



UNIVERSIDAD TECNOLÓGICA ISRAEL
ESCUELA DE POSGRADOS "ESPOG"

MAESTRÍA EN ELECTRÓNICA Y AUTOMATIZACIÓN
Resolución: RPC-SO-09-No.265-2021

PROYECTO DE TITULACIÓN EN OPCIÓN AL GRADO DE MAGÍSTER

Título del proyecto:
Sistema de videovigilancia comunitaria mediante visión artificial
Línea de Investigación:
Automatización, visión artificial, seguridad y a la sociedad
Campo amplio de conocimiento:
Ingeniería, Industria y construcción
Autor/a:
Ing. Edwin Santiago Sarabia Velásquez
Tutor/a:
Mg. René Ernesto Cortijo

Quito – Ecuador

2023

APROBACIÓN DEL TUTOR



Yo, **Rene Ernesto Cortijo Leyva** con C.I: **1719010108** en mi calidad de Tutor del proyecto de investigación titulado: **Sistema de videovigilancia comunitario mediante visión artificial,**

Elaborado por: **Edwin Santiago Sarabia Velásquez**, de C.I: **1718186115**, estudiante de la Maestría: **ELECTRÓNICA Y AUTOMATIZACIÓN**, resolución: **RPC-SO-09-No.265-2021**, de la **UNIVERSIDAD TECNOLÓGICA ISRAEL (UISRAEL)**, como parte de los requisitos sustanciales con fines de obtener el Título de Magister, me permito declarar que luego de haber orientado, analizado y revisado el trabajo de titulación, lo apruebo en todas sus partes.

Quito D.M., 2 de septiembre del 2023

RENE
ERNEST
O
CORTIJO
LEYVA

Firmado digitalmente por RENE ERNESTO CORTIJO LEYVA
Fecha: 2023.09.03
0956.09-05100

Firma

DECLARACIÓN DE AUTORIZACIÓN POR PARTE DEL ESTUDIANTE



Yo, **Ing. Edwin Santiago Sarabia Velásquez** con C.I: 1718186115, autor del proyecto de titulación denominado: **Sistema de videovigilancia comunitario mediante visión artificial**.
Previo a la obtención del título de Magister en **Electrónica y Automatización**.

1. Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar el respectivo trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.
2. Manifiesto mi voluntad de ceder a la Universidad Tecnológica Israel los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador, artículos 4, 5 y 6, en calidad de autor@ del trabajo de titulación, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital como parte del acervo bibliográfico de la Universidad Tecnológica Israel.
3. Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de prosperidad intelectual vigentes.

Quito D.M., 2 de septiembre del 2023

Firma

DEDICATORIA

El presente trabajo va dedicado a mi familia, a mi madre Martha por ser la persona quien me ha enseñado a no desfallecer en mis sueños y siempre estar perseverante brindándome sus sabios consejos, demostrándome su cariño, amor, apoyo incondicional y por ser el pilar más importante en todo el trayecto de vida y carrera profesional, siendo quien corrige mis fallas y celebra mis triunfos y alegrías.

A mis hermanos, Evelyn y Javier por ser mis grandes amigos y brindarme su apoyo para desarrollarme en el ámbito profesional y personal en cada uno de los momentos buenos, malos y estar durante todo este proceso demostrándome el valor de tener una gran familia.

A Fernando, un gran amigo a quien estimo y valoro tener su amistad, por ser una persona que demuestra que a pesar de las circunstancias se pueden lograr cada una de las metas soñadas gracias por su apoyo y ayuda incondicional.

A mi compañera Liseth por la ayuda brindada en las situaciones y momentos más difíciles, por motivarme, eres una persona muy importante en mi vida que siempre te agradeceré por todo.

Agradezco a mis maestros y tutor, Magíster Rene Cortijo, que nos brindaron paciencia tiempo, confianza y sobre todo sus conocimientos en el desarrollo del proyecto.

ING. SANTIAGO SARABIA

Tabla de contenidos

APROBACIÓN DEL TUTOR.....	2
DECLARACIÓN DE AUTORIZACIÓN POR PARTE DEL ESTUDIANTE	3
INFORMACIÓN GENERAL	1
Contextualización del tema	1
Problema de investigación	2
Objetivo general.....	3
Objetivos específicos.....	3
Vinculación con la sociedad y beneficiarios directos.....	4
CAPÍTULO I: DESCRIPCIÓN DEL PROYECTO.....	6
Contextualización general del estado del arte	6
Proceso investigativo metodológico	8
CAPÍTULO II: PROPUESTA	10
Fundamentos teóricos aplicados.....	10
Inteligencia Artificial	10
Visión Artificial	11
Machine Learning	11
Deep Learning	12
Redes Neuronales Convolucionales	12
Librerías	13
Python	14
Google Drive	15
Edge IA	15
Protocolo de comunicación RSTP	15
Jetson Nano NVIDIA	16
Ubuntu	17
Descripción de la propuesta	17
DVR	19
Cámaras Color VU	20
Sirena de 30W	21
Jetson Nano	22
Validación de la propuesta.....	34
Matriz de articulación de la propuesta.....	37

Análisis de resultados. Presentación y discusión.....	38
CONCLUSIONES.....	41
RECOMENDACIONES.....	42
Referencias	43
ANEXOS.....	45

Índice de tablas

Tabla 1	Características técnicas del DVR	19
Tabla 2	Especificaciones técnicas de las cámaras tipo bullet	20
Tabla 3	Especificaciones técnicas de la sirena audible de 30 w	21
Tabla 4	Especificaciones técnicas de la tarjeta jetson nano	22
Tabla 5	Perfiles de profesionales validadores del proyecto	34
Tabla 6	Criterios de evaluación de los expertos.....	35
Tabla 7	Escala de evaluación del proyecto.....	35
Tabla 8	Matriz del proyecto de propuesta y articulación.....	37
Tabla 9	Datos obtenidos de la primera cámara en monitoreo	38
Tabla 10	Datos obtenidos de la segunda cámara en monitoreo	38
Tabla 11	Resultados de la implementación del sistema.....	39
Tabla 12	Resultados del funcionamiento del sistema	39

Índice de figuras

Figura. 1 Inteligencia artificial	10
Figura. 2 Redes neuronales convolucionales	13
Figura. 3 Jetson Nano Nvidia.....	16
Figura. 4 Diagrama de flujo del sistema de videovigilancia mediante visión artificial.....	18
Figura. 5 Arquitectura del Proyecto	18
Figura. 6 DVR eSSD de 4 canales 1080 lite	20
Figura. 7 Cámara color VU de tipo bala para exterior	21
Figura. 8 Sirena de 30W	21
Figura. 9 Jetson nano Nvidia	23
Figura. 10 Adquisición de imágenes del grabador a la jetson nano	23
Figura. 11 Detección de personas mediante técnicas basadas en redes neuronales.....	24
Figura. 12 Estimación de la postura humana	24
Figura. 13 Señal de auxilio detectada y activación de la alarma	25
Figura. 14 Interfaz de usuario creada para seleccionar el área de detección de la señal.....	25
Figura. 15 Api de google drive.....	26
Figura. 16 Flasheo en la micro sd	27
Figura. 17 Interfaz del sistema operativo en la jetson nano	27
Figura. 18 Librerías de Python	28
Figura. 19 Modelo ultralytics YoloV8	28
Figura. 20 Modelos de pose de yoloV8.....	29
Figura. 21 Modelo yoloV8x pose.....	29
Figura. 22 Programación del modelo pose.....	30
Figura. 23 Programación para el recorte del áreas mediante los puntos de la interfaz gráfica	31
Figura. 24 Conexión de las cámaras con la jetson nano	31
Figura. 25 Configuración de la guide de la interfaz grafica	32
Figura. 26 Interfaz de usuario del sistema de videovigilancia	32
Figura. 27 Configuración del Api de google drive.....	33
Figura. 28 Interfaz gráfica de usuario.....	33

INFORMACIÓN GENERAL

Contextualización del tema

La seguridad es un tema crucial cada vez más relevante en las ciudades, comunidades que buscan mejorar su seguridad y la tecnología de videovigilancia comunitaria teniendo un principio fundamental en la prevención y reducción de delitos. La visión artificial es una tecnología emergente que permite la supervisión y seguridad utilizando cámaras y software de reconocimiento de imágenes, esto se ha vuelto cada vez más popular en las ciudades debido al aumento de la delincuencia y los robos, lo que ha llevado a los residentes a buscar soluciones innovadoras para garantizar su seguridad y la de sus propiedades. Mediante el uso de esta herramienta es posible implementar sistemas de seguridad más eficientes y efectivos que puedan detectar patrones de comportamiento y alertar a las autoridades o vecinos sobre posibles amenazas.

Este trabajo de investigación tiene como propósito un sistema de videovigilancia comunitaria mediante visión artificial que implica la integración de varios componentes, como cámaras de alta resolución, alarma, software de reconocimiento de imágenes y almacenamiento de datos. Estos componentes trabajan juntos para capturar imágenes de alta calidad y analizarlas para detectar comportamientos sospechosos o potencialmente peligrosos.

El sistema debe ser capaz de detectar comportamientos anormales, como detectar señal de auxilio ante algún evento de robo y activar la alarma o enviar una notificación a las autoridades o vecinos. También se pueden incluir funciones como reconocimiento facial para identificar a la persona que activó el sistema.

En este contexto, el presente proyecto se desarrollará en la provincia de Pichincha, cantón Mejía parroquia de Alóag en la cooperativa Nuevo Alóag, está conformado por alrededor de 20 casas y una población aproximada de 60 habitantes, la cooperativa cuenta con un pasaje y una calle principal que es la bahía de caráquez donde se ubican algunos locales comerciales. El barrio en la actualidad tiene entornos de inseguridad, robos a los transeúntes y habitantes del sector con una falta de respuesta ante estos sucesos por parte de la policía nacional.

En vista de los fundamentos y la falta de seguridad antes mencionados, se desea desarrollar un sistema de videovigilancia comunitaria mediante visión artificial. Para ello, se llevará a cabo una investigación detallada sobre las técnicas de visión artificial, algoritmos de aprendizaje, seleccionar la tecnología adecuada en hardware y software, se diseñará el sistema de visión

artificial, estableciendo protocolos de respuesta y coordinación, se implementará y validará el funcionamiento del sistema garantizando la privacidad y protección de datos. Con este proyecto se espera contribuir significativamente a mejorar la seguridad en los barrios, disminuir los robos, promoviendo la participación y colaboración de la ciudadanía en la implementación de soluciones innovadoras y efectivas.

Además de mejorar la seguridad en el barrio, el sistema de videovigilancia comunitaria mediante visión artificial puede tener otras ventajas. Por ejemplo, puede ayudar a reducir los costos de seguridad al permitir que la ciudadanía supervise y responda a situaciones de alerta en tiempo real. También puede mejorar la eficiencia de los servicios de emergencia al proporcionar información valiosa sobre la ubicación e imagen del evento de anomalía que se suscite.

Problema de investigación

A nivel global, la delincuencia presente en diversos países conlleva a un incremento significativo en los niveles de inseguridad, resaltando así la vital importancia de abordar este desafío. En el contexto latinoamericano, la evolución constante de la tecnología se hace evidente y en este sentido, la adopción de sistemas de videovigilancia comunitaria basados en visión artificial emerge como una medida para proporcionar seguridad y confianza frente a potenciales incidentes que puedan surgir en la comunidad.

En Ecuador, específicamente en las ciudades como Quito, Guayaquil y Cuenca que son las más grandes del país no escapan de este problema. Durante el último año, se han incrementado los incidentes de robos a ciudadanos y extranjeros, una tendencia que se refleja de manera destacada en las publicaciones que emite la prensa, noticias, redes sociales, artículos y videos, estas perturbaciones que se tienen en la sociedad han generado preocupación al salir por el miedo de ser víctimas de la delincuencia.

En consecuencia, el proyecto busca indagar en la efectividad y aplicabilidad de la visión artificial en la videovigilancia comunitaria, analizando su potencial para mejorar la detección de comportamientos anómalos y eventos inusuales. Al mismo tiempo, se pretende explorar la integración de sistemas de alerta temprana que permitan una respuesta más eficaz por parte de las autoridades o los operadores. La investigación también considerará la interacción entre la visión artificial y otros componentes tecnológicos, como la comunicación con dispositivos móviles o sistemas de automatización de seguridad. En última instancia, el objetivo es ofrecer un enfoque integral para implementar un sistema de videovigilancia comunitaria mediante

visión artificial que sea eficiente, preciso y éticamente sólido, brindando una contribución valiosa a la mejora de la seguridad en el barrio.

En la cooperativa de vivienda nueva Alóag, se tiene un problema de seguridad preocupante y persistente. Los residentes han sido testigos de un aumento constante en incidentes delictivos, como robos a viviendas, vandalismo y actos de violencia callejera. Estos eventos han generado un clima de inseguridad que afecta la calidad de vida de todos los habitantes. La falta de un sistema de vigilancia adecuado ha dificultado la identificación y seguimiento de los delincuentes, lo que ha llevado a una sensación generalizada de vulnerabilidad. Los residentes evitan salir de noche y experimentan ansiedad por la seguridad de sus hogares y seres queridos. Además, el barrio ha experimentado una disminución en la participación en actividades sociales y cívicas debido al temor a posibles incidentes. Por lo tanto, el problema crítico a abordar es cómo implementar un sistema de videovigilancia mediante visión artificial que brinde una respuesta efectiva y disuade la actividad delictiva, restaurando la sensación de seguridad y promoviendo una mayor unión comunitaria.

Es por eso que surge la necesidad de diseñar un sistema de videovigilancia comunitaria mediante visión artificial que implica abordar una serie de desafíos técnicos para desarrollar un sistema eficiente y responsable que satisfaga las necesidades de seguridad de una comunidad o institución. Este prototipo debe ser capaz de detectar patrones sospechosos y enviar alertas en tiempo real a las autoridades o vecinos. Este sistema debe ser seguro, eficiente, flexible y escalable, que pueda ser implementado de manera colaborativa con la comunidad y ciudadanía.

Por lo antes expuesto, el desarrollo de este trabajo es factible por lo que se pretende incorporar un prototipo con visión artificial que se integren a los grabadores tradicionales de videovigilancia que no poseen este tipo de tecnología, esto debido que en la actualidad se tienen equipos de videovigilancia con inteligencia artificial muy costosos, y realizar el cambio de estos equipos significa una inversión elevada, por tal motivo la implementación de este proyecto es muy importante por el aporte a la comunidad y a la industria de la seguridad electrónica.

Objetivo general

Desarrollar un sistema de videovigilancia comunitaria mediante visión artificial.

Objetivos específicos

Definir los parámetros y reglas de análisis en la imagen para desarrollar el algoritmo de visión artificial

Crear el algoritmo de visión artificial mediante la biblioteca de OpenCV y el lenguaje de programación de Python.

Implementar el algoritmo de visión artificial en un servidor con Linux con almacenamiento de imágenes en drive e interfaz gráfica para la visualización.

Realizar la instalación del sistema de videovigilancia comunitaria mediante visión artificial en puntos estratégicos de la comunidad con dos cámaras color Vu.

Validar el funcionamiento del sistema de videovigilancia para verificar su robustez y efectividad.

Vinculación con la sociedad y beneficiarios directos

El desarrollo del sistema de videovigilancia comunitaria mediante visión artificial tiene varios beneficiarios directos. En primer lugar, los residentes de la cooperativa de vivienda nuevo Aloag, serán los principales beneficiarios, ya que el sistema ayudará a mejorar la seguridad en la zona. Con la instalación de cámaras de seguridad y la utilización de tecnología de visión artificial, se podrá monitorear la actividad en la zona y detectar cualquier actividad sospechosa. Esto puede ayudar a disuadir a los delincuentes y reducir el número de robos, vandalismo y otros delitos que se pueden cometer en la zona.

Además, la policía y otros organismos de seguridad también se beneficiarán del sistema. Al contar con acceso a la información y datos recopilados por las cámaras y la tecnología de visión artificial, podrá responder de manera más efectiva a cualquier actividad sospechosa o delictiva en la zona. Esto permitirá una respuesta más rápida y eficaz ante cualquier situación de emergencia o peligro. En general, el sistema de videovigilancia comunitaria mediante visión artificial tiene el potencial de mejorar la calidad de vida y la seguridad de los residentes del barrio, así como mejorar la capacidad de respuesta y el trabajo de la policía y otros organismos de seguridad en la zona.

Por otro lado, la vinculación con la sociedad también puede ayudar a garantizar la aceptación del sistema y reducir las preocupaciones que se tiene dentro de la comunidad. Es importante comunicar los beneficios del sistema y cómo se utilizarán los datos y la información capturada. También es necesario garantizar que se cumplan los estándares y cumplir con la privacidad y protección de datos, y que se involucre a la comunidad en el monitoreo y control del sistema.

Al implementar este sistema, se aspira a fomentar una mayor solidaridad entre los vecinos al dejar un papel activo en la protección de su entorno. La expectativa es que la percepción de seguridad aumente, mientras que la sensación de unidad y cooperación se intensifique entre los

residentes. La visión de contar con un sistema de videovigilancia colectivo puede motivar a la comunidad a trabajar en conjunto para prevenir situaciones delictivas.

En esencia, el desarrollo del proyecto no solo busca mejorar la seguridad física, sino también construir lazos sociales más sólidos y generar confianza entre los vecinos. La tecnología actúa como un catalizador para lograr un objetivo más amplio: promover la participación cívica y la solidaridad en un esfuerzo conjunto para desalentar la delincuencia.

CAPÍTULO I: DESCRIPCIÓN DEL PROYECTO

Contextualización general del estado del arte

En la elaboración de este proyecto, se realizó una revisión minuciosa de investigaciones previas relacionadas con sistemas de videovigilancia comunitaria. Se exploraron diversos artículos científicos a nivel nacional e internacional, así como tesis de grado tanto a nivel de pregrado como de doctorado, que se enmarcan dentro del campo de la videovigilancia. Durante esta revisión, se identificaron trabajos cuyas variables y enfoques guardan relevancia con la presente investigación.

En esta perspectiva, se presenta el trabajo titulado *“Diseño e Implementación de un prototipo de sistema de seguridad por videovigilancia automático dentro de un área externa restringida, utilizando visión artificial para la búsqueda, reconocimiento, seguimiento y neutralización de personas”*. Para lograr este objetivo, se adquirió la metodología de desarrollo de *software* ágil *SCRUM*, implementando un algoritmo de detección, reconocimiento, seguimiento y neutralización de personas mediante el uso de la librería de visión artificial *OpenCV* y una interfaz gráfica de usuario desarrollada en Qt. Los resultados muestran que el sistema es capaz de detectar y seguir personas con una alta precisión, incluso en condiciones de poca luz y largas distancias. Además, el sistema mecánico diseñado permite que la persona detectada coincida con el centro de la imagen y se pueda utilizar el sistema de neutralización mediante una marcadora de paintball. En conclusión, este prototipo demuestra la efectividad de la visión artificial y la videovigilancia para la implementación de sistemas de seguridad automatizados en áreas restringidas (FIGUEROA & ARIAS, 2018).

Por otra parte, la investigación titulada *“Diseño de un Sistema de Seguridad a través de Cámaras de Videovigilancia para el Laboratorio 14 de la Carrera de Ingeniería en Computación y Redes”*. El objetivo del proyecto es diseñar e implementar un sistema de seguridad mediante la instalación de cámaras de videovigilancia para el monitoreo en tiempo real del laboratorio 14 de la carrera. La metodología utilizada en este estudio es mixta, cuantitativa y cualitativa, utilizando métodos deductivos, de análisis, bibliográficos y propositivos. Se llevaron a cabo encuestas y entrevistas para determinar la mejor tecnología para la conexión del sistema de videovigilancia y el tipo de cámaras a utilizar. Como resultado, se logró la implementación exitosa del sistema de seguridad a través de cámaras de videovigilancia, permitiendo la visualización en tiempo real desde cualquier dispositivo conectado a la misma red de datos, brindando seguridad a la comunidad estudiantil. En conclusión, este proyecto es de gran

importancia para la seguridad del laboratorio y de los estudiantes de la carrera de Ingeniería en Computación y Redes (LUCAS, 2019).

Se presenta a continuación el proyecto de investigación titulado "*Sistema de vigilancia mediante visión por computadora para el Centro de Educación Inicial N°1 del Ministerio de Educación*". Su objetivo principal radica en la implementación de un sistema de vigilancia que sea confiable y seguro, con la finalidad de reducir los riesgos de seguridad existentes en dicho establecimiento. Este sistema se apoya en las últimas tendencias tecnológicas relacionadas con la inteligencia artificial. Para lograr este propósito, se perfeccionó la metodología de investigación experimental, aplicando técnicas de *Machine Learning* y *Deep Learning* para el entrenamiento previo del sistema de reconocimiento facial y la detección de personas. Además, se utilizó el lenguaje de programación *Python* y el *framework Tensor Flow*. Los resultados obtenidos destacaron la eficacia del sistema de vigilancia, permitiendo la detección y reconocimiento de personas, así como la apertura automática de puertas mediante cerraduras magnéticas. Este avance contribuyó de manera significativa a mejorar la seguridad en el centro educativo. En conclusión, el sistema de vigilancia basado en visión por computadora implementado en el Centro de Educación Inicial N°1 del Ministerio de Educación se revela como una solución efectiva y confiable, que garantiza la seguridad del establecimiento y proporciona tranquilidad a la comunidad educativa. (Arellano, 2020).

El artículo de investigación, titulado "*Control de Seguridad y Acceso Vehicular Mediante Visión Artificial*", tuvo como principal objetivo desarrollar un software de seguridad y acceso vehicular que se basara en algoritmos de reconocimiento de caracteres utilizando tecnología de visión artificial. Para llevar a cabo este proyecto, se utilizó una maqueta a escala de la puerta de ingreso número 3 en la UNJFSC, dos cámaras de grabación, la herramienta de adquisición de imágenes de Matlab, sistemas operativos Windows Server y una base de datos MySQL. El enfoque de la investigación fue descriptivo, con un diseño experimental de aplicación práctica. Los resultados obtenidos demostraron una precisión del 94% en la detección del patrón de las placas vehiculares y un 92% de precisión en la lectura de los caracteres en todas las pruebas realizadas. Además, se evaluó cómo afectaba el ángulo de inclinación de la cámara en la eficacia de lectura, encontrando que un ángulo de 35° respecto a la matrícula alcanzaba una eficacia del 88%, mientras que la colocación frontal lograba un 95% de eficacia. En resumen, se concluyó que esta aplicación mejoraba notablemente la seguridad y el control de acceso, reduciendo en un 87% el tiempo de congestión vehicular al ingresar a las instalaciones. Esto sugiere que puede ser implementada con éxito en otras instituciones que enfrenten desafíos similares. (Torres, Ríos, Vega, & Flores, 2020).

En un contexto donde la seguridad y la prevención de robos se han convertido en preocupaciones fundamentales en entornos comerciales, se llevó a cabo la investigación titulada *"Implementación de un sistema de visión artificial para la prevención de robos en el Centro Comercial Mariscal Cáceres, Lima - 2021"*. El objetivo principal de esta investigación fue mejorar la prevención de robos en el mencionado centro comercial mediante la aplicación de un sistema basado en visión artificial. Para lograr esto, se empleó una metodología de enfoque cuantitativo con un diseño experimental pre-experimental. La población de estudio consistió en 90 registros de actividades de seguridad, y se realizó un muestreo censal. La recopilación de datos se llevó a cabo mediante una lista de verificación que fue validada por expertos y sometida a pruebas estadísticas utilizando el *software SPSS*, incluyendo el análisis Alfa de *Cronbach* y la prueba no paramétrica de *Wilcoxon*. Los resultados indicaron que la implementación del sistema de visión artificial mejoró significativamente la prevención de robos en el Centro Comercial Mariscal Cáceres en Lima durante el año 2021, con un nivel de significancia menor al 0.05 en las tres dimensiones que representan la variable dependiente. (Cordova, 2021).

Otro trabajo de investigación tuvo como título *"Desarrollo e Implementación de un Sistema CCTV Antirrobo Inteligente Capaz de Detectar Armas de Manera Eficiente y Bajo Costo al Interior de la Joyería Chavelis de Arequipa usando un Raspberry Pi 4"*. El objetivo principal fue desarrollar e implementar un sistema de videovigilancia que, mediante visión artificial y la *Raspberry Pi 4*, pudiera detectar armas como pistolas o cuchillos en una joyería y tomar decisiones en caso de robo a mano armada, activando un sistema de alarma y notificando del evento a las personas o autoridades correspondientes. Para lograrlo, se aplicó una metodología basada en el reconocimiento de objetos y la comunicación por medio de *WhatsApp*, permitiendo al usuario controlar el sistema de alarma de manera remota. Como resultado, se obtuvo un sistema eficiente de seguridad electrónica que aumentó la eficacia de los sistemas de videovigilancia y ayudó a combatir la delincuencia y la inseguridad ciudadana en una joyería. En conclusión, el sistema desarrollado puede ser aplicado en distintos tipos de negocios, tales como farmacias, mini mercados, tiendas de electrodomésticos, bancos, entre otros, para aumentar su seguridad y proteger sus bienes de valor (PACCO, 2022).

Proceso investigativo metodológico

El proceso metodológico que se utiliza para el diseño de este prototipo es altamente participativo y enfocado en las necesidades de la comunidad. Se proporciona un enfoque de investigación de acción participativa para involucrar a la comunidad en todo el proceso y llevar

a cabo un análisis detallado de los componentes del sistema para asegurarse de que se cumplan los objetivos del proyecto.

El proceso investigativo del proyecto de sistema de videovigilancia comunitaria mediante visión artificial se divide en tres fases: la fase exploratoria, la fase descriptiva y la fase explicativa.

En la fase exploratoria, se realiza un estudio previo sobre las necesidades de seguridad en el barrio y se identifican los problemas existentes en los sistemas de videovigilancia actuales. Se ejecuta una revisión bibliográfica sobre las tecnologías de visión artificial y se determina las soluciones tecnológicas existentes en el mercado.

En la fase descriptiva, se recopilan datos mediante encuestas y entrevistas a los residentes de la comunidad, autoridades locales y expertos en seguridad. Se identifican los puntos críticos y las áreas de mayor incidencia delictiva en el barrio, y se seguirán los requisitos para el sistema de videovigilancia comunitaria.

En la fase explicativa, se realiza una recopilación detallada de los datos en la fase descriptiva y se definen las variables que influyen en la implementación del sistema de videovigilancia comunitaria. También un análisis detallado de los diferentes componentes del sistema de videovigilancia, como las cámaras, el *software* de reconocimiento de objetos y la infraestructura de red. Se evaluaron las opciones disponibles en el mercado y se eligieron los componentes que mejor se adaptan a las necesidades del sistema y al presupuesto disponible.

Posteriormente, se procede a la implementación del prototipo de sistema de videovigilancia comunitaria mediante visión artificial y se realizan pruebas de campo para demostrar su eficacia. Se evalúa la calidad de las imágenes capturadas, la capacidad de reconocimiento de personas y la capacidad de detección de la señal de auxilio en tiempo real.

Finalmente, se realizará una evaluación de la eficacia del sistema de videovigilancia comunitaria mediante visión artificial, comparando los resultados obtenidos con los sistemas de videovigilancia tradicionales. Se concluirá que el sistema de videovigilancia comunitaria mediante visión artificial es una solución tecnológica eficaz y de bajo costo para mejorar la seguridad en las comunidades.

CAPÍTULO II: PROPUESTA

Fundamentos teóricos aplicados

El presente proyecto de investigación se centra en el desarrollo de un sistema de videovigilancia comunitaria mediante visión artificial, con el objetivo de optimizar la seguridad en entornos urbanos y fomentar la participación activa de la comunidad en la prevención del delito. Para lograr este propósito, se aplicarán diversos fundamentos teóricos provenientes de distintas áreas de conocimiento, con el fin de diseñar un sistema eficiente, ético y adaptado a las necesidades específicas de la comunidad.

En esta investigación, se hará uso de conceptos de inteligencia artificial, visión artificial, aprendizaje automático, reconocimiento de patrones y procesamiento de imágenes, entre otros. La visión artificial permitirá dotar al sistema de la capacidad de identificar y analizar señales de auxilio de personas, reconocimiento facial, posibilitando una respuesta rápida y eficaz ante situaciones de riesgo.

Inteligencia Artificial

La inteligencia artificial (IA) es una rama de la ciencia de la computación que busca crear sistemas y programas capaces de realizar tareas que, normalmente, requieren de la inteligencia humana. Estos sistemas están diseñados para aprender de la experiencia, adaptarse al cambio, razonar y resolver problemas, y pueden abarcar áreas tan diversas como el procesamiento de lenguaje natural, la visión por computadora, el reconocimiento de voz, la toma de decisiones, entre otras. La IA ha experimentado un crecimiento significativo en las últimas décadas debido a avances en algoritmos, hardware y el aumento en la disponibilidad de grandes cantidades de datos. Su objetivo es mejorar la eficiencia, la precisión y la automatización de tareas complejas, transformando la forma en que interactuamos con la tecnología y ofreciendo oportunidades para mejorar diversos aspectos de la vida humana. (Russell & Norvig, 2020).

Figura. 1

Inteligencia artificial



Nota: La figura detalla cómo se puede hacer uso de la inteligencia artificial. *Fuente:* (Rouhiainen, 2018)

Visión Artificial

La visión artificial es una rama de la inteligencia artificial que se centra en el desarrollo de algoritmos y técnicas que permiten a las máquinas interpretar y comprender el contenido visual del mundo que les rodea. Esta disciplina se inspira en la forma en que los seres humanos procesan y comprenden la información visual a través de los ojos y el cerebro. Los sistemas de visión artificial utilizan cámaras y sensores para capturar imágenes o vídeos, y luego aplican algoritmos de procesamiento de imágenes y aprendizaje automático para analizar y extraer características relevantes de esas imágenes. Esto puede abarcar diversas tareas, como el reconocimiento de objetos, la detección de personas, el seguimiento de objetos en movimiento, la segmentación de imágenes y la reconstrucción tridimensional del entorno. (Szeliski, 2022)

La visión artificial ha demostrado ser una tecnología poderosa con aplicaciones en diversos campos, incluyendo la robótica, la medicina, la conducción autónoma, la seguridad, la industria manufacturera y el entretenimiento.

La visión artificial ha demostrado ser una tecnología poderosa con aplicaciones en diversos campos, incluyendo la robótica, la medicina, la conducción autónoma, la seguridad, la industria manufacturera y el entretenimiento.

Machine Learning

Machine learning es una rama de la inteligencia artificial que se enfoca en el desarrollo de algoritmos y modelos que permiten a las computadoras aprender de datos y experiencias previas, lo que les permite llevar a cabo tareas específicas sin necesidad de una programación explícita. En lugar de seguir reglas fijas, los sistemas de aprendizaje automático utilizan patrones y relaciones que encuentran en los datos para mejorar su rendimiento y tomar decisiones de manera informada. A través de algoritmos y técnicas estadísticas, estos sistemas pueden identificar patrones complejos en los datos y hacer predicciones precisas.

El aprendizaje automático ha demostrado ser una herramienta poderosa con aplicaciones en una amplia variedad de campos, como el reconocimiento de voz, el procesamiento de imágenes, el análisis de datos, la detección de fraudes, la recomendación de productos y mucho más. Su capacidad para adaptarse a nuevos datos y aprender de experiencias pasadas lo convierte en una tecnología versátil y prometedora para el futuro.

Para el desarrollo del algoritmo se emplea esta técnica de machine learning como es para el reconocimiento facial cuando el sistema detecta una señal de auxilio, permitiendo identificar individuos específicos a partir de características faciales únicas. El implementar el uso de un

modelo para generar alertas personalizadas basadas en el comportamiento registrado, lo que permite tomar medidas rápidas y efectivas.

Deep Learning

El *deep learning* es una rama de la inteligencia artificial (IA) que se enfoca en el desarrollo de algoritmos y modelos de redes neuronales artificiales altamente complejos y capaces de aprender representaciones jerárquicas de datos. Estas redes neuronales se inspiran en la estructura y función del cerebro humano, con múltiples capas de neuronas interconectadas que procesan la información de manera no lineal. (Goodfellow, Bengio, & Courville, 2018)

A diferencia de los enfoques tradicionales de aprendizaje automático, donde los algoritmos dependen en gran medida de la extracción manual de características, el deep learning utiliza técnicas automáticas de extracción y abstracción de características, lo que lo hace especialmente adecuado para grandes conjuntos de datos y tareas complejas.

Las aplicaciones del deep learning son variadas y abarcan diversos campos, como el reconocimiento de voz, el procesamiento de imágenes y vídeo, la traducción automática, el diagnóstico médico, la conducción autónoma, el análisis de redes sociales, entre otros. Su capacidad para realizar tareas complejas y descubrir patrones ocultos en grandes conjuntos de datos lo ha convertido en una herramienta poderosa y revolucionaria en el campo de la inteligencia artificial.

Esta técnica dentro del desarrollo del proyecto se utiliza algoritmos de aprendizaje profundo para aumentar la precisión en la detección de personas que se requiere para mapear cuando una persona realice la acción de levantar los brazos y el sistema alerte de esta situación y se produzca la alerta o notificación.

Redes Neuronales Convolucionales

Las Redes Neuronales Convolucionales (CNN) son un tipo de arquitectura de redes neuronales profundas diseñadas específicamente para tareas de visión por computadora, como el procesamiento de imágenes y el reconocimiento de patrones. Estas redes se inspiran en el funcionamiento del sistema visual humano y utilizan capas convolucionales para extraer características significativas de las imágenes de manera automática. (LeCun, Bengio, & Hinton, 2015)

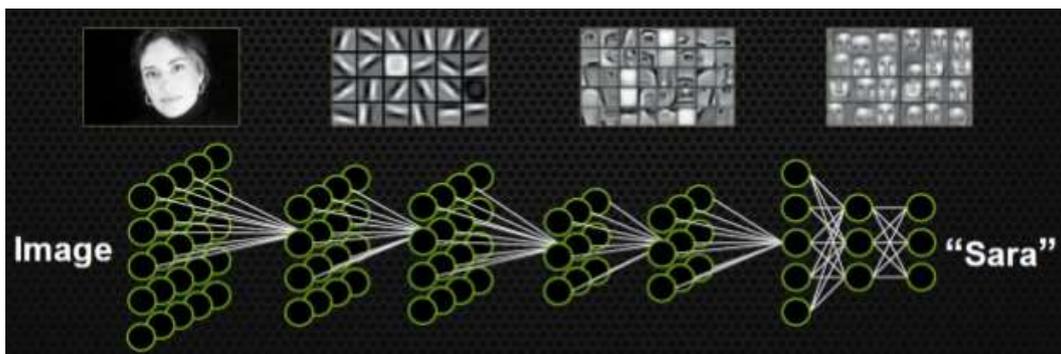
La clave de las CNN radica en la convolución, un proceso que implica la aplicación de filtros o kernels a una imagen para detectar características locales, como bordes, esquinas o texturas. Estos filtros se deslizan sobre toda la imagen para generar mapas de características que resaltan

patrones específicos. Luego, las capas subsiguientes de la red neuronal se encargan de combinar y procesar estas características para realizar tareas más complejas, como el reconocimiento de objetos o el etiquetado de imágenes.

Las redes neuronales convolucionales (CNN) son particularmente útiles para analizar imágenes en detalle y con alta precisión, esta técnica se usa en el desarrollo del algoritmo para segmentar la imagen mediante la cámara y así lograr incluir la detección de las personas.

Figura. 2

Redes neuronales convolucionales



Nota: En la imagen se puede ver como es el uso de un algoritmo mediante machine learning.

Fuente: (Moreno, 2019)

Librerías

OpenCV

OpenCV (Open Source Computer Vision Library) es una biblioteca de código abierto de visión por computadora y procesamiento de imágenes. Fue desarrollada originalmente por Intel y actualmente es mantenida por Willow Garage y Itseez. OpenCV proporciona un conjunto de funciones y algoritmos que permiten a los desarrolladores trabajar con imágenes y videos de manera eficiente. La biblioteca es altamente popular y ampliamente utilizada en la comunidad de visión por computadora debido a su gran cantidad de funcionalidades y su versatilidad.

OpenCV cuenta con implementaciones de algoritmos para tareas comunes de visión por computadora, como detección de objetos, seguimiento de movimiento, reconocimiento facial, calibración de cámaras, segmentación de imágenes, entre otras. La biblioteca está escrita en C++ y cuenta con interfaces para C++, Python y otras plataformas, lo que la hace accesible para desarrolladores de diferentes lenguajes de programación. (OpenCV, 2023)

Gracias a su licencia de código abierto, OpenCV ha sido adoptada en una amplia gama de aplicaciones y proyectos, desde la investigación académica hasta el desarrollo de aplicaciones comerciales en diversas industrias.

Numpy

NumPy es una poderosa biblioteca de código abierto para Python que se utiliza ampliamente en el ámbito de la ciencia de datos, el análisis numérico y la computación científica. Esta biblioteca proporciona estructuras de datos y funciones para realizar operaciones matemáticas y numéricas de manera eficiente y rápida. La característica central de NumPy es el arreglo multidimensional (array), que permite representar datos en forma de matrices y realizar cálculos vectorizados de manera sencilla.

Cuda

CUDA es una potente biblioteca de programación desarrollada por NVIDIA, diseñada específicamente para aprovechar el poder de las unidades de procesamiento gráfico (GPU) en aplicaciones de alto rendimiento y cómputo paralelo. Esta biblioteca permite a los desarrolladores escribir código que se ejecuta de manera eficiente en las GPU, lo que resulta en un aumento significativo en el rendimiento de tareas intensivas en cálculos. Con CUDA, es posible acelerar una amplia variedad de aplicaciones, como procesamiento de imágenes, simulaciones numéricas, aprendizaje automático y más. Al aprovechar la capacidad de paralelismo masivo que ofrecen las GPU modernas, CUDA se ha convertido en una herramienta esencial para aquellos que buscan resolver problemas complejos y realizar cálculos a gran escala con un alto grado de eficiencia y velocidad.

Python

Python es un lenguaje de programación de alto nivel, interpretado y de propósito general. Fue creado en la década de 1990 por Guido van Rossum y desde entonces ha ganado una enorme popularidad en la comunidad de desarrolladores. La filosofía detrás de Python se centra en la legibilidad del código y la simplicidad de la sintaxis, lo que lo convierte en un lenguaje muy accesible para principiantes y a la vez potente y flexible para profesionales. (Foundation, 2021)

Una de las principales ventajas de Python es su amplia biblioteca estándar, que incluye una gran cantidad de módulos y paquetes que facilitan la implementación de diferentes tareas, como el procesamiento de texto, la manipulación de archivos, la programación de redes, el acceso a bases de datos, entre muchas otras. Además, Python es conocido por su comunidad activa y solidaria, lo que ha llevado al desarrollo de numerosas bibliotecas y frameworks de terceros que amplían aún más su funcionalidad y permiten abordar una gran variedad de proyectos.

Python se utiliza en una amplia gama de aplicaciones, desde el desarrollo web y científico, hasta el aprendizaje automático, la automatización de tareas, la inteligencia artificial y más. Su

versatilidad, facilidad de uso y su enfoque en la legibilidad del código lo convierten en una opción popular para proyectos de cualquier escala.

Se utiliza Python como lenguaje de programación para el desarrollo del algoritmo por lo que este es de alto nivel y es muy utilizado para este tipo de algoritmos, siendo una de las tecnologías más empleadas en la actualidad.

Google Drive

Google Drive es un servicio de almacenamiento en la nube proporcionado por Google. Permite a los usuarios almacenar, sincronizar y compartir archivos y documentos en línea, lo que facilita el acceso a la información desde cualquier dispositivo con conexión a internet.

Google Drive se utiliza en el proyecto de la siguiente manera, cuando se emita una alerta de la persona que alzó los brazos para notificar que sufre de algún robo o problema, el sistema capturará la imagen de la persona y la enviará a una carpeta compartida donde se almacenarán estas alertas para tenerlas como evidencias ante cualquier situación que se presente en el barrio.

Edge IA

Edge AI o Inteligencia Artificial en el Borde, está impulsando una nueva era de dispositivos inteligentes y autónomos. En lugar de depender completamente de la nube o servidores remotos, la IA se implementa directamente en los dispositivos en el extremo de la red, lo que permite un procesamiento de datos más rápido y eficiente. Esto ha dado lugar a una serie de aplicaciones transformadoras en diversas industrias. (Electrónica, 2023)

En el ámbito de la seguridad, los sistemas de Edge AI pueden detectar intrusos o incidentes de forma autónoma y enviar alertas en tiempo real, mejorando la vigilancia y la respuesta a situaciones críticas. En la industria, los robots equipados con Edge AI pueden realizar tareas de manera autónoma y adaptarse a cambios en su entorno sin intervención humana, lo que aumenta la eficiencia y la productividad.

Esta tecnología en el borde de la red abre un mundo de posibilidades, mejorando la seguridad, la eficiencia y la precisión en aplicaciones que van desde la seguridad y la salud hasta la industria y el transporte.

Protocolo de comunicación RSTP

El Protocolo RTSP es un estándar de comunicación utilizado en aplicaciones de videovigilancia y streaming de medios. Su función principal es facilitar la transmisión de video y audio en tiempo

real entre una cámara de seguridad u otro dispositivo de captura y un sistema de gestión o visualización de video.

RTSP actúa como un "control remoto" para la reproducción de medios en tiempo real. Permite la solicitud y el control de flujos multimedia, lo que significa que puede iniciar, pausar, detener o adelantar una transmisión de video o audio en vivo. Además, RTSP puede utilizarse para acceder a cámaras de seguridad desde ubicaciones remotas a través de una red, lo que es esencial en aplicaciones de videovigilancia.

El protocolo RTSP funciona en conjunto con otros protocolos, como el Protocolo de Transferencia de Hipertexto (HTTP) para la entrega de datos multimedia. Esto significa que, mediante una solicitud RTSP, se puede obtener la ubicación de un flujo multimedia y luego utilizar HTTP o UDP para recibir los datos de video en tiempo real.

Jetson Nano NVIDIA

La Jetson Nano de NVIDIA es una potente placa de desarrollo de inteligencia artificial diseñada para aplicaciones de computación de borde. Es una opción popular para proyectos de visión artificial, aprendizaje profundo, robótica y otras aplicaciones que requieren un alto rendimiento en un factor de forma compacto. La Jetson Nano cuenta con una GPU de NVIDIA con arquitectura CUDA, lo que la convierte en una plataforma excepcional para acelerar tareas intensivas en cálculos. La Jetson Nano es ideal para aquellos que desean desarrollar e implementar soluciones de inteligencia artificial en el borde, ofreciendo un rendimiento sorprendente en un formato compacto y accesible para estudiantes, desarrolladores y entusiastas por igual.

Figura. 3

Jetson Nano Nvidia



Nota: Tarjeta de desarrollo para desarrollo de proyectos de IA. *Fuente:* (Rehman, Razzaq, & Hussian, 2021)

Ubuntu

Ubuntu es un sistema operativo de código abierto basado en Linux que se ha convertido en una de las distribuciones más populares y ampliamente utilizadas en todo el mundo. Es conocido por su enfoque en la facilidad de uso, la estabilidad y la seguridad, lo que lo hace adecuado tanto para usuarios principiantes como para usuarios avanzados. Ubuntu ofrece una interfaz gráfica intuitiva y amigable, lo que permite a los usuarios acceder y gestionar sus aplicaciones, archivos y configuraciones de manera sencilla. Ubuntu es utilizado en una variedad de entornos, desde computadoras personales y portátiles hasta servidores y dispositivos IoT (Internet de las cosas), lo que demuestra su versatilidad y adaptabilidad en diferentes escenarios.

Descripción de la propuesta

El presente proyecto tiene como finalidad que el sistema combine tecnologías de visión artificial y aprendizaje profundo para monitorear y analizar las imágenes capturadas por las cámaras de seguridad en tiempo real. Está diseñado para detectar y reconocer la presencia de personas en el área vigilada, y específicamente para identificar situaciones de emergencia cuando alguien alza los brazos en señal de auxilio.

Este sistema combina la potencia del procesamiento de imágenes en tiempo real con algoritmos avanzados de aprendizaje profundo para brindar una detección precisa de señales de auxilio en situaciones de seguridad. Su capacidad para identificar señales de auxilio y reaccionar ante emergencias potenciales lo convierte en una herramienta valiosa para la vigilancia y la seguridad en diversos entornos.

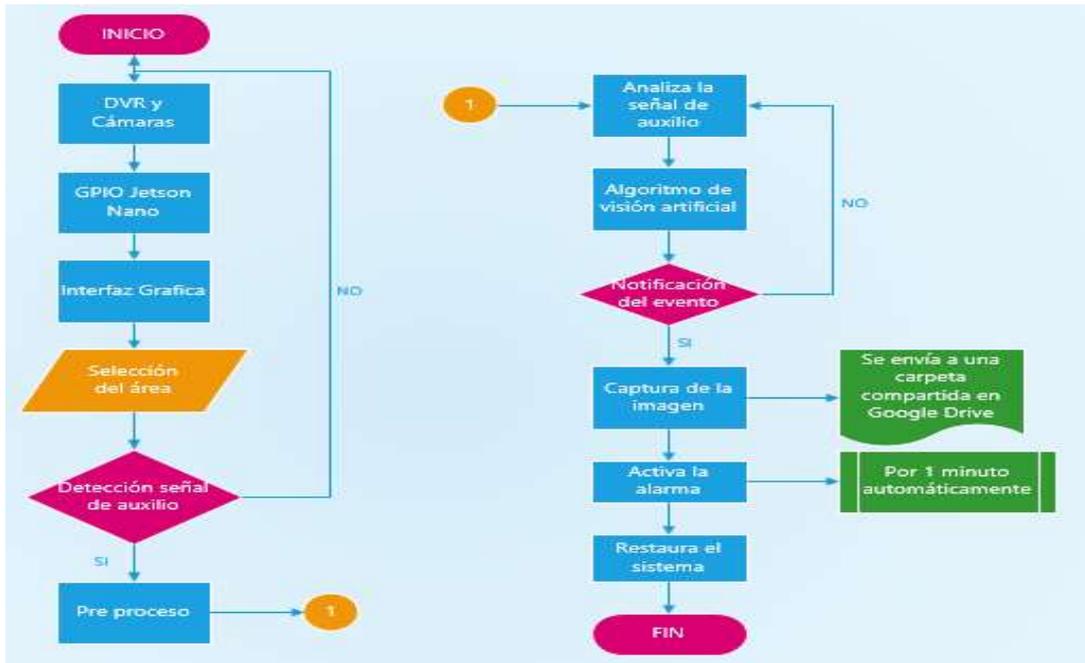
Adicionalmente, este sistema puede estar configurado para correr en una Jetson Nano, aprovechando su capacidad de procesamiento y cómputo en el borde. Las GPIO (Entradas/Salidas de Propósito General) de la Jetson Nano pueden ser utilizadas para activar sirenas o dispositivos de alarma en respuesta a una señal de auxilio detectada.

a. Estructura general

A continuación, se detalla la estructura general del prototipo con el proceso de configuración y funcionamiento del algoritmo de visión artificial en el sistema de videovigilancia comunitaria mediante el siguiente diagrama de flujo.

Figura. 4

Diagrama de flujo del sistema de videovigilancia mediante visión artificial



Nota: Detalle del funcionamiento del sistema de videovigilancia mediante la implementación del algoritmo de visión artificial. Fuente: Elaboración Propia.

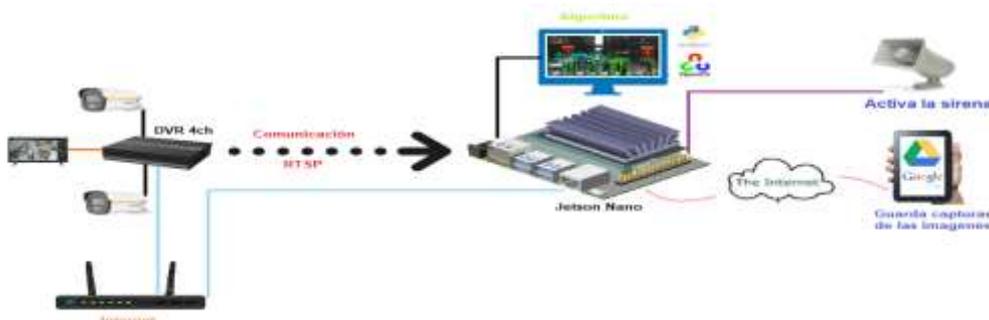
b. Explicación del aporte

El presente proyecto técnico abarca el desarrollo, implementación, configuración, puesta en marcha del sistema de videovigilancia comunitario mediante visión artificial. A continuación, se detalla cómo se implementará el algoritmo de visión artificial en el sistema de videovigilancia.

A continuación, se muestra la arquitectura del sistema de videovigilancia comunitario mediante visión artificial con los diferentes componentes y equipos.

Figura. 5

Arquitectura del Proyecto



Nota: se establece la arquitectura que tendrá el sistema dentro del proyecto. Fuente: elaboración propia.

Para la selección de los equipos en el diseño del proyecto, se optó por tecnologías apropiadas en el ámbito de la visión artificial, que tengan las características técnicas requeridas, sean adecuadas en costo y beneficio que permitan mejorar la calidad de servicio en su instalación e implementación del sistema de seguridad. A continuación, se proporcionan los detalles específicos de estos elementos utilizados en el proyecto.

DVR

Es el dispositivo que se encarga de grabar y digitalizar las imágenes mediante las cámaras, poseen un protocolo de seguridad donde se visualiza y registra todos los acontecimientos que se tengan en el barrio, el grabador que se utiliza para el proyecto es de la marca *hikvision* que se describe a continuación sus características:

Tabla 1

Características técnicas del DVR

Características técnicas	
Entradas Video	4 canales HD
Resolución	Turbo HD 1080P
Detección de Movimiento	Multi zona
Salidas de video	HDMI, VGA
Entrada y Salida	Audio
Puerto IP	RJ45
Interfaz	USB
Manejo de Protocolos	TCP/IP, PPPoE, DHCP, Hik Cloud P2P
Alimentación	12 Vdc
Consumo del Disco	5.4W
Disco	Con tecnología eSSD

Nota: especificaciones técnicas del dvr tomadas de la página oficial del *Hikvision*. *Fuente:* Elaboración Propia.

Figura. 6

DVR eSSD de 4 canales 1080 lite



Nota: imagen del grabador de la nueva serie eSSD de la marca Hikvision. *Fuente:* (Hikvision, 2023)

Cámaras Color VU

Estos dispositivos son de gran importancia, ya que permiten monitorear y son de tipo analógica y digital. Para el desarrollo del proyecto se utilizan las cámaras análogas con las características requeridas en el diseño como es el uso de cámaras de video vigilancia de tipo bullet para exterior con las siguientes características:

Tabla 2

Especificaciones técnicas de las cámaras tipo bullet

Características técnicas	
Consumo Eléctrico	3 W máximo
Resolución	Turbo HD 1080P
Sensor	Turbo HD 2 megapíxeles
Cobertura	30 m
Tipo de Material	Metálico
Sistema	PAL/NTSC
Alimentación	12 Vdc
Día y Noche	Color VU
Protección	IP67

Nota: descripción general de las especificaciones de la cámara hacer utilizada en el proyecto.
Fuente: elaboración propia

Figura. 7

Cámara color VU de tipo bala para exterior



Nota: imagen de la cámara a color las 24 horas para la implementación del proyecto. *Fuente:* (Hikvision, 2023)

Sirena de 30W

Este equipo sirve para emitir una alerta sonora cuando la central de alarma presente alguna anomalía en su funcionamiento, se tiene las características que se detallan:

Tabla 3

Especificaciones técnicas de la sirena audible de 30 w

Características técnicas	
Voltaje	6 – 12 Vdc
Corriente	1100 mA
Nivel de sonido	120 dB
Tono constante	30 W

Nota: descripción de las especificaciones técnicas a tomar en cuenta en la conexión del equipo.

Fuente: elaboración propia.

Figura. 8

Sirena de 30W



Nota: Sirena audible de 30 w cuando se activa el sistema. *Fuente:* elaboración Propia.

Jetson Nano

La Jetson Nano es una potente plataforma informática desarrollada por NVIDIA que se destaca por su capacidad para implementar aplicaciones de inteligencia artificial y aprendizaje automático en dispositivos de tamaño reducido. Equipada con un procesador ARM Cortex-A57 de cuatro núcleos y una GPU NVIDIA Maxwell con 128 núcleos CUDA, la Jetson Nano tiene el rendimiento necesario para tareas de procesamiento intensivo. Además, cuenta con 4 GB de memoria RAM y ofrece múltiples puertos USB, HDMI y Ethernet para la conectividad. Esta plataforma también es compatible con diversas bibliotecas y frameworks de aprendizaje automático, lo que permite a los desarrolladores implementar soluciones innovadoras en campos como la visión por computadora y el procesamiento del lenguaje natural. A continuación, se detallan las especificaciones técnicas de la tarjeta.

Tabla 4

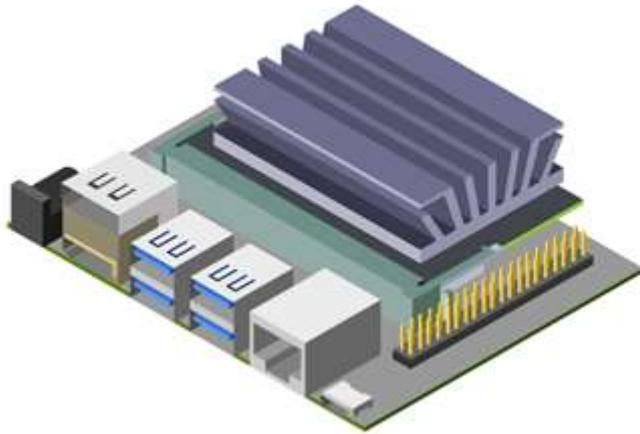
Especificaciones técnicas de la tarjeta jetson nano

	Característica Técnica
Procesador	ARM Cortex-A57 de 4 núcleos a 1.43 GHz
GPU	NVIDIA Maxwell con 128 núcleos CUDA
Memoria RAM	4 GB LPDDR4
Almacenamiento	Ranura para tarjeta microSD (no incluida)
Conectividad	Gigabit Ethernet, WLAN (mediante adaptador USB)
Puertos USB	4 x USB 3.0
Puertos HDMI	1 x HDMI 2.0 (con soporte para 4K a 60 Hz)
GPIO	40 pines GPIO
Cámara	Conector MIPI CSI-2 para cámara
Sistema operativo	Soporte para Linux basado en Ubuntu
Consumo de energía	5V/4A (adaptador de corriente no incluido)

Nota: descripción de las especificaciones para el uso de la tarjeta de desarrollador de la jetson nano. *Fuente:* Elaboración Propia

Figura. 9

Jetson nano Nvidia



Nota: imagen de la tarjeta de la jetson nano Nvidia. *Fuente:* (Franklin, 2019)

A continuación, se describe el desarrollo del algoritmo de visión artificial para el sistema de video vigilancia comunitario de la siguiente manera:

En primer lugar, se realiza la adquisición de imágenes por lo que el sistema se conecta a las cámaras de seguridad ubicadas en diferentes puntos de vigilancia. Estas cámaras capturan continuamente imágenes de vídeo en tiempo real. Esta adquisición se realiza mediante el protocolo de comunicación RTSP que permite visualizar las imágenes del grabador en la tarjeta jetson nano para realizar el siguiente proceso de análisis.

Figura. 10

Adquisición de imágenes del grabador a la jetson nano



Nota: mediante el protocolo RTSP se transmiten las imágenes y vídeo del grabador a la jetson nano para su posterior análisis. *Fuente:* elaboración propia.

Se obtiene la detección de personas utilizando técnicas de detección de objetos y personas basadas en redes neuronales convolucionales (CNN), esta tecnología avanzada se enfoca en procesar imágenes de manera similar a cómo funciona el sistema visual humano. Las redes neuronales convolucionales son capaces de aprender patrones y características distintivas en los datos visuales, lo que permite identificar y localizar personas en las imágenes capturadas por las cámaras de vigilancia. Este enfoque, basado en la detección de objetos y personas, potencia la eficacia del sistema de videovigilancia comunitario al proporcionar la capacidad de identificar personas de manera precisa y en tiempo real. El sistema analiza las imágenes para identificar la presencia de individuos en el campo de visión de las cámaras. Cuando se detecta una persona, se delimita y se realiza un seguimiento en el video.

Figura. 11

Detección de personas mediante técnicas basadas en redes neuronales



Nota: mediante el uso de las redes neuronales se realiza la detección de personas u objetos para el caso del proyecto se realiza la detección de personas. *Fuente:* (Franklin, 2019)

Se efectúa un análisis de posturas el sistema utiliza modelos de aprendizaje profundo para reconocer la postura y posición de las personas detectadas. En particular, se busca la señal de auxilio, que en este caso es cuando una persona alza los brazos.

Figura. 12

Estimación de la postura humana

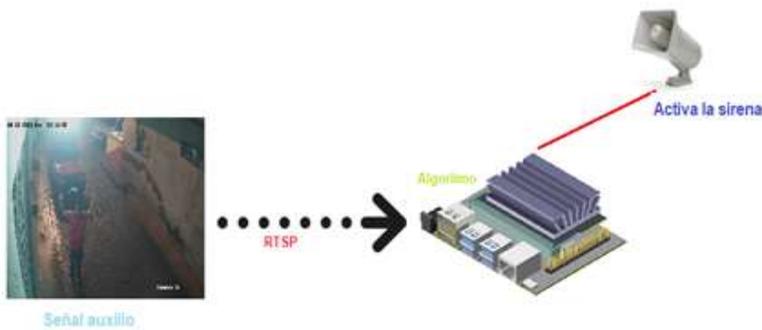


Nota: mediante el uso de la estimación humana y mediante una técnica de aprendizaje profundo se puede analizar esta postura. *Fuente:* (Sivaraman, Shetty, Shah, & Pathak, 2021)

Obteniendo la Detección de señales de auxilio cuando el sistema identifica una persona con los brazos levantados, activa una alerta o notificación en el interfaz de usuario. Esto puede ser una alerta visual en la interfaz de monitoreo, un mensaje de texto o una notificación a un dispositivo móvil.

Figura. 13

Señal de auxilio detectada y activación de la alarma



Nota: cuando una persona alce los brazos la cámara capta esta imagen, se realiza el análisis una vez comprobado esta señal emite una alerta de alarma y captura la imagen y envía al api del google drive. *Fuente:* elaboración propia.

En la interfaz gráfica los operadores de seguridad pueden acceder a una interfaz de usuario que muestra las imágenes en tiempo real de las cámaras, junto con marcadores que indican la ubicación y la postura de las personas detectadas. Cuando se detecta una señal de auxilio, la interfaz resalta la situación de emergencia y permite a los operadores tomar medidas adecuadas.

Figura. 14

Interfaz de usuario creada para seleccionar el área de detección de la señal



Nota: mediante la interfaz gráfica se puede seleccionar el área donde se desea detectar la señal de auxilio para el funcionamiento del sistema. *Fuente:* elaboración propia.

Por último, la capacidad de respuesta en caso de que se detecte una señal de auxilio, el sistema puede estar configurado para activar protocolos de respuesta predefinidos, como notificar a los servicios de emergencia, reproducir mensajes de alerta en altavoces

Además de los pasos que ya mencioné en la descripción anterior, el sistema también puede integrar la capacidad de conectar con Google Drive para guardar imágenes en una carpeta compartida. Una vez que se detecta una señal de auxilio y se activa una alerta, el sistema captura imágenes de la situación y las envía automáticamente a una carpeta designada en Google Drive.

Figura. 15

Api de google drive



Nota: mediante este api el sistema captura la imagen y esta se guarda en una carpeta compartida dentro del google drive. *Fuente:* google drive

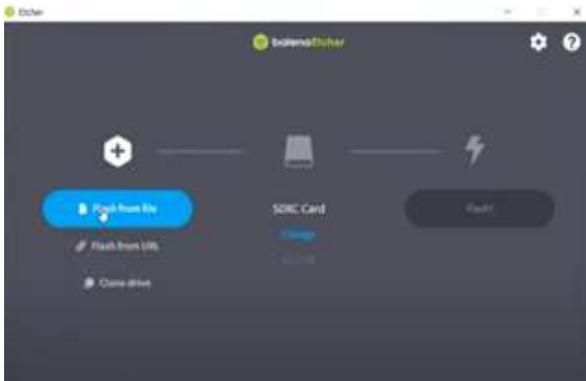
c. Estrategias y/o técnicas

Las técnicas en un sistema de videovigilancia comunitaria mediante visión artificial se centran en el uso de herramientas avanzadas de procesamiento de imágenes y análisis de datos para mejorar la seguridad y el monitoreo en la comunidad. Estas técnicas abarcan desde la detección y reconocimiento de personas hasta la interpretación de patrones de comportamiento de señales de auxilio. A continuación, se detalla estas técnicas y desarrollo del algoritmo.

En primer lugar, se instala el sistema operativo en la tarjeta de desarrollador como es la jetson nano, que en este caso es *Ubuntu*, una vez descargado la imagen del sistema se debe realizar un flasheo de una micro sd para montar el sistema operativo a la tarjeta de la siguiente manera:

Figura. 16

Flasheo en la micro sd



Nota: se realiza el flasheo para instalar el sistema operativo en la tarjeta mediante la micro sd. *Fuente:* elaboración propia.

Una vez realizado el flasheo se procede a instalar el sistema operativo mediante la micro sd, colocando en la ranura de la entrada de la tarjeta para empezar a utilizar la interfaz como se muestra a continuación.

Figura. 17

Interfaz del sistema operativo en la jetson nano



Nota: visualización de la interfaz dentro de la tarjeta jetson nano para su uso. *Fuente:* elaboración propia.

Uso de librerías de Python

Las librerías de Python son conjuntos de recursos que albergan una amplia variedad de código preexistente, destinado a ser reutilizado en la creación de nuevos scripts y desarrollos. Estas librerías consisten en módulos previamente creados y disponibles en línea. La incorporación de estos depósitos de código varía según la distribución específica de Python que se esté empleando. En el marco de este proyecto, se han empleado diversas librerías para alcanzar los objetivos planteados. Entre ellas se encuentran *Gitpython*, *numpy*, *opencv-python*, *cuda*, *nvidia-pyindex* y *ultralytics* entre otras.

Figura. 18

Librerías de Python

```
import sys
import argparse
import cv2
from scipy.spatial import distance
import numpy as np

import numpy as np
from imutils.video import VideoStream
from imutils import face_utils
import cv2

import datetime
import imutils

import sys
import argparse
```

Nota: importación de las librerías principales para el desarrollo del algoritmo. *Fuente:* elaboración propia.

Modelo Ultralytics YoloV8

Ultralytics YOLOv8 representa un modelo de vanguardia, que se ha construido a partir de los logros conseguidos por ediciones previas de YOLO. Este nuevo modelo incorpora innovaciones y mejoras con el propósito de ampliar su rendimiento y versatilidad. YOLOv8 se distingue por su capacidad para ser veloz, preciso y de uso sencillo, lo que lo convierte en una elección excepcional para diversas tareas, como la detección y el seguimiento de objetos, la segmentación de instancias, la clasificación de imágenes y la estimación de poses.

Figura. 19

Modelo ultralytics YoloV8

```
#####
directorio_credenciales = 'credentials_module.json'
##### LECTURA DE MODELO #####
model = YOLO('yolov8x-pose.pt')
```

Nota: instalación del modelo en el entorno de Python. *Fuente:* (GitHub, 2023)

Modelo YoloV8x Pose

La estimación de pose se refiere a la tarea de identificar las posiciones de puntos clave en una imagen, que a menudo se denominan puntos específicos. Estos puntos clave pueden representar diversas partes del objeto, como articulaciones, puntos de referencia u otras características distintivas. Sus ubicaciones se expresan típicamente mediante coordenadas 2D [x, y] o 3D [x, y, visible]. El resultado de un modelo de estimación de pose consiste en un conjunto de puntos que indican los puntos clave del objeto en la imagen, usualmente acompañados de puntuaciones de confianza para cada punto. La estimación de pose es especialmente valiosa

cuando se requiere identificar partes específicas de un objeto en una escena y comprender su disposición relativa.

Figura. 20

Modelos de pose de yoloV8

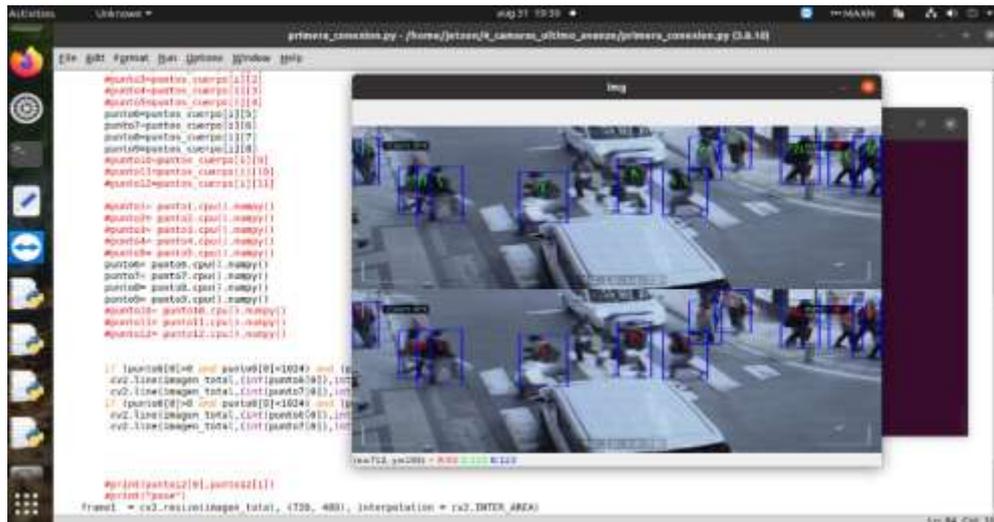
Modelo	tamaño (pixeles)	pose del mapa 50-95	pose del mapa 50	Velocidad CPU ONNX (ms)	Velocidad A100 TensorRT (ms)	parámetros (M)	FLOP (B)
YOLOv8n-pose	640	50,4	80,1	131,8	1,18	3,3	9,2
YOLOv8s-pose	640	60,0	86,2	233,2	1,42	11,6	30,2
YOLOv8m-pose	640	65,0	88,8	456,3	2,00	26,4	81,0
YOLOv8l-pose	640	67,6	90,0	784,5	2,59	44,4	168,6
YOLOv8x-pose	640	69,2	90,2	1607,1	3,73	69,4	263,2
YOLOv8x-pose-p6	1280	71,6	91,2	4088,7	10,04	99,1	1066,4

Nota: existen diferentes modelos de pose de yoloV8. *Fuente:* (cochard, 2021)

Mediante este modelo se procede a identificar la posición de los puntos claves de las personas dentro de la imagen que muestra la cámara para representar las articulaciones que para el proyecto serán los brazos del cuerpo humano de la siguiente manera.

Figura. 21

Modelo yoloV8x pose



Nota: mediante este modelo se obtiene los puntos claves de las personas dentro de la imagen de la cámara. *Fuente:* elaboración propia.

A continuación se muestra el código de la programación del modelo de pose de la estimación de la postura de las personas.

Figura. 22

Programación del modelo pose

```
if point_y_codo_izquierdo < point_y_hombro_izquierdo:
    angulo_izquierda = math.atan(distancia_y_izquierda/distancia_x_izquierda) * ( 180.0 / math.pi )
else:
    angulo_izquierda = 0
if point_y_codo_derecho < point_y_hombro_derecho:
    angulo_derecha = math.atan(distancia_y_derecha/distancia_x_derecha) * ( 180.0 / math.pi )
else:
    angulo_derecha = 0

cv2.line(imagen_total, (int(punto6[0]),int(punto6[1])), (int(punto8[0]),int(punto8[1])), (0,0,255), thickness=2)
cv2.line(imagen_total, (int(punto7[0]),int(punto7[1])), (int(punto9[0]),int(punto9[1])), (0,255,0), thickness=2)

if angulo_izquierda > 15 and angulo_derecha > 15:
    cv2.rectangle(imagen_total, c1, c2, (0,0,255), thickness=2)
    if (punto6[0] > 0 and punto6[0] < 1024) and (punto6[1] > 0 and punto6[1] < 720):
        contador_camara_1 = contador_camara_1 + 1
        #print("imagen1")
        deteccion_area_aux_1 = imagen_total[c1[1]:c1[1]+abs(c1[1]-c2[1]), c1[0]:c1[0]+abs(c1[0]-c2[0])]
        #cv2.imshow("deteccion_area_aux", deteccion_area_aux)
    if (punto6[0] > 0 and punto6[0] < 1024) and (punto6[1] > 720 and punto6[1] < 1440):
        contador_camara_2 = contador_camara_2 + 1
        deteccion_area_aux_2 = imagen_total[c1[1]:c1[1]+abs(c1[1]-c2[1]), c1[0]:c1[0]+abs(c1[0]-c2[0])]
        #print("imagen2")
```

Nota: mediante el uso del modelo pose se realiza la programación con las técnicas adecuadas para realizar la estimación de la postura de las personas. *Fuente:* elaboración propia.

La técnica de visión artificial que se utiliza para recortar áreas basándose en puntos de la interfaz gráfica es una estrategia avanzada que permite la automatización precisa de la segmentación de regiones específicas en imágenes o pantallas. Esta técnica se inicia mediante la detección de puntos de referencia visuales en la interfaz gráfica, como esquinas, bordes o elementos gráficos distintivos. Estos puntos de referencia sirven como marcadores clave que indican las áreas de interés. Una vez que se han identificado estos puntos, se utilizan algoritmos de procesamiento de imágenes para conectarlos y crear una región delimitada. Luego, esta región se recorta de la imagen original, lo que permite aislar y trabajar con precisión en la zona de interés.

Figura. 23

Programación para el recorte del áreas mediante los puntos de la interfaz gráfica

```
for result in results:
    for box, pose in zip(result.bboxes, result.keypoints):
        p=box.xyxy[0]
        c1, c2 = (int(p[0]), int(p[1])), (int(p[2]), int(p[3]))
        cv2.rectangle(imagen_total, c1, c2, (255,0,0), thickness=2)
        for i, k in enumerate(pose):
            #print(k[0])
            puntos_cuerpo=k[0].xy
            punto6=puntos_cuerpo[i][5]
            punto7=puntos_cuerpo[i][6]
            punto8=puntos_cuerpo[i][7]
            punto9=puntos_cuerpo[i][8]

            punto6= punto6.cpu().numpy()
            punto7= punto7.cpu().numpy()
            punto8= punto8.cpu().numpy()
            punto9= punto9.cpu().numpy()

            point_x_hombro_izquierdo=punto6[0]
            point_y_hombro_izquierdo=punto6[1]

            point_x_hombro_derecho=punto7[0]
            point_y_hombro_derecho=punto7[1]

            point_x_codo_izquierdo=punto8[0]
            point_y_codo_izquierdo=punto8[1]
```

Nota: se programa utilizando el box del modelo donde se realizar los marcados de las áreas hacer detectadas para el caso del proyecto. *Fuente:* elaboración propia.

Para realizar la conexión entre las imágenes y videos captadas por las cámaras de videovigilancia desde el grabador a la tarjeta jetson nano se realiza mediante el uso del protocolo de comunicación que es el RTSP y el puerto de comunicación 554, que se programa de la siguiente manera.

Figura. 24

Conexión de las cámaras con la jetson nano

```
p1="rtsp://admin:"
p2="@#"
cam1=":554/Streaming/Channels/102"
cam2=":554/Streaming/Channels/202"
cam3=":554/Streaming/Channels/302"
cam4=":554/Streaming/Channels/402"
direccion_total=p1+clave+p2+direccion_ip+cam1
direccion_total1=p1+clave+p2+direccion_ip+cam2
direccion_total2=p1+clave+p2+direccion_ip+cam3
direccion_total3=p1+clave+p2+direccion_ip+cam4
print(direccion_total3)
longitud=0
print(direccion_total)
try :
    direccion=ping(direccion_ip, verbose=False, count=1, timeout=0.1)
    direccion=str(direccion)
    longitud=len(direccion)
    print("longitudddd", longitud)
except Exception as e:
    longitud=0
    print("longitudddd", longitud)
if longitud>85:
    try:
        camara1_r = VideoStream(direccion_total).start()
        camara11 = camara1_r.read()
        #print(camara11.shape)
```

Nota: como se evidencia mediante el uso del protocolo RTSP y el puerto 554 se configura y programa para recibir la imagen del grabador en tiempo real. *Fuente:* elaboración propia.

A continuación, se realiza la interfaz de usuario que tendrá el sistema de video vigilancia para el monitoreo usando el lenguaje de programación de Python, con bibliotecas gráficas para el desarrollo de la interfaz como se muestra a continuación.

Figura. 25

Configuración de la guide de la interfaz grafica

```
#####  
##### configuracion de la ventana de la guide #####  
mainClock = pygame.time.Clock()  
  
pygame.init()  
pg.init()  
  
##### configuracion de la camara para hacer las capturas #####  
  
color=False#True#False  
  
camera_index = 0  
camera=cv2.VideoCapture(camera_index)  
camera.set(3,640)  
camera.set(4,480)  
  
#####  
  
camera_index_1 = 2  
  
camera_1=cv2.VideoCapture(camera_index_1)  
camera_1.set(3,640)  
camera_1.set(4,480)  
retval,frame_1=camera_1.read()
```

Nota: se realiza la programación de la interfaz gráfica usando librerías de Python. Fuente: elaboración propia.

Figura. 26

Interfaz de usuario del sistema de videovigilancia



Nota: mediante esta interfaz gráfica el usuario o el operador podrá realizar el monitoreo y determinar el área de detección. Fuente: elaboración propia.

Para recibir las notificaciones de las señales de auxilio que detecte el modelo del sistema se lo realiza mediante la aplicación de google drive donde se guardaran las imágenes que se

capturen de este evento, siendo compartida esta carpeta con los diferentes residentes de la comunidad, a continuación, se muestra el código.

Figura. 27

Configuración del Api de google drive

```
##### conexion con drive #####
# INICIAR SESION
def login():
    GoogleAuth.DEFAULT_SETTINGS['client_config_file'] = directorio_credenciales
    gauth = GoogleAuth()
    gauth.LoadCredentialsFile(directorio_credenciales)

    if gauth.credentials is None:
        gauth.LocalWebserverAuth(port_numbers=[8092])
    elif gauth.access_token_expired:
        gauth.Refresh()
    else:
        gauth.Authorize()

    gauth.SaveCredentialsFile(directorio_credenciales)
    credenciales = GoogleDrive(gauth)
    return credenciales

def subir_archivo(ruta_archivo,id_folder):
    credenciales = login()
    archivo = credenciales.CreateFile({'parents': [{"kind": "drive#fileLink",\
                                                "id": id_folder}]})
    archivo['title'] = ruta_archivo.split("/")[-1]
    archivo.SetContentFile(ruta_archivo)
    archivo.Upload()
#####
```

Nota: configuración dentro del programa para el envío y captura de las imágenes a la carpeta de google drive. *Fuente:* elaboración propia.

Finalmente, el sistema de videovigilancia comunitaria con visión artificial utiliza algoritmos avanzados para mejorar la seguridad en el barrio al detectar y responder automáticamente a señales de auxilio ante eventos sospechosos o peligrosos en las imágenes captadas por cámaras de seguridad. Esta tecnología proporciona una capa adicional de protección y vigilancia para los residentes del barrio. A continuación, se muestra la interfaz de usuario terminado con la integración de las señales de las cámaras para su funcionamiento.

Figura. 28

Interfaz gráfica de usuario



Nota: una vez realizada la programación con las bibliotecas gráficas se muestra la interfaz gráfica final del proyecto. *Fuente:* elaboración propia.

Validación de la propuesta

La validación del proyecto se realiza con tres expertos con amplios conocimientos en el área, mediante la evaluación del proyecto. Donde se busca la opinión y la retroalimentación de los profesionales con experiencia y conocimientos relevantes en el campo específico del proyecto. Esta validación busca asegurarse de que el enfoque, la metodología y los resultados del proyecto sean sólidos y estén alineados con las mejores prácticas y el conocimiento existente en el campo.

A continuación, se detallan los perfiles de los profesionales que validan el proyecto.

Tabla 5

Perfiles de profesionales validadores del proyecto

Nombres y Apellidos	Años de experiencia	Titulación Académica	Cargo
Liseth Daniela Albarracín Palma	5	Ingeniera en Electrónica Digital y Telecomunicaciones Master en Industria 4.0	Jefa de Proyectos
Luis Xavier Valarezo Rodríguez	4	Ingeniero Electrónico en Automatización y Control Industrial Especialista en Inteligencia Artificial	Programador
Renato Fernando Córdova Sosa	12	Ingeniero Electrónico en Telecomunicaciones Maestría en tecnologías de la información	Gerente General de la empresa Inn-3

Nota: descripción de los expertos evaluadores del proyecto con su formación y años de experiencia. *Fuente:* elaboración propia.

Durante la validación con expertos, se presenta el proyecto a estos profesionales, se comparten detalles sobre los objetivos, los métodos utilizados y los resultados obtenidos hasta el momento. Los expertos aportan su experiencia y perspectiva, ofreciendo críticas constructivas y sugerencias para mejorar el proyecto. Este proceso es valioso para identificar posibles errores, identificar áreas de mejora y garantizar que el proyecto cumpla con los estándares y expectativas de la comunidad experta en ese campo. En última instancia, la validación con expertos contribuye a la calidad y la credibilidad del proyecto.

Mediante la siguiente tabla que se presenta a continuación, los expertos validaran el proyecto con los distintos criterios de evaluación que son los siguientes.

Tabla 6*Criterios de evaluación de los expertos*

Criterios	Descripción
Impacto	Representa el alcance que tendrá el modelo de gestión y su representatividad en la generación de valor público.
Aplicabilidad	La capacidad de implementación del modelo considerando que los contenidos de la propuesta sean aplicables
Conceptualización	Los componentes de la propuesta tienen como base conceptos y teorías propias de la gestión por resultados de manera sistémica y articulada.
Actualidad	Los contenidos de la propuesta consideran los procedimientos actuales y los cambios científicos y tecnológicos que se producen en la nueva gestión pública.
Calidad Técnica	Miden los atributos cualitativos del contenido de la propuesta.
Factibilidad	Nivel de utilización del modelo propuesto por parte de la Entidad.
Pertinencia	Los contenidos de la propuesta son conducentes, concernientes y convenientes para solucionar el problema planteado.

Nota: mediante distintos criterios se evalúa el proyecto con los expertos. *Fuente:* Universidad Tecnológica Israel maestría en Electrónica y Automatización.

Los expertos validarán el proyecto mediante una escala que se presenta en la siguiente tabla para esta evaluación.

Tabla 7*Escala de evaluación del proyecto*

CRITERIOS	EVALUACIÓN SEGÚN IMPORTANCIA Y REPRESENTATIVIDAD				
	En Total desacuerdo	En Desacuerdo	Ni de Acuerdo Ni en Desacuerdo	De Acuerdo	Totalmente Acuerdo
Impacto					X
Aplicabilidad					X
Conceptualización					X
Actualidad					X
Calidad Técnica					X
Factibilidad					X
Pertinencia					X

Nota: mediante esta tabla se describe la escala de evaluación del proyecto. *Fuente:* Ing. Wilmer Fabián Albarracín Guarochico MBA.

Mediante actas de validación de cada profesional que valida el proyecto estas se encuentran en la parte de anexos al final del documento como evidencia de esta evaluación y validación.

Matriz de articulación de la propuesta

En la presente matriz se sintetiza la articulación del desarrollo del proyecto realizado con los sustentos teóricos, metodológicos, técnicas de visión artificial y herramientas tecnológicas empleadas en el proyecto de sistema de videovigilancia comunitario mediante visión artificial.

Tabla 8

Matriz del proyecto de propuesta y articulación

Ejes o partes principales del proyecto		Breve descripción de los resultados	Sustento teórico que se aplicó en la construcción del proyecto	Metodologías, herramientas técnicas y tecnológicas que se emplearon
1	Definición: de los diversos fundamentos teóricos de visión artificial, técnicas, reconocimiento de señales de auxilio, procesamiento de imagen, lenguaje de programación y elementos electrónicos.	1.1. Identificación de los patrones de señal de auxilio 1.2. Selección de las técnicas de visión artificial y algoritmos. 1.3. Análisis de costos beneficio de los equipos del sistema.	Inteligencia Artificial Visión Artificial Machine Learning Deep learning Python, TensorFlow, Cuda, Numpy	Redes neuronales convolucionales PoseResnet Protocolo RTSP Api de google drive
2	Diseño: adquisición de las imágenes y videos del grabador en tiempo de las cámaras instaladas en el barrio de la cooperativa nuevo Alóag, aplicación del algoritmo de visión artificial, programación.	2.1. Instalación de Ubuntu 2.2. Uso de aplicaciones y librerías de Python. 2.3. Algoritmos de reconocimiento de postura humana. 2.4 Aplicación en Opencv	Programación en Python Aplicaciones de OpenCV	Algoritmo de reconocimiento de señales de auxilio. Postura Humana
3	Implementación: instalación de las cámaras de videovigilancia, cableado, pruebas de funcionamiento y puesta en marcha del sistema de videovigilancia comunitario.	3.1. Compra de cámaras análogas tipo bala color VU 3.2. Sirena de 30W para alarma 3.4 Jetson nano 3.5. Instalaciones eléctricas y electrónicas	Cableado del sistema Instalaciones eléctricas y electrónicas Protocolo de comunicación RTSP	Jetson nano procesador ARM Cortex-A57 de cuatro núcleos y una GPU NVIDIA Maxwell con 128 núcleos CUDA, 4 GB de memoria RAM, 40 pines GPIO

Nota: elaboración de la matriz de articulación y propuesta del proyecto. *Fuente:* Universidad Tecnológica Israel maestría en Electrónica y Automatización.

Análisis de resultados. Presentación y discusión.

En el contexto de nuestra investigación, el análisis de resultados es una fase crítica que permite evaluar el desempeño y la eficacia del sistema de videovigilancia comunitario mediante visión artificial. Durante este proceso, se han recopilado y examinado una cantidad significativa de datos generados por el sistema en funcionamiento. Estos datos incluyen la detección de la señal de auxilio, así como métricas de rendimiento, como la tasa de detección, falsos positivos y la precisión.

La presentación y discusión de estos resultados son fundamentales para validar la utilidad y la eficiencia del sistema. En la sección de presentación, destacamos las principales observaciones y hallazgos, respaldados con tablas que ilustran el comportamiento del sistema en diferentes escenarios. Que se muestra a continuación.

Tabla 9

Datos obtenidos de la primera cámara en monitoreo

Cámara	Escenario	Señales Reales (SR)	Señales Detectadas (SD)	Falsos Positivos (FP)
Pasaje	Día soleado	8	8	0
	Día oscuro	6	5	1
	Noche con poca luz	9	9	0
	Lluvia intensa	6	5	1

Nota: los datos que se obtuvieron de la cámara del pasaje son recolectados de una semana.

Fuente: elaboración propia.

Tabla 10

Datos obtenidos de la segunda cámara en monitoreo

Cámara	Escenario	Señales Reales (SR)	Señales Detectadas (SD)	Falsos Positivos (FP)
Calle principal	Día soleado	15	15	0
	Día oscuro	10	9	1
	Noche con poca luz	10	9	1
	Lluvia intensa	8	7	1

Nota: los datos que se tiene en la tabla son de la calle principal donde se tiene más movimiento de personas en el barrio. *Fuente:* elaboración propia.

A continuación, en la sección de discusión, interpretamos los resultados a la luz de nuestros objetivos de investigación y los comparamos con la literatura relevante. Analizamos los factores que influyen en el rendimiento del sistema y propone posibles mejoras y recomendaciones para futuros desarrollos.

Tabla 11

Resultados de la implementación del sistema

Cámara	Escenario	Señales Reales (SR)	Señales Detectadas (SD)	Falsos Positivos (FP)	Tasa de Detección (TD)	Tasa de Falsos Positivos (TFP)	Precisión (P)
Pasaje	Día soleado	8	8	0	100%	0%	100%
	Día oscuro	6	5	1	83.33%	20%	80%
	Noche con poca luz	9	9	0	100.00%	0%	100.00%
	Lluvia intensa	6	5	1	83.33%	20%	80%
	Promedio				92%	10%	90%

Nota: los resultados obtenidos se los calculo con métricas que miden la funcionalidad del sistema. *Fuente:* elaboración propia.

Tabla 12

Resultados del funcionamiento del sistema

Cámara	Escenario	Señales Reales (SR)	Señales Detectadas (SD)	Falsos Positivos (FP)	Tasa de Detección (TD)	Tasa de Falsos Positivos (TFP)	Precisión (P)
Calle principal	Día soleado	15	15	0	100%	0%	100%
	Día oscuro	10	9	1	90%	11%	89%
	Noche con poca luz	10	9	1	90%	11%	89%
	Lluvia intensa	8	7	1	88%	14%	86%
	Promedio				92%	9%	91%

Nota: calculamos con métricas la funcionalidad que el sistema tiene en el barrio. *Fuente:* elaboración propia.

En última instancia, este análisis y discusión refuerzan la validez y la aplicabilidad de nuestro sistema de videovigilancia comunitario, subrayando su potencial como una herramienta eficaz para mejorar la seguridad y la vigilancia en el barrio.

Los resultados revelan una tasa de detección promedio del 92%, lo que indica la capacidad de nuestro sistema para identificar con precisión la señal de auxilio de la persona. Además, se registró una baja tasa de falsos positivos, en promedio, de aproximadamente el 9.5%, lo que sugiere que el sistema minimiza la alarma innecesaria. Por último, se puede evidenciar que el sistema tiene una precisión en promedio de más del 91%. Estos datos numéricos respaldan la efectividad de la implementación del sistema de videovigilancia mediante visión artificial siendo una herramienta efectiva y prometedora para mejorar la seguridad y la vigilancia en el barrio.

Las imágenes obtenidas para el análisis y discusión de estos resultados se encuentran en la parte de anexos al final del documento.

CONCLUSIONES

En el proceso de definir los parámetros y reglas de análisis en la imagen para la creación del algoritmo de visión artificial, se lograron avances significativos. Se identificaron y seleccionaron criterios clave para la detección como señal de auxilio y las imágenes capturadas. Esto incluyó la definición de postura humana, la identificación de formas y patrones relevantes, y la delimitación de áreas de interés. Estos parámetros proporcionaron una base sólida para el desarrollo del algoritmo.

La implementación del algoritmo de visión artificial utilizando la biblioteca OpenCV y el lenguaje de programación Python fue un hito crucial en el proyecto. Se logró diseñar un algoritmo eficiente y preciso que utilizó los parámetros y reglas definidos previamente para la detección de señales de auxilio en tiempo real. Esto implicó la manipulación de imágenes, la aplicación de técnicas de procesamiento y la optimización del rendimiento del algoritmo.

La implementación exitosa del algoritmo en la tarjeta jetson nano marcó un paso importante hacia la funcionalidad práctica del sistema de videovigilancia comunitaria. Se logró configurar el sistema para procesar y almacenar imágenes de manera eficiente, permitiendo la visualización a través de una interfaz gráfica amigable. Esto facilitó el acceso y la gestión de datos, lo que es esencial para el monitoreo y la seguridad comunitaria.

La instalación física del sistema de videovigilancia en puntos estratégicos de la comunidad con dos cámaras color Vu se llevó a cabo según lo planificado. La colocación estratégica de las cámaras permitió una cobertura efectiva de áreas críticas, asegurando la captura de imágenes relevantes. La integración de las cámaras con la tarjeta jetson nano y el algoritmo de visión artificial se realizó de manera satisfactoria, creando un sistema funcional de vigilancia en tiempo real.

La validación del sistema de videovigilancia fue un proceso crucial para garantizar su robustez y efectividad. Durante las pruebas y evaluaciones, se confirmó que el sistema detectó con precisión las señales de auxilio y otros eventos de seguridad, manteniendo una baja tasa de falsos positivos. Esto respalda la confiabilidad del sistema como una herramienta efectiva para mejorar la seguridad comunitaria. Además, se identificaron áreas de mejora que podrían implementarse en futuras iteraciones del sistema para optimizar aún más su desempeño.

RECOMENDACIONES

Para futuras investigaciones, sería beneficioso profundizar en la adaptación dinámica de los parámetros de análisis de imagen. Esto permitiría al sistema ajustarse automáticamente a diferentes condiciones de iluminación y escenarios, mejorando aún más su capacidad de detección.

Se sugiere investigar la implementación de algoritmos de aprendizaje profundo (deep learning) para la detección de más señales de auxilio. Esto podría aumentar la precisión y la capacidad del sistema para identificar eventos de seguridad en imágenes más complejas.

Para mejorar la accesibilidad y la divulgación de resultados, se debe considerar desarrollar una aplicación móvil que permita a los residentes de la comunidad acceder a la interfaz gráfica y recibir notificaciones en tiempo real sobre el evento de inseguridad que se suscitó.

En investigaciones posteriores, explorar la posibilidad de integrar tecnologías de cámaras más avanzadas, como cámaras de alta resolución y cámaras IP, para mejorar aún más la capacidad de detección del sistema.

Para una divulgación más efectiva y la socialización del proyecto, se debe considerar organizar reuniones en la comunidad para educar a los residentes sobre el uso y los beneficios del sistema de videovigilancia. Esto promoverá una mayor conciencia y participación comunitaria en la seguridad.

Referencias

- Arellano, L. R. (2020). SISTEMA DE VIDEO VIGILANCIA MEDIANTE VISIÓN POR COMPUTADOR PARA EL CENTRO DE EDUCACIÓN INICIAL N°1 DEL MINISTERIO DE EDUCACIÓN. (*Tesis de Ingeniería*). UNIVERSIDAD TECNOLÓGICA ISRAEL, Quito.
<http://repositorio.uisrael.edu.ec/handle/47000/2433>
- cochard, d. (25 de 05 de 2021). *PoseResnet: un modelo de aprendizaje automático de arriba hacia abajo para la estimación de poses*. <https://medium.com/axinc-ai/poseresnet-a-top-down-machine-learning-model-for-skeletal-detection-9454f391ae4d>
- Cordova, J. (2021). *Implementación de un sistema con visión artificial para la prevención de hurtos en el centro comercial Mariscal Cáceres, Lima - 2021*.
https://repositorio.ucv.edu.pe/bitstream/handle/20.500.12692/85147/Cordova_LJL-SD.pdf?sequence=1&isAllowed=y
- Electrónica, V. (2023). *Edge IA*. <https://www.vencoel.com/que-es-edge-ai-y-como-esta-revolucionando-la-industria/>
- FIGUEROA, Y., y ARIAS, L. (2018). DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE SISTEMA DE SEGURIDAD POR VIDEOVIGILANCIA AUTOMÁTICO DENTRO DE UN ÁREA EXTERNA RESTRINGIDA, UTILIZANDO VISIÓN ARTIFICIAL PARA LA BÚSQUEDA, RECONOCIMIENTO, SEGUIMIENTO Y NEUTRALIZACIÓN DE PERSONAS. (*TESIS DE INGENIERIA*). UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE, LATACUNGA.
<http://repositorio.espe.edu.ec/xmlui/bitstream/handle/21000/13961/T-ESPEL-MEC-0123.pdf?sequence=1&isAllowed=y>
- Foundation, P. S. (2021). *Python Language Reference, Version 3.10.0*.
<https://docs.python.org/3/>
- Franklin, D. (18 de 03 de 2019). *Jetson Nano lleva la informática de IA a todos*.
<https://developer.nvidia.com/blog/jetson-nano-ai-computing/>
- GitHub. (2023). *ultralytics*. <https://github.com/ultralytics/ultralytics>
- Goodfellow, I., Bengio, Y., y Courville, A. (2018). *Deep learning*.
<https://doi.org/10.1007/s10710-017-9314-z>
- Hikvision. (2023). *Hikvision*. <https://www.hikvision.com/en/products/Turbo-HD-Products/DVR/edvr-series/ds-e04hghi-b/>
- LeCun, Y., Bengio, Y., y Hinton, G. (28 de 08 de 2015). *Deep learning*.
<https://www.researchgate.net/publication/277411157>
- LUCAS, L. Y. (2019). DISEÑO DE UN SISTEMA DE SEGURIDAD A TRAVÉS DE CÁMARAS DE VIDEOVIGILANCIA PARA EL LABORATORIO 14 DE LA CARRERA DE INGENIERÍA EN COMPUTACIÓN Y REDES. (*Tesis de Ingeniería*). Universidad Estatal del Sur de Manabi, Manabi. <http://repositorio.unesum.edu.ec/bitstream/53000/1595/1/UNESUM-ECU-REDES-2019-41.pdf>

- Moreno, Á. A. (2019). *Clasificación de imágenes usando redes neuronales convolucionales en Python*. <https://idus.us.es/bitstream/handle/11441/89506/TFG-2402-ARTOLA.pdf?sequence=1&isAllowed=y>
- OpenCV. (2023). *Open Source Computer Vision Library*. <https://opencv.org/>
- PACCO, J. (2022). DESARROLLO E IMPLEMENTACIÓN DE UN SISTEMA CCTV ANTIRROBO INTELIGENTE CAPAZ DE DETECTAR ARMAS DE MANERA EFICIENTE Y A BAJO COSTO AL INTERIOR DE LA JOYERÍA CHAVELIS DE AREQUIPA USANDO UN RASPBERRY PI 4. (*Tesis de Ingeniería*). UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA, AREQUIPA – PERÚ.
<http://repositorio.unsa.edu.pe/bitstream/handle/20.500.12773/14858/IEpaenja.pdf?sequence=1&isAllowed=y>
- Rehman, S. U., Razzaq, M. R., y Hussian, M. H. (31 de Mayo de 2021). *TRAINING OF SSD(SINGLE SHOT DETECTOR) FOR FACIAL DETECTION USING NVIDIA JETSON NANO*.
<https://arxiv.org/pdf/2105.13906.pdf>
- Rouhiainen, L. (2018). *Inteligencia artificial*.
https://static0planetadelibroscom.cdnstatics.com/libros_contenido_extra/40/39308_inteligencia_artificial.pdf
- Russell, S. J., y Norvig, P. (2020). *Artificial Intelligence A Modern Approach* (Vol. 4th edición). Person.
- Sivaraman, S., Shetty, R., Shah, C., y Pathak, N. (21 de 06 de 2021). *Entrenamiento y optimización de un modelo de estimación de pose 2D con NVIDIA TAO Toolkit, Parte 1*.
<https://developer.nvidia.com/blog/training-optimizing-2d-pose-estimation-model-with-tao-toolkit-part-1/>
- Szeliski, R. (2022). *Computer Vision* (Vol. Second Edition).
<https://link.springer.com/bookseries/3191>
- Torres, J. C., Ríos, M. S., Vega, H. A., y Flores, E. E. (2020). Control de Seguridad y Acceso Vehicular, Mediante Visión Artificial. *Big Bang Faustiniiano*, 41.
<file:///C:/Users/USUARIO/Downloads/615-Texto%20del%20art%C3%ADculo-1612-1-10-20210203.pdf>

ANEXOS

A continuación, se presenta los anexos del proyecto.

ANEXO 1

VALIDACIÓN DEL PROYECTO CON PROFESIONALES EXPERTOS



Quito D.M., 25 de agosto del 2023

Yo, **Liseth Daniela Albarracín Palma**, con C.I 0503469785, en mi calidad de validador de la propuesta del proyecto titulado: **Sistema de videovigilancia comunitario mediante visión artificial**.

Elaborado por el Ing. **Edwin Santiago Sarabia Velásquez**, con C.I 1718186115, estudiante de la Maestría en Electrónica y Automatización de la Universidad Tecnológica Israel (UISRAEL), como parte de los requisitos para obtener el Título de Magister, me permito declarar haber revisado el proyecto y realizado la evaluación de criterios.

EVALUACIÓN SEGÚN IMPORTANCIA Y REPRESENTATIVIDAD

CRITERIOS	En Total desacuerdo	En Desacuerdo	Ni de Acuerdo Ni en Desacuerdo	De Acuerdo	Totalmente Acuerdo
Impacto					X
Aplicabilidad					X
Conceptualización					X
Actualidad					X
Calidad Técnica					X
Factibilidad					X
Pertinencia					X



Presado digitalizado por:
**LISETH DANIELA
ALBARRACIN
PALMA**

Mg. Liseth Daniela Albarracín Palma

C.I 0503469785



Quito D.M., 25 de agosto del 2023

Yo, Renato Fernando Córdova Sosa, con C.I 1715466811, en mi calidad de validador de la propuesta del proyecto titulado: Sistema de videovigilancia comunitario mediante visión artificial.

Elaborado por el Ing. Edwin Santiago Sarabia Velásquez, con C.I 1718186115, estudiante de la Maestría en Electrónica y Automatización de la Universidad Tecnológica Israel (UISRAEL), como parte de los requisitos para obtener el Título de Magister, me permito declarar haber revisado el proyecto y realizado la evaluación de criterios.

EVALUACIÓN SEGÚN IMPORTANCIA Y REPRESENTATIVIDAD					
CRITERIOS	En Total desacuerdo	En Desacuerdo	Ni de Acuerdo Ni en Desacuerdo	De Acuerdo	Totalmente Acuerdo
Impacto				X	
Aplicabilidad				X	
Conceptualización					X
Actualidad					X
Calidad Técnica					X
Factibilidad					X
Pertinencia					X



FIRMA ELECTRÓNICAMENTE POR:
RENATO FERNANDO
CORDOVA SOSA

Ing. Renato Córdova Sosa

C.I 1715466811



Quito D.M., 29 de agosto del 2023

Yo, Luis Xavier Valarezo Rodríguez, con C.I 0706611241, en mi calidad de validador de la propuesta del proyecto titulado: Sistema de videovigilancia comunitario mediante visión artificial.

Elaborado por el Ing. Edwin Santiago Sarabia Velásquez, con C.I 1718186115, estudiante de la Maestría en Electrónica y Automatización de la Universidad Tecnológica Israel (UISRAEL), como parte de los requisitos para obtener el Título de Magister, me permito declarar haber revisado el proyecto y realizado la evaluación de criterios.

EVALUACIÓN SEGÚN IMPORTANCIA Y REPRESENTATIVIDAD					
CRITERIOS	En Total desacuerdo	En Desacuerdo	Ni de Acuerdo Ni en Desacuerdo	De Acuerdo	Totalmente Acuerdo
Impacto					X
Aplicabilidad					X
Conceptualización					X
Actualidad					X
Calidad Técnica					X
Factibilidad					X
Pertinencia					X

Ing. Luis Xavier Valarezo Rodríguez

C.I 0706611241

ANEXO 2

CÓDIGO DEL ALGORITMO DE VISIÓN ARTIFICIAL PARA EL SISTEMA DE VIDEOVIGILANCIA

```
import pygame
from pygame.locals import *
import cv2

import pygame as pg
import numpy as np
import serial, time
import datetime

import imutils
import math
from scipy.spatial import distance
import pygame

import sqlite3, datetime

import multiprocessing
import time

import cv2
import threading
import numpy as np
import os
import pygame
from imutils.video import VideoStream
from imutils import face_utils

import sys
import argparse
import cv2
from scipy.spatial import distance
from pythonping import ping
import math
from scipy.spatial import distance

#####

from pydrive2.auth import GoogleAuth
from pydrive2.drive import GoogleDrive
from pydrive2.files import FileNotUploadedError
from pydrive2.auth import GoogleAuth
from pydrive2.drive import GoogleDrive
from pydrive2.files import FileNotUploadedError
from datetime import timedelta
import datetime
import RPi.GPIO as GPIO
import time
#####

from ultralytics import YOLO
import time
import imageio
from ultralytics.yolo.engine.results import Results
from ultralytics.yolo.utils import DEFAULT_CFG, ROOT, ops
from ultralytics.yolo.v8.detect.predict import DetectionPredictor
import numpy as np

##### configuracion del relay #####

output_pin = 18 # BCM pin 18, BOARD pin 12 real

GPIO.setmode(GPIO.BCM)

GPIO.setup(output_pin, GPIO.OUT, initial=GPIO.LOW)

#####

directorio_credenciales = 'credentials_module.json'
```

```

##### conexion con drive #####
# INICIAR SESION
def login():
    GoogleAuth.DEFAULT_SETTINGS['client_config_file'] = directorio_credenciales
    gauth = GoogleAuth()
    gauth.LoadCredentialsFile(directorio_credenciales)

    if gauth.credentials is None:
        gauth.LocalWebserverAuth(port_numbers=[8092])
    elif gauth.access_token_expired:
        gauth.Refresh()
    else:
        gauth.Authorize()

    gauth.SaveCredentialsFile(directorio_credenciales)
    credenciales = GoogleDrive(gauth)
    return credenciales

def subir_archivo(ruta_archivo,id_folder):
    credenciales = login()
    archivo = credenciales.CreateFile({'parents': [{"kind": "drive#fileLink",\
        "id": id_folder}]})

    archivo['title'] = ruta_archivo.split("/")[-1]
    archivo.SetContentFile(ruta_archivo)
    archivo.Upload()
#####

#####
def conexion_rtsp_inicial(clave,direccion_ip):

    print("iniciando funcion")
    conexion_r=0

#####

def obtener_tiempo():
    tiempo_fin=datetime.datetime.now()

    tiempo_fin = tiempo_fin - timedelta(hours=0)

    tiempo_h_m_s=tiempo_fin.strftime("%H:%M:%S")
    tiempo_y_m_d=tiempo_fin.strftime("%Y:%m:%d")

    resta=tiempo_h_m_s.split(':')

    horas=str(resta[0])
    minutos=str(resta[1])
    segundos=str(resta[2])

    # horas=float(horas)
    # minutos=float(minutos)
    segundos=float(segundos)

    # horas=int(minutos)
    # minutos=int(minutos)
    segundos=int(segundos)

    segundos=str(segundos)

    resta=tiempo_y_m_d.split(':')

    año=str(resta[0])
    mes=str(resta[1])
    dia=str(resta[2])

    tiempo_activacion=año+"-"+mes+"-"+dia+"-----"+horas+"-"+minutos
    fecha=tiempo_activacion+".jpg"
    return fecha

#####

```

```

def calibrar_clik_off():
    global click_c
    click_c= 0

def salir_interfaz():
    pygame.quit()

    cv2.destroyAllWindows()

##### VARIABLES DE LADDER #####
etapa_vision=0
etapa_deteccion_vision=0
etapa_posicion_cero=0
etapa_posicion_cero_1=0
cont_deteccion_vision=0
etapa_posicion_cero_2=0

sensor_vision_artificial=0
sensor_posicion_cero_banda=0
sensor_posicion_cero_banda_despachado=0
tiempo_inicio_vision=datetime.datetime.now()
tiempo_fin_vision=datetime.datetime.now()
control_cubiculo_lleno=0

control_cubiculo_lleno=0

contador_cubiculos_rojo=0
contador_cubiculos_verde=0
contador_cubiculos_azules=0

#####

def visual_camara_ip_1(frame,screen):

    frame=np.rot90(frame)
    frame_espejo=np.flip(frame, axis=0)

    frame_espejo=cv2.cvtColor(frame_espejo,cv2.COLOR_RGB2BGR)
    frame=pygame.surfarray.make_surface(frame_espejo)
    screen.blit(frame,(30,230))
    return screen

def visual_camara_ip_1_area(frame,screen):
    frame= cv2.resize(frame, (170, 120), interpolation = cv2.INTER_AREA)
    frame=np.rot90(frame)
    frame_espejo=np.flip(frame, axis=0)
    frame_espejo=cv2.cvtColor(frame_espejo,cv2.COLOR_RGB2BGR)

    frame=pygame.surfarray.make_surface(frame_espejo)
    screen.blit(frame,(470,245))
    return screen

##### CAMARA 1 RSOTROS #####

def visual_camara_ip_1_rostro_1(frame,screen):

    frame=np.rot90(frame)
    frame_espejo=np.flip(frame, axis=0)

    frame_espejo=cv2.cvtColor(frame_espejo,cv2.COLOR_RGB2BGR)
    frame=pygame.surfarray.make_surface(frame_espejo)
    screen.blit(frame,(525,420))
    return screen

```

```

##### TITULO ESPOCH COORDENADAS ###

coordenada_x_espoch=250
coordenada_y_espoch=50

color_espoch=(25, 170, 65)

coordenada_x_espoch_logo=150
coordenada_y_espoch_logo=50

##### TITULO FACULTAD #####

coordenada_x_facultad=320
coordenada_y_facultad=85

color_facultad=(200, 55, 16)

coordenada_x_facultad_logo=1050
coordenada_y_facultad_logo=40

##### TITULO DE PROYECTO #####

coordenada_x_titulo=260
coordenada_y_titulo=120

color_titulo=(46, 65, 188)

#####

#####
#####

rect_area= pg.Rect(1050, 400, 520, 480)

rect_rojo= pg.Rect(1440, 415, 80, 30)
rect_verde= pg.Rect(1440, 450, 80, 30)

```

ANEXO 3

DATASHEET CÁMARA COLOR VU TIPO BALA



DS-2CE10DF0T-FS
2MP ColorVu Audio Fixed Mini Bullet Camera











- High quality imaging with 2 MP, 1920 × 1080 resolution
- 24/7 color imaging with F1.0 aperture
- 3D DNR technology delivers clean and sharp images
- 2.8 mm, 3.6 mm fixed focal lens
- Up to 20 m white light distance for bright night imaging
- High quality audio with audio over coaxial cable, built-in mic
- One port for four switchable signals (TVI/AHD/CVI/CVBS)
- Water and dust resistant (IP67)

• Specification

Camera	
Image Sensor	2 MP CMOS
Max. Resolution	1920 (H) × 1080 (V)
Min. Illumination	0.001 Lux @ (F1.0, AGC ON), 0 Lux with white light
Shutter Time	PAL: 1/25 s to 1/50,000 s; NTSC: 1/30 s to 1/50,000 s
Angle Adjustment	Pan: 0° to 360°, Tilt: 0° to 90°, Rotate: 0° to 360°
Signal System	PAL/NTSC
Lens	
Lens Type	2.8 mm, 3.6 mm fixed focal lens
Focal Length & FOV	2.8 mm, horizontal FOV: 105°, vertical FOV: 55°, diagonal FOV: 125° 3.6 mm, horizontal FOV: 84°, vertical FOV: 45°, diagonal FOV: 99°
Lens Mount	M12
Illuminator	
Supplement Light Type	White Light
Supplement Light Range	Up to 20 m
Smart Supplement Light	Yes
Image	
Image Parameters Switch	STD/HIGH-SAT/HIGHLIGHT
Image Settings	Brightness, Sharpness, Anti-banding, AGC
Frame Rate	TVI: 1080p @25 fps/30 fps CVI: 1080p @25 fps/30 fps AHD: 1080p @25 fps/30 fps CVBS: PAL/NTSC
Day/Night Mode	Color
Wide Dynamic Range (WDR)	Digital WDR
Image Enhancement	DWDR, BLC, HLC, Global
Noise Reduction	3D DNR/2D DNR
White Balance	Auto, Manual
Interface	
Video Output	Switchable TVI/AHD/CVI/CVBS
Alarm	Solid Light Alarm (White Light)
General	
Material	Front cover: Plastic, Main body: Metal
Power	12 VDC ± 25%, max. 2 W, *You are recommended to use one power adapter to supply the power for one camera.
Dimension	161.1 mm × 68.4 mm [6.34" × 2.69"]
Weight	Approx. 380 g (0.84 lb.)
Operating Condition	-40 °C to 60 °C (-40 °F to 140 °F). Humidity: 90% or less (non-condensing)
Communication	HIKVISION-C
Language	English
Approval	
Protection	IP67

ANEXO 4

DATASHEET DVR 4 CANALES

HIKVISION

DS-E04HGHI-B 4-ch 1080p Lite 1U H.265 eSSD DVR

Key Feature

- Powered by eSSD technology
- Deep learning based human and vehicle targets classification of Motion Detection 2.0
- Scene-adaptive bitrate control video compression
- Up to 1080p Lite@30 fps encoding capability
- Low power consumption
- Audio via coaxial cable



• Specification

Motion Detection 2.0	
Human/Vehicle Analysis	Deep learning-based motion detection 2.0 is enabled by default for all analog channels. It can classify human and vehicle, and extremely reduce false alarms caused by objects like leaves and lights; Quick search by object or event type is supported.
Recording	
Video Compression	H.265 Pro/H.265
Encoding Resolution	1080p Lite/720p/720p Lite/WD1/4CIF/CIF/QVGA
Frame Rate	Main stream: 1080p Lite/720p/720p Lite/WD1/4CIF@25 fps (P)/30 fps (N) Note: 1080p Lite/720p is 15 fps by default Sub-stream: 4CIF@15 fps; CIF/QVGA@25 fps (P)/30 fps (N)
Video Bitrate	According to the actual scene, scene-adaptive bitrate control technology would automatically allocates appropriate bitrate to ensure the stability of recording period.
Stream Type	Video, Video & Audio Note: In order to protect privacy, the device will only record video (without audio) by default. If you want to record audio, you shall manually enable audio recording. For operation details, refer to the user manual.
Audio Compression	G.711u
Audio Bitrate	64 Kbps
Video and Audio	
IP Video Input	1-ch Up to 1080p resolution Support H.265+/H.265 IP cameras
Analog Video Input	4-ch BNC interface (1.0 Vp-p, 75 Ω), supporting coaxitron connection
HDTVI Input	1080p25, 1080p30, 720p25, 720p30
AHD Input	1080p25, 1080p30, 720p25, 720p30
HDCVI Input	1080p25, 1080p30, 720p25, 720p30
CVBS Input	Support
HDMI Output	1-ch, 1920 × 1080/60Hz, 1280 × 1024/60Hz, 1280 × 720/60Hz
VGA Output	1-ch, 1920 × 1080/60Hz, 1280 × 1024/60Hz, 1280 × 720/60Hz
Video Output Mode	HDMI/VGA simultaneous output

ANEXO 5

DATASHEET JETSON NANO NVIDIA

DATA SHEET

NVIDIA Jetson Nano System-on-Module

Maxwell GPU + ARM Cortex-A57 + 4GB LPDDR4 + 16GB eMMC

Maxwell GPU¹

128-core GPU | End-to-end lossless compression | Tile Caching | OpenGL[®] 4.6 | OpenGL ES 3.2 | Vulkan™ 1.1 | CUDA[®] | OpenGL ES Shader Performance (up to): 512 GFLOPS (FP16) | Maximum Operating Frequency: 921MHz

CPU

ARM[®] Cortex[®] -A57 MPCore (Quad-Core) Processor with NEON Technology | L1 Cache: 48KB L1 instruction cache (I-cache) per core; 32KB L1 data cache (D-cache) per core | L2 Unified Cache: 2MB | Maximum Operating Frequency: 1.43GHz

Audio

Industry standard High Definition Audio (HDA) controller provides a multichannel audio path to the HDMI interface.

Memory

Dual Channel | System MMU | Memory Type: 4ch x 16-bit LPDDR4 | Maximum Memory Bus Frequency: 1600MHz | Peak Bandwidth: 25.6 GB/s | Memory Capacity: 4GB

Storage

eMMC 5.1 Flash Storage | Bus Width: 8-bit | Maximum Bus Frequency: 200MHz (HS400) | Storage Capacity: 16GB

Boot Sources

eMMC and USB (recovery mode)

Networking

10/100/1000 BASE-T Ethernet | Media Access Controller (MAC)

Imaging

Dedicated RAW to YUV processing engines process up to 1400Mpix/s (up to 24MP sensor) | MIPI CSI 2.0 up to 1.5Gbps (per lane) | Support for x4 and x2 configurations (up to four active streams).

Operating Requirements

Temperature Range (T_j): -25 – 97C° | Module Power: 5 – 10W | Power Input: 5.0V

Display Controller

Two independent display controllers support DSI, HDMI, DP, eDP: MIPI-DSI (1.5Gbps/lane): Single x2 lane | Maximum Resolution: 1920x960 at 60Hz (up to 24bpp) | HDMI 2.0a/b (up to 6Gbps) | DP 1.2a (HBR2 5.4 Gbps) | eDP 1.4 (HBR2 5.4Gbps) | Maximum Resolution (DP/eDP/HDMI): 3840 x 2160 at 60Hz (up to 24bpp)

Clocks

System clock: 38.4MHz | Sleep clock: 32.768kHz | Dynamic clock scaling and clock source selection

Multi-Stream HD Video and JPEG

Video Decode

H.265 (Main, Main 10): 2160p 60fps | 1080p 240fps
H.264 (BP/MP/HP/Stereo SEI half-res): 2160p 60fps | 1080p 240fps
H.264 (MVC Stereo per view): 2160p 30fps | 1080p 120fps
VP9 (Profile 0, 8-bit): 2160p 60fps | 1080p 240fps
VP8: 2160p 60fps | 1080p 240fps
VC-1 (Simple, Main, Advanced): 1080p 120fps | 1080i 240fps
MPEG-2 (Main): 2160p 60fps | 1080p 240fps | 1080i 240fps

Video Encode

H.265: 2160p 30fps | 1080p 120fps
H.264 (BP/MP/HP): 2160p 30fps | 1080p 120fps
H.264 (MVC Stereo per view): 1440p 30fps | 1080p 60fps
VP8: 2160p 30fps | 1080p 120fps

JPEG (Decode and Encode): 600 MP/s

Peripheral Interfaces

xHCI host controller with integrated PHY: 1 x USB 3.0, 3 x USB 2.0 | USB 3.0 device controller with integrated PHY | EHCI controller with embedded hub for USB 2.0 | 4-lane PCIe: one x1/2/4 controller | single SD/MMC controller (supporting SDIO 4.0, SD HOST 4.0) | 3 x UART | 2 x SPI | 4 x I2C | 2 x I2S: support I2S, RJM, LJM, PCM, TDM (multi-slot mode) | GPIOs

Mechanical

Module Size: 69.6 mm x 45 mm | PCB: 8L HDI | Connector: 260 pin SO-DIMM

ANEXO 6

PRUEBAS DE FUNCIONAMIENTO DEL PROYECTO





