



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES
INDUSTRIALES

“DISEÑO Y CONSTRUCCIÓN DE NODOS INTELIGENTES PARA
DETECCIÓN DE ARMAS DENTRO DE UNA RED DE VIDEO-
VIGILANCIA UTILIZANDO VISIÓN ARTIFICIAL”

Trabajo de Titulación

Tipo: Dispositivo Tecnológico

Presentado para optar al grado académico de:

INGENIERO EN ELECTRÓNICA, CONTROL Y REDES
INDUSTRIALES

AUTOR: FERNANDO PATRICIO OÑATE MIRANDA

TUTOR: ING. EDWIN VINICIO ALTAMIRANO SANTILLÁN.

Riobamba – Ecuador

2020

@ 2020, Fernando Patricio Oñate Miranda

Se autoriza la reproducción total o parcial, con fines académicos, por cualquier medio o procedimiento, incluyendo la cita bibliográfica del documento, siempre y cuando se reconozca el Derecho de Autor.

Yo, Fernando Patricio Oñate Miranda, declaro que el presente trabajo de titulación es de mi autoría y los resultados del mismo son auténticos. Los textos en el documento que provienen de otras fuentes están debidamente citados y referenciados.

Como autor (a) asumo la responsabilidad legal y académica de los contenidos de este trabajo de titulación; El patrimonio intelectual pertenece a la Escuela Superior Politécnica de Chimborazo.

Riobamba, 22 de enero de 2020



Fernando Patricio Oñate Miranda

180497066-1

ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
CARRERA DE INGENIERÍA ELECTRÓNICA EN CONTROL Y REDES
INDUSTRIALES

El Tribunal de Trabajo de Titulación certifica que: El trabajo de titulación: **“DISEÑO Y CONSTRUCCIÓN DE NODOS INTELIGENTES PARA DETECCIÓN DE ARMAS DENTRO DE UNA RED DE VIDEO-VIGILANCIA UTILIZANDO VISIÓN ARTIFICIAL”**, de responsabilidad del señor **FERNANDO PATRICIO OÑATE MIRANDA**, ha sido minuciosamente revisado por los Miembros del Tribunal de Trabajo de Titulación, quedando autorizada su presentación.

| | FIRMA | FECHA |
|--|--------------|--------------|
| Ing. Wilson Armando Zuñiga Vinuesa. PRESIDENTE DEL TRIBUNAL | | |
| Ing. Edwin Vinicio Altamirano Santillán. DIRECTOR DE TRABAJO DE TITULACIÓN | | |
| Ing. José Luis Tinajero León MIEMBRO DE TRIBUNAL | | |

DEDICATORIA

A mis padres Holger y Teresa por ser el pilar fundamental de mi vida, en toda mi educación, tanto de vida como académica, su apoyo incondicional a través del tiempo. Con amor el presente triunfo es para ustedes.

Fernando

AGRADECIMIENTO

Agradezco a Dios por haberme guiado y acompañado a lo largo de mi carrera, por ser la mayor fortaleza en los momentos de debilidad, a mi familia quienes de una u otra manera han contribuido a la realización de mi trabajo de titulación, por haberme dado su fuerza y apoyo incondicional, me han ayudado a llegar hasta donde estoy ahora, logrando cumplir mi meta. Gracias a todos.

Fernando

TABLA DE CONTENIDO

| | |
|---|------|
| ÍNDICE DE TABLAS | viii |
| ÍNDICE DE FIGURAS | x |
| ÍNDICE DE GRÁFICOS | xii |
| ÍNDICE DE ANEXOS..... | xiii |
| ÍNDICE DE ECUACIONES..... | xiv |
| ÍNDICE DE ABREVIATURAS | xv |
| RESUMEN | xvi |
| SUMMARY | xvii |
| INTRODUCCIÓN | 1 |
| FORMULACIÓN DEL PROBLEMA | 2 |
| SISTEMATIZACIÓN DEL PROBLEMA | 2 |
| JUSTIFICACIÓN TEÓRICA..... | 2 |
| JUSTIFICACIÓN APLICATIVA..... | 3 |
| OBJETIVOS..... | 4 |
| OBJETIVO GENERAL | 4 |
| OBJETIVOS ESPECIFICOS | 4 |
| CAPÍTULO I | |
| 1 MARCO TEÓRICO REFERENCIAL..... | 5 |
| 1.1 Nodos | 5 |
| 1.2 Detección de armas..... | 5 |
| 1.3 Sistemas de seguridad electrónicos | 5 |
| <i>Sistemas CCTV.....</i> | 6 |
| 1.4 Visión Artificial..... | 6 |
| <i>Componentes de la visión artificial</i> | 6 |
| <i>Etapas de la visión artificial.....</i> | 15 |
| 1.5 Adquisición de imágenes para el entrenamiento del clasificador..... | 19 |
| 1.6 Pre procesamiento de imágenes para el entrenamiento del clasificador | 20 |
| 1.7 Ubicación de los nodos de vigilancia | 23 |
| 1.8 Tarjetas embebidas | 24 |
| 1.9 Acceso a los servicios SMS | 26 |
| 1.10 Acceso a los servicios de una base datos | 27 |
| CAPÍTULO II | |
| 2 PROPUESTA Y DISEÑO DEL PROTOTIPO..... | 28 |

| | | |
|---------------------|--|-----------|
| 2.1 | Recopilación de imágenes para el entrenamiento..... | 28 |
| 2.2 | Pre procesamiento de las imágenes..... | 29 |
| 2.3 | Marcar la posición del objeto dentro de la imagen positiva..... | 30 |
| 2.4 | Creación del vector de entrenamiento..... | 31 |
| 2.5 | Obtención del clasificador Haar..... | 32 |
| 2.6 | Creación del archivo XML..... | 34 |
| 2.7 | Algoritmo del detector..... | 35 |
| 2.8 | Implementación del sistema de detección..... | 37 |
| | <i>Ubicación de los nodos.....</i> | <i>39</i> |
| CAPÍTULO III | | |
| 3 | VALIDACIÓN DEL PROTOTIPO..... | 45 |
| 3.1 | Prueba de precisión y sensibilidad del detector..... | 45 |
| 3.2 | Prueba del tiempo de procesamiento..... | 61 |
| 3.3 | Prueba del tiempo de envío y recepción de la alerta GSM..... | 63 |
| 3.4 | Prueba del tiempo de envío y almacenamiento de la imagen en la base de datos de Gmail..... | 64 |
| 3.5 | Prueba de consumo total de corriente..... | 66 |
| CAPITULO IV | | |
| 4 | EVALUACIÓN ECONÓMICA..... | 68 |
| 4.1 | Relación beneficio/costo..... | 68 |
| | CONCLUSIONES..... | 70 |
| | RECOMENDACIONES..... | 71 |
| GLOSARIO | | |
| BIBLIOGRAFÍA | | |
| ANEXOS | | |

ÍNDICE DE TABLAS

| | | |
|--------------------|---|----|
| Tabla 1-1: | Comparación de cámaras web..... | 10 |
| Tabla 2-1: | Comparación de cámaras IP..... | 11 |
| Tabla 3-1: | Comparación de módulos de cámaras para tarjetas embebidas..... | 12 |
| Tabla 4-1: | Comparación de cámaras de teléfonos inteligentes..... | 13 |
| Tabla 5-1: | Matriz de confusión..... | 19 |
| Tabla 6-1: | Programas para el redimensionamiento de imágenes en masa..... | 21 |
| Tabla 7-1: | Programas para recorte de imágenes..... | 21 |
| Tabla 8-1: | Hardware para entrenamientos más usados..... | 22 |
| Tabla 9-1: | Software para entrenamientos más usados..... | 22 |
| Tabla 10-1: | Características de los Arduinos..... | 25 |
| Tabla 11-1: | Características de los modelos de Raspberry Pi..... | 25 |
| Tabla 12-1: | Comparación de módulos GSM..... | 26 |
| Tabla 13-1: | Comparación de bases de datos..... | 27 |
| Tabla 1-2: | Parámetros formar el vector de entrenamiento..... | 31 |
| Tabla 2-2: | Parámetros para formar el clasificador..... | 32 |
| Tabla 3-2: | Parámetros que se obtienen en el entrenamiento..... | 33 |
| Tabla 4-2: | Librerías usadas en el algoritmo detector..... | 36 |
| Tabla 1-3: | Resultados con variable a plena luz del día oficina..... | 47 |
| Tabla 2-3: | Matriz de confusión con variable plena luz de día oficina..... | 48 |
| Tabla 3-3: | Resultados con variable de día nublado en oficina..... | 48 |
| Tabla 4-3: | Matriz de confusión con variable de día nublado en oficina..... | 49 |
| Tabla 5-3: | Resultados con variable de día con lluvia en oficina..... | 49 |
| Tabla 6-3: | Matriz de confusión con variable día con lluvia en oficina..... | 50 |
| Tabla 7-3: | Resultados con variable a plena luz del día en automóvil..... | 52 |
| Tabla 8-3: | Matriz de confusión con variable plena luz en automóvil..... | 53 |
| Tabla 9-3: | Resultados con variable de día nublado en automóvil..... | 53 |
| Tabla 10-3: | Matriz de confusión con variable día nublado en automóvil..... | 54 |
| Tabla 11-3: | Resultados con variable de día con lluvia en automóvil..... | 54 |
| Tabla 12-3: | Matriz de confusión con variable día con lluvia en automóvil..... | 55 |
| Tabla 13-3: | Resultados con variable a plena luz del día en parque..... | 57 |
| Tabla 14-3: | Matriz de confusión con variable a plena luz en parque..... | 57 |
| Tabla 15-3: | Resultados con variable de día nublado en parque..... | 58 |
| Tabla 16-3: | Matriz de confusión con variable de día nublado en parque..... | 58 |
| Tabla 17-3: | Resultados con variable de día con lluvia en parque..... | 59 |

| | | |
|--------------------|--|----|
| Tabla 18-3: | Matriz de confusión con variable día con lluvia en parque | 59 |
| Tabla 19-3: | Resumen de resultados | 60 |
| Tabla 20-3: | Resultados de la prueba de detección de objetos..... | 61 |
| Tabla 21-3: | Resultados de la prueba de procesamiento de imágenes | 62 |
| Tabla 22-3: | Resultados de la prueba de envío y recepción alerta GSM | 64 |
| Tabla 23-3: | Resultados de la prueba de envío y almacenamiento de Gmail..... | 65 |
| Tabla 24-3: | Resultados de la prueba de consumo de corriente del dispositivo..... | 67 |
| Tabla 1-4: | Análisis de costos | 68 |
| Tabla 2-4: | Relación beneficio/costo | 68 |

ÍNDICE DE FIGURAS

| | | |
|---------------------|--|----|
| Figura 1-1. | Componentes de la Visión Artificial..... | 7 |
| Figura 2-1. | Características Objetivo Fotográfico..... | 8 |
| Figura 3-1. | Objetivos en función del ángulo..... | 9 |
| Figura 4-1. | Etapas de la visión artificial..... | 15 |
| Figura 5-1. | Patrones básicos de características HAAR..... | 17 |
| Figura 6-1. | Conversión de imagen con detector HOG..... | 17 |
| Figura 7-1. | Ejemplo del cálculo LBP..... | 18 |
| Figura 8-1. | Imagen positiva..... | 20 |
| Figura 9-1. | Imagen negativa..... | 20 |
| Figura 10-1: | Acceso a los servicios SMS..... | 26 |
| Figura 1-2. | Recopilación de imágenes positivas para el entrenamiento..... | 28 |
| Figura 2-2. | Recopilación de imágenes negativas para el entrenamiento..... | 29 |
| Figura 3-2. | Localización del objeto en la imagen positiva..... | 30 |
| Figura 4-2. | Obtención del clasificador..... | 33 |
| Figura 5-2. | Obtención del clasificador..... | 34 |
| Figura 6-2. | Clasificador XML final generado..... | 34 |
| Figura 7-2. | Diagrama de flujo del algoritmo detector..... | 35 |
| Figura 8-2. | Implementación del sistema en Raspberry PI 3 B+..... | 38 |
| Figura 9-2. | Conexión del módulo GSM con la Raspberry PI..... | 38 |
| Figura 10-2. | Base de datos creada para la recopilación de imágenes detectadas..... | 39 |
| Figura 11-3. | Ubicación de los nodos en la oficina..... | 40 |
| Figura 12-3. | Ubicación de los nodos en el automóvil..... | 40 |
| Figura 13-3. | Ubicación de los nodos en un parque..... | 41 |
| Figura 14-2. | Sistema implementado (oficina)..... | 42 |
| Figura 15-2. | Sistema implementado (oficina)..... | 42 |
| Figura 16-2. | Sistema implementado (automóvil)..... | 43 |
| Figura 17-2. | Sistema implementado (parque)..... | 43 |
| Figura 18-2. | Sistema implementado (parque)..... | 44 |
| Figura 1-3. | Detección positiva de rifle y metralleta en ambiente de oficina..... | 45 |
| Figura 2-3. | Detección positiva de pistola y recortada en ambiente de oficina..... | 46 |
| Figura 3-3. | Detección positiva de cuchillo y pistola en ambiente de oficina..... | 46 |
| Figura 4-3. | Detección positiva de escopeta y pistola en ambiente de oficina..... | 46 |
| Figura 5-3. | Recorte facial de la imagen en ambiente de oficina..... | 47 |
| Figura 6-3. | Detección positiva de metralleta y recortada en ambiente de automóvil..... | 50 |

| | | |
|---------------------|--|----|
| Figura 7-3. | Detección positiva de pistola y cuchillo en ambiente de automóvil..... | 51 |
| Figura 8-3. | Detección positiva de escopeta y cuchillo en ambiente de automóvil | 51 |
| Figura 9-3. | Detección positiva de pistola y rifle en ambiente de automóvil..... | 51 |
| Figura 10-3. | Recorte facial de la imagen en ambiente de automóvil..... | 52 |
| Figura 11-3. | Detección positiva de metralleta y recortada en ambiente de parque..... | 55 |
| Figura 12-3. | Detección positiva de cuchillo y escopeta en ambiente de parque..... | 56 |
| Figura 13-3. | Detección positiva de rifle y pistola en ambiente de parque | 56 |
| Figura 14-3. | Recorte facial de la imagen en ambiente de parque | 57 |
| Figura 15-3. | Prueba de medición del tiempo de procesamiento | 62 |
| Figura 16-3. | Medición de tiempo de envío y recepción de alerta GSM | 63 |
| Figura 17-3. | Medición de tiempo de envío y almacenamiento en Gmail | 65 |
| Figura 18-3. | Medición del consumo de corriente del dispositivo | 66 |

ÍNDICE DE GRÁFICOS

| | | |
|---------------------|---|----|
| Gráfico 1-3. | Resumen de la prueba de precisión y sensibilidad | 61 |
| Gráfico 2-3. | Resumen de la prueba de tiempo de procesamiento | 63 |
| Gráfico 3-3. | Resumen de la prueba tiempo de envío y recepción GSM | 64 |
| Gráfico 4-3. | Resumen de la prueba de envío y almacenamiento en Gmail | 66 |

ÍNDICE DE ANEXOS

| | |
|-----------------|----------------------------------|
| ANEXO A. | MANUAL DE USUARIO |
| ANEXO B. | CÓDIGO FUENTE |
| ANEXO C. | TOMA DE MUESTRAS |
| ANEXO D. | PRUEBAS DE FUNCIONAMIENTO |

ÍNDICE DE ECUACIONES

| | | |
|-----------------------|---|----|
| Ecuación 1-1. | Fórmula de cálculo de precisión | 19 |
| Ecuación 2-1. | Fórmula de cálculo de TVP | 19 |
| Ecuación 1-2. | Fórmula de cálculo de parámetros de entrenamiento..... | 31 |
| Ecuación 1-3. | Precisión con variable a plena luz de día en oficina | 48 |
| Ecuación 2-3. | Sensibilidad con variable a plena luz de día en oficina..... | 48 |
| Ecuación 3-3. | Precisión con variable de día nublado en oficina..... | 49 |
| Ecuación 4-3. | Sensibilidad con variable de día nublado en oficina..... | 49 |
| Ecuación 5-3. | Precisión con variable de día con lluvia en oficina..... | 50 |
| Ecuación 6-3. | Sensibilidad con variable de día con lluvia en oficina..... | 50 |
| Ecuación 7-3. | Precisión con variable a plena luz de día en automóvil | 53 |
| Ecuación 8-3. | Sensibilidad con variable a plena luz de día en automóvil | 53 |
| Ecuación 9-3. | Precisión con variable de día nublado en automóvil..... | 54 |
| Ecuación 10-3. | Sensibilidad con variable de día nublado en automóvil..... | 54 |
| Ecuación 11-3. | Precisión con variable de día con lluvia en automóvil..... | 55 |
| Ecuación 12-3. | Sensibilidad con variable de día con lluvia en automóvil..... | 55 |
| Ecuación 13-3. | Precisión con variable a plena luz de día en parque..... | 58 |
| Ecuación 14-3. | Sensibilidad con variable a plena luz de día en parque..... | 58 |
| Ecuación 15-3. | Precisión con variable de día nublado en parque | 59 |
| Ecuación 16-3. | Sensibilidad con variable de día nublado en parque | 59 |
| Ecuación 17-3. | Precisión con variable de día con lluvia en parque | 60 |
| Ecuación 18-3. | Sensibilidad con variable de día con lluvia en parque | 60 |

ÍNDICE DE ABREVIATURAS

| | |
|---------------|--|
| RAM: | Read Only Memory / Memoria solo de Lectura |
| FLANN: | Fast Library for Approximate Nearest Neighbors / Biblioteca rápida para los vecinos más cercanos aproximados |
| FPS: | Frames for Second / Cuadros por segundo |
| USB: | Universal Serial Bus / Bus Universal en Serie |
| ANT: | National Transit Agency / Agencia Nacional de Tránsito |
| GIF: | Graphics Interchange Format / Formato de Intercambio de Gráficos |
| GPIO: | General Purpose Input-Output / Entrada-salida de uso general |
| ISO: | International Organization for Standardization / Organización Internacional para la Estandarización. |
| PWM: | Pulse Width Modulation / Modulación por Ancho de Pulsos |
| RAM: | Random Access Memory / Memoria de Acceso Aleatorio |
| RGB: | Red, Green, Blue / Rojo, Verde y Azul |
| SPI: | Serial Peripheral Interface / Interfaz Periférica Serial |
| SURF: | Speeded Up Robust Features / Características Robustas Aceleradas |
| UART: | Universal Asynchronous Receiver Transmitter / Receptor-Transmisor Asíncrono Universal |

RESUMEN

En el presente trabajo de titulación tuvo como objetivo diseñar e implementar nodos inteligentes para detección de armas dentro de una red de video-vigilancia utilizando visión artificial. Para lo cual se implementó nodos de detección mediante los requisitos necesarios para su ejecución, así como el acople de varios dispositivos electrónicos. Se implementó un dispositivo de captura de imágenes que permite obtener las entradas del sistema, analizando dichas imágenes con herramientas de visión artificial orientadas al aprendizaje automático para seleccionar características de los objetos con ayuda de librerías de OpenCV y clasificadores HAAR los cuales se ejecutan sobre un software libre Linux, se ha diseñado un algoritmo detector el cual se ha programado sobre una tarjeta embebida de bajo costo. El sistema envía una alerta GSM al momento que se detecta un arma. Para lo cual se envía la imagen hacía una base de datos de Gmail para su análisis tomando en cuenta que, si se identifica un rostro dentro de la imagen que contiene el objeto en estudio se procederá a realizar un recorte facial y un acercamiento para identificar de mejor manera al individuo. Con las pruebas se determinó que a pesar que los objetos en análisis son pequeños y de difícil detección a comparación de vehículos o personas, los nodos cumplen con la detección de armas en diferentes ambientes. Se concluye que el sistema logra enviar la alerta GSM en un tiempo promedio de 6.5 segundos, enviar y almacenar la imagen detectada en la base de datos de Gmail en un tiempo promedio de 10 segundos lo que nos permite tener una detección en tiempo real, se recomienda utilizar imágenes propias de cada ambiente en análisis para obtener clasificadores con mayor fuerza de detección.

Palabras clave: <VISIÓN ARTIFICIAL>, <CLASIFICADOR HAAR (ESTRUCTURA DE FUNCIONES)>, <VIDEO VIGILANCIA>, <SISTEMA GLOBAL PARA LAS COMUNICACIONES (GSM)>, <APRENDIZAJE AUTOMÁTICO>, <OPENCV (SOFTWARE)>.



SUMMARY

In this titling work, the objective was to implement intelligent nodes for weapons detection within a video surveillance network using artificial vision. For which nodes of detection were designed determining the requirements and implementing them through the coupling of several electronic devices, an image capture device was implemented that allows to obtain the inputs of the system, analyzing said images with characteristics of the objects with the help of OpenCV libraries and HAAR classifiers which run on free Linux software, a detector algorithm has been designed which has been programmed on a low-cost embedded card. The system sends a GSM alert when a weapon is detected, then sends the image to a Gmail database for analysis taking into account that, if a face is identified within the image that contains the object of study, a facial cut and an approach will be carried out to better identify the individual. With the tests it was determined that although the objects under analysis are small and difficult to detect compared to vehicles or people, the nodes comply with the provision of weapons in different environments. It is concluded that the system manages to send the GSM alert in an average time of 6.5 seconds, send and store the image detected in the Gmail database in an average time of 10 seconds, which allows us to have a real-time detection, It recommends using own images of each environment in analysis to obtain classifiers with greater detection force.

Key words: <ARTIFICIAL VISION>, <HAAR CLASSIFIER (FUNCTIONS STRUCTURE)>, <VIDEO SURVEILLANCE>, <GLOBAL SYSTEM FOR COMMUNICATIONS (GSM)>, <AUTHOMATIC LEARNING>, <OPENCV (SOFTWARE)>.



INTRODUCCIÓN

En la actualidad en la mayoría de los países los sistemas de seguridad están instalados en ambientes distintos los cuales están compuestos por cámaras de vigilancia, son instalados para prevenir situaciones de peligro como asaltos a mano armada siendo cada vez más orientados a mejorar la seguridad social y mayormente utilizados en entidades comerciales. En el caso de nuestro país según Instituto Nacional de Estadísticas y Censos (2018, p. 2), se realiza un monitoreo por parte de un personal capacitado de eventos suscitados convirtiendo a este sistema en una herramienta pasiva, el cual no presta la ayuda necesaria al momento de darse una situación de peligro, el sistema guarda sus registros para usarlos posteriormente.

Según (Martín, 2012; ANT, 2017) para reducir este tipo de peligros una forma seria la detección de armas en un tiempo prudente el cual proporcione una mejor respuesta de ayuda por parte de las entidades de seguridad encargadas. Los eventos de este tipo duran aproximadamente 3 minutos o menos con lo cual varios de los casos no llegan a ser resueltos debido al problema de respuesta por parte del personal de seguridad encargado. Debido a su elevado costo, a las limitaciones humanas en el aspecto de monitoreo y a las limitaciones de tiempo, el monitoreo en vivo es una opción muy poco viable para la mayoría de instituciones que lo requieren especialmente para los pequeños negocios que no cuentan con los recursos necesarios.

La detección instantánea permitiría reducir los casos no resueltos de eventos de este tipo, notificando a los organismos encargadas de seguridad en el momento en el cual está sucediendo el evento y no luego como actualmente se lo viene realizando, además de ello guardar en una base de datos las imágenes detectadas por el sistema para su posterior análisis. Así obtener una función disuasiva hacia quienes realizan aquellas acciones. Con ello se busca mejorar el perfil de seguridad social y obtener una mayor confiabilidad por parte de los usuarios en los sistemas de video vigilancia. Si bien el presente proyecto tiene la limitación de solo detectar pequeñas armas en videos posteriormente se lo podría aplicar para la detección de cualquier tipo de arma dentro de ambientes video vigilados en tiempo real.

FORMULACIÓN DEL PROBLEMA

Cómo diseñar y construir nodos inteligentes para detección de armas dentro de una red de video-vigilancia utilizando visión artificial.

SISTEMATIZACIÓN DEL PROBLEMA

¿Cuáles son los algoritmos y métodos de detección utilizados en visión artificial que permiten detectar armas y se puedan usar en el prototipo?

¿Cuáles son los requerimientos que deben cumplir los nodos utilizando el algoritmo de detección adecuado?

¿Cuál es el diseño necesario que cubra los requerimientos de los nodos a implementar?

¿Cuál es el software y hardware que permita implementar el diseño de los nodos?

¿Los nodos desarrollados cumplen con los requerimientos?

JUSTIFICACIÓN TEÓRICA

En múltiples países se han registrado un incremento de criminalidad donde la principal herramienta de atraco es un arma de fuego, de acuerdo con Statistic (2017, p. 2), Estados Unidos en el año 2017 se registraron 200.000 robos con armas de fuego de los cuales aproximadamente 66.000 casos cada año presentan lesiones fatales por arma de fuego. En Ecuador de acuerdo con el Instituto Nacional de Estadísticas y Censos (2018, p. 2), en el año 2018 la tendencia de robo a unidades económicas a crecido de manera considerable, esto dado a varios factores siendo uno de ellos la baja economía del país, dándose un total de aproximadamente 450 robos cada mes y un total de aproximadamente 5400 cada año esperándose un aumento para el año 2020, debido a que los sistemas de monitoreo en donde un personal observa son muy costosos y de baja eficiencia las entidades comerciales han optado por no utilizarlos dejando así más vulnerable la integridad del personal y de sus bienes. De acuerdo con R.K Tiwari (2015, p. 703-712), un operador de monitoreo en promedio perderá hasta un 45% de actividad de pantalla al transcurso de 12 minutos de continuo

monitoreo, a menudo que el tiempo aumenta tendrá una tasa de fallo de 90% al transcurso de 22 minutos.

Actualmente los sistemas de video vigilancia se utilizan para la obtención y análisis de datos importantes, coincidiendo con algunos autores (Martin, 2012; ANT, 2017) siendo orientados con mayor frecuencia con la seguridad privada y con menor frecuencia la seguridad pública. Además de ello los sistemas de video vigilancia se los localiza en sitios de trabajo con el objetivo de controlar la productividad de los trabajadores, así también de registrar datos relevantes que puedan darse en el transcurso de sus actividades. Dado al grado de inseguridad que presentan dichos sistemas las entidades comerciales han tratado de implementar sistemas de seguridad siendo principalmente cámaras de video vigilancia con monitoreo, lamentablemente son elementos pasivos que no proporcionan la ayuda necesaria al momento que se requiere la misma.

El proyecto proporcionará un sistema de seguridad capaz de detectar pequeñas armas al momento que se produzca un robo o un atraco y al emitir una alarma a las entidades encargadas para así socorrer a la víctima.

JUSTIFICACIÓN APLICATIVA

Es un sistema de detección de armas dentro de un ambiente video vigilado, el cual está integrado por nodos inteligentes. Cada nodo es capaz de captar información, procesarla y enviarla hacia la base de datos.

Para captar la información el sistema deberá monitorear el ambiente en tiempo real con ayuda del dispositivo de captura de imágenes, durante el procesamiento se busca en cada imagen el objeto en análisis, al momento que el sistema detecte el objeto será capaz de enviar una alerta GSM hacia los dispositivos vinculados al mismo, esto se lo realizará en el menor tiempo posible.

Para su posterior análisis el sistema realizará una captura de la imagen detectada, procediendo a su envío y almacenamiento en la base de datos de Gmail, así el personal encargado del sistema podrá reconocer las características en las cuales se detectó el arma y utilizar esta información como mejor les convenga.

OBJETIVOS

OBJETIVO GENERAL

Diseñar y construir nodos inteligentes para detección de armas dentro de una red de video-vigilancia utilizando visión artificial.

OBJETIVOS ESPECIFICOS

Investigar y obtener el algoritmo para la detección de armas mediante visión artificial.

Determinar los requerimientos que deben cumplir los nodos utilizando el algoritmo de detección adecuado.

Establecer el diseño que cubra los requerimientos de los nodos a implementar.

Seleccionar el software y hardware que permita implementar el diseño de los nodos.

Evaluar si los nodos desarrollados cumplen con los requerimientos.

CAPÍTULO I

1 MARCO TEÓRICO REFERENCIAL

1.1 Nodos

Se refiere a la conexión de al menos dos elementos entre sí, ya sean informáticos, electrónicos, eléctricos, hidráulicos, neumáticos, etc., en algunos casos existe la posibilidad que los nodos estén formados por elementos de distintas áreas, considerando la capacidad de obtener información, procesarla y enviarla hacia el otro u otros nodos, abarcando información determinada que permiten una conexión hacia otros puntos. (Jorquera y Pérez, 2012, p.9)

1.2 Detección de armas

La detección de armas por lo general aparece en situaciones que demandan peligro, en la actualidad se puede decir que existen dos tipos:

- Detección electrónica que se lo realiza mediante cámaras de seguridad implementados en transporte público, pequeños emprendimientos o negocios, entre otros. (Owuye 2020)
- Detección física, que implementa en su estructura guardias de seguridad. (Owuye 2020)

1.3 Sistemas de seguridad electrónicos

Un sistema de seguridad electrónico viene a ser la interconexión de recursos, redes y dispositivos cuyo principal objetivo es brindar la protección de la integridad de las personas y su entorno, previniéndolas de posibles peligros y presiones externas que alteren la normalidad de su entorno. (Cámara Nacional de Comercio y Servicios, 2017, p. 7)

Los sistemas de seguridad electrónicos desde su inicio fueron destinados a disuadir, reducir y prevenir un acontecimiento delictivo mismo que resulta disminuir la probabilidad de pérdidas humanas, económicas en establecimientos comerciales públicos y privados. El tipo de seguridad

electrónica que se realiza para detectar acontecimientos en los cuales se puedan detectar armas vienen a ser los sistemas CCTV (Closed Circuit Television). (Hidalgo, 2012, p. 14)

Sistemas CCTV

Es un sistema de video vigilancia que en su desarrollo adopto la supervisión, el control y el riesgo eventual de actividad física dentro de un entorno. El sistema está compuesto por uno o varios dispositivos de captura de imágenes conectadas a uno o varios monitores, lo cual permite tener una red de video vigilancia. (Hidalgo, 2012, p. 16)

El sistema CCTV en la actualidad es el más usado para prevenir acontecimientos de peligro como asaltos en donde hacen presencia armas, lamentablemente posee desventajas como es el alto costo, mantenimiento complejo y la necesidad de la vigilancia constante por parte de los operadores. (Hidalgo, 2012, p. 16)

1.4 Visión Artificial

Otra manera por la cual es mundialmente conocida es como visión por computador, misma que, mediante el uso de la programación detalla características relevantes de un objeto contenido en una imagen, llegando a ello mediante varios procesos de discriminación. Comúnmente un sistema de visión artificial está constituido por un dispositivo que refleja la luz sobre una muestra en donde la imagen es capturada y posteriormente procesada e interpretada por un computador. (Gallo y Fernández, 2012, p.16)

Componentes de la visión artificial

Los sistemas de visión artificial presentan cuatro componentes: procesador, dispositivos de captura de imágenes, iluminación y objeto, lo cual se muestran en la figura 1-1.

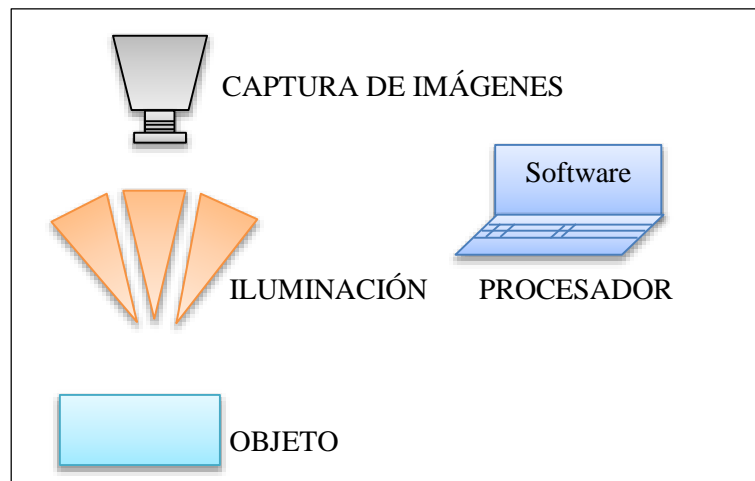


Figura 1-1. Componentes de la Visión Artificial

Realizado por: OÑATE, Fernando 2020

1.4.1.1 *Procesador de imágenes*

Es aquel que procesa la imagen tomada por el dispositivo de captura digital, puede presentarse de varias formas entre las más conocidas entran los sistemas embebidos o un computador. El mismo tiene la capacidad de realizar el procesamiento de imágenes además de técnicas de visión artificial y su desempeño se verá directamente relacionado a las características del sistema además de ello el lenguaje de programación utilizado también desempeña un papel importante en la velocidad del sistema. (Paguay & Valarezo, 2018, p. 17)

1.4.1.2 *Dispositivo de captura de imágenes*

Es uno de los elementos más importantes e imprescindibles de la visión artificial y tiene la función de realizar la captura de imágenes del objeto misma que se la realiza mediante la captura de luz reflejada sobre la superficie de un chip de silicio fotosensible. Actualmente podemos encontrar una gama muy amplia de estos dispositivos de captura de imagen, sin importar su marca o el tipo todos estos dispositivos comparten por lo general sus componentes básicos los cuales se detallan a continuación. (Carrillo, 2011)

- **Objetivo fotográfico:** Consiste en un sistema óptico el cual está conformado por un conjunto de lentes, mismos que pueden realizar divergencia o convergencia con el único objetivo de

formar la imagen que va a capturar el sensor, tienen una principal característica que es su longitud focal la cual viene expresada en milímetros misma que establece el ángulo de por segundo, cobertura, así como el factor de ampliación del objetivo a ser capturado, como se puede observar en la figura 2-1. (Landoni, 2015, p.5)

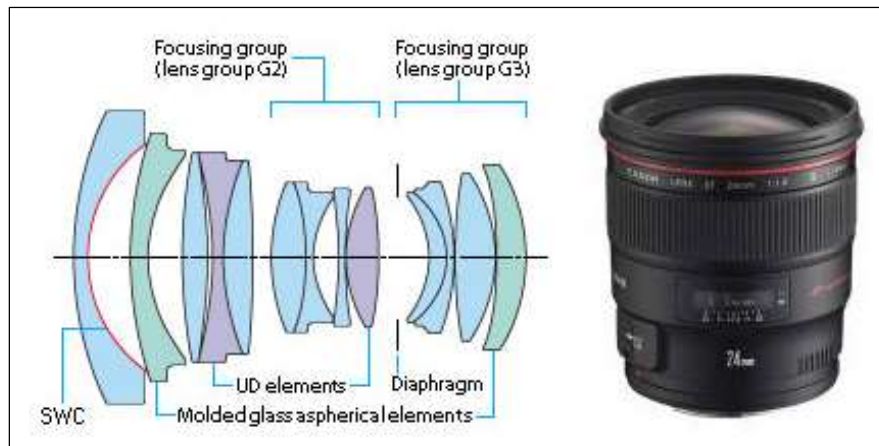


Figura 2-1. Características Objetivo Fotográfico

Fuente: (<http://cefire.edu.gva.es>)

Se define al ángulo de cobertura como el campo visual que abarca el objetivo, cuanto menor es la longitud focal mayor es el ángulo de cobertura de un objetivo. (Landoni, 2015, p.6)

Los objetivos en función del ángulo de cobertura se muestran en la figura 3-1 y se pueden clasificar en:

- **Normales.** Su ángulo de cobertura está en torno a los 45 grados muy similar al ángulo de visión humana. (Landoni, 2015, p.5)
- **Angulares.** Su ángulo de cobertura está en torno de 60 a 90 grados y existen objetivos angulares que poseen un ángulo de cobertura de 180 grados son comúnmente conocidos como ojos de pez. (Landoni, 2015, p.5)
- **Teleobjetivos.** Su ángulo de cobertura está en torno a los 8 y 28 grados. (Landoni, 2015, p.5)

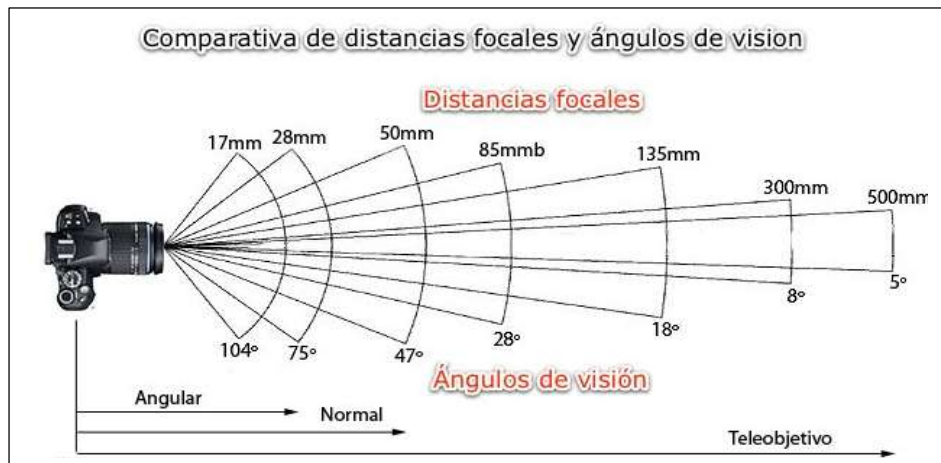


Figura 3-1. Objetivos en función del ángulo

Fuente: <https://n9.cl/cmsg>

- **Sensor fotográfico:** Es el elemento encargado de captar toda la información del escenario, en otras palabras, recoge toda la luz que el objetivo sea capaz de llevar hacia el convirtiéndola en señales eléctricas, el sensor está formado por píxeles sensibles a la luz, la calidad de la imagen dependerá de gran manera de la cantidad de megapíxeles que posea la misma. (García, 2015. p.36)
- **Dispositivo de conversión ADC:** Es aquel que transforma las señales analógicas en digitales mismas que serán compatibles con el lenguaje del computador. (García, 2015. p.36)
- **Dispositivo procesador digital de señales:** Controla y regula el sistema electrónico de la cámara y procesa las imágenes digitalizadas para enviarlas a un dispositivo de almacenamiento. (García, 2015. p.36)
- *Adquisición de imágenes mediante cámaras web.*

La cámara web es una pequeña cámara digital que está conectada a una computadora y que fue construida con el objetivo de capturar imágenes para transmitir las a través de internet, dependiendo del modelo y marca de la cámara web la calidad de las imágenes irán variando, en la tabla 1-1 se muestra una comparación de las cámaras web más económicas de gama media. (Rosales, 2017, p. 29)

Tabla 1-1: Comparación de cámaras web.

| Nombre | Imagen | Descripción | Precio (USD) |
|--------------------------|---|---|--------------|
| Logitech C920 HD Pro |  | Dispone de 15MP, que permite grabar en Full HD 1080P. Es compatible con Windows 7 o posterior, MacOS 10.6 o posterior, Chrome OS y Android V5.0 o superior. | 70,00 |
| Papalook PA452 Full HD |  | Resolución de 1080P, es capaz de capturar video en HD a 720P y tomar fotografías en 1080P. Es compatible con Windows, desde la versión XP SP2. | 30,00 |
| Logitech C525 HD |  | Tiene 8MP y grabar videos en HD 720p en formato panorámico, es compatible con sistemas Windows SP2 en adelante. | 40,00 |
| Logitech C525 HD |  | Graba a 720P, en una resolución panorámica 16.9 y tecnología, incluso en situaciones de poca luz. | 42,00 |
| Logitech C922 Pro Stream |  | Toma fotos y graba video a 720P en 30fps. Es compatible con Windows XP en adelante y Mac OS X 10.4 o posterior | 33,00 |

Realizado por: OÑATE, Fernando 2020







Fuente: <https://wifibit.com/mejores-webcams-baratas/>

- **Adquisición de imágenes mediante cámaras IP**

También conocidas como cámaras de red, al igual que un ordenador la cámara de red posee su propia dirección IP misma que se conecta directamente a la red, las cámaras IP se las puede usar en cualquier parte que exista conexión a la red, esta es la característica que las diferencia de las cámaras web que solo pueden ser ejecutadas cuando están conectadas al ordenador mediante el

puerto USB o IEEE 1394, las cámaras de gama alta que cuentan con entradas para alarmas y salidas de relé, además de incorporar la detección de movimiento, en la tabla 2-1 se muestra una comparación de las cámaras IP más utilizadas actualmente . (Rosales, 2017, p. 30)

Tabla 2-1: Comparación de cámaras IP.

| Nombre | Imagen | Descripción | Precio (USD) |
|-------------------------------------|---|---|--------------|
| Bagotte Cámara IP WiFi |  | Posee una resolución 1920x1080P Captura las imágenes en 720P HD- | 60,00 |
| Lensoul Cámara IP |  | Su resolución de 1080P HD permite gran claridad de imagen. | 50,00 |
| APEMAN 1080P IP Cámara |  | La resolución 1920x1080P permite a la esta cámara IP garantizar imágenes de alta calidad. Con ángulo horizontal de 355 grados y vertical de 80. | 35,00 |
| Gakov GA829X |  | Los 1080P en 25fps permiten una claridad, permite una excelente visión nocturna de hasta 8 metros. | 44,00 |
| QZT cámara de vigilancia WiFi 1080P |  | Resolución en 1080p. Contiene todos los elementos para brindar un satisfactorio monitoreo gracias a una rotación de 360 grados, | 40,00 |
| Mibao 1080P |  | Ofrece alta resolución en 1080p, controla el espacio monitoreado desde la palma de tu teléfono inteligente o Tablet. | 45,00 |



Realizado por: OÑATE, Fernando 2020

Fuente: <https://www.alarmadoo.com/es/camaras-ip/>

- *Adquisición de imágenes mediante módulos de cámaras de sistemas embebidos*

Los módulos de cámara para sistemas embebidos son la nueva tendencia de los sistemas informáticos, poseen varias ventajas sobre las descritas anteriormente como bajo costo, pequeño tamaño y alta calidad, en la tabla 3-1 se muestra los módulos de cámara para tarjetas embebidas más usadas. (Álvarez, 2014, p. 30)

Tabla 3-1: Comparación de módulos de cámaras para tarjetas embebidas.

| Nombre | Imagen | Descripción | Precio (USD) |
|--|---|--|--------------|
| See3CAM_ CU55M |  | 5.0 MP monocromo. +Excelente imagen NIR rendimiento. Eficiencia cuántica mayor del 90% | 150,00 |
| See3CAM_ CU135 |  | 13MP con buen rendimiento. Tiene un soporte de lente con montura que se usa de acuerdo a sus requerimientos. | 180,00 |
| See3CAM_ 130 – 4K Autofocus |  | 13MP basada que puede transmitir en Full HD (1920x1080) a 60 fps. | 240,00 |
| Raspberry PI Camera Board v 1.3 |  | Resolución de 5MP o grabación de video HD de 1080p a 30 fps presenta 5MP (2592–1944 píxeles). | 20,00 |
| The Raspberry Pi Camera Module v2 |  | El módulo de cámara v2 tiene un sensor IMX219 de 8 megapíxeles producido por la empresa Sony. | 30,00 |
| Mgsystem Modulo Camara Vga Ov7670 |  | Presentado por la empresa Omnivision, resolución de 640x480 pixeles (0.3MPx) | 8,00 |
| Mivm Camara Ov2640 |  | Soporte de salida para formatos RGB, RGB formatos de compresión JPEG que permite el redimensionamiento de la imagen. | 52,00 |

Realizado por: ONATE, Fernando 2020

Fuente: <https://www.e-consystems.com/See3CAM-USB-3-Camera.asp>

- *Adquisición de imágenes mediante teléfonos inteligentes*

Los teléfonos inteligentes han revolucionado por completo los sistemas informáticos en la actualidad, poseen capacidades como adquirir y almacenar datos, mismos que pueden enviarse a través de la red, en la captura de imágenes los teléfonos inteligentes han llegado a un nivel bastante alto ya que poseen cámaras de gama alta con múltiples funcionalidades, en la tabla 4-1 se muestra la comparación de las cámaras de los teléfonos inteligentes más destacados. (Álvarez, 2014, p. 34)

Tabla 4-1: Comparación de cámaras de teléfonos inteligentes.

| Nombre | Imagen | Descripción | Precio (USD) |
|--------------------|---|--|--------------|
| HUAWEI P30 Pro |  | Posee una cámara de 40 MP, el sistema y trabaja sobre software libre. Es de gama alta y de precio relativamente bajo. | 900,00 |
| Samsung Galaxy S10 |  | Incorpora tres cámaras de 16 MP, y trabaja sobre software libre, uno de los teléfonos más buscados del mercado por su alta eficiencia. | 930,00 |
| Iphone 11 pro |  | Contiene tres cámaras de 12 MP y trabaja sobre software con licencia, es un teléfono de gama alta y muy conocido por, la nitidez de imagen que ofrece su cámara. | 1300,00 |

Realizado por: ONATE, Fernando 2020

Fuente: <https://andro4all.com/2018/12/mejores-moviles-gama-alta-android>

1.4.1.3 Iluminación de la imagen

Un factor muy importante a considerar dentro de visión artificial es la iluminación ya que es la parte más crítica dentro del sistema de visión. El propósito de la iluminación es controlar la forma en la cual se va a ver el objeto, cada objeto tiene su propia iluminación óptima, ya que no es lo mismo iluminar un objeto hecho de metal que iluminar una hoja de papel. (COGNEX, 2018)

1.4.1.4 Software de programación y visualización

Es necesario una vez que el sistema de procesamiento esté listo escoger un programa que soporte varias librerías y con ellas acceder a un conjunto de funciones que permitan realizar varias aplicaciones de visión artificial.

- Torch3vision: Estructurado en base al lenguaje de programación C++ y dispone de operaciones con imágenes básicas, se registra la versión 2.1 que se la desarrollo en el año 2007. (Rosales, 2017, p. 19)
- VLX: No es solo una única librería, más bien es un conjunto de librerías lo cual ofrece mayor cobertura en el área de visión artificial, se desarrolló con el principal objetivo de tener un sistema rápido y fiable. (Álvarez, 2014, p. 24)
- OpenCV: Es una biblioteca de librerías con funciones orientadas completamente al campo de la visión artificial, cuenta con más de 500 funciones que permiten su desarrollo en varias áreas como: reconocimiento de objetos, calibración de cámaras, entre otras. (Paredes & Guerrero, 2012, p. 20)
- Raspbian: Incorpora más de 35,000 paquetes, además incorpora un software de licencia libre Linux, el cual ha desarrollado un entorno muy amigable y funcional para la ejecución sobre tarjetas embebidas. (Ubuntu, 2015)
- Windows 10 IoT Core: Utiliza una estructura muy compacta y funcional, lo que da facilidad de construir dispositivos de bajo costo con menos recursos, pero de alta eficiencia, por lo cual se puede instalar en tarjetas embebidas actuales. (Microsoft, 2013)

Etapas de la visión artificial

Está compuesta por 5 etapas en las que el pre procesamiento la segmentación la extracción y el reconocimiento forman un solo bloque como se muestra en la figura 4-1.

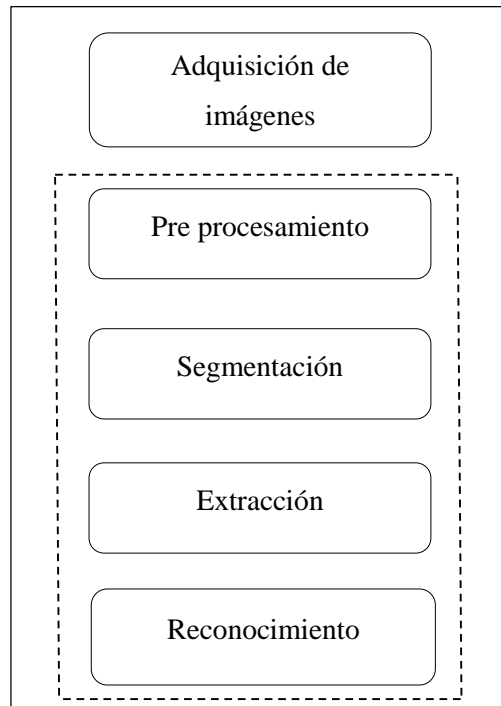


Figura 4-1. Etapas de la visión artificial

Realizado por: OÑATE, Fernando 2020

1.4.1.5 Adquisición de imágenes

El sensor de imagen convierte energía electromagnética en señales eléctricas, mismas que serán procesadas, mostradas e interpretadas como imágenes digitales. (Juan & Gustavo, s.f.)

1.4.1.6 Pre-procesamiento

Se llama así al proceso inicial de una imagen que no ha sido modificada, la imagen llega al computador en forma digital la cual se encuentra conformada por dígitos los cuales posteriormente se llamaran píxeles, donde 0 es negro y 255 es blanco. Cada pixel en color está

representado por 3 dígitos RGB y también están en el rango de 0 a 255 dependiendo directamente de la intensidad de color que los mismos posean. (Juan & Gustavo, s.f.)

1.4.1.7 Segmentación

Proceso en el cual la imagen es dividida en objetos de interés, se obtienen características convenientes para diferenciar un tipo de otro mediante los procesos de descripción.

Se entiende por descriptores a la extracción de características de imagen desde los componentes propios que describen a la imagen, uno de los factores importantes a considerar es el costo computacional que requiere, por ello, otros ámbitos como la resolución y la cantidad de imágenes intervienen de manera directa en la eficiencia de los descriptores. (Juan & Gustavo, s.f.)

1.4.1.8 Extracción y selección de características

Como su nombre lo indica se selecciona y se extrae características propias de interés, reduciendo así la información no deseada y obteniéndose un vector e características, mismo que represente de una manera eficiente al patrón original del objeto, este proceso se lo realiza con el único objetivo de poder diferenciar un objeto de otro. (Aguilar, E 2018 pp 26)

- Detectores histológicos tipo HAAR cascade

Son detectores que presentan características rectangulares de imagen para el reconocimiento de objetos, se basan en la generación de áreas o parámetros que resultan del análisis y coincidencias encontradas dentro de las imágenes como se muestra en la figura 5-1, los patrones básicos de las características de tipo HAAR, las áreas de color blanco equivalen a las áreas positivas mientras que su opuesto de color negro equivalen a las áreas negativas, en primera instancia los valores de cada pixel dentro de las áreas negras se suman para posteriormente realizar la misma acción con los valores que se encuentran dentro de las áreas blancas, el resultado de la suma de las áreas blancas la diferencia con el de las áreas negras y se analiza el resultado de las subregiones de cada imagen clasificadas y con ello se pueden crear clasificadores más eficientes con capacidad de detectar objetos con variaciones de escala, color o iluminación. (Kannan, R. 2018 pp. 2605)

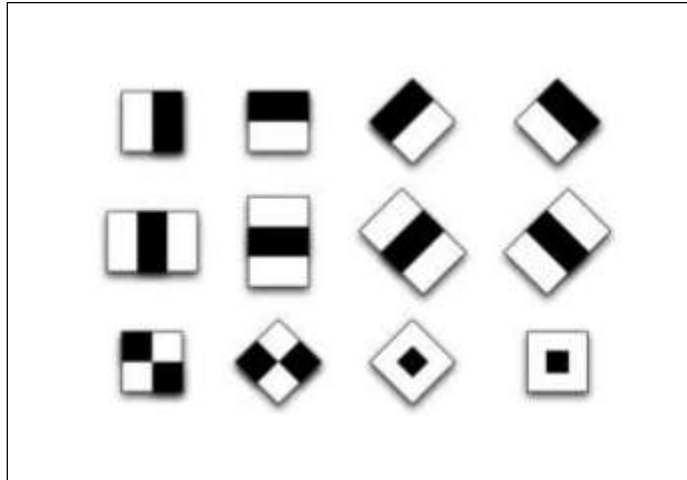


Figura 5-1. Patrones básicos de características HAAR

Fuente: (Kannan, R. 2018 pp. 2605)

- Detectores tipo HOG-SVM

Histogram of oriented Gradients o más conocidos como detectores tipo HOG, las formas de los objetos y las apariencias que presentan dentro de una imagen suelen describirse mediante distribución de las direcciones mediante vectores de cada borde de objeto mismos que están divididos dentro de secciones en la imagen entonces un descriptor está formado por el conjunto de todos ellos y convierte la imagen con sus direcciones y distribución, ya que estos predominan en el contorno de los objetos se pueden identificar así como se observa en la figura 6-1. (Reinius, S. 2013 pp. 16)

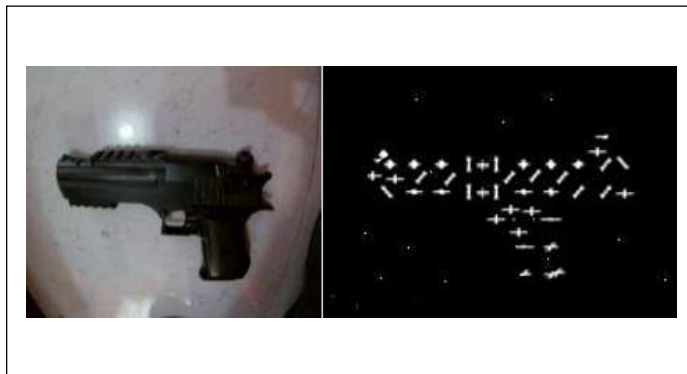


Figura 6-1. Conversión de imagen con detector HOG

Fuente: (Reinius, S. 2013 pp. 16)

- Detectores tipo LBP-Cascade

Local Binary Patterns, realizan una comparación de píxeles dentro de una imagen detectando un umbral de vecindad cada 3x3 píxeles en el cual existe un píxel central que contienen un valor resultante considerado como un número binario como se muestra en la figura 7-1. (Reinius, S. 2013 pp. 20)

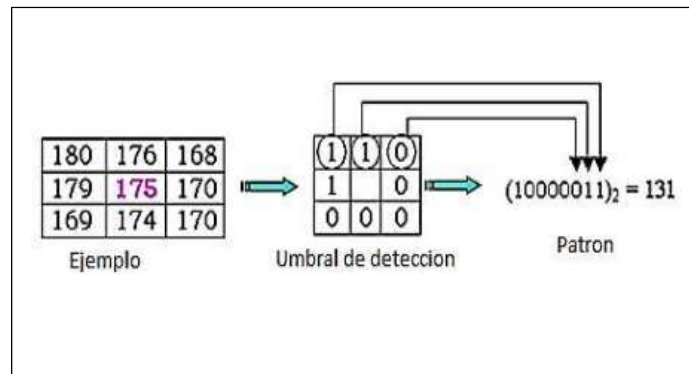


Figura 7-1. Ejemplo del cálculo LBP

Fuente: (Reinius, S. 2013 pp. 20)

1.4.1.9 Reconocimiento

Una vez seleccionados los descriptores en la etapa anterior se procede a la clasificación de la imagen utilizando generalmente algoritmos genéticos, redes neuronales y métodos estadísticos. Así también varias de estas técnicas de procesamiento utilizan los histogramas Orientados o Gradientes (HOG) y el conocido algoritmo de clasificación de Máquinas de Soporte Vectorial (SVM) misma que se basa en la división de la imagen en pequeñas zonas obteniendo así un mejor reconocimiento, además con la implementación de estos dos métodos se ha logrado crear clasificadores en cascada mucho más eficientes como el clasificador HAAR cuyo entrenamiento es muy fácil y sus aplicaciones son muy amplias. (Jiménez, 2005, p.29)

Para evaluar la calidad de la predicción de un modelo entrenado previamente se puede usar la matriz de confusión la cual basa su análisis en la clasificación de los resultados reales o de predicción, donde las filas representan los valores de predicción y las columnas los reales de cada evento ya sean positivos o negativos, en la tabla 5-1 se muestra los elementos de la matriz de confusión. (Jiménez, 2005, p.29)

Tabla 5-1: Matriz de confusión

| MATRIZ DE CONFUSIÓN OBJETO | | Valor de predicción | |
|----------------------------|-----------|---------------------|-----------|
| | | Positivos | Negativos |
| Valor real | Positivos | VP | FN |
| | Negativos | FP | VN |

Fuente: (Torres, 2018)

Donde:

VP: Es la cantidad de positivos correctamente clasificados como positivos por el modelo.

VN: Muestra la cantidad de negativos correctamente clasificados como negativos por el modelo.

FN: Nos muestra la cantidad de positivos clasificados incorrectamente como negativos.

FP: Se considera a la cantidad de negativos clasificados incorrectamente como positivos.

Precisión: Muestra un valor cuando se predicen eventos positivos. (Torres, 2018)

Ecuación 1-1. Fórmula de cálculo de precisión

$$Precisión = \frac{VP}{VP + FP}$$

TVP: Comúnmente conocida como sensibilidad o exhaustividad y muestra la tasa de verdaderos positivos, conlleva el valor cuando el evento es positivo en que porcentaje se ha logrado clasificar.

(Torres, 2018)

Ecuación 2-1. Fórmula de cálculo de TVP

$$TVP = \frac{VP}{Total\ Positivos}$$

1.5 Adquisición de imágenes para el entrenamiento del clasificador

Para el entrenamiento del clasificador se necesita adquirir imágenes positivas y negativas:

- Imágenes positivas

Se refiere a toda imagen que contenga específicamente el objeto que se desea detectar, en la imagen 8-1 se muestra una imagen considerada como positiva.



Figura 8-1. Imagen positiva

Realizado por: OÑATE, Fernando 2020

- Imágenes negativas

Se refiere a toda imagen que no contenga específicamente el objeto que se desea detectar, en la imagen 9-1 se muestra una imagen considerada como negativa.



Figura 9-1. Imagen negativa

Realizado por: OÑATE, Fernando 2020

1.6 Pre procesamiento de imágenes para el entrenamiento del clasificador





Dentro del pre procesamiento de imágenes para el entrenamiento del clasificador se pueden encontrar varias acciones que dependerán de cómo se hayan obtenido las imágenes entre las más usadas tenemos:

- Redimensionamiento de imágenes

Dependiendo del dispositivo usado para obtener las imágenes este guardará cada imagen de un tamaño y formato específico, para optimizar el tiempo de entrenamiento de un clasificador en

cascada se puede redimensionar las imágenes a un tamaño y formato menor dependiendo de las capacidades del ordenador en el cual se vaya a realizar el entrenamiento, en la tabla 6-1 se muestran algunos programas para el redimensionamiento de imágenes en masa. (Kannan, R. 2018 pp. 2605)

Tabla 6-1: Programas para el redimensionamiento de imágenes en masa

| Nombre | Imagen | Descripción | Licencia |
|---------------------|---|--|----------|
| Light Image Resizer |  | Actualmente presenta una versión 5.0.5.1 trabaja sobre plataforma Windows superiores a la versión 7. Es un software de fácil utilización y buen rendimiento, | Gratuito |
| Image Resizer |  | Se dispone de una versión 2.9, trabaja sobre Windows 7 y superiores y es compatible además con otros sistemas operativos. | Gratuito |
| BeFunky |  | El programa trabaja en línea, y da soporte a los principales formatos que se conoce como el JPG. | Gratuito |
| Advanced Renamer |  | Es una herramienta capaz de renombrar múltiples archivos y carpetas a la vez. | Pagado |



Realizado por: ONATE, Fernando 2020

Fuente: <https://es.digitaltrends.com/fotografia/programas-para-editar-fotos-2/>

- Recorte del área de interés

Para poder entrenar el clasificador en cascada se debe delimitar el objeto a detectar de la imagen, para ello existen herramientas pre establecidas, mismas que se muestran en la tabla 7-1. (Kannan, R. 2018 pp. 2605)

Tabla 7-1: Programas para recorte de imágenes

| Nombre | imagen | Descripción | Licencia |
|---------------|---|--|----------|
| Object Marker |  | Se creó para automatizar y simplificar el proceso de exportación de los objetos. | Gratuito |
| Image clipper |  | Se utiliza para delimitar objetos dentro de una imagen. | Gratuito |

Realizado por: ONATE, Fernando 2020

Fuente: <https://sapui5.hana.ondemand.com/#/api/sap.m.ObjectMarker>

- Hardware y software para el entrenamiento

Para el entrenamiento del clasificador en cascada se debe optar por un hardware y software tomando en cuenta que un buen hardware con un buen software disminuirá de manera considerable el tiempo que se demora en entrenar el clasificador. En las tablas 8-1 y 9-1 se muestran distintos (hardware y software) que se pueden utilizar en el entrenamiento del clasificador, tomando en cuenta que existen un sin número de modelos y versiones solo se muestran los más usados.



Tabla 8-1: Hardware para entrenamientos más usados

| Nombre | Imagen | Descripción | Precio (USD) |
|-------------------|---|---|--------------|
| Dell XPS |  | Incorpora un procesador de octava generation Intel Core i7 con una memoria RAM de 8GB a 16GB | 1100 |
| Huawei MateBook |  | Tiene un procesador de octava generation Intel Core i7 con una memoria RAM de 8GB con ampliación hasta 32 GB. | 930 |
| HP Spectre |  | Contiene un procesador Intel Core i7 con memoria RAM de 8GB a 16GB. | 2300 |
| Apple MacBook Pro |  | Su procesador es un Intel Core i7 o con una memoria RAM de 16GB. | 2500 |

Realizado por: ONATE, Fernando 2020

Fuente: <https://www.techradar.com/news/mobile-computing/laptops/best-laptops-1304361>

Tabla 9-1: Software para entrenamientos más usados

| Nombre | Imagen | Descripción | Licencia |
|---------|--|---|----------|
| Windows |  o | Es el software más utilizado en el país y trabaja con una licencia pagada, actualmente se maneja con la versión 10. | Pagada |
| Linux |  | Es un sistema operativo no muy usado en el país pero que presenta varias ventajas además de no necesitar la constante actualización de licencias. | Libre |

Realizado por: ONATE, Fernando 2020

Fuente: <http://xml.cie.unam.mx/xml/Linux/glinux-2.html>

1.7 Ubicación de los nodos de vigilancia

Una buena ubicación de los nodos de vigilancia permitirá que el sistema sea efectivo o no, para ello se considera los puntos más críticos y que demanden mayor vigilancia, en algunos ambientes se tienen varios puntos críticos que demandan mayor vigilancia que otros puntos, en estos casos la eficiencia del sistema dependerá del número de nodos implementados para cubrir dichos puntos. Si el campo de visión no es bueno los nodos no podrán recabar la información de sucesos, para la correcta ubicación de los nodos de debe considerar algunos aspectos como son:

- Determinar los puntos críticos que demanden mayor vigilancia.

Dichos puntos variarán de acuerdo al ambiente por ejemplo en el caso de una oficina su punto crítico será la entrada a la misma además del área que ocupa el oficinista, en el caso de un automóvil los puntos críticos serán el área que ocupa el conductor y el área que ocupan los pasajeros, en el caso de un parque público sus puntos críticos serán los lugares con menor visión por parte de los vigilantes, etc. (GrupoPrevent, 2014)

- Evitar que los nodos estén al alcance de los usuarios

Es muy relevante en los casos en que los nodos de vigilancia se instalan en ambientes en donde pueden ser víctimas de vandalismo o sabotaje, los nodos deben ser ubicados en lugares altos y en otros casos en lugares poco accesibles para que no sean víctimas de vandalismo. (GrupoPrevent, 2014)

- No ubicar los nodos de vigilancia a contraluz

A pesar que en la actualidad muchas de las cámaras que utilizan los nodos de vigilancia cuentan con *autoris* que adapta la visión a la cantidad de luz al igual que el ojo humano, durante un instante breve puede perder la imagen lo que conlleva a obtener una debilidad del sistema, para ello se recomienda colocar un mayor número de nodos frente con frente para evitar este efecto. (GrupoPrevent, 2014)

- Enfrentar los nodos de vigilancia

De esta manera si uno de los nodos de vigilancia sufre algún desperfecto o es víctima de algún acto vandálico la otra cámara podrá capturar el hecho e identificar al responsable. (GrupoPrevent, 2014)

- Aprovechar el poder disuasorio de los nodos de vigilancia




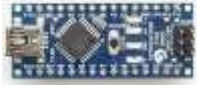
El principal objetivo de los nodos de vigilancia es dar con el responsable de un delito o actos vandálicos, por lo tanto, mientras más visibles estén tendrán un mayor efecto disuasorio en las personas. (GrupoPrevent, 2014)

1.8 Tarjetas embebidas

Para poder considerarse como una tarjeta embebida todos sus componentes deben estar unidos en una sola placa o PCB (Printed Circuit Board), entre las más conocidas encontramos las FPGA mismas que contaban con un procesador central (CPU) además de conexiones internas mediante buses de datos, memoria interna y diversos canales de entrada y salida uno de sus modelos más comerciable son las tarjetas Arduino. Los programas de sistemas embebidos se enfrentan normalmente a tareas de procesamiento en tiempo real. Existen también plataformas desarrolladas por distintos fabricantes que proporcionan herramientas para el desarrollo y diseño de aplicaciones y prototipos con sistemas embebidos desde ambientes gráficos. Varias empresas desarrolladoras de hardware y software como INTEL, ASUS, Nvidia, Raspberry Pi Foundation entre otras han marcado nuevos desafíos para la producción e implementación de sistemas embebidos cada vez más pequeñas y de menor costo como lo es Raspberry Pi que han llegado a constituir estructuras no modulares como las computadoras tradicionales, ya que trabajan sobre software libre su aplicación conlleva a varios ámbitos investigativos y comerciales aunque conllevan algunas desventajas como la difícil actualización y reemplazo de sus componentes. (Maldonado, L. 2018 pp. 16)

Existen varios tipos de Arduinos que poseen distintas características, los más importantes están enunciados a continuación en la tabla 10-1:

Tabla 10-1: Características de los Arduinos




| Especificaciones | Arduino UNO | Arduino MEGA | Arduino Leonardo | Arduino NANO |
|----------------------|---|---|--|---|
| Imagen |  |  |  |  |
| Micro controlador | ATmega 328 | ATmega 2560 | ATmega32u4 | ATmega 328 |
| Voltaje operativo | 5V | 5V | 5V | 5V |
| Tensión de entrada | 7-12V | 7-12V | 7-12V | 7-12V |
| Voltaje de entrada | 6-20V | 6-20V | 6-20V | 6-20V |
| Pines digitales | 14 | 54 | 20 | 22 |
| Pines análogos | 6 | 16 | 7 | 8 |
| Corriente DC por pin | 40mA | 40mA | 40 Ma | 40 mA |
| Corriente DC | 50mA | 50mA | 50Ma | 19mA |
| Capacidad de memoria | 32 KB | 256 KB | 32 KB | 32 KB |
| SRAM | 2 KB | 8KB | 2.5 KB | 2 KB |
| EEPROM | 1 KB | 4KB | 1 KB | 1 KB |
| Clock Speed | 16MHz | 16MHz | 16MHz | 16MHz |

Realizado por: Oñate, F. 2020

Fuente: <https://www.electrontools.com/Home/WP/2018/06/19/arduino-mega-2560-caracteristicas/>

Las características principales de los modelos de Raspberry Pi se muestran en la tabla 11-1:

Tabla 11-1: Características de los modelos de Raspberry Pi

| MODELOS RASPBERRY PI | | | |
|----------------------|---|--|---|
| Características | Modelo A+ | Modelo B | RPI Modelo B+ |
| Imagen |  |  |  |
| SoC | Broadcom BCM2835 | Broadcom BCM2835 | Broadcom BCM2836 |
| CPU | ARM11 ARMv6 700MHz | ARM11 ARMv6 700MHz | Cortex-A7 4 núcleos 900MHz |
| GPU | Broadcom VideoCore IV 250MHz OpenGL ES 2.0 | | |
| RAM | 256 MB | 512 MB | 1 GB |
| USB | 1 | 4 | 4 |
| Salidas de video | HDMI 1.4 @ 1920X1200 px | | |
| Almacenamiento | Micro SD | | |
| Ethernet | No | Si 10/100 Mbps | Si 10/100 Mbps |
| Tamaño | 85,60 x 56,5 mm | 85,60 x 56,5 mm | 85,60 x 56,5 mm |
| Peso | 23g | 45g | 45g |

Realizado por: Oñate, F. 2020

Fuente: <http://bibing.us.es/proyectos/abreproy/90330/fichero/Memoria+TFG.pdf>

1.9 Acceso a los servicios SMS

Los SMS (Short Message Service) es un servicio inalámbrico que consiste en el envío y recepción de mensajes de texto desde y hacia teléfonos móviles, fijos y a otros sistemas externos, donde el mensaje puede contener números, palabras o una combinación de caracteres alfanuméricas. Se los llama mensajes cortos ya que poseen entre 100 y 250 caracteres de largo. Mediante un módem GSM podemos conectar cualquier sistema digital a la red GSM (Global System for Mobile) no solo para enviar mensajes SMS si no para transmitir datos. Para acceder a estos servicios se requiera una conexión en serie como se aprecia en la figura 10-1 para así poder utilizar el servicio. (Díez. S.2016. pp 30)

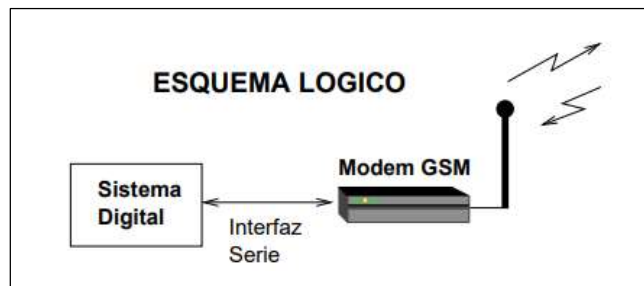





Figura 10-1: Acceso a los servicios SMS

Fuente: <http://www.learobotics.com/personal/juan/doctorado/sms/sms.pdf>

En la tabla 12-1 se muestra una comparación de características de módulos GSM existentes en el mercado.

Tabla 12-1: Comparación de módulos GSM

| | Sim 900 Shield | Sim 808 Shield | Sim 800L |
|--------------------|---|--|---|
| Imagen |  |  |  |
| Costo | \$ 45 | \$60 | \$25 |
| Chip sim | 900 | 808 | 800 |
| Banda de operación | 850 a 1900 MHz | 850 a 1900 MHz | 850 a 1900 MHz |
| Dimensiones | 8.5x5.7x2 cm | 6.5x5.4x1.3cm | 1.5 cm x 1.8 cm x 0.25 cm |
| Voltaje de trabajo | 5v | 5v | 5v |
| Comunicación | Serial | Serial | Serial |
| Red | GSM/GPRS | GSM/GPRS | GSM/GPRS |




Realizado por: ONATE, Fernando 2020

Fuente: <http://electrotekmega.com/producto/modulo-gprs-gsm-sim800/>

1.10 Acceso a los servicios de una base datos

Se entiende como una base de datos a un conjunto de datos estructurado y almacenado de forma sistemática con el único objetivo de facilitar su posterior acceso, por lo tanto, puede estar constituida por cualquier tipo de datos. Una base de datos constituye una mayor disponibilidad, independencia, seguridad, eficiencia y mayor reutilización de datos, en la actualidad se puede contar con bases de datos pre definidas como Gmail, Yahoo!, la nube, entre otras y que son de acceso libre para todo tipo de usuarios, de esta manera facilitan su acceso para fines investigativos y comerciales, en la tabla 13-1 se puede observar una comparación de las características de algunas bases de datos más usadas en la actualidad. (Oyala.V. 2011)

Tabla 13-1: Comparación de bases de datos

| | Gmail | Nube | Base Física |
|---------|--|---|--|
| Imagen |  |  |  |
| Precio | Sin costo | Depende del tamaño requerido | Depende del tamaño requerido |
| Espacio | 100 GB | Depende del usuario | Depende usuario |
| Acceso | Libre | Libre | Libre |

Realizado por: ONATE, Fernando 2020

Fuente: https://volaya.github.io/libro-sig/chapters/Cliente_servidor.html

CAPÍTULO II

2 PROPUESTA Y DISEÑO DEL PROTOTIPO

2.1 Recopilación de imágenes para el entrenamiento

Se trata de adquirir imágenes negativas y positivas que serán usadas para el entrenamiento del clasificador, entendiéndose como negativas a imágenes en las que no esté contenido el objeto a detectar y como positivas a imágenes que contienen específicamente el objeto el cual se desea detectar.

Las imágenes positivas se obtuvieron mediante la captura desde un teléfono inteligente HUAWEI P30 PRO, se seleccionó este medio ya que posee una gran definición para la captura de imágenes lo cual es ideal para obtener imágenes claras para el entrenamiento. Las imágenes se pueden apreciar en la figura 1-2. Se capturaron alrededor de 4500 imágenes positivas y 8000 imágenes negativas que serán usadas para el entrenamiento del clasificador. Dentro de las imágenes negativas se utilizó una cantidad de 2000 imágenes descargadas de internet debido a que las imágenes negativas pueden ser de cualquier origen o tipo.

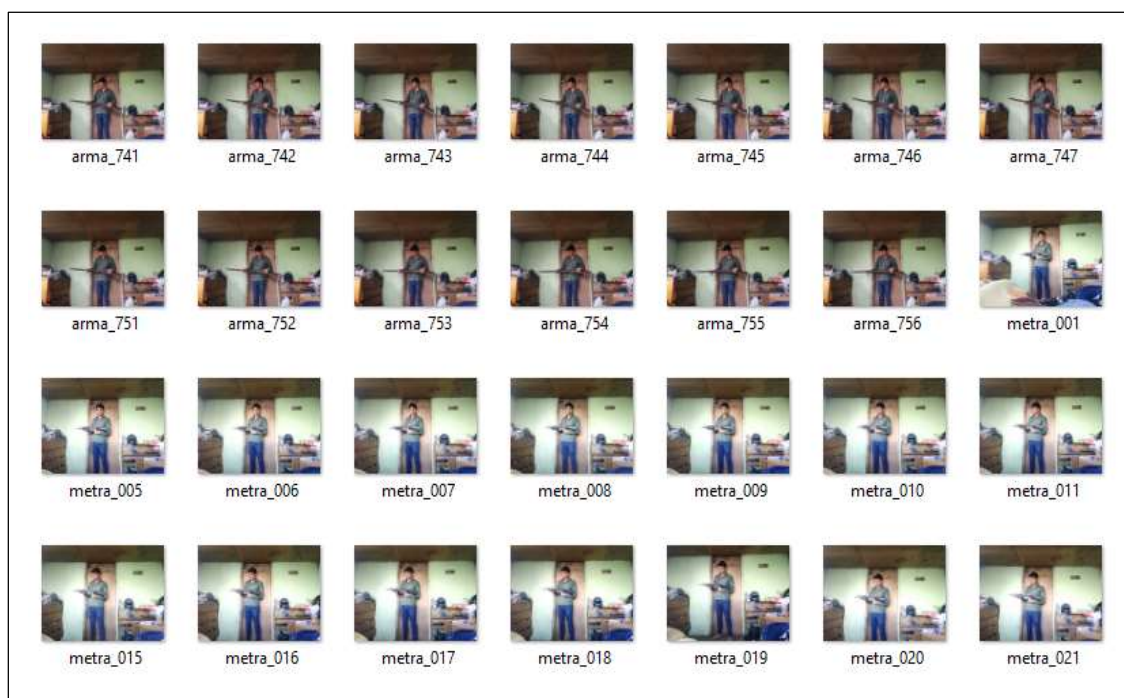


Figura 1-2. Recopilación de imágenes positivas para el entrenamiento

Realizado por: OÑATE, Fernando 2020

Una parte de las imágenes negativas fueron descargadas de diferentes páginas web complementando así a las imágenes capturadas por el teléfono inteligente, las imágenes deben cumplir con la condición de no contener objetos considerados como positivos, las imágenes se proceden a almacenar como se observa en la figura 2-2, cabe recalcar que es de suma importancia utilizar imágenes reales que contengan variación entre sí de entornos, iluminación, fondos, color etc. De estas variaciones dependerá la adaptación del sistema a implementar ya que posteriormente los datos de dichas imágenes serán utilizados por el clasificador y dependerá directamente de ello su eficiencia.

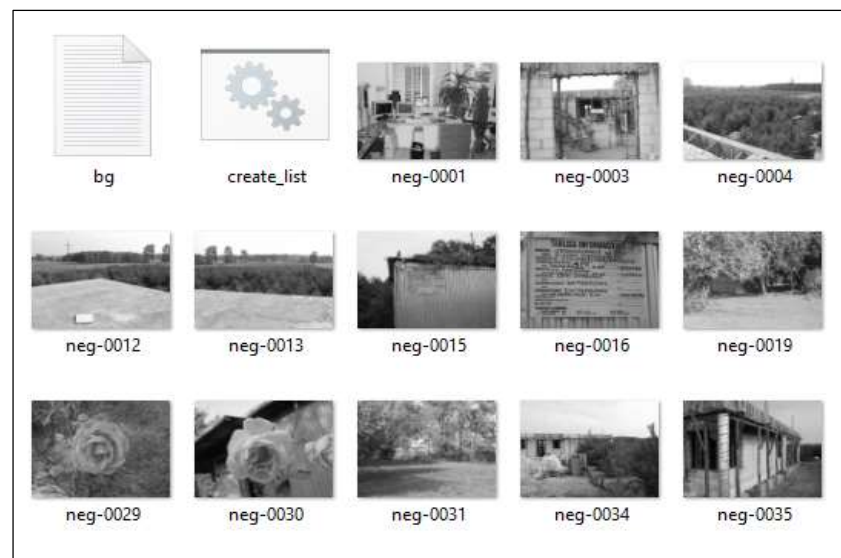


Figura 2-2. Recopilación de imágenes negativas para el entrenamiento

Realizado por: OÑATE, Fernando 2020

2.2 Pre procesamiento de las imágenes

Luego de la recopilación de las imágenes se debe realizar un primer pre procesamiento de las mismas para lo cual se utilizó el ordenador HP Spectre x360, inicialmente se redujo el tamaño de las imágenes a 640 x 480 píxeles y dándoles un formato .JPG, en el caso de que las imágenes recopiladas presentan un tamaño mayor el entrenamiento del clasificador demandara una cantidad elevada de recursos informáticos y de tiempo para el entrenamiento del clasificador.

Para que el proceso de entrenamiento sea más fácil de procesar por el ordenador se procede a modificar previamente ciertas características de las imágenes como son: el nombre siempre buscando que los mismos no contengan caracteres especiales ni espacios en blanco además de estar respectivamente enumeradas y realizar la conversión del formato .JPG a .BMP con

dimensiones de 640 x 480 con ayuda de un programa de nombre *Ligth Image Resizer*, lo que agiliza la velocidad de extracción de información de las imágenes analizadas. Este paso es muy importante para que en el transcurso de localización del objeto dentro de la imagen no se generen errores en la lectura de datos.

2.3 Marcar la posición del objeto dentro de la imagen positiva

Para la delimitación del objeto dentro de la imagen se utilizó el programa *object_marker*, los datos que se obtienen de la delimitación corresponden a los píxeles de la imagen, el desarrollador del programa recomienda que el marcado se realice de izquierda hacia la derecha evitando así errores en los datos del objeto, por lo tanto, tomando el ejemplo de la imagen del arma se puede obtener la información del marcado del objeto mismo que muestra que el objeto a analizar se encuentra ubicado en la posición 354, 217 píxeles en el plano (x, y) además muestra el tamaño de 186x110 píxeles de largo y alto en el plano (w, h), hay que tener muy en cuenta que los objetos a ser analizados no presentan la misma forma, por lo tanto, se debe realizar un entrenamiento por separado. Esto influirá directamente en la eficiencia del entrenamiento si bien para la pistola podemos hacer una relación de tamaño del objeto de uno de alto por dos de largo (1,2) y para el cuchillo podemos darle una relación de uno de alto por tres de ancho (1,3) y así para las demás armas, además se utilizó parámetros establecidos ya que el sistema tiene un entrenamiento por defecto de 24x24 píxeles mismo que es utilizado en la mayoría de entrenamientos de este tipo como se puede observar en la figura 3-2.



Figura 3-2. Localización del objeto en la imagen positiva

Realizado por: OÑATE, Fernando 2020

2.4 Creación del vector de entrenamiento

La creación del vector de entrenamiento que contiene las características de las imágenes como son el nombre, tamaño y posición; es muy importante, ya que contienen el número de muestras que se van a crear para el entrenamiento dicho número sigue una fórmula aproximada la cual consta de parámetros como el número de imágenes positivas, negativas, número de etapas de cascada y valores de aproximación deseados como los falsos positivos y niveles de error. Para la creación del vector se deben tomar en cuentas varios parámetros que requiere el aplicativo `createsamples` como se muestra a continuación en la tabla 1-2.

Tabla 1-2: Parámetros formar el vector de entrenamiento

| | |
|------|---|
| info | Ruta para el archivo de información positiva |
| vec | Ruta del vector de salida |
| num | Número de archivos positivos para ser empaquetados en el vector |
| w | Anchura del objeto |
| h | Altura del objeto |

Realizado por: ONATE, Fernando 2020

Fuente: https://www.cs.auckland.ac.nz/~m.rezaei/Tutorials/Creating_a_Cascade_of_Haar-Like_Classifiers_Step_by_Step.pdf

En muchos de los casos, la creación del vector presenta problema. Una de las maneras de verificar su correcto funcionamiento es comprobar que el número de imágenes positivas estén empaquetadas dentro del vector, para realizar una estimación de los vectores se puede realizar una aproximación mediante una fórmula misma que contiene los valores que debemos utilizar para un correcto funcionamiento, la formula fue propuesta por Maria Dimashova en OpenCV.org. (Dimashova, 2012)

Ecuación 1-2. Fórmula de cálculo de parámetros de entrenamiento

$$vecfile \text{ has to contain } \geq (num.P + (num.Stages - 1) * (1 - minHitRate) * num.P)S$$

Donde S viene a ser un parámetro de recuento de las imágenes que se utilizan de fondo ya que al modificar valores de entrenamiento dicho valor genera complicaciones de ahí la importancia de utilizar la fórmula 1-2 para obtener valores reales para el entrenamiento. Si bien la fórmula no se la considera como guía para crear los parámetros de un clasificador, esta se basa principalmente en las experiencias de investigadores realizadas con anterioridad, pero es de mucha ayuda para la

realización de clasificadores en cascada por lo cual fue tomada como referencia para la realización del clasificador.

2.5 Obtención del clasificador Haar

Una vez obtenido el vector de entrenamiento se procede a obtener el clasificador para ello se utilizó la aplicación haartraining el cual genera un único archivo en cascada en base a los parámetros solicitados como se muestra en la tabla 2-2.

Tabla 2-2: Parámetros para formar el clasificador

| | |
|---------|---|
| data | Indica la ruta de almacenamiento para los clasificadores en cascada |
| vec | Indica la ruta que indica la localización del archivo vector |
| bg | Indica la ruta del archivo de fondo |
| npos | Indica el número de muestras positivas \leq no. Archivos positivos bmp |
| nneg | Indica el número de muestras negativas (patches) \geq npos |
| nstages | Indica la número de etapas previstas para la obtención |
| Mem | Indica la cantidad de memoria asignada en MB |
| mode | Parámetro no asignado |
| W, h | Insertar las medidas de las muestras |
| Nonsym | Use esto si el objeto no es simétrico horizontalmente |

Realizado por: ONATE, Fernando 2020

Fuente: https://www.cs.auckland.ac.nz/~m.rezaei/Tutorials/Creating_a_Cascade_of_Haar-Like_Classifiers_Step_by_Step.pdf

Los valores que no son asignados por el usuario automáticamente se colocan por defecto, el rendimiento del clasificador se verá afectado directamente por características de las muestras como son la iluminación los tamaños del objeto y la calidad de imagen además de ello la configuración ideal influirá en la calidad del mismo como se muestra en la figura 4-2 y sus parámetros de detallan en la tabla 3-2.

```

Parent node: 0
*** 1 cluster ***
POS: 200 200 1.000000
NEG: 200 0.243605
BACKGROUND PROCESSING TIME: 0.01
Precalculation time: 8.09
+-----+
| N |%SMP|F| ST.THR | HR | FA | EXP. ERR|
+-----+
| 1|100%|-|-0.915344| 1.000000| 1.000000| 0.067500|
+-----+
| 2|100%|+|-1.761648| 1.000000| 1.000000| 0.050000|
+-----+
| 3|100%|-|-1.040223| 1.000000| 0.325000| 0.027500|
+-----+
Stage training time: 4.79
Number of used features: 3

Parent node: 0
Chosen number of splits: 0

Total number of splits: 0

Tree Classifier
Stage
+-----+
| 0| 1|
+-----+

```

Figura 4-2. Obtención del clasificador

Realizado por: OÑATE, Fernando 2020

Tabla 3-2: Parámetros que se obtienen en el entrenamiento

| | |
|--------------|--|
| Parent node: | Define la etapa actual del proceso de obtención |
| N: | Muestra el número de funciones utilizadas en esta etapa |
| %SMP: | Muestra el porcentaje de muestras (Porcentaje de muestras usadas por este parámetro) |
| F: | Indica “+” volteado positivo (cuando se aplica la simetría) y “-“ si no se ha volteado |
| ST.THR: | Muestra el umbral de etapa |
| HR: | Muestra la tasa de aciertos basado en el umbral de etapa |
| FA: | Indica la falsa alarma basada en el umbral de etapa |
| EXP. ERR: | Muestra el error exponencial de Exponencial de clasificador fuerte |

Realizado por: OÑATE, Fernando 2020

Fuente: https://www.cs.auckland.ac.nz/~m.rezaei/Tutorials/Creating_a_Cascade_of_Haar-Like_Classifiers_Step_by_Step.pdf

En la figura 5-2 se puede observar que las características usadas han incrementado en el encabezado nodos, en general la detección falsa ha disminuido y el tiempo computacional de obtención ha incrementado, de esta manera se realizara hasta llegar a la última muestra, el proceso va avanzando y se encuentra en la muestra 10.

```

Parent node: 9
*** 1 cluster ***
POS: 200 200 1.000000
NEG: 200 0.000574246
BACKGROUND PROCESSING TIME: 2.60
Precalculation time: 7.99
+-----+-----+-----+-----+-----+-----+
| N | %SMP | P | ST.THR | HR | FA | EXP. ERR |
+-----+-----+-----+-----+-----+-----+
| 1 | 100% | - | -0.554502 | 1.000000 | 1.000000 | 0.207500 |
+-----+-----+-----+-----+-----+-----+
| 2 | 100% | + | -0.883580 | 1.000000 | 1.000000 | 0.180000 |
+-----+-----+-----+-----+-----+-----+
| 3 | 100% | - | -1.647806 | 1.000000 | 1.000000 | 0.122500 |
+-----+-----+-----+-----+-----+-----+
| 4 | 83% | + | -1.357607 | 1.000000 | 0.785000 | 0.095000 |
+-----+-----+-----+-----+-----+-----+
| 5 | 91% | - | -1.956339 | 1.000000 | 0.810000 | 0.100000 |
+-----+-----+-----+-----+-----+-----+
| 6 | 76% | + | -1.634170 | 1.000000 | 0.630000 | 0.055000 |
+-----+-----+-----+-----+-----+-----+
| 7 | 73% | - | -1.235653 | 1.000000 | 0.435000 | 0.057500 |
+-----+-----+-----+-----+-----+-----+
Stage training time: 10.40
Number of used features: 7
Parent node: 9
Chosen number of splits: 0
Total number of splits: 0
Tree Classifier
Stage
+-----+-----+-----+-----+-----+-----+
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
+-----+-----+-----+-----+-----+-----+

```

Figura 5-2. Obtención del clasificador

Realizado por: OÑATE, Fernando 2020

2.6 Creación del archivo XML

Luego de que ha finalizado el proceso previo se busca los archivos pistola.xml, cuchillo.xml, rifle.xml, metra.xml los cuales nos servirán en la programación del sistema, como se muestra en la figura 6-2.

| | | | |
|---------------------|--------------|------------------------|----------|
| cascares | 30/09/201... | Carpeta de archivos | |
| negative | 30/09/201... | Carpeta de archivos | |
| positive | 10/09/201... | Carpeta de archivos | |
| vector | 30/09/201... | Carpeta de archivos | |
| 01 samples_creation | 30/09/201... | Archivo por lotes ... | 1 KB |
| 02 haarTraining | 30/09/201... | Archivo por lotes ... | 1 KB |
| 03 convert | 30/09/201... | Archivo por lotes ... | 1 KB |
| createsamples | 26/07/200... | Aplicación | 72 KB |
| cuchillo | 28/06/201... | Documento XML | 118 KB |
| cv097.dll | 26/07/200... | Extensión de la apl... | 785 KB |
| cxcore097.dll | 26/07/200... | Extensión de la apl... | 1.017 KB |
| haarconv | 24/11/200... | Aplicación | 60 KB |
| haartraining | 02/12/200... | Aplicación | 88 KB |
| highgui097.dll | 26/07/200... | Extensión de la apl... | 393 KB |
| libguide40.dll | 06/11/200... | Extensión de la apl... | 184 KB |
| pistola | 28/06/201... | Documento XML | 421 KB |

Figura 6-2. Clasificador XML final generado

Realizado por: OÑATE, Fernando 2020

2.7 Algoritmo del detector

En la figura 7-2 se muestra la estructura del algoritmo detector por medio de un diagrama de flujo.

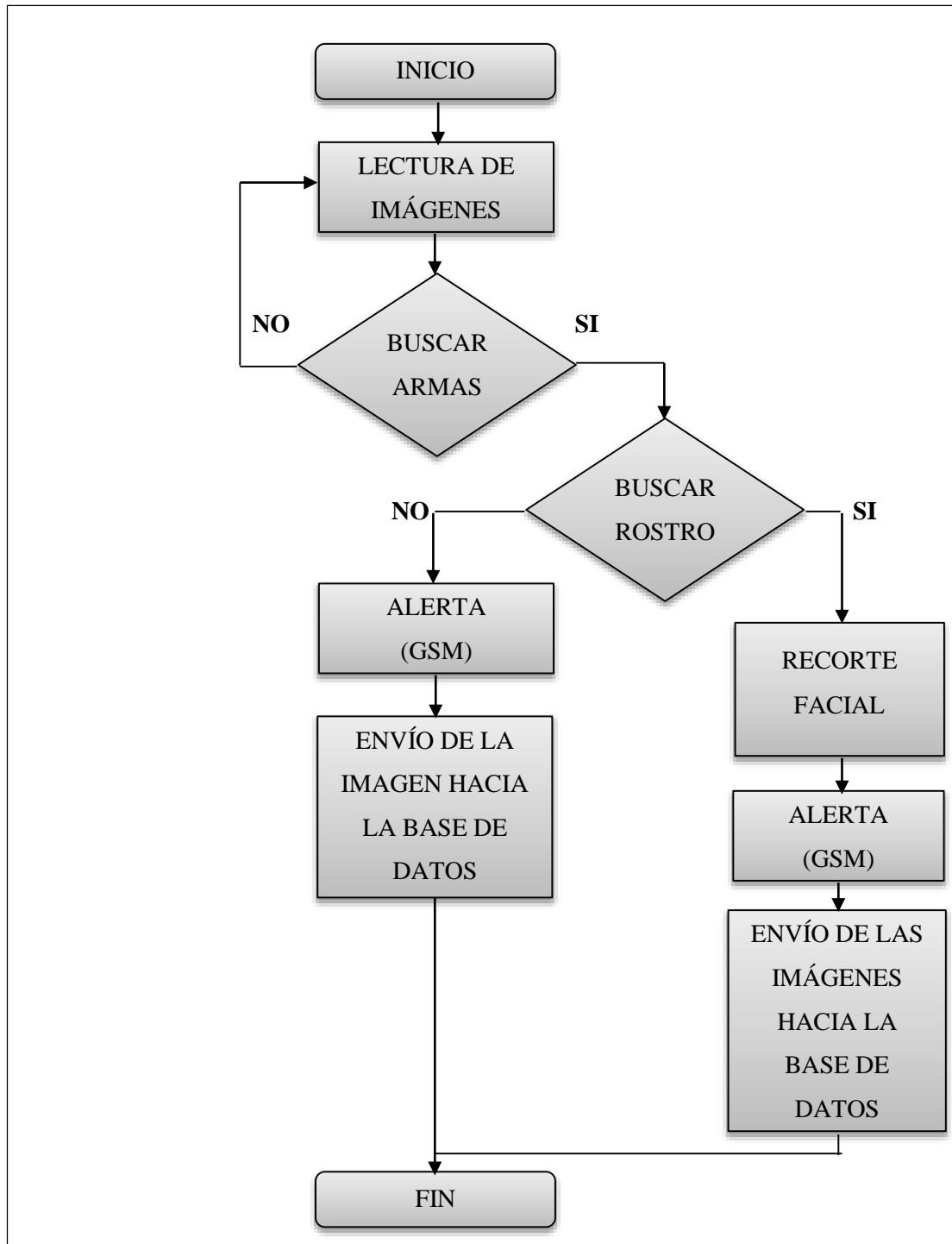


Figura 7-2. Diagrama de flujo del algoritmo detector

Realizado por: OÑATE, Fernando 2020

- Inicio

Para que el detector cumpla con su función principalmente se necesita inicializar librerías de Python que permitirán la correcta ejecución de instrucciones necesarias las cuales se encuentran en el algoritmo detector, mediante las siguientes instrucciones que se observan en la tabla 4-2.

Tabla 4-2: Librerías usadas en el algoritmo detector

| Librería | Descripción |
|---|--|
| <code>from picamera import PiCamera</code> | Sirve para tener conexión entre el puerto de cable Ribbon y el módulo de cámara Raspberry PI |
| <code>import time</code> | Sirve generar lapsos de tiempo o esperas |
| <code>import smtplib</code> | Permiten la conexión con Gmail |
| <code>from email.mime.image import MIMEImage</code> | Permite adjuntar imágenes en Gmail |
| <code>from email.mime.multipart import MIMEMultipart</code> | Permite enviar el correo electrónico de Gmail |
| <code>import cv2</code> | Permite utilizar funciones de OpenCV |
| <code>import numpy as np</code> | Permite trabajar rápidamente con matrices |
| <code>import RPi.GPIO as GPIO</code> | Permite tener conexión entre los puertos GPIO de Raspberry |
| <code>import serial</code> | Permite tener conexión con los puertos seriales de Raspberry |

Realizado por: OÑATE, Fernando 2020

- Lectura de imágenes

Este proceso se lo realiza dentro de un bucle while, lo que conlleva a que el código se esté ejecutando constantemente, a partir de ello se leen las imágenes por segundo que ingresan desde el dispositivo de captura de imágenes, para que el proceso sea más rápido y siempre pensando en un mayor rendimiento del mismo se ha transformado las imágenes adquiridas a escala de grises o RGB.

- Buscar armas

En este proceso el algoritmo compara las características de las imágenes de entrada con los datos de los clasificadores, en caso de detectar un arma se procede a dibujar un rectángulo para su

identificación, luego se busca rostros en la imagen que se detectó el objeto en estudio y si la detección facial es positiva se procede a realizar el recorte y redimensionamiento del rostro detectado para su correcta visualización, si la detección del rostro es negativa solo se enviará la imagen donde se muestra el objeto detectado. Hay que tomar en cuenta que al momento de llamar al clasificador se deben colocar parámetros del mismo como son: la escala y los vecinos, estos parámetros varían de acuerdo al ambiente, calidad de imagen de entrada y agudeza de clasificación.

- Alerta GSM

Una vez que se ha realizado la detección del objeto en análisis en cada caso se envía un mensaje de texto mismo que alertará al personal encargado que se ha detectado un arma, al ser los nodos independientes se coloca en el mensaje la dirección en donde ha ocurrido el evento.

- Envío de la imagen a la base de datos

Con fines de corroborar la información enviada por la alerta, se procede a enviar la imagen detectada hacia la base de datos de Gmail, el algoritmo es capaz de detectar rostros y en caso de que ello suceda se hará un recorte facial y lo enviará junto a la imagen original de detección, con esta información el usuario puede comprobar que se ha detectado un arma y obtener detalles veraces del portador, así proceder a utilizar esta información como mejor le convenga.

- Fin

El programa se ejecuta constantemente, se lo puede dar fin pulsando la tecla escape.

2.8 Implementación del sistema de detección

La implementación del sistema se ha realizado sobre el lenguaje de Python en su versión 3.7.3, al ser un software libre nos brinda múltiples beneficios entre uno de ellos la utilización de sus librerías más importantes, en este caso como librería principal OpenCV en su versión 4.1.0, como se mencionó anteriormente el sistema será implementado sobre una tarjeta embebida en este proyecto se utilizará la Raspberry Pi 3 Modelo B+ en su última versión 2018, se escogió esta tarjeta debido a su capacidad de procesamiento su bajo costo y por su fácil conexión a diversos periféricos de entrada y salida, una vez instaladas todas las librerías en la tarjeta embebida podemos editar el código y realizar las pruebas correspondientes, además de realizar modificaciones del programa con valores propios del entorno ya que con variaciones de

iluminación y características de la cámara a utilizar las características cambian, como se puede observar en la figura 8-2.

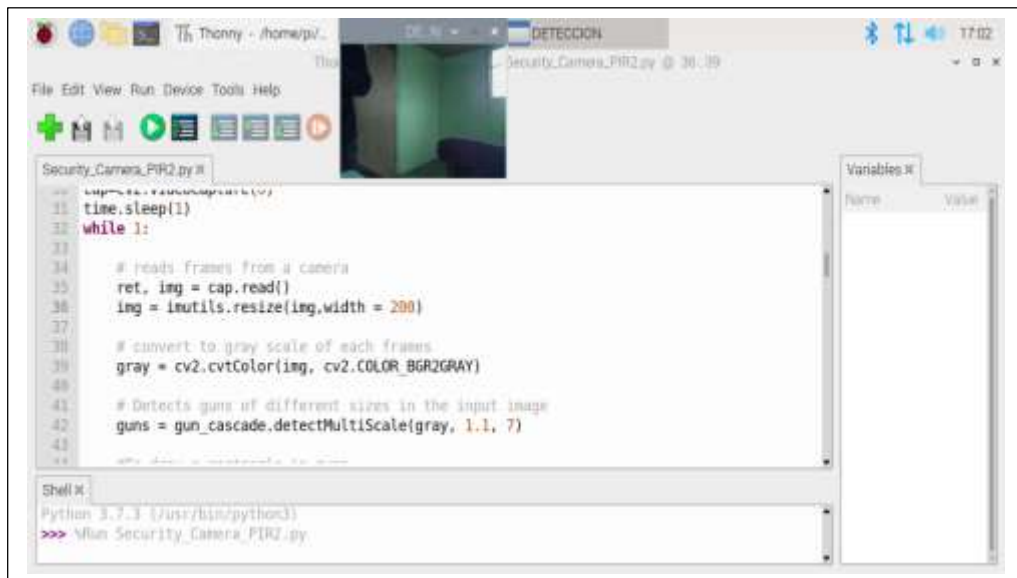


Figura 8-2. Implementación del sistema en Raspberry PI 3 B+

Realizado por: OÑATE, Fernando 2020

Para el envío del mensaje de texto se ha seleccionado el módulo GSM 800L, mismo que se encuentra fácilmente en el mercado y es de bajo costo, utiliza la comunicación serial para intercambiar información con la Raspberry y poder enviar el SMS, en la figura 9-2 se puede observar el esquema de conexión del módulo GSM con la Raspberry.

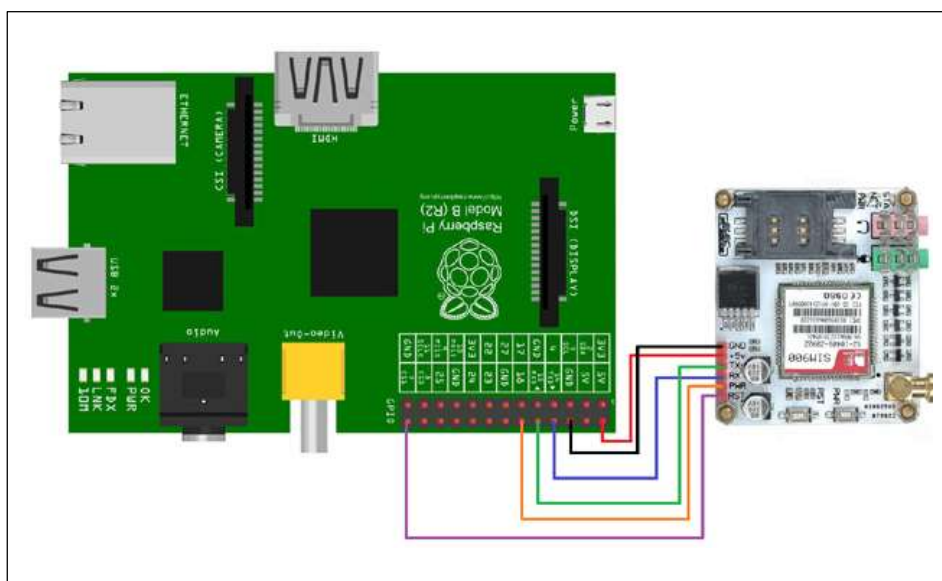


Figura 9-2. Conexión del módulo GSM con la Raspberry PI

Realizado por: OÑATE, Fernando 2020

Una vez realizada la detección y enviada la alerta como mensaje de texto, se procede a guardar la imagen en una base de datos. Se ha seleccionado la base de datos de Gmail, que cuenta con un volumen alto de almacenamiento y es de acceso gratuito. Además de ser de fácil acceso y estar presente en la mayoría de terminales informáticos a nivel mundial, para acceder a sus servicios se procedió a crear una cuenta de Gmail, la cual sirve para que el encargado del sistema pueda utilizar las imágenes, en la figura 10-2 se puede observar la 000cuenta de Gmail creada para el desarrollo de esta investigación.

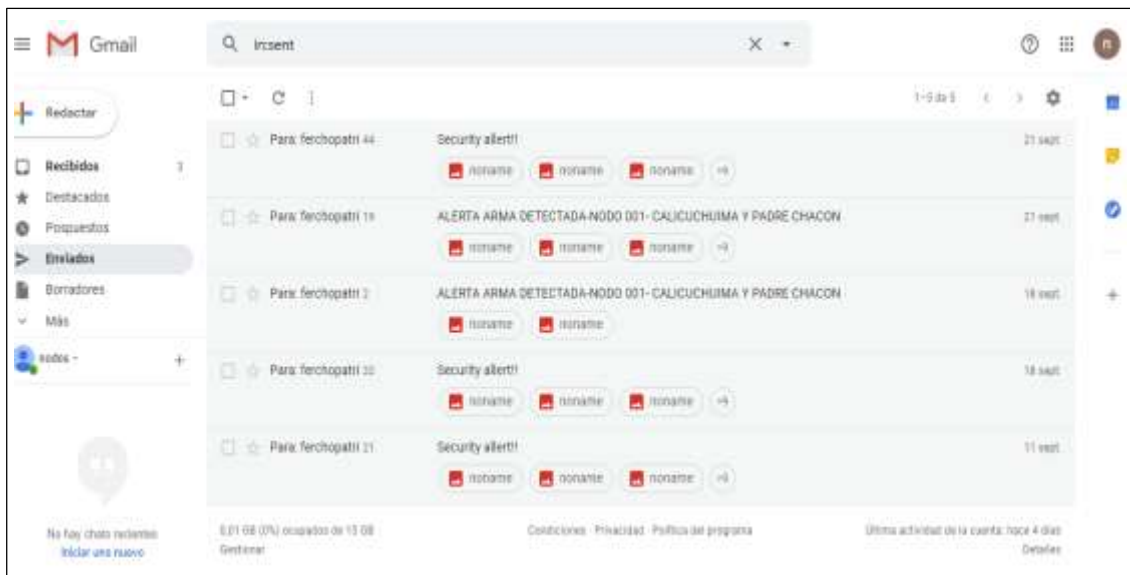


Figura 10-2. Base de datos creada para la recopilación de imágenes detectadas

Realizado por: OÑATE, Fernando 2020

Ubicación de los nodos

- Ubicación de los nodos en una oficina

Se realizó la ubicación a una altura de 2.30 metros con respecto al piso, el nodo uno con un ángulo 30 grados y el nodo dos con un ángulo de 38 grados de inclinación de la cámara hacia abajo, ubicación que permitió un correcto ángulo de visión de cada una de las cámaras en la entrada de la oficina, como se muestra en la figura 11-3.

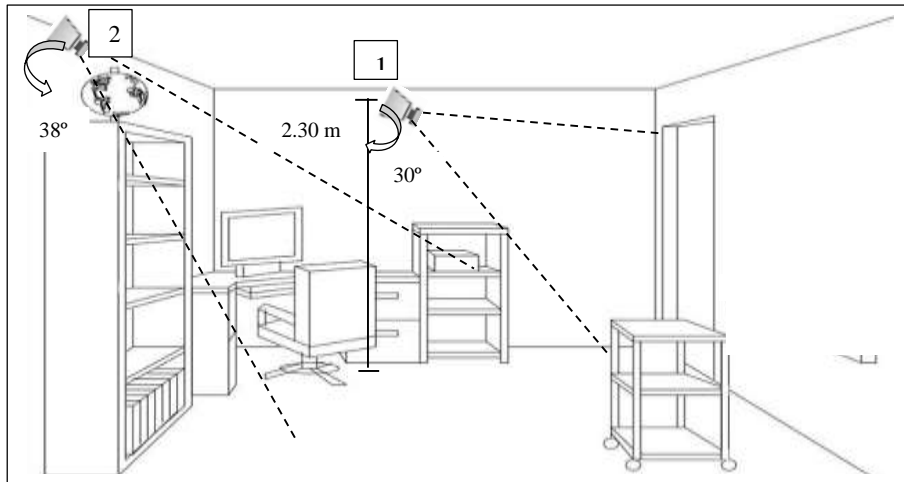


Figura 11-2. Ubicación de los nodos en la oficina

Realizado por: OÑATE, Fernando 2020

Fuente: <https://www.idibujosparacolorear.com/dibujos-de-oficinas-para-colorear>

- Ubicación del nodo en un automóvil

Se realizó la ubicación del nodo a una altura de 0.90 metros de altura con respecto al piso del auto, el nodo se encuentra al costado derecho del automóvil a lado del asiento del copiloto con un ángulo de inclinación de la cámara de 36 grados hacia la izquierda, ubicación que permitió un correcto ángulo de visión de la cámara en el automóvil, como se muestra en la figura 12-3.

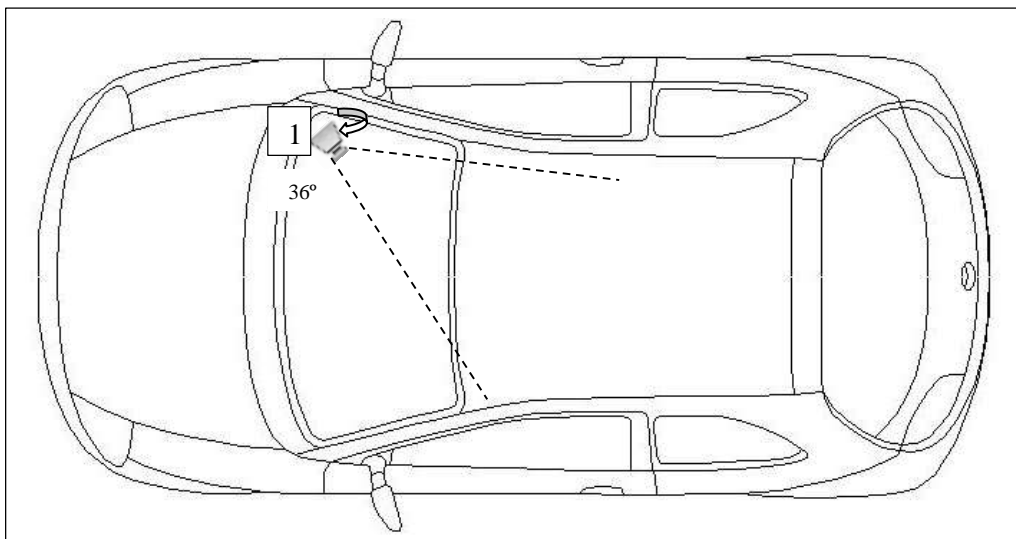


Figura 12-2. Ubicación de los nodos en el automóvil

Realizado por: OÑATE, Fernando 2020

Fuente: <https://www.bloquesautocad.com/coche-en-planta-ford-ka/>

- Ubicación del nodo en un parque público

Se realizó la ubicación de los nodos a una altura de 2.50 metros de altura con respecto al piso del auto, con un ángulo de inclinación de la cámara de 37 grados hacia abajo, ubicación que permitió un correcto ángulo de visión de la cámara en un parque público, como se muestra en la figura 13-3.

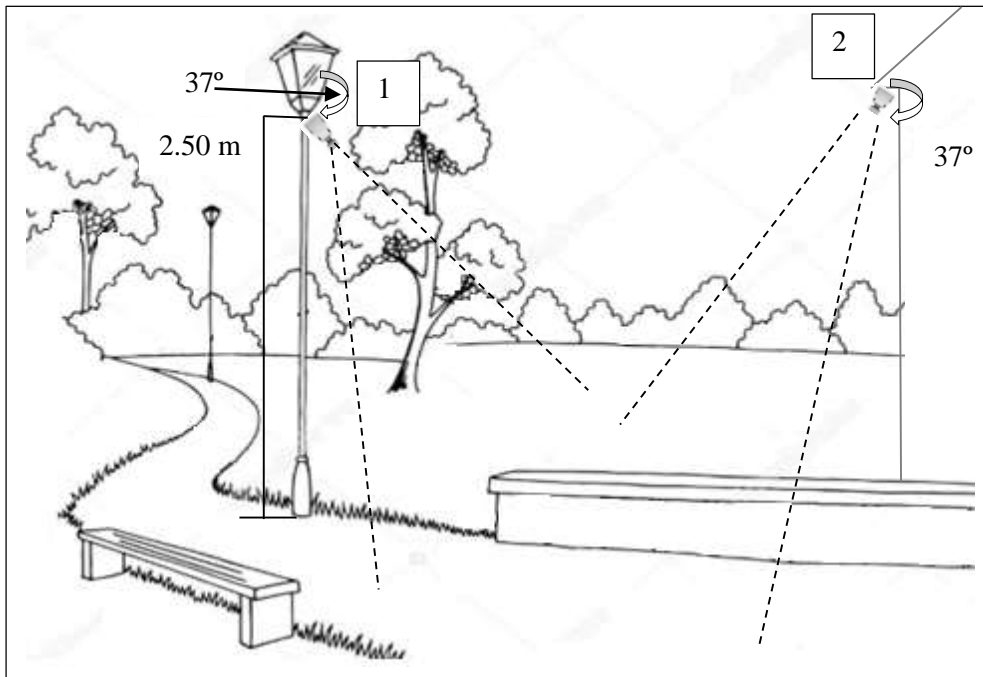


Figura 13-2. Ubicación de los nodos en un parque

Realizado por: OÑATE, Fernando 2020

Fuente: <https://sp.depositphotos.com/166721412/stock-illustration-park-fountain-graphic-black-white.html>

Como se mencionó anteriormente el sistema busca ser adaptativo a diferentes entornos, por lo tanto, se implementó en tres ambientes que utilizan video vigilancia con mayor frecuencia como son: una oficina, un automóvil y un parque público; a continuación, se describe la implementación y las pruebas realizadas, en las figuras 14-2, 15-2, 16-2, 17-2, 18-2 y 19-2 se muestran el sistema implementado en los ambientes de oficina, automóvil y parque respectivamente.



Figura 14-2. Sistema implementado (oficina)

Realizado por: OÑATE, Fernando 2020

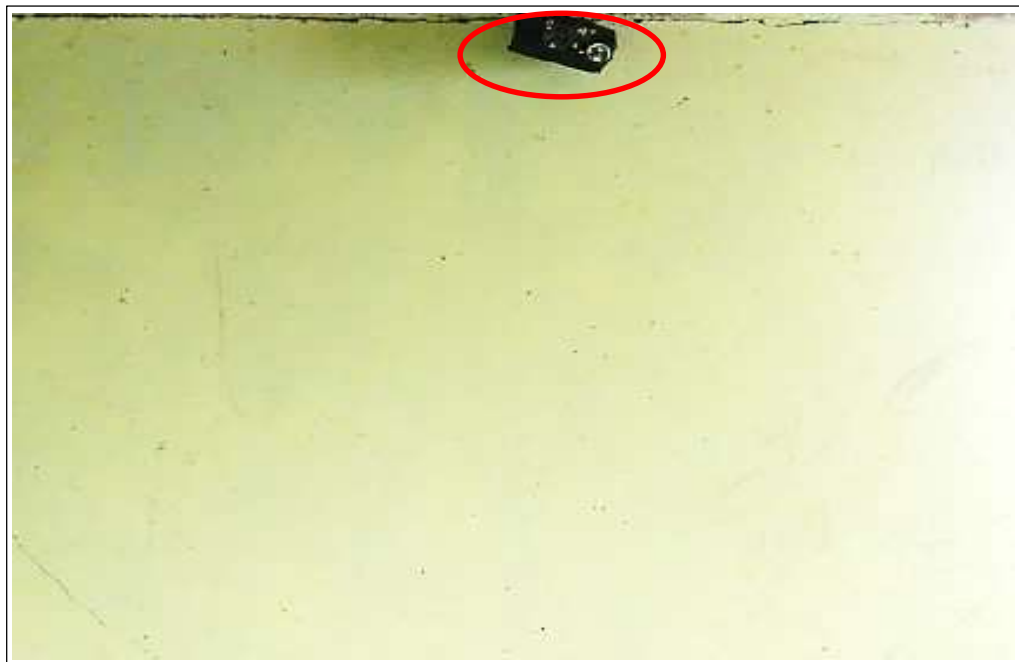


Figura 15-2. Sistema implementado (oficina)

Realizado por: OÑATE, Fernando 2020



Figura 16-2. Sistema implementado (automóvil)

Realizado por: OÑATE, Fernando 2020



Figura 17-2. Sistema implementado (parque)

Realizado por: OÑATE, Fernando 2020



Figura 18-2. Sistema implementado (parque)

Realizado por: OÑATE, Fernando 2020

CAPÍTULO III

3 VALIDACIÓN DEL PROTOTIPO

En el presente capítulo se muestra los resultados obtenidos del prototipo en operación, al operar en tiempo real las pruebas consistieron en medir: precisión y sensibilidad del detector, tiempo de procesamiento, tiempo de envío y recepción de la alerta GSM, tiempo de envío y almacenamiento de la imagen en la base de datos de Gmail, consumo total de corriente.

3.1 Prueba de precisión y sensibilidad del detector

Se ha utilizado una matriz de confusión para determinar la precisión del detector, hay que considerar que la precisión depende de gran manera de las configuraciones del programa de acuerdo al ambiente que se desee adaptar, al tener un ambiente con mayor iluminación se aumentó el número de vecinos, esto produce que el sistema detecte de una mejor manera el objeto en análisis y al disminuir la iluminación en un ambiente se redujo el número de vecinos, aunque al realizar esto el sistema está propenso a aumentar el número de falsos positivos.

Los resultados variarán de acuerdo con el ambiente en análisis, en las figuras 1-3, 2-3, 3-3 y 4-3 se muestra la correcta detención del objeto en ambiente de oficina.



Figura 1-3. Detección positiva de rifle y metralleta en ambiente de oficina

Realizado por: OÑATE, Fernando 2020



Figura 2-3. Detección positiva de pistola y recortada en ambiente de oficina

Realizado por: OÑATE, Fernando 2020



Figura 3-3. Detección positiva de cuchillo y pistola en ambiente de oficina

Realizado por: OÑATE, Fernando 2020



Figura 4-3. Detección positiva de escopeta y pistola en ambiente de oficina

Realizado por: OÑATE, Fernando 2020

En la figura 5-3 se muestra el rostro enviado a la base de datos de Gmail junto con la imagen que contiene el objeto detectado, por lo tanto, se realizó el recorte facial y un redimensionamiento de la imagen.



Figura 5-3. Recorte facial de la imagen en ambiente de oficina

Realizado por: OÑATE, Fernando 2020

En las tablas 1-3, 3-3, y 5-3 se muestran los resultados obtenidos de la prueba en el ambiente de automóvil y en las tablas 2-3, 4-3 y 6-3 se muestran las matrices de confusión que permiten evaluar la precisión y sensibilidad del sistema, para la prueba se realizaron 10 muestras de cada evento.

Tabla 1-3: Resultados con variable a plena luz del día oficina

| Objeto | Detección Positiva | Detección Negativa | Falso Positivo | Falso negativo |
|------------|--------------------|--------------------|----------------|----------------|
| Pistola | 10 | 0 | 0 | 0 |
| Cuchillo | 10 | 0 | 0 | 0 |
| Metrallata | 10 | 0 | 0 | 0 |
| Rifle | 10 | 0 | 0 | 0 |
| Escopeta | 10 | 0 | 0 | 0 |
| Recortada | 9 | 1 | 1 | 0 |
| Total | 59 | 1 | 1 | 0 |

Realizado por: OÑATE, Fernando 2020

Tabla 2-3: Matriz de confusión con variable plena luz de día oficina

| MATRIZ DE CONFUSIÓN ARMAS | | Valor de predicción | |
|---------------------------|-----------|---------------------|-----------|
| | | Positivos | Negativos |
| Valor real | Positivos | 59 | 0 |
| | Negativos | 1 | 1 |

Realizado por: OÑATE, Fernando 2020

Para obtener un valor cuantitativo de utilizó la ecuación de precisión descrita anteriormente:

Ecuación 1-3. Precisión con variable a plena luz de día en oficina

$$Precisión = \frac{59}{60} * 100$$

De donde se obtuvo un valor de: 98.33 %

El valor cuantitativo de sensibilidad o TVP se obtuvo mediante:

Ecuación 2-3. Sensibilidad con variable a plena luz de día en oficina

$$TVP = \frac{59}{60} * 100$$

Dando como resultado un valor de: 98.33 %

Tabla 3-3: Resultados con variable de día nublado en oficina

| Objeto | Detección Positiva | Detección Negativa | Falso Positivo | Falso negativo |
|------------|--------------------|--------------------|----------------|----------------|
| Pistola | 9 | 1 | 0 | 1 |
| Cuchillo | 9 | 1 | 1 | 0 |
| Metralleta | 9 | 1 | 1 | 0 |
| Rifle | 10 | 0 | 0 | 0 |
| Escopeta | 10 | 0 | 0 | 0 |
| Recortada | 9 | 1 | 1 | 0 |
| Total | 56 | 4 | 3 | 1 |

Realizado por: OÑATE, Fernando 2020

Tabla 4-3: Matriz de confusión con variable de día nublado en oficina

| MATRIZ DE CONFUSIÓN ARMAS | | Valor de predicción | |
|---------------------------|-----------|---------------------|-----------|
| | | Positivos | Negativos |
| Valor real | Positivos | 56 | 1 |
| | Negativos | 3 | 4 |

Realizado por: OÑATE, Fernando 2020

Para obtener un valor cuantitativo de utilizó la ecuación de precisión descrita anteriormente:

Ecuación 3-3. Precisión con variable de día nublado en oficina

$$Precisión = \frac{56}{59} * 100$$

De donde se obtuvo un valor de: 94.91 %

El valor cuantitativo de sensibilidad o TVP se obtuvo mediante:

Ecuación 4-3. Sensibilidad con variable de día nublado en oficina

$$TVP = \frac{56}{60} * 100$$

Dando como resultado un valor de: 93.33 %

Tabla 5-3: Resultados con variable de día con lluvia en oficina

| Objeto | Detección Positiva | Detección Negativa | Falso Positivo | Falso negativo |
|------------|--------------------|--------------------|----------------|----------------|
| Pistola | 9 | 1 | 0 | 1 |
| Cuchillo | 8 | 2 | 1 | 1 |
| Metralleta | 9 | 1 | 1 | 0 |
| Rifle | 10 | 0 | 0 | 0 |
| Escopeta | 9 | 1 | 0 | 1 |
| Recortada | 8 | 2 | 1 | 1 |
| Total | 53 | 7 | 3 | 4 |

Realizado por: OÑATE, Fernando 2020

Tabla 6-3: Matriz de confusión con variable día con lluvia en oficina

| MATRIZ DE CONFUSIÓN ARMAS | | Valor de predicción | |
|---------------------------|-----------|---------------------|-----------|
| | | Positivos | Negativos |
| Valor real | Positivos | 53 | 4 |
| | Negativos | 3 | 7 |

Realizado por: OÑATE, Fernando 2020

Para obtener un valor cuantitativo de utilizó la ecuación de precisión descrita anteriormente:

Ecuación 5-3. Precisión con variable de día con lluvia en oficina

$$Precisión = \frac{53}{56} * 100$$

De donde se obtuvo un valor de: 94.64 %

El valor cuantitativo de sensibilidad o TVP se obtuvo mediante:

Ecuación 6-3. Sensibilidad con variable de día con lluvia en oficina

$$TVP = \frac{53}{60} * 100$$

Dando como resultado un valor de: 88.33 %

En la figura 6-3, 7-3, 8-3 y 9-3 se puede mirar los resultados obtenidos de la prueba de precisión y sensibilidad del detector en el ambiente de automóvil.



Figura 6-3. Detección positiva de metralleta y recortada en ambiente de automóvil

Realizado por: OÑATE, Fernando 2020



Figura 7-3. Detección positiva de pistola y cuchillo en ambiente de automóvil

Realizado por: OÑATE, Fernando 2020



Figura 8-3. Detección positiva de escopeta y cuchillo en ambiente de automóvil

Realizado por: OÑATE, Fernando 2020



Figura 9-3. Detección positiva de pistola y rifle en ambiente de automóvil

Realizado por: OÑATE, Fernando 2020

En la figura 10-3 se muestra el rostro enviado a la base de datos de Gmail junto con la imagen que contiene el objeto detectado, por lo tanto, se realizó el recorte facial y un redimensionamiento de la imagen.



Figura 10-3. Recorte facial de la imagen en ambiente de automóvil

Realizado por: OÑATE, Fernando 2020

En las tablas 7-3, 9-3, y 11-3 se muestran los resultados obtenidos de la prueba en el ambiente de automóvil y en las tablas 8-3, 10-3 y 12-3 se muestran las matrices de confusión que permiten evaluar la precisión y sensibilidad del sistema, para la prueba se realizaron 10 muestras de cada evento.

Tabla 7-3: Resultados con variable a plena luz del día en automóvil

| Objeto | Detección Positiva | Detección Negativa | Falso Positivo | Falso negativo |
|------------|--------------------|--------------------|----------------|----------------|
| Pistola | 10 | 0 | 0 | 0 |
| Cuchillo | 10 | 0 | 0 | 0 |
| Metralleta | 10 | 0 | 0 | 0 |
| Rifle | 10 | 0 | 0 | 0 |
| Escopeta | 10 | 0 | 0 | 0 |
| Recortada | 9 | 1 | 1 | 0 |
| Total | 59 | 1 | 1 | 0 |

Realizado por: OÑATE, Fernando 2020

Tabla 8-3: Matriz de confusión con variable plena luz en automóvil

| MATRIZ DE CONFUSIÓN ARMAS | | Valor de predicción | |
|---------------------------|-----------|---------------------|-----------|
| | | Positivos | Negativos |
| Valor real | Positivos | 59 | 0 |
| | Negativos | 1 | 1 |

Realizado por: OÑATE, Fernando 2020

Para obtener un valor cuantitativo de utilizó la ecuación de precisión descrita anteriormente:

Ecuación 7-3. Precisión con variable a plena luz de día en automóvil

$$Precisión = \frac{59}{60} * 100$$

De donde se obtuvo un valor de: 98.33 %

El valor cuantitativo de sensibilidad o TVP se obtuvo mediante:

Ecuación 8-3. Sensibilidad con variable a plena luz de día en automóvil

$$TVP = \frac{59}{60} * 100$$

Dando como resultado un valor de: 98.33 %

Tabla 9-3: Resultados con variable de día nublado en automóvil

| Objeto | Detección Positiva | Detección Negativa | Falso Positivo | Falso negativo |
|------------|--------------------|--------------------|----------------|----------------|
| Pistola | 9 | 1 | 0 | 1 |
| Cuchillo | 9 | 1 | 1 | 0 |
| Metralleta | 9 | 1 | 1 | 0 |
| Rifle | 10 | 0 | 0 | 0 |
| Escopeta | 10 | 0 | 0 | 0 |
| Recortada | 8 | 2 | 2 | 0 |
| Total | 55 | 5 | 4 | 1 |

Realizado por: OÑATE, Fernando 2020

Tabla 10-3: Matriz de confusión con variable día nublado en automóvil

| MATRIZ DE CONFUSIÓN ARMAS | | Valor de predicción | |
|---------------------------|-----------|---------------------|-----------|
| | | Positivos | Negativos |
| Valor real | Positivos | 55 | 1 |
| | Negativos | 4 | 5 |

Realizado por: OÑATE, Fernando 2020

Para obtener un valor cuantitativo de utilizó la ecuación de precisión descrita anteriormente:

Ecuación 9-3. Precisión con variable de día nublado en automóvil

$$Precisión = \frac{55}{59} * 100$$

De donde se obtuvo un valor de: 93.22 %

El valor cuantitativo de sensibilidad o TVP se obtuvo mediante:

Ecuación 10-3. Sensibilidad con variable de día nublado en automóvil

$$TVP = \frac{55}{60} * 100$$

Dando como resultado un valor de: 91.66 %

Tabla 11-3: Resultados con variable de día con lluvia en automóvil

| Objeto | Detección Positiva | Detección Negativa | Falso Positivo | Falso negativo |
|------------|--------------------|--------------------|----------------|----------------|
| Pistola | 8 | 2 | 1 | 1 |
| Cuchillo | 8 | 2 | 1 | 1 |
| Metralleta | 8 | 2 | 2 | 0 |
| Rifle | 10 | 0 | 0 | 0 |
| Escopeta | 9 | 1 | 0 | 1 |
| Recortada | 8 | 2 | 1 | 1 |
| Total | 51 | 9 | 5 | 4 |

Realizado por: OÑATE, Fernando 2020

Tabla 12-3: Matriz de confusión con variable día con lluvia en automóvil

| MATRIZ DE CONFUSIÓN ARMAS | | Valor de predicción | |
|---------------------------|-----------|---------------------|-----------|
| | | Positivos | Negativos |
| Valor real | Positivos | 51 | 4 |
| | Negativos | 5 | 9 |

Realizado por: OÑATE, Fernando 2020

Para obtener un valor cuantitativo de utilizó la ecuación de precisión descrita anteriormente:

Ecuación 11-3. Precisión con variable de día con lluvia en automóvil

$$Precisión = \frac{51}{56} * 100$$

De donde se obtuvo un valor de: 91.07 %

El valor cuantitativo de sensibilidad o TVP se obtuvo mediante:

Ecuación 12-3. Sensibilidad con variable de día con lluvia en automóvil

$$TVP = \frac{51}{60} * 100$$

Dando como resultado un valor de: 85.00 %

En las figuras 11-3, 12-3 y 13-3 se puede observar los resultados obtenidos de la prueba de precisión y sensibilidad del detector en el ambiente de parque.



Figura 11-3. Detección positiva de metralleta y recortada en ambiente de parque

Realizado por: OÑATE, Fernando 2020



Figura 12-3. Detección positiva de cuchillo y escopeta en ambiente de parque

Realizado por: OÑATE, Fernando 2020



Figura 13-3. Detección positiva de rifle y pistola en ambiente de parque

Realizado por: OÑATE, Fernando 2020

En la figura 14-3 se muestra el rostro enviado a la base de datos de Gmail junto con la imagen que contiene el objeto detectado, por lo tanto, se realizó el recorte facial y un redimensionamiento de la imagen.



Figura 14-3. Recorte facial de la imagen en ambiente de parque

Realizado por: OÑATE, Fernando 2020

En las tablas 13-3, 15-3, y 17-3 se muestran los resultados obtenidos de la prueba en el ambiente de automóvil y en las tablas 14-3, 16-3 y 18-3 se muestran las matrices de confusión que permiten evaluar la precisión y sensibilidad del sistema, para la prueba se realizaron 10 muestras de cada evento.

Tabla 13-3: Resultados con variable a plena luz del día en parque

| Objeto | Detección Positiva | Detección Negativa | Falso Positivo | Falso negativo |
|------------|--------------------|--------------------|----------------|----------------|
| Pistola | 9 | 1 | 1 | 0 |
| Cuchillo | 8 | 2 | 1 | 1 |
| Metralleta | 10 | 0 | 0 | 0 |
| Rifle | 10 | 0 | 0 | 0 |
| Escopeta | 10 | 0 | 0 | 0 |
| Recortada | 9 | 1 | 1 | 0 |
| Total | 56 | 4 | 3 | 1 |

Realizado por: OÑATE, Fernando 2020

Tabla 14-3: Matriz de confusión con variable a plena luz en parque

| MATRIZ DE CONFUSIÓN ARMAS | | Valor de predicción | |
|---------------------------|-----------|---------------------|-----------|
| | | Positivos | Negativos |
| Valor real | Positivos | 56 | 1 |
| | Negativos | 3 | 4 |

Realizado por: OÑATE, Fernando 2020

Para obtener un valor cuantitativo de utilizó la ecuación de precisión descrita anteriormente:

Ecuación 13-3. Precisión con variable a plena luz de día en parque

$$Precisión = \frac{56}{59} * 100$$

De donde se obtuvo un valor de: 94.91 %

El valor cuantitativo de sensibilidad o TVP se obtuvo mediante:

Ecuación 14-3. Sensibilidad con variable a plena luz de día en parque

$$TVP = \frac{56}{60} * 100$$

Dando como resultado un valor de: 93.33 %

Tabla 15-3: Resultados con variable de día nublado en parque

| Objeto | Detección Positiva | Detección Negativa | Falso Positivo | Falso negativo |
|------------|--------------------|--------------------|----------------|----------------|
| Pistola | 8 | 2 | 1 | 1 |
| Cuchillo | 8 | 2 | 2 | 0 |
| Metralleta | 9 | 1 | 1 | 0 |
| Rifle | 10 | 0 | 0 | 0 |
| Escopeta | 10 | 0 | 0 | 0 |
| Recortada | 8 | 2 | 2 | 0 |
| Total | 53 | 7 | 6 | 1 |

Realizado por: OÑATE, Fernando 2020

Tabla 16-3: Matriz de confusión con variable de día nublado en parque

| MATRIZ DE CONFUSIÓN ARMAS | | Valor de predicción | |
|---------------------------|-----------|---------------------|-----------|
| | | Positivos | Negativos |
| Valor real | Positivos | 53 | 1 |
| | Negativos | 6 | 7 |

Realizado por: OÑATE, Fernando 2020

Para obtener un valor cuantitativo de utilizó la ecuación de precisión descrita anteriormente:

Ecuación 15-3. Precisión con variable de día nublado en parque

$$Precisión = \frac{53}{59} * 100$$

De donde se obtuvo un valor de: 89.83 %

El valor cuantitativo de sensibilidad o TVP se obtuvo mediante:

Ecuación 16-3. Sensibilidad con variable de día nublado en parque

$$TVP = \frac{53}{60} * 100$$

Dando como resultado un valor de: 88.33 %

Tabla 17-3: Resultados con variable de día con lluvia en parque

| Objeto | Detección Positiva | Detección Negativa | Falso Positivo | Falso negativo |
|------------|--------------------|--------------------|----------------|----------------|
| Pistola | 7 | 3 | 2 | 1 |
| Cuchillo | 6 | 4 | 2 | 2 |
| Metralleta | 7 | 3 | 2 | 1 |
| Rifle | 8 | 2 | 1 | 1 |
| Escopeta | 8 | 2 | 1 | 1 |
| Recortada | 6 | 4 | 2 | 2 |
| Total | 42 | 18 | 10 | 8 |

Realizado por: ONATE, Fernando 2020

Tabla 18-3: Matriz de confusión con variable día con lluvia en parque

| MATRIZ DE CONFUSIÓN ARMAS | | Valor de predicción | |
|---------------------------|-----------|---------------------|-----------|
| | | Positivos | Negativos |
| Valor real | Positivos | 42 | 8 |
| | Negativos | 10 | 18 |

Realizado por: ONATE, Fernando 2020

Para obtener un valor cuantitativo de utilizó la ecuación de precisión descrita anteriormente:

Ecuación 17-3. Precisión con variable de día con lluvia en parque

$$Precisión = \frac{42}{52} * 100$$

De donde se obtuvo un valor de: 80.76 %

El valor cuantitativo de sensibilidad o TVP se obtuvo mediante:

Ecuación 18-3. Sensibilidad con variable de día con lluvia en parque

$$TVP = \frac{42}{60} * 100$$

Dando como resultado un valor de: 70.00 %

En la tabla 19-3 se puede observar un resumen de los resultados obtenidos en las pruebas de precisión y sensibilidad del detector en los diferentes ambientes.

Tabla 19-3: Resumen de resultados

| VARIABLES | AMBIENTE | PRECISIÓN (%) | SENSIBILIDAD (%) |
|-----------|-----------|---------------|------------------|
| PLENA LUZ | OFICINA | 98,33 | 98,33 |
| | AUTOMÓVIL | 98,33 | 98,33 |
| | PARQUE | 94,91 | 93,33 |
| NUBLADO | OFICINA | 94,91 | 93,33 |
| | AUTOMÓVIL | 93,22 | 91,66 |
| | PARQUE | 89,83 | 88,33 |
| LLUVIA | OFICINA | 94,64 | 88,63 |
| | AUTOMÓVIL | 91,07 | 87,00 |
| | PARQUE | 80,76 | 70,00 |
| PROMEDIO | | 92,89 | 89,88 |

Realizado por: OÑATE, Fernando 2020

En base a los resultados de la tabla 19-3 se puede apreciar el porcentaje de precisión que posee el sistema, ello varía de acuerdo al ambiente dependiendo de factores externos propios de cada uno de ellos como son: iluminación, distancia del nodo hacia el área de supervisión y ángulo de visión.

En el gráfico 1-3 se puede observar el resumen de las evaluaciones realizadas en el sistema en cada uno de los ambientes.

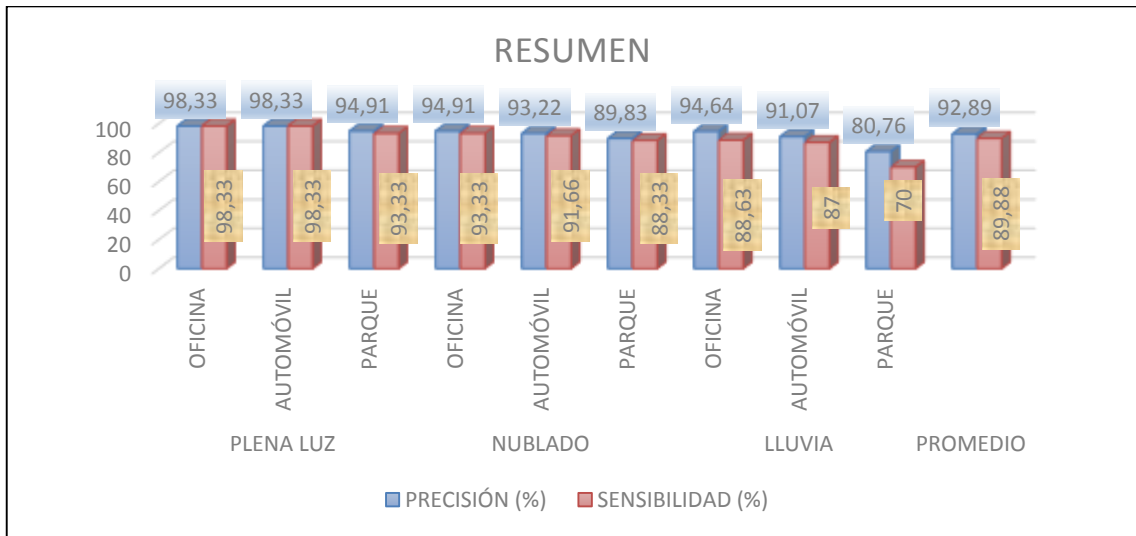


Gráfico 1-3. Resumen de la prueba de precisión y sensibilidad

Realizado por: OÑATE, Fernando 2020

La detección de objetos se ve afectado directamente por la distancia en la que se encuentra, mientras más lejos está ubicado el elemento de la cámara menor será la capacidad de detección. Mediante esta prueba se determina la distancia mínima de detección que en este caso de estudio resultado ser de 3,7 metros para objetos pequeños y de 4,7 metros para objetos de mayor tamaño, como se muestra en la tabla 20-3.

Tabla 20-3: Resultados de la prueba de detección de objetos

| OBJETOS | DISTANCIA / DETECCIÓN | | | | | | | | |
|------------|-----------------------|------|------|------|------|------|------|------|------|
| | 2,5m | 2,7m | 3,0m | 3,5m | 3,7m | 4,0m | 4,5m | 4,7m | 5,0m |
| Pistolas | Si | Si | Si | Si | Si | Si | No | No | No |
| Rifle | Si | Si | Si | Si | Si | Si | Si | Si | No |
| Escopeta | Si | Si | Si | Si | Si | Si | Si | Si | No |
| Cuchillo | Si | Si | Si | Si | Si | No | No | No | No |
| Metralleta | Si | Si | Si | Si | Si | Si | No | Si | No |
| Recortada | Si | Si | Si | Si | Si | No | No | No | No |

Realizado por: OÑATE, Fernando 2020

3.2 Prueba del tiempo de procesamiento

La prueba se realizó con el único objetivo de medir el tiempo de procesamiento que transcurre durante la ejecución del programa, el dispositivo que se está usando para obtener las imágenes trabaja a 30 cuadros por segundo (FPS) por lo tanto el tiempo de procesamiento de cada imagen

va a depender directamente de ello, como se tiene una cantidad limitada de procesamiento de la tarjeta embebida se ha reducido a 8 cuadros por segundo con 1280x720 pixeles de resolución, en la figura 15-3 se muestra la obtención de datos para realizar la prueba.



Figura 15-3. Prueba de medición del tiempo de procesamiento

Realizado por: OÑATE, Fernando 2020

El tiempo de procesamiento cuando el programa no detecta armas (negativo) es de 80.10 ms, y el tiempo de procesamiento cuando se ha detectado un arma (positivo) es de 130.23 ms, dándonos la conclusión que aun en su máximo tiempo de procesamiento no afecta en el rendimiento del sistema lo que no produce retardos de procesamiento, en la tabla 21-3 se observa los datos de los tiempos obtenidos en la prueba.

Tabla 21-3: Resultados de la prueba de procesamiento de imágenes

| Muestra | Estado | Tiempo(ms) |
|---------|----------|------------|
| 1 | Negativo | 80.10 |
| 2 | Positivo | 130.23 |
| 3 | Negativo | 80.40 |
| 4 | Positivo | 131.40 |
| 5 | Negativo | 80.70 |
| 6 | Positivo | 131.67 |

Realizado por: OÑATE, Fernando 2020

Como se observa en la tabla 21-3 el sistema procesa las imágenes en un muy poco tiempo lo que es muy satisfactorio ya que el sistema debe mantener el monitoreo en un tiempo real, cuando el sistema detecta el objeto en análisis el tiempo de procesamiento aumenta, aun así, el tiempo sigue siendo muy corto por lo tanto no afecta el correcto funcionamiento y desempeño del mismo, en

el gráfico 2-3 se puede observar el resumen de la evaluación del tiempo de procesamiento del sistema.

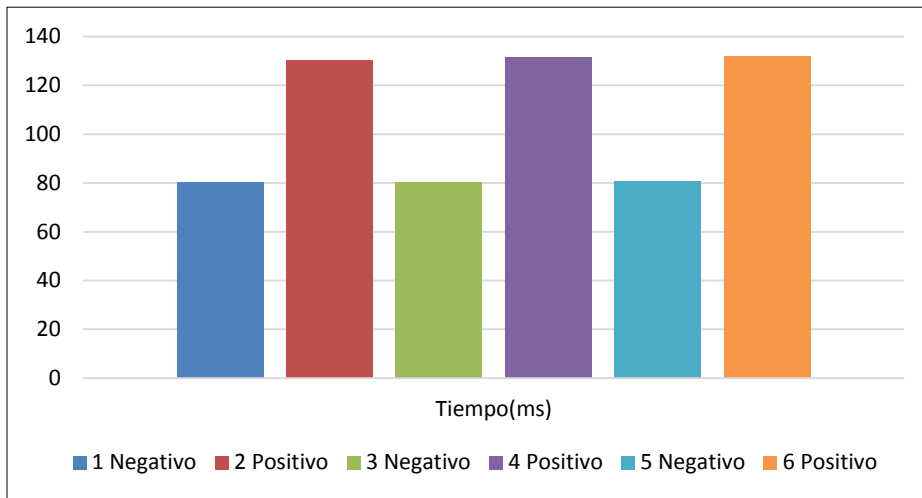


Gráfico 2-3. Resumen de la prueba de tiempo de procesamiento

Realizado por: OÑATE, Fernando 2020

3.3 Prueba del tiempo de envío y recepción de la alerta GSM

Se realizó esta prueba para determinar el tiempo de envío y recepción de la alerta GSM tomando en cuenta que el sistema requiere que se envíe la alerta en el menor tiempo posible, en la figura 16-3 se puede observar cómo se midió el tiempo de envío y recepción de la alerta GSM.



Figura 16-3. Medición de tiempo de envío y recepción de alerta GSM

Realizado por: OÑATE, Fernando 2020

En la tabla 22-3 se observa los resultados obtenidos de la prueba, hay que tomar en cuenta que el proceso de recepción de la alerta GSM depende en gran parte de la operadora telefónica del mismo, para esta prueba se utilizó un chip GSM de la operadora telefónica Moviestar.

Tabla 22-3: Resultados de la prueba de envío y recepción alerta GSM

| Muestra | Hora de envío | Hora de llegada | Tiempo (segundos) |
|---------|---------------|-----------------|-------------------|
| 1 | 8:43 | 8:49 | 6 |
| 2 | 9:11 | 9:18 | 7 |
| 3 | 9:25 | 9:32 | 7 |
| 4 | 9:40 | 9:46 | 6 |
| 5 | 9:51 | 9:57 | 6 |

Realizado por: OÑATE, Fernando 2020

Con los resultados obtenidos se puede constatar que el tiempo de envío y recepción de la alerta GSM es un tiempo óptimo, en el gráfico 3-3 se puede observar el resumen de los tiempos obtenidos durante la evaluación del sistema.

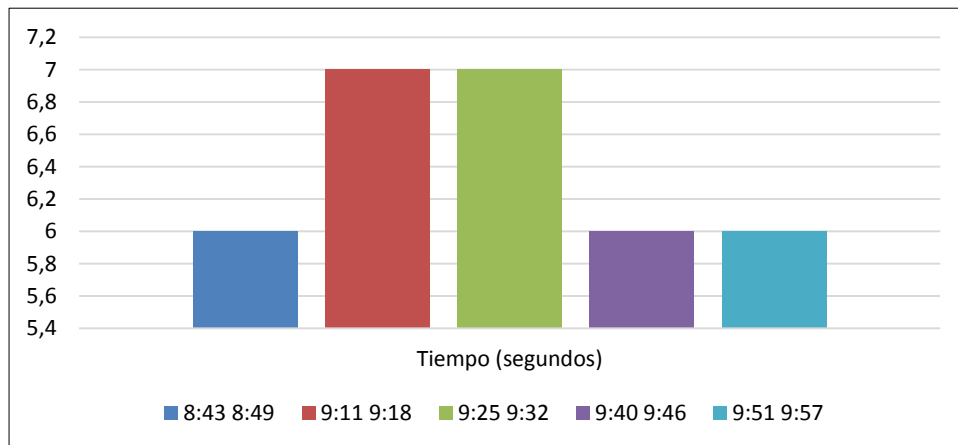


Gráfico 3-3. Resumen de la prueba tiempo de envío y recepción GSM

Realizado por: OÑATE, Fernando 2020

3.4 Prueba del tiempo de envío y almacenamiento de la imagen en la base de datos de Gmail

Se realizó esta prueba con el único objetivo de medir el tiempo que se demora el sistema en enviar la imagen positiva a la base de datos de Gmail, en la figura 17-3 se observa cómo se realizó la medición del tiempo de esta prueba.



Figura 17-3. Medición de tiempo de envío y almacenamiento en Gmail

Realizado por: OÑATE, Fernando 2020

Hay que tomar en cuenta que el sistema envía la imagen en muy poco tiempo, pero dependerá directamente de la velocidad del internet que se esté usando para que la imagen llegue a su destino, las pruebas se las realizó con un internet residencial de 3 Megabytes por segundo, en la tabla 23-3 se observa los resultados obtenidos en esta prueba.

Tabla 23-3: Resultados de la prueba de envío y almacenamiento de Gmail

| Muestra | Hora de envío | Hora de almacenamiento | Tiempo (segundos) |
|---------|---------------|------------------------|-------------------|
| 1 | 8:43 | 8:53 | 10 |
| 2 | 9:11 | 9:20 | 9 |
| 3 | 9:25 | 9:34 | 9 |
| 4 | 9:40 | 9:51 | 11 |
| 5 | 9:51 | 10:00 | 9 |

Realizado por: OÑATE, Fernando 2020

Con los resultados de las evaluaciones se puede constatar que el envío y almacenamiento de la imagen en la cual se ha detectado el objeto en análisis es muy corto, lo que permite que el sistema ofrezca sea en tiempo real, cumpliendo con el objetivo planteado en el diseño. En el gráfico 5-3 se puede observar un resumen de los tiempos obtenidos durante la ejecución de la prueba.

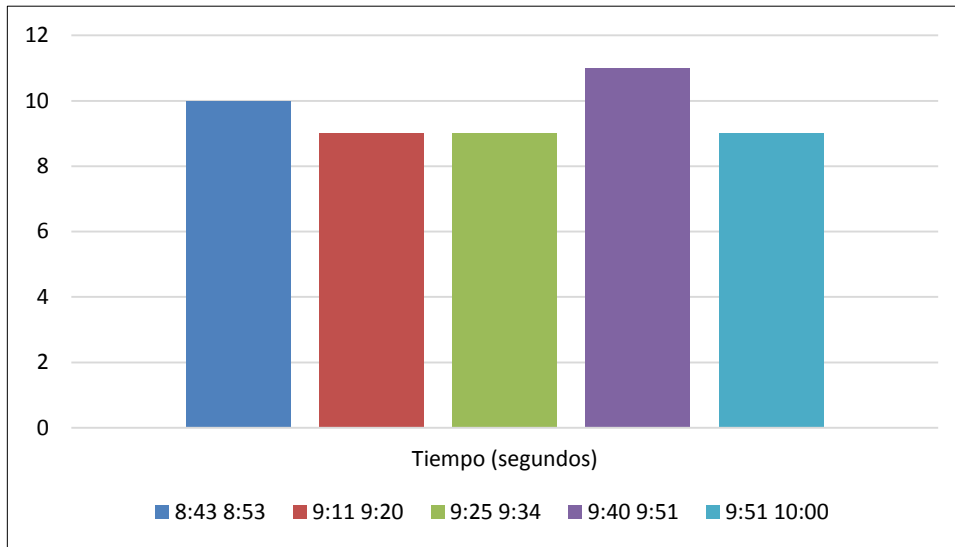


Gráfico 4-3. Resumen de la prueba de envío y almacenamiento en Gmail

Realizado por: OÑATE, Fernando 2020

3.5 Prueba de consumo total de corriente

La prueba se la realizo para verificar el consumo de energía del sistema mediante la ayuda de un multímetro digital UT139 como se observa en la figura 18-3. La prueba muestra una comparación con los datos descritos en la hoja técnica de los dispositivos usados para de esta manera poder obtener el valor del consumo total de energía.

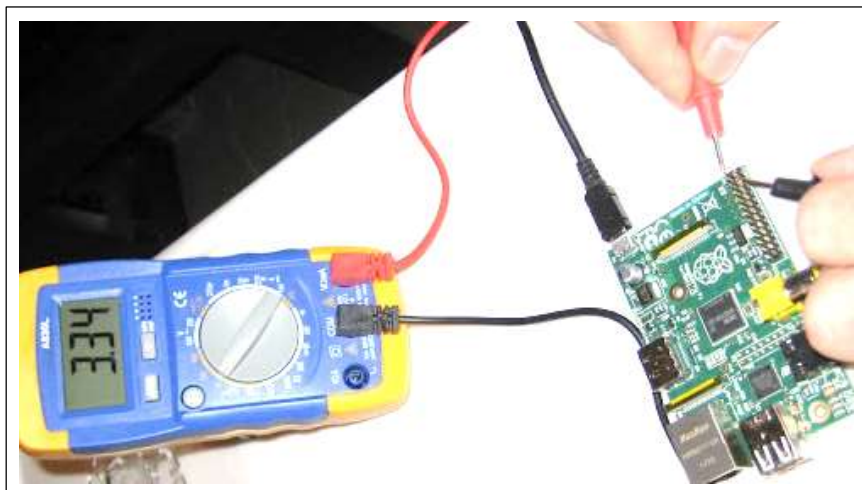


Figura 18-3. Medición del consumo de corriente del dispositivo

Realizado por: OÑATE, Fernando 2020

En la tabla 24-3 se muestran los valores de corriente total consumida por el dispositivo en funcionamiento, hay que tomar en cuenta que el dispositivo consume corriente mínima en reposo y una corriente máxima a pleno rendimiento.

Tabla 24-3: Resultados de la prueba de consumo de corriente del dispositivo

| N | Dispositivo | Corriente mínima (mA) | Voltaje mínimo (V) | Corriente máxima (mA) | Voltaje máximo (V) |
|-----------------|--------------------------|-----------------------|--------------------|-----------------------|--------------------|
| 1 | Raspberry pi 3 modelo B+ | 230 | 4.76 | 350 | 3.83 |
| 2 | Módulo GSM | 1 | 4.2 | 1000 | 3.7 |
| 3 | Módulo de cámara | 100 | 4.5 | 500 | 3.96 |
| Corriente total | | 331 | | 1850 | |

Realizado por: OÑATE, Fernando 2020

CAPITULO IV

4 EVALUACIÓN ECONÓMICA

En el presente capítulo se analizará el beneficio/costo del dispositivo, los costos varían de acuerdo al número de nodos implementados, la tabla 25-3 muestra el análisis económico de un nodo y posteriormente el costo del sistema, donde se detallan sus componentes y sus respectivos precios de implementación.

Tabla 1-4: Análisis de costos

| N | Módulos | Cantidad | Precio (USD) |
|--|----------------------------------|----------|--------------|
| 1 | Raspberry Pi3 modelo B+ | 1 | 80 |
| 2 | Cámara de raspberry V 1.2 | 1 | 30 |
| 3 | Tarjeta micro SD High Speed | 1 | 10 |
| 4 | Módulo GSM 800L | 1 | 30 |
| 5 | Case de raspberry Pi 3 modelo B+ | 1 | 10 |
| 6 | Varios | | 40 |
| COSTO TOTAL POR NODO | | | 200 |
| COSTO TOTAL DEL SISTEMA (5 nodos) | | | 1000 |

Realizado por: OÑATE, Fernando 2020

4.1 Relación beneficio/costo

Como se puede observar el precio del prototipo es de 200 dólares americanos, el costo del sistema de detección va a variar de acuerdo con el número de nodos implementados, en la tabla 2-4 se muestra una comparación beneficio/costo entre el sistema de detección y un sistema de vigilancia compuesto por personal humano.

Tabla 2-4: Relación beneficio/costo

| | Costo Anual | Extras | Total |
|------------------------|-------------|--------|-------------|
| Prototipo | 1000 | 300 | 1300 |
| Operador de monitoreo | 4800 | 800 | 5600 |
| Beneficio/Costo | | | 4300 |

Realizado por: OÑATE, Fernando 2020

Tomando en cuenta el valor del beneficio/costo y recalando que el sistema posee una precisión mayor al 90% y un operador de monitoreo en promedio perderá hasta un 45% de actividad de pantalla al transcurso de 12 minutos de continuo monitoreo, a menudo que el tiempo aumenta tendrá una tasa de fallo de 90% al transcurso de 22 minutos; se puede indudablemente tener un beneficio/costo muy amplio en comparación al sistema tradicional compuesto por operadores de monitoreo.

El prototipo construido invita a los investigadores a seguir desarrollando algoritmos más avanzados para la detección de objetos junto con el avance de la tecnología y desarrollo de nuevos dispositivos electrónicos, con ello se busca el inicio de las ciudades inteligentes, mismas que en un futuro próximo predominarán a nivel mundial.

CONCLUSIONES

Se ha logrado diseñar y construir nodos inteligentes para detección de armas dentro de una red de video-vigilancia utilizando visión artificial orientado a varios entornos en los cuales se logró detectar satisfactoriamente el objeto en análisis con un beneficio/costo muy aceptable.

Se obtuvo el algoritmo para la detección de armas mediante visión artificial que utilizó el sistema, en base a la investigación previa de conocimientos teóricos y prácticos sobre el procesamiento de imágenes aplicados en la detección de objetos.

El prototipo es capaz de realizar la detección de armas en un ambiente en tiempo real mediante el uso de un algoritmo de detección que envía una alerta GSM a un dispositivo inteligente.

El software y hardware utilizados en el prototipo es de bajo costo y alta eficiencia. Por el avance de la tecnología se ha logrado cumplir con la selección de la tarjeta embebida Raspberry Pi 3 modelo B+, la cámara de Raspberry Pi V1.3 y el módulo GSM 800L. Y para el software se ha utilizado el lenguaje de programación Python 3.7.4 y con ayuda de la librería OpenCV 4.1.1.

Con las pruebas se determinó que a pesar que los objetos en análisis son pequeños y de difícil detección a comparación de vehículos o personas, los nodos cumplen con la detección de armas en los diferentes ambientes. El sistema logra enviar la alerta GSM en un tiempo promedio de 6.5 segundos, enviar y almacenar la imagen detectada en la base de datos de Gmail en un tiempo promedio de 10 segundos lo que nos permite tener una detección en tiempo real, envío de una alerta GSM y almacenamiento de la imagen en la base de datos de Gmail en un tiempo bastante corto, dando así cumplimiento con todos los objetivos planteados al inicio de la investigación.

RECOMENDACIONES

Trabajar sistemas embebidos con capacidad de procesamiento en paralelo como son las tarjetas Nvidia diseñadas para visión artificial de la serie Jetson las cuales brindan muchas prestaciones como un gran número de núcleos de procesamiento GPUs, NPUs, además de programación en paralelismo con librerías disponibles y listas para utilizar conjuntamente en Python y que además brindan soporte en línea desde su página oficial.

Para reducir el costo computacional de la tarjeta embebida se recomienda realizar programación estructurada en base a funciones, de esta manera solo se ejecutarán los procesos a su llamado y no constantemente.

Para la identificación de armas durante la noche se recomienda entrenar al clasificador con imágenes positivas propias del ambiente y usar una cámara nocturna de similares características que la usada en esta investigación.

Incorporar un sistema de alimentación ininterrumpida, así el sistema puede seguir funcionando en caso de fallas eléctricas.

Para la implementación inalámbrica se recomienda utilizar un internet mínimo de 5 Megabytes por segundo para asegurar un correcto desempeño del sistema.

GLOSARIO

| | |
|------------------------|---|
| Datasheet: | Hoja técnica u hoja de datos. |
| Arrays: | Conjunto de objetos de la misma clase. |
| Bit: | Unidad de medida de información. |
| Bucle: | Ejecuta repetidas veces un código. |
| Buffer: | Espacio en un disco reservado para almacenamiento. |
| Ethernet: | Estándar de redes de área local. |
| Fotosensible: | Sensibilidad a la acción de la luz. |
| Python: | Lenguaje de programación. |
| Monocromáticos: | Colores de un solo matiz o tono. |
| Pixel: | Unidad homogénea en color que forma parte de una imagen digital. |
| Umbral: | Cantidad mínima de señal que ha de estar presente para ser registrada por un sistema. |

BIBLIOGRAFÍA

Aguilar Cabrera, Edison Jacinto. *Identificación en tiempo real de personas en posesión de pequeñas armas dentro de un ambiente video-vigilado* [En línea] (Trabajo de titulación) (Grado) Escuela Superior Politécnica del Ejército. Sangolquí - Ecuador 2018. Pp 20-60. [Consulta: 9 febrero 2019]. Disponible en: <http://repositorio.espe.edu.ec/handle/21000/15377>

Stadista. *Number of robberies in the U.S. 2017, by weapon* Statistic. [en línea]. 2017 [Consulta: 9 febrero 2019]. Disponible en: <https://www.statista.com/statistics/251914/number-of-robberies-in-the-us-by-weapon/>.

Instituto Nacional de Estadísticas y Censos – INEC. *Delitos de mayor connotación psicosocial enero 2018.* [en línea]. 2018. [Consulta: 11 febrero 2019]. Disponible en: <https://www.ecuadorencifras.gob.ec/banco-de-informacion/>

R. K. Tiwari and G. K. Verma. *A Computer Vision based Framework for Visual Gun Detection Using Harris Interest Point Detector.* *Procedia Comput. Sci.* vol. 54, 2015, pp. 703–712.

ANT, e. *Transporte seguro (ant).* [blog]. [Consulta: 11 febrero 2019]. Disponible en: <http://www.ecu911.gob.ec/transporte-seguro/>

Cámara Nacional de Comercio y Servicios. *Sistema Seguridad Electrónico* [en línea]. Segunda Edición. Uruguay, 2017. [Consulta: 12 febrero 2019]. Disponible en: <https://docplayer.es/75908807-Segunda-edicion-manual-de-recomendaciones-sistema-eguridad-electronico.html>

Hidalgo Gallo, Esteban Fernando. *Sistema CCTV (Circuito Cerrado de Televisión) entre edificios, para la seguridad y vigilancia en el aeropuerto internacional Cotopaxi.* [En línea]. (Trabajo de titulación). (Pregrado) Universidad Técnica de Ambato. Ambato - Ecuador. 2012. pp. 16 – 17. [Consulta: 12 febrero 2019]. Disponible en: http://repo.uta.edu.ec/bitstream/123456789/2338/1/Tesis_t691ec.pdf

González, C. & Zarama, D. *Aplicación de sistemas de visión por computador (vir/nir) para la inspección visual de uchuvas.* [En línea] (Tesis) (Pregrado) Pontificia Universidad Javeriana Bogotá D.C Colombia 2014. [Consulta: 12 febrero 2019]. Disponible en: <https://repository.javeriana.edu.co/bitstream/handle/10554/16472/GonzalezMeybergCamiloEduardo2014.pdf;sequence=1>

Paguay, D. & Valarezo, L. *Diseño e implementación de un prototipo clasificador de huevo de gallina basado en las imperfecciones de la cáscara aplicando visión artificial.* (Tesis) (Pregrado) Escuela Superior Politécnica de Chimborazo. Riobamba, Ecuador 2018

Carrilo, R. *todo - fotografía.* [En línea] [Consulta: 15 febrero 2019]. Disponible en: <http://todo-fotografia.com/tecnica/componentes-de-una-camara-fotografica/>

Landoni, B. *Taller de Fotografía I.* [En línea] [Consulta: 15 febrero 2019]. Disponible en: http://fido.palermo.edu/servicios_dyc/blog/docentes/trabajos/42863_165800.pdf

Villamonte, J. *Componentes de la cámara fotográfica digital.* [En línea] [Consulta: 25 febrero 2019]. Disponible en: <https://es.slideshare.net/Julianalsola/componentes-de-la-camara-fotograficadigital>

García, E. *Qué cámara digital escoger,* 2015. [En línea] [Consulta: 25 febrero 2019]. Disponible en: http://www.asifunciona.com/practico/pr_camara_digital/camdig_4.htm

COGNEX. *Introducción a la Visión Artificial,* 2018 [En línea] [Consulta: 25 febrero 2019]. Disponible en http://www.ikusmen.com/documentos/descargas/3cbb38_Introduction%20to%20Machine%20Vision.pdf

BCNVISIÓN. *Sistemas de iluminación para aplicaciones de visión artificial,* 2017. [En línea]. [Consulta: 26 febrero 2019]. Disponible en: <http://www.bcnavision.es/blog-vision-artificial/iluminacion-vision-artificial2/>

Rosales, C. 2017. *Prototipo de detección de expresiones corporales mediante visión artificial para mejorar la comunicación con niños que tienen parálisis cerebral infantil.* [En línea] (Tesis) (Pregrado). Universidad Nacional de Loja. Loja, Ecuador 2017. [Consulta: 27 febrero 2019]. Disponible en: <http://dspace.unl.edu.ec/jspui/bitstream/123456789/18591/1/Rosales%20Romero%20Cristhian%20Eduardo.pdf>

Álvarez, M. *Análisis, Diseño e implementación de un sistema de control de ingreso de vehículos basado en visión artificial y reconocimiento de placas en el parqueadero de la universidad Politécnica Salesiana.* [En línea] (Tesis) (Pregrado) Universidad Politécnica Salesiana, Cuenca, Ecuador. 2014. [Consulta: 27 febrero 2019]. Disponible en: <https://dspace.ups.edu.ec/bitstream/123456789/7060/1/UPS-CT003790.pdf>.

Paredes, J. & Guerrero, L. *Estudio comparativo entre algoritmos de reconocimiento de borde para identificación de placas de autos.* [En línea] (Tesis) (Pregrado). Escuela Superior Politécnica de Chimborazo. Riobamba, Ecuador. 2012. [Consulta: 27 febrero 2019]. Disponible en: <http://dspace.espoch.edu.ec/handle/123456789/1944>

Juan, M. & Gustavo, R. *Imagen Digital.* [En línea] [Consulta: 01 marzo 2019]. Disponible en: https://ocw.unican.es/pluginfile.php/171/course/section/75/Bloque_1._Tema_1._

Valdivia, C. *Diseno de un Sistema de Visión Artificial para l clasificación de chirimoyas basado en medidas.* [En línea] (Tesis) (Pregrado) Pontificia Universidad Católica Del Perú Lima, Perú 2017 [Consulta: 01 marzo 2019]. Disponible en: <http://repositorio.concytec.gob.pe/bitstream/CONCYTEC/573/1/Tesis%20Valdivia%20Arias%20C%20C3%A9sar%20Javier.pdf>

Fu, k., Gonzales, C. & Lee, G. *ROBÓTICA: Control, detección, visión e inteligencia.* 1994. pp.306 – pp.310. Primera ed. México: s.n.

Jiménez, M. *Desarrollo de un sistema de visión artificial para la detección de aglomeración de personas en un semáforo.* [En línea] (Tesis) (Pregrado) Universidad Nacional de Loja. Loja, Ecuador. 2015 [Consulta: 03 marzo 2019]. Disponible en: <http://dspace.unl.edu.ec/jspui/bitstream/123456789/11225/1/Jim%20C3%A9nez%20Ochoa%20Magaly%20Gabriela.pdf>

Sobrado, E. *Sistema de Visión Artificial para el Reconocimiento y Manipulación de Objetos Utilizando Un Brazo Robot.* [En línea] (Tesis) (Maestria)Pontificia Universidad Católica Del Perú Lima, Perú 2003 [Consulta: 05 marzo 2019]. Disponible en: http://tesis.pucp.edu.pe/repositorio/bitstream/handle/123456789/68/SOBRAO_EDDIE_VISION_ARTIFICIAL_BRAZO_ROBOT.pdf

Torres, j. *Deep learning, introduccion practica con keras.* Barcelona-España: Watch This Space, 2018, pp. 50-60.

RASPBERRYPI.ORG, 2016. FrontPage - Raspbian. [en línea]. [Consulta: 10 marzo 2019]. Disponible en: <https://www.raspbian.org/>.

About Ubuntu Mate. [en línea], 2015. [Consulta: 10 marzo 2019]. Disponible en: <https://ubuntumate.org/about/>.

UBUNTU, 2015. *Ubuntu Core documentation*. [en línea]. [Consulta: 13 marzo 2019]. Disponible en: <https://www.ubuntu.com/core>.

MICROSOFT, 2013. *Download Windows 10 IoT Core*. [en línea]. [Consulta: 13 marzo 2019]. Disponible en: <https://www.microsoft.com/en-us/software-download/windows10IoTCore>.

Díez, Sergio. *Implementación del envío de sms mediante plataforma software defined radio*. (Tesis) (Pregrado) Universidad Carlos III de Madrid. Leganés, España 2016. [en línea]. [Consulta: 13 marzo 2019]. Disponible en: https://e-archivo.uc3m.es/bitstream/handle/10016/26824/TFG_Sergio_Diez_Sanz.pdf

Oyala, Víctor. *Sistemas de información geográfica*. [en línea], 2011. [Consulta: 13 marzo 2019]. Disponible en: <https://volaya.github.io/libro-sig/index.html>

Diego Marcos Jorquera1, Francisco Maciá Pérez. *Sistema de Regeneración de Nodos de Red*. [en línea]. [Consulta: 13 marzo 2019]. Disponible en: https://www.researchgate.net/publication/228704370_Sistema_de_Regeneracion_de_Nodos_de_Red.

REINIUS, S. *Object recognition using the OpenCV Haar cascade-classifier on the iOS platform*. [en línea] Haar classification. 2013 pp. 12 – 13 [Consulta: 06 de febrero de 2019]. Disponible en: <http://www.diva-portal.org/smash/get/diva2:601707/fulltext01.pdf>

KANNAN, R; et al. Performance Evaluation of SIFT & FLANN and HAAR Cascade Image Processing Algorithms for Object Identification in Robotic Applications [en línea] USA: Volumen 118 No. 18 2018. pp. 2606 - 2606 [Consulta: 13 de marzo de 2019]. Disponible en: <https://acadpubl.eu/jsi/2018-118-18/articles/18c/37.pdf>

GRUPOPREVENT. *¿Cuál es la mejor ubicación de las cámaras de vigilancia?, 2014*. [en línea] [Consulta: 13 marzo 2019]. Disponible en: <https://www.prevent.es/cual-es-la-mejor-ubicacion-de-las-camaras-de-vigilancia>

MALDONADO, L. Diseño experimental de un dispositivo que por medio de una fuente de radiación infrarroja externa permita visualizar una imagen térmica de un objetivo [en línea]

(Trabajo de titulación). (Postgrado) Escuela Superior Politécnica de Chimborazo, Facultad de Informática y Electrónica, Escuela de Ingeniería Electrónica en Control y Redes Industriales. Ecuador, Chimborazo. 2018 pp. 16 – 23 [Consulta: 13 de marzo de 2019].

ANEXOS

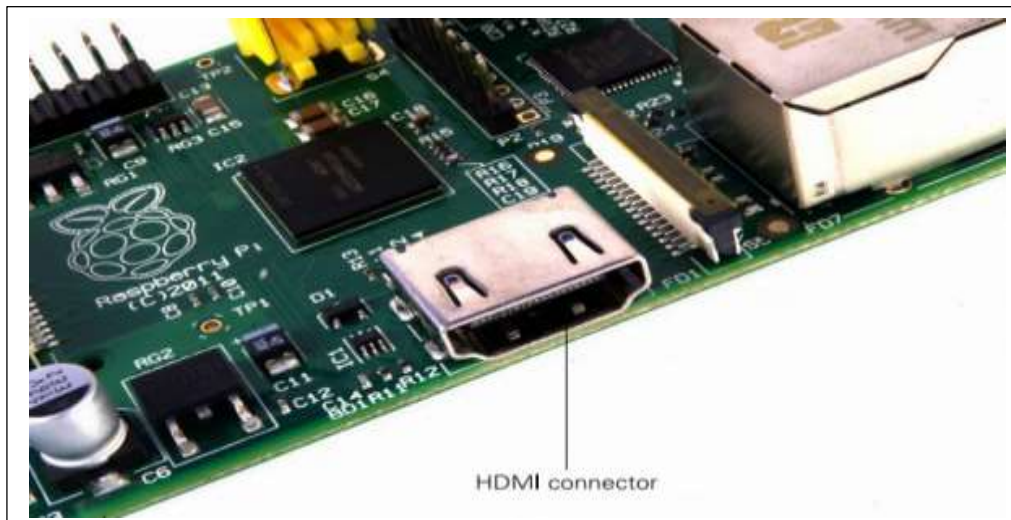
ANEXO A: MANUAL DE USUARIO

El dispositivo de detección tiene como objetivo principal identificar armas dentro de un ambiente video vigilado, generando automáticamente alarmas al momento de detectar una. El prototipo se caracteriza porque es accesible, dinámico, interactivo y sencillo de utilizar, ya que sus interfaces son dinámicas y la aplicación se encuentra desarrollada para ser ejecutada modo escritorio.

El propósito de este Manual es facilitar al usuario la operación del dispositivo y configuración del mismo.

Conexión de la pantalla

Antes de empezar a usar el dispositivo, necesita conectar una pantalla para configurar las condiciones del ambiente. El dispositivo es compatible con tres salidas de video diferentes: video compuesto, video HDMI y video DSI. El video compuesto y video HDMI son fácilmente accesibles para el usuario final, como se describe en esta sección; mientras que el video DSI requiere de un hardware especializado.



Conexión de teclado y el ratón

Ahora que ya tiene los dispositivos de salida del dispositivo bien clasificados, es el momento de pensar en la entrada. Como mínimo, va a necesitar un teclado, y para la mayoría de los usuarios,

un ratón o trackball también será necesario, el dispositivo espera que sus periféricos se conecten a través del puerto USB.



Conexión del Almacenamiento Externo

Como se habrá dado cuenta, el dispositivo no tiene un disco duro tradicional. En su lugar utiliza una tarjeta de memoria SD (Secure Digital), un sistema de almacenamiento de estado sólido que se usa habitualmente en las cámaras digitales. Casi cualquier tarjeta SD funcionará con el dispositivo, pero debido a que alberga todo el sistema operativo, es necesario que la tarjeta sea de al menos 4 GB de capacidad para almacenar todos los archivos requeridos.

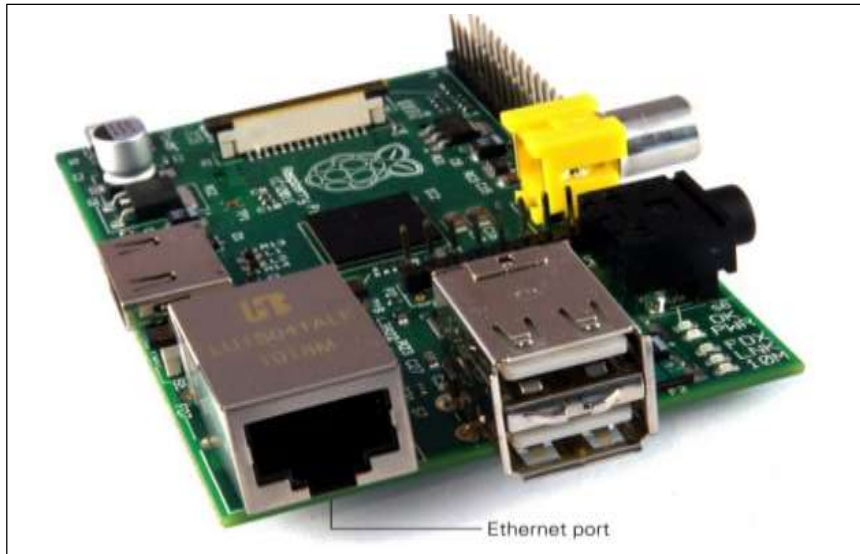
Algunas tarjetas SD funcionan mejor que otras y algunos modelos se niegan en absoluto a trabajar con el dispositivo. Para ver una lista actualizada de los modelos de tarjetas SD que funcionan con el dispositivo, visite la página eLinux Wiki: http://www.elinux.org/RPi_SD_cards



Conexión de la Red

Red cableada

Para conectar su dispositivo en Red, necesitará conectar un cable Ethernet RJ45 entre el dispositivo y el concentrador (switch), router o hub. Si no cuenta con un router o hub puede conseguir comunicar su PC o laptop con el dispositivo conectando ambos equipos de manera directa con un cable Patch Cord.



Red inalámbrica

El dispositivo cuenta con la característica integrada para gestionar redes inalámbricas, es posible añadir soporte Wi-Fi al dispositivo utilizando la tarjeta para red inalámbrica incorporada.

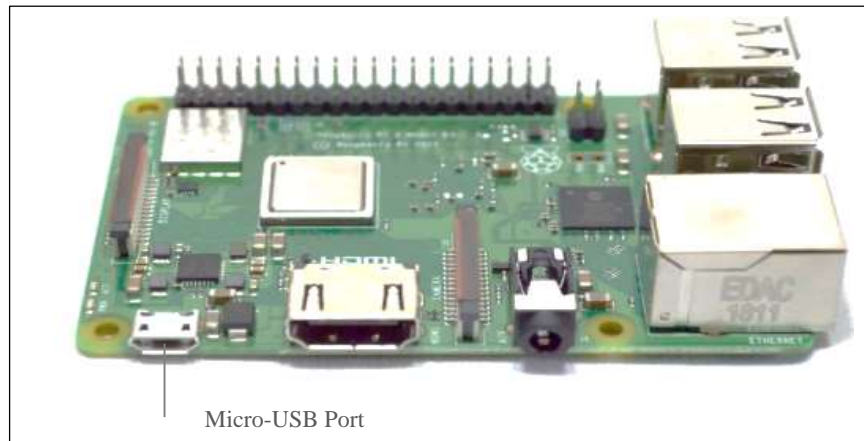
Conexión de la Fuente de Alimentación

El dispositivo se alimenta por el pequeño conector micro-USB ubicado en la parte inferior izquierda de la placa de circuito. Este conector es el mismo que se encuentra presente en la mayoría de los smartphones y algunas tablets.

Muchos cargadores diseñados para smartphones funcionarán con el dispositivo, pero no todos. El dispositivo consume más energía que la mayoría de los dispositivos micro-USB y requiere de al menos 700mA para funcionar. Algunos cargadores sólo pueden suministrar hasta 500mA, ocasionando problemas intermitentes en el funcionamiento del dispositivo.

Es posible conectar el dispositivo al puerto USB de una PC o laptop, pero no es recomendable. Al igual que con los cargadores de menor potencia, los puertos USB de una computadora no

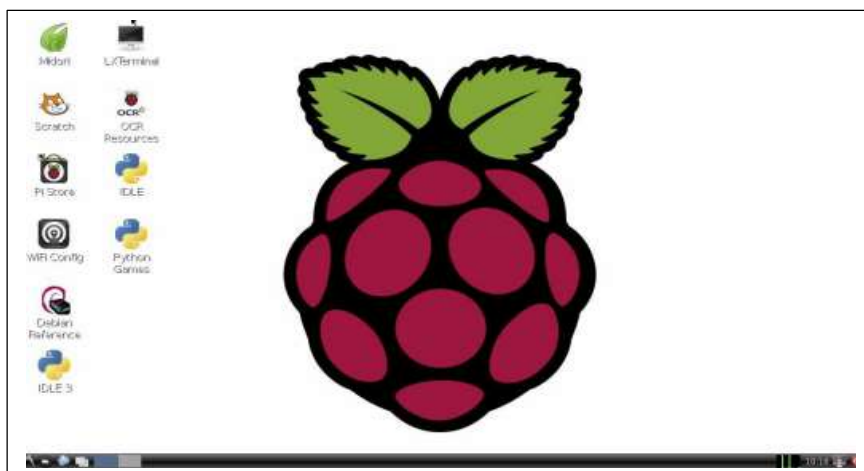
pueden suministrar la energía necesaria para que el dispositivo funcione adecuadamente. Sólo conecte la fuente de alimentación micro-USB cuando se encuentre listo para comenzar a utilizar su dispositivo. Al no contar con un botón de encendido, el dispositivo comenzará a trabajar en el mismo instante en que la fuente de alimentación sea conectada.



Instalación del Sistema Operativo

Si ya adquirió el dispositivo junto con una tarjeta SD con un sistema operativo precargado, sólo tiene que insertar la tarjeta en la ranura para tarjetas SD en la parte inferior del dispositivo. Si adquirió únicamente el dispositivo, necesitará instalar un sistema operativo a la tarjeta SD antes de que pueda utilizar su dispositivo.

Si es la primera vez que ejecuta RASPBIAN en aquella tarjeta SD, habrá un retardo en el arranque mientras la partición de la tarjeta se redimensiona para hacer espacio a su sistema operativo seleccionado; no desconecte la alimentación del dispositivo mientras se encuentra en este proceso, ya que corre el riesgo de dañar su tarjeta SD.



Actualización del software

Además de instalar y desinstalar paquetes, apt puede utilizarse para mantenerlos actualizados. La actualización de un paquete a través de apt le garantiza que ha recibido las últimas actualizaciones, correcciones de errores y parches de seguridad.

Antes de intentar actualizar un paquete, asegúrese que la caché de apt sea la más reciente llevando a cabo una actualización:

```
sudo apt-get update
```

Al actualizar el software, tiene dos opciones: puede actualizar de una vez todo en el sistema o actualizar los programas individualmente. Si sólo desea mantener su distribución actualizada, la forma de conseguirlo es escribiendo lo siguiente:

```
sudo apt-get upgrade
```

Instalación de OpenCV

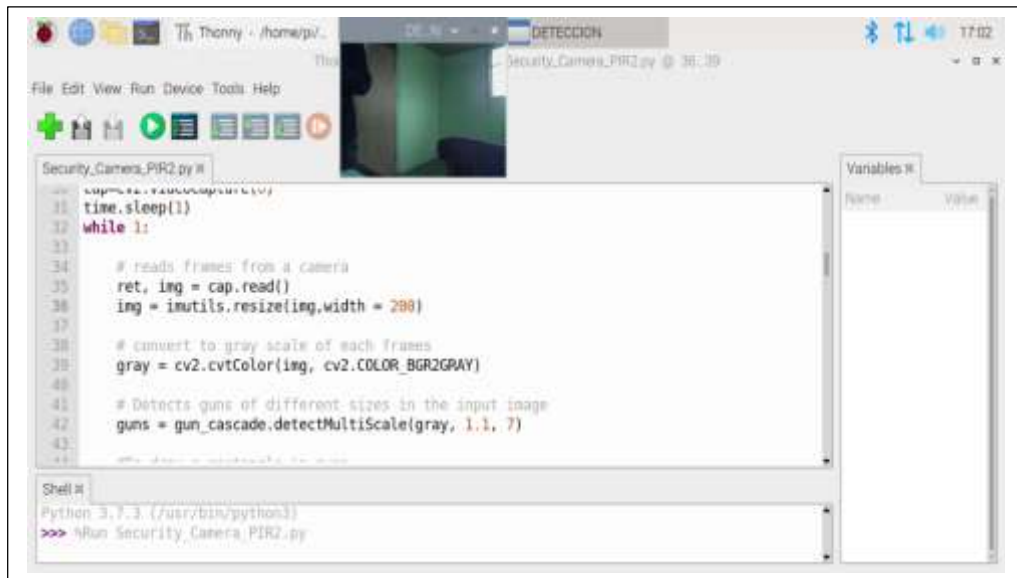
La instalación del software es un privilegio permitido sólo para el usuario root, ya que afecta a todos los usuarios del dispositivo. Por consiguiente, los comandos necesitan ser precedidos con sudo para indicarle al sistema operativo que debe de ejecutarse como el usuario root.

```
sudo apt-get install opencv
```

```
LXTerminal
File Edit Tabs Help
libiso9660-7 libjack-jackd2-0 libjbig2dec0 libkabc4 libkcal4 libkdcraw8
libkde3support4 libkdecore5 libkdesu5 libkdeui5 libkdnssd4 libkfile4
libkhtml5 libkimap4 libkio5 libkjsapi4 libkjsembed4 libkldap4
libkmediaplayer4 libkmime4 libknewstuff2-4 libknewstuff3-4 libknotifyconfig4
libkntlm4 libkparts4 libkpimutils4 libkpty4 libkresources4 libkrosscore4
libkrossui4 libktexteditor4 libkutils4 libmailtransport4 libmicroblog4
libmodplug1 libmpcdec6 libmysqlclient16 libnepomuk4 libnepomukquery4a
liborc-0.4-0 libpaper-utils libpaper1 libphonon4 libplasma3 libpolkit-qt-1-0
libpoppler-qt4-3 libpostproc51 libpq5 libpqxx-3.0 libpulse-mainloop-glib0
libqca2 libqimageblitz4 libqt4-dbus libqt4-designer libqt4-network
libqt4-opengl libqt4-qt3support libqt4-script libqt4-sql libqt4-sql-mysql
libqt4-svg libqt4-webkit libqt4-xml libqt4-xmlpatterns libqtcore4 libqtgui4
libreadline5 libruby1.8 libschrödinger-1.0-0 libsolid4 lib soprano4
libspeex1 libssh-4 libstreamanalyzer0 libstreams0 libtheora0
libthreadweaver4 libutempter0 libv4l-0 libvcdinfo0 libwavpack1 libxbase2.0-0
libxcb-shape0 libxcb-shm0 libxcb-xv0 libxine1 libxine1-bin libxine1-console
libxine1-ffmpeg libxine1-misc-plugins libxine1-plugins libxine1-x
libxml2-utils libxvnc1 mysql-common mysql-server-core-5.1 oxygen-icon-theme
phonon phonon-backend-xine plasma-scriptengine-javascript qt4-qtconfig ruby
ruby1.8 shared-desktop-ontologies soprano-daemon
0 upgraded, 161 newly installed, 0 to remove and 39 not upgraded.
Need to get 138 MB of archives.
After this operation, 362 MB of additional disk space will be used.
Do you want to continue [Y/n]? █
```

Ejecutar el código de detección

Una vez que se ha instalado OpenCV se procede a ejecutar el código de detección para ello solo basta buscar el archivo y abrirlo con Tony Python, una vez que se abra el código en Tony Python se puede pulsar sobre el botón de Run y el algoritmo de detección empezará a ejecutarse.



ANEXO B: CÓDIGO FUENTE

```
from time import sleep
import smtplib
import time
from datetime import datetime
from email.mime.image import MIMEImage
from email.mime.multipart import MIMEMultipart
import cv2
import numpy as np
import time,sys
import matplotlib.pyplot as plt
import imutils

gmail_user = 'nodosseguridad2@gmail.com'
gmail_password = 'nodos2020'
gun_cascade = cv2.CascadeClassifier('pistola.xml')
```

```

knife_cascade = cv2.CascadeClassifier('cuchillo.xml')
rifle_cascade = cv2.CascadeClassifier('rifle.xml')
metra_cascade = cv2.CascadeClassifier('metralleta.xml')
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
toaddr = 'nodoseguridad@gmail.com'
me = 'nodoseguridad2@gmail.com'
Subject='alerta'
cap=cv2.VideoCapture(0)
time.sleep(1)
while 1:
    # reads frames from a camera
    ret, img = cap.read()
    # Convert the image to RGB colorspace
    image = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    # Make a copy of the original image to draw face detections on
    image_copy = np.copy(image)
    # Convert the image to gray
    gray_image = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)
    # Detect faces in the image using pre-trained face detector
    #img = imutils.resize(img,width = 900)
    # convert to gray scale of each frames
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
    # Detects guns of different sizes in the input image
    guns = gun_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5, minSize=(30,
30))
    #To draw a rectangle in guns
    for (ex,ey,ew,eh) in guns:
        cv2.rectangle(img,(ex,ey),(ex+ew,ey+eh),(255,255,0),2)
        font=cv2.FONT_HERSHEY_SIMPLEX
        cv2.putText(img, "ARMA DETECTADA",(250,20),
                    cv2.FONT_HERSHEY_SIMPLEX,1,(0,0,255),2)
        cv2.putText(img, "pistola",(ex,ey),
                    cv2.FONT_HERSHEY_SIMPLEX,1,(255,255,0),2)
    # Get the bounding box for each detected face
    faces = face_cascade.detectMultiScale(gray, 1.08, 4)
    print('Number of faces detected:', len(faces))
    if (len(faces)>0):

```

```

face_crop = []
for (rx,ry,rw,rh) in faces:
    cv2.rectangle(img,(rx,ry),(rx+rw,ry+rh),(255,255,0),2)
    face_crop.append(image_copy[ry:ry+rh, rx:rx+rw])
for face in face_crop:
    face = imutils.resize(face,width = 300)
    print("Motion...")
    p=cv2.imwrite('movement.jpg',face)
    p1=cv2.imwrite('movement1.jpg',image_copy)

    subject='Security alert!!'
    msg = MIMEMultipart()
    msg['Subject'] = subject
    msg['From'] = me
    msg['To'] = toaddr
    fp= open('movement.jpg','rb')
    img2= MIMEImage(fp.read())
    fp.close()
    msg.attach(img2)
    fp1= open('movement1.jpg','rb')
    img3= MIMEImage(fp1.read())
    fp1.close()
    msg.attach(img3)
    server = smtplib.SMTP_SSL('smtp.gmail.com',465)
    server.login(gmail_user, gmail_password)
    server.send_message(msg)
    server.quit()
    print("gmail enviado...")
else:
    print("Motion...")
    p=cv2.imwrite('movement.jpg',image_copy)
    subject='Security alert!!'
    msg = MIMEMultipart()
    msg['Subject'] = subject
    msg['From'] = me
    msg['To'] = toaddr
    fp= open('movement.jpg','rb')

```

```

img2= MIMEImage(fp.read())
fp.close()
msg.attach(img2)
server = smtplib.SMTP_SSL('smtp.gmail.com',465)
server.login(gmail_user, gmail_password)
server.send_message(msg)
server.quit()
print("gmail enviado...")

# Detects knives of different sizes in the input image
knives = knife_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5,
minSize=(50, 50))
for (x,y,w,h) in knives:
# To draw a rectangle in a knife
cv2.rectangle(img,(x,y),(x+w,y+h),(0,0,255),2)
font=cv2.FONT_HERSHEY_SIMPLEX
cv2.putText(img, "ARMA DETECTADA",(250,20),
cv2.FONT_HERSHEY_SIMPLEX,1,(0,0,255),2)
cv2.putText(img, "cuchillo",(x,y),
cv2.FONT_HERSHEY_SIMPLEX,1,(0,0,255),2)

# Get the bounding box for each detected face
faces = face_cascade.detectMultiScale(gray, 1.08, 4)
print('Number of faces detected:', len(faces))
if (len(faces)>0):
face_crop = []
for (rx,ry,rw,rh) in faces:
cv2.rectangle(img,(rx,ry),(rx+rw,ry+rh),(255,255,0),2)
face_crop.append(image_copy[ry:ry+rh, rx:rx+rw])
for face in face_crop:
face = imutils.resize(face,width = 300)
print("Motion...")
p=cv2.imwrite('movement.jpg',face)
p1=cv2.imwrite('movement1.jpg',image_copy)
subject='Security alert!!'
msg = MIMEMultipart()
msg['Subject'] = subject
msg['From'] = me
msg['To'] = toaddr

```



```

        fp= open('movement.jpg','rb')
        img2= MIMEImage(fp.read())
        fp.close()
        msg.attach(img2)
        fp1= open('movement1.jpg','rb')
        img3= MIMEImage(fp1.read())
        fp1.close()
        msg.attach(img3)
        server = smtplib.SMTP_SSL('smtp.gmail.com',465)
        server.login(gmail_user, gmail_password)
        server.send_message(msg)
        server.quit()
        print("gmail enviado...")
    else:
        print("Motion...")
        p=cv2.imwrite('movement.jpg',image_copy)
        subject='Security alert!!'
        msg = MIMEMultipart()
        msg['Subject'] = subject
        msg['From'] = me
        msg['To'] = toaddr
        fp= open('movement.jpg','rb')
        img2= MIMEImage(fp.read())
        fp.close()
        msg.attach(img2)
        server = smtplib.SMTP_SSL('smtp.gmail.com',465)
        server.login(gmail_user, gmail_password)
        server.send_message(msg)
        server.quit()
        print("gmail enviado...")
# Detects knives of different sizes in the input image
metras = metra_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=10,
minSize=(100, 100))
for (ox,oy,ow,oh) in metras:
# To draw a rectangle in a metras
cv2.rectangle(img,(ox,oy),(ox+ow,oy+oh),(0,0,255),2)
font=cv2.FONT_HERSHEY_SIMPLEX

```

```

cv2.putText(img, "ARMA DETECTADA",(250,20),
            cv2.FONT_HERSHEY_SIMPLEX,1,(0,0,255),2)
cv2.putText(img, "metralleta",(ox,oy),
            cv2.FONT_HERSHEY_SIMPLEX,1,(0,0,255),2)
# Get the bounding box for each detected face
faces = face_cascade.detectMultiScale(gray, 1.08, 4)
print('Number of faces detected:', len(faces))
if (len(faces)>0):
    face_crop = []
    for (rx,ry,rw,rh) in faces:
        cv2.rectangle(img,(rx,ry),(rx+rw,ry+rh),(255,255,0),2)
        face_crop.append(image_copy[ry:ry+rh, rx:rx+rw])
    for face in face_crop:
        face = imutils.resize(face,width = 300)
        print("Motion...")
        p=cv2.imwrite('movement.jpg',face)
        p1=cv2.imwrite('movement1.jpg',image_copy)
        subject='Security alert!!'
        msg = MIMEMultipart()
        msg['Subject'] = subject
        msg['From'] = me
        msg['To'] = toaddr
        fp= open('movement.jpg','rb')
        img2= MIMEImage(fp.read())
        fp.close()
        msg.attach(img2)
        fp1= open('movement1.jpg','rb')
        img3= MIMEImage(fp1.read())
        fp1.close()
        msg.attach(img3)
        server = smtplib.SMTP_SSL('smtp.gmail.com',465)
        server.login(gmail_user, gmail_password)
        server.send_message(msg)
        server.quit()
        print("gmail enviado...")
else:
    print("Motion...")

```

```

p=cv2.imwrite('movement.jpg',image_copy)
subject='Security alert!!'
msg = MIMEMultipart()
msg['Subject'] = subject
msg['From'] = me
msg['To'] = toaddr
fp= open('movement.jpg','rb')
img2= MIMEImage(fp.read())
fp.close()
msg.attach(img2)
server = smtplib.SMTP_SSL('smtp.gmail.com',465)
server.login(gmail_user, gmail_password)
server.send_message(msg)
server.quit()
print("gmail enviado...")

# Detects knives of different sizes in the input image
rifles = rifle_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=15,
minSize=(200, 400))
for (ax,ay,aw,ah) in rifles:
# To draw a rectangle in a rifles
cv2.rectangle(img,(ax,ay),(ax+aw,ay+ah),(0,0,255),2)
font=cv2.FONT_HERSHEY_SIMPLEX
cv2.putText(img, "ARMA DETECTADA",(250,20),
cv2.FONT_HERSHEY_SIMPLEX,1,(0,0,255),2)
cv2.putText(img, "rifle",(ax,ay),
cv2.FONT_HERSHEY_SIMPLEX,1,(0,0,255),2)

# Get the bounding box for each detected face
faces = face_cascade.detectMultiScale(gray, 1.08, 4)
print('Number of faces detected:', len(faces))
if (len(faces)>0):
face_crop = []
for (rx,ry,rw,rh) in faces:
cv2.rectangle(img,(rx,ry),(rx+rw,ry+rh),(255,255,0),2)
face_crop.append(image_copy[ry:ry+rh, rx:rx+rw])
for face in face_crop:
face = imutils.resize(face,width = 300)
print("Motion...")

```

```
p=cv2.imwrite('movement.jpg',face)
p1=cv2.imwrite('movement1.jpg',image_copy)
subject='Security alert!!'
msg = MIMEMultipart()
msg['Subject'] = subject
msg['From'] = me
msg['To'] = toaddr
fp= open('movement.jpg','rb')
img2= MIMEImage(fp.read())
fp.close()
msg.attach(img2)
fp1= open('movement1.jpg','rb')
img3= MIMEImage(fp1.read())
fp1.close()
msg.attach(img3)
server = smtplib.SMTP_SSL('smtp.gmail.com',465)
server.login(gmail_user, gmail_password)
server.send_message(msg)
server.quit()
print("gmail enviado...")
```

else:

```
print("Motion...")
p=cv2.imwrite('movement.jpg',image_copy)
subject='Security alert!!'
msg = MIMEMultipart()
msg['Subject'] = subject
msg['From'] = me
msg['To'] = toaddr
fp= open('movement.jpg','rb')
img2= MIMEImage(fp.read())
fp.close()
msg.attach(img2)

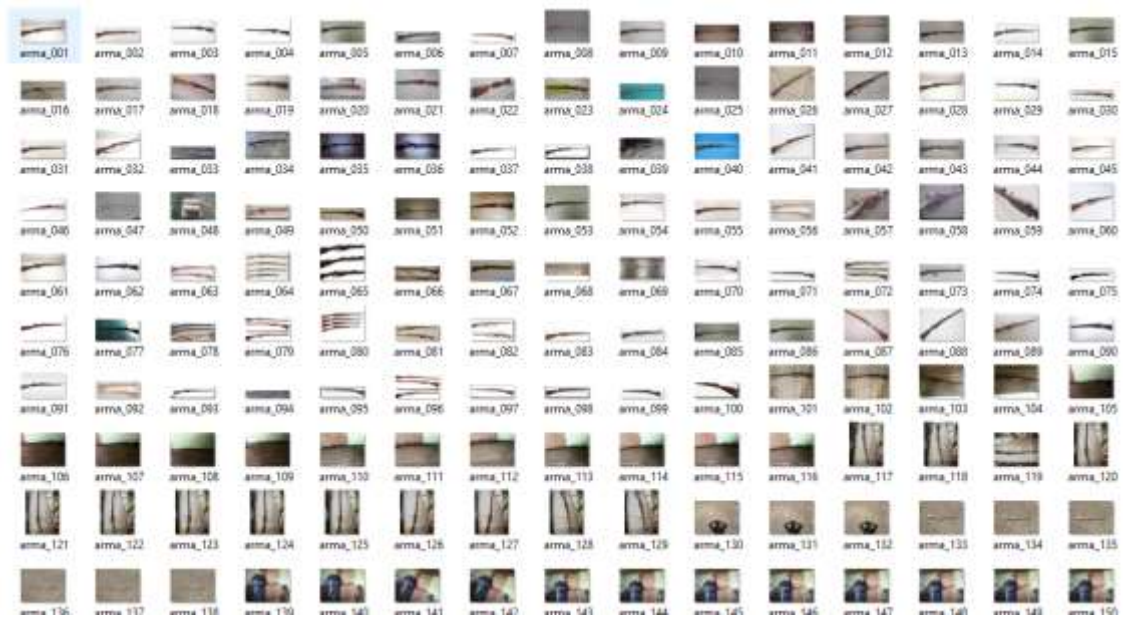
server = smtplib.SMTP_SSL('smtp.gmail.com',465)
server.login(gmail_user, gmail_password)
server.send_message(msg)
server.quit()
```

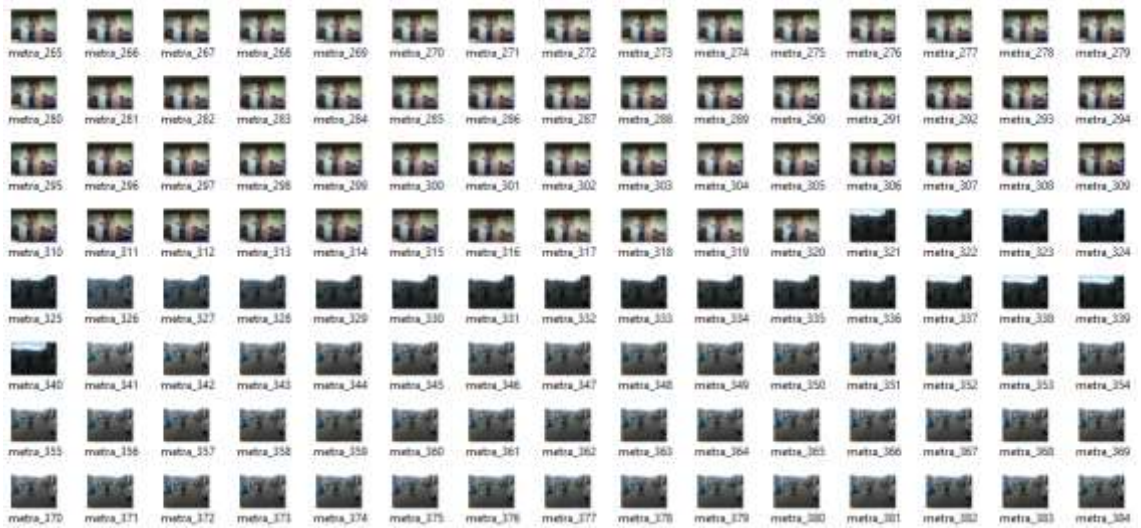
```

    print("gmail enviado...")
# Display an image in a window
cv2.imshow('DETECCION',img)
# Wait for Esc key to stop
k = cv2.waitKey(30) & 0xff
if k == 27:
    break
# Close the window
cap.release()
# De-allocate any associated memory usage
cv2.destroyAllWindows()

```

ANEXO C: Toma de muestras





ANEXO D: Pruebas de detección



mirandafernando177@gmail.com

para mí ▾

2 archivos adjuntos





mirandafernando177@gmail.com

para mí ▾

2 archivos adjuntos



mirandafernando177@gmail.com

para mí ▾

2 archivos adjuntos



mirandafernando177@gmail.com

para mí ▾

2 archivos adjuntos





mirandafernando177@gmail.com

para mí ▾



mirandafernando177@gmail.com

para mí ▾



mirandafernando177@gmail.com

para mí ▾

2 archivos adjuntos





**ESCUELA SUPERIOR POLITÉCNICA DE
CHIMBORAZO**



**DIRECCIÓN DE BIBLIOTECAS Y RECURSOS
PARA EL APRENDIZAJE Y LA INVESTIGACIÓN**

UNIDAD DE PROCESOS TÉCNICOS
REVISIÓN DE NORMAS TÉCNICAS, RESUMEN Y BIBLIOGRAFÍA

Fecha de entrega: 22 /01 /2020

| |
|---|
| INFORMACIÓN DEL AUTOR/A (S) |
| Nombres – Apellidos: FERNANDO PATRICIO OÑATE MIRANDA |
| INFORMACIÓN INSTITUCIONAL |
| Facultad: INFORMÁTICA Y ELECTRÓNICA |
| Carrera: INGENIERÍA ELECTRÓNICA CONTROL Y REDES INDUSTRIALES |
| Título a optar: INGENIERO EN ELECTRÓNICA, CONTROL Y REDES INDUSTRIALES |
| f. Analista de Biblioteca responsable: |