



UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE QUITO
CARRERA DE ELECTRÓNICA Y AUTOMATIZACIÓN

**SISTEMA DE VIGILANCIA INTELIGENTE IMPLEMENTADO EN EL
LOCAL COMERCIAL EL CHAVAL**

Trabajo de titulación previo a la obtención del
Título de Ingeniero en Electrónica y Automatización

AUTOR: Sebastián Mauricio Lovato Granizo

TUTOR: William Manuel Montalvo López

Quito-Ecuador

2024

**CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL
TRABAJO DE TITULACIÓN**

Yo, Sebastián Mauricio Lovato Granizo con documento de identificación No. 0604354407 manifiesto que:

Soy el autor y responsable del presente trabajo; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Quito, 17 de septiembre del año 2024

Atentamente,



Sebastián Mauricio Lovato Granizo

0604354407

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL
TRABAJO DE TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA
SALESIANA**

Yo, Sebastián Mauricio Lovato Granizo con documento de identificación No. 0604354407, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del Proyecto Técnico: “Sistema de vigilancia inteligente implementado en local comercial El Chaval”, el cual ha sido desarrollado para optar por el título de: Ingeniero en Electrónica y Automatización, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribo este documento en el momento que hago la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana

Quito, 17 de septiembre del año 2024

Atentamente,



Sebastián Mauricio Lovato Granizo

0604354407

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, William Manuel Montalvo López con documento de identificación N°1712789989, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: SISTEMA DE VIGILANCIA INTELIGENTE IMPLEMENTADO EN LOCAL COMERCIAL EL CHAVAL, realizado por Sebastián Mauricio Lovato Granizo con documento de identificación N° 0604354407, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Quito, 17 de septiembre del año 2024

Atentamente,

A handwritten signature in blue ink, appearing to be 'W. Montalvo', is written over a faint, light blue circular stamp or watermark.

Ing. William Manuel Montalvo López, MSc

1712789989

DEDICATORIA

Dedico la culminación de mi trabajo de titulación a mis padres que siempre me brindaron su apoyo incondicional durante toda mi etapa universitaria, además de mi hermana que gracias a su ayuda pude conllevar las vicisitudes que tuve en esta etapa académica.

ÍNDICE

INTRODUCCIÓN.....	IV
CAPÍTULO 1	1
ANTECEDENTES	1
1.1 Problema	1
1.1.1 Antecedentes.....	1
1.1.2 Importancia y alcances	2
1.1.3 Delimitación	3
1.2 Objetivos del proyecto	3
1.2.1 Objetivo general	3
1.2.2 Objetivos específicos	3
CAPÍTULO 2	4
FUNDAMENTOS TEÓRICOS.....	4
2.1 Sistema de seguridad inteligente.....	4
2.2 Visión Artificial	4
2.3 YOLO V5.....	5
2.4 Principales librerías de YOLOV5	5
2.4.1 PyTorch	5
2.4.2 OpenCv.....	5
2.4.3 Numpy	6
2.5 Reconocimiento de voz.....	6
2.6 Cámara de vigilancia Logitech 920	7
2.7 Jetson Nano.....	7
2.8 Micrófono USB.....	8
2.9 Protocolo de comunicación Ethernet	8
2.10 Arduino Nano.....	8
2.11 SIM800L	9
CAPÍTULO 3	10
MARCO METODOLÓGICO.....	10
3.1 Propuesta del proyecto.....	10
3.2 Variables para tomar en cuenta.....	11
3.3 Partes del sistema distribuido.....	11

3.4	Lenguaje de programación.....	11
3.5	Implementación electrónica.....	12
3.5.1	Tarjeta de desarrollo	12
3.5.2	Algoritmo de entrenamiento de la red neuronal	13
3.5.3	Programa de reconocimiento de objetos y voz	15
3.6	Sistema de alarma	16
3.7	Estructura del equipo	18
CAPÍTULO 4		19
PRUEBAS Y RESULTADOS.....		19
4.1	Resultados del reconocimiento de objetos	19
4.2	Resultados para el reconocimiento de voz.....	24
RECOMENDACIONES		29
Referencias Bibliográficas.....		30
Anexos.....		32

ÍNDICE DE TABLAS

Tabla 3.1 Comparativa de variables	11
Tabla 3.2 Comparación entre las tarjetas de desarrollo.....	12
Tabla 4.1 Resultados de la detección de objetos	24
Tabla 4.2 Escala de precisión para el reconocimiento de voz	25
Tabla 4.3 Resultados de la distancia del reconocimiento de voz	25
Tabla 4.4 Escala de nivel de ruido.....	25
Tabla 4.5 Resultados del nivel de ruido en el reconocimiento de voz	26

ÍNDICE DE FIGURAS

Figura 2.1 Cámara Logitech	7
Figura 2.2 Jetson Nano	7
Figura 2.3 Micrófono USB	8
Figura 2.4 Arduino NANO	8
Figura 2.5 SIM800L	9
Figura 3.1 Esquema general del sistema electrónico distribuido	10
Figura 3.2 Diagrama de flujo el algoritmo del entrenamiento de imágenes.....	13
Figura 3.3 Etiquetado de imagen.....	14
Figura 3.4 Creación de clases	14
Figura 3.5 Diagrama de flujo del algoritmo para reconocimiento de voz y objetos	15
Figura 3.6 Diagrama de flujo del algoritmo para el sistema de alarma.....	16
Figura 3.7 Diseño PCB en Proteus	17
Figura 3.8 PCB con las pistas conductoras.....	17
Figura 3.9 Elementos soldados en la PCB.....	17
Figura 3.10 Estructura del sistema de alarma.....	18
Figura 4.1 Detección de mascarilla a 2 metros de distancia.....	19
Figura 4.2 Detección de mascarilla a 1.5 metros de distancia.....	20
Figura 4.3 Detección de mascarilla a 1 metros de distancia.....	20
Figura 4.4 Detección de gorra a 2 metros de distancia.....	21
Figura 4.5 Detección de gorra a 1.5 metros de distancia.....	21
Figura 4.6 Detección de gorra a 1 metro de distancia	22
Figura 4.7 Detección de gorra a 2 metros de distancia.....	22
Figura 4.8 Detección de pasamontañas a 1.5 metros de distancia.....	23
Figura 4.9 Detección de pasamontañas a 1 metros de distancia.....	23
Figura 4.10 Mensaje de alerta de la detección de objetos	26
Figura 4.11 Mensaje de alerta del reconocimiento de voz	26

RESUMEN

El desarrollo del proyecto técnico busca implementar un sistema de seguridad inteligente en un pequeño local comercial. El sistema desarrollado busca brindar seguridad a los empleados y clientes del local al utilizar una cámara de seguridad que detecte objetos que sirven para ocultar el rostro como gorras, mascarillas y pasamontañas, además de un micrófono que detecte palabras y frases amenazantes. Con la detección de los patrones establecidos, se enviará un mensaje de alerta vía SMS al encargado de la seguridad del local. Los resultados de las pruebas de campo demostraron que la eficiencia en la detección de objetos depende de factores como la luz, orientación del objeto y movimiento de la cámara, mientras que la detección de voz se ve afectada por el ruido y la distancia.

Se utilizó el sistema de detección de objetos código abierto YOLO V5 para el entrenamiento y procesamiento de imágenes, y para la detección de patrones de voz se empleó Google Speech Recognition. Para procesar los algoritmos, se utilizó una Jetson Nano, ya que cuenta con una GPU que facilita y mejora el rendimiento para la detección de objetos en tiempo real.

PALABRAS CLAVES: YOLO V5, Google Speech Recognition, sistema de seguridad, detección de objetos, Jetson Nano.

ABSTRACT

The development of the technical project seeks to implement an intelligent security system in a small commercial establishment. The developed system seeks to provide security to employees and customers by using a security camera that detects objects used to hide the face such as hats, masks, and balaclavas, as well as a microphone that detects threatening words and phrases. Upon detection of the established patterns, an alert message will be sent via SMS to the venue's security manager. Field test results showed that object detection efficiency depends on factors such as light, object orientation and camera movement, while voice detection is affected by noise and distance.

The open source object detection system YOLO V5 was used for training and image processing, and Google Speech Recognition was used for voice pattern detection. A Jetson Nano was used to process the algorithms, since it has a GPU that facilitates and improves performance for object detection in real time.

KEYWORDS: YOLO V5, Google Speech Recognition, security system, object detection, Jetson Nano.

INTRODUCCIÓN

Con el avance tecnológico se descubren cada vez más formas de innovar en los campos científicos, técnicos e industriales. En el campo de la seguridad, se están empezando a desarrollar sistemas que utilizan las herramientas tecnológicas más avanzadas, buscando mejorar las capacidades y la eficiencia de los sistemas de seguridad convencionales.

Actualmente el sector del comercio se ha visto gravemente afectado por robos y extorsiones, lo que provoca grandes pérdidas económicas. Esta situación puede llevar incluso al cierre total del negocio. Por esta razón es fundamental contar con un sistema de seguridad que brinde un espacio de confianza y comodidad para los empleados y clientes.

En este contexto se, busca desarrollar un sistema de vigilancia inteligente que pueda ser implementado en pequeños locales comerciales, ya que estos pequeños locales son los más vulnerables a ataques delictivos.

La inteligencia artificial no solo mejora la capacidad de respuesta ante posibles amenazas, sino que también reduce la carga de trabajo el personal de seguridad, permitiendo una supervisión continua y sin interrupciones. Los sistemas de seguridad que emplean inteligencia artificial pueden aprender y adaptarse a nuevos patrones de comportamiento, lo que los hace cada vez más precisos y confiables. Además, estos sistemas pueden integrarse con otras soluciones tecnológicas, para crear un entorno de seguridad más integral.

El sistema desarrollado realiza un monitoreo en tiempo real del negocio utilizando algoritmos de visión artificial y reconocimiento de voz para detectar comportamientos sospechosos que indiquen una amenaza a la seguridad del local. Al detectar cualquier amenaza se envía un mensaje de alerta vía SMS al encargado o propietario del establecimiento.

CAPÍTULO 1

ANTECEDENTES

1.1 Problema

1.1.1 Antecedentes

El aumento de la inseguridad es considerable en el país, las cifras demuestran el aumento de actividades delictivas como robos, homicidios, extorsiones entre otros. Esto incluye los robos a unidades económicas, los cuales se han ido incrementando cada año. Según la Fiscalía General del Estado en el año 2022 se registraron 5353 robos a unidades económicas, mientras que en el año 2023 la cifra aumentó a 5756 mostrando un incremento del 7% (Primicias, 2023).

Los locales comerciales además de sufrir robos que afectan directamente a su estabilidad financiera también son susceptibles a perder clientes y colaboradores por la pérdida de confianza ante la vulnerabilidad que existe en estos pequeños locales. Contar con un sistema de vigilancia eficiente puede prevenir en su mayoría cualquier infracción que afecte al personal o patrimonio del local comercial.

Contar con un sistema de vigilancia funcional y eficaz resulta un gasto muy elevado para ser cubierto por negocios pequeños por lo que optan por no implementar ningún sistema o implementar alguno que no brinde la seguridad necesaria.

De acuerdo con la problemática, existen varios estudios que han planteado varios sistemas de vigilancia inteligentes implementados en diferentes entornos, como (Palacios, 2023), que diseñó e implementó un sistema de seguridad inteligente para un centro estudiantil, el cual es capaz de detectar intrusos. El sistema, mediante inteligencia artificial detecta la intrusión y envía la imagen del intruso al encargado de seguridad del centro, o carga la imagen a la nube

para que sea visible en una aplicación de acceso limitado, y guarde la fecha y hora de la intrusión en una base de datos.

También (Quimiz & Trujillo, 2022) llevaron a cabo un prototipo de sistema de seguridad que utiliza las redes IOT y GSM, que funciona con energía solar dentro de un domicilio. El sistema cuenta con dos filtros de seguridad que se encuentran en la puerta principal de la casa el primero es una clave de acceso y el segundo es la identificación RFID, en caso de una entrada no autorizada dos sensores de movimiento envían un mensaje SMS y activan una sirena, con el mensaje SMS se pueden activar las luces de alarma del sistema además de tener acceso a una página web donde también es posible activar las luces de alarma.

Además (Gutierrez, 2021) en su investigación realizó la implementación de un sistema de seguridad inteligente para un edificio, analizó la factibilidad de implementar el sistema en el edificio e identificó las características técnicas de los equipos que fueron usados en la implementación del sistema. Para la implementación ubicó 4 cámaras de alta definición en puntos estratégicos con vista panorámica de todo el ambiente monitoreado, también colocó un DVR para tener un circuito cerrado de televisión, finalmente se instaló un monitor que muestra las imágenes obtenidas de las cámaras en tiempo real.

1.1.2 Importancia y alcances

El proyecto técnico pretende desarrollar un sistema de vigilancia inteligente, a través de visión artificial para el reconocimiento de prendas que oculten el rostro y algoritmos capaces de reconocer lenguaje natural para identificar palabras clave como amenazas o insultos, por medio de la detección de estos dos patrones se enviará un mensaje de alerta vía SMS al encargado de la seguridad del local comercial.

1.1.3 Delimitación

Para el desarrollo del sistema de vigilancia, se construye un conjunto de datos robusto que sirva para entrenar de manera eficaz a la red neuronal. Con los datos entrenados, se utilizan algoritmos que van a determinar si los objetos y patrones de voz son detectados. Dependiendo de si la detección es favorable o no, el sistema de alerta activará el mensaje de auxilio en los dispositivos correspondientes.

Por lo explicado, el sistema es ideal para brindar seguridad a pequeños locales comerciales de cualquier tipo. Lo que se debe tener en cuenta es que, al utilizar visión artificial, es sensible a los cambios de luz, y al reconocimiento de voz le afectan las alteraciones como el ruido. Por lo que es muy importante realizar una correcta calibración del sistema para su ideal funcionamiento.

1.2 Objetivos del proyecto

1.2.1 Objetivo general

- Implementar un sistema de vigilancia inteligente mediante el uso de visión artificial y reconocimiento de lenguaje natural en el local comercial El Chaval.

1.2.2 Objetivos específicos

- Sintetizar información relevante de los sistemas inteligentes de alarma empleando textos científicos para la selección de desarrollo de la metodología del proyecto.
- Modificar algoritmos de visión artificial para la detección de patrones de objetos mediante el uso de herramientas de inteligencia artificial de código abierto.
- Modificar algoritmos de procesamiento de lenguaje natural para la detección de palabras mediante el uso de herramientas de inteligencia artificial de código abierto.
- Verificar el sistema de alarma inteligente para su validación mediante pruebas de campo en la Pescadería El Chaval.

CAPÍTULO 2

FUNDAMENTOS TEÓRICOS

2.1 Sistema de seguridad inteligente

Los sistemas de seguridad cumplen con las funciones de detectar y alertar el comportamiento extraño de una persona que perturbe el desarrollo normal de actividades en domicilios o negocios. Con el avance tecnológico los sistemas de seguridad inteligentes usan los recursos técnicos más novedosos que amplían las funciones y aplicaciones de los sistemas de seguridad convencionales.

Una gran ventaja con la que cuentan los sistemas de seguridad inteligentes es que se pueden conectar con otros dispositivos que se encuentren en negocios o domicilios, lo que permite monitorear y controlar el sistema desde un teléfono inteligente o utilizando comandos de voz (Cohen, 2023).

Gracias a la capacidad que poseen los sistemas de seguridad inteligentes para adaptar otros dispositivos a su entorno, son capaces de cumplir tareas extra, como un control de acceso remoto, la detección de incendios, y la recepción de notificaciones en tiempo real (Cohen, 2023).

Al utilizar tecnologías emergentes, los sistemas inteligentes no son completamente efectivos en comparación a los sistemas convencionales, que se han ido perfeccionando a lo largo del tiempo hasta asegurar una eficiencia total. Por este motivo, es que se necesita seguir explorando e implementando nuevas ideas que ayuden a mejorar el rendimiento de los sistemas de seguridad inteligentes.

2.2 Visión Artificial

Es una disciplina de la inteligencia artificial que permite extraer información de imágenes. Esto amplía las tareas que puede realizar un computador, que antes se limitaban a guardar, editar o reproducir imágenes. Con la información obtenida y mediante su procesamiento, se pueden desarrollar una gran cantidad de aplicaciones en distintos campos (Mínguez, 2021).

2.3 YOLO V5

Es una red neuronal que es capaz de predecir los cuadros delimitadores y la probabilidad de clase a partir de una imagen. El algoritmo de detección es capaz de procesar más de 60 fotogramas por segundo, lo que los convierte en una opción excelente para la detección de objetos en video (V-alje, 2023).

2.4 Principales librerías de YOLOV5

2.4.1 PyTorch

PyTorch es una herramienta de aprendizaje profundo de código abierto que se usa para crear redes neuronales, fusionando la biblioteca Torch de aprendizaje automático con una interfaz de alto nivel basada en Python. PyTorch es compatible con una amplia variedad de estructuras de redes neuronales, desde simples modelos lineales hasta complejas redes convolucionales y modelos generativos transformadores utilizados en áreas como visión artificial y procesamiento del lenguaje natural (NLP) (IBM, 2022).

Al estar basado en Python y ofrecer extensas bibliotecas de modelos preconfigurados (incluso previamente entrenados), PyTorch permite a los científicos de datos crear y ejecutar sofisticadas redes neuronales profundas, reduciendo notablemente el tiempo y esfuerzo necesarios para el desarrollo del código y la estructura matemática (IBM, 2022).

2.4.2 OpenCv

OpenCV (Open Source Computer Vision Library) es una biblioteca de código abierto diseñada para facilitar la implementación de aplicaciones de visión por computadora y procesamiento de imágenes. Desarrollada inicialmente por Intel, OpenCV es ahora mantenida por una comunidad activa y cuenta con el respaldo de múltiples empresas tecnológicas. La biblioteca está escrita en C++ y ofrece enlaces para Python, Java y MATLAB, permitiendo su uso en una amplia variedad de aplicaciones y entornos. Con más de 2,500 algoritmos optimizados, OpenCV es una herramienta robusta para tareas de visión por computadora (Imagina, 2024).

La principal utilidad de OpenCV radica en su capacidad para procesar y analizar imágenes y videos. Esta biblioteca permite realizar operaciones básicas como la lectura, escritura y manipulación de imágenes, así como tareas más complejas como la detección de bordes, el reconocimiento de objetos, el seguimiento de movimiento y la reconstrucción 3D. Además,

OpenCV incluye funcionalidades avanzadas para el análisis de imágenes en tiempo real, lo que es crucial para aplicaciones que requieren procesamiento rápido y eficiente, como la realidad aumentada, la vigilancia de seguridad y los sistemas de conducción autónoma (Imagina, 2024).

2.4.3 Numpy

NumPy, que significa "Numerical Python", es una biblioteca fundamental para la computación científica en Python. Proporciona soporte para grandes matrices y matrices multidimensionales junto con una amplia colección de funciones matemáticas de alto nivel para operar con estos arreglos de manera eficiente. NumPy es esencial para cualquier trabajo que involucre álgebra lineal, transformaciones de Fourier y otras operaciones matemáticas avanzadas, sirviendo como la base sobre la que se construyen muchas otras bibliotecas de Python orientadas a la ciencia de datos y el aprendizaje automático (Numpy, 2020).

La principal utilidad de NumPy radica en su capacidad para realizar operaciones matemáticas rápidas y eficientes en grandes conjuntos de datos. Sus objetos de matriz, conocidos como ndarray, son mucho más eficientes que las listas de Python en términos de consumo de memoria y velocidad de ejecución. Además, NumPy permite realizar operaciones vectorizadas, que son operaciones aritméticas aplicadas a arrays enteros sin necesidad de escribir bucles explícitos en Python, lo que mejora significativamente el rendimiento y facilita la escritura de código más limpio y conciso (Numpy, 2020).

2.5 Reconocimiento de voz

Es una tecnología en donde intervienen varios procesos para la interpretación y comprensión del habla humana por parte de las computadoras, se basa en dos procesos fundamentales, el primero es el ASR que transforma los comandos de voz a texto para brindar una mejor interpretación a la computadora, el segundo se trata del NLP que utiliza el machine learning que ayuda a una mejor comprensión del lenguaje humano (Amazon, 2023).

2.6 Cámara de vigilancia Logitech 920

Esta es una cámara que ofrece gran nitidez y detalle en la calidad de imagen, tiene una resolución de 1080p a 30 fps, su campo visual es de 78° y posee corrección de iluminación automática HD, además cuenta con la característica de ajuste automático para tener una definición consistente y video fluido (Logitech, 2024). En la Figura 2.1 se muestra la Cámara Logitech.



Figura 2.1 Cámara Logitech

2.7 Jetson Nano

Es una pequeña computadora que tiene el sistema operativo Ubuntu que cuenta con librerías que permiten desarrollar proyectos de visión artificial y procesamiento de voz. El módulo tiene una memoria RAM de 4Gb, un CPU ARM A57 y una GPU Maxwell de 128 núcleos lo que mejora considerablemente el rendimiento de la detección de objetos en tiempo real (NVIDIA, 2024). En la Figura 2.2 se muestra la Jetson Nano.

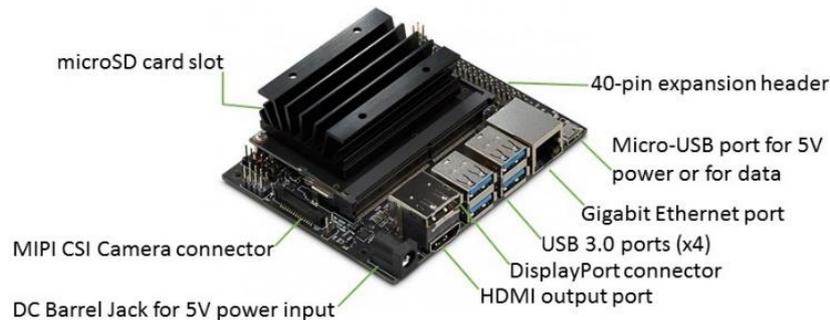


Figura 2.2 Jetson Nano

2.8 Micrófono USB

Un micrófono USB es un tipo de micrófono que se conecta a una computadora o dispositivo similar mediante un puerto USB. A diferencia de los micrófonos tradicionales que requieren una interfaz de audio o una tarjeta de sonido separada, los micrófonos USB tienen una interfaz de audio integrada. Esto significa que pueden convertir la señal de audio analógica directamente a una señal digital que puede ser interpretada por la computadora. En la Figura 2.3 se muestra el micrófono USB.



Figura 2.3 Micrófono USB

2.9 Protocolo de comunicación Ethernet

Es un protocolo que permite el intercambio de datos entre dispositivos que se encuentran conectados a una red local mediante cables de redes LAN. Este protocolo puede alcanzar velocidades de transmisión de hasta 1000 megabits por segundo y alcanzar una distancia de transmisión de 10 km gracias a el uso de fibra de vidrio (IONOS, 2022).

2.10 Arduino Nano

Es una placa de microcontrolador compacta y versátil, basada en ATmega328. Tiene características similares a placas Arduino más completas pero su tamaño reducido permite su adaptación a una mayor cantidad de entornos. En la Figura 2.4 se muestra el Arduino Nano.

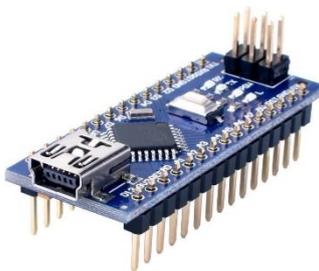


Figura 2.4 Arduino NANO

2.11 SIM800L

Es un módulo GSM/GPRS versátil y eficiente para añadir capacidades de comunicación móvil a proyectos electrónicos, ofreciendo una amplia gama de funcionalidades en un paquete compacto. En la Figura 2.5 se muestra el módulo SIM800L



Figura 2.5 SIM800L

CAPÍTULO 3

MARCO METODOLÓGICO

3.1 Propuesta del proyecto

Se plantea, mediante el desarrollo del proyecto, un sistema de seguridad inteligente para pequeños locales comerciales que utilice visión artificial y reconocimiento de voz para detectar posibles amenazas a clientes y empleados, y dar una alerta rápida al encargado del establecimiento.

Mediante la visión artificial, el sistema es capaz de reconocer tres objetos que generalmente se utilizan para cubrir el rostro que son mascarillas, pasamontañas y gorras. Para los patrones de voz, se utilizaron 10 frases amenazantes para empleados y clientes.

Se utiliza una cámara Logitech para grabar el entorno del local comercial y un micrófono USB que tomara las frases establecidas. Para el sistema de alerta se utiliza un Arduino Nano y un módulo celular en miniatura SIM800L que permite la transmisión GPRS. El sistema de control principal es una Jetson Nano que, gracias a su GPU, mejora el rendimiento de la detección de objetos, y también posee herramientas que mejoran la detección de patrones de voz.

Para lograr la mayor eficiencia en la detección de objetos y de patrones de voz, es necesario realizar algunas pruebas y comparar los resultados de estas, ya que los factores climáticos y las condiciones sonoras del establecimiento afectan directamente en la respuesta eficaz del sistema. En la Figura 3.1 se muestra el esquema general del sistema electrónico distribuido.

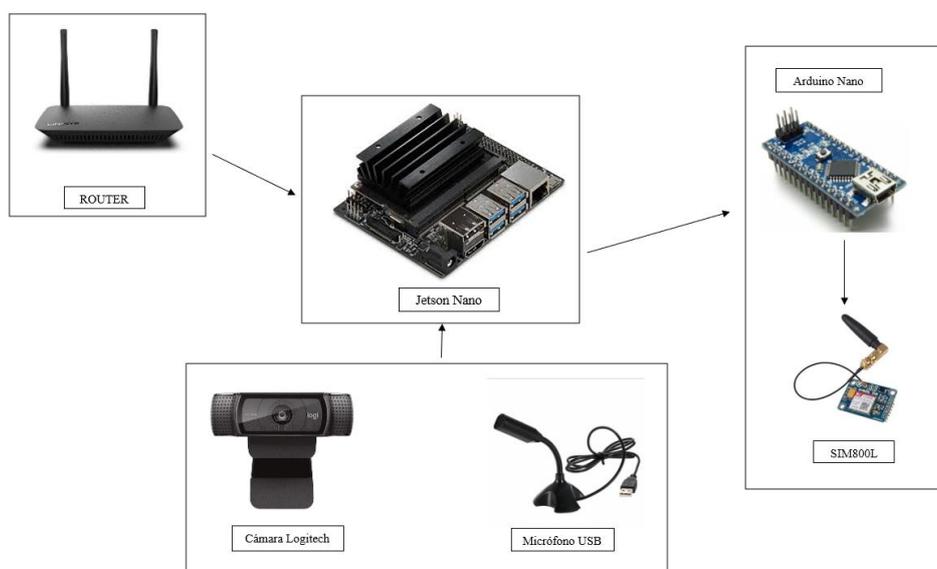


Figura 3.1 Esquema general del sistema electrónico distribuido

3.2 Variables para tomar en cuenta

Existen varias variables que se deben tomar en cuenta para implementar la visión artificial y el reconocimiento de voz. Estas variables son fundamentales, ya que darán pistas precisas de como configurar el sistema para que el sistema logre funcionar a toda su capacidad. Esto se detalla en la Tabla 3.1.

Variable	Tipo	Invasivo	Dispositivo
Ruido	Ambiental	Si	Micrófono
Distancia	Ambiental	No	Cámara y micrófono
Luminosidad	Ambiental	Si	Cámara

Tabla 3.1 Comparativa de variables

3.3 Partes del sistema distribuido

El sistema se distribuye en dos partes principales que son: el sistema de detección y el sistema de alarma. Para la implementación de la visión artificial, se deben construir una base de datos con las imágenes que se desea reconocer y después entrenar la red neuronal. Para el reconocimiento de voz se plantean las frases y palabras para ser analizadas mediante algoritmos avanzados de reconocimiento de voz. Para el sistema de alarma se desarrolla el algoritmo que controla el módulo celular.

3.4 Lenguaje de programación

El lenguaje de programación utilizado para el desarrollo del proyecto es Python, ya que cuenta con una gran cantidad de herramientas y librerías especializadas para el reconocimiento de voz y la visión artificial. Además, su simplicidad y legibilidad lo han convertido en el lenguaje de programación más extendido en el desarrollo de la inteligencia artificial, por lo que se encuentran gran cantidad de algoritmos que han sido implementados exitosamente en distintos campos.

El sistema de código abierto para detección de objetos YOLO ha aumentado enormemente las capacidades de la visión artificial. En este proyecto se utilizó la versión YOLO V5 para el entrenamiento de los datos.

Para la detección de patrones de voz se, utilizaron algoritmos avanzados que transforman la voz a texto, específicamente Google Speech Recognition que es capaz de reconocer una gran gama de idiomas y ofrece APIs que los desarrolladores pueden implementar en sus proyectos de forma gratuita.

3.5 Implementación electrónica

3.5.1 Tarjeta de desarrollo

Existen algunas tarjetas de desarrollo capaces de ejecutar algoritmos de inteligencia artificial, como Arduino, Esp32, Rasperry Pi entre otras. Sin embargo, las dos mejores opciones en cuanto a capacidad de procesamiento son la Rasperry Pi 4 y la Jetson Nano. En la Tabla 3.2 se muestran las principales ventajas y desventajas de cada una.

Tarjeta de desarrollo	Ventajas	Desventajas
Jetson Nano	<ul style="list-style-type: none"> ▪ Tiene una GPU de 128 núcleos que es excelente para tareas IA y Deep Learning ▪ Adecuada para tareas intensivas en computación como la visión artificial y el reconocimiento de voz. ▪ Compatible con frameworks de Deep Learning 	<ul style="list-style-type: none"> ▪ Es más costosa y su aplicación se limita a proyectos especializados ▪ Tiene un gran consumo energético
Rasperry Pi	<ul style="list-style-type: none"> ▪ Mas económica por lo que se puede emplear en proyectos limitados ▪ Cuenta con una gran comunidad de usuarios que facilita el encontrar recursos y soporte técnico. ▪ Tiene un consumo energético relativamente bajo. 	<ul style="list-style-type: none"> ▪ No tiene los requerimientos para aplicar tareas IA intensivas. ▪ La ejecución de algoritmos de visión artificial es mucho más lenta.

Tabla 3.2 Comparación entre las tarjetas de desarrollo

3.5.2 Algoritmo de entrenamiento de la red neuronal

Para el entrenamiento de la red neuronal se recolecto un conjunto de imágenes que contengan los objetos que se busca reconocer, después se utilizó YOLO V5 para entrenar la red neuronal. El algoritmo responde al diagrama de flujo de la Figura 3.2.

Las imágenes recopiladas se almacenan en dos carpetas que son images y valid, la carpeta images contiene las imágenes con las que se entrena la red neuronal y la carpeta valid cuenta con las imágenes que evalúan el rendimiento del modelo, con los datos recolectados hay que etiquetar cada imagen en formato YOLO para la red entrenar la red neuronal.

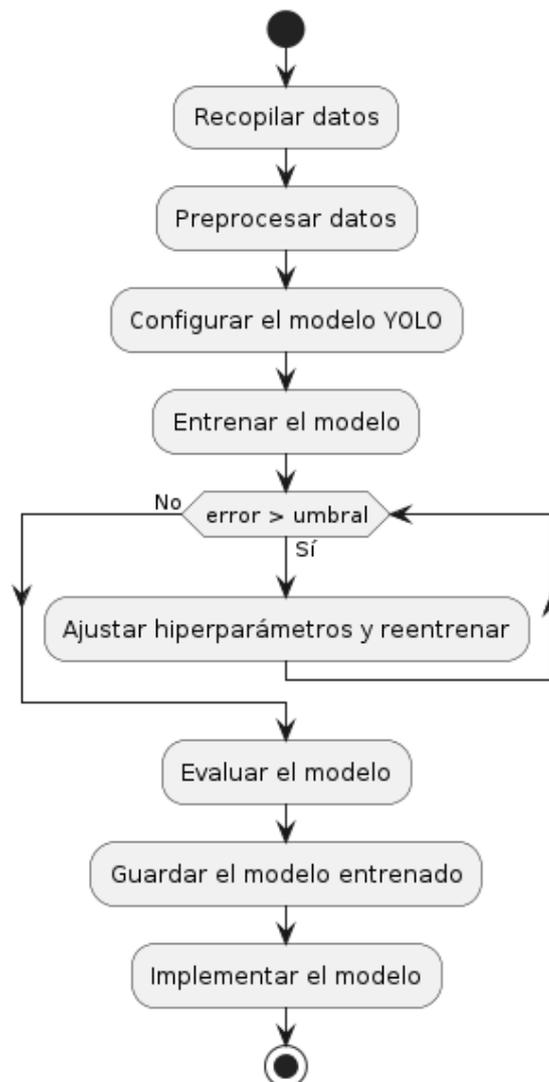


Figura 3.2 Diagrama de flujo el algoritmo del entrenamiento de imágenes

Para realizar el etiquetado de las imágenes se utilizó Makesense.ai que permite etiquetar las imágenes en formato YOLO, para esto es necesario subir las imágenes de los objetos que se va a reconocer y después seleccionar el área de la imagen que se necesita, como se muestra en la Figura 3.3.

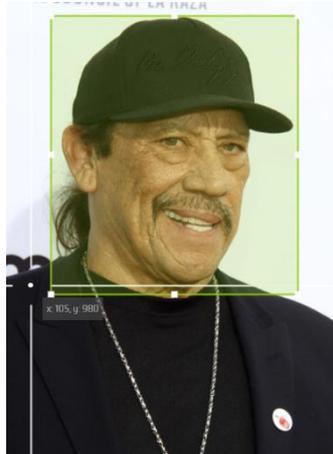


Figura 3.3 Etiquetado de imagen

Ya con las etiquetas colocadas se procede a entrenar la red neuronal, se deben instalar todas las librerías necesarias para el funcionamiento de YOLO V5, con las librerías instaladas se deben crear las clases que van a ser usadas para el entrenamiento en este caso las clases son gorras, mascarillas y pasamontañas. En la Figura 3.4 se muestra el algoritmo para la creación de las clases.

```
▶ # Crear archivo de configuración del dataset
dataset_config = """
train: /content/dataset/train/images
val: /content/dataset/val/images
nc: 3 # Número de clases (gorra, mascarilla, pasamontañas)
names: ['hat', 'mask', 'balaclava']
"""

with open('masks_hats_balaclavas.yaml', 'w') as f:
    f.write(dataset_config)
```

Figura 3.4 Creación de clases

Al crear las clases se genera un archivo .yaml, es este archivo el que se va a entrenar, se escogió 100 épocas para entrenar la red neuronal, al terminar un entrenamiento se genera un archivo .pt con el que se realizara la inferencia de las imágenes.

3.5.3 Programa de reconocimiento de objetos y voz

En la etapa de reconocimiento de voz no es necesario entrenar ninguna red neuronal ya que Google Speech Recognition transforma directamente el dialogo a texto y compara las frases establecidas en su base de datos.

El objetivo del código es combinar estas dos disciplinas de la inteligencia artificial sin que entren en conflicto para brindar un sistema de seguridad confiable, por lo que estos dos procesos no se realizan en simultaneo, ya que existe un pequeño delay entre la ejecución de la detección de objetos y el reconocimiento de patrones de voz.

El primer programa en ejecutarse es el de reconocimiento de objetos. Al finalizar su ejecución el programa, se envía un mensaje de alerta media, ya que las personas que usen gorras o mascarillas pueden ser clientes o asaltantes. Posteriormente, se ejecuta programa de detección de voz. Si se detecta una amenaza, se enviará un mensaje de alerta máxima, indicando que todo el local corre peligro. En la Figura 3.5 se muestra el diagrama de flujo del código.

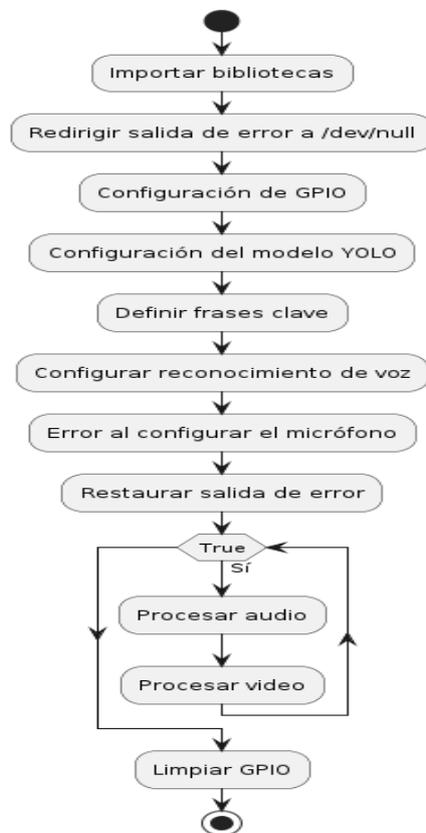


Figura 3.5 Diagrama de flujo del algoritmo para reconocimiento de voz y objetos

3.6 Sistema de alarma

Para la programación del sistema de alarma, se incluye la librería para la comunicación con el módulo SIM800L y se definen los pines de transmisión y recepción de voz e imágenes. Los pines se configuran como entradas con resistencias pull-up. Se envía un comando AT al SIM800L para verificar su funcionamiento. Se configura un bucle que envía el mensaje “Sujeto Sospechoso” cuando detecta un objeto y envía el mensaje “palabras detectadas” cuando detecta el comando de voz. Finalmente, se configura la función de enviar el mensaje, con un retardo de 30 segundos para no sobrecargar el módulo con múltiples mensajes en poco tiempo. En la Figura 3.6 se muestra el diagrama de flujo del código.

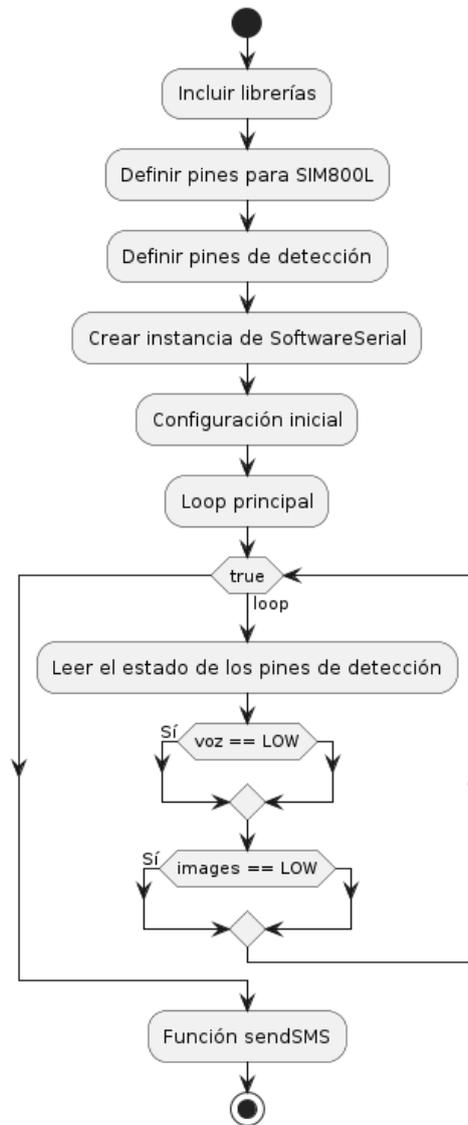


Figura 3.6 Diagrama de flujo del algoritmo para el sistema de alarma

El diseño de la PCB se realizó en el software PROTEUS, como se muestra en la Figura 3.7.

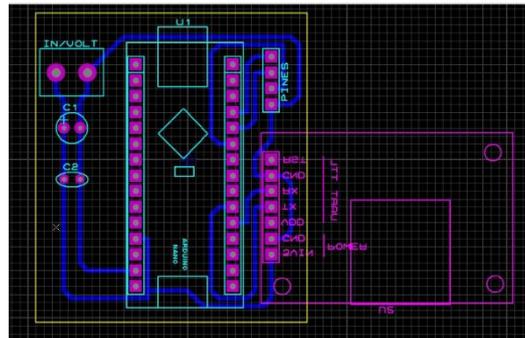


Figura 3.7 Diseño PCB en Proteus

Se utilizó el método de transferencia térmica para realizar la PCB. El resultado se muestra en la Figura 3.8.

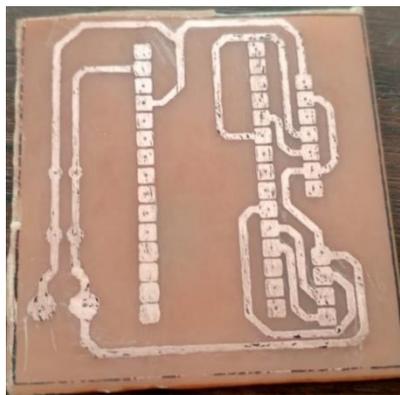


Figura 3.8 PCB con las pistas conductoras

La Figura 3.9 muestra los elementos soldados en la PCB.



Figura 3.9 Elementos soldados en la PCB

3.7 Estructura del equipo

En la Figura 3.10 se muestra la estructura que contiene la Jeston Nano y la placa PCB del sistema de alarma. La estructura cuenta con una ranura que para que los puertos y puntos de conexión del sistema sean fáciles de manipular.



Figura 3.10 Estructura del sistema de alarma

CAPÍTULO 4

PRUEBAS Y RESULTADOS

Es desarrollo del proyecto arrojó varios resultados que determinaron la efectividad del sistema de seguridad inteligente, se modificaron las variables más importantes que intervienen en el proceso de detección de objetos y reconocimiento de voz los cuales se explican a continuación.

4.1 Resultados del reconocimiento de objetos

Para determinar la calidad del reconocimiento se tomaron 3 muestras variando la distancia entre la cámara y el objeto. Se obtuvieron los siguientes resultados: la Figura 4.1 muestra el porcentaje de confianza de la detección de mascarillas a una distancia de dos metros.



Figura 4.1 Detección de mascarilla a 2 metros de distancia

La segunda prueba se realizó a 1.5 metros de distancia. El porcentaje de confianza de la detección se muestra en la Figura 4.2.



Figura 4.2 Detección de mascarilla a 1.5 metros de distancia

La tercera prueba se realizó a una distancia de 1 metro. El porcentaje de confianza de la detección se muestra en la Figura 4.3.



Figura 4.3 Detección de mascarilla a 1 metros de distancia

Se utilizaron las mismas distancias para determinar el porcentaje de confianza en la detección de gorras. En la Figura 4.4 se muestra el porcentaje de confianza a 2 metros de distancia.



Figura 4.4 Detección de gorra a 2 metros de distancia

El porcentaje de confianza a una distancia de 1.5 metros se muestra en la Figura 4.5.



Figura 4.5 Detección de gorra a 1.5 metros de distancia

La última prueba se realizó a 1 metro de distancia. En la Figura 4.6 se muestra el porcentaje de confianza

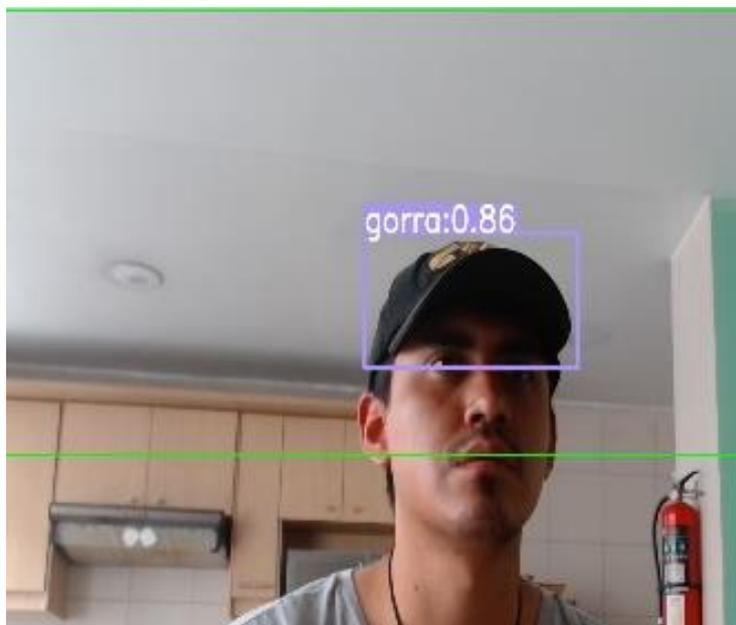


Figura 4.6 Detección de gorra a 1 metro de distancia

El porcentaje de confianza de detección del pasamontaña a una distancia de 2 metros se muestra en la Figura 4.7



Figura 4.7 Detección de gorra a 2 metros de distancia

El porcentaje de confianza de detección del pasamontaña a una distancia de 1.5 metros se muestra en la Figura 4.8



Figura 4.8 Detección de pasamontañas a 1.5 metros de distancia

El porcentaje de confianza de detección del pasamontaña a una distancia de 1 metro se muestra en la Figura 4.9

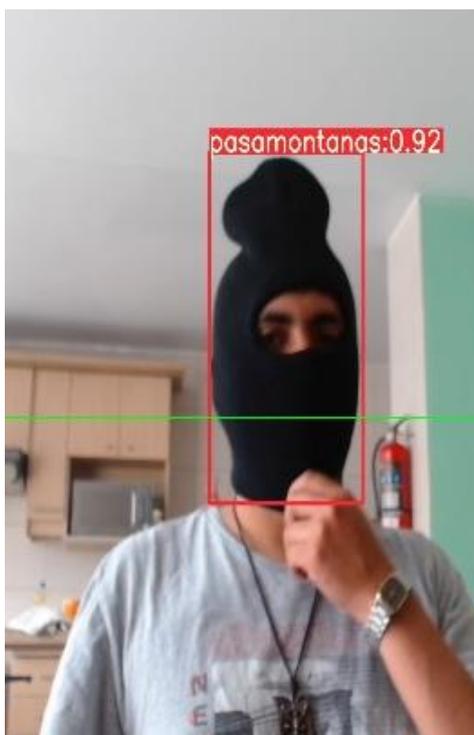


Figura 4.9 Detección de pasamontañas a 1 metros de distancia

Con las pruebas realizadas se obtuvieron los siguientes resultados:

OBJETO	DISTANCIA	% DE DETECCIÓN
MASCARILLA	2m	0.72
	1.5m	0.74
	1m	0.85
GORRA	2m	0.80
	1.5m	0.86
	1	0.92
PASAMONTAÑAS	2m	0.89
	1.5m	0.92
	1m	0.92

Tabla 4.1 Resultados de la detección de objetos

En base a los resultados obtenidos se observa que el porcentaje de detección en mascarillas es afectado por la distancia que hay entre el objeto y la cámara reduciendo un 13% la detección del objeto.

El porcentaje de detección en gorras se ve afectado pero su el error en la detección se reduce al 7%.

El objeto que presento un muy bajo porcentaje de error fue el pasamontaña ya que la distancia no influyo en la detección del objeto.

4.2 Resultados para el reconocimiento de voz

Se determinó que la distancia y el ruido afectan considerablemente el reconocimiento de los patrones de voz, la distancia causa una pérdida de la calidad de audio y recoge menos detalles de la voz, también reduce el volumen en la voz lo que hace partes del discurso inaudibles y difíciles de procesar.

La interferencia de ruido de fondo y el ruido estocástico generaron problemas en la detección de voz ya que causaron el solapamiento de las frecuencias de voz humanas lo que redujo la precisión del reconocimiento.

Se realizó una escala para determinar la precisión en el reconocimiento de voz afectado por la distancia y el ruido, los valores de la escala se muestran en la Tabla 4.2.

Escala	Precisión del reconocimiento
0	Nula
1	Baja
2	Media
3	Alta

Tabla 4.2 Escala de precisión para el reconocimiento de voz

Con la escala establecida se realizaron varias pruebas cambiando la distancia entre la fuente de voz y el micrófono. Los resultados obtenidos se muestran en la Tabla 4.3.

Distancia	Precisión del reconocimiento
0.5 m	3
0.75 m	3
1 m	2
1.5 m	1
1.75m	1
2 m	0

Tabla 4.3 Resultados de la distancia del reconocimiento de voz

En la Tabla 4.4 se muestran los valores de la escala del nivel de ruido utilizados para realizar las pruebas del reconocimiento de voz.

Escala	Nivel de ruido
0	Sin ruido
1	Ruido ambiente
2	Ruido moderado
3	Ruido fuerte

Tabla 4.4 Escala de nivel de ruido

Los resultados obtenidos de las pruebas realizadas con distintos niveles de ruido se muestran en la Tabla 4.5.

Escala	Precisión del reconocimiento
0	3
1	2
2	1
3	0

Tabla 4.5 Resultados del nivel de ruido en el reconocimiento de voz
Al ser detectados los objetos se envió exitosamente el mensaje “Sujeto sospechoso” vía SMS como se muestra en la Figura 4.10.

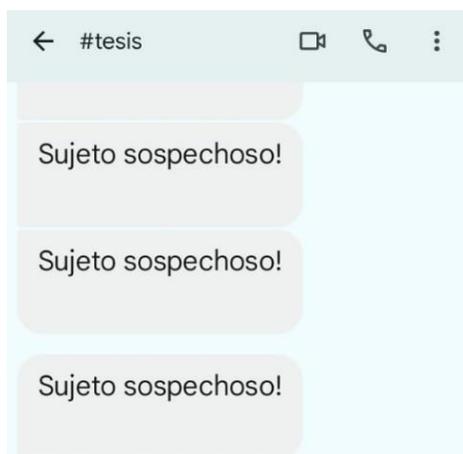


Figura 4.10 Mensaje de alerta de la detección de objetos

Mediante el reconocimiento de los comandos de voz se envió exitosamente el mensaje SMS “palabras detectadas” como se muestra en la Figura 4.11.



Figura 4.11 Mensaje de alerta del reconocimiento de voz

Las pruebas realizadas demostraron la eficiencia del sistema de seguridad y las condiciones que afectan su rendimiento, se comprobó que en la detección de objetos la distancia afecta la precisión del reconocimiento otro factor como el cambio de luz no influye en su rendimiento. En el reconocimiento de patrones de voz se determinó que el ruido y la distancia influyen significativamente en la precisión al detectar los comandos de voz.

La vinculación de la visión artificial y el reconocimiento de voz resultó exitosa ya que no existen conflictos de comunicación entre los algoritmos implementados, además el sistema de alarma tiene una respuesta rápida y eficaz lo que permite al personal de seguridad tomar acciones rápidas para prevenir ataques al local. Los resultados e información obtenidos de la implementación del sistema de seguridad inteligente en un ambiente real pueden ser la base para optimizar y mejorar nuevos sistemas.

CONCLUSIONES

Se desarrollo un sistema de vigilancia integrando la visión artificial para la detección precisa de objetos y el reconocimiento de lenguaje natural para la interacción automática, el sistema no solo mejora la eficiencia operativa, sino que también fortalece la capacidad de respuesta ante situaciones de seguridad. Este enfoque innovador no solo optimiza la supervisión del entorno físico del establecimiento, sino que también eleva el estándar de seguridad mediante la automatización inteligente de procesos críticos.

La revisión y síntesis de literatura científica sobre el desarrollo e implementación de sistemas de seguridad inteligentes en distintos campos mostraron varios conceptos fundamentales para el desarrollo de estos sistemas, lo que permitió establecer las herramientas y estrategias más adecuadas para el desarrollo e implementación del sistema.

Se modificaron algoritmos de visión artificial para la detección precisa y efectiva de objetos. Se emplearon las librerías del sistema código abierto YOLO V5 para el entrenamiento de la red neuronal y las librerías que realizan la inferencia del modelo entrenado. Además, el rendimiento de la programación mejoró al utilizar una GPU para el procesamiento de imágenes.

Se modificaron algoritmos de reconocimiento de lenguaje natural para la detección de patrones de voz, utilizando herramientas disponibles en Google Speech Recognition, lo cual permite realizar la transcripción de voz en tiempo real proporcionando una base para aplicaciones de seguridad y control mediante comandos de voz.

Se verifico el rendimiento del sistema de vigilancia inteligente mediante varias pruebas de campo, modificando las variables que influyen directamente en el rendimiento de la detección de objetos y en el reconocimiento de los patrones de voz, se determinó una eficiencia total del 75%.

RECOMENDACIONES

Se recomienda continuar con la investigación en las disciplinas de visión artificial el reconocimiento de voz aplicadas a los sistemas de seguridad. Conocer nuevos métodos de desarrollo e implementación de estos sistemas permitiría sugerir mejoras adicionales al rendimiento del sistema.

Generar medios informativos que expliquen de manera sencilla y concisa las ventajas y desventajas de los sistemas de vigilancia inteligentes, permitiendo a la comunidad tomar una decisión informada al elegir entre un sistema de seguridad convencional y uno inteligente.

Optimizar el entrenamiento de la red neuronal para la detección de objetos con un conjunto de datos más robusto que incluya imágenes en distintas condiciones de luz y orientación, de modo que el reconocimiento de objetos no se vea afectado por el clima u otras variables.

Implementar el sistema de seguridad en distintos entornos y verificar los resultados obtenidos en cada uno de ellos para establecer los cambios y mejoras necesarias que permitan que el sistema sea adaptable y pueda ser implementado en establecimientos distintos a pequeños locales comerciales.

Realizar una planificación para el mantenimiento del sistema de seguridad para prevenir fallas y para reparar daños en el sistema, manteniendo la eficiencia del sistema y alargando el funcionamiento de todos sus componentes.

Referencias Bibliográficas

- Amazon. (2023). *¿Qué es la NLP?* Obtenido de Amazon: [https://aws.amazon.com/es/what-is/nlp/#:~:text=El%20procesamiento%20de%20lenguaje%20natural%20\(NLP\)%20es%20fundamental%20para%20analizar,t%C3%ADpicas%20en%20las%20conversaciones%20cotidianas.](https://aws.amazon.com/es/what-is/nlp/#:~:text=El%20procesamiento%20de%20lenguaje%20natural%20(NLP)%20es%20fundamental%20para%20analizar,t%C3%ADpicas%20en%20las%20conversaciones%20cotidianas.)
- Cohen, A. (5 de Mayo de 2023). *Linkedin Pulse*. Obtenido de *¿Qué es un sistema de seguridad inteligente?*: <https://www.linkedin.com/pulse/qu%C3%A9-es-un-sistema-de-seguridad-inteligente-ariel-cohen/>
- IBM. (2022). *¿Qué es PyTorch?* Obtenido de IBM: <https://www.ibm.com/es-es/topics/pytorch#:~:text=%C2%BFC%C3%B3mo%20funciona%20PyTorch%3F,de%20las%20redes%20neuronales%20profundas.>
- IBM. (2022). *IBM*. Obtenido de *¿Qué es PyTorch?*: <https://www.ibm.com/mx-es/topics/pytorch>
- Imagina. (3 de Marzo de 2024). *Procesamiento de imágenes con OpenCV en Python*. Obtenido de *Imagina Informacion* : <https://imaginaformacion.com/tutoriales/opencv-en-python>
- IONOS. (12 de Diciembre de 2022). *Digital Guide IONOS*. Obtenido de *¿Qué es Ethernet?*: <https://www.ionos.es/digitalguide/servidores/know-how/ethernet-ieee-8023/#:~:text=Ethernet%20designa%20a%20una%20tecnolog%C3%ADa,se%20crea%20mediante%20conexiones%20Ethernet.>
- Logitech. (2024). *Logitech*. Obtenido de *WEBCAM C920 HD PRO*: <https://www.logitech.com/es-es/products/webcams/c920-pro-hd-webcam.960-001055.html>
- Mínguez, T. D. (2021). *Visión Artificial Aplicaciones Prácticas Con OpenCV-Python*. Barcelona : Marcombo .
- Numpy. (20 de Abril de 2020). *What is Numpy?* Obtenido de Numpy: <https://numpy.org/doc/stable/user/whatisnumpy.html>
- NVIDIA. (2024). *NVIDIA*. Obtenido de *Jetson Nano*: <https://www.nvidia.com/es-la/autonomous-machines/embedded-systems/jetson-nano/product-development/>

Primicias. (28 de 07 de 2023). *Hay más robos a negocios los lunes por la mañana*. Obtenido de Primicias: <https://www.primicias.ec/noticias/economia/robo-negocios-inseguridad-empresas/#:~:text=Los%20robo%20a%20negocios%2C%20f%C3%A1bricas,la%20Fiscal%3ADa%20General%20del%20Estado>.

V-alje, Q. (11 de Noviembre de 2023). *Learn-microsoft*. Obtenido de Entrenamiento de un modelo de detección de objetos con TensorFlow: <https://learn.microsoft.com/es-es/windows/ai/windows-ml/tutorials/tensorflow-train-model>

Anexos

En el Anexo 1 se muestra el Código en Python para la detección de objetos y comandos de voz.

Anexo 1. Código en Python para la detección de objetos y comandos de voz

```
import cv2
import speech_recognition as sr
import Jetson.GPIO as GPIO
import time
from yoloDet import YoloTRT # Asegúrate de que esta clase esté
correctamente definida y disponible
import os
import sys

# Redirigir la salida de error a /dev/null para silenciar los
mensajes de ALSA
sys.stderr = open(os.devnull, 'w')

# Configuración de GPIO
LED_PIN_IMAGE = 18
LED_PIN_AUDIO = 21
GPIO.setmode(GPIO.BOARD)
GPIO.setup(LED_PIN_IMAGE, GPIO.OUT)
GPIO.setup(LED_PIN_AUDIO, GPIO.OUT)
GPIO.output(LED_PIN_IMAGE, GPIO.HIGH)
GPIO.output(LED_PIN_AUDIO, GPIO.HIGH)

# Configuración de YOLO para detección de imágenes
model = YoloTRT(library="yolov5/build/libmyplugins.so",
engine="yolov5/build/yolov5s.engine", conf=0.7, yolo_ver="v5")
```

```

# Configuración del reconocimiento de voz
frases_clave = [
    "dame el dinero",
    "ten cuidado",
    "no te muevas",
    "no grites",
    "estoy armado",
    "abre la caja",
    "haz lo que te digo",
    "date prisa",
    "manos arriba",
    "robo"
]

try:
    recognizer = sr.Recognizer()
    mic_index = 12 # Asegúrate de que este es el índice correcto
para tu micrófono USB
    mic = sr.Microphone(device_index=mic_index)
finally:
    # Restaurar la salida de error estándar
    sys.stderr = sys.__stderr__

def encender_led(pin):
    print(f"Encendiendo el LED en el pin {pin}")
    GPIO.output(pin, GPIO.LOW)
    time.sleep(5) # Mantén el LED encendido por 5 segundos
    GPIO.output(pin, GPIO.HIGH)
    print(f"Apagando el LED en el pin {pin}")

```

```

def detectar_frases_en_texto(text):
    for frase in frases_clave:
        if frase in text:
            print(f"Frase clave detectada: {frase}")
            encender_led(LED_PIN_AUDIO)
            return True # Frase detectada, detener la búsqueda
    return False

```

En el Anexo 2 se muestra la programación del Arduino Nano para el sistema de alarma.

Anexo 2. Código del Arduino Nano del sistema de alarma

```
#include <SoftwareSerial.h>
```

```
// Configuración del pin para el SIM800L
```

```
#define SIM800_RX 3
```

```
#define SIM800_TX 2
```

```
// Pin de entrada para la detección
```

```
#define DETECTION_PIN 7
```

```
#define DETECTION_PIN_2 8
```

```
// Crear una instancia de SoftwareSerial para la comunicación con el SIM800L
```

```
SoftwareSerial sim800(SIM800_RX, SIM800_TX);
```

```
void setup() {
```

```
    // Configurar el pin de detección como entrada
```

```
    pinMode(DETECTION_PIN, INPUT_PULLUP);
```

```
    pinMode(DETECTION_PIN_2, INPUT_PULLUP);
```

```

// Iniciar la comunicación serial para monitoreo
Serial.begin(9600);

// Iniciar la comunicación con el SIM800L
sim800.begin(9600);

// Esperar un momento para que el SIM800L se inicie
delay(1000);

// Verificar que el SIM800L responda
sim800.println("AT");
delay(1000);
if (sim800.available()) {
  Serial.println("SIM800L inicializado correctamente.");
  Serial.println(sim800.readString());
} else {
  Serial.println("No se pudo inicializar el SIM800L.");
}
}

void loop() {
  // Leer el estado del pin de detección
  int voz = digitalRead(DETECTION_PIN);
  int images = digitalRead(DETECTION_PIN_2);
  Serial.print("Voz: ");
  Serial.println(voz);
  Serial.print("Images: ");
  Serial.println(images);
}

```

```

// Si se detecta un estado HIGH en el pin de detección, enviar un SMS
if (voz == LOW) {
    sendSMS('palabras detectadas');
    delay(5000); // Esperar 5 segundos para evitar múltiples mensajes en poco tiempo
}

if (images == LOW) {
    sendSMS('Sujeto sospechoso!');
    delay(5000); // Esperar 5 segundos para evitar múltiples mensajes en poco tiempo
}

}

void sendSMS(String message) {
    // Configurar el SIM800L para modo de mensaje de texto
    sim800.println("AT+CMGF=1");
    delay(1000);

    // Indicar el número de teléfono al que se enviará el mensaje
    sim800.println("AT+CMGS=\"+593961701007\"); // Reemplaza con el número de
teléfono real
    delay(1000);

    // Enviar el mensaje
    sim800.println(message);
    delay(1000);

    // Enviar el comando de terminación de mensaje
    sim800.write(26); // Ctrl+Z en ASCII

```

```
delay(1000);
```

```
// Leer la respuesta del SIM800L
```

```
if (sim800.available() {
```

```
    Serial.println(sim800.readString());
```

```
}
```

```
delay(30000);
```

```
Serial.println("Termino de enviar");
```

```
}
```

En el Anexo 3 se muestra el entorno de instalación del Sistema de vigilancia

Anexo 3. Entorno de instalación del local comercial “El Chaval”



En el Anexo 4 se muestra la instalación de la cámara en la parte superior del local para tener un mayor rango visión.

Anexo 4. Colocación de la cámara



En el Anexo 5 se muestra la instalación del micrófono

Anexo 5. Colocación del micrófono



En el Anexo 6 se muestran las conexiones del sistema de vigilancia.

Anexo 6. Conexiones del sistema de vigilancia



En el Anexo 7 se muestran el sistema instalado en el local comercial

Anexo 7. Sistema instalado

