el sistema de comunicación que permita transmitir las imágenes y videos de una cámara ip conectada a la red de un computador, se procesará las imágenes de manera que se desarrolle y seleccione el mejor algoritmo para la detección y reconocimiento de personas, se realizará pruebas para entrenar al algoritmo de manera que se pueda determinar la distancia entre dos personas (entre 1m y 2m) y finalmente, se evaluara el prototipo en la zona comercial de Pelileo donde existe la mayor aglomeración de personas, para realizar un análisis de los resultados obtenidos en cuanto al porcentaje de aciertos de detección de personas y la distancia mínima.

1.4. Objetivos

1.4.1. Objetivo General.

Desarrollar un Sistema de detección de personas mediante el uso de algoritmos de inteligencia artificial, para controlar el distanciamiento social en la zona comercial del cantón Pelileo.

1.4.2. Objetivos Específicos.

- Realizar un estudio sobre los tipos de algoritmos de inteligencia artificial para el reconocimiento de objetos o personas.
- Implementar el sistema de adquisición que permita transmitir y recibir las imágenes y videos de una cámara ip conectada la red de un computador.
- Seleccionar el algoritmo para la detección y reconocimiento de personas en el sistema basado en inteligencia artificial.
- Evaluar el sistema en la zona comercial de Pelileo para analizar los resultados obtenidos respecto a los aciertos en la detección de personas y la distancia mínima.

Capítulo 2

Fundamento teórico.

2.1. Estado del arte

Existen diversos estudios en los que se han desarrollado prototipos o sistemas que logran el objetivo de reconocer e identificar personas [18]. El trabajo realizado por [19], se enfoca en la detección de peatones a partir de imágenes estáticas, es decir, no clasifican a los peatones por el movimiento que estos realizan. Este sistema tiene como aplicación principal la detección de peatones para vehículos autónomos. De la misma manera, en el trabajo realizado por [20], se presenta un estudio a gran escala desde un vehículo en movimiento el cual permite la identificación de peatones, así como la estimación de la trayectoria y evaluación de riesgos para el conductor. Los sistemas de conducción autónoma se han desarrollado para evitar accidentes con los peatones [21]. Otro sistema es el realizado por [22], en donde presentan un sistema de bajo costo que utiliza únicamente una cámara óptica. En este sistema se diseñó un clasificador en cascada para conseguir una detección eficaz y fiable de los peatones. A través de la integración de información espacial y temporal capturada por un sistema de visión. Se pueden realizar sistemas confiables con el fin de prevenir accidentes en los cruces viales [23].

Existen diversos métodos utilizados para la detección de personas, entre ellos se encuentra la Red Neuronal Convolutiva (CNN) [24]. En el estudio realizado por [25] y [26], se logra una alta precisión al optimizar automáticamente la representación de características para la tarea de detección, y así regularizar la red neuronal. Las Wavelets al tener propiedades invariantes y gran eficiencia computacional son una herramienta eficaz para la detección de objetos y muy utilizadas para este tipo de aplicaciones [27].

Los algoritmos de clasificación SVM pueden ser entrenados con procesados de imagen como los “Histograma of Oriented Gradients” o “Local Binary Patterns”, los cuales permiten extraer las características de la imagen [28]. También existen algoritmos como You Only Look Once (YOLO) [29], Viola-Jones [30] o Mean Shift, los cuales generaron buenos resultados en un análisis cualitativo realizado por [31]. Estos algoritmos pueden ser implementados en varias plataformas computacionales como ChangeDetection.net (CDNET) y People Detection Benchmark repository (PDbm), que forman parte del grupo de investigación Video Processing Understanding Lab (VPU Lab) [32]. Otro ejemplo de ello es el sistema realizado por [33] en el cual utiliza una tarjeta Raspberry PI para implementar

un detector YOLO.

Los dispositivos como el KINET son herramientas que permiten realizar aplicaciones de detección, este puede generar un perfil de color en el espacio HSV en función del color de su ropa. El algoritmo realizado por [34] permite ubicar y seguir a una persona dentro de una tarea específica de trabajo. Este tipo de aplicaciones son muy utilizadas en la robótica. En el trabajo de [35] se utilizan técnicas de inteligencia artificial para la detección de personas en un robot manipulador de 6 grados de libertad, el cual clasifica las imágenes obtenidas y diferenciando a las personas de los objetos inertes.

Las investigaciones actuales han permitido desarrollar prototipos que detectan personas para establecer el distanciamiento social requerido por el virus COVID-19 [36]. Estos prototipos pueden ir desde vehículos aéreos no tripulados [37], a sistemas de vigilancia que permiten identificar la densidad social de cierta zona en la que se está evaluando [38].

Estos sistemas de video vigilancia permiten visualizar a las personas desde varias cámaras, midiendo el distanciamiento con la ayuda de algoritmos de IA [39]. Otros prototipos como el realizado por [40], permite identificar el uso de mascarilla en las personas utilizando algoritmos como SSD MobileNetV1-COCO, SORT, para generar alertas en lugares cerrados. Los datos que se obtienen de todas estas aplicaciones pueden ser procesados para generar estadísticas que permitan advertir de zonas en donde se cumple o no con las medidas establecidas por los gobiernos y así lograr reducir y prevenir contagios de esta enfermedad [41].

2.2. Marco Teórico

2.2.1. Procesamiento digital de imágenes.

El procesamiento digital de imágenes se encuentra en diversas aplicaciones como las industriales, agricultura, ganadería, medicina, satélites de observación, así como en diferentes aplicaciones de la índole social. Por tal motivo, en la actualidad se enfatiza el desarrollo de nuevas y mejores técnicas para lograr un adecuado tratamiento a las imágenes dependiendo de la aplicación para la cual se las requiere utilizar. El objetivo principal del procesamiento digital de imágenes es el de aplicar técnicas o algoritmos para lograr identificar patrones que se ajusten a la búsqueda de información requerida de la imagen [42].

Para lograr comprender el procesamiento que se puede realizar a una imagen se requieren conocer varios aspectos básicos que conforman el patrón de estas, para lo cual, a

continuación, se exponen las principales características que conforman las imágenes, así como los principales procesos que se siguen para obtener información de estas.

Pixel

Un pixel es la unidad más pequeña de una imagen que posee luminiscencia y colores específicos. Su medida depende del patrón de puntos que conforman una imagen, así como la resolución a lo largo de cada línea de la misma. Un píxel también puede ser el elemento más pequeño de un dispositivo sensible a la luz, como las cámaras que usan dispositivos acoplados por carga [43].

Imagen

Una imagen es una matriz formada por pixeles con diferente intensidad luminosa, es decir, es un conjunto de números los cuales guardan información como tamaño, forma, color, textura y pueden ser manipulados para alterar las características conjuntas de estas. Una imagen digitalizada puede ser descompuesta en matrices numéricas representadas en números binarios, decimales o hexadecimales que permiten identificar las características antes mencionadas y así poder desarrollar la identificación de patrones que permiten el procesamiento de las mismas [44].

Brillo

El brillo es un atributo de una imagen que permiten identificar la cantidad de luz o luminosidad que posee una imagen, es decir, permite identificar los tonos claros y oscuros. El brillo tiene que ver con la percepción del ojo humano, bajo este criterio, puede variar la sensación de brillo en una imagen de una persona a otra, sin embargo, se utiliza el contraste para definir valores que puedan ser utilizados para la caracterización de las imágenes [45].

Tono

El tono es el máximo cambio de color que soporta una imagen entre la gama de colores de longitud de onda como los son: rojo, verde, azul, cian, magenta, amarillo, naranja. Estos valores cambian dependiendo de la proporción de esta gama de colores y la transición entre estos [46].

Espacio RGB

El modelo RGB es un espacio conformado por los valores de los colores principales de representación de una imagen los cuales son: rojo, verde y azul. Los valores de RGB pueden ser representados en un espacio tridimensional como se observa en la Figura 1.

En base a estos colores y sus valores respectivos se puede obtener la información de un color específico, el cual es la combinación de la escala de estos tres. Como referencia el color negro está posicionado en el origen del sistema, lo que quiere decir que va a tener un valor de $R=0$, $G=0$, $B=0$; de esta forma se puede representar cualquier tipo de color en una escala de números enteros del 0 al 255, lo que permite generar una gama de 16,777.216 colores diferentes [1].

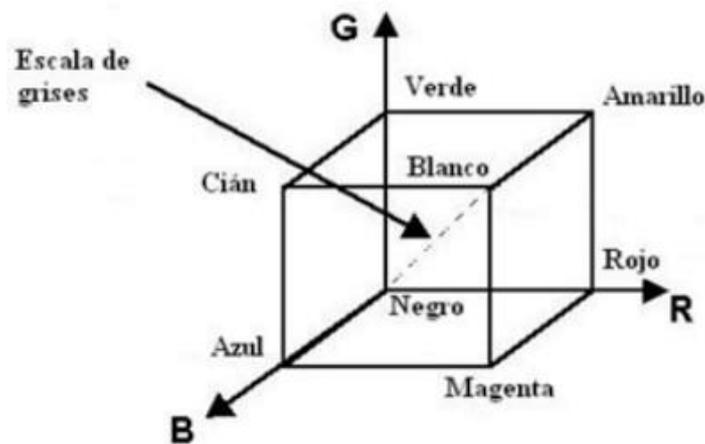


Figura 1. Representación de Colores RGB en escala tridimensional [1]

Mapa de Color

Un mapa de color o también conocido como BITMAP, es una composición de números asociados a un código de color representado con los valores de los colores básicos de la escala RGB. Como se observa en la Figura 2, un mapa de bits se caracteriza por el alto y el ancho de una imagen en pixeles, así como el número de bits que representan el color de cada uno de estos [2].

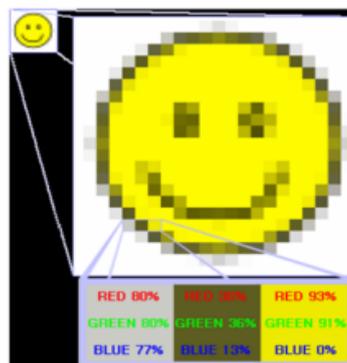


Figura 2. Mapa de Color de una Imagen [2]

2.2.2. Inteligencia Artificial

La inteligencia artificial se puede definir como la capacidad que poseen las máquinas de procesar datos para ejecutar algoritmos que les permitan aprender y tomar decisiones como las que realizará en su lugar un ser humano. Sin embargo, la gran diferencia que existe en este aprendizaje y toma de decisiones, es que las máquinas no requieren de descanso, pueden procesar volúmenes muy grandes de información y cometen una menor cantidad de errores que podría cometer su contraparte humana [47]. Por esta razón las tecnologías de IA se pueden encontrar en diversas aplicaciones en la actualidad como:

- Reconocimiento y clasificación de imágenes
- Procesamiento de datos médicos
- Mantenimiento Predictivo
- Detección de objetos
- Protección contra seguridad cibernética

Otra de las aplicaciones importantes que en la actualidad maneja la IA, es la realización de sugerencias o predicciones relacionadas con la salud, trabajo, relaciones interpersonales y compras. Un ejemplo de ello, son los algoritmos de aprendizaje que usan las redes sociales para mostrarnos publicidad o publicaciones de temas que nos atraen, identificándolos con nuestras últimas búsquedas en la red [47].

Así también, el uso de estas herramientas de IA, permiten obtener ventajas competitivas en el ámbito de negocios, ya que con la comprensión e interpretación de datos podemos lograr una mejor toma de decisiones sobre otros competidores [47].

Finalmente, dentro de la industria y manufactura existen máquinas o robots que realizan ciertas tareas, las cuales pueden ser riesgosas, aburridas o poseer cierto grado de dificultad para un operador humano. En la industria automotriz se utilizan estas máquinas para áreas de ensamble, pintura, etc. lo que ha permitido la mejora en la productividad, reducido el número de accidentes y generado menor cantidad de errores en el producto final [47].

2.2.3. Técnicas de inteligencia artificial

La visión por computador está definida como un campo de estudio el cual permite el desarrollo de técnicas que permitan al computador, entender u obtener información de imágenes y videos digitales. Esta disciplina es un subcampo de la inteligencia artificial y del

aprendizaje por computador, la cual desarrolla métodos de algoritmos de aprendizaje. El principal objetivo de la visión por computador es entender el contenido de las imágenes digitales extrayendo información como objetos, texto o alguna modelo tridimensional [48].

Las técnicas para analizar una imagen digital constan de herramientas las cuales se distribuyen según el nivel de procesamiento que se requiere realizar para analizar la información contenida en una imagen digital como observa en la Figura 3.

- **Pre procesamiento.** Secuencia que permite adaptar la información de una imagen para una mejora en los siguientes procesos.
- **Segmentación.** Proceso para realizar la partición de una imagen en varias regiones que contengan la información requerida para resolver el problema.
- **Detección de objetos y clasificación.** Determinación y clasificación de los objetos contenidos en la imagen.

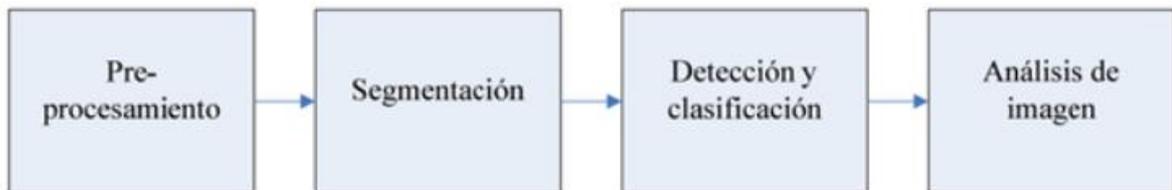


Figura 3. Niveles de procesamiento de una imagen [2]

- **Análisis de imagen.** Extraer la información necesaria acerca de lo que la imagen muestra.

Para realizar estos procesos, existen diversos métodos que se han desarrollado actualmente para lograr el procesamiento de imágenes utilizando inteligencia artificial. Estos métodos permiten el mejoramiento y clasificación de ciertas regiones u objetos con el objetivo de obtener la información requerida de la imagen en análisis. A continuación, se presentan los principales métodos utilizados para el procesamiento digital de imágenes [49].

Sistemas difusos

Los sistemas difusos son muy utilizados actualmente ya que permiten identificar y controlar sistemas lineales y no lineales que no puedan ser modelados matemáticamente. Este sistema está compuesto por reglas, las cuales permiten obtener un valor de la salida del sistema dependiendo de las entradas, estas reglas son consideradas un puente de transición entre las entradas y las salidas del sistema analizado.

En el procesamiento de imágenes, la lógica difusa ha encontrado una gran cantidad de

aplicaciones entre las cuales destacan la medicina, las telecomunicaciones e incluso permitiendo hallazgos espaciales. La razón principal para utilizar la lógica difusa en el procesamiento de imágenes es que, este tipo de algoritmos son una herramienta muy útil para la representación y permiten manejar eficientemente la vaguedad y ambigüedad [3].

El procesamiento difuso de imágenes consta de tres etapas principales como se observa en la Figura 4, Fuzzyficación de la imagen; Procesamiento de la máquina de inferencia y la Defuzzificación de la Imagen.

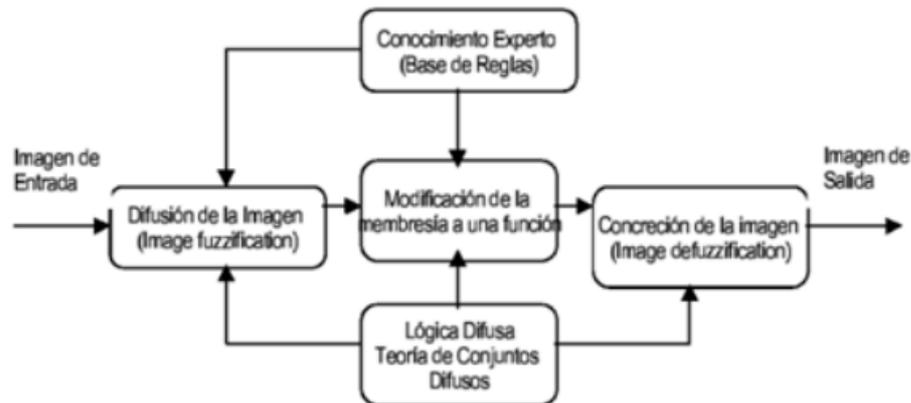


Figura 4. Niveles de procesamiento de una imagen [3]

La problemática sobre si un píxel analizado es más oscuro o brillante, o preguntas como la ubicación de cierto objeto en la perspectiva de la imagen son situaciones en donde la lógica difusa puede ajustarse más a la forma en la cual se maneja la incertidumbre de decisión del sistema. En aplicaciones de procesamiento de imágenes como la detección de objetos, análisis de escenas y análisis de imágenes médicas se requiere mucho más conocimiento o ser especialista en el tema en cuestión para lograr que el sistema difuso tenga una correcta toma de decisiones [3].

Redes Neuronales

El sistema nervioso de los animales está estructurado por neuronas que se enlazan para formar redes que procesan la información generada de los diferentes estímulos detectados por los sistemas sensoriales. Estas características han servido como base para modelar las redes neuronales artificiales (RNA). Los modelos comunes de estas redes poseen la característica de ser supervisados, es decir, existe un proceso de referencia que ya fue entrenado con anterioridad para obtener una salida en base a una entrada determinada del sistema [49].

Las imágenes digitales contienen demasiada información la cual en ciertos casos es

difícil de interpretar, por lo cual, la mayoría de las aplicaciones de las RNAs en imágenes digitales son las de segmentación y clasificación. La segmentación de una imagen puede diferenciarse en las siguientes funciones:

- **Segmentación de forma.** Consiste en obtener una nueva imagen binarizada, la cual está dividida en regiones que representan la información que se quiere analizar.
- **Detección de bordes.** Extracción de altas frecuencias en la imagen que representan los límites de objetos y contornos.
- **Segmentación de color.** División de la imagen en regiones que agrupan colores iguales o parecidos.
- **Segmentación de Textura.** División de la imagen basado en los patrones de variación espacial de las intensidades de color y grises.
- **Segmentación de movimiento.** detección de objetos dinámicos en una secuencia de imágenes.

En los últimos años, se han desarrollado varias estructuras de redes neuronales para el procesamiento de imágenes, en las cuales destacan la red neuronal celular (CNN), percepción multicapa (MLP) y redes neuronales recurrentes(RNN) como las más conocidas, y de las cuales se mencionaría más a detalle a continuación [49].

Red neuronal convolucional (CNN)

Basada en la teoría de los autómatas celulares, esta red permite la interacción de los vecinos de cada unidad en la red. Contiene una cantidad fija de unidades llamadas células, las cuales son sistemas no lineales, dinámicos, donde la información de entrada es codificada para conocer su comportamiento. Cada célula está localmente interconectada con múltiples entradas que vienen de otras células. Un aspecto interesante de la CNN además de sus variantes, es que se considera muy propicia para implementaciones en hardware, por lo cual es común encontrar diversas implementaciones de algoritmos CNN en chips o en GPUs [4]. La arquitectura de la CNN es la siguiente:

- **Capa de Convolución:** Analiza las imágenes de entrada y detecta si existen las características buscadas. En la salida de esta capa se obtiene un conjunto llamado mapa de características.
- **Capa de Pooling:** Es una operación que se aplica entre capas de convolución. Esta recibe el mapa de características, reduce el tamaño y filtra las características más importantes. En la salida de esta capa se obtiene un mapa de características comprimido.
- **Capa de Activación ReLU:** Esta capa cambia los valores negativos recibidos por ceros.

- **Capa Fully Conected:** Esta capa está colocada al final de la CNN y se conectan hacia las neuronas de salida. Al recibir los datos de entrada, esta aplica una combinación lineal y una función de activación para lograr clasificar la imagen de entrada. Finalmente, entrega a su salida un vector de tamaño igual al número de clases.

Perceptrón multicapa (MLP)

El modelo perceptrón multicapa, está compuesto por capas ocultas y capas de salida como se observa en la Figura 5, las cuales provienen de una serie de neuronas que reciben, procesan y envían datos hacia las neuronas consecutivas, procesando la información por medio de diversas operaciones matemáticas. En el procesamiento de imágenes las neuronas de entrada reciben los valores digitales que representan cierto píxel de una imagen [4].

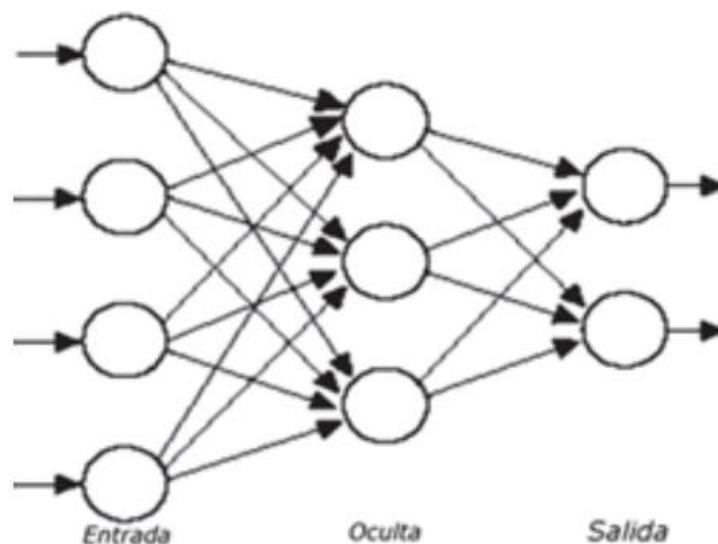


Figura 5. Modelo de una red neuronal perceptrón multicapa [4]

Por otro lado, las capas ocultas son las encargadas de identificar el nivel de complejidad en la relación que posee la capa de entrada con la capa de salida. Finalmente, la capa de salida es la encargada de representar el resultado de clasificación de la red neuronal; por lo cual existe una relación directa entre la capa de salida con el número de posibles clasificaciones que tendrá el sistema. Las conexiones que unen las neuronas se representan mediante pesos numéricos que indican el nivel de correlación entre cada muestra que se pretende clasificar; estos pesos se establecen durante la etapa de entrenamiento de la red. Cada neurona envía a la siguiente capa un valor conocido como valor de activación. Finalmente, en la etapa de entrenamiento se establece la mejor relación entre cada muestra y sus respectivas clases a través de un método de aprendizaje [4].

Redes neuronales recurrentes (RNN)

Las Redes neuronales recurrentes (RNN) tiene una gran capacidad de procesamiento y

obtención de datos secuenciales. Existen varias aplicaciones en donde se utiliza una RNN, como el análisis de vídeo, el análisis de la música y la subtitulación de imágenes, sin embargo, dependen de la capacidad de la RNN.

A diferencia de las redes neuronales antes mencionadas, las cuales buscan establecer la independencia entre los datos de entrada, las RNN guardan activamente sus valores secuenciales y temporales. Estas generalmente, incrementan la arquitectura de una red multicapa tradicional, con el aumento de iteraciones que conectan nodos adyacentes o pasos de tiempo. Estas iteraciones conforman la memoria interna de la red, la cual permite evaluar las características del dato actual con respecto al dato inmediatamente anterior.

Es importante considerar que la mayoría de las redes neuronales tradicionales del tipo MLP están limitadas a un mapeo uno a uno de la entrada y la salida; mientras que, las RNN pueden realizar mapeos de uno a varios, de varios a varios (traducción de idiomas), y de varios a uno (identificación de voz). Se utiliza un esquema computacional para representar los mapeos entre las “n” entradas, las salidas y la pérdida [50].

2.2.4. Algoritmos para detección de personas

Las técnicas de detección y clasificación de personas son importantes para el desarrollo de diversos aparatos tecnológicos que permiten ejecutar tareas en los campos de la robótica, interacción humano máquina, sistemas de video vigilancia, entre otros.

Estos dispositivos tienen como función principal el vigilar continuamente la zona especificada en donde se realizará la detección de las personas que aparezcan en la escena o fotograma analizado; sin ningún tipo de restricción como vestimenta, edad o sexo o de condiciones ambientales como lluvia, calor, neblina, u otras. Por la complejidad que requiere este procesamiento de las imágenes existen algoritmos más confiables que poseen alta tasa de fidelidad en la detección los cuales se presentará a continuación [51].

Descriptor HOG

El histograma de gradientes orientados (HOG) es ampliamente utilizado desde su implementación, la cual consiste en el cálculo de los gradientes horizontal y vertical, formando matrices de orientación y magnitud. En otras palabras, consiste dividir la imagen en pequeños bloques distribuidos a lo largo y ancho de la misma, y con un ligero solape entre ellos. Cada bloque se subdivide en celdas, y en estos últimos se calcula la magnitud y orientación de los gradientes en cada píxel. En cada una de estas divisiones se calcula el

HOG promediado por un peso gaussiano, y luego se guarda en el vector de características de la imagen. Con esta información se puede comparar las características de los espacios analizados con otras imágenes para generar el aprendizaje deseado [52]. El proceso de extracción del descriptor HOG evalúa un conjunto normalizado de histogramas locales de orientación de los gradientes. Las etapas de este proceso se describen a continuación:

- **Normalización color/gamma:** Consiste en aplicar un filtro a la imagen de origen para reducir la influencia de la iluminación. La normalización realiza una compresión gamma aplicada bien a los canales RGB o a la imagen en escala de grises.
- **Cálculo de Gradientes:** El objetivo de esta etapa es obtener la información de contorno y silueta de la imagen calculando los gradientes de primer o segundo orden píxel a píxel. En imágenes con canales RGB, se asigna el gradiente del píxel con mayor norma entre los canales.
- **Codificación de los histogramas locales de orientación:** La ventana de detección se divide en celdas que son regiones espaciales más pequeñas. Cada una de estas celdas, obtiene un histograma unidimensional de orientación de los gradientes de cada píxel existente en esa región.
- **Normalización del bloque:** La etapa de normalización permite mejorar el cambio en la iluminación y sombreado, así como el contraste de los bordes. Se calcula la energía acumulada en un bloque, el cual está conformado por un grupo de celdas. Esta energía permite normalizar cada celda del bloque, es decir cambia la pertenencia de un bloque a otro.

Red de canales agregados

El algoritmo de detección de personas llamado Agrégate Channel Features (ACF), es un detector que realiza una búsqueda exhaustiva, mientras que el modelo de persona que maneja es general, es decir toma en cuenta una base de datos determinada para generar el aprendizaje.

En cuanto a los canales empleados, ACF utiliza 6 canales HOG, un canal para la magnitud de gradiente, y 3 canales de color LUV. El conjunto de estos canales al ser aplicados sobre un conjunto de datos de imagen “I”, se combinan para determinar la región de interés [5]. En la Figura 6, se pueden ver el funcionamiento de la detección mediante los canales mencionados.

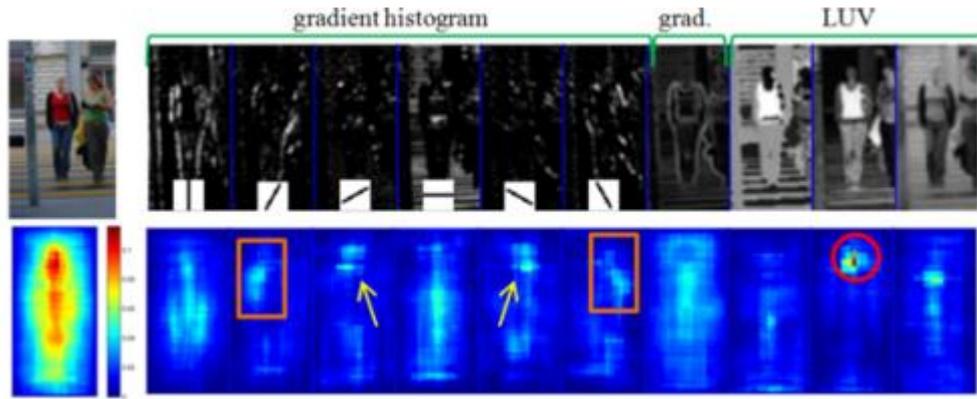


Figura 6. Canales del Algoritmo ACF para detección de personas [5]

Filtro de GABOR

El filtro de GABOR es una técnica de filtrado bidimensional utilizado en las imágenes, la cual permite resaltar los bordes del objeto buscado, aunque existan cambios de iluminación en la imagen. La primera parte de este procedimiento consiste en la obtención de los coeficientes mediante la convolución de los datos de la imagen “I”, con las wavelets correspondientes a esta. Las wavelets son filtros bidimensionales pasabanda provenientes del producto de una onda plana por una gaussiana.

Posteriormente, se aplica la transformada de Fourier Bidimensional Discreta a los coeficientes y a la imagen, obteniendo así números complejos como salida. Se realiza la convolución circular bidimensional entre los coeficientes resultantes de la imagen y el filtro. Finalmente, se obtienen los datos de la imagen a partir de la magnitud, realizando la transformada de Fourier inversa, a partir de la convolución circular entre los coeficientes de la imagen y el filtro [6].

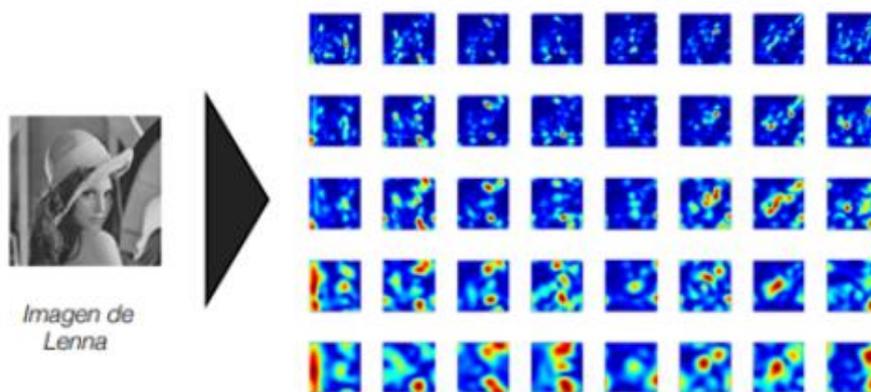


Figura 7. Ejemplo de filtro Gabor [6]

En la Figura 7 e presenta un ejemplo de la utilización de filtro de Gabor, en donde se puede analizar la salida de las convoluciones. Se observa que los filtros de Gabor que se desplazan en sentido horizontal; en frecuencias bajas tiene una respuesta elevada, consecuencia de la barra vertical situada al fondo de la imagen. Con ayuda de estos filtros se puede identificar información en diferentes zonas de la imagen analizada [53].

Máquina de vectores soporte (SVM)

Las SVM son una de las técnicas más utilizadas en el aprendizaje automático. Este método consiste en elaborar un “hiperplano” en un espacio con dimensiones infinitas o de valor muy elevado, el cual permita separar las clases del sistema. Una prudente separación entre clases permite una clasificación correcta de la nueva muestra. Por lo tanto, las Maquinas de Vectores Soporte maximizan la distancia del hiperplano a los puntos más cercanos de la muestra, como se observa en la Figura 8, el conseguir con iteraciones constantes llegar al punto más cercano de la muestra del plano se conoce como optimización del sistema [7]. Existen diversas situaciones en las que se puede utilizar una SVM dependiendo de los datos dentro del hiperplano, las cuales se va a detallar a continuación.

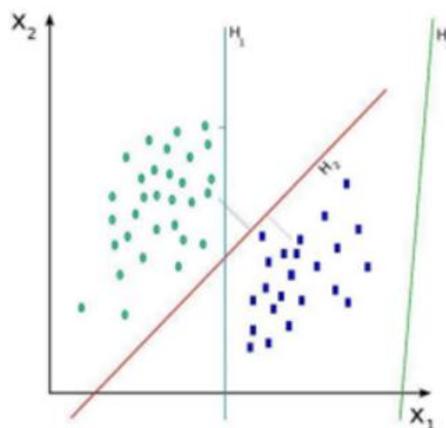


Figura 8. Hiperplano para representación de SVM [7]

- ❖ **Casos perfectamente separables** Si la distribución de los datos permite separar linealmente de forma perfecta en dos clases. En estos casos lineales se obtiene un número infinito de posibles hiperplanos, por lo que se requiere un método que permita seleccionar uno de ellos como clasificador Óptimo, el cual se conoce como máxima margin hyperplane o hiperplano óptimo de separación. Para obtenerlo se calcula la distancia perpendicular de cada muestra a un determinado hiperplano. La distancia más pequeña o margen determina que tan

alejado está el hiperplano de las muestras de entrenamiento.

- ❖ **Casos Cuasi-Separables linealmente** En la mayoría de los casos aplicables a la vida real, los datos no se pueden separar linealmente de forma perfecta, por lo que no se puede definir un hiperplano óptimo para estas aplicaciones. Para esto, requiere que se clasifique correctamente la mayoría de las observaciones a excepción de unas pocas que se encuentran fuera de los rangos de un hiperplano definido. La optimización permite que los datos que se encuentran en el margen o que lo violan, influyen sobre el hiperplano. A estos datos se les conoce como vectores soporte y son los que definen el clasificador.

Comparativa de Algoritmos para detección de personas

Todos los algoritmos analizados en la sección anterior permiten la detección de objetos, incluida la detección de personas. Existen ejemplos simples que permiten evaluar el funcionamiento de estos algoritmos; en este caso se puede observar en la tabla 1 la comparativa de los diferentes algoritmos evaluados con imágenes patrones propias del software MATLAB. Los algoritmos que mejor trabajan en la detección de personas son los de RCNN, los cuales generan mayor cantidad de falsos positivos pero mejor porcentaje de detección; mientras que el algoritmo ACF genera una mínima cantidad de falsos positivos y negativos, pero un porcentaje de detección de personas del 50 %, siendo este último un valor entregado por el algoritmo que indica en que porcentaje se parece la imagen evaluada con respecto a la base de datos con la que se realiza el entrenamiento.

Tabla 1 Comparativo de algoritmos para detección de personas basado en MATLAB

Algoritmo	SVM - HOG	ACF - HOG	RCNN - GABOR
Falsos Negativos Imagen 1	2	0	0
Falsos Negativos Imagen 2	2	0	0
Falsos Positivos Imagen 1	0	0	1
Falsos Positivos Imagen 2	0	1	3
Porcentaje de detección Imagen 1	8%	58.6%	90%
Porcentaje de detección Imagen 2	3%	49.7%	92%

En la siguiente sección se evaluó más a detalle los algoritmos para identificar dentro del sistema de adquisición de video, cual es el que mejor trabaja en la detección de personas. Sin embargo, en base al estudio de estado del arte y los ejemplos observados en el software MATLAB, los más relevantes para el diseño de la aplicación serán los de RCNN y ACF, de los cuales se menciona y evaluar en la sección siguiente de metodología.

Capítulo 3

Metodología.

3.1. Tipo de investigación.

La propuesta del proyecto se basa en diseñar un sistema de vigilancia, que permita la detección de personas en zonas de alta aglomeración, como en el caso de la zona comercial de Pelileo. Para el desarrollo de este estudio, se aplica una investigación de tipo bibliográfico y de índole descriptiva, por lo que permite identificar patrones de incumplimiento del distanciamiento social, implementando el prototipo de detección con la ayuda de investigaciones basadas en algoritmos de inteligencia artificial, con el objetivo de recolectar datos de píxeles y distancias que permitan el análisis de las zonas que tengan mayor riesgo de contagio de COVID 19 en la población.

3.2. Métodos de la investigación.

En el presente trabajo la investigación se basa en documentos o información secundaria tales como bases de datos, libros, artículos, publicaciones, manuales, estadísticas, tesis, etc. Otro fundamento de este proyecto es la investigación científica, ya que se analiza diversa documentación referente a algoritmos de inteligencia artificial y más en concreto, a la detección de objetos en diferentes aplicaciones de visión por computador. Finalmente, se utiliza estadística descriptiva, para obtener datos de error y valor medio; y realizar un análisis de estos con el objetivo de determinar el funcionamiento del prototipo implementado.

3.2.1. Técnicas de recolección de datos primarios y secundarios.

- **Primaria:** Para el presente proyecto de investigación se recopila la información a través de los datos de píxeles y distancias, obtenidos por la plataforma implementada. Por medio de esto, se puede recolectar datos cualitativos específicos que permiten un análisis de la situación actual de la población.
- **Secundaria:** En este proyecto se utiliza información obtenida de proyectos de grado, artículos, publicaciones de revistas, libros, internet y datos estadísticos.

3.3. Escenario de prueba.

Este proyecto de investigación se basa en el diseño e implementación de un sistema de detección de distanciamiento social mediante el uso de algoritmos de inteligencia artificial para controlar y prevenir contagios dentro de la zona comercial de Pelileo. Como se observa en la Figura 9, este sistema está compuesto por una cámara IP ubicada a una altura de 3 metros desde el suelo. Esta permite la captura y envío de imágenes o videos hacia un ordenador, a través de protocolo IP por una red WIFI de 2.4 GHZ.

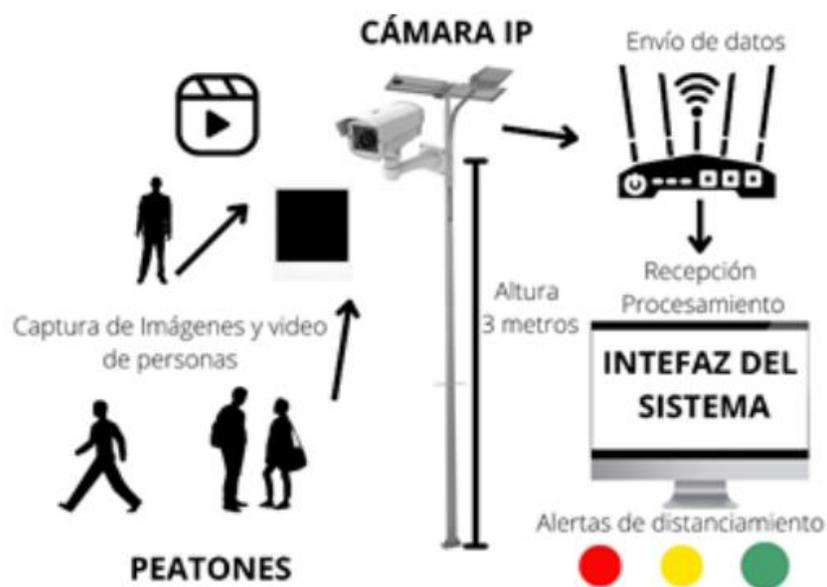


Figura 9. Esquema del Sistema de Detección de Distanciamiento Social

Este computador se encarga de recibir y procesar los datos de imágenes y píxeles para obtener, a través de una interfaz gráfica, la detección de las personas mediante un algoritmo de inteligencia artificial y así identificar la cantidad de transeúntes que han incumplido el distanciamiento social, generando alarmas dentro de la interfaz de usuario.

La programación de la interfaz gráfica y el algoritmo de inteligencia artificial están realizados en la plataforma MATLAB, ya que tiene una serie de herramientas y funciones de visión artificial que facilitan el desarrollo del sistema. Una de estas es Computer Vision Tool- box, la cual posee algoritmos, funciones y aplicaciones que permiten el diseño y evaluación de sistemas de procesamiento de video y visión artificial.

Entre las principales características de esta herramienta, es que se puede realizar detección y seguimiento de objetos, detección y clasificación de características; también permite la

automatización de flujos de trabajo en la calibración de cámaras simples, estéreo y ojo de pez. Las aplicaciones de visión artificial pueden generar etiquetado y validación de datos de entrenamiento, o también conocidos como “ground-truth”.

Existen funciones que posibilitan entrenar detectores de objetos utilizando algoritmos como YOLO, SSD y ACF; estos permiten también la detección y segmentación de objetos para el análisis de imágenes que poseen demasiada información. Se pueden realizar sistemas que ejecuten el reconocimiento facial y detección de peatones u otros objetos comunes. Finalmente, se puede configurar la capacidad gráfica del ordenador para poder conseguir un procesamiento más eficaz de los datos de píxeles y distancias analizados.

Finalmente, el sistema se despliega para obtener datos de distancias y píxeles, con el objetivo de comprobar la fiabilidad del sistema. Estas se las realiza en la Zona de Pelileo, específicamente en el barrio el Tambo, donde se tiene la concentración de locales de venta de Jeans más conocida. Específicamente en la Avenida Confraternidad, es la ubicación seleccionada ya que esta es en donde se tiene la mayor concentración de peatones debido a la alta cantidad de locales comerciales, como se observa en la Figura 10, esta posee una distancia de 400.29 metros.



Figura 10. Distancia de la Avenida Confraternidad de Pelileo El Tambo.

A continuación, se presentan las etapas que servirán de guía para el desarrollo del sistema de control de distanciamiento de personas en el sector comercial del barrio el tambo del cantón Pelileo.

■ *Recolección y Análisis de información.*

Realizar una investigación por medio de fuentes bibliográficas que tengan información acerca del diseño de sistemas de visión por computador y que apliquen algoritmos de inteligencia artificial para la detección de personas, con el objetivo de reunir la información necesaria para comprender el funcionamiento de los diversos algoritmos y dar cumplimiento a los objetivos planteados.

■ *Etapa II. Recopilación y Análisis de Datos.*

Realizar la recolección y procesamiento de datos de imágenes, píxeles y distancias en el cantón Pelileo barrio el tambo, en donde se capturó varias grabaciones (videos), los cuales fueron procesados y analizados para poder subdividirlos en una secuencia de imágenes, las mismas que a través de diversos algoritmos fueron evaluadas para obtener el que mejor permita la detección de personas.

■ *Etapa III. Diseño de la Interfaz.*

Una vez obtenido el algoritmo que mejor se ajusta a la aplicación, se procede al diseño de la interfaz de usuario que permite visualizar, realizar grabaciones, abrir videos previamente grabados y procesar la detección de personas, visualizando cuando una de estas se encuentre en un rango de contagio de 1.5 a 2 metros con una advertencia en rojo, mientras que, si está a una distancia mayor a esta, la aplicación indicara que no existe peligro. En la aplicación se muestran los mensajes y alertas de colores cuando existan incumplimiento del distanciamiento social.

■ *Etapa IV. Pruebas y Análisis de Datos.*

Finalmente, se desarrolla el despliegue del prototipo realizando diversas pruebas en la zona comercial de Pelileo, recopilando datos de píxeles y distancias para realizar un análisis de los resultados obtenidos con el sistema y poder elaborar la documentación final incluyendo las conclusiones, recomendaciones y trabajos futuros del presente proyecto.

3.4. Hipótesis y operacionalización de variables.

3.4.1. Hipótesis de investigación H.

¿Las lecturas de las distancias entre personas obtenidas por el prototipo de distanciamiento social desarrollado en este proyecto con relación a las distintas distancias que puede captar la cámara son fiables?

Hipótesis nula Ho.

¿Las lecturas de distancias entre personas obtenidas por el prototipo de distanciamientos social desarrollado en este proyecto no son fiables a cierta distancia de objetivo a cámara?

Variable independiente.

Distancia medida desde el dispositivo de captura de video hacia el objetivo de estudio

Variable dependiente.

Error Calculado entre los objetivos de estudio

Operacionalización de variables.

A continuación, se detalla la variable dependiente e independiente en la Tabla 2.

Tabla 2 Variables del proyecto de investigación

Variable independiente			
Variable	Concepto	Indicadores	Técnica e Instrumento
Distancia medida desde el dispositivo de captura de video hacia el objetivo de estudio.	Distancia de separación existente entre los objetivos de estudio y el dispositivo de captura de video.	-Longitud medida en el suelo hasta el dispositivo de captura de video (m). - Altura de ubicación del dispositivo de captura de video -Angulo del dispositivo de captura de video	-Flexómetro -Algoritmo de distanciamiento -Software Matlab - Transportador de ángulos (graduador)
Variable dependiente			
Variable	Concepto	Indicadores	Técnica e Instrumento
Error Calculado entre los objetivos de estudio	Error de Distancia que existe entre las personas para la detección del distanciamiento la cual va ser medida en el escenario de estudio.	- Porcentaje de error	-Computador -Video (mp4) -Imágenes (jpg.png) -Dispositivo de captura de video (cámara)

Población y muestra.

■ Población.

La población considerada para este proyecto se ha determinado con la recolección de los datos obtenidos por el sistema desarrollado. Se han realizado varias pruebas en donde se relaciona las variables de distancia de cámara hacia objetivo, distancia entre objetivos, distancia en píxeles y el error calculado; mismas que durante las pruebas realizadas a una distancia de 5, 10, 15 y 20 metros se obtuvo una población finita de 300 datos.

■ Muestra.

La muestra en este caso de estudio se toma a partir de la población de datos mencionada en el punto anterior. En donde se ha seleccionado un grupo de datos de distancia de cámara a objetivo desde 5 metros hasta 20 metros, y distancia entre personas de 1 metro hasta 2 metros consideradas como medidas reales. Para obtener la muestra a partir de los 300 datos se aplica un nivel de confianza del 90 % y un margen de error del 8 % y se calcula utilizando la ecuación 3.6 en donde se obtiene una muestra total de 80 datos cómo se observa en 3.9.

$$\text{Población} = N = 300 \quad (3.1)$$

$$\text{StdDev} = 0,5 \quad (3.2)$$

$$\text{nivel Confianza} = 90 \% \quad (3.3)$$

$$\text{Puntuación } Z = 1,65 \quad (3.4)$$

$$\text{margen Error} = 8 \% = 0,08 \quad (3.5)$$

$$N * (\text{Puntuación})^2 * \text{StdDev} * (1 - \text{StdDev})$$

$$(\text{margenError})^2 * (N - 1) + \text{Puntuación}Z^2 * \text{StdDev} * (1 - \text{StdDev})$$

$$300 * (1,65)^2 * 0,5 * (1 - 0,5)$$

$$(0,08)^2 * (300 - 1) + 1,65^2 * 0,5 * (1 - 0,5) \quad (3.7)$$

$$n = \frac{204,19}{2,59} \quad (3.8)$$

$$n = 78,83 \text{ w } 80 \quad (3.9)$$

3.5. Sistema de adquisición de video

El sistema de adquisición de video permite obtener las imágenes y videos desde la cámara y enviarlos a través de una puerta de enlace por protocolo IP al ordenador, que mediante el software MATLAB almacena los archivos JPG o PNG para las imágenes, y los archivos AVI o MP4, como los formatos para almacenar los videos del sistema como se observa en la Figura 11.



Figura 11. Esquema del sistema de adquisición de video.

Para la implementación del sistema de adquisición de video se requiere tomar en cuenta que debe haber una fuente de energía en el sitio donde se va a colocar la cámara web, así como un ROUTER que sirva como puerta de enlace para la transmisión de los datos hacia el ordenador.

Cámara IP.

La cámara IP es el elemento de que permite capturar imágenes, video, y transmitirlos a través de una red local hacia el ordenador que procesara estas posteriormente. En el mercado se pueden encontrar diversas cámaras que permiten cumplir este objetivo, sin embargo, se han considerado varios aspectos para la selección de esta.

Los requerimientos necesarios para la selección de una cámara IP para el sistema, se basan en el lugar donde se va a ubicar la cámara; como se ha mencionado anteriormente, esta va a estar ubicada a tres metros del suelo, esto indica que se requiere tener un punto de conexión a la red eléctrica en el sitio donde se va a colocar la cámara, sin embargo, se consideró colocar una cámara que posea autonomía propia, es decir que tenga una batería y un cargador. En la actualidad existen cámaras IP que poseen un panel solar que permite la carga de la batería de este dispositivo por lo que le proporciona una autonomía las 24 horas del día, esto serviría para trabajos futuros y para este proyecto utilizamos una cámara IP que se conectara en una toma eléctrica para su funcionamiento.

Otro aspecto importante para considerar es la protección que posee la cámara, ya que va a estar expuesta a la intemperie, esta requiere cierta protección industrial que evite danos debido a los cambios de clima repentinos de nuestro país por lo que se requiere una protección IP62 como mínimo.

Se requiere que esta cámara pueda transmitir la información mediante una red de 2,4GHZ ya que es la más común en dispositivos inalámbricos y es compatible con todos los Routers que se posee actualmente en las redes domésticas y empresariales.

Para esto se seleccionó la Cámara IP PTZ Homkeep que se observa en la Figura12, esta cámara posee grandes características que permiten la captura de video bajo circunstancias de poca luz, clima lluvioso y con una gran resolución.



Figura 12. Cámara IP SMART ONLY

Como se observa en la Tabla 3, la cámara selecciona tiene las siguientes características técnicas:

Tabla 3 Características Técnicas de la cámara IP PTZ HOMKEEP

Descripción	Cámara IP SMART ONLY.
Batería	19200 mAh
Antena	5dBi
Red WIFI	2.4 GHz
Resolución	1080p
Zoom	Digital 4X
Nivel de Protección	IP68
Precio	\$ 30

Esta sección se basa en determinar la fiabilidad y confiabilidad técnica de las medidas realizadas por el sistema implementado para la detección de distanciamiento social. Se ha desplegado el sistema en la Zona Comercial de el Tambo como se mencionó anteriormente, considerando que, una cámara ubicada en la Av. Confraternidad puede medir 20 metros hasta el objetivo, y al tener aproximadamente 10 metros la distancia entre veredas; con una única cámara se puede cubrir un área de 200 metros cuadrados.



Figura 13. Cámara Seleccionada para la Obtención de Datos de Prueba.

Para realizar estas pruebas se seleccionó la cámara número 11 como se observa en la Figura 13, debido a que en esta ubicación se encuentra la mayor concentración de locales comerciales de la Avenida, la ubicación Real se la puede observar en la Figura 14.



Figura 14. Ubicación Real de la Cámara de Pruebas

3.6. Diseño del algoritmo para detección de personas

La implementación del algoritmo de Visión Artificial para la detección de personas se realiza en la plataforma de programación MATLAB, con el fin de tener un procesamiento de datos en tiempo real. Como se observa en el marco teórico, existen diversos métodos o

algoritmos, que permiten detectar a peatones dentro del cuadro de visión de una imagen. Para seleccionar el mejor método que provee MATLAB, como se observa en la Figura 15, se realiza una comparación entre los métodos de red de canales agregados (ACF) y redes neuronales convolucionales (RCNN) basadas en regiones, ya que son las más adecuadas según el estudio realizado.

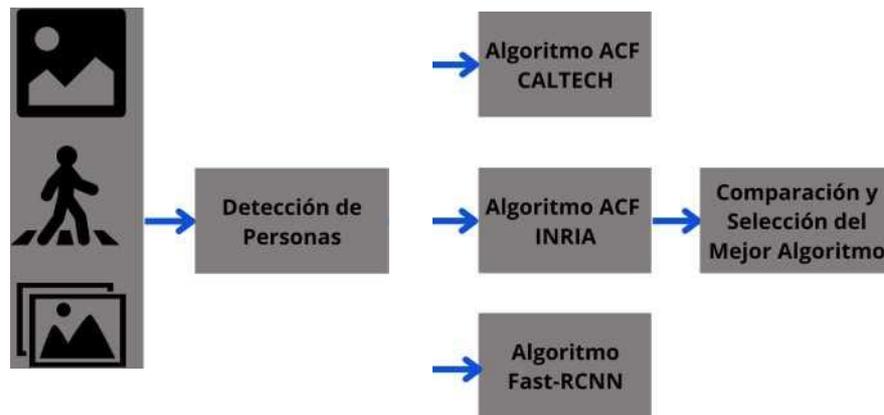


Figura 15. Diagrama de Bloques de comparativa y selección de algoritmos.

La estructura de un sistema de visión basado en ACF, está compuesto de la aplicación de histogramas de gradiente y magnitudes de gradiente con características de color; siendo los canales más comunes el RGB, tono - saturación (HSV) y luminancia (LUV). Su eficiencia computacional se incrementa por aproximaciones o uso de múltiples modelos a través de diferentes escalas y ventanas vecinas.

Los píxeles de los canales ACF que se transmiten al espacio de la submuestra se transforman en un vector de características, el cual será entrenado por el modelo de cascadas AdaBoost. Esta es una técnica en donde se combinan múltiples clasificadores débiles y se obtiene un clasificador fuerte, siendo esta la que se procesa en la mayoría de los programas. En el software MATLAB se utiliza la función detectPeopleACF, la cual posee dos modelos de detectores entrenados. El primer modelo es el conjunto de datos de peatones INRIA, el cual consta de 614 detecciones de personas para entrenamiento y 288 para pruebas con un tamaño de 100x41 píxeles cada imagen, siendo este modo el que utiliza la función por defecto.

El segundo modelo es el de CALTECH, debe ser configurado como galtech-50x21”; este consta de aproximadamente 10 horas de video en resolución 640x480, a una frecuencia de 30Hz tomado de un vehículo en circulación en un entorno urbano, en donde se capturaron alrededor de 250.000 fotogramas de los cuales constan 2300 peatones únicos. Esta función

en cualquiera de los dos modelos, entrega como datos de salida un vector que posee las coordenadas x, y, así como las dimensiones de ancho y alto de la caja en donde se identifican a la persona.

Por otro lado, las R-CNN se utilizan para detectar varios objetos en una imagen. Esta permite encontrar en la región de la imagen el objeto. Luego extrae las características CNN de la región y, por último, clasifica el objeto utilizando las características extraídas. Existen tres variantes de este algoritmo: R-CNN, Fast R-CNN y Faster R-CNN, en donde la velocidad de entrenamiento y detección mejora de la primera a la última. La función de Matlab que permite realizar la identificación de objetos mediante este método es `visión.PeopleDetectorACF`. Esta función permite la identificación de personas en base a dos modelos entrenados los cuales son `UprightPeople_128x64` y `UprightPeople_96x48`, dependiendo del tamaño de imagen que se está analizando. Esta función en ambos modelos, entrega como datos de salida un vector que posee las coordenadas x, y, así como las dimensiones de ancho y alto de la caja en donde se identifica a la persona.

Una vez analizados los algoritmos se procede a la implementación en MATLAB para establecer una comparativa entre todos los métodos y poder seleccionar el mejor para la investigación. El protocolo de pruebas establecido utiliza un banco de 10 fotografías diferentes con un tamaño de 400 x 266px para la primera prueba como se observa en la Figura 16, a continuación, tenemos las imágenes de 800 x 533px y se pueden visualizar en la Figura 17 que se utiliza para la segunda prueba y por último se observa la Figura 18 con imágenes de una resolución de 1080 x 720px para la última prueba. En este conjunto de imágenes se encuentran diferentes personas ejecutando varias actividades como se observan en el diferente grupo de imágenes, esto se utiliza para poder evaluar el algoritmo de detección de personas de una manera correcta.



Figura 16. Diagrama de Bloques de comparativa y selección de algoritmos.



Imagen 4

Imagen 5

Imagen 6

Figura 17. Imágenes de prueba de una resolución 800x533 px.



Imagen 7

Imagen 8

Imagen 9

Imagen 10

Figura 18. Imágenes de prueba de una resolución 1080x720 px.

Para la evaluación de los diferentes algoritmos, se tomó en consideración los siguientes aspectos que nos permiten analizar cuál de los tres algoritmos es el más estable y obteniendo resultados significativos y se detallan a continuación:

- **Falsos Positivos:** Existe un falso positivo cuando se detecta a un objeto que no corresponde a la búsqueda.
- **Falsos Negativos:** Existe un falso negativo cuando no se detecta al objeto que se está buscando.
- **Porcentaje de Detección:** Es un valor que entrega el algoritmo de reconocimiento, el cual indica el porcentaje de exactitud que tiene el objeto detectado comparándolo con la base de datos de entrenamiento. En este caso, se calcula el valor medio de todas las detecciones, para tener un porcentaje general de la imagen.

Una vez evaluados los diferentes sistemas de detección de personas basados en los algoritmos analizados con anterioridad, se puede observar en la Tabla 4, los resultados para imágenes de 400x250 píxeles, mismos que se graficaron y analizaron en base de un diagrama de caja. Como se observa en la Figura 19 se obtuvo un promedio de detección en el algoritmo ACF CALTECH del 70 %, una media de falsos positivos de 0 (Figura 20), y una media de falsos negativos de 1 Figura 21 para los algoritmos ACF CALTECH e INRIA.

Tabla 4 Tabla comparativa de imágenes de 400x250px

	ACF INRIA			ACF CALTECH			R-CNN		
	% Detección	Falsos Positivos	Falsos Negativos	% Detección	Falsos Positivos	Falsos Negativos	% Detección	Falsos Positivos	Falsos Negativos
Imagen 1	58,64 %	0	0	31,41%	0	2	28,96 %	1	2
Imagen 2	47,34 %	0	0	78,57%	0	4	62,21 %	1	0
Imagen 3	20,01 %	0	2	67,80%	2	0	24,29 %	2	0
Imagen 4	86,21 %	0	0	71,62%	0	0	61,80%	0	0
Imagen 5	26,75 %	1	2	73,58%	1	0	55,58%	1	1
Imagen 6	15,03 %	1	6	73,95%	1	2	12,21 %	1	3
Imagen 7	0%	0	0	70,23%	0	0	18,57%	0	1
Imagen 8	29,29 %	0	2	58,23 %	0	1	37,42 %	0	2
Imagen 9	80,87 %	0	2	53,51 %	0	2	79,58%	0	3
Imagen10	65,16%	0	0	87,50%	0	1	54,93 %	1	0

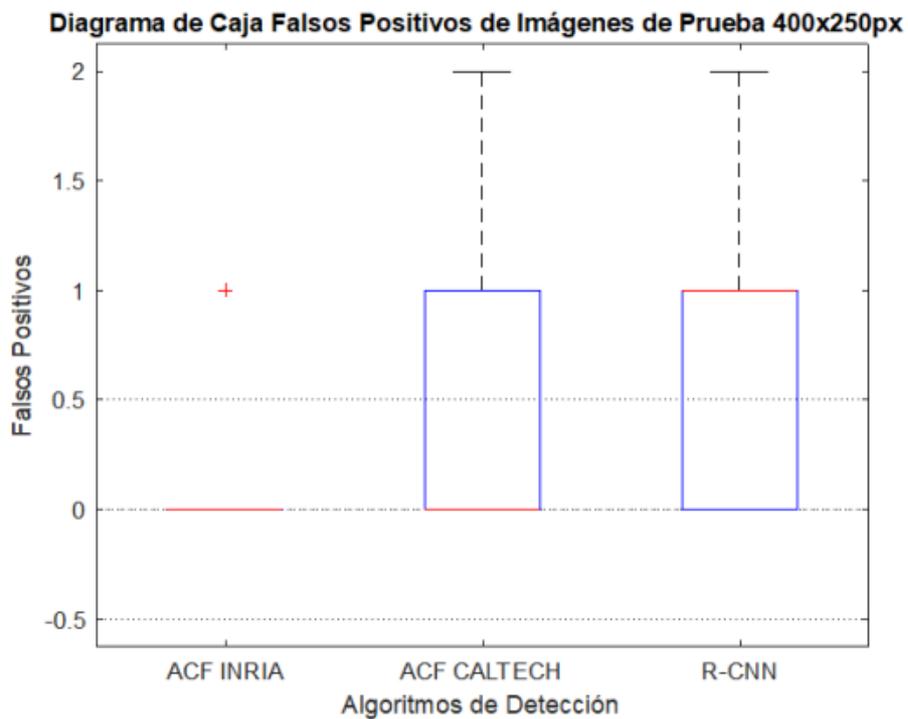


Figura 19. Diagrama de Caja Falsos Positivos en Imágenes de 400px.

Diagrama de Caja Falsos Negativos de Imágenes de Prueba 400x250px

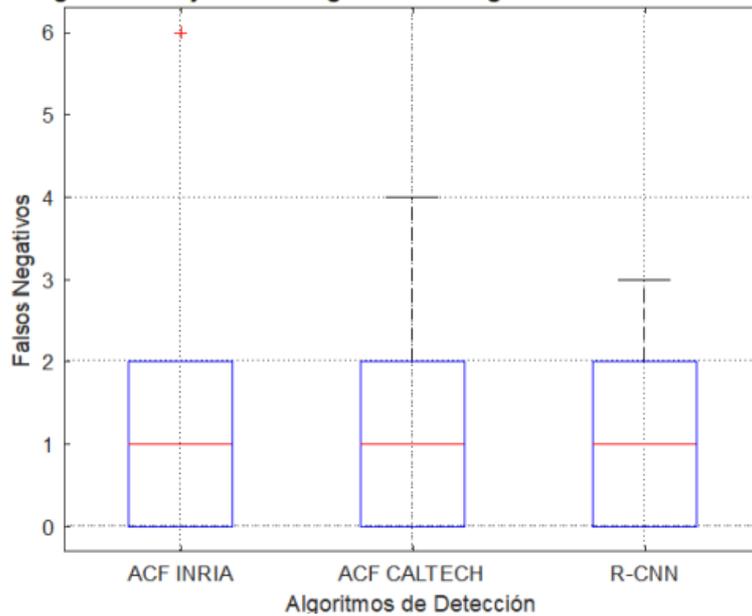


Figura 20. Diagrama de Caja Falsos Negativos en Imágenes de 400px.

Diagrama de Caja %Detección de Imágenes de Prueba 400x250px

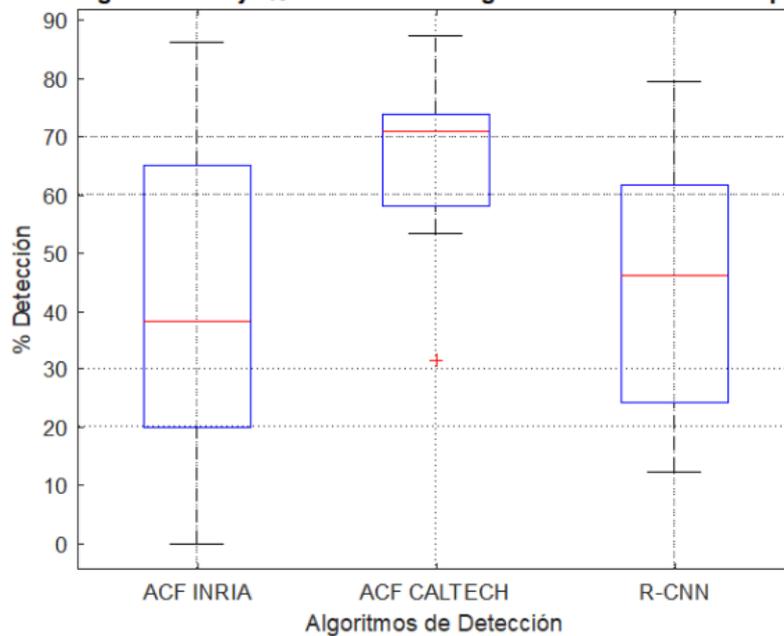


Figura 21. Diagrama de Caja % Detección en Imágenes de 400px.

Por otro lado, en las imágenes de 800x520 píxeles, se puede observar en la Tabla 5 y en las gráficas Figura 22, Figura 23 y Figura 24, se mantiene un promedio de falsos positivos de 0 en el algoritmo ACF CALTECH, el menor promedio de falsos negativos es del algoritmo ACF INRIA con un valor de 0 y se tiene un porcentaje de detección de 62 % para ACF CALTECH.

Tabla 5 Tabla comparativa de imágenes de 800x520px

	ACF INRIA			ACF CALTECH			R-CNN		
	% Detección	Falsos Positivo	Falsos Negativo	% Detección	Falsos Positivo	Falsos Negativo	% Detección	Falsos Positivo	Falsos Negativo
Imagen 1	56.68%	0	1	57.39 %	0	3	49.37%	0	1
Imagen 2	39.92%	1	0	70.24 %	0	4	55.52 %	1	1
Imagen 3	51.42%	1	0	79.43%	1	0	62.28%	1	0
Imagen 4	83.35 %	0	0	56.76%	0	0	47.29 %	0	0
Imagen 5	36.21 %	2	0	59.03%	1	0	37.25 %	2	1
Imagen 6	31.54%	1	1	72.49%	2	0	66.98%	1	0
Imagen 7	39.30%	0	0	59.95%	0	0	42.73 %	0	1
Imagen 8	37.58%	1	1	61.88%	0	1	28.15%	0	1
Imagen 9	30.91 %	1	0	55.95%	0	1	62.29 %	1	0
Imagen	48.38%	1	0	97.67%	0	1	78.98 %	0	0

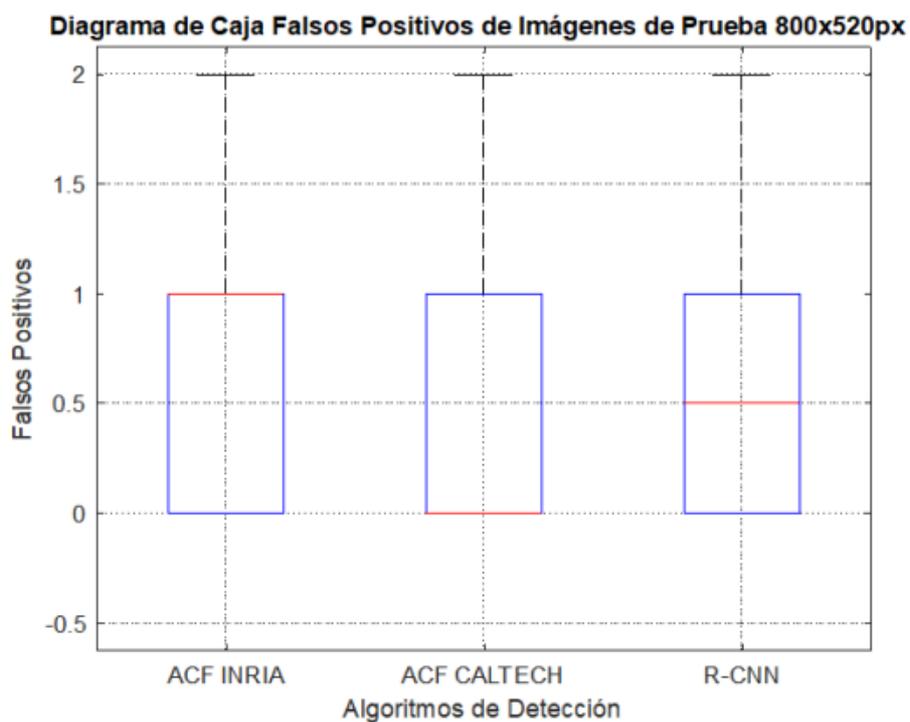


Figura 22. Diagrama de Caja Falsos Positivos en Imágenes de 800px.

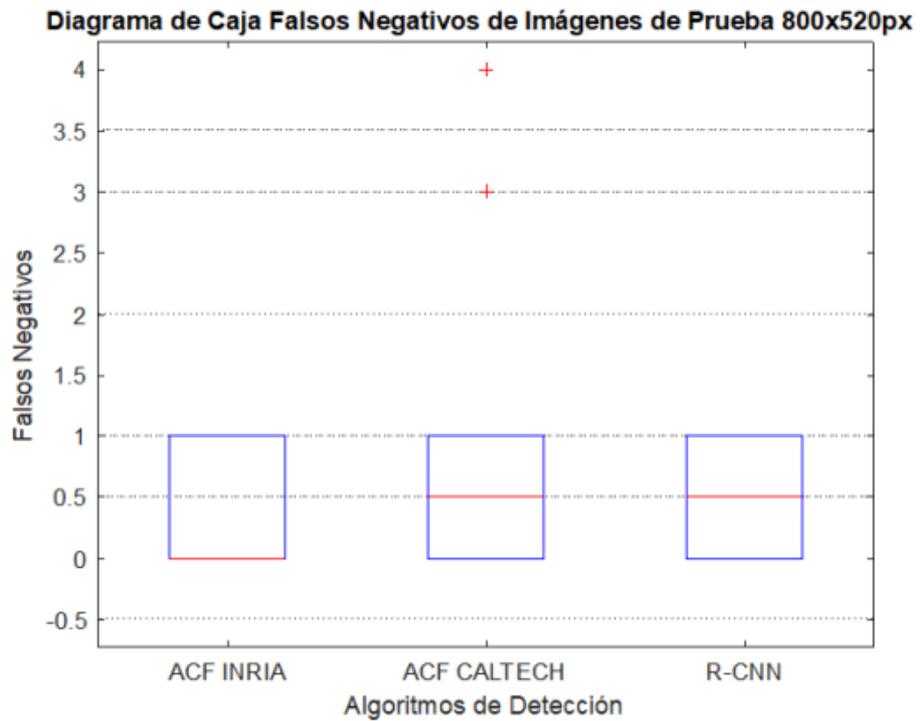


Figura 23. Diagrama de Caja Falsos Negativos en Imágenes de 800px.

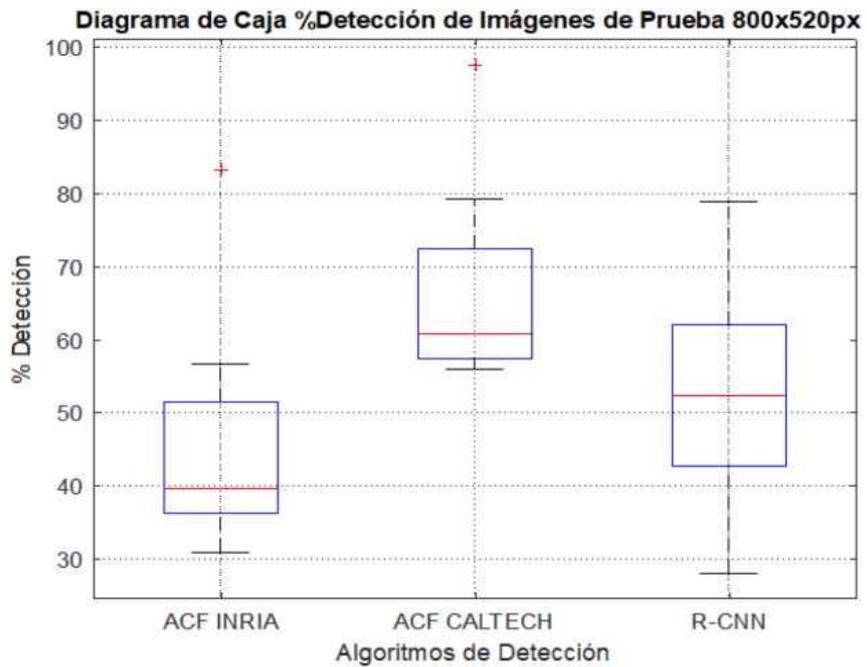


Figura 24. Diagrama de Caja %Detección en Imágenes de 800px.

Finalmente se observa en la Tabla 6, Figura 25, Figura 26, Figura 27, los resultados de las imágenes en tamaño 1080x720 pixeles, en donde los falsos positivos y negativos observados son casi nulos, salvo en casos en particular donde existe alto tráfico de personas y vehículos en el cual se obtuvieron 3 falsos positivos. Los algoritmos más estables son los ACF con su base de datos INRIA y CALTECH, sin embargo, este último demostró porcentajes de detección estables y superando el 60 %; por lo cual se decidió la elección de este para la implementación de la plataforma de video realizada en este proyecto.

Tabla 6 Tabla comparativa de imágenes de 1080x720px

	ACF INRIA			ACF CALTECH			R-CNN		
	% Detección	Falsos Positivos	Falsos Negativos	% Detección	Falsos Positivos	Falsos Negativos	% Detección	Falsos Positivos	Falsos Negativos
Imagen 1	56.92%	0	0	27.15%	0	1	14.29 %	0	0
Imagen 2	50.53 %	0	0	51.01%	0	3	61.34%	0	2
Imagen 3	65.11%	0	0	91.69%	1	0	41.96%	0	0
Imagen 4	92.28%	0	0	51.93%	0	0	28.67%	0	1
Imagen 5	38.54%	1	0	59.52%	2	0	53.47 %	0	1
Imagen 6	54.32 %	3	0	66.80%	3	0	29.75 %	3	1
Imagen 7	48.51 %	0	0	63.31 %	1	0	59.36 %	1	1
Imagen 8	40.80 %	2	0	62.22%	0	1	44.79 %	0	1
Imagen 9	27.59 %	1	0	54.41 %	0	1	68.54 %	0	2
Imagen 10	39.91 %	1	0	65.19%	2	1	64.76 %	0	0

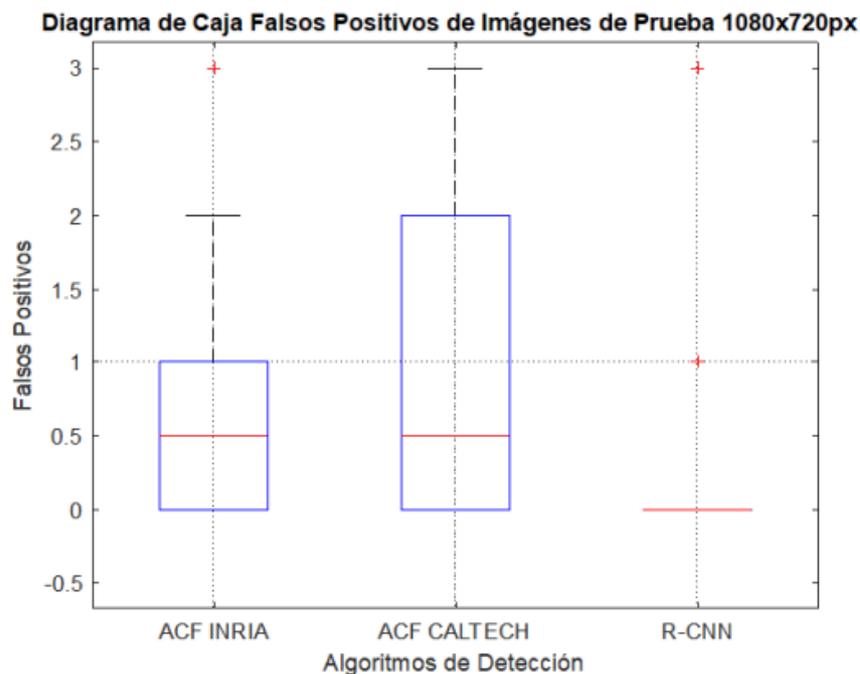


Figura 25. Diagrama de Caja Falsos Positivos en Imágenes de 1080px.

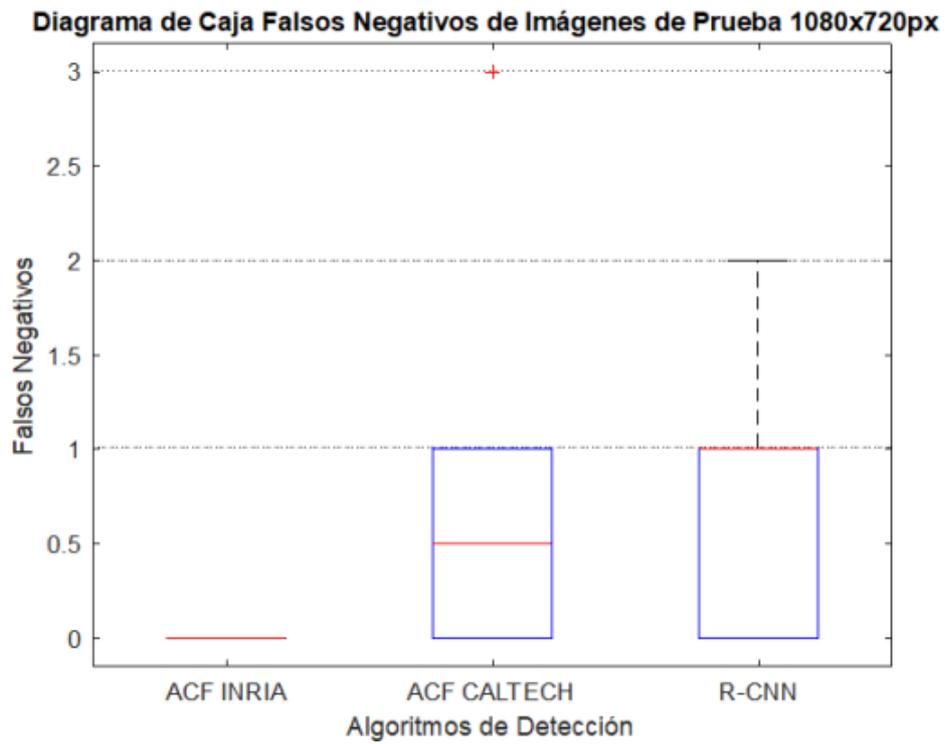


Figura 26. Diagrama de Caja Falsos Negativos en Imágenes de 1080px.

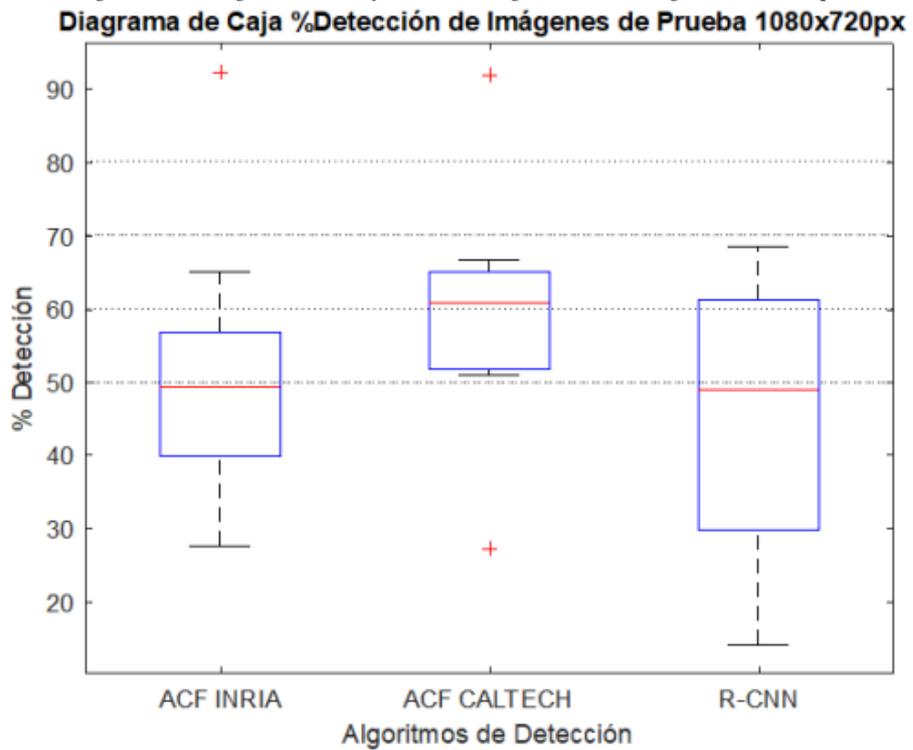


Figura 27. Diagrama de Caja %Detección en Imágenes de 1080px.

3.7. Cálculo de distancia entre personas

Una vez seleccionado el algoritmo de detección, se procede a realizar las operaciones que van a permitir el cálculo de la distancia entre personas. Para esto se debe lograr la conversión de píxeles a una variable medida en nuestro entorno, en este caso metros (m). La función `detectPeopleACF` entrega un vector con los datos de coordenadas (x, y) del punto de referencia del objeto detectado, el ancho y el alto de la caja que contiene el segmento de la imagen donde se encuentra lo requerido por la función.

Para lograr obtener la relación requerida de píxeles a metros se realizaron las siguientes pruebas:

- Se colocó la cámara a una altura de 3 metros.
- Se procedió a tomar medidas de 0.5, 1, 1.5, 2 metros de distancia marcando el punto inicial y el punto final.
- Se colocó a dos personas en los extremos marcados para tomar las medidas correspondientes.
- Se procedió a realizar capturas de video en las diferentes posiciones de medida, con el fin de obtener 60 datos diferentes para cada una de las distancias.
- Se realiza el análisis de datos para obtener una transformación estimada que convierte los píxeles en distancia.

Para poder obtener un punto de referencia y comenzar a medir la distancia entre los individuos, se busca calcular el punto medio del segmento en donde se detectó una persona. Para esto se utilizaron las coordenadas en (x, y) y el tamaño de la caja donde se encuentra el individuo, para poder calcular dos puntos, uno superior y uno inferior como se observa en la Figura 28, una vez obtenidos estos puntos se calcula la distancia entre ellos y su punto medio. Finalmente se calcula la distancia entre los puntos medios calculados entre cada individuo y esa será la distancia en píxeles que analizaremos.



Figura 28. Calculo de la distancia entre objetivos en pixeles.

Una vez logrado obtener los datos en pixeles requeridos, se procede a evaluar las imágenes de prueba para distancias fijas de 0.5, 1, 1.5 y 2 metros como se observa en la Figura 29.



Figura 29. Pruebas de algoritmo a distancia de 0.5, 1, 1.5, 2 metros.

Los resultados obtenidos de la medición de estas imágenes de prueba se pueden observar en la Tabla 7, en esta se encuentra el promedio calculado de cada imagen observando un incremento proporcional entre los pixeles y la distancia fija medida.

Tabla 7 Datos obtenidos en las pruebas de medición a 0.5, 1, 1.5 y 2 metros

	0,5 metros	1 metro	1,5 metros	2 metros
Prueba 1	123,09	164,01	267,32	312,28
Prueba 2	121,77	188,67	255,28	318,67
Prueba 3	123,98	171,96	234,72	308,55
Prueba 4	116,47	183,28	232,02	329,72
Prueba 5	120,94	196,94	232,90	312,85
Prueba 6	125,18	173,52	232,75	312,09
Prueba 7	122,99	162,29	244,62	307,08
Prueba 8	119,26	181,21	251,51	315,29
Prueba 9	123,84	190,86	219,44	316,77
Prueba 10	124,62	168,34	254,33	315,99

Con estos datos se procede a calcular la función que aproxima el valor de pixeles en metros, para esto se sacó el promedio de las 10 medidas primeras y 10 medidas últimas. Una vez obtenidos estos datos, se precedió a calcular la ecuación de la siguiente forma:

$$metros = \left(\frac{y^2 - y^1}{x^2 - x^1} \right) * (pixeles - x_1) + y_1 \quad (3.10)$$

$$metros = \left(\frac{2 - 0,5}{314,92 - 122,21} \right) * (pixeles - 122,21) + 0,5 \quad (3.11)$$

$$\text{metros} = \left(\frac{2 - 0,5}{314,92 - 122,21} \right) * (\text{pixeles} - 122,21) + 0,5 \quad (3.12)$$

$$\text{metros} = 0,0078 * \text{pixeles} - 0,9532 + 0,5 \quad (3.13)$$

$$\text{metros} = 0.0078 * \text{pixeles} - 0.4532 \quad (3.14)$$

Finalmente, para comprobar el correcto funcionamiento de la estimación obtenida, se realiza 5 pruebas cambiando la distancia de los individuos como se observa en la Tabla 8. Los resultados obtenidos indican que el porcentaje de error calculado entre la distancia real

$$\% \text{error} = \frac{\text{Valor Medido} - \text{Valor Estimado}}{\text{Valor Medido}} * 100\% \quad (3.15)$$

y la estimación del sistema (ecuación 3.15) no supera un redondeo del 11% a 12% en todos los casos. Posteriormente se realiza pruebas para comprobar la fiabilidad del sistema que se menciona en el capítulo 4 de resultados y discusión.

Tabla 8. Prueba de estimación del algoritmo de pixeles a metros

	Medida Real	Estimación	%Error
Prueba 1	0.75 metros	0.6892 metros	8.1067%
Prueba 2	1.30 metros	1.4372 metros	10.5538%
Prueba 3	1.80 metros	1.7127 metros	4.85 %
Prueba 4	2.20 metros	2.0916 metros	4.9273 %
Prueba 5	3 metros	2.9377 metros	2.0767%

Una vez conseguida la distancia entre individuos, se procede a la programación, integración e implementación de la interfaz de usuario que permite observar los casos de aglomeración o incumplimiento del distanciamiento que se verá en las siguientes secciones.

3.8. Lógica de Programación del Sistema

Para comprender el funcionamiento del sistema de detección implementado, se han desarrollado una serie de diagramas de flujo para comprender de manera más gráficas la implementación de los algoritmos que contienen la lógica de programación del sistema. Como se observa en la Figura 30, la plataforma sigue el siguiente diagrama de flujo, mismo que contiene procesos de los que se hablara a continuación.

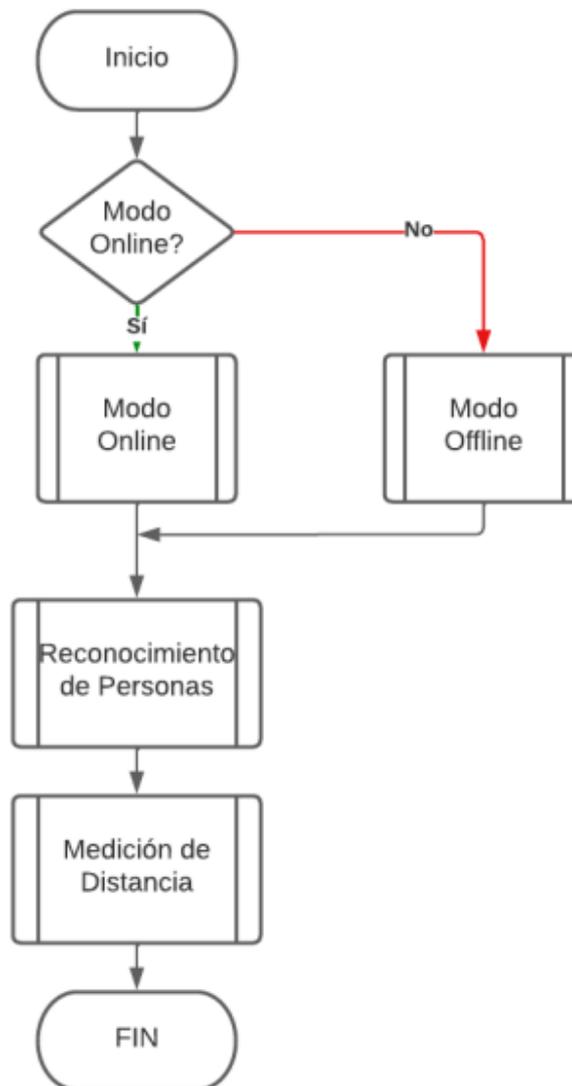


Figura 30. Diagrama de Flujo Principal del Sistema.

3.8.1. Diagrama de Flujo Detección de Personas

Como se mencionó anteriormente se ha seleccionado el algoritmo ACF CALTECH para la detección de personas. Como se observa en la Figura 31, el proceso para detectar a las personas transforma los datos del frame de video actual en un valor numérico. Después aplica la función detect para obtener los datos de posición de las cajas que contienen a la persona detectada. Se utiliza la función de cálculo de distancia, se muestra el frame, la caja y las leyendas respectivas en pantalla y pasa al siguiente frame para continuar con la detección.

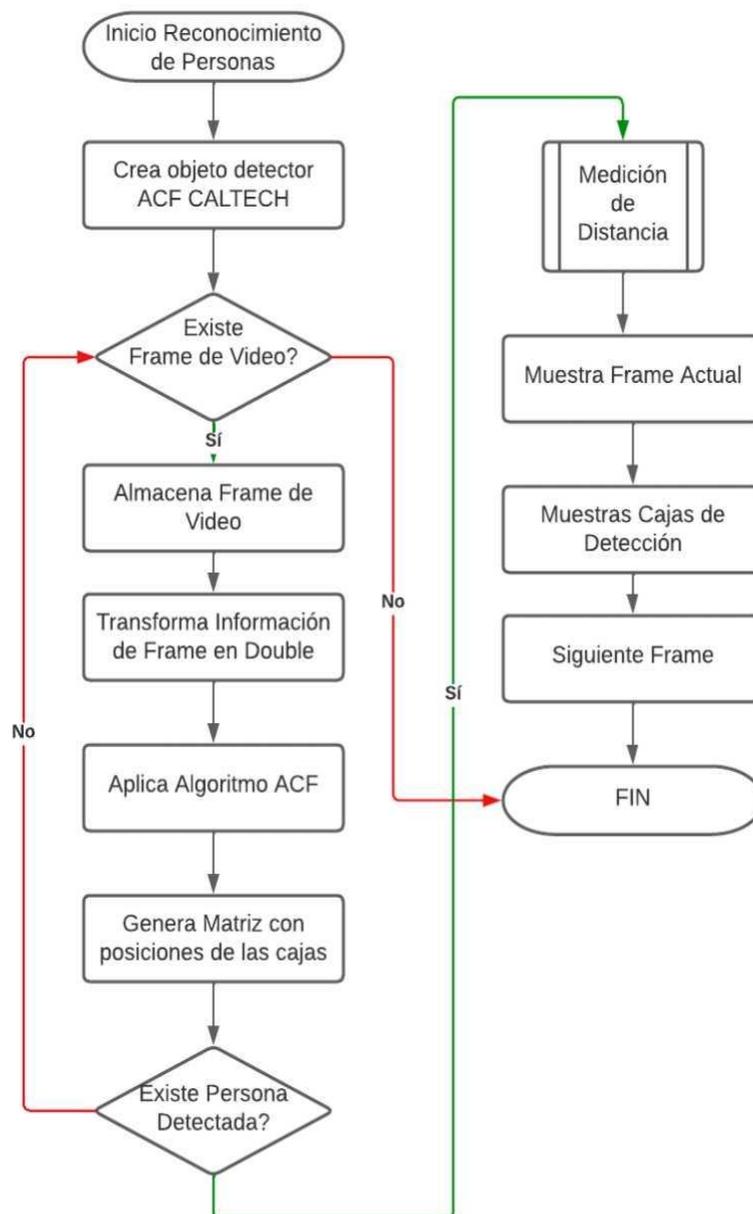


Figura 31. Diagrama de Flujo del Algoritmo de Detección de Personas.

3.8.2. Diagrama de Flujo Medición Distancia

Para realizar el cálculo de la distancia, se aplica lo desarrollado en secciones anteriores, para esto se sigue el diagrama que se muestra en la Figura 32. Aquí se crea un vector de igual tamaño que la cantidad de cajas obtenidas en la detección de personas. Se utiliza un ciclo For para realizar el cálculo de las distancias horizontales y verticales existentes entre la caja actual a analizarse y la caja más próxima. Se realiza el cálculo de la distancia entre ellas haciendo un cálculo del punto medio entre cada una de ellas y aplicando la ecuación que permite obtener la distancia entre dos puntos. Se transforma esta distancia de pixeles a metros aplicando la ecuación 3.14. Se continúa con la estimación para las siguientes cajas, comprobando si ya se realizó el cálculo de alguna de ellas, hasta que se haya conseguido todas las distancias posibles dependiendo de la cantidad de cajas detectadas.

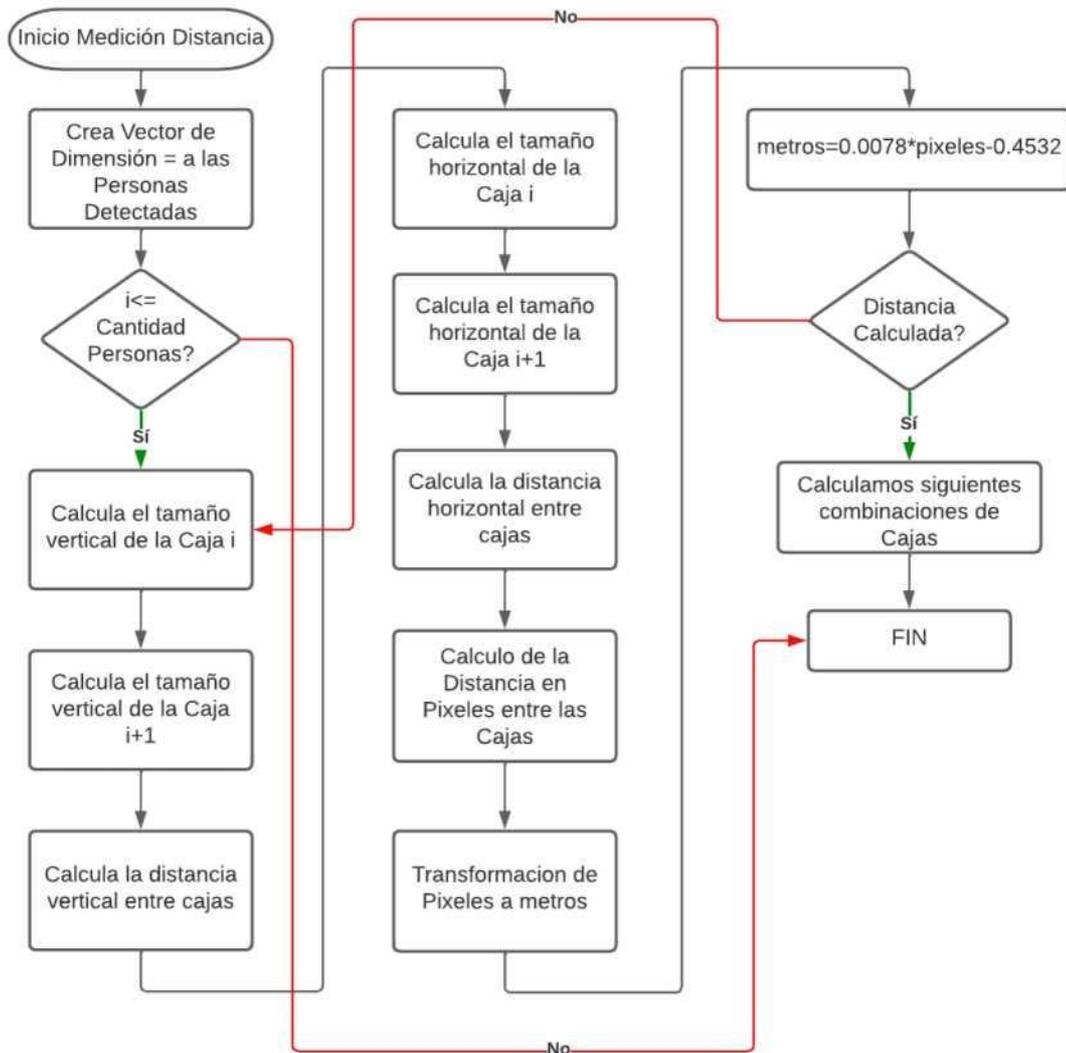


Figura 32. Diagrama de Flujo del Algoritmo para la Medición de Distancia.

3.8.3. Diagrama de Flujo Modo Offline

El modo Offline sigue la lógica que se muestra en la Figura 33, en donde se selecciona el archivo de video para obtener los datos respectivos. Si se da la acción de reproducir el video, crea el objeto que permite obtener datos y reproducir videos en pantalla, y así usar la función *Reconocimiento de Personas* seguida de *Medición de Distancia*. Se determina si el distanciamiento es apropiado, es decir la distancia es menor a 1.50 metros” Peligroso mayor a ese valor” Distanciados”. Si el distanciamiento es incumplido, un acumulador cuenta las veces que se ha activado esta condición. El algoritmo muestra el frame actual, los recuadros con las leyendas y colores de” Peligroso” Distanciados se procede a comprobar el valor actual del acumulador. Si este está entre 0 y 4 significa poco incumplimiento, si está entre 4 y 10 significa incumplimiento regular, y si es mayor que 10 significa que hay demasiado incumplimiento en la zona por lo que activa una alarma sonora que puede ser detenida con un botón controlador por el operador. El acumulador vuelve a 0 para poder seguir analizando la zona y se pueden extraer los datos en un archivo de texto para analizarlos. En la siguiente sección se detalla de mejor forma todas las funciones de este modo.

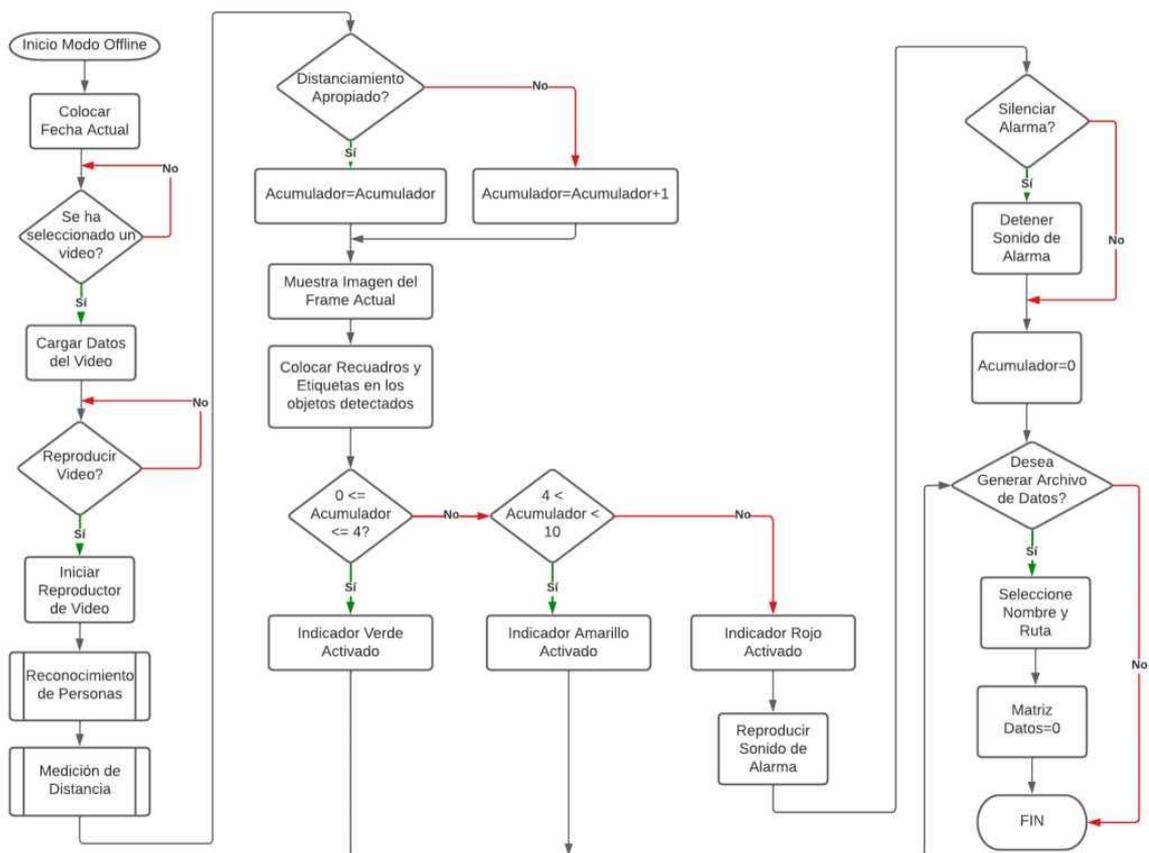


Figura 33. Diagrama de Flujo Modo Offline.

3.8.4. Diagrama de Flujo Modo Online

El modo Online sigue la misma lógica que el modo Offline, como se muestra en la Figura 34, Este permite la conexión a una cámara ip, misma que debe coincidir con la dirección colocada en el programa, en este caso *192.168.188.8:8080*. Una vez establecida la conexión se procede a la detección de personas, cálculo de distancias y validación respectiva para activar las alarmas tal y como se menciona en el modo Offline. Sin embargo, en este se puede realizar la grabación de videos para analizarlos posteriormente. Si se decide iniciar la grabación, los datos se irían almacenando en un vector hasta que se detenga la grabación y se cree un archivo con la fecha y hora actual. En la siguiente sección se hablará más a detalle de este modo del sistema.

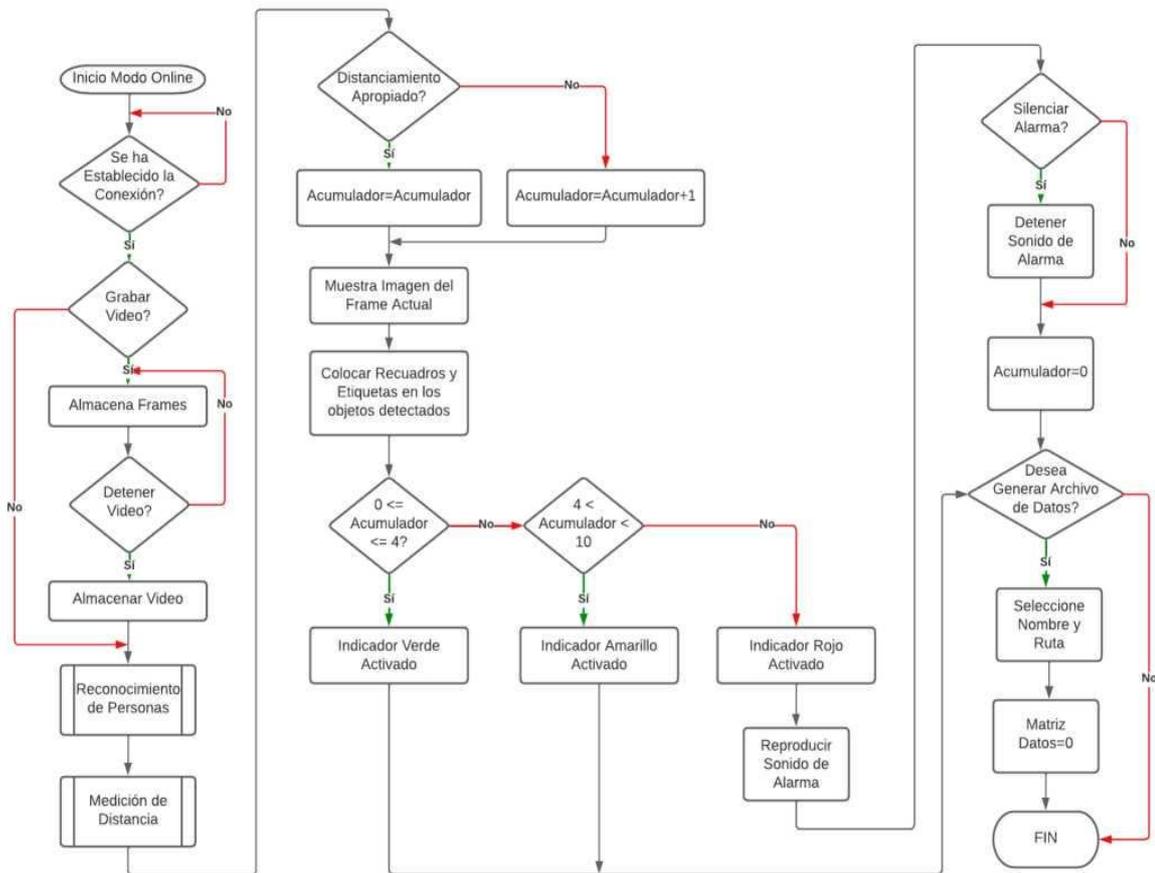


Figura 34. Diagrama de Flujo Modo Online.

3.9. Interfaz de usuario para monitoreo de aglomeración

La interfaz gráfica está compuesta de dos ventanas principales donde el usuario puede trasladarse. La primera pestaña es la del Modo OFFLINE, misma que permite observar y analizar videos pregrabados. La segunda pestaña es la del Modo ONLINE, en esta se puede establecer la conexión con un servidor de cámara IP para poder visualizar y analizar el video en tiempo real. A continuación, se presentan la descripción y guía para poder moverse dentro de cada una de las pestañas.

3.9.1. Modo OFFLINE

En el Modo OFFLINE del sistema de detección de distanciamiento se puede visualizar un video pregrabado, para poder analizar los datos en situaciones donde el operador del sistema no pueda realizarlo en tiempo real. Como se observa en la Figura 35, este modo posee un recuadro de reproducción del video, un recuadro de información de fecha, cuatro botones de acción, un recuadro de información del video cargado, un contador de alarmas y tres indicadores visuales tipo semáforo que indican las condiciones de distanciamiento que se observan.

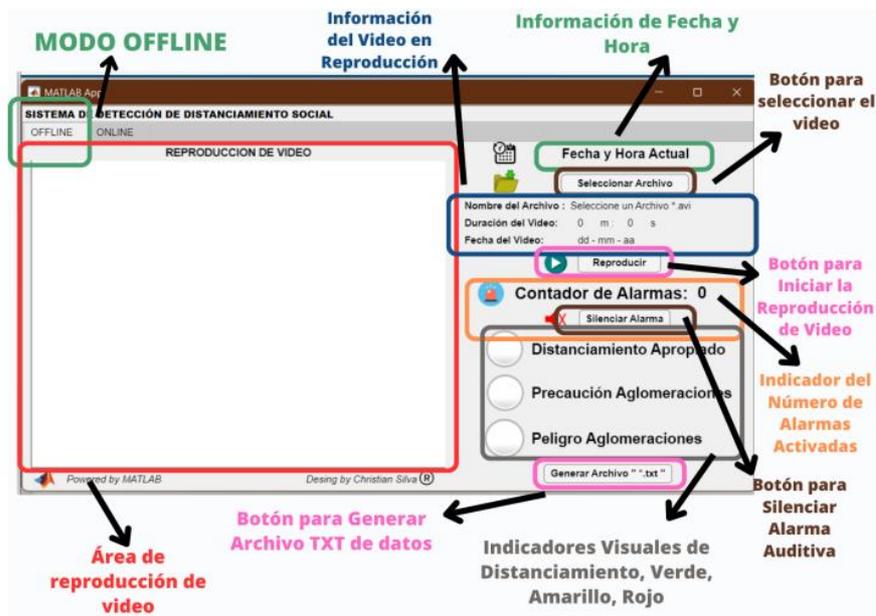


Figura 35. Componentes de Pestaña OFFLINE del sistema

Una vez observada la composición de la pestaña del modo OFFLINE, se procede a dar una explicación del funcionamiento de cada uno de los componentes, así como los pasos que se requiere para el manejo de la plataforma.

- **Paso 1:** Hacer “Clic” en el botón “Seleccionar Archivo”, el cuál desplegará una ventana de emergencia de la carpeta de videos grabados por el sistema.
- **Paso 2:** Seleccionar el Video que se requiere analizar y dar “Clic” en Abrir.

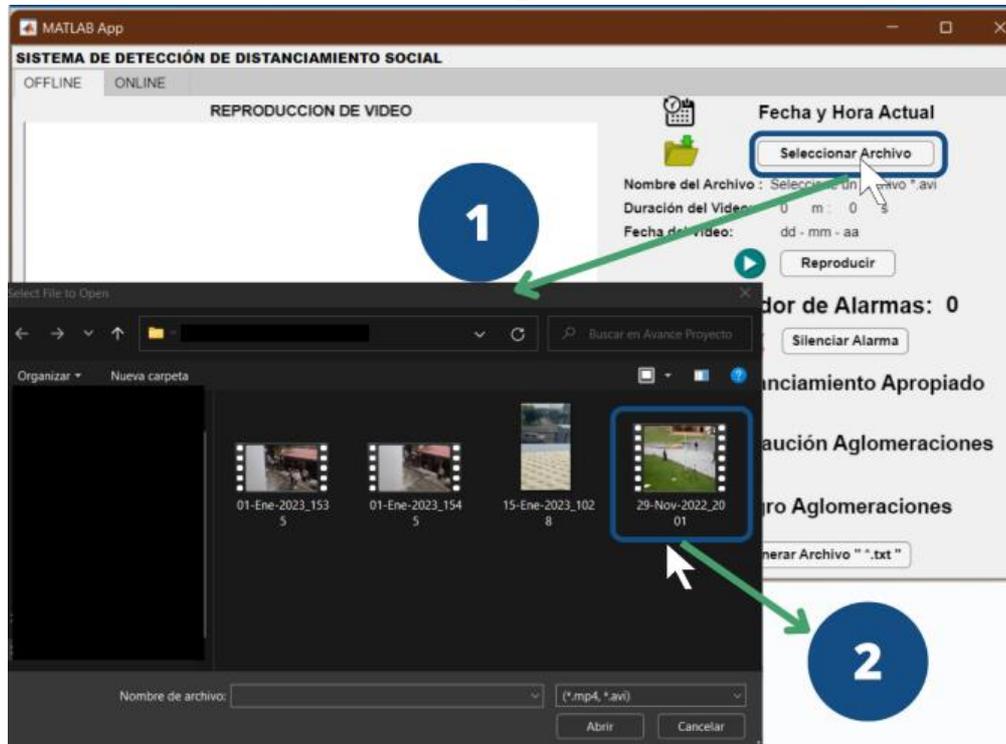


Figura 36. Selección de Archivo de video Offline

- **Paso 3:** La ventana emergente se cierra y se proceden a cargar los datos del video seleccionado, los cuales son: Nombre del Archivo, Duración del Video y Fecha del Video. Se debe tomar en cuenta que el nombre del archivo y la fecha del video están relacionados para poder extraer estos datos, mismos que son grabados de tal forma desde la ventana ONLINE.
- **Paso 4:** Dar “Clic” en el Botón de Reproducir Video para Iniciar.
- **Paso 5:** Una vez iniciada la reproducción del video seleccionado, el algoritmo de detección de distanciamiento comienza a procesar los datos, mostrando en el área de reproducción de video un recuadro de color verde y una etiqueta que dice “Distanciados”, misma que encierra a una persona cuando es detectada. Cuando se detectan varias personas, el algoritmo realiza el cálculo de la distancia entre ellos y si es menor a 1.5m, el recuadro se convierte en color rojo y cambia la etiqueta a “Peligro”.
- **Paso 6:** El indicador que representa “Distanciamiento Apropriado”, se coloca de color

Verde indicando que no existe demasiados casos de aglomeraciones en ese instante, mientras que los otros dos indicadores se colocan en color Gris indicando su inactividad.

- **Paso 7:** El contador de alarmas inicia su cuenta en “0”, y va aumentando conforme se presenten los casos de incumplimiento del distanciamiento.
- **Paso 11:** El algoritmo continuo su ejecución, identificando personas dentro del video y calculando las distancias correspondientes.
- **Paso 12:** El Contador continua la cuenta de situaciones de contagio, sin embargo, este tiene un límite, el cual reinicia al contador cuando las alarmas detectadas son mayores a “10”.

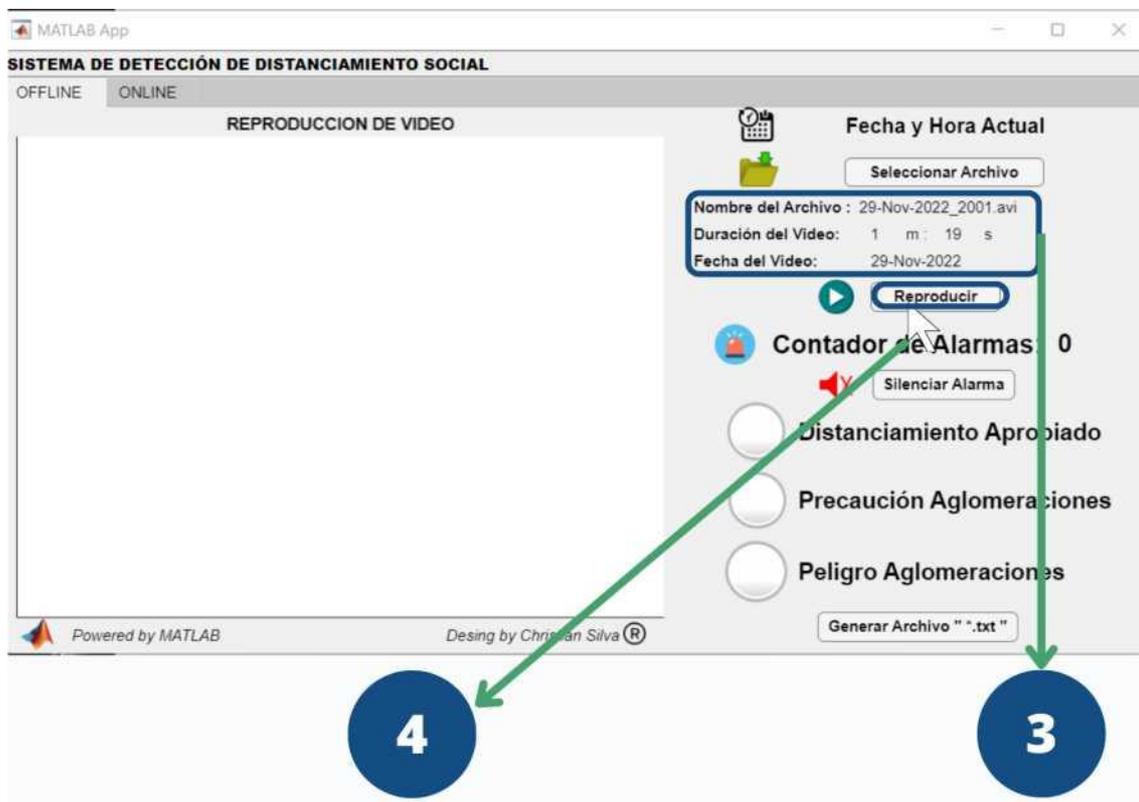


Figura 37. Información del video e inicio de reproducción

- **Paso 13:** Una vez que el Contador llega a “10” incidencias, el indicador de Precaución Aglomeraciones se coloca en color Gris, mientras que el indicador de Peligro Aglomeraciones se coloca en color Rojo indicando advertencia. Este color se mantiene hasta que el Contador de Alarmas reinicie su cuenta. Por otro lado, al llegar a la situación de Peligro superando las 10 incidencias de aglomeración se activa una alarma auditiva que va a alertar tanto al operador del sistema como a las personas que se encuentran en el

lugar donde está ubicada la cámara.

- **Paso 14:** Si existe la molestia de la alarma auditiva o está ya cumplió con su propósito de alertar a las personas, se la puede silenciar dando “Clic” en el Botón de Silenciar Alarma.
- **Paso 15:** Una vez obtenidos los datos por parte del algoritmo se puede generar un archivo de texto (.txt) para el análisis fuera de la plataforma con algún programa especializado. Para esto se da “Clic” en el Botón Generar Archivo “.txt”.
- **Paso 16:** Se procederá a abrir una ventana emergente que indica la ruta de los archivos generados por el sistema. Se requiere colocar un nombre para el archivo a generarse.
- **Paso 17:** Se da “Clic” en el Botón Guardar, y se genera el archivo correspondiente. Dentro de este archivo se pueden visualizar los datos de: fecha, tiempo, distancia en píxeles, distancia en metros, porcentaje de detección, estado (Peligro o Distanciados) y el acumulador del contador en el instante medido.

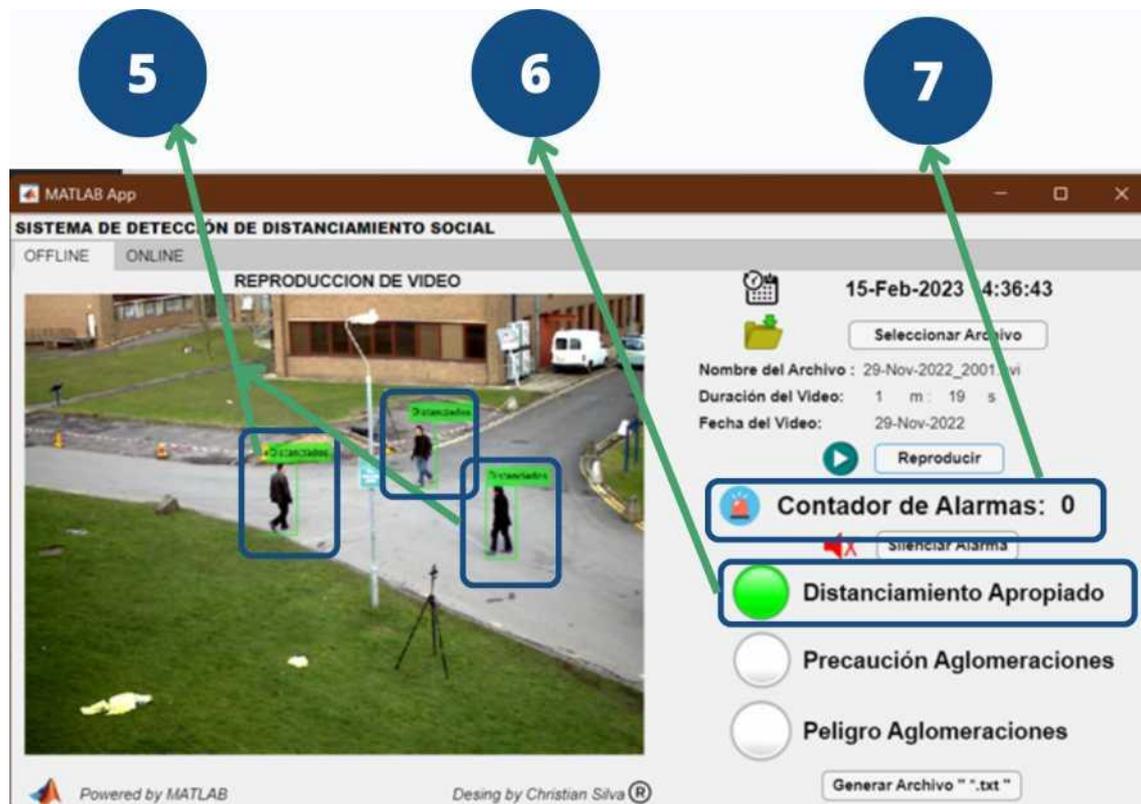


Figura 38. Inicio de reproducción de video offline

- **Paso 8:** El algoritmo continuo en ejecución y cuando este detecta incumplimiento de la distancia mínima, el cuadro que recubre la persona cambia a color ROJO y posee una

etiqueta que indica PELIGRO.

- **Paso 9:** Una vez detectado el incumplimiento, el Contador de Alarmas comienza a aumentar su valor dependiendo de la reincidencia de casos.
- **Paso 10:** Cuando el Contador de Alarmas supera las “5” incidencia de casos el indicador de Distanciamiento Apropiado cambia a color Gris, mientras que el indicador de Precaución Aglomeraciones se coloca de color Amarillo como advertencia.

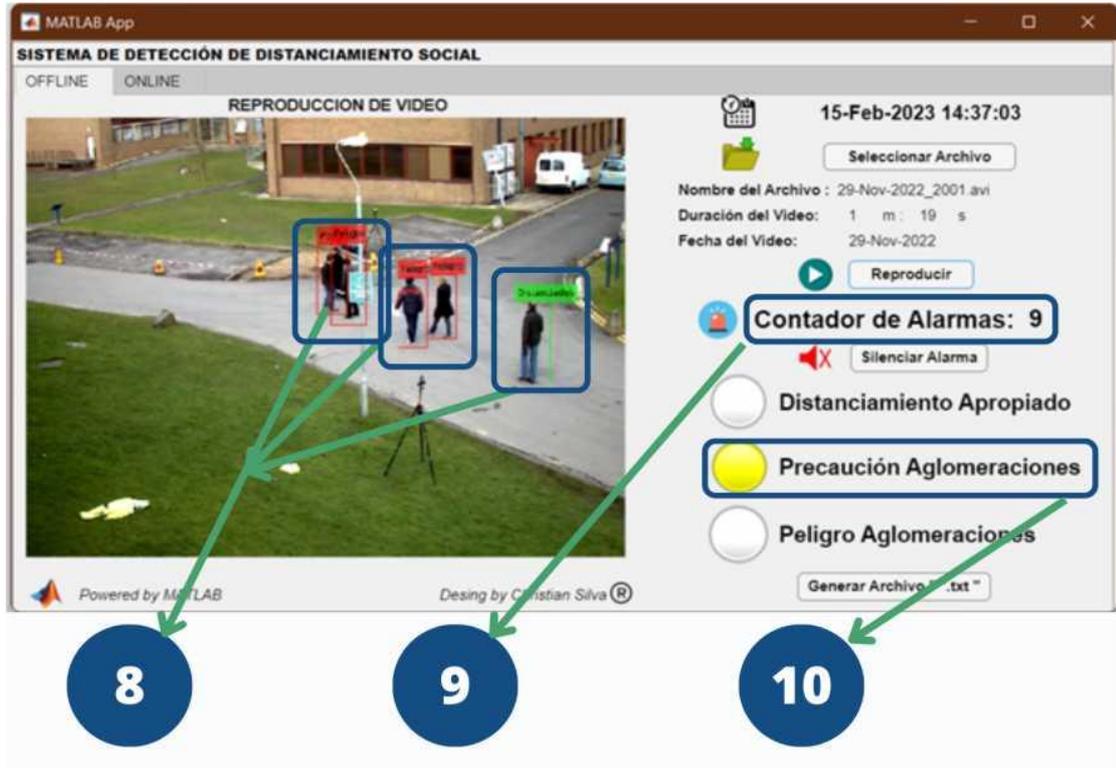


Figura 39. Acumulación de Aglomeraciones para Precaución



Figura 40. Sistema detectando Peligro de Aglomeraciones modo Offline



Figura 41. Generación de Archivo txt en Modo Offline

3.9.2. Modo ONLINE

En el Modo ONLINE del sistema de detección de distanciamiento se puede visualizar en tiempo real, las imágenes o videos transmitidos por la Cámara IP PTZ Homkeep seleccionada para este proyecto. Como se observa en la Fig. 3.34, este modo posee un recuadro de reproducción del video online, un recuadro de información de fecha, cinco botones de acción, un indicador de estado de conexión, un indicador de estado de grabación, un contador de alarmas y tres indicadores visuales tipo semáforo que indican las condiciones de distanciamiento actuales de la zona.

Una vez observada la composición de la pestaña del modo ONLINE, se procede a dar una explicación del funcionamiento y procedimientos requeridos por el usuario para comprender el manejo de la plataforma.

- **Paso 1:** Dar “Clic” en el Botón Conexión Cámara IP. Para esto debe estar encendida y conectada a la red local la Cámara IP PTZ Homkeep. Así también se debe comprobar la dirección IP de la misma y que coincida con la colocada en el programa principal. En este caso se coloca la dirección IP 192.168.188.8:8080.
- **Paso 2:** Verificar que el Indicador de Estado de Conexión se encuentre desconectado y la luz indicadora de color Rojo, que representa que se puede realizar la conexión y no hay ninguna establecida por el momento.
- **Paso 3:** Una vez Establecida la Conexión con la Cámara IP, el Indicador de Estado de Conexión cambia a CAMARA CONECTADA y la luz indicadora se cambia a color Verde, lo que indica una conexión exitosa.

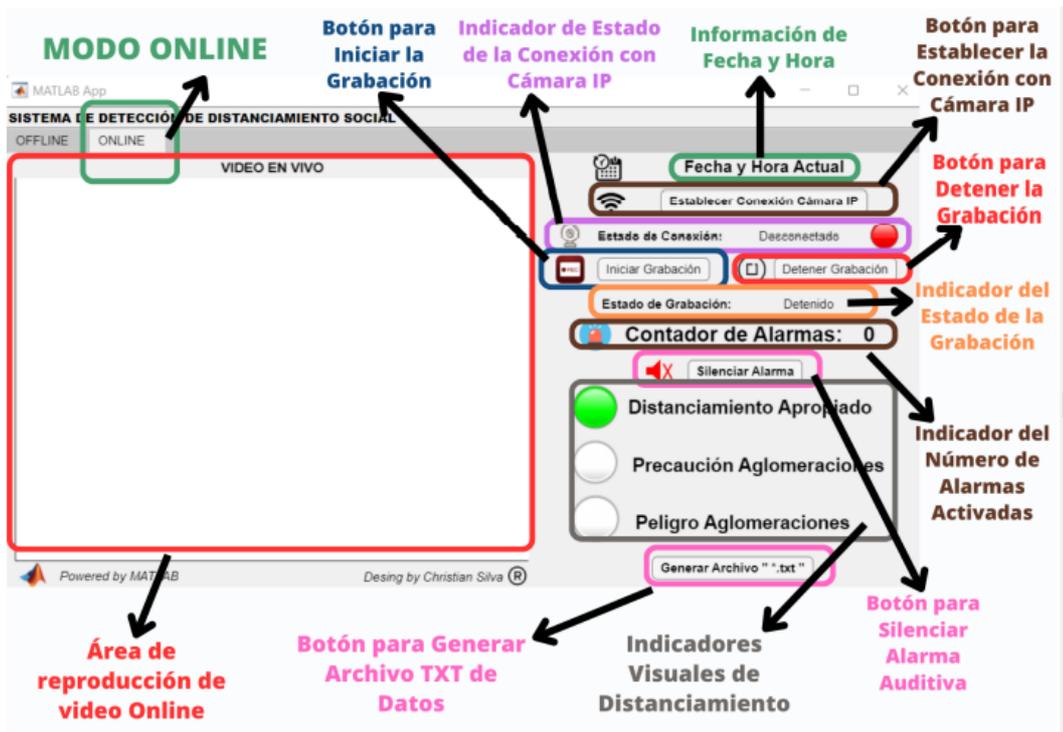


Figura 42. Componentes de la pestaña del modo ONLINE del sistema

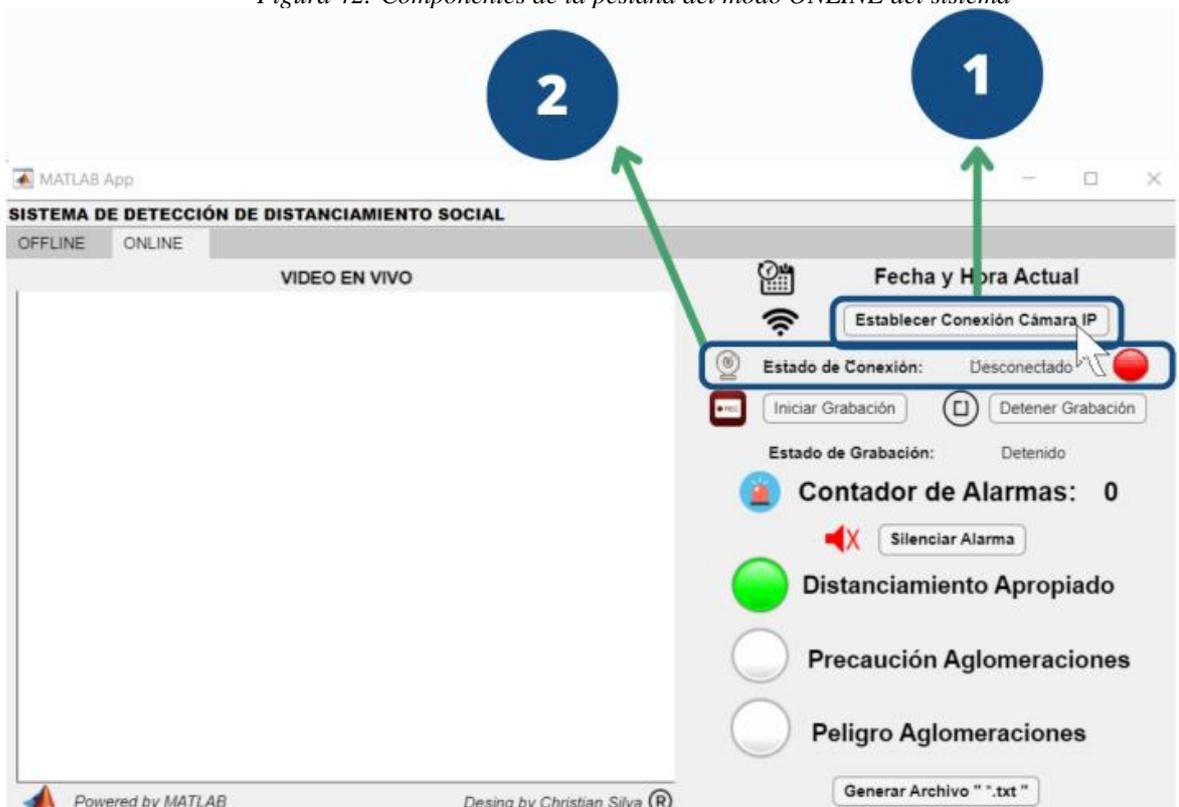


Figura 43. Establecer Conexión con Cámara IP

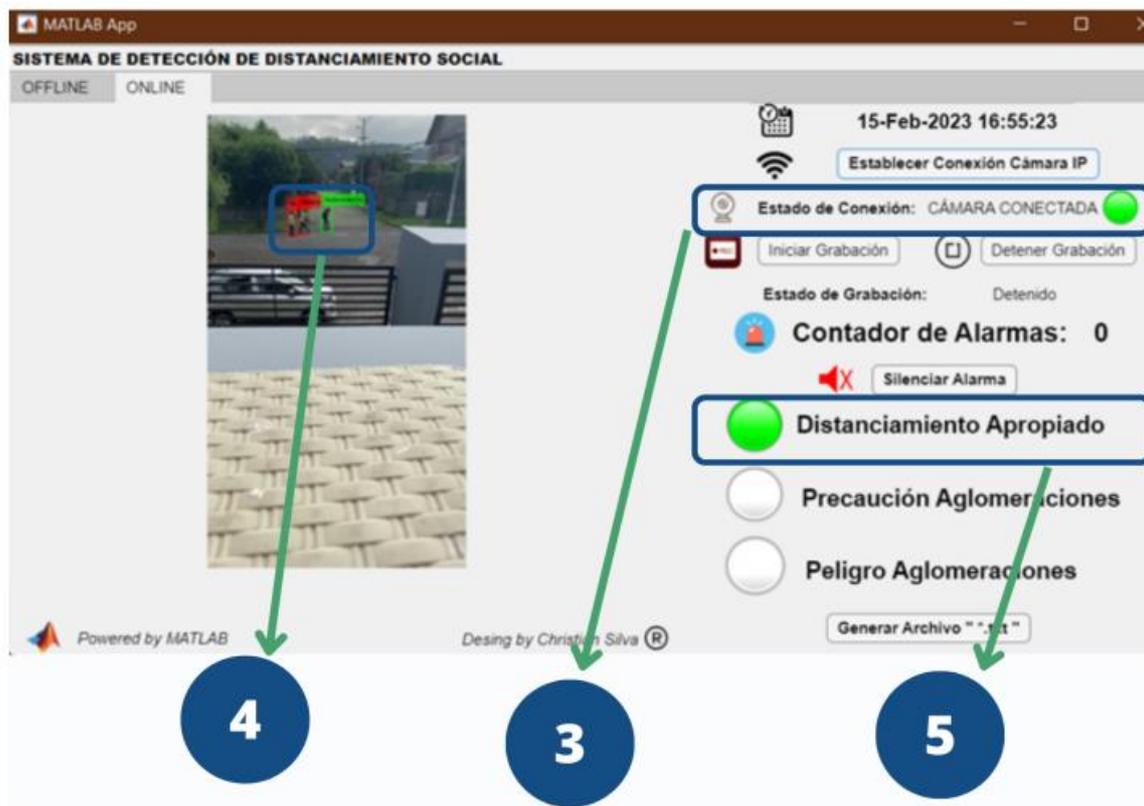


Figura 44. Sistema de Detección de distanciamiento en tiempo real inicializado

Paso 4: Dentro del área de reproducción de video ONLINE se muestran las imágenes en tiempo real de lo que captura la cámara IP, al mismo tiempo que se comienza a ejecutar el algoritmo de detección de personas. Colocando un recuadro de color Verde y una etiqueta que dice “Distanciados”, cuando la distancia es mayor de 1.5 metros; mientras que se coloca un recuadro de color Rojo y una etiqueta que dice “Peligro”, cuando la distancia es menor a 1.5 metros.

- **Paso 5:** El Contador de Alarmas se inicializa en “0” y procede a aumentar la cuenta cuando existe incumplimiento del distanciamiento. De igual forma el Indicador Visual de Distanciamiento Apropiado se coloca de color Verde mientras el contador no supere las 4 Alarmas detectadas.

- **Paso 6:** El algoritmo continúa en ejecución y cuando este detecta incumplimiento de la distancia mínima, el cuadro que recubre la persona cambia a color ROJO y posee una etiqueta que indica PELIGRO, caso contrario sigue de color VERDE con la etiqueta de DISTANCIADOS.

- **Paso 7:** Una vez detectado el incumplimiento por el sistema, el Contador de Alarmas comienza a aumentar su valor dependiendo de la reincidencia de casos.
- **Paso 8:** Cuando el Contador de Alarmas supera las “5” incidencia de casos el indicador de Distanciamiento Apropiado cambia a color Gris, mientras que el indicador de Precaución Aglomeraciones se coloca de color Amarillo como advertencia de la situación en la zona.
- **Paso 9:** El algoritmo continuo en ejecución y cuando este detecta incumplimiento de la distancia mínima.

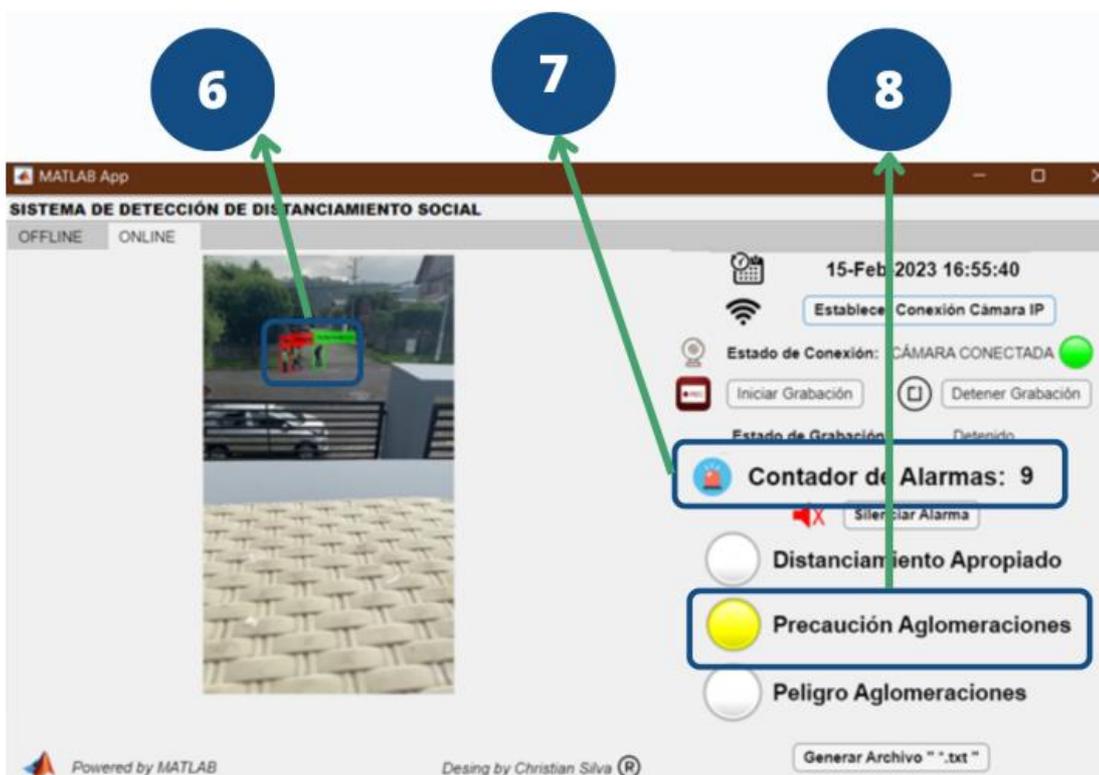


Figura 45. Precaución de Aglomeraciones en modo Online

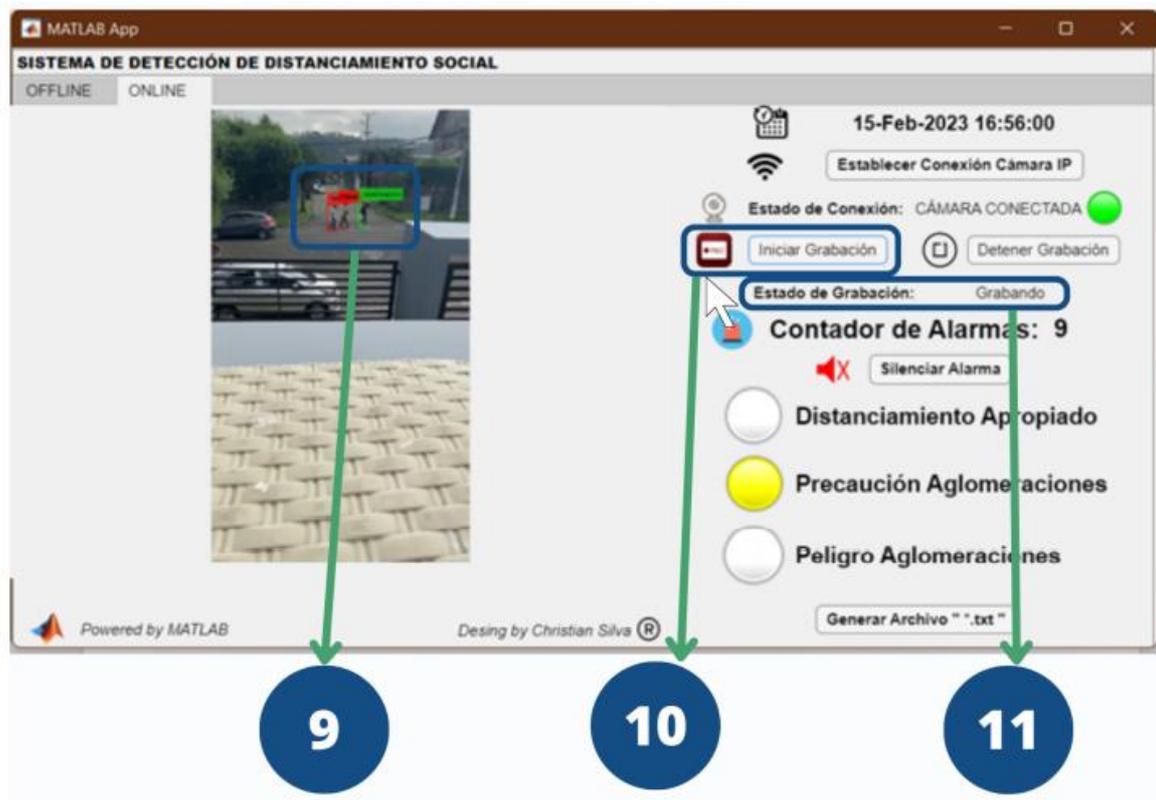


Figura 46. Inicio de Grabación de Video modo Online

- **Paso 10:** Si se requiere la grabación del video que se está observando en ese instante, se puede grabar dando “Clic” en el Botón de Iniciar Grabación.
- **Paso 11:** Se verifica que el Indicador de Grabación haya cambiado su estado a “Grabando” después de dar “Clic” al Botón Detener Grabación. Este detiene hasta ese instante la grabación y guarda el archivo de video en formato avi dentro de una carpeta donde se almacenan todas las grabaciones realizadas por el usuario.

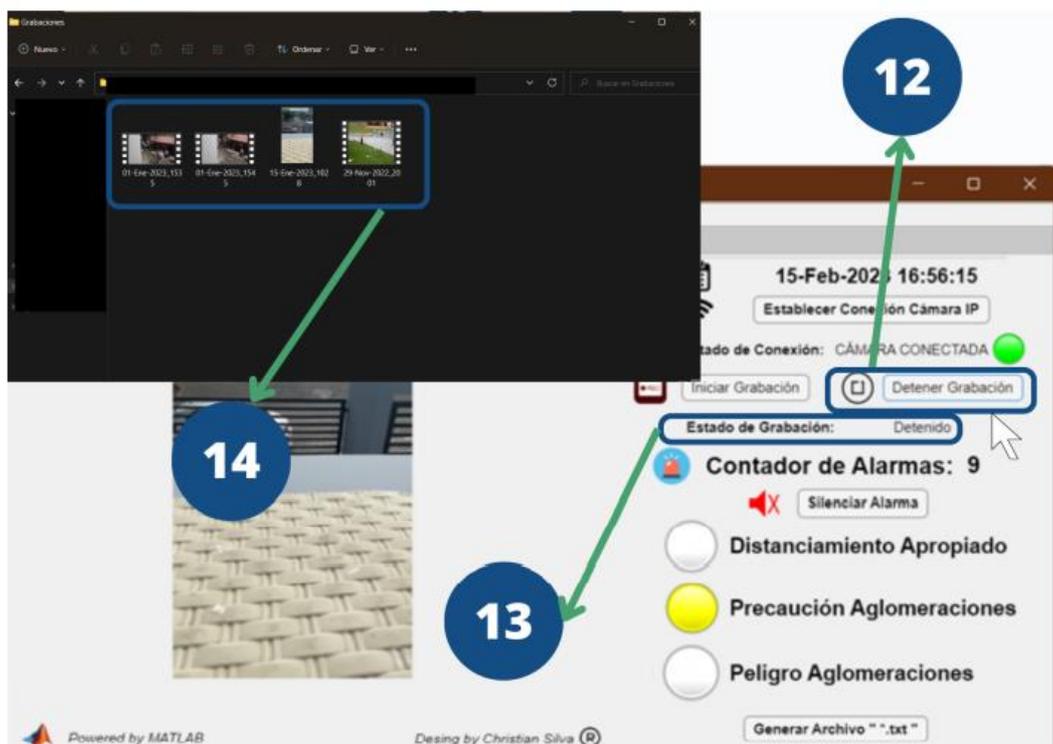


Figura 47. Detención y Generación de archivo de Video Grabado modo Online

- **Paso 13:** Se verifica el Indicador de Estado de Grabación el cual cambia a Detenido.
- **Paso 14:** El archivo se guarda dentro de una carpeta específica que contiene los archivos grabados con anterioridad. El nombre del archivo contiene la fecha y hora en la que se realizó la grabación para poder usar estos datos en el modo Offline.
- **Paso 15:** El Contador continua la cuenta de situaciones de contagio, sin embargo, este tiene un límite, el cual reinicia al contador cuando las alarmas detectadas son mayores a “10”.
- **Paso 16:** Una vez que el Contador supera las “10” incidencias, el indicador de Precaución Aglomeraciones se coloca en color Gris, mientras que el indicador de Peligro Aglomeraciones se coloca en color Rojo indicando una situación de Contagio. Este color se mantiene hasta que el Contador de Alarmas reinicie su cuenta. Por otro lado, al llegar a la situación de Peligro superando las 10 incidencias de aglomeración se activa una alarma auditiva que alerta tanto al operador del sistema como a las personas que se encuentran en el lugar donde está ubicada la cámara.



Figura 48. Peligro de Aglomeraciones en modo Online

- **Paso 17:** Si existe molestia de la alarma auditiva o, está ya cumplió con su propósito de alertar a las personas, se la puede silenciar dando “Clic” en el Botón de Silenciar Alarma.
- **Paso 18:** Una vez obtenidos los datos por parte del algoritmo se puede generar un archivo de texto (.txt) para el análisis. Para esto se da “Clic” en el Botón Generar Archivo “.txt”.
- **Paso 19:** Se procede a abrir una ventana emergente que indica la ruta de los archivos generados por el sistema. Se requiere colocar un nombre para el archivo a generarse. También se debe verificar que la extensión del archivo corresponda a “.txt”.
- **Paso 20:** Se da “Clic” en el Botón Guardar, y se genera el archivo correspondiente.

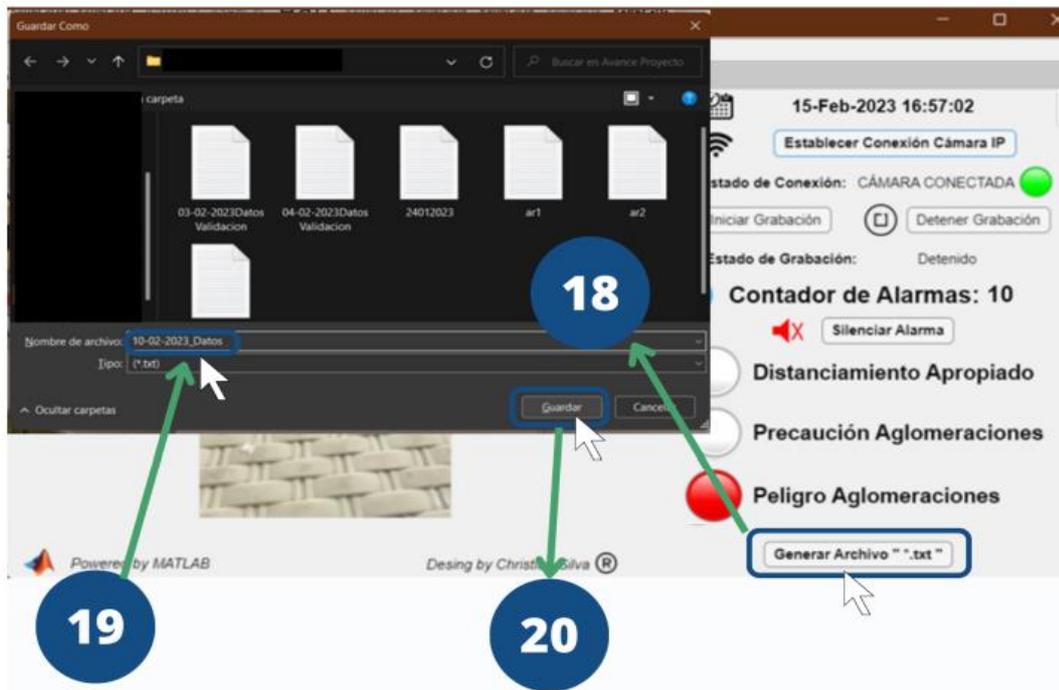


Figura 49. Generación de archivo txt en modo Online

- Dentro de este archivo se pueden visualizar los datos de: fecha, tiempo, distancia en píxeles, distancia en metros, porcentaje de detección, estado (Peligro o Distanciados) y el acumulador del contador en el instante medida

Capítulo 4

Resultados y discusión.

Para poder cubrir los 400 metros que abarcan la Av. Confraternidad, se realizaron Pruebas en donde se identificaron los puntos de visión de la cámara, observando que se tiene una cobertura de aproximadamente 20 metros con una sola cámara. Como se observa en la Figura 50, se requiere una cantidad de 20 cámaras para lograr cubrir el área total de la Zona Comercial.



Figura 50. Ubicación de las Cámaras a lo largo de la Zona Comercial de el Tambo.

Finalmente, para analizar los datos de distancias obtenidos se utilizan herramientas de software especializado que analizan los datos generados por el sistema a partir de un archivo de texto. Los datos para analizar son principalmente la distancia calculada entre personas y la distancia entre la cámara y el objetivo detectado.

4.1. Resultados Fiabilidad.

Para la comprobación de los resultados se utiliza el software IBM SPSS Statistics, mismo que permite obtener datos estadísticos de las pruebas realizadas para comprobar la fiabilidad del sistema. En este caso se realizar varias pruebas a diferentes distancias para poder calcular el error de la medida del sistema. Para esto se tomaron medidas de 1 y 2 metros entre personas, mientras las distancias entre la cámara y las personas son de 5, 10, 15, 20 metros. A continuación, se presentan los resultados obtenidos para cada escenario.

4.1.1. Escenario de Prueba 1

En esta prueba se realizaron diferentes mediciones del valor que se obtiene del sistema al medir una distancia fija de 1 metro entre dos objetivos. Se realizaron pruebas a distancias de cámara de 5, 10, 15, 20 metros y se calculó el error obtenido para cada caso como se observa en la Tabla 9.

Tabla 9. Datos Estadísticos del Error Obtenido en el Escenario de Prueba 1.

	N	Media	Desviación típica	Error típico	Intervalo de confianza para la media al 95 %			Máximo
					Límite inferior	Límite superior	Mínimo	
5 m	10	4,8817	3,59161	1,13577	2,3124	7,4510	1,07	11,74
10 m	10	3,8498	2,67687	,84650	1,9349	5,7647	,43	7,17
15 m	10	3,5290	2,51907	,79660	1,7270	5,3310	,51	7,53
20 m	10	12,6181	3,44212	1,08849	10,1558	15,0804	7,84	18,10
Total	40	6,2197	4,80436	,75964	4,6831	7,7562	,43	18,10

En la Figura 51, se puede observar el diagrama de caja de los datos de error obtenidos en las mediciones de este escenario de prueba, en donde se puede verificar que existe una media del porcentaje de error menor al 5% para distancias de cámara de 5, 10 y 15 metros. Sin embargo, para una distancia de 20 metros se obtuvo un porcentaje de error de 13 % con valores máximos que llegan al 19%.

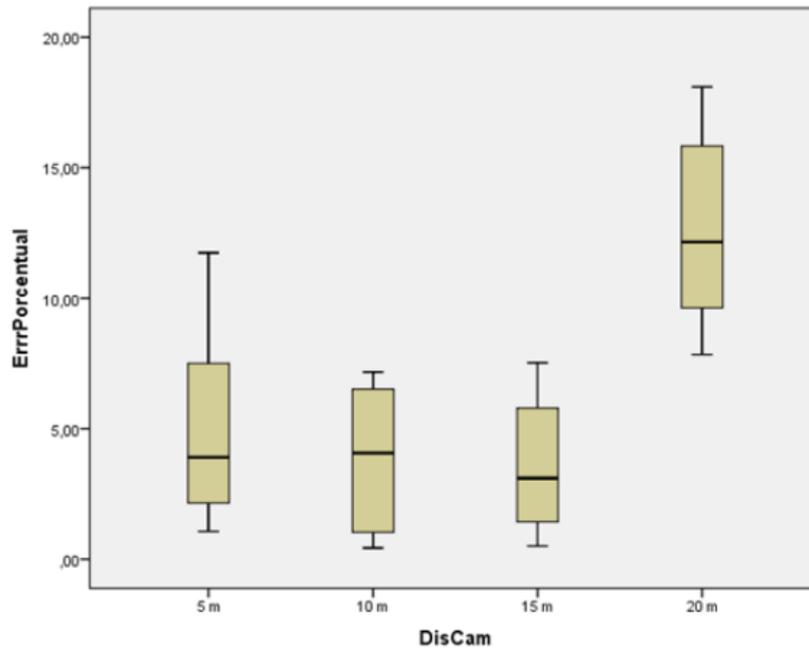


Figura 51. Diagrama de Caja del Error en el Escenario de Prueba 1.

4.1.2. Escenario de Prueba

En esta prueba se realizaron diferentes mediciones del valor que se obtiene del sistema al medir una distancia fija de 2 metros entre dos objetivos. Se realizaron pruebas a distancias de cámara de 5, 10, 15, 20 metros y se calculó el error obtenido para cada caso como se observa en el Tabla 10.

Tabla 10. Datos Estadísticos del Error Obtenido en el Escenario de Prueba 2.

	N	Media	Desviación típica	Error típico	Intervalo de confianza		Mínimo	Máximo
					para la Límite inferior	media al 95 % Límite superior		
5 m	10	4,5040	,84587	,26749	3,8989	5,1091	3,15	5,48
10 m	10	3,3720	,93114	,29445	2,7059	4,0381	2,36	4,71
15 m	10	3,8175	,62885	,19886	3,3676	4,2674	3,08	4,81
20 m	10	20,2105	12,27021	3,88018	11,4329	28,9881	4,25	36,30
Total	40	7,9760	9,30274	1,47089	5,0008	10,9512	2,36	36,30

En la Figura 52 se puede observar el diagrama de caja de los datos de error obtenidos en las mediciones de este escenario de prueba, en donde se puede verificar que existe una media del porcentaje de error menor al 5 % con valores máximos cercanos al 8 % para distancias de cámara de 5, 10 y 15 metros. Sin embargo, para una distancia de 20 metros se

obtuvo un porcentaje medio de error del 17% con valores máximos que sobrepasan el 30 %.

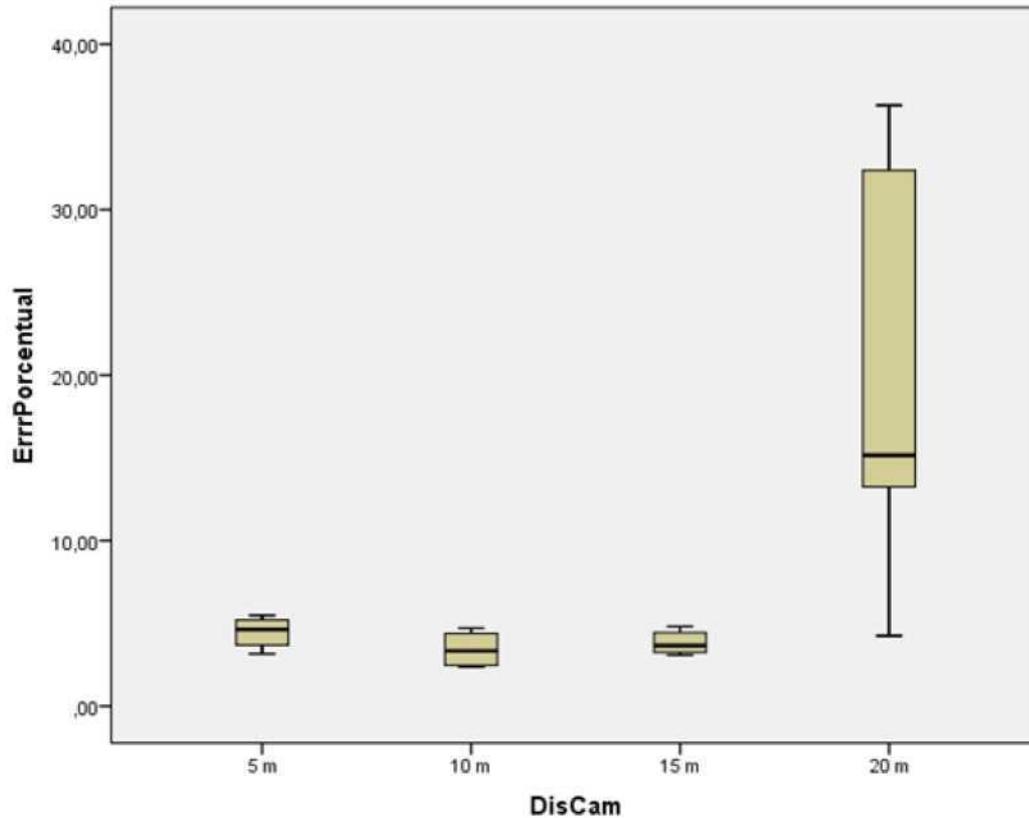


Figura 52. Diagrama de Caja del Error en el Escenario de Prueba 2.

4.2. Comprobación de Hipótesis.

Para la validación de la hipótesis se utiliza la prueba estadística de ANOVA, misma que se realiza en el software IBM SPSS Statistics. Esta prueba permite evaluar los datos obtenidos sobre observaciones múltiples, perfecto para comparar las medias obtenidas, en este caso la lectura de distancia del sistema y la distancia entre la cámara y el objetivo.

4.2.1. Escenario de Prueba 1 Para esta prueba de hipótesis se realizaron 10 mediciones a diferentes distancias de la cámara con respecto a los objetivos (personas) que se encuentran a una distancia fija de **1 metro**. Las distancias de la cámara a estos objetivos son: 5, 10, 15, y 20 metros.

Hipótesis Nula: H_0

$H_0: \mu_1 = \mu_2 = \mu_3 = \mu_4$

Donde:

μ_1 media de las distancias leídas por la cámara a una distancia de 5m.

μ_2 media de las distancias leídas por la cámara a una distancia de 10m.

μ_3 media de las distancias leídas por la cámara a una distancia de 15m.

μ_4 media de las distancias luidas por la cámara a una distancia de 20m.

Hipótesis Alternativa: H_1

H_1 : Existe i, j tal que $\mu_i \neq \mu_j$ para $i, j = 1, 2, 3, 4$

Tabla 11. Datos Estadísticos de las pruebas realizadas para el Escenario de Prueba 1.

Distancia con Respecto a la Cámara	N	Media	Desviación típica	Error típico	Intervalo confianza		Mínimo	Máximo
					para la m Límite	Media al 95 % Límite		
5 m	10	,9577	,04412	,01395	,9261	,9892	,88	1,02
10 m	10	1,0360	,03044	,00962	1,0142	1,0577	,99	1,07
15 m	10	1,0271	,03470	,01097	1,0022	1,0519	,98	1,08
20 m	10	,8738	,03442	,01088	,8492	,8984	,82	,92
Total	40	,9736	,07458	,01179	,9498	,9975	,82	1,08

Los datos estadísticos obtenidos de las pruebas realizadas a las personas con un distanciamiento de 1 metro se pueden observar en el Tabla 11 y son representados mediante el diagrama de caja que se observa en la Figura 53, En este se puede identificar que existen dispersiones de datos similares en distancias de 10 y 15 metros con un rango que está alrededor de 0.97 a 1.07 metros. Para valores de 20 metros, se observa que la mayoría de los valores se concentran en un rango de 0.82 a 0.92 metros. Finalmente se observa que para una distancia de 5 metros se tienen valores dispersos entre 0.88 y 1.03.

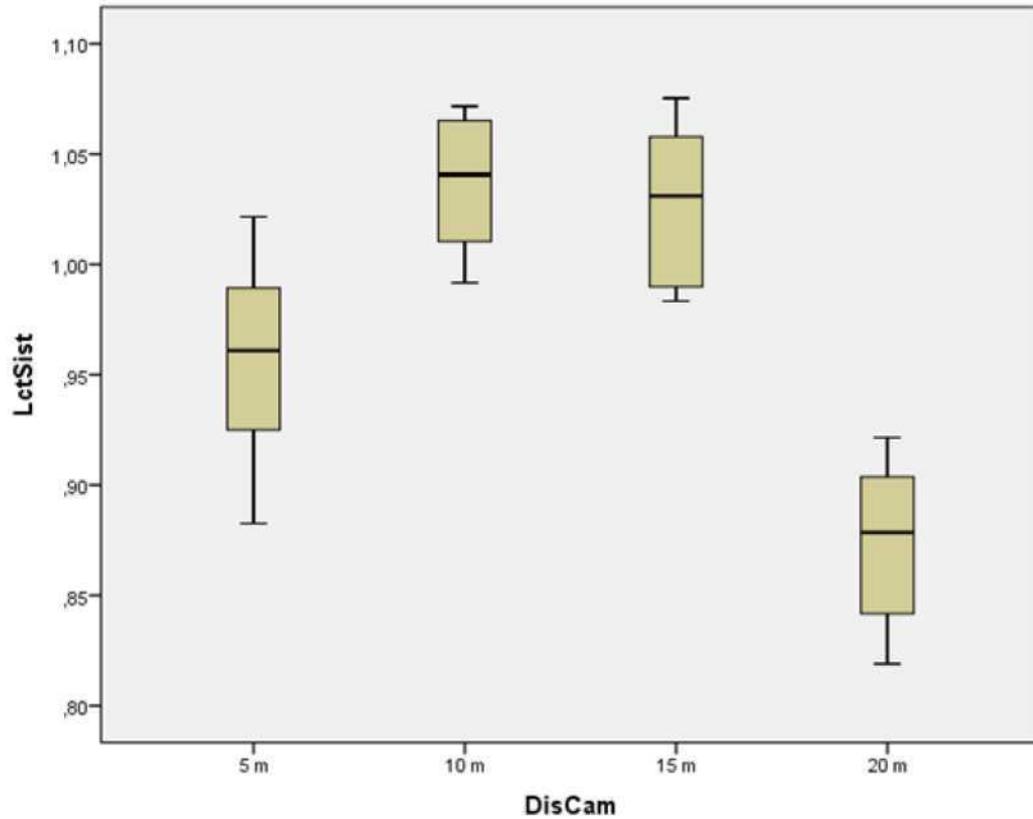


Figura 53. Diagrama de Caja de los Datos del Escenario de Prueba 1.

Como se observa en la Tabla 12 el **P valor** obtenido es de 0.000 por lo que cumple con la condición: $Pvalor = 0,000 < 0,05$, es decir que significativamente las medias no son iguales y se rechaza la hipótesis H_0 . Esto se produce debido a que existe una dispersión demasiado elevada con respecto a las mediciones sobre todo en una distancia de 20 metros debido a que se llega al límite de la zona de detección y el algoritmo comienza a cometer una mayor cantidad de errores en esa zona.

4.2.2. Escenario de Prueba 2

Para esta prueba de Hipótesis se realizaron 10 mediciones a diferentes distancias de la cámara con respecto a los objetivos (personas) que se encuentran a una distancia fija de **2 metros**. Las distancias de la cámara a estos objetivos son: 5, 10, 15, y 20 metros.

Hipótesis Nula: H_0

$$H_0: \mu_1 = \mu_2 = \mu_3 = \mu_4$$

Donde:

μ_1 media de las distancias leídas por la cámara a una distancia de 5m.

μ_2 media de las distancias leídas por la cámara a una distancia de 10m.

μ_3 media de las distancias leídas por la cámara a una distancia de 15m.

μ_4 media de las distancias leídas por la cámara a una distancia de 20m.

Hipótesis Alternativa: H_1

H_1 : Existe i, j tal que $\hat{\mu}_i = p, j$ para $i, j = 1, 2, 3, 4$

Los datos estadísticos obtenidos de las pruebas realizadas a las personas con un distanciamiento

Tabla 12. ANOVA para el Escenario de Prueba 1.

	Suma de cuadrados	Gl	Media Cuadrática	F	Sig.
Entre Grupos	,170	3	,057	42,974	,000
Dentro de Grupos	,047	36	,001		
Total	,217	39			

de 2 metros se pueden observar en la Tabla 12 y son representados mediante el diagrama de caja que se observa en la Figura 54, En este se puede identificar que existen dispersiones de datos similares en distancias de 5, 10 y 15 metros con un rango que está alrededor de 1.85 a 1.95 metros. Para valores de distancia de 20 metros, se observa que la mayoría de los valores se concentran en el rango de 2.10 hasta los 2.70, con una media de 2.30.

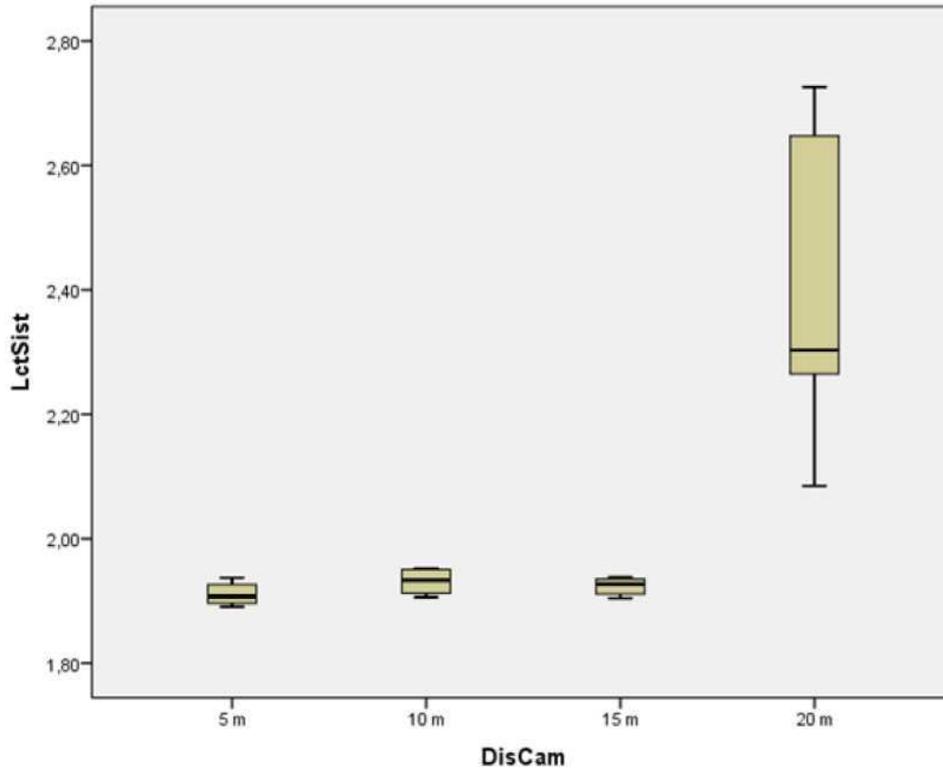


Figura 54. Diagrama de Caja de los Datos del Escenario de Prueba 2.

Como se observa en la Tabla 13 el **Pvalor** obtenido es de 0,000 por lo que cumple con la condición: $P \text{ valor} = 0,000 < 0,05$, es decir que significativamente las medias no son iguales y se rechaza la hipótesis **H0**. Esto se produce debido a que existe una dispersión demasiado elevada con respecto a las mediciones sobre todo en una distancia de 20 metros debido a que se llega al límite de la zona de detección y el algoritmo comienza a cometer una mayor cantidad de errores con valores muy altos con respecto al valor real.

Tabla 13. ANOVA para el Escenario de Prueba 2.

	Suma de cuadrados	Gl	Media Cuadratica	F	Sig.
Entre Grupos	1,746	3	,582	38,160	,000
Dentro de Grupos	,549	36	,015		
Total	2,295	39			

4.3. Resultado de Agrupación

4.3.1. Escenario de Prueba 1

Anteriormente se realizó el estudio ANOVA en el cual se obtuvo un valor con una significancia de $0,000 < 0,05$, entonces rechazamos H_0 y se toma la hipótesis alternativa H_i y mediante ese resultado podemos realizar un estudio posterior y así podemos saber con una clasificación para saber que grupos de las medidas serán diferentes. Como podemos observar en la Tabla 14.

Tabla 14. Análisis de grupos post hoc para el Escenario 1

DisCam	N	Subconjunto para alfa = 0,05	
		1	2
15 m	10	3,5290	
10 m	10	3,8498	
5 m	10	4,8817	
20 m	10		12,6181
sig.		,763	1,000

De la misma manera se puede visualizar la agrupación de las medias en el Figura 55, de las cuales se tomaron en cuenta las medias obtenidas anterior mente del análisis de ANOVA.

En la Figura 55, se puede observar que se agruparon las medias de las medidas tomadas de cámara hacia el objetivo de 5m, 10m, 15m en el grupo 1. Así mismo para el grupo 2 solo están las tomas de la distancia de 20m, claramente se visualiza una diferencia mayor debido a que el sistema ya tienen dificultad con la resolución de imagen que toma a esa distancia.

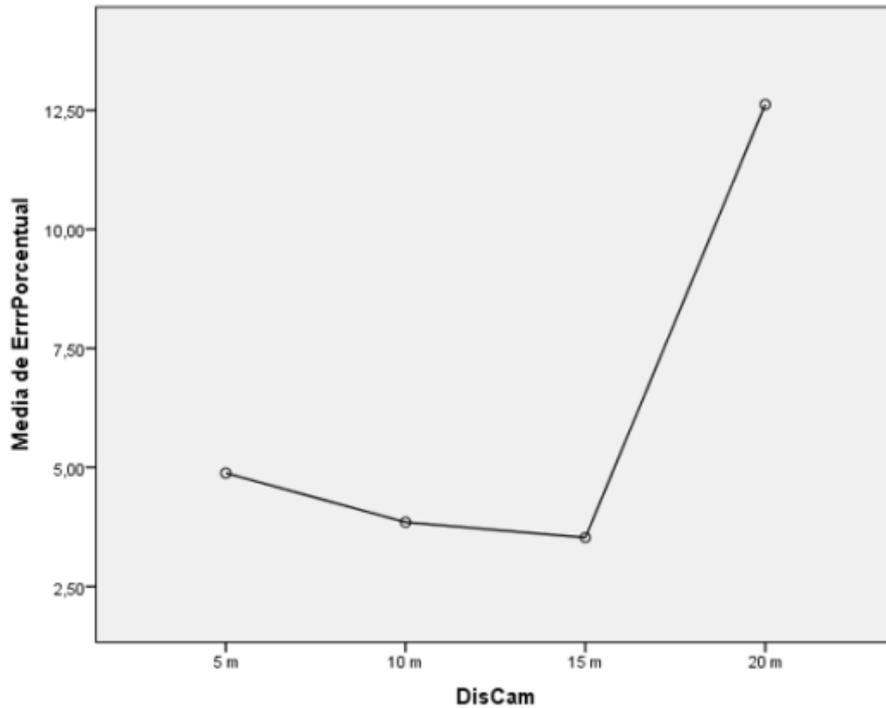


Figura 55. Agrupación de medias.

4.3.2. Escenario de Prueba 2

Anteriormente se realizó el estudio ANOVA en el cual se obtuvo un valor con una significancia de $0,000 < 0,05$, entonces rechazamos H_0 y se toma la hipótesis alternativa H_i y mediante ese resultado podemos realizar un estudio posterior y así podemos saber con una clasificación para saber que grupos de las medidas serían diferentes. Como podemos observar en el Tabla 15.

Tabla 15. Análisis de grupos post hoc para el Escenario 2

DisCam	N	Subconjunto para alfa = 0,05	
		1	2
15 m	10	3,3720	
10 m	10	3,8175	
5 m	10	4,5040	
20 m	10		20,2105
sig.		,976	1,000

De la misma manera se puede visualizar la agrupación de las medias en el Figura 56, de las cuales se tomaron en cuenta las medias obtenidas anterior mente del análisis de ANOVA.

Las distancias de 5m, 10m, 15m, pertenecen al grupo 1 siendo estas con una similitud en las medias, mientras que la medida de 20m tienen una diferencia notoria como se observa en la Figura 56, esto se debe a que la resolución de la imagen analizada dentro del sistema de detección, se distorsiona y no se tiene una buena apreciación de las personas y esto produce que el porcentaje de error crezca con relación a las anteriores.

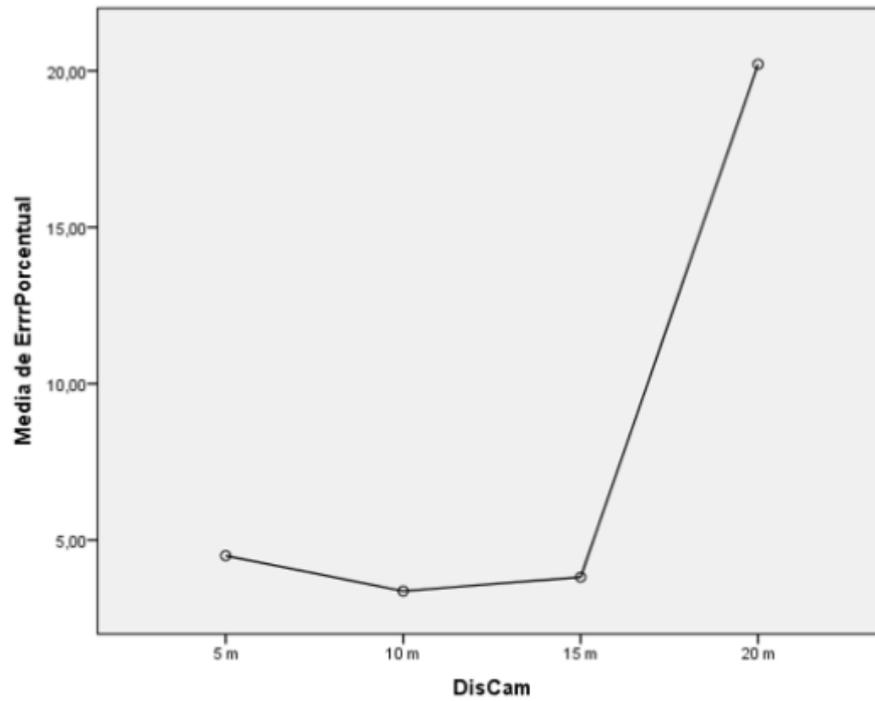


Figura 56. Agrupación de medias

4.4. Resultados en Campo.

Una vez determinada la fiabilidad del sistema con el análisis estadístico realizado en las secciones anteriores. Se procede al despliegue del prototipo en la Zona Comercial del Tambo en Pelileo. Se colocó la cámara a una altura de 3 metros y se procedió a capturar y analizar los videos para determinar el comportamiento en campo del sistema.

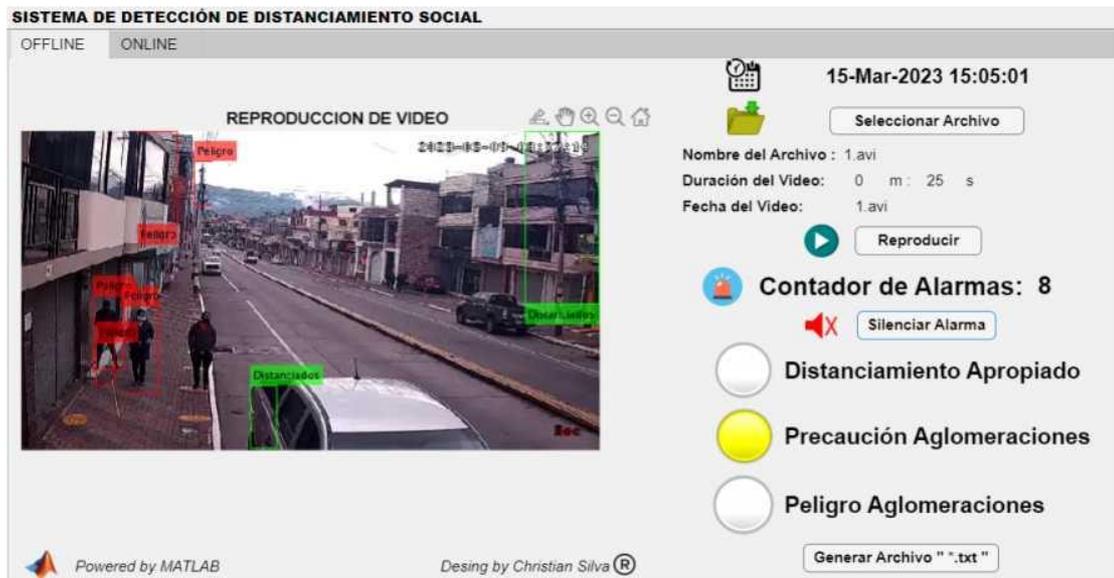


Figura 57. Prueba 1 de Sistema Desplegado.

En la Figura 57, se puede observar el sistema en funcionamiento, el cual muestra la zona comercial en donde se ubicó el sistema. En esta imagen de prueba se observa la presencia de cuatro falsos positivos, un falso negativo y dos personas detectadas correctamente. Los reflejos en los vehículos y de las ventanas provocan que existan estas fallas sin haber personas al rededor.



Figura 58. Prueba 2 de Sistema Desplegado.

En la Figura 58, se puede observar el sistema detectando siete objetos. Tres son personas detectadas correctamente, sin embargo, al igual que el caso anterior se tienen cuatro falsos positivos que se produce por los reflejos que se encuentran en la zona. Se puede observar también que la distancia entre objetivos indica que hay un correcto distanciamiento ya que estos objetos se encuentran a una distancia de 1 metro.



Figura 59. Prueba 3 de Sistema Desplegado.

En la Figura 59, se puede observar el sistema detectando seis objetivos, los cuales están tres están detectados de manera correcta. Se puede observar también que la distancia entre los objetivos de atrás que se encuentran juntos tiene el indicador de Distanciados, esto debido a que la cámara se encuentra llegando a su zona máxima de detección.



Figura 60. Prueba 4 de Sistema Desplegado.

En la Figura 60, se puede observar el sistema detectando cinco objetivos, los cuales tres están detectados de manera correcta, sin embargo, se tiene dos falsos positivos, debido a los reflejos generados por los vidrios. Se puede observar también que la distancia entre los objetivos de atrás que se encuentran juntos tiene el indicador de Peligro.



Figura 61. Prueba 5 de Sistema Desplegado.

En la Figura 61, se puede observar el sistema detectando siete objetivos, los cuales están detectados tres están detectados de manera correcta. Se tiene un falso negativo y tres falsos positivos, ya que la detección de una de las personas que se encuentra junto al vehículo no se realizó, y se han colocado cajas adicionales que no corresponden a ningún objeto válido. Se puede observar también que la distancia entre los objetivos posee el indicador de Peligro, ya que se observa que la distancia entre ellos es menor a 1.5 metros

Capítulo 5

Conclusiones y Recomendaciones

5.1. Conclusiones

Se puede afirmar con la realización del presente trabajo de investigación que se puede realizar un sistema de detección del distanciamiento entre personas a bajo costo y que puede ser implementado en varios lugares donde exista alta afluencia de personas. Los conocimientos técnicos aplicados en el diseño del sistema son coherentes con lo estudiado durante diversas ramas de la carrera de Electrónica y Telecomunicaciones.

Como se ha podido observar en el desarrollo de esta investigación se ha logrado obtener un sistema fiable para poder detectar la presencia de personas, sino que también permite el cálculo de la distancia entre estos objetivos utilizando el software Matlab como principal herramienta para cumplir con el objetivo principal. Por lo que, una vez cumplidos todos los objetivos de estudio del proyecto se puede concluir lo siguiente:

- Se estudió cada uno de los algoritmos de inteligencia artificial existentes que permiten la detección de personas, y se estableció que el más efectivo para el diseño del sistema en base a las pruebas preliminares es el algoritmo ACF de CALTECH.
- Se logró la estimación de la distancia entre objetivos, utilizando cálculos matemáticos básicos que permitieron conseguir al sistema un porcentaje de error promedio del 12 %, el cual consideramos aceptable debido a que no existen prototipos o estudios previos que indique una base de error mínima para este tipo de proyectos.
- En las pruebas realizadas, se obtuvieron mejores resultados y más estables para distancias de cámara a objetivo de 5, 10 y 15 metros, las cuales se encuentran en valores menores del 10 % por lo que se afirma que el sistema puede detectar de una mejor manera cuando los objetivos se encuentran más lejos, esto se debe a las imágenes de entrenamiento de la base de datos CALTECH que son de menor tamaño.
- Se afirma que el sistema es fiable en la mayoría de los casos debido a la comprobación de hipótesis realizada por ANOVA, ya que indica que la estimación que realizamos de distancia entre personas no depende de la distancia de la cámara a dichos objetivos, esto se cumplió en todos los casos excepto para una distancia de 20 metros en donde los datos obtenidos en las pruebas indicaron únicamente para este caso las medias no

son significativamente iguales.

- Al desplegar el sistema en el área comercial de Pelileo - El Tambo, se pudo comprobar que la alerta auditiva pudo persuadir a las personas que se encontraban transitando por el sector e incumplían el distanciamiento social.
- Se pudo comprobar que el sistema, pese al tener una estabilidad aceptable, cuando exista multitud de alta densidad, las personas se superponen entre sí, por lo que es bastante difícil distinguir a las personas de manera confiable, también tiene incidencias bajas de falsos positivos cuando existen prendas de vestir cerca del cuadro de visión de la imagen, así como cuando existe algún objeto reflejante que puede confundir al sistema.

5.2. Recomendaciones

En el proceso de la elaboración del presente trabajo de investigación, se ha logrado identificar posibles ramificaciones que pueden complementar o escalar al sistema implementado. Como continuación directa del trabajo actual se pueden realizar pruebas masivas en diferentes puntos de la ciudad de Ambato para poder determinar una base de datos en donde se pueda detectar y prevenir puntos de aglomeraciones de manera predictiva.

Por otro lado, se puede recomendar complementar el estudio realizando sistemas parecidos con otro tipo de algoritmos de inteligencia artificial como los mencionados en este proyecto para determinar una base de datos y delimitar los errores en este tipo de proyectos para futuras implementaciones.

Finalmente, el sistema puede ser modificado e implementado en la industria 4.0, para poder controlar el flujo de personal que existe dentro de la planta industrial, así como en aplicaciones de seguridad industrial en donde los operadores requieran estar a cierta distancia de un objetivo y el supervisor pueda controlar, advertir y prevenir algún accidente.

Bibliografía

- [1] M. A. Pérez, *Espacios de Color RGB, HSI y sus Generalizaciones a n-Dimensiones*. Puebla: INADE, 2009.
- [2] UNED, *Gráficos Vectoriales y Mapa de Bits*. España: Universidad Nacional de Educación a Distancia, 2007.
- [3] B. Uscaítegui, *Procesamiento difuso de imágenes: filtro difuso para eliminar el ruido impulsivo*. España: Universidad de La Rioja, 2003.
- [4] I. L. T. G, “Evaluación de las redes neuronales artificiales perceptrón multicapa y fuzzy-arma en la clasificación de imágenes satelitales,” *Ingeniería*, pp. 61-72, 2012.
- [5] E. S. Atienza, *Detección de personas mediante redes convolucionales*. Madrid: Universidad Autónoma de Madrid, 2017.
- [6] D. C. Maroco, *Detector automático de rasgos oculares mediante filtros de Gabor*. Móstoles: Universidad Rey Juan Carlos, 2010.
- [7] Y. G. García, *Algoritmos SVM para problemas sobre big data*. Madrid: Universidad Autónoma de Madrid, 2013.
- [8] K. F. M. Bustillos, *Diseño de una aplicación móvil para el control de un dron con gestos*. Sangolquí: Universidad de las Fuerzas Armadas ESPE, 2021.
- [9] M. A. Cayambe, *Desarrollo de un prototipo de una aplicación móvil híbrida de reconocimiento facial para la DINASED de la ciudad de Riobamba*. Riobamba: Escuela Superior Politécnica de Chimborazo, 2018.
- [10] J. A. Julia, *Dron de vuelo autónomo con reconocimiento basado en inteligencia artificial*. Madrid: Universidad Complutense de Madrid, 2020.
- [11] Emergencias, *Brote de enfermedad por coronavirus (COVID-19): orientaciones para el público*, Centro de Operaciones de Emergencia, Ecuador, 2020.
- [12] A. Sánchez, “Covid-19: cuarentena, aislamiento, distanciamiento social y confinamiento, ¿son lo mismo?” *Anales de Pediatría*, pp. 73-74, 2020.
- [13] E. 911, *Sistema Ecuatoriano de Vigilancia*, ECU911, Ecuador, May 2022. [Online]. Available: <https://www.ecu911.gob.ec/camaras-de-videovigilancia/>
- [14] M. Viscaíno, *Sistema de patrullaje de drones para detección de personas en espacios restringidos*. Sangolquí: Universidad de las Fuerzas Armadas ESPE, 2022.
- [15] COE, *Resolución 09 de febrero del 2022*, Comité de Operaciones de Emergencia, Ecuador, feb 2022. [Online]. Available: <https://www.gestionderiesgos.gob.ec/wp-content/uploads/2022/02/Coe-Nacional-Resolucion-miercoles-9-de-febrero-del-2022.pdf/>
- [16] AME, *Pelileo 'la ciudad azul', tierra que viste al ecuador*, Asociación de

- Municipalidades del Ecuador, Ecuador, feb 2022. [Online]. Avaliable: <https://ame.gob.ec/Pelileo-la-ciudad-azul-tierra-que-viste-al-ecuador/>
- [17] Guardiania, *VMS - Detección de personas*, May 2020. [Online]. Avaliable: <https://guardianva.com/es/vms-deteccion-de-personas>
- [18] C. W. B. S. a. P. P. P. Dollar, "Pedestrian detection: An evaluation of the state of the art," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 743-761, 2012.
- [19] T. E. T. P. C. Papageorgiou, *A Trainable Pedestrian Detection System*. Proceedings of Intelligent Vehicles, 1999.
- [20] J. G. y. S. M. D. Gavrila, "Vision-based pedestrian detection: the protector system," *IEEE Intelligent Vehicles Symposium*, pp. 13-18, 2004.
- [21] C. W. B. S. a. P. P. P. Dollar, "Pedestrian detection: A benchmark," *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 304-311, 2009.
- [22] H. Q. a. J. K. X. B. Cao, "A low-cost pedestrian-detection system with a single optical camera," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 1, pp. 58-67, 2008.
- [23] C.-J.-R.-M.-Y. MarkLiao, "Pedestrian detection and tracking at crossroads," *Pattern Recognition*, vol. 37, no. 5, pp. 1025-1034, 2004.
- [24] J. D. L. C. L. C. L. Roberto, *Detección de personas en imágenes de profundidad mediante redes neuronales convolucionales*. Madrid: Universidad de Alcalá, 2019.
- [25] A. Y. M. Y. a. J. O. M. Szarvas, "Pedestrian detection with convolutional neural networks," *IEEE Proceedings. Intelligent Vehicles Symposium*, pp. 224-229, 2005.
- [26] A. E. D. Andróes, *Visión artificial aplicada a la detección e identificación de personas en tiempo real*. Quito: Escuela Politécnica Nacional, 2019.
- [27] C. P. P. S. E. O. a. T. P. M. Oren, "Pedestrian detection using wavelet templates," *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 193-199, 1997.
- [28] M. F. Clara, *Desarrollo de un sistema de detección de personas en ambientes de interior usando cámaras ojo de pez en plano cenital y algoritmos basados en Deep Learning*. Madrid: Universidad Politécnica de Madrid, 2018.
- [29] C. J. G. Alirio, *Sistema de detección de aglomeración y conteo de personas basado en procesamiento de imágenes e internet de las cosas IOT para el grupo TIGUM*. Colombia: Universidad del Bosque, 2020.
- [30] R. Ramiro, *Video sensor para la detección y seguimiento de personas*. Cuba: Universidad de las Ciencias Informáticas, 2017.
- [31] S. A. C. W. R. Parra, *Sistema de visión artificial integrado a plataforma aérea para la detección de personas en tiempo real*. Colombia: Universidad Militar Nueva Granada, 2018.

- [32] A. Mendieta, *Plataforma de evaluación de algoritmos de detección de personas*. Madrid: Universidad Autónoma de Madrid, 2017.
- [33] P. Germán, *Prototipo De Sistema Embebido Para La Detección De Personas En Tiempo Real Para El Control De Espacios Mediante Técnicas De Aprendizaje Profundo*. Quito: Universidad San Francisco de Quito, 2021.
- [34] P. E. H. A. L. Karla, “Detección y seguimiento de personas con análisis de color en datos RGB-d,” *Memorias del XVI Congreso Latinoamericano de Control Automático*, 2014.
- [35] J. M. G. d.-G. y. A. G.-C. Juan M. Gandarias, “Clasificación de información táctil para la detección de personas,” in *Jornadas de Automática*, Oviedo, 2017.
- [36] A. K. a. B. Brik, “An edge-based social distancing detection service to mitigate covid-19 propagation,” *IEEE Internet of Things Magazine*, vol. 3, no. 3, pp. 35-39, 2020.
- [37] G. C. J. M. Z. W. J. W. a. D. L. Z. Shao, “Real-time and accurate uav pedestrian detection for social distancing monitoring in covid-19 pandemic,” *IEEE Transactions on Multimedia*, vol. 24, pp. 2069-2083, 2022.
- [38] K. R. y. U. O. D. Yang, E. Yurtsever, “A vision-based social distancing and critical density detection system for covid-19,” *Sensors*, 2021.
- [39] S. E. A. G. Saponara, “Implementing a real-time, ai-based, people detection and social distancing measuring system for covid-19,” *Real-Time Image Proc*, vol. 18, pp. 1937-1947, 2021.
- [40] Z. Andreas, *Desarrollo de un prototipo para detección de mascarillas y control de aforo de personas en espacios cerrados*. Sangolquí: Universidad de las Fuerzas Armadas ESPE, 2022.
- [41] P. Mauricio, “Detección de concentración de personas aplicando biga data para evitar brotes epidemiológicos covid-19,” *Identidad Bolivariana*, ISSN-e 2550-6749, vol. 4, no. 2, pp. 5-19, 2020.
- [42] P. Q. Simbeni, *Procesamiento Digital de Imágenes*. Bahía Blanca: Universidad Nacional del Sur, 2015.
- [43] W. L. Hosch, *Bitmap*, Britannica. [Online]. Available: <https://www.britannica.com/technology/bitmap>
- [44] M. C. Alberdi, *La imagen digital. ¿Nuevos interrogantes para la semiótica?* Rosario: UNR, 2004.
- [45] U. Europea, *iluminación, contraste, tamaño y color en el medio ambiente*. Mérida: Consejería de Hacienda y Administración Pública.
- [46] C. Comunicación, *Imagen Digital*. Argentina: Universidad Nacional de Lomas De Zamora, 2014.
- [47] L. Rouhiainen, *Inteligencia Artificial*. Barcelona: Editorial Planeta, 2018.
- [48] J. Brownlee, *Deep Learning for Computer Vision: Image Classification, Object*

Detection and Face Recognition in Python. Machine Learning Mastery, 2019.

- [49] C. M. R. Q, “Redes neuronales artificiales para el procesamiento de imágenes, una revisión de la última década,” *Revista de ingeniería eléctrica, electrónica y computación*, pp. 7-16, 2011.
- [50] A. Carlos, *Redes neuronales recurrentes: Análisis de los modelos especializados en datos secuenciales*. Buenos Aires: Universito of CEMA, 2021.
- [51] D. R. F. Marco, “Histograma del gradiente con múltiples orientaciones (hogo) detección de personas,” *Revista Vínculos*, pp. 138-147, 2015.
- [52] J. I. Tartaria, *Detección de personas*. Euskadi: Universidad del País Vasco, 2013.
- [53] S. González, *Transformada Wavelet Gabor*. Colombia: Fundación Universitaria Juan de Castellanos, 2004.

Anexos

A.1. Datos generados por el Sistema

CistanciaCamaraObjett!	CistarciaFeal	Distarte i aPisel a	Cist an ti aM etrn: E st ad □	%Detecciones Base Dato
3	0,5	116,9264	0,45833 pefigro	40,8863
3	1	173,2532	0,83818 peljgro	72,5846
3	1,5	232,0324	1,357 peligre	63,5031
3	2	295,9654	1,8551 distanciados	84,3217
3	2,5	355,139	2,3174 distanciEdzs	62,181-
3	3	409,0916	2,7377 distanciados	35,2995
3,5	0,5	113,3401	Q46835 peljgro	78,6571
^5	1	165,4992	0,83769 peljgro	78,7033
3,5	1,5	247,9751	1,431 peljgro	73,5432
^5	2	300,7SS9	1,893 distanciados	80,9391
4*	2,5	364,5902	2,3906 d i sta nc i ados	54,2613
3,5	3	405,6324	2,7107 distanciEdzs	82,3365
4	0,5	129,3043	0,55537 paliare	9CJ8443
4	1	184,1359	0,93306 peljgro	54,6165
4	1,5	236, AS	1,3903 p E 1 igro	46,543
4	2	302,9355	1,9097 distanciados	53,3656
4	2,5	341,5562	2,2109 dEta nejados	83,86-3
4	3	413,6618	2,3124 d i sta nc i ados	57,77-3
¿,5	0,5	113,0525	0,42861 pefigro	62,9933
¿,5	1	181,6105	0,96336 peljgro	46,6239
¿,5	1,5	235,4436	1,3833 peligre	73,9229
¿,5	2	299,8251	1,8862 distanciados	42,0023
¿,5	2,5	347,2313	2,2552 distanciEdzs	57,9798
¿,5	3	AS, 2947	2,7393 distanciados	96,9527
üil	0,5	118,6239	0,47207 peligre	67,267
5	1	178,0402	0,93551 pElr^rz	93,8041
5	1,5	239,424	1,4143 peljgro	87,8106
S	2	293,3291	1,8738 d i sta nc i ados	83,4457
S	2,5	343,8129	2,2285 distanciados	60,5183
3	3	396,6246	2,6405 distanciEdzs	57,461

10	1	195, +943	1,0717	pe igio	57,8059
10	1	194,0224	1,0602	pe igro	60,0079
10	1	185,238	0,99166	pe igrc	45,8559
10	1	185,7517	0,99566	pe igru	40,0979
10	1	188,437	1,017	pe igrc	91,9206
10	1	194,8349	1,0665	pe igru	98,0296
10	1	194,6718	1,0652	pe igrc	62,48
10	1	190,9331	1,0361	pe igru	67,2285
10	1	192,1C54	1,0452	pe igru	78,136
10	1	187,6366	1,0104	pe igru	93,6352
10	2	308,4595	1,9528	distanciado:	75,2668
10	2	305,2383	1,9277	distanciado:	81,3072
10	2	306,3611	1,9364	distanciado:	51,5648
10	2	302,4449	1,9059	distanciado:	97,9873
10	2	306,115	1,9501	distanciado:	96,4777
10	2	303,2614	1,9122	distanciado:	53,6689
10	2	302,8135	1,9087	distanciado:	63,6252
10	2	306,2276	1,951	distanciado:	91,8559
10	2	306,184	1,9506	distanciado:	59,1714
10	2	305,5685	1,9302	distanciado:	55,283

15	1	186,9621	1,0051	pe igru	55,0126
15	1	195,9591	1,0753	pe igru	43,8493
15	1	185,0131	0,9899	pe igro	85,8979
15	1	188,5138	1,0172	pe igru	66,0872
15	1	192,0635	1,0449	pe igru	58,8506
15	1	184,1756	0,98337	pe igro	76,56
15	1	193,7367	1,0579	pe igro	90,7835
15	1	193,7904	1,0584	pe igru	48,4556
15	1	193,1035	1,053	pe igro	46,2653
15	1	184,466	0,98563	pe igro	79,4758
15	2	306,6107	1,9384	distanciado:	43,6235
15	2	303,6319	1,9151	distanciado:	74,2167
15	2	305,0314	1,926	distanciado:	65,5886
15	2	302,9775	1,91	distanciado:	84,1
15	2	305,9002	1,9328	distanciado:	48,9703
15	2	305,2472	1,9277	distanciado:	72,1897
15	2	303,1098	1,9111	distanciado:	81,2148
15	2	306,2089	1,9352	distanciado:	62,2416
15	2	306,3486	1,9363	distanciado:	79,7166
15	2	302,188	1,9089	distanciado:	68,2037

20	1	170,427	0,87613pe igrc	99,0956
20	1	174,0938	0,90473pe igrc	94,7801
20	1	173,9631	0,90371pe igrc	91,1224
20	1	166,0169	0,84173pe igru	70.2892
20	1	168,6972	0,86264 pe igrc	80,9518
20	1	171,0227	0,88078pe igru	79.4647
20	1	176,2531	0,92157pe igrc	68,6457
20	1	173,0464	0,89656pe igrc	50,1114
20	1	163,1078	0,81904 pe igru	82,7232
20	1	164,6301	0,8313pe igrc	93,9758
20	2	403,6401	2,6952distanciado;	56,2515
20	2	328^1777	2,1066distanciado;	46,6792
20	2	350,8617	2,2835distanciado;	83,6333
20	2	325,3962	2,0849distanciado;	49,8896
20	2	397,5203	2,6475distanciado;	90,3779
20	2	407,5911	2,726distanciado;	56,1816
20	2	394,9073	2,6271distanciado;	71,3404
20	2	348,4791	2,2649distanciado;	79,6346
20	2	353,413	2,3034distanciado;	95,6008
20	2	353,3552	2,303distanciado;	54,8013

A.2. Código del Programa

```
classdef AplicacionI < matlab.apps.AppBase

    % Properties that correspond to app components properties (Access = public)
    UIFigure                                matlab.ui.Figure

    SISTEMADEDETECCINDEDISTANCIAMIENTOSOCIALPanel matlab.ui.container.Panel

    TabGroup                                matlab.ui.container.TabGroup
    OFFLINETab                              matlab.ui.container.Tab
    Image_15                                matlab.ui.control.Image
    Image_14                                matlab.ui.control.Image
    DesingbyChristianSilvaLabel Powe       matlab.ui.control.Label
    redbyMATLAB Label                       matlab.ui.control.Label
    SilenciarAlarmaButton                  matlab.ui.control.Button
    Image_6                                  matlab.ui.control.Image
    GenerarArchivotxtButton                matlab.ui.control.Button
    ddmmaaLabel                             matlab.ui.control.Label
    FechadelVideoLabel                     matlab.ui.control.Label
    Image_4                                  matlab.ui.control.Image
    Image_3                                  matlab.ui.control.Image
    PeligroAglomeracionesLabel             matlab.ui.control.Label
    PrecaucinAglomeracionesLabel           matlab.ui.control.Label
    DistanciamientoApropiadoLabel          matlab.ui.control.Label
    duracionvideoLabel_4                    matlab.ui.control.Label
    segundosLabel                           matlab.ui.control.Label
    duracionvideoLabel_2                    matlab.ui.control.Label
    FechayHoraActualLabel                   matlab.ui.control.Label
    minutosLabel                            matlab.ui.control.Label
    DuracindelVideoLabel                    matlab.ui.control.Label
    Image_2                                  matlab.ui.control.Image
    Image                                    matlab.ui.control.Image
    SeleccioneunArchivoaviLabel             matlab.ui.control.Label
    NombredelArchivoLabel                   matlab.ui.control.Label
    SeleccionarArchivoButton                matlab.ui.control.Button
    Lamp 2                                   matlab.ui.control.Lamp
    Lamp33                                   matlab.ui.control.Lamp
```

Lamp	matlab.ui.control.Lamp
Label	matlab.ui.control.Label
ContadordeAlarmasLabel	matlab.ui.control.Label
ReproducirButton	matlab.ui.control.Button
UIAxes	matlab.ui.control.UIAxes
ONLINETab	matlab.ui.container.Tab
Image_17	matlab.ui.control.Image
Image_16	matlab.ui.control.Image
DesingbyChristianSilvaLabel_2 Power re d	matlab.ui.control.Label
byHAT LAB L a b e 1_2	matlab.ui.control.Label
Image_13	matlab.ui.control.Image
Image_12	matlab.ui.control.Image
DetenerGrabacinButton	matlab.ui.control.Button
IniciarGrabacinButton DetenidoLabel	matlab.ui.control.Button
EstadodeGrabacinLabel	matlab.ui.control.Label
GenerarArchivotxtButton_2	matlab.ui.control.Label
PeligroAglomeracionesLabel_2	matlab.ui.control.Button
PrecaucinAglomeracionesLabel_2	matlab.ui.control.Label
DistanciamientoApropiadoLabel_ Image_II	matlab.ui.control.Label
SilenciarAlarmaButton_2	matlab.ui.control.Label
ContadordeAlarmasLabel_2 Image_16	matlab.ui.control.Image
Image_9	matlab.ui.control.Button
Image_8	matlab.ui.control.Label
FechayHoraActualLabel_2	matlab.ui.control.Image
Image_7	matlab.ui.control.Image
Lamp_6	matlab.ui.control.Image
DesconectadoLabel	matlab.ui.control.Label
EstadodeConexinLabel	matlab.ui.control.Image
Lamp2_2	matlab.ui.control.Lamp
Lamp_4	matlab.ui.control.Label
Lamp_3	matlab.ui.control.Label
Label 2	matlab.ui.control.Lamp
	matlab.ui.control.Lamp
	matlab.ui.control.Lamp
	matlab.ui.control.Label

```

EstablecerConexinCmaralPButton matlab.ui.control.Button
UIAxes_2 matlab.ui.control.UIAxes
end

% Callbacks that handle component events methods (Access = private)

% Code that executes after component creation function startupFcn(app)
ele
clear all
cióse all
global fecha contnombreach contnombreach=0;
fecha=datetime(now,"ConvertFrom","datenum"); fechastr=datestr(date);
aplicación.FechayHoraActualLabel.Text=fechastr;
app.FechayHoraActualLabel 2. Text=fechastr;
end

% Button pushed function: ReproducirButton
function ReproducirButtonPushedfappj event)
global videoPath video inicio matrizDatos distancia estado detección player
%Lectura del video dentro de la carpeta
[y, Fs] = audioread('alerta.mp3');
player = audioplayer(y, Fs);
videoReader = visión.VideoFileReader(videoPath);
inicio=0;
matrizDatos= [];
^Posición en la pantalla del reproductor de video
videoPlayer = visión.VideoPlayerC'Position", [300 100 1000 500]);
%CREA EL OBJETO DETECTOR PARA LA detección de personas mediante algoritmo RED DE CANALES
AGREGADOS C %DE DATOS DE CALTECH
detector = peopleDetectorACF("caltech-50x21");

acumulador=0;
actual=0;
tnediav=0;
mediah=0;
«hile ~isDcne(videoReader)
«MIENTRAS EL VIDEO SE PUEDA REPRODUCIR SE ALMACENA CADA FRAME EN UNA / ;VARIABLE PARA LA
DETECCIPON DE PERSONAS
aux=0;
fecha=datetime(now,"ConvertFrom","datenum");
fechastr=datestr(-Fecha);
app.FechayHoraActualLabel.Text=fechastr;
trame = step(videoReader);
l=double(trame);
frame2=readFrame(video); actúa1=video.CurrentTime;
[bboxesSjSeores] = detectCdetector,);
/ ;EXTRAE LOS DATOS DE Puntuación y LIMITES DONDE SE PUEDA DETECTAR A UNA «PERSONA
DENTRO DEL FRAME ANALIZADO "l" Y EL DETECTOR CREADoR! cond=zeros(size(bboxes,l),1);
/ ;VALOR AUXILIAR DEPENDIENDO DE CUANTAS PERSONAS DETECTO EN if ~isempty(bboxes)
EL FRAME
for i=1:(size(bboxes,l)-l)
for j=(i+1):(size(bboxes,l)-l) disl_v=abs(bboxes(i,l)+bboxes(ij,3)-bboxesfj,1)) «CALCULA LA DISTANCIA
VERTICAL DE PíxeLES ENTRE «ANALIZADA Y LA CAJA,
MÁS PROXIMA dis2_v=abs(bboxes(j,l)+bboxes(j,3)-LA CAJA DE DETECCIÓN
bboxes(ij,1)) mediav=(disl_v+dis2_v)/2;
«CALCULA UNA DISTANCIA AUXILIAR ENTRE CATAS disih=abs(bboxes(i,2)-bboxes(i,2));
/ ;CALCULA LA DISTANCIA HORIZONTAL ENTRE LAS CAIAS PROXIMAS dis2_h=abs(bboxes(ij,2)
+bboxes(ij,2)-bboxesfj,2)-bboxes(j,2)); mediah=(disl_h+dis2_h)/2;
distancia=sqrt(niediah^2+mediav^2);
«CALCULA LA DISTANCIA HORIZONTAL ENTRE LAS CAIAS PROXIMAS

```

```

        if ((disl_v<75 || dis2_v<40)&&(disl_h<40 || dis2_h<75)) cond(i)=cond(i)+1; cond(j)=cond(j)+1;
            %SI LA DISTANCIA ENTRE LOS PÍXELES ES MENOR QUE 50
            %HORIZONTALMENTE SE DETERMINA EXISTE DISTANCIAMIENTO ENTRE ESAS
            PERSONAS aux=(aux+1) /2; ! QUE 75 Y
            estado=" No Distanciados"; QUE NO
        else
            cond(i)=cond(i)+0; aux=(aux+0) /2; estado="Distanciados";
            XCONDICIÓN EQUIVALENTE A DISTANCIADOS end
        detección = mean(scores);
    end
end end I=insertObjectAnnotation (I,"rectangle"3 bboxes((cond>0), :),"Peli I=insertObjectAnnotation
(I,"rectangle", bboxes((cond==0), :))/"Dis XCOLOCA RECTANGULOS DE COLOR PARA DIFERENCIAR ENTRE
DISTANCIADOS Y NO ^DISTANCIADOS step (videoPlayer, I); imshow .gro', 'color','r');
(I)'Parent', app. UIAxes); itanciados','color','g');
XPASA AL SIGUIENTE FRAME acuuulador=acumulador+aux;
acumuladorentero=round(acumulador); app. Label.Text=num2str(acumuladorentero); datcs= [fechastr distancia
estado acumuladorentero detección]; %matrizDatos= [matrizDatos; datos];
if (acumulador>4) &&(acumulador<10)
    play(player);
    app.Lamp2.Color='white';
    app. Lamp. Color='white'; app. Lamp33.Color='yellow'.

elseif acumulador>10
    app. Lamp2.Color='white'; app. Lamp.Color='red'; app. Lamp33.Color='white'; acumulador=0.
else
    app. Lamp2.Color='green'.
    app. Lamp.Color='white'; app. Lamp33.Color='white'.
end
end
end
release(videoReader).
release(videoPlayer); acuinulador=0.
end

% Button pushed function: SeleccionarArchivoButton function SeleccionarArchivoButtonPushed (app, event)
global videoPath video inicio tipos= {'1 *.mp4; *.avi'};
[file, path] =uigetfile(tipos).
if isequal(file,0)
    app. SeleccionarArchivoLabel.Text='NO SE HA SELECCIONADO NINGÚN ARCHIVO';
else
    videoPath=fullfile(path,file).
    app. SeleccionarArchivoLabel.Text=file; app. OFFLINETab.OuterPosition.
    video=VideoReader(videoPath); dur=video. Duration/60.
    durm=fix(dur).
    app. minutosLabel1.Text=num2str(durm).
    durs=round(abs(durm-dur) *60).
    app. segundosLabel.Text=num2str(durs);
    fechvideo=strsplit (file,
    app. ddmmaaLabel.Text=fechvideo(I);

```

```

end
end

% Button pushed function: EstablecerConexinCmaralPButton function EstablecerConexinCmaralPButtonPushed
(appj event)
global videoPath video inicio matrizDatosON distancia estado detección player cam ;Lectura del video dentro
de la carpeta cam = ipcam('http://192.168.188.11:8080/video');
estadocam=;
app. Lamp_6. Color='green'; app. DesconectadoLabel.Text='CÁMARA CONECTADA';
videoPlayer = visión. VideoPlayer ("Position", [300 100 1000 500])
%CREA EL OBJETO DETECTOR PARA LA detección de personas mediante algoritmo RED DE CANALES AGREGADOS C
%DE DATOS DE CALTECH
detector = peopleDetectorACF("caltech-50x21");
[y, Fs] = audioread('alerta.mp3');
player = audioplayer (y, Fs);
inicio 0;
matrizDatosON= [];
acumulador=0;
actual=0; mediav=0;
mediah=0; distanciametros=0;
estadocam==l
aux=0;
fecha=datetime(non,"ConvertFrom""datenum"); fechastr=datestr (fecha);
app. FechayHoraActualLabel 2. Text=fechastr.
trame = snapshot(cam).
l=double (trame).
[bboxesSjSCores] = detect (detector, l);
/ ¡EXTRAE LOS DATOS DE PUNTUACIÓN Y LIMITES DONDE SE PUEDA DETECTAR A UNA
¡¡PERSONA DENTRO DEL FRAME ANALIZADO "l" Y EL DETECTOR CREADOR!! cond=zeros (size
(bboxes, l),1);

if ~isempty(bboxes)
for i=l:(size (bboxes, l)-l)
for j=(i+1) :(size (bboxes, l) 1)
disl_v=abs (bboxes (i, l) +bboxes(i,3)-bboxes(j,1)) / ¡CALCULA LA DISTANCIA VERTICAL
DE PÍXELES ENTRE / ¡ANALIZADA Y LA CAJA MÁS LA CAJA DE DETECCIÓN
PROXIMA dis2_v=abs (bboxes (j, l) +bboxes(j,3)-
bboxes(i,1)) mediav= (disl v+dis2 v) /2;
/ ¡CALCULA UNA DISTANCIA AUXILIAR ENTRE CAJAS disl_h=abs(bboxes(i,2)-
bboxes(j,2));
¡¡CALCULA LA DISTANCIA HORIZONTAL ENTRE LAS CAJAS PROXIMAS
dis2_h=abs(bboxes(i,2) +bboxes(i,4)-bboxes(j,2)-bboxes(j,4)); mediah=(disl_h+dis2_h) /2;
distância=sqrt(mediah^2+mediav^2); distanciametros=0.5078*distância-0.4532;
if distanciametros<0.5
estadc="No Distanciados";
else
estadc="Distanciados";
end
/¡CALCULA LA DISTANCIA HORIZONTAL ENTRE LAS CAJAS PROXIMAS if ((disl_v<75
|| dis2_v<40)&&(disl_h<40 || dis2_h<75)) cond(i)=cond(i)+1; cond(j)=cond(j)+1;
SÍ SI LA DISTANCIA ENTRE LOS PÍXELES ES MENOR %MENOR QUE 50
HORIZONTALMENTE SE DETERMINA QUE NO ¡¡EXISTE DISTANCIAMIENTO
ENTRE ESAS PERSONAS!! aux=(aux+1) /2; QUE 75 Y
estadc="No Distanciados";
else
cond(i)=cond(i)+0; aux=(aux+0) /2; estadc="Distanciados";
¡¡CONDICIÓN EQUIVALENTE A DISTANCIADOS
end

```

```

        detección = mean(scores); end
    end end l=insertObjectAnnotation (l,"rectangle", bboxes((cond>0), :), "Peli l=insertObjectAnnotation (l,
"rectangle", bboxes((cond==0), :), "Dis «COLOCA RECTANGULOS DE COLOR PARA DIFERENCIAR ENTRE
DISTANCIADOS Y NO «DISTANCIADOS . gro','color','r');
step (videoPlayer, l); itanciados','color','g');
    outpict = imfuse (l, frame,'blend'); imshow (outpict,'Parent', app. UIAxes_2);
    acumulador=acumulador+aux; acumuladorentero=round(acumulador);
    app. Label_2. Text=num2str(acumuladorentero); datos= [fechastr distancia estado acumuladorentero detección];
    matrizDatosON= [matrizDatosON; datos];
    if (acumulador>4) &&(acumulador<10) play(player);
        app. Lamp_3. Color='white'.
        app. Lamp_4. Color='white'.
        app. Lamp2_2. Color='yellow'.
    elseif acumulador>10
        app. Lamp2_2. Color='white'.
        app. Lamp_4. Colo r='red'.
        app. Lamp_3. Color='white'; acumulador=0.
    else
        app. Lamp_3. Color='green'.
        app. Lamp2_2. Color='white'.
        app. Lamp_4. Color='white'.
    end
end
end
release(videoPlayer); acumulador=0.
end

% Button pushed function: GenerarArchivotxtButton function GenerarArchivotxtButtonPushed (app, event) global
contnombreach matrizDatos contnombreach=contnombreach+1.
    tablaDatos=array2table (matrizDatos,"VariableNames", ["Hora","dis [nombreachivotxt, direccionarchivotxt]=uiputfile
("*.txt","Guard writetable (tablaDatos, [direccionarchivotxt nombreachivotxt],"Delimiter' matrizDatos= [];
end tancia","estado", "acumuladorentero", "■ ar
Como");

% Button pushed function: SilenciarAlarmaButton function \t");
SilenciarAlarmaButtonPushedfapp, event)
    global player
    stop(player)
end

% Button pushed function: IniciarGrabacinButton function IniciarGrabacinButtonPushed (app, event) global detener cam
imagenaux vidWriter app. DetenidoLabel.Text='Grabando'; detener=l.
    vidWriter = VideoWriter('aaaa.avi'); open(vidWriter).
    while detener==l
        imagenaux = snapshot(cam); writeVideo (vidWriter, imagenaux); end
end

% Button pushed function: DetenerGrabacinButton function DetenerGrabacinButtonPushed (app, event) global detener
cam imagenaux vidWriter app. DetenidoLabel.Text='Detenido'; writeVideo (vidWriter, imagenaux);

```

```

    detener=0;
    clear cam
end

% Button pushed function: GenerarArchivotxtButton_2
function GenerarArchivotxtButton_2Pushed(app, event)
    global contnombreach matrizDatosON
    contnombreach=contnombreach+1;
    tablaDatos=array2table(matrizDatosON,"VariableNames",["Hora","distancia","estado","acumuladorentero"
[nombreachivotxt,direccionarchivotxt]=uiputfile(''.txt',"Guardar Como");
writetable(tablaDatos,[direccionarchivotxt nombreachivotxt],"Delimiter","\t");
matrizDatosON=[];
end

% Button pushed function: SilenciarAlarmaButton_2
function SilenciarAlarmaButton_2Pushed(app, event)
    global player
    stop(player)
end

end

% Component initialization
methods (Access = private)

% Create UIFigure and components
function createComponents(app)

% Get the file path for locating images
pathToMLAPP = fileparts(mfilename('fullpath'));

% Create UIFigure and hide until all components are created
app.UIFigure = uifigure('Visible', 'off');
app.UIFigure.Position = [100 100 884 467];
app.UIFigure.Name = 'MATLAB App';

% Créate SISTEMADEDETECCINDEDISTANCIAMIENTOSOCIALPanel app.
SISTEMADEDETECCINDEDISTANCIAMIENTOSOCIALPanel = uipanel(UIFigure);
(app
    app.EMA DE DETECCIÓN DE
SISTEMADEDETECCINDEDISTANCIAMIENTOSOCIALPanel.Title = 'SISTDISTANCIAMIENTO S rial Black';
    app.
        'bold';
SISTEMADEDETECCINDEDISTANCIAMIENTOSOCIALPanel.FontName =-35 881 503];
'A
    app.
SISTEMADEDETECCINDEDISTANCIAMIENTOSOCIALPanel.FontWeight =
    app.
SISTEMADEDETECCINDEDISTANCIAMIENTOSOCIALPanel.Position = [1

% Create TabGroup
app.TabGroup = uitabgroup (app. SISTEMADEDETECCINDEDISTANCIAMIENTOSOCIALPanel).
app.TabGroup.Position = [11 880 482],

% Create OFFLINETab
app.OFFLINETab = uitab (app.TabGroup),
app.OFFLINETab.Title = 'OFFLINE'.

% Create UIAxes
app.UIAxes = uiaxes (app.OFFLINETab).
title (app.UIAxes, 'REPRODUCCION DE VIDEO')
app.UIAxes.XTick = [].
app.UIAxes.YTick = [].
app.UIAxes.Position = [5 59 509 397],

% Create ReproducirButton
app.ReproducirButton = uibutton (app.OFFLINETab, 'push').
app.ReproducirButton.ButtonPushedFcn = createCallbackFcn (app, @ReproducirButtonPushed, true); app.
ReproducirButton.FontWeight = 'bold'.
app.ReproducirButton.Position = [668 300 100 23].
app.ReproducirButton.Text = 'Reproducir'.

% Create ContadordeAlarmasLabel
app.ContadordeAlarmasLabel = uilabel (app.OFFLINETab);
app.ContadordeAlarmasLabel.FontSize = 20;
app.ContadordeAlarmasLabel.FontWeight = 'bold';

```

```

app.FechayHoraActualLabel.FontWeight = 'bold';
app.FechayHoraActualLabel.Position = [600 431 251 26];
app.FechayHoraActualLabel.Text = 'Fecha y Hora Actual';

% Create duracionvideoLabel_2
app.duracionvideoLabel_2 = uilabel(app.OFFLINETab);
app.duracionvideoLabel_2.HorizontalAlignment = 'center';
app.duracionvideoLabel_2.Position = [691 349 25 22];
app.duracionvideoLabel_2.Text = 'm:';

% Create segundosLabel
app.segundosLabel = uilabel(app.OFFLINETab);
app.segundosLabel.HorizontalAlignment = 'center';
app.segundosLabel.Position = [715 349 32 22];
app.segundosLabel.Text = '0';

% Create duracionvideoLabel_4
app.duracionvideoLabel_4 = uilabel(app.OFFLINETab);
app.duracionvideoLabel_4.HorizontalAlignment = 'center';
app.duracionvideoLabel_4.Position = [746 349 25 22];
app.duracionvideoLabel_4.Text = 's';

% Create DistanciamientoApropiadoLabel
app.DistanciamientoApropiadoLabel = uilabel(app.OFFLINETab);
app.DistanciamientoApropiadoLabel.FontSize = 18;
app.DistanciamientoApropiadoLabel.FontWeight = 'bold';
app.DistanciamientoApropiadoLabel.Position = [612 196 241 23];
app.DistanciamientoApropiadoLabel.Text = 'Distanciamiento Apropiado';

% Create PrecaucinAglomeracionesLabel
app.PrecaucinAglomeracionesLabel = uilabel(app.OFFLINETab);
app.PrecaucinAglomeracionesLabel.FontSize = 18;
app.PrecaucinAglomeracionesLabel.FontWeight = 'bold';
app.PrecaucinAglomeracionesLabel.Position = [612 143 249 23];
app.PrecaucinAglomeracionesLabel.Text = 'Precaución Aglomeraciones'; % Create
PeligroAglomeracionesLabel
app.PeligroAglomeracionesLabel = uilabel(app.OFFLINETab);
app.PeligroAglomeracionesLabel.FontSize = 18;
app.PeligroAglomeracionesLabel.FontWeight = 'bold';
app.PeligroAglomeracionesLabel.Position = [612 88 216 23];
app.PeligroAglomeracionesLabel.Text = 'Peligro Aglomeraciones';

% Create Image_3
app.Image_3 = uiimage(app.OFFLINETab);
app.Image_3.Position = [546 259 35 31];
app.Image_3.ImageSource = fullfile(pathToMLAPP,'alarmaicon.png');

% Create Image_4
app.Image_4 = uiimage(app.OFFLINETab);
app.Image_4.Position = [552 431 55 27];
app.Image_4.ImageSource = fullfile(pathToMLAPP,'dateicon.png');

% Create FechadelVideoLabel
app.FechadelVideoLabel = uilabel(app.OFFLINETab);
app.FechadelVideoLabel.HorizontalAlignment = 'center';
app.FechadelVideoLabel.FontWeight = 'bold';
app.FechadelVideoLabel.Position = [529 328 104 22];
app.FechadelVideoLabel.Text = 'Fecha del Video: ';

% Create ddmmaaLabel
app.ddmmaaLabel = uilabel(app.OFFLINETab); app.ddmmaaLabel.Position = [668 328 163 22]; app.
ddmmaaLabel.Text = 'dd - mm - aa';

% Create GenerarArchivotxtButton
app.GenerarArchivotxtButton = uibutton(app.OFFLINETab, 'push'); app.
GenerarArchivotxtButton.ButtonPushedFcn = createCallbackFcn(app, @GenerarArchivotxtButtonPushed
GenerarArchivotxtButton.FontWeight = 'bold';
app.GenerarArchivotxtButton.Position = [627 45 154 23]; app.GenerarArchivotxtButton.Text = 'Generar
Archivo " * .txt'

```

```

% Create ImageO
app.Image_6 = uiimage(app.OFFLINETab);
app.Image_6.Position = [619 233 45 21];
app.Image_6.ImageSource = fullfile(pathToMLAPP, 'silencio.png') % Create SilenciarAlarmaButton
App. SilenciarAlarmaButton = uibutton(app. OFFLINETab, 'push') ;
App. SilenciarAlarmaButton.ButtonPushedFcn = createCallbackFcn (app, @SilenciarAlarmaButtonPushed I, tru.
App. SilenciarAlarmaButton.FontWeight = 'bold' ;
App. SilenciarAlarmaButton.Position = [669 232 110 23] ;
App. SilenciarAlarmaButton.Text = 'Silenciar Alarma' ;

% Create PoweredbyñATLABLabel app. PoweredbyMATLABLabel = uilabel (app. OFFLINETab) ; app. Powe
redbyMAT LAB Label.FontAngle = 'italic' ;
app. PoweredbyMATLABLabel.Position = [53 39 119 22]; app. Poweredby MATLABLabel.Text = 'Powered by
MATLAEI'.

% Create DesingbyChristianSilvaLabel
app. DesingbyChristianSilvaLabel = uilabel (app. OFFLINETab).
app. DesingbyChristianSilvaLabel.FontAngle = 'italic'.
app. DesingbyChristianSilvaLabel.Position = [341 39 139 22].
app. DesingbyChristianSilvaLabel.Text = 'Desing by Christian Silva'.

% Create Image_14
app. Image_14 = uiimage (app. OFFLINETab).
app. Image_14. Position = [-1 37 55 27].
app. Image_14. ImageSource = fullfile (pathToMLAPP, 'Matlab.png').

% Create Image_15 app. Image_15 = uiimage (app. OFFLINETab); app. Image_15. Position = [466 42 45 18].
app. Image_15. ImageSource = fullfile (pathToMLAPP, 'Copyright.png % Create ONLINETab

% Create ONLINETab
app. ONLINETab = uitab (app. TabGroup).
app. ONLINETab.Title = 'ONLINE'.

% Create UIAxes_2
app. UIAxes_2 = uiaxes (app. ONLINETab).
title (app. UIAxes_2, 'VIDEO EN VIVO')
app. UIAxes_2. XTick = [].
app. UIAxes_2. YTick = [].
app. UIAxes_2. Position = [4 60 502 394].

% Create EstablecerConexinCmaralPButton
app. EstablecerConexinCmaralPButton = uibutton (app. ONLINETab, 'push').
app. EstablecerConexinCmaralPButton.ButtonPushedFcn = createCallbackFcn (app, @EstablecerConexinCmara
app. EstablecerConexinCmaralPButton.FontWeight = 'bold'.
app. EstablecerConexinCmaralPButton.Position = [627 400 198 23];
app. EstablecerConexinCmaralPButton.Text = 'Establecer Conexión Cámara IP';

% Create Label_2
app. Label_2 = uilabel (app. ONLINETab).
app. Label_2. Tag = 'alarma'.
app. Label_2. HorizontalAlignment = 'center'.
app. Label_2. FontSize = 20.
app. Label_2. FontWeight = 'bold'.
app. Label_2. Position = [808 271 38 26].
app. Label_2. Text = '0'.

% Create Lamp_3
app. Lamp_3 = uilamp (app. ONLINETab).
app. Lamp_3. Position = [543 191 43 43].

% Create Lamp_4
app. LampA = uilamp (app. ONLINETab).
app. Lamp_4. Position = [542 82 46 46].
app. Lamp 4. Color = [11 1];

```

```

% Create Lamp2_2
app.Lamp2_2 = uilamp (app.ONLINETab);
app.Lamp2_2.Position = [543 138 43 43];
app.Lamp2_2.Color = [1 1 1];

% Create EstadodeConexinLabel
app.EstadodeConexinLabel = uilabel (app.ONLINETab);
app.EstadodeConexinLabel.FontWeight = 'bold';
app.EstadodeConexinLabel.Position = [567 367 125 22];
app.EstadodeConexinLabel.Text = 'Estado de Conexión:';

% Create DesconectadoLabel
app.DesconectadoLabel = uilabel (app.ONLINETab); app.DesconectadoLabel.HorizontalAlignment = 'center';
app.DesconectadoLabel.Position = [691 367 138 22];
app.DesconectadoLabel.Text = 'Desconectado';

% Create Lamp é
app.Lampó = uilamp (app.ONLINETab);
app.Lamp_6.Position = [828 365 27 27];
app.Lamp_6.Color = [10 B];

% Create Image 7
app.Image_7 = uiimage (app.ONLINETab);
app.Image?.Position = [549 430 55 27];
app.Image?.ImageSource = fullfile (pathToMLAPP, 'dsteicon.png');

% Create FechayHoraActiialLabel_2
app.FechayHoraActualLabel_2 = uilabel (app.ONLINETab);
app.FechayHoraActualLabel_2.HorizontalAlignment = 'center';
app.FechayHoraActualLabel_2.FontSize = 16;
app.FechayHoraActualLabel_2.FontWeight = 'bold';
app.FechayHoraActualLabel_2.Position = [601 432 251 26];
app.FechayHoraActualLabel_2.Text = 'Fecha y Hora Actual';

% Create Image_8
app.Image_8 = uiimage (app.ONLINETab);
app.Image_8.Position = [552 397 55 27];
app.Image_8.ImageSource = fullfile (pathToMLAPP, 'wifi.png');

% Create Image 9
app.Image_9 = uiimage (app.ONLINETab);
app.Image_9.Position = [513 365 55 27];
app.Image_9.ImageSource = fullfile (pathToMLAPP, 'cam.png');

% Create Image_10
app.Image_10 = uiimage (app.ONLINETab);
app.Image_10.Position = [547 267 35 31];
app.Image_10.ImageSource = full-file (pathToMLAPP, 'alarmaicon.png');

% Create ContadordeAlarmasLabel_2
app.ContadordeAlarmasLabel_2 = uilabel (app.ONLINETab);
app.ContadordeAlarmasLabel_2.FontSize = 20;
app.ContadordeAlarmasLabel_2.FontWeight = 'bold';
app.ContadordeAlarmasLabel_2.Position = [593 271 215 26];
app.ContadordeAlarmasLabel_2.Text = 'Contador de Alarmas:';

% Create SilenciarAlarmaButton_2
app.SilenciarAlarmaButton_2 = uibutton (app.ONLINETab, 'push'); app.SilenciarAlarmaButton_2.ButtonPushedFcn = createCallbackFcn app.SilenciarAlarmaButton_2.FontWeight = 'bold'; (app, @SilenciarAlarmaButton_2Pushed);
app.SilenciarAlarmaButton_2.Position = [653 235 110 23]; app.SilenciarAlarmaButton_2.Text = 'Silenciar Alarma';

% Create Image_II
app.Imagell = uiimage (app.ONLINETab);
app.Imagell.Position = [604 237 45 21];
app.Imagell.ImageSource = full file (pathToMLAPP, 'silencio.png');

```

```

% Create DistanciamientoApropiadoLabel_2 app. DistanciamientoApropiadoLabel_2 = uilabel (app. ONLINETab);
app. DistanciamientoApropiadoLabel_2. FontSize = 18; app. DistanciamientoApropiadoLabel_2. FontWeight = 'bold';
app. DistanciamientoApropiadoLabel_2. Position = [596 201 241 23]

app.DistanciamientoApropiadoLabel_2.Text = 'Distanciamiento Apropiado';

% Create PrecaucinAglomeracionesLabel_2 app. PrecaucinAglomeracionesLabel_2 = uilabel (app. ONLINETab);
app. PrecaucinAglomeracionesLabel_2. FontSize = 18;
app. PrecaucinAglomeracionesLabel_2. FontWeight = 'bold';
app. PrecaucinAglomeracionesLabel_2. Position = [600 145 249 23];
app. PrecaucinAglomeracionesLabel_2. Text = 'Precaución Aglomeraciones';

% Create PeligroAglomeracionesLabel_2 app. PeligroAglomeracionesLabel_2 = uilabel (app. ONLINETab); app.
PeligroAglomeracionesLabel_2. FontSize = 18;
app. PeligroAglomeracionesLabel_2. FontWeight = 'bold';
app. PeligroAglomeracionesLabel_2. Position = [603 90 216 23];
app. PeligroAglomeracionesLabel_2. Text = 'Peligro Aglomeraciones % Create GenerarArchivotxtButton_2
app. GenerarArchivotxtButton_2 = uibutton (app. ONLINETab, 'push').
app. GenerarArchivotxtButton_2. ButtonPushedFcn = createCallbackFcn (appj @GenerarArchivotxtButton_2Pus
app. GenerarArchivotxtButton_2. FontWeight = 'bold'.
app. GenerarArchivotxtButton_2. Position = [619 46 154 23].
app. GenerarArchivotxtButton_2. Text = 'Generar Archivo " *.txt % Create EstadodeGrabacinLabel
app. EstadodeGrabacinLabel = uilabel (app. ONLINETab).
app. EstadodeGrabacinLabel.FontWeight = 'bold'.
app. EstadodeGrabacinLabel.Position = [571 301 130 22].
app. EstadodeGrabacinLabel.Text = 'Estado de Grabación.';

% Create DetenidoLabel

% Create DetenidoLabel
app. DetenidoLabel = uilabel (app. ONLINETab);
app. DetenidoLabel.HorizontalAlignment = 'center';
app. DetenidoLabel.Position = [700 301 138 22];
app. DetenidoLabel.Text = 'Detenido';

% Create IniciarGrabacinButton
app. IniciarGrabacinButton = uibutton (app. ONLINETabj 'push').
app. IniciarGrabacinButton.ButtonPushedFcn = createCallbackFcn (appj @IniciarGrabacinButtonPushedj tru app.
IniciarGrabacinButton.Position = [567 334 107 23].
app. IniciarGrabacinButton.Text = 'Iniciar Grabación';

% Create DetenerGrabacinButton
app. DetenerGrabacinButton = uibuttontapp. ONLINETabj 'push').
app. DetenerGrabacinButton.ButtonPushedFcn = createCallbackFcnCapp, @DetenerGrabacinButtonPushed_, tru«
app. DetenerGrabacinButton.Position = [736 334 117 23].
app. DetenerGrabacinButton.Text = 'Detener Grabación'.

% Create Image_12
app. Image_12 = uiimage (app. ONLINETab).
app. Image_12. Position = [513 332 55 27].
app. Image_12. ImageSource = -Full-fileCpathToñLAPP, 'recicon.png').

% Create Image_13
app. Image_13 - uiimage (app. ONLINETab).
app. Image_13. Position = [687 332 55 27].
app. Image_13. ImageSource = -FullfilepathToMLAPP, 'stopicon.png').

% Create PoweredbyMATLABLabel_2
app. PoweredbyMATLABLabel_2 = uilabel (app. ONLINETab).
app. PoweredbyMATLABLabel_2. FontAngle = 'italic'.
app. PoweredbyMATLABLabel_2. Position = [52 39 119 22].
app. PoweredbyMATLABLabel_2. Text = 'Powered by MATLAB';

```

```

app.DesingbyChristianSilval_abel_2 = uilabel (app.ONLINETab).
app.DesingbyChristianSilval_abel_2.FontAngle = 'italic'; app.DesingbyChristianSilval_abel_2.Position = [343
38 139 22].
app.DesingbyChristianSilval_abel_2.Text = 'Desing          by Christian Silva';

% Create Image_16
app.Image_16 = uiimage (app.ONLINETab).
app.Image_16.Position = [470 41 40 18].
app.Image_16.ImageSource = fullfile(pathToMLAPP,'Copyright.png').

% Create Image_17
app.Image_17 = uiimage (app.ONLINETab).
app.Image_17.Position = [-1 37 55 27].
app.Image_17.ImageSource = fullfile (pathToMLAPP,'Matlab.png').

% Show the figure after all components are created app.UIFigure.Visible = 'on'.
end
end

% App creation and deletion methods (Access = public)

% Construct app function app = AplicacionI

% Create UIFigure and components createComponents(app)

% Register the app with App Designer registerApp (app, app.UIFigure)

% Execute the startup function runStartupFcn (app_, @startupFcn)
if nargin == 0 clear app
end
end

% Code that executes before app deletion function delete(app)

% Delete UIFigure when app is deleted delete (app.UIFigure)
end
end

```

A.3. Código de Pruebas Preliminares

```
clc
clear all
I = imread('personas1.jpg');
[bboxes, scores] = detectPeopleACF(I); RGB=insertObjectAnnotation
(I,"rectangle", bboxes,scores); puntomediox=zeros(size(bboxes,1),1).
puntomedioy=zeros(size(bboxes,1),1); distancias=zeros (size (bboxes,
1)-1,1);
for i=1: size(bboxes,1)
    puntomediox (i, 1) =bboxes (i, 3)/2+bboxes (i, 1); puntomedioy (i,
    1) =bboxes(i,4)/2+bboxes(i,2).
    if i>=2 distancias(i-1,1) =sqrt ((puntomediox (i-1, 1)-
    puntomediox(i,1)) ^2 +(puntomedioy(i-1,1)-puntomedioy(i,1)) ^2);
    end
end
for j=1
end
end
RGB1 = insertMarker (RGB, [puntomediox puntomedioy],"x-mark"> "Size",5
    /'Color"/'red");
imshow (RGB);
```

A.4. Capturas del Sistema en funcionamiento

