



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO**

TÍTULO

DISEÑO DE UN PROTOTIPO DE SEGURIDAD CON INTELIGENCIA
ARTIFICIAL PARA MONITOREAR EL INGRESO DE VEHÍCULOS A LA
EMPRESA CBVISION SUCURSAL SALCEDO.

AUTOR

Guanoluisa Huertas, Edwin Eduardo

TRABAJO DE TITULACIÓN

Previo a la obtención del grado académico en
MAGÍSTER EN ELECTRÓNICA Y AUTOMATIZACIÓN

TUTOR

Morales Escobar, Luis Alberto

Santa Elena, Ecuador

Año 2025



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO
TRIBUNAL DE SUSTENTACIÓN**

**Ing. Alicia Andrade Vera, Mgtr.
COORDINADORA DEL
PROGRAMA**

**Ing. Luis Morales Escobar, Ph. D.
TUTOR**

**Ing. Luis Chuquimarca Jiménez, Mgtr.
DOCENTE
ESPECIALISTA**

**Ing, Junior Figueroa Olmedo, Mgtr.
DOCENTE
ESPECIALISTA**

**Abg. María Rivera González, Mgtr.
SECRETARIA GENERAL
UPSE**



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO**

CERTIFICACIÓN

Certifico que luego de haber dirigido científica y técnicamente el desarrollo y estructura final del trabajo, este cumple y se ajusta a los estándares académicos, razón por el cual apruebo en todas sus partes el presente trabajo de titulación que fue realizado en su totalidad por GUANOLUISA HUERTAS EDWIN EDUARDO, como requerimiento para la obtención del título de Magíster en Electrónica y Automatización.

TUTOR

Ing. Luis Alberto Morales Escobar, Ph. D.

Santa Elena, 14 de febrero del 2025



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO**

DECLARACIÓN DE RESPONSABILIDAD

Yo, **Guanoluisa Huertas Edwin Eduardo**

DECLARO QUE:

El trabajo de Titulación, DISEÑO DE UN PROTOTIPO DE SEGURIDAD CON INTELIGENCIA ARTIFICIAL PARA MONITOREAR EL INGRESO DE VEHÍCULOS A LA EMPRESA CBVISION SUCURSAL SALCEDO, previo a la obtención del título en Magister en Electrónica y Automatización, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Santa Elena, 14 de febrero del 2025

EL AUTOR

Edwin Eduardo Guanoluisa Huertas



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE CIENCIAS DE LA INGENIERÍA
INSTITUTO DE POSTGRADO**

CERTIFICACIÓN DE ANTIPLAGIO

Certifico que después de revisar el documento final del trabajo de titulación denominado DISEÑO DE UN PROTOTIPO DE SEGURIDAD CON INTELIGENCIA ARTIFICIAL PARA MONITOREAR EL INGRESO DE VEHÍCULOS A LA EMPRESA CBVISION SUCURSAL SALCEDO, presentado por el estudiante, GUANOLUISA HUERTAS EDWIN EDUARDO, fue enviado al Sistema Antiplagio COMPILATIO, presentando un porcentaje de similitud correspondiente al 8%, por lo que se aprueba el trabajo para que continúe con el proceso de titulación.



TUTOR

Ing. Luis Alberto Morales Escobar, Ph. D.



**UNIVERSIDAD ESTATAL PENÍNSULA
DE SANTA ELENA
FACULTAD DE SISTEMAS Y TELECOMUNICACIONES
INSTITUTO DE POSTGRADO**

AUTORIZACIÓN

Yo, Guanoluisa Huertas Edwin Eduardo

Autorizo a la Universidad Estatal Península de Santa Elena, para que haga de este trabajo de titulación o parte de él, un documento disponible para su lectura consulta y procesos de investigación, según las normas de la Institución.

Cedo los derechos en línea patrimoniales del proyecto de titulación con componentes de investigación aplicada y/o desarrollo, además apruebo la reproducción de este proyecto de titulación dentro de las regulaciones de la Universidad, siempre y cuando esta reproducción no suponga una ganancia económica y se realice respetando mis derechos de autor

Santa Elena, 14 de febrero del 2025

EL AUTOR

Edwin Eduardo Guanoluisa Huertas

AGRADECIMIENTO

Agradezco a Dios y al Príncipe San Miguel por sus infinitas bendiciones, guiarme siempre por el camino del bien y permitirme alcanzar un objetivo más en mi vida.

A la Universidad Península de Santa Elena por abrirme las puertas y dejarme ser parte de esta formación académica y fortalecer mis conocimientos profesionales.

A mi tutor Ph. D. Luis Morales por sus enseñanzas impartidas a lo largo de este proyecto de titulación.

A los docentes, autoridades y compañeros quienes han sido un soporte fundamental para cumplir este sueño.

Edwin Eduardo, Guanoluisa Huertas

DEDICATORIA

El presente proyecto de titulación lo dedico de todo corazón a mi esposa Alejandra, por ser mi compañera de vida, mi apoyo incondicional y mi fuente inagotable de inspiración.

A mi hijo Mathias, porque su alegría y amor me han impulsado a seguir adelante, eres mi mayor motivación y la razón por la que me esfuerzo cada día.

A mis padres, Edwin y María, por su amor incondicional, por cada sacrificio hecho por mí y por inculcarme valores que hoy me han permitido alcanzar esta meta.

A mis hermanos, Stiven y Odalis, por su compañía, sus palabras de aliento y por estar siempre presentes en mi vida, compartiendo alegrías y desafíos.

A mis queridos abuelitos, Manuel e Inés, por su amor infinito y por ser un ejemplo de sabiduría, fortaleza y dedicación.

A mis suegros Efraín y Maruja por estar siempre pendientes y apoyándome a lo largo de esta formación académica.

A mi cuñado Christopher por el apoyo incondicional y sus aportes de conocimientos en todo momento.

Edwin Eduardo, Guanoluisa Huertas

ÍNDICE GENERAL

TRIBUNAL DE SUSTENTACIÓN.....	II
CERTIFICACIÓN	III
DECLARACIÓN DE RESPONSABILIDAD	IV
CERTIFICACIÓN DE ANTIPLAGIO.....	V
AUTORIZACIÓN	VI
AGRADECIMIENTO.....	VII
DEDICATORIA	VII
ÍNDICE GENERAL	IX
ÍNDICE DE TABLAS	XII
ÍNDICE DE FIGURAS.....	XII
RESUMEN	XV
ABSTRACT	XV
INTRODUCCIÓN	1
CAPÍTULO 1. MARCO TEÓRICO REFERENCIAL.....	3
1.1. Revisión de literatura	3
1.2. Desarrollo teórico y conceptual	7
1.2.1. Inteligencia Artificial	7
1.2.2. Redes Neuronales.....	8
1.2.3. Procesamiento Digital De Imágenes	9
1.2.4. Reconocimiento Óptico De Caracteres (Ocr).....	10
1.2.5. Visión Artificial	10
1.2.6. Componentes De La Visión Artificial.....	10
1.2.7. Técnicas De Iluminación.....	11
1.2.8. Dirección IP.....	12
1.2.9. Programador De Horario Digital	12
1.2.10. Relé De 11 Pines.....	13
1.2.11. Sensor De Movimiento.....	14
1.2.12. Cámara IP	15
1.2.13. NVIDIA Jetson Nano	16
1.2.14. Estación De Energía.....	16
CAPÍTULO 2. METODOLOGÍA	18
2.1. Contexto de la investigación.....	18
2.2. Diseño y alcance de la investigación	18
2.3. Tipo y métodos de investigación	19
2.4. Población y muestra	19
2.5. Técnicas e instrumentos de recolección de datos	19

2.6. Procesamiento de la evaluación: Validez y confiabilidad de los instrumentos aplicados para el levantamiento de información.	20
2.7. Metodología Del Desarrollo	21
2.7.1. Diseño principal de comunicación entre computador y cámara de videovigilancia.....	21
2.7.2. Diseño del sistema de iluminación automático.....	22
2.7.3. Montaje de los elementos de control del sistema de iluminación automático.....	23
2.7.4. Programación del programador semanal de horas digital	24
2.7.5. Montaje de la cámara de videovigilancia en la zona de garaje.....	25
2.7.6. Configuración del router para comunicaciones ethernet entre el NVR y el ISP	26
2.7.7. Configuración de los parámetros de conexión entre el software HIKLOOK y la red de internet	30
2.7.8. Activación de la cámara	32
2.7.9. Configuración de los parámetros de tomas de imagen la cámara.....	33
2.7.10. Capturas de imagen en tiempo real	35
2.7.11. Open CV	35
2.7.12. Diseño del programa y red de entrenamiento neuronal.....	36
2.7.13. Dividir videos en frames	38
2.7.14. Determinación de las zonas de interés	45
2.7.15. Configuración de NVIDIA Jetson Nano	49
2.7.16. Configuración inicial del sistema operativo en la pantalla principal Jetson	52
2.7.17. Entrenamiento de detectnet.....	53
2.7.19. Organizar las Imágenes y las Etiquetas	58
2.7.20. Entrenamiento personalizado de la red neuronal	60
2.7.21. Base de datos obtenidos del prototipo.....	63
2.7.22. Subprocesos de ejecución de la programación para el prototipo.....	64
2.7.23. Diseño de interfaz de usuario.....	66
CAPÍTULO 3. RESULTADOS Y DISCUSIÓN	70
3.1. Pruebas de funcionamiento y validación del sistema de vigilancia.....	70
3.1.1. Pruebas de fiabilidad y confiabilidad del sistema con vehículos no registrados.....	82
3.1.2. Ventajas del prototipo	86
3.1.3. Limitaciones del prototipo.....	87
CONCLUSIONES	88
RECOMENDACIONES	89
REFERENCIAS.....	90
ANEXOS.....	93

ÍNDICE DE TABLAS

TABLA 1. Registro de usuarios para las pruebas del prototipo.....	70
TABLA 2. Resultados obtenidos en la base de datos del día 12/12/2024 por la mañana.....	71
TABLA 3. Resultados ingreso de vehículos empresariales en el horario de la tarde el día 12/12/2024	72
TABLA 4. Datos obtenidos en la mañana del día 13/12/2025 con vehículos personales	73
TABLA 5. Enfoque y entrenamiento de la red neuronal con las placas vehiculares que presentan inconvenientes (primer vehículo)	74
TABLA 6. Enfoque y entrenamiento de la red neuronal con las placas vehiculares que presentan inconvenientes (segundo vehículo)	75
TABLA 7. Análisis general de los datos obtenidos el día 08/01/2025 luego del entrenamiento y modificaciones	76
TABLA 8. Registro ingreso de vehículos empresariales posterior a modificaciones	77
TABLA 9. Datos obtenidos el último día de pruebas del prototipo 12/01/2025	78
TABLA 10. Ejemplo demostrativo de la base de datos	81
TABLA 11. Resultados de las pruebas con vehículos no registrados	82
TABLA 12. Ejemplo demostrativo de la base de datos con vehículos no registrados	84
TABLA 13. Prueba con vehículos registrados y no registrados	85

ÍNDICE DE FIGURAS

FIGURA 1. Red Neuronal.....	7
FIGURA 2. Etapas del procesamiento digital de imágenes.....	8
FIGURA 3. Proceso del Reconocimiento Óptico De Caracteres.....	9
FIGURA 4. Iluminación Frontal.....	11
FIGURA 5. Programador horario digital.....	12
FIGURA 6. Relé de 11 pines planos.....	13
FIGURA 7. Sensor de movimiento.....	14
FIGURA 8. Cámara IP 2 MP.....	15
FIGURA 9. Tarjeta NVIDIA Jetson Nano.....	16
FIGURA 10. Estación de energía 300 W.....	17
FIGURA 11. Ubicación geográfica de la oficina de CBVISION sucursal Salcedo.....	18
FIGURA 12. Diagrama general del sistema de videovigilancia y prototipo de seguridad.....	21
FIGURA 13. Montaje de los elementos de control del sistema de iluminación automático.....	24
FIGURA 14. Cable de red para conexión de la cámara con el NVR, esquema T568B.....	25
FIGURA 15. Montaje en techo de la cámara.....	26
FIGURA 16. Símbolo del sistema y comando ipconfig para obtener la puerta de enlace.....	27
FIGURA 17. Ventana de configuración inicial del router.....	27
FIGURA 18. Configuración de red WIFI para invitados.....	28
FIGURA 19. Configuración del equipo en modo Router.....	28
FIGURA 20. Configuración del router para designación del puerto de administración remota..	29
FIGURA 21. Conexión de internet establecida.....	29
FIGURA 22. Configuración del puerto LAN del router para conexión con el NVR.....	30
FIGURA 23. Ejecución del software HIKLOOK.....	31
FIGURA 24. Configuración del software HIKLOOK para conexión de red.....	31
FIGURA 25. Configuración de parámetros de conexión de red.....	32
FIGURA 26. Configuración del puerto de administración remota.....	32
FIGURA 27. Activación de la cámara con SADP.....	33
FIGURA 28. Parámetros por defecto de la calidad de imagen.....	33
FIGURA 29. Ajustes de imagen para la captura de video y calidad.....	34
FIGURA 30. Configuración de la velocidad de obturación.....	34
FIGURA 31. Configuración luz infrarroja.....	35
FIGURA 32. Diagrama de flujo de funcionamiento de OpenCV.....	36
FIGURA 33. Obtención de videos.....	38
FIGURA 34. Pantalla principal VLC.....	39

FIGURA 35. Menú herramientas VLC	39
FIGURA 36. Configuración preferencias de video	40
FIGURA 37. Configuración de capturas de pantalla en formato png.....	40
FIGURA 38. Reproducción de un video en VLC.....	41
FIGURA 39. Captura de pantalla manualmente	41
FIGURA 40. Preferencias en la barra de herramientas.....	42
FIGURA 41. Seleccionar todo en la pestaña de preferencias	42
FIGURA 42. Pestaña video, opción filtros.....	43
FIGURA 43. Filtro captura de imagen FPS.....	43
FIGURA 44. Definición del tiempo de captura de pantalla automático	44
FIGURA 45. Capturas de pantalla automáticas con VLC	44
FIGURA 46. Nombre de la clase etiquetada	46
FIGURA 47. Abrir el directorio en LabelImg	47
FIGURA 48. Selección de carpeta de trabajo para el directorio en LabelImg	47
FIGURA 49. Configuración formato Pascal/VOC	47
FIGURA 50. Etiqueta de la zona de interés	48
FIGURA 51. Etiqueta de la zona de interés	48
FIGURA 52. Información guardada en el archivo .xml	49
FIGURA 53. Preparación del HARDWARE	50
FIGURA 54. Pantalla principal programa ETCHER, opción Flash para completar el proceso ..	51
FIGURA 55. Mensaje de Flash completado en el programa ETCHER.....	51
FIGURA 56. Escritorio, pantalla principal de la tarjeta JETSON NANO.....	52
FIGURA 57. Terminal de Jetson Nano con el comando git clone.....	53
FIGURA 58. Comando Jetson Nano ejecutado el Docker/run.sh.....	53
FIGURA 59. Interfaz de Model Downloader selección all models	54
FIGURA 60. Archivos formato .xml.....	56
FIGURA 61. Script con las carpetas organizadas.....	58
FIGURA 62. 9,600 imágenes en formato jpg.....	58
FIGURA 63. Jetson Nano ejecutando la red de entrenamiento neuronal.....	60
FIGURA 64. Archivos de salida al finalizar el entrenamiento de la red neuronal	61
FIGURA 65. Ejecución de la conversión a un modelo onnx	62
FIGURA 66. Aplicación de la red neuronal sobre un video en marcha.....	63
FIGURA 67. Nuevo archivo en el programa DB Browser para SQLite y campos requeridos en la base de datos temporal	64
FIGURA 68. Indicadores de la interface al ingreso del aplicativo	66
FIGURA 69. Mensaje al agregar un nuevo usuario a la base de datos	67

FIGURA 70. Mensaje al ingresar un usuario nuevo.....	67
FIGURA 71. Mensaje cuando hace falta ingresar datos de los nuevos usuarios	68
FIGURA 72. Mensaje de error al no marcar o seleccionar un usuario para eliminar o editar.....	68
FIGURA 73. Mensaje de error al ingresar credenciales de acceso incorrectos.....	69
FIGURA 74. Mensaje de error cuando el usuario o contraseña no son ingresados	69
FIGURA 75. Muestra de funcionamiento del prototipo desde el interior del vehículo durante el día	80
FIGURA 76. Muestra de funcionamiento del prototipo desde el interior del vehículo durante la noche.....	80
FIGURA 77. Representación gráfica del sistema de iluminación automático	93
FIGURA 78. Diseño en AutoCAD de la zona donde se ha implementado el proyecto	94
FIGURA 79. Diagrama de flujo de la base de datos permanente	95
FIGURA 80. Diagrama de flujo de la base de datos temporal.....	96
FIGURA 81. Diagrama de flujo del funcionamiento total del prototipo	97
FIGURA 82. Carta de validación por parte de la empresa	98

RESUMEN

El desarrollo de este proyecto presenta el diseño e implementación de un prototipo de seguridad basado en inteligencia artificial para el análisis automatizado de datos, optimizando el control vehicular a través del reconocimiento óptico de caracteres, mediante el entrenamiento de redes neuronales y aprendizaje automático. El prototipo emplea sensores, cámaras, análisis de videos en tiempo real y algoritmos de programación para identificar patrones en las imágenes, obtener los datos y registrarlos en un archivo histórico, el sistema basa su funcionamiento por medio de una cámara de videovigilancia con iluminación automatizada para mejorar la captura de imágenes en distintas condiciones de luz, y DetectNet para la detección de placas.

Palabras claves: Inteligencia Artificial, Redes Neuronales, DetectNet.

ABSTRACT

The development of this project presents the design and implementation of a security prototype based on artificial intelligence for automated data analysis, optimizing vehicle control through optical character recognition, by training neural networks and machine learning. The prototype uses sensors, cameras, real-time video analysis and programming algorithms to identify patterns in the images, obtain the data and record them in a historical archive. The system bases its operation on a video surveillance camera with automated lighting to improve image capture in different light conditions, and DetectNet for license plate detection.

Keywords: Artificial Intelligence, Neural Networks, DetectNet.

INTRODUCCIÓN

Los sistemas de seguridad son tecnologías, procedimientos y estrategias diseñados para proteger personas, bienes e información frente a amenazas y riesgos, estos sistemas pueden ser físicos, como cámaras de vigilancia, alarmas y controles de acceso, o digitales, como firewalls, software de detección de intrusos y encriptación de datos, su objetivo principal es prevenir incidentes, detectar posibles ataques o focos de inseguridad y responder de manera rápida y eficiente en caso de que ocurra algún problema. (Vieites, 2014)

El procesamiento digital de imágenes ha cobrado gran relevancia en la tecnología y la informática en los últimos años, impulsado por la reducción en el costo de las computadoras, actualmente, puede realizarse desde una PC hasta un teléfono celular, su origen se remonta a 1964 en el Jet Propulsion Laboratory, cuando las imágenes de la Luna, capturadas por el Ranger 7, fueron procesadas para corregir distorsiones, estas técnicas sentaron las bases para la mejora de imágenes y fueron aplicadas en misiones como el Surveyor, los vuelos a Marte de los Mariner y las misiones Apolo a la Luna. (Guerra Cantero & Vanegas Lago, 2014)

(Roldán Cordero, 2023) desarrolla un sistema web para el registro automático de placas vehiculares utilizando cámaras de vigilancia en parqueaderos, el mismo que como parte de la detección de vehículos se utilizan un algoritmo de sustracción de fondo, para obtener el movimiento en un área específica, una vez detectado el movimiento se extrae el cuadro de imagen que será almacenado y procesado mediante transformación a escalas de grises y reducción de ruido, al completarse este proceso se aplica el algoritmo de detección de bordes de Canny, que identifica los 4 contornos pertenecientes a una placa vehicular.

Actualmente la sucursal Salcedo de la empresa CBVISION no cuenta con un sistema de seguridad que identifique los vehículos por número de placa al ingresar a los patios de garaje, lo que dificulta llevar un registro preciso de los ingresos, hoy en día la sucursal dispone de cinco vehículos propios, cuatro camionetas y una furgoneta, que regresan a su zona de parqueo al finalizar la jornada laboral, sin embargo, también ingresan vehículos de trabajadores y unidades de transporte de materiales y equipos de la agencia matriz, en muchos casos estos vehículos llegan durante la madrugada, aumentando el riesgo de seguridad para la empresa.

Los sistemas de control de acceso convencionales suelen ser ineficientes, propensos a errores humanos y vulnerables a fallos de seguridad, lo que compromete la integridad de las instalaciones y del personal, la falta de vigilancia nocturna y la entrada esporádica de vehículos ajenos a la agencia representan un punto crítico de inseguridad, por ello se propone el diseño e implementación de un prototipo de sistema de seguridad basado en inteligencia artificial, que permita registrar y gestionar automáticamente una base de datos de los vehículos que ingresan.

Formulación del problema de investigación

¿Cuál es la contribución de la Inteligencia Artificial y redes neuronales en los sistemas de videovigilancia en el monitoreo de ingresos o salidas de vehículos?

Objetivo General:

Implementar un prototipo de sistema de seguridad integral para monitorear el acceso de automóviles, utilizando visión artificial para la vigilancia y toma de datos automática en la agencia de CBVISION Salcedo.

Objetivos Específicos:

- Investigar el estado del arte de los sistemas de seguridad por medio de inteligencia artificial para obtener parámetros de diseño.
- Diseñar el prototipo de seguridad considerando variables de entrada y salida para la toma y análisis de datos.
- Diseñar un sistema de compensación de iluminación automático para un correcto desempeño del prototipo en las noches.
- Realizar el entrenamiento de la red neuronal que permita reconocer las placas de los vehículos que ingresan al garaje de la empresa

Planteamiento hipotético

¿Por qué razón se requiere implementar un sistema de seguridad videovigilado con inteligencia artificial en la agencia de la empresa de CBVISION de Salcedo?

¿La digitalización de imágenes y extracción de datos hacia una base de muestreos ayudará a un mejor registro del ingreso de los vehículos y personal trabajador al finalizar su jornada laboral?

CAPÍTULO 1. MARCO TEÓRICO REFERENCIAL

1.1. Revisión de literatura

En (Guerra Cantero & Vanegas Lago, 2014) se presenta el proceso de extracción de texto de una imagen digitalizada aplicando un proceso OCR, el mismo que permite extraer y editar el texto de imágenes, dicho proceso consta de 5 etapas: adquisición de la imagen, binarización, fragmentación o segmentación de la imagen, representación digital o extracción de características y distinción del carácter contenido en la imagen, este proceso aplican varios softwares libres y algoritmos de extracción de texto, dentro de los software para este proceso tenemos ABBYFineReader, Free Online OCR, Softi FreeOCR, FreeOCR, entre otros. El algoritmo OCR tiene implementado una red neuronal que permite el reconocimiento acertado de caracteres, el mismo que es capaz de leer una amplia variedad de formatos de imagen y convertirlo a texto en más de 40 idiomas, los lenguajes de programación más utilizados en estos procesos son C, C++, Python. Se desarrolla una aplicación utilizando el framework Qt, el cual está diseñado para ser ejecutado en varios sistemas operativos, utiliza C++ como lenguaje nativo, teniendo además soporte como Python, Java o C#.

(Roldán Cordero, 2023) hace referencia en su investigación a la tecnología OCR (Reconocimiento Óptico de Caracteres) se ha convertido en una herramienta esencial en la industria gracias a su capacidad para automatizar la lectura y procesamiento de datos, esta tecnología permite la digitalización de documentos, detección de caracteres, trazabilidad, lo que reduce los tiempos de proceso y aumenta la eficiencia en una variedad de sectores industriales, el entrenamiento de un modelo consiste en alimentarlo con un conjunto de datos de entrenamiento y ajustar los parámetros del modelo para que pueda hacer predicciones precisas y generalizadas en nuevos datos, esta etapa se puede organizar en los siguientes pasos:

1. Elección del modelo
2. Etiquetado de
3. Entrenamiento del modelo
4. Validación del modelo
5. Ajuste del modelo

6. Interpretación de resultados

En la investigación de (Duran Gomez, 2009) se habla sobre el reconocimiento a partir de la digitalización del texto que puede ser a través de una cámara, un escáner o capturando la imagen del carácter escrito por medio de un lápiz óptico, como primer paso, se realiza un tratamiento de la imagen capturada, pasando desde la binarización de la imagen hasta el filtrado que elimina el ruido que la distorsiona, el sistema de reconocimiento desarrollado es dependiente del escritor, por lo que cada sistema de clasificación escogido para reconocer el carácter debe ser previamente adaptado para que pueda realizar el proceso de reconocimiento de la manera correcta.

Sin embargo, se tiene en estos métodos de clasificación una gran variedad de ejemplos de entrenamiento, muestras y plantillas que permitirán en un alto índice, el reconocimiento de caracteres manuscritos de terceros en el programa. Se habla de dos métodos de clasificación muy conocidos para relacionar imágenes y crear variables de similitud para la correcta etiquetación de los caracteres a reconocer, estos dos métodos son los de correlación y la de redes neuronales. El método de correlación da como resultado un valor de relación entre los datos o patrones de las imágenes y dicha relación puede calcularse mediante diversos coeficientes que miden el grado de correlación entre las mismas, por otro lado, las redes neuronales serán entrenadas de tal forma que el sistema pueda identificar de un conjunto de posibles soluciones la mejor de las salidas en relación al dato de entrada del sistema.

(Zainal, Atul, Prashengit, & Shahadat, 2017) han llevado a cabo un estudio sobre un sistema de reconocimiento de matrículas vehiculares implementado en el entorno Python utilizando OpenCV, este sistema está diseñado para procesar imágenes de matrículas fuera de línea, capturadas en diferentes condiciones de iluminación, escenarios viales y vehículos de distintos colores.

El rendimiento del sistema se evalúa en función de cuatro métricas clave: tasa de detección, tasa de segmentación, tasa de reconocimiento y tiempo de ejecución, la detección de matrículas se basa en técnicas de procesamiento de imágenes, como el procesamiento morfológico, el suavizado gaussiano, el umbral adaptativo y el filtrado de contornos de los caracteres ubicados en la parte inferior de la placa. Para la etapa de

reconocimiento, se emplea un modelo de aprendizaje profundo basado en redes neuronales convolucionales multicapa.

En la implementación de (León, Huaman, & Castro, 2023) desarrollan un prototipo de visión artificial mediante el estudio, desarrollo de técnicas de filtrado y simplificación de datos para el reconocimiento de placas aplicado en varias ciudades de Perú, para el funcionamiento de este prototipo se ha implementado lenguaje de programación Python, Visual Studio Code, mediante estas librerías el sistema captura las imágenes de las placas, procesa las imágenes, compara el texto, lo extra y lo almacena en una base de datos, para su validación.

En el estudio llevado a cabo por (Agbeyangi, Alashiri, & Otunuga, 2020) denominado "Automatic Identification of Vehicle Plate Number using Raspberry Pi", el procedimiento de reconocimiento automático de matrículas de vehículos, se resalta que, pese a que el formato de las placas puede variar entre naciones, los métodos de reconocimiento generalmente se rigen por tres etapas esenciales: identificación, segmentación y reconocimiento de caracteres.

Las imágenes de las placas de los vehículos se capturan mediante la cámara, luego estas imágenes se almacenan en una tarjeta de memoria SD para su procesamiento y reconocimiento óptico de caracteres, con el objetivo de mostrar los datos de la placa identificada.

(Pech, y otros, 2021) exponen los hallazgos experimentales alcanzados en la elaboración y valoración de un sistema de visión artificial diseñado para la detección de caracteres en las placas de vehículos. Este sistema utiliza técnicas de identificación óptica de caracteres mediante la librería pytesseract, además de herramientas de procesamiento de imágenes proporcionadas por OpenCV, todo esto se lleva a cabo en un entorno de programación Python. Además, para la administración de datos y el almacenamiento de registros de matrículas detectados, se emplea una base de datos MySQL. Su operación se fundamenta en una conexión entre el código Python y el servidor de MySQL.

Según (Cáceres, 2021) en el ámbito de la programación, Python ofrece herramientas avanzadas para la implementación de OCR, una de las más utilizadas es Tesseract, un motor de reconocimiento óptico de caracteres de código abierto desarrollado por Google. Permitiendo a los desarrolladores extraer texto de imágenes y documentos escaneados

con gran precisión, digitalización de textos de manera automática, aplicar el OCR para la gestión de datos. En la actualidad el OCR se ha convertido en una solución poderosa para múltiples aplicaciones a través de Python, desde la digitalización de documentos hasta la automatización de procesos administrativos.

Para la gestión y tratado de datos se utiliza la herramienta MySQL, según (Casillas, Gilbert, & Perez, 2021) MySQL es un sistema gestor de bases de datos (SGBD) ampliamente utilizado debido a su simplicidad, alto rendimiento y estabilidad, aunque no cuenta con algunas funciones avanzadas que ofrecen otros gestores, su facilidad de uso y rápida implementación lo convierten en una opción ideal para diversas aplicaciones, tanto comerciales como de entretenimiento, uno de los principales beneficios de MySQL es su distribución libre bajo licencia GPL, lo que permite un desarrollo continuo y una gran comunidad de soporte, además, es compatible con múltiples plataformas, siendo GNU/Linux una de las más utilizadas

Para trabajar con MySQL, es fundamental contar con las credenciales del usuario administrador de la base de datos, aunque algunos procesos requieren privilegios limitados, la administración completa del sistema demanda permisos de administrador, especialmente en tareas avanzadas de gestión y configuración del SGBD.

(Rodrigo, 2021) habla acerca del entrenamiento de una red neuronal, el mismo que es un proceso iterativo en el cual se ajustan los valores de los pesos y los sesgos (bias) con el objetivo de minimizar el error en las predicciones del modelo, este ajuste se realiza mediante algoritmos de optimización, como el descenso de gradiente, los cuales modifican los parámetros de la red con base en la retropropagación del error, evaluado a través de una función de costo, permitiendo a la red neuronal entrenada la identificación de algoritmos de mayor relevancia en las capas de salida.

(Loaiza & Ibañez, 2024) en su proyecto que se enfoca en el desarrollo e implementación de un sistema inteligente basado en la placa de desarrollo Jetson Nano y un modelo de entrenamiento de red neuronal, se puede evidenciar que esta combinación de hardware y software permite la ejecución eficiente de algoritmos de inteligencia artificial, facilitando el procesamiento de datos en tiempo real y la toma de decisiones automatizadas en aplicaciones de visión por computadora, robótica, automatización industrial, entre otros.

1.2. Desarrollo teórico y conceptual

1.2.1. Inteligencia Artificial

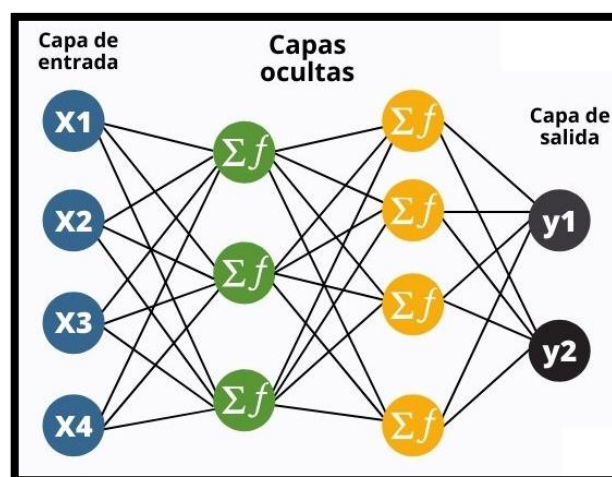
La inteligencia artificial es una rama de la informática que desarrolla programas capaces de emular procesos propios de la inteligencia humana, es decir, las máquinas pueden analizar el entorno y realizar determinadas acciones de manera más o menos autónoma con el fin de lograr objetivos concretos, por ejemplo, un chatbot es capaz de interpretar los problemas que plantean los clientes de un comercio o empresa y ofrecerles respuestas más precisas, así como un motor de recomendación realiza sugerencias acordes con las preferencias que el usuario ha mostrado con anterioridad. (Caiafa, 2020)

1.2.2. Redes Neuronales

Una red neuronal artificial es un modelo computacional inspirado en el funcionamiento del cerebro humano, diseñado para aprender a partir de experiencias y mejorar su desempeño en tareas específicas. Su principal objetivo es emular el proceso de aprendizaje y toma de decisiones mediante la interconexión de múltiples unidades de procesamiento, llamadas neuronas artificiales. Estas unidades trabajan en conjunto para reconocer patrones, clasificar datos y realizar predicciones con un alto grado de precisión y exactitud. (Berenguer, 2023)

FIGURA 1

Red Neuronal



Nota. Por Rubén Cañadas, 2021, Entrenamiento de las redes neuronales, AbDatum, (<https://abdatum.com/media/images/red-neuronal-multicapa.jpeg>)

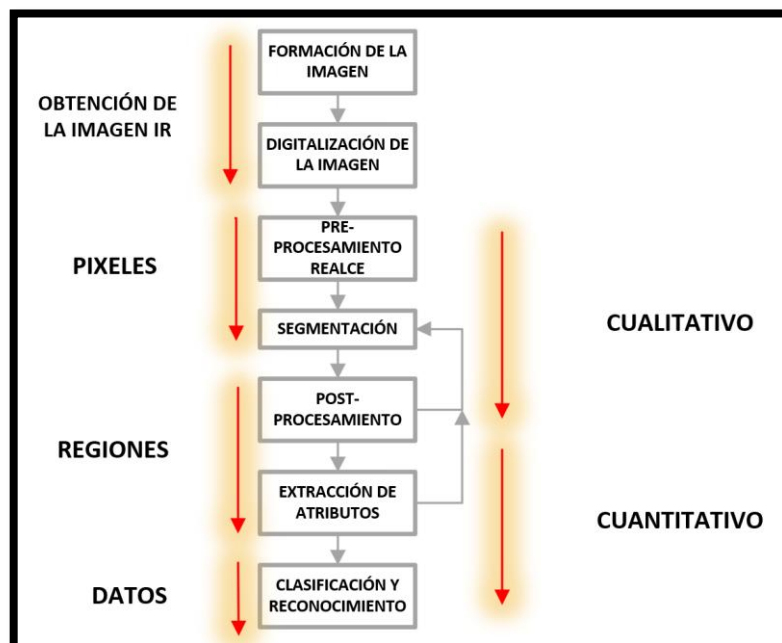
En la Figura 1 se muestra la arquitectura de una red neuronal artificial compuesta por una capa de entrada, dos capas ocultas y una capa de salida, esta estructura permite procesar la información de manera jerárquica, transformando los datos iniciales en representaciones cada vez más abstractas y significativas, la estructura interna de una red neuronal está determinada por cuatro elementos fundamentales: pesos sinápticos, sumador, función de activación, umbral o bias. (Cañadas, 2021)

1.2.3. Procesamiento Digital De Imágenes

El procesamiento de imágenes tiene como objetivo mejorar el aspecto de las imágenes y hacer más evidentes ciertos detalles sobresalientes, el procesamiento de las imágenes se puede en general hacer por medio de métodos ópticos, o bien por medio de métodos digitales, en una computadora el procesamiento digital de imágenes se efectúa dividiendo la imagen en un arreglo rectangular de elementos, cada elemento de la imagen así dividida se conoce con el nombre de pixel, el siguiente paso es asignar un valor numérico a la luminosidad promedio, así, los valores de la luminosidad de cada pixel, con sus coordenadas que indican su posición, definen completamente la imagen. (Rubio, 2020)

FIGURA 2

Etapas del procesamiento digital de imágenes



Nota. Por Pérez, E. R., 2017, Etapas del procesamiento digital de imágenes, visión artificial, ResearchGate, (https://www.researchgate.net/figure/Secuencia-de-procesamiento-y-analisis-de-imagenes-digitales-infrarrojas_fig1_262874787)

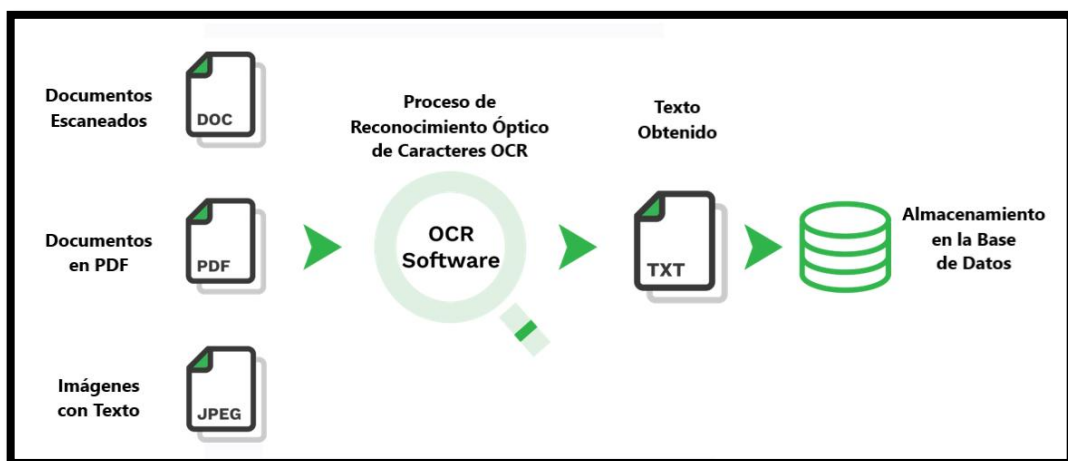
En la Figura 2 se presenta el flujo de procesamiento digital de imágenes, el cual comprende varias etapas clave, el proceso inicia con la captura de la imagen a través de un dispositivo de adquisición, posteriormente, la imagen obtenida es sometida a un tratamiento específico que incluye técnicas de preprocesamiento, como mejora de contraste, reducción de ruido y corrección de iluminación, finalmente, en la última fase, se aplica el reconocimiento óptico de caracteres (OCR), un procedimiento basado en inteligencia artificial que permite identificar y extraer texto dentro de la imagen, como números y letras de placas vehiculares.

1.2.4. Reconocimiento Óptico De Caracteres (Ocr)

El reconocimiento óptico de caracteres (OCR) es el proceso por el cual se convierte una imagen de texto en un formato de texto que pueden leer las máquinas, por ejemplo como se muestra en la Figura 3, si escanea el formulario o un recibo, su computadora guarda el escaneo como un archivo de imagen, no se puede utilizar un editor de texto para editar, buscar o contar las palabras del archivo de imagen, sin embargo, se puede utilizar el OCR para convertir la imagen en un documento de texto con su contenido como datos de texto. (Maghni, 2024)

FIGURA 3

Proceso del Reconocimiento Óptico De Caracteres



Nota. Por Yeelen, R., 2024, Flujo del Reconocimiento óptico de caracteres OCR, Klippa, (https://www.researchgate.net/figure/Secuencia-de-procesamiento-y-analisis-de-imagenes-digitales-infrarrojas_fig1_262874787)

El sistema realiza la transformación de los archivos impresos a digitales gracias al reconocimiento óptico de caracteres, una tecnología que ha aprendido a distinguir diferentes símbolos, como letras, números y signos, este proceso requiere un entrenamiento previo, durante el cual se ha almacenado la información en una base de datos, posteriormente, la herramienta utiliza un algoritmo de coincidencia de patrones para comparar los elementos del documento con su base de datos interna, logrando así identificar y extraer la información deseada.

En base a la Figura 3, las etapas del OCR son:

1. Digitalización del documento con un escáner o fotografía que contenga texto.
2. Procesamiento de la imagen, convirtiéndola en una versión en blanco y negro.
3. El sistema identifica las zonas oscuras y dentro de ellas segmenta los caracteres, reconociendo los patrones y características que coinciden con su base de datos.
4. Reconstruye el texto y lo convierte en un formato editable almacenado en una base de datos. (Ayerdi, 2024)

1.2.5. Visión Artificial

La visión artificial es un campo de la inteligencia artificial (IA) que permite a los ordenadores y sistemas extraer información significativa a partir de imágenes digitales, videos y otras entradas visuales, y tomar medidas o realizar recomendaciones en función de esa información, si la IA permite a los ordenadores pensar, la visión artificial les permite ver, observar y comprender, la visión artificial entrena a las máquinas para realizar estas funciones, pero tiene que hacerlo en mucho menos tiempo con cámaras, datos y algoritmos en lugar de retinas, nervios ópticos y una corteza visual. (Gomez, 2024)

1.2.6. Componentes De La Visión Artificial

Los diferentes componentes de la visión artificial trabajan juntos para adquirir, procesar y analizar características en las imágenes con la finalidad de tomar decisiones correctas, el éxito de una aplicación de visión depende del uso correcto de cada uno de los componentes de la visión artificial y de su funcionamiento conjunto.

Los principales componentes de un sistema de visión artificial incluyen:

- Iluminación especial para sistemas de visión artificial
- Lentes y ópticas para visión artificial.
- Sensores de imagen.
- Procesadores de visión.
- Comunicaciones. (Laura, 2023)

1.2.7. Técnicas De Iluminación

Las técnicas de iluminación son fundamentales a la hora de determinar la imagen que se obtiene ya que, en última instancia, la luz es la materia prima con la que trabajará cualquier cámara de visión artificial, pertenezca a un rango y otro del espectro de luz, cuando deseamos implementar visión artificial en una aplicación, lo primero que debemos tener en cuenta es el tipo de iluminación que vamos a emplear, la técnica más común de todas las técnicas de iluminación es la iluminación frontal, en este caso, la cámara se ubica mirando al objeto y en la dirección de la luz. (Cervilla, 2017)

FIGURA 4

Iluminación Frontal



Nota. Por Cervilla, D., 2017, Técnica de iluminación frontal, BCNVISION, (<http://www.bcnvision.es/blog-vision-artificial/wp-content/uploads/2017/04/luz-frontal.png>)

Como se puede observar en la Figura 4, para la iluminación frontal la cámara se posiciona mirando al objeto en la misma dirección que la luz, esto reduce las sombras, suaviza las texturas y minimiza la influencia de rayas, polvo e imperfecciones que pueda tener el objeto, la cámara recibe la luz reflejada del objeto, este tipo de iluminación se consigue mediante anillos de luz. El uso de la iluminación frontal mediante anillos de luz es una técnica eficaz en la captura de imágenes, debido a que al iluminar el objeto desde la misma

dirección en la que la cámara captura la imagen, se minimiza la formación de sombras que podrían afectar la visibilidad de los detalles, además los anillos de luz distribuyen la iluminación de manera uniforme alrededor del objeto, evitando áreas sobreexpuestas o subexpuestas, lo que contribuye a una mejor detección y procesamiento digital de la imagen.

1.2.8. Dirección IP

La dirección IP permite que los dispositivos se comuniquen entre sí a través de la red, ya sea para enviar correos electrónicos, transmitir videos o simplemente navegar por la web, es como una especie de código postal digital que permite acceder a diferentes servicios en línea y compartir información en todo el mundo, aunque también lo impide si en los servidores está aplicándose algún tipo de filtrado de IP.

- Una IP estática es fija y permanente que se asigna manualmente a un dispositivo.
- Una IP dinámica se asigna automáticamente a un dispositivo cuando se conecta a internet. (Wong, 2023)

1.2.9. Programador De Horario Digital

FIGURA 5

Programador horario digital



Nota. Por Romero, F., 2024, Programador de Horario Semanal, ELECTROCENTER, (<https://electrocenter.ec/producto/sassin-v-1350-programador-digital-kg316t-110v-a1/>)

El programador horario digital mostrado en la Figura 5, es un dispositivo que se puede comparar como un reloj programable semanal (24 horas y 7 días) recomendado para la

automatización del encendido y apagado de equipos del hogar, el comercio o la industria. Este programador es capaz de realizar hasta 20 eventos (10 encendidos y 10 apagados) con una frecuencia diaria y/o semanal, de acuerdo con las programaciones previamente establecidas.

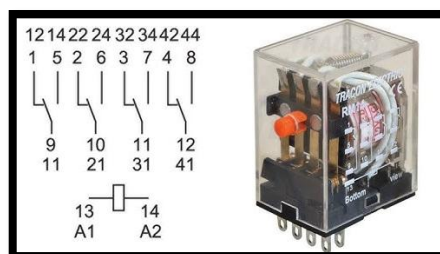
Características principales:

- Modelo: KG316.
- Voltaje de Suministro: 110/220 VAC.
- Consumo: 1.5 VA.
- Pantalla: LCD.
- Batería Interna 1. 1.5V.
- Operación: 20 eventos de encendido/apagado (día/semana).
- Contacto: 1 contacto normalmente abierto de capacidad de 10 Amperios. (Herco, 2021)

1.2.10. Relé De 11 Pines

FIGURA 6

Relé de 11 pines planos



Nota. Por Arias, K., 2024, Relé de 11 pines planos con base, INGECOMSA, (<https://ingecomsas.com/producto/relevo-relpol-11-pines-planos-110v-ac/>)

El relé de 11 pines planos con bobina de 120 VAC mostrado en la Figura 6, es un dispositivo electromecánico diseñado para controlar circuitos eléctricos de manera eficiente y confiable, este relé es ideal para aplicaciones que requieren una conmutación precisa y segura, tanto en entornos industriales como comerciales.

Características Principales:

- Número de Pines: 11 planos
- Tipo de Bobina: 120 VAC
- Capacidad de Contacto: Hasta 5A

- Configuración de Pines: Planos, lo que facilita la conexión y el montaje en diversas aplicaciones. (MAZ, 2023)

1.2.11. Sensor De Movimiento

FIGURA 7

Sensor de movimiento



Nota. Por Steren, L., 2023, Sensor de Movimiento, Electrónica Steren, (<https://www.steren.com.ec/sensor-de-movimiento-pir-para-intemperie.html>)

Un sensor de presencia o sensor de movimiento como el mostrado en la Figura 7, es un dispositivo electrónico que pone en funcionamiento un sistema (encendido o apagado) cuando detecta movimiento en el área o ambiente en el que está instalado, se utilizan con frecuencia para optimizar el consumo y la eficiencia energética de diversos sistemas como la ventilación, la iluminación o el aire acondicionado en el hogar o en la oficina, aunque también tiene aplicaciones en el ámbito de la seguridad.

Características Principales:

- Ajuste de sensibilidad de luz ambiente para encender con baja luminosidad
- El sensor es ajustable en su direccionamiento
- Detecta movimiento en 180°
- Ajusta el tiempo que permanecerá encendido al detectar movimiento: 5 s a 7 min
- Distancia máxima de detección: 11 m
- Carga máxima de 800 W
- Alimentación 120 VAC. (Steren, 2024)

1.2.12. Cámara IP

Una cámara IP (Internet Protocol Camera) como la que se puede observar en la Figura 8, es un tipo de cámara de video digital que se conecta a una red (local o a Internet) y transmite imágenes o videos en tiempo real mediante el Protocolo de Internet (IP), estas cámaras son ampliamente utilizadas en sistemas de videovigilancia y monitoreo remoto, este tipo de cámaras están diseñadas especialmente para enviar video y audio a través de internet con la ayuda de un Router o a través de una red local para poder monitorear en directo desde cualquier parte del mundo por medio de cualquier equipo que esté conectado a internet. (HIKVISION, 2023)

FIGURA 8

Cámara IP 2 MP



Nota. HIKVISION, 2023, Cámara 2 MP tipo domo con sensor Infrarrojo, HIKVISION, (<https://www.hikvision.com/es-co/products/IP-Products/Network-Cameras/Pro-Series-EasyIP-/ds-2cd1343g2-i-uf-/?subName=DS-2CD1343G2-IUF>)

Características:

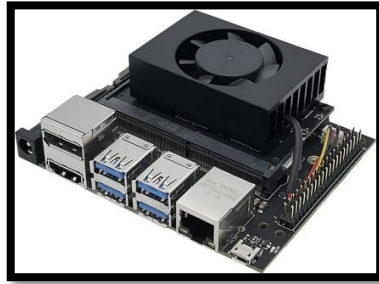
- Resolución: HD1080P
- NTSC: 1920(H)*1080(V)
- Lente: 2.8mm
- Ángulo de Visión: 103°
- Iluminación: 0.01Lux@F1.2/ 0Lux IR On (24un.)
- Velocidad de Obturador: 1/30 a 1/50,000
- Smart IR

- Alimentación: 12Vdc +/- 15%
- Consumo: 4 Watt.

1.2.13. NVIDIA Jetson Nano

FIGURA 9

Tarjeta NVIDIA JETSON nano



Nota. Por López, L., 2024, NVIDIA Jetson Nano Módulo, MOUSER Electronics, (<https://www.mouser.ec/new/seeed-studio/seeed-studio-jetson-nano-module/>)

El módulo Jetson Nano como se puede observar en la Figura 9, es una computadora pequeña y potente para aplicaciones integradas e IoT con inteligencia artificial que ofrece la potencia de la inteligencia artificial moderna y la capacidad de trabajar con redes neuronales de manera paralela y procesar información de varios sensores.

Características:

- Procesador de cuatro núcleos ARM Cortex-A57
- GPU NVIDIA Maxwell de 128 núcleos.
- Memoria y almacenamiento: 4 GB de memoria y soporte para tarjetas microSD.
- Desempeño gráfico y soporte para IA.
- Bajo consumo energético, con un consumo de energía de solo 5-10 vatios.
- Conectividad mediante puertos USB 3.0, HDMI, Ethernet, y GPIOs. (NVIDIA, 2021)

1.2.14. Estación De Energía

La estación de energía portátil tiene como objetivo principal garantizar un suministro eléctrico continuo, especialmente en caso de cortes energéticos, su función es proporcionar una fuente de alimentación confiable para mantener en operación los

sistemas críticos, asegurando la estabilidad y el funcionamiento ininterrumpido de los dispositivos conectados. (Stereon, 2024)

FIGURA 10

Estación de energía 300 W.



La estación de energía mostrada en la Figura 10, cuenta con una capacidad de 300 vatios, 299 Wh y 80,818 mAh, lo que le permite alimentar una variedad de equipos durante un tiempo prolongado. Además, ofrece tres modos de carga, mediante la red eléctrica, a través de un panel solar o utilizando la toma de cigarrera de un vehículo, lo que la convierte en una solución flexible para diferentes entornos y necesidades energéticas.

CAPÍTULO 2. METODOLOGÍA

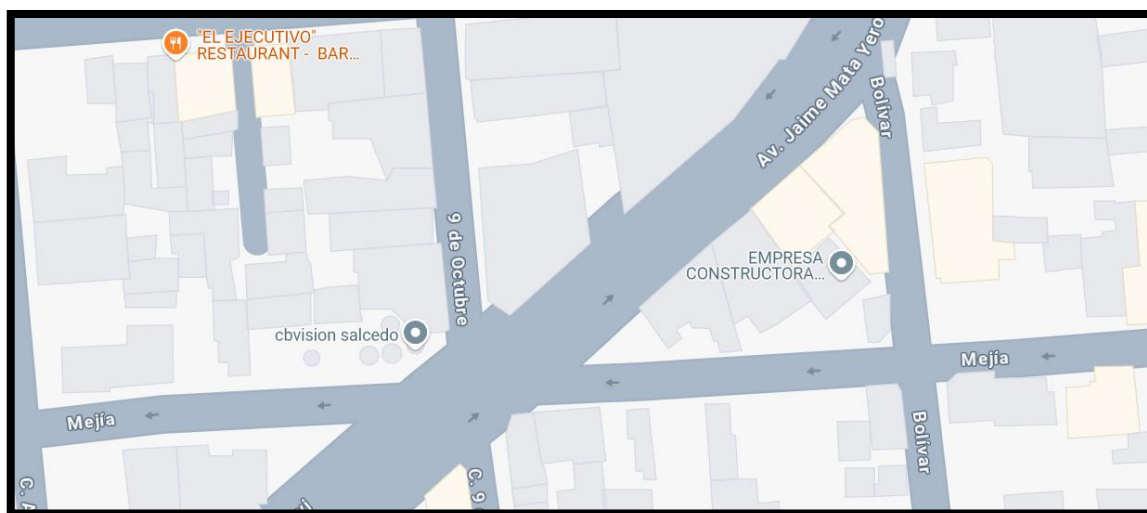
2.1. Contexto de la investigación

La empresa CBVISION es una de las más grandes en redes de televisión a nivel nacional y en la actualidad una de las empresas en el top 10 de telecomunicaciones ISP a nivel nacional, su agencia matriz se encuentra ubicada en la ciudad de Paute, provincia de Azuay y una de sus sucursales en la ciudad de Salcedo, Parroquia San Miguel, en las calles 9 de Octubre y Mejía.

Esta sucursal cuenta con un edificio propio, una planta habitacional de dos pisos y una planta edificada para el área de equipos, cajas, oficinas administrativas y bodega de un solo piso, además cuenta con un espacio amplio destinado a zona de parqueo y bodega general, sus coordenadas geográficas son -1.0463442425711935, -78.59172249369242.

FIGURA 11

Ubicación geográfica de la oficina de CBVISION sucursal Salcedo.



2.2. Diseño y alcance de la investigación

El presente proyecto de investigación es de carácter experimental, debido a que se conjuga varios caminos investigativos y a su vez diferentes escenarios de implementación, así como también se deberá considerar al momento de la implementación los factores climáticos, tomas de las fotografías, entre otros, y de ello ir calibrando y programando nuestro prototipo de seguridad.

El alcance que se quiere obtener es de tipo analítico, ya que, se van a analizar diferentes datos y llevarlos a una base de almacenamiento en común para su posterior aprobación por parte de los ejecutivos encargados de la agencia, esto implica un enfoque centrado en la recopilación, procesamiento y análisis de datos para obtener información relevante que sirva de base para la toma de decisiones.

2.3. Tipo y métodos de investigación

Tipo de investigación cuantitativo

Usa recolección de datos para probar hipótesis con base a la medición numérica y el análisis estadístico para establecer patrones de comportamiento, en este presente proyecto de investigación se aplica un enfoque cuantitativo ya que se realiza diferentes tomas de fotografías, posterior digitalizarlas y extraer datos alfa numéricos, así también se presenta los diferentes estados climáticos en la zona, posibles atenuaciones de iluminación, y deterioro en los objetos a muestrear. Todo ello implica un sin-número de datos y su análisis, por lo tanto, este enfoque ofrece una perspectiva diferente para estudiar la inteligencia artificial de manera más profunda y los resultados obtenidos de los modelos entrenados

2.4. Población y muestra

La población objetivo para el presente proyecto está claramente definida, lo cual es crucial para garantizar la relevancia y aplicabilidad de los resultados, en este caso, se trata de todo el personal administrativo y técnico de la agencia CBVISION en la ciudad de Salcedo, conformado por un total de 14 personas, este grupo constituye los beneficiarios directos del proyecto.

Al ser una población pequeña, es posible abordar a toda la población sin necesidad de tomar una muestra, esto asegura que los resultados sean representativos del grupo completo, por tanto, este proyecto es de carácter Muestreo No probabilístico debido que el enfoque o de la selección de las unidades de la muestra no depende del azar, sino de criterios subjetivos o conveniencia del investigador.

2.5. Técnicas e instrumentos de recolección de datos

En este apartado se utiliza una técnica de recolección de datos mixta, se aplica encuestas a los trabajadores previo a la implementación y una vez implementado el prototipo de

seguridad, también se aplica entrevistas a los ejecutivos de la agencia así mismo previo y posterior a la implementación.

Como punto de partida y en una encuesta mediante WhatsApp se preguntó a los trabajadores sobre el implementar un prototipo de seguridad que les permita registrar el ingreso vehicular automáticamente en la agencia y se obtuvo 14 aciertos de factibilidad, siendo esto un 100% de aprobación, además, en una entrevista por parte de gerencia hay toda la predisposición y apoyo para el desarrollo del presente proyecto, esto parte como una necesidad de incrementar los factores de seguridad en la empresa, tanto en el día, como en la noche.

La base de recolección de datos se encuentra registrado en el repositorio virtual en el dispositivo NVIDIA Jetson Nano, donde el prototipo captura la imagen mediante la cámara y con ayuda del entrenamiento de la red neuronal procesa la imagen y extrae el texto, una vez con el texto extraído se compara con una base de datos y lo almacena según sea personal de la agencia o personal no registrado.

2.6. Procesamiento de la evaluación: Validez y confiabilidad de los instrumentos aplicados para el levantamiento de información.

Para validar la propuesta de manera inicial se basa en los resultados obtenidos en las investigaciones realizadas, donde se analiza la parte de resultados estadísticos, y en base a ello podremos ver la factibilidad de seguir aplicando ciertos elementos para el diseño y la implementación, podremos basarnos en los errores anteriores para realizar mejoras y nuevas implementaciones e innovaciones, en general la mayoría de investigaciones realizadas tienen muchos datos en común y programas aplicados en común, además para garantizar el correcto funcionamiento del prototipo se utilizará materiales y equipos de marcas reconocidas a nivel local, siendo el punto más importante y crítico la cámara de toma de imágenes, la cual deberá ser de una gama alta y con grado IP alto.

Una vez implementado el prototipo previo a su entrega y puesta en marcha se hará varias pruebas-error, esto con la finalidad de garantizar el 100% de efectividad, considerando los diferentes escenarios de funcionamiento, de aquí se tomará muestras de datos y se los pasará a tablas de tabulación para su respectivo análisis estadístico.

2.7. Metodología Del Desarrollo

El desarrollo del proyecto conjuga la inteligencia artificial y el entrenamiento de redes neuronales mediante la aplicación de métodos de aprendizaje automático, el prototipo en su etapa de funcionamiento es capaz de detectar imágenes, identificar las placas de los vehículos, registrar la información en una base de datos para su posterior validación y aprobación. El prototipo se ha creado con una perspectiva flexible, facilitando la interacción del usuario y su personalización para futuras modificaciones o aplicaciones en las otras sucursales de la empresa

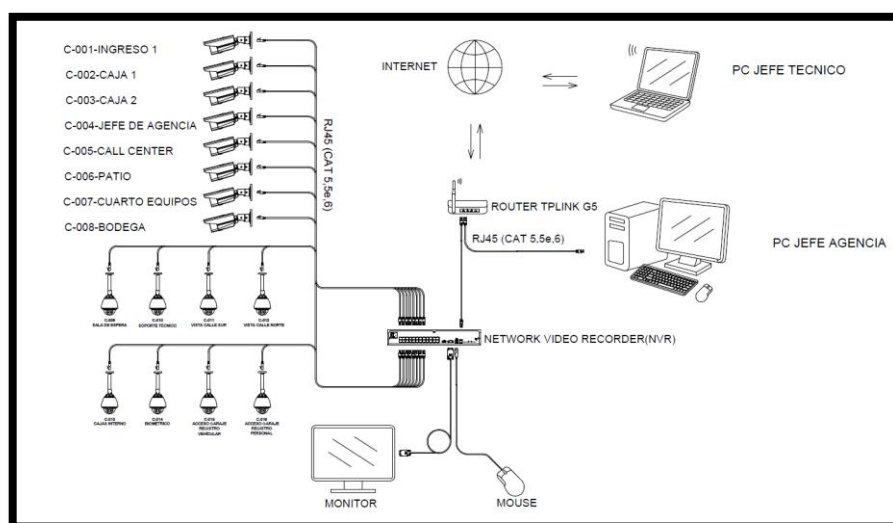
2.7.1. Diseño principal de comunicación entre computador y cámara de videovigilancia

La empresa CBVISION agencia Salcedo al ser una entidad ISP, cuenta con internet para distribución residencial y comercial, en este apartado se omiten varios criterios de programación y configuraciones por temas de confidencialidad, se muestran únicamente los pasos más relevantes e importantes para la configuración y ejecución del proyecto.

El esquema de diseño de nuestro sistema de videovigilancia está basado en una comunicación ethernet entre el router, la cámara y la computadora principal como se muestra en la Figura 12, las configuraciones de enlace se realizan mediante asignación de IP estáticas tanto al router como a nuestro programa de vigilancia, permitiendo un enlace con la red de internet y posterior la transmisión de datos de imágenes captadas.

FIGURA 12

Diagrama general del sistema de videovigilancia y prototipo de seguridad



2.7.2. Diseño del sistema de iluminación automático

El sistema de iluminación mostrado en el Anexo 1, está diseñado para entrar en funcionamiento desde las 18h30 hasta las 6 am, esta programación se la realiza por medio de un programador de horas, donde podemos ejecutar varias opciones de programación, en este proyecto se lo realiza para toda la semana en el periodo de tiempo antes descrito.

- Función Primaria: Encender el sistema de iluminación automáticamente a las 18h30 y apagarlo a las 6h00 del día siguiente.
- Programación Horaria: Configurable para que opere de manera consistente durante toda la semana.
- El Relé de Control viene interconectado con el programador para activar o desactivar el circuito de iluminación.
- Asegurar una conexión estable y protegida contra sobretensiones.
- Configurar el programador para que este ciclo se repita automáticamente de lunes a domingo.

Es muy importante que el funcionamiento del controlador del sistema de iluminación funcione todos los días, ya que la empresa tiene un horario de trabajo de lunes a sábado y los días domingos soporte técnico en casos emergentes, esto también en días feriados, además en ciertos casos ingresan camionetas o camiones a descargar materiales provenientes desde la agencia matriz.

Para el control de encendido del reflector led se lo realiza por medio de un sensor de movimiento, este emite una señal de encendido a nuestro relé de control y este a su vez se enclava y permite que la lámpara se encienda y así mejorar la calidad y toma de imágenes para nuestro prototipo.

- El sensor de movimiento detecta actividad en su área de cobertura y emite una señal eléctrica.
- El relé recibe la señal del sensor, se activa y envía corriente al reflector LED.
- El relé asegura que la lámpara permanezca encendida durante el tiempo definido.
- El reflector led ilumina el área de interés, mejorando la calidad visual y captación de imágenes por la cámara.

El ajuste del sensor de movimiento se lo realiza con las siguientes consideraciones:

- Configurar la sensibilidad del sensor para evitar activaciones innecesarias.
- Ajustar el tiempo de retardo según el uso, en este caso se trabaja a 10 segundos.
- Ajustar el modo de funcionamiento para que la fotocélula trabaje con un poco de luz, es decir no ajustar la oscuridad al 100%, se regulará al 75% aproximadamente.
- Realizar pruebas para verificar que el sistema detecte movimiento y encienda el reflector correctamente.
- Ajustar los parámetros del sensor y el tiempo de enclavamiento del relé según sea necesario.

Es muy indispensable contar con un suministro estable para el sensor, relé, programador digital de horas y reflector, por lo cual se ha implementado una estación de energía, la cual se puede observar en la Figura 10, la misma que está destinada a suplir los cortes energéticos y mantener activo el NVR, y el sistema de iluminación automático, la computadora de almacenamiento de datos funciona con UPS propios de la empresa y el generador instalado, no se pudo trabajar con este sistema de transferencia de energía por temas de cargas eléctricas ya que el sistema eléctrico del cuarto de equipos y oficinas administrativas están operando dentro de la capacidad límite del generador y UPS.

2.7.3. Montaje de los elementos de control del sistema de iluminación automático

El sistema de control de iluminación automático se encuentra implementado dentro del tablero de control y transferencia, recalcando que el sistema de iluminación va a trabajar con energía de la red y energía de la estación suplementaria, aquí la implementación se divide por secciones donde tenemos la parte de protección, esquema de control y por último salidas a los diferentes elementos, se describe a continuación los elementos que se encuentran enmarcados dentro de nuestro tablero de control mostrado en la Figura 13.

1. Breaker de protección 1 polo 10 Amperios, montaje en riel DIN, el mismo que está destinado a la protección de nuestro circuito de iluminación automático.
2. Relé de 11 pines, este relé es utilizado para la parte de fuerza y suplir a un contactor por temas de costos y funcionalidad, además el reflector al ser led y presentar un consumo en corriente muy bajo en estos casos no es común encontrar un contactor para dicha funcionalidad.

3. Borneras de salida, estas borneras son utilizadas con la finalidad de tener una mayor facilidad al momento de mantenimientos o cambios dentro de nuestro sistema de control.

FIGURA 13

Montaje de los elementos de control del sistema de iluminación automático



2.7.4. Programación del programador semanal de horas digital

Encendido del dispositivo:

- Conectar el programador digital a una fuente de energía.

Configuración inicial:

- Ajustar la hora actual en el programador.
- Usar los botones etiquetados como "Clock", "Hour", "Min" o similares.
- Seleccionar el formato de hora (12/24 horas) si el dispositivo lo permite.

Programación de horarios:

- Esto se lo realiza con el botón de programación, generalmente marcado como "Timer" o "Prog".
- Configurar el primer horario presionando el botón "Prog"

Establecer:

- Hora de encendido: Ajusta las horas y minutos para el encendido del dispositivo.
- Hora de apagado: Configura las horas y minutos para el apagado.

Seleccionar días de la semana:

- En este caso se va trabajar con la programación: Todos los días.

Modo automático:

- Una vez configurados los programas, cambiar el modo del programador a "Auto" o "Run" para que funcione según la programación establecida.

2.7.5. Montaje de la cámara de videovigilancia en la zona de garaje

La cámara implementada está mostrada mediante una flecha en el Anexo 2, la misma que también muestra de manera gráfica la estructura arquitectónica de la empresa CBVISION sucursal Salcedo, una vez adquirida la cámara, se verifica que todos los componentes estén correctos, especialmente en proyectos de vigilancia o prototipos de seguridad, una vez lista la cámara se procede al montaje de la misma en el área de mayor enfoque para la captura de imágenes, siendo el lugar destinado bajo la cubierta del garaje, siendo este el punto más óptimo para el montaje como se puede observar en la Figura 14.

FIGURA 14

Montaje en techo de la cámara



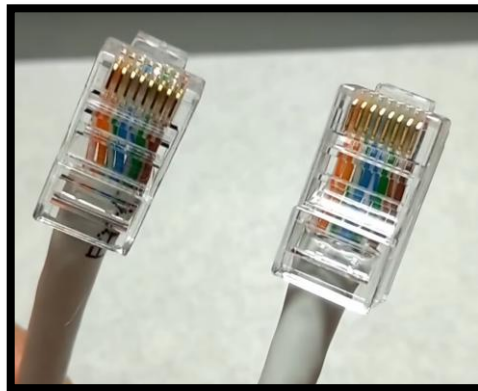
El cable ethernet de la cámara se conecta directamente desde el NVR, así como también el puerto de alimentación 12 V DC, para la conexión de datos se realiza una conexión directa desde el NVR y la cámara.

Para conectar el cable Cat 5e, se utilizan conectores RJ45, existen dos esquemas estándar, T568A y T568B, en este caso se emplea el esquema T568B mostrando en la Figura 15, siendo este el más común en redes de transmisión de datos.

- Blanco/Naranja, Naranja, Blanco/Verde, Azul, Blanco/Azul, Verde, Blanco/Marrón, Marrón.

FIGURA 15

Cable de red para conexión de la cámara con el NVR, esquema T568B



Para garantizar una buena conexión de red es muy importante asegurarse de que el cable Ethernet esté en buen estado y con conectores RJ45 crimpados correctamente tal como se observa en la Figura 18, una vez realizado el cableado de red para la cámara y el NVR este se inserta un extremo del cable Ethernet en uno de los puertos LAN disponibles en el NVR y el otro extremo en el puerto LAN de la cámara, el cable Ethernet suministra tanto energía como datos, eliminando la necesidad de un adaptador de 12V DC, como recomendación confirmar que la cámara es compatible con PoE para evitar problemas de alimentación, en este caso la cámara utilizada para el proyecto es compatible con PoE.

2.7.6. Configuración del router para comunicaciones ethernet entre el NVR y el ISP

Como primer paso se configura el router, el mismo que es de marca TPLINK AC 1300, G5, esta configuración se la puede realizar por varios métodos alternativos, siendo la más utilizada por medio del navegador web e ingresando a su dirección IP, para conocer su IP de acceso a configuraciones nos conectamos via cable de red ethernet o wifi al router y en nuestro computador ingresamos al símbolo del sistema y digitamos el comando ipconfig, donde nos arroja la IP de acceso a configuraciones del equipo como se puede observar en la Figura 16.

FIGURA 16

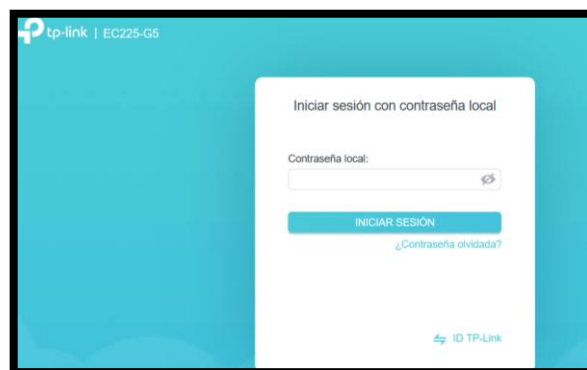
*Símbolo del sistema y comando **ipconfig** para obtener la puerta de enlace*

```
Adaptador de LAN inalámbrica Wi-Fi:  
Sufijo DNS específico para la conexión. . . :  
Vínculo: dirección IPv6 local. . . . . : fe80::ac42:a453:a252:17b7%4  
Dirección IPv4. . . . . : 192.168.0.103  
Máscara de subred . . . . . : 255.255.255.0  
Puerta de enlace predeterminada . . . . . : 192.168.0.1
```

Posterior se abre el navegador web y en el buscador digitamos la IP 192.168.0.1 y se presiona la tecla enter, ahí redirige a las configuraciones del equipo como se observa en la Figura 17, se establece una conexión al router (vía cable Ethernet o Wi-Fi), es importante destacar que al acceder a la configuración del router se debe iniciar sesión utilizando las credenciales predeterminadas y cambiarlas inmediatamente para mejorar la seguridad de la red.

FIGURA 17

Ventana de configuración inicial del router



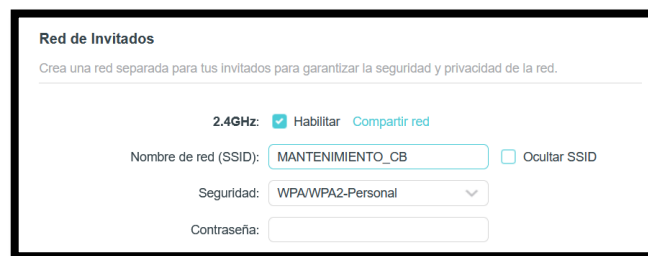
Como primer paso se crea una contraseña de acceso a las configuraciones, la contraseña solicitada por el jefe de agencia y el jefe técnico es confidencial por lo cual se omite la descripción en este apartado, al ingresar la contraseña ya se puede observar las diferentes pestañas de configuración de nuestro equipo router.

Dentro de las configuraciones inalámbricas es muy importante establecer un nombre identificable, pero que no revele información personal o empresarial, elegir el estándar más seguro disponible, en este caso vamos a trabajar con WPA2, definir una contraseña robusta siguiendo las mismas recomendaciones de seguridad mencionadas para el acceso al router, seleccionar un canal manualmente para minimizar interferencias, especialmente en entornos con muchas redes cercanas.

Además de estas configuraciones se puede configurar una lista de control de acceso (ACL) para limitar los dispositivos que pueden conectarse al router mediante sus direcciones MAC, como complemento a la ACL y posibles conexiones de terceros a la red podemos configurar una red de invitados como se muestra en la Figura 18, la misma que configura una red separada para visitantes o dispositivos temporales, lo que evita que accedan a la red principal.

FIGURA 18

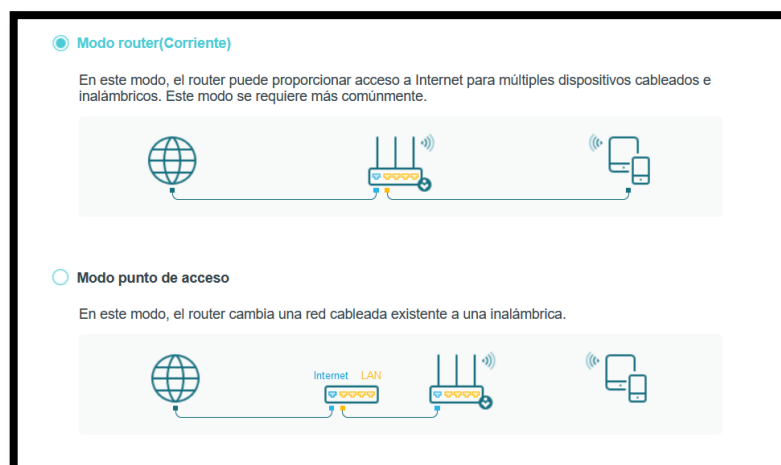
Configuración de red WIFI para invitados



Dentro de configuraciones avanzadas configuramos nuestro equipo en modo router que nos permite tener comunicaciones LAN y WIFI, la opción de configurarlo en modo router como se muestra en la Figura 19, es esencial para habilitar tanto la comunicación por LAN (red cableada) como por Wi-Fi (red inalámbrica), en este modo, el equipo actúa como un punto de conexión central que gestiona la asignación de direcciones IP, controla el acceso a Internet y conecta los dispositivos entre sí en la red local.

FIGURA 19

Configuración del equipo en modo Router

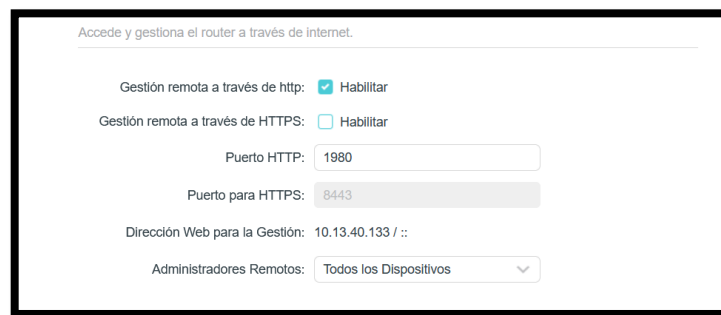


La configuración de administración remota mostrada en la Figura 20, en un router permite que el dispositivo sea administrado desde una ubicación externa, lo cual es útil en entornos corporativos o sucursales donde el personal técnico necesita acceso sin estar físicamente presente.

- Dentro del apartado de Configuraciones Avanzadas, selecciona la opción Sistema.
- Ir a la pestaña Administración y elegir Administración Remota.
- Marcar la casilla para activar esta función y colocar el puerto en 1980.

FIGURA 20

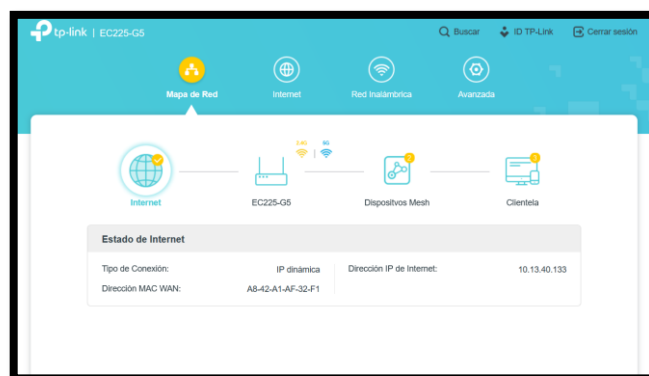
Configuración del router para designación del puerto de administración remota



Tras completar todas las configuraciones básicas del router, se procede a activar el servicio de internet, este proceso incluye la instalación de una conexión de fibra óptica, conectando el puerto asignado por el jefe técnico de la empresa a nuestra unidad receptora de fibra óptica bridge, un ícono de internet con una marca de verificación confirma que la conexión con el proveedor de servicios de internet (ISP) se ha establecido correctamente como se muestra en la Figura 21.

FIGURA 21

Conexión de internet establecida



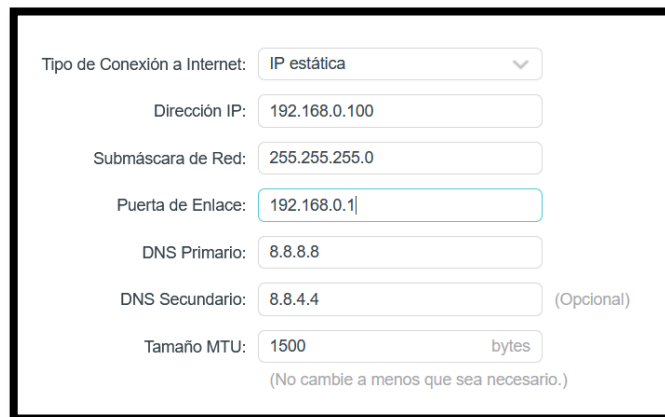
2.7.7. Configuración de los parámetros de conexión entre el software HIKLOOK y la red de internet

En primer lugar, en las configuraciones del router, dentro de la pestaña Red, en la opción LAN, identificar el puerto conectado al NVR y realizar los ajustes correspondientes., cambiamos la asignación de IP dinámica a IP estática para garantizar una comunicación estable con la cámara de videovigilancia, para este caso en particular, se asignará la dirección IP 192.168.0.100.

Además, se configuran los servidores DNS con los valores predeterminados: 8.8.8.8 como DNS primario y 8.8.4.4 como DNS secundario, ya que ofrecen una excelente velocidad y confiabilidad, estos ajustes aseguran una conectividad eficiente y estable para el sistema de videovigilancia, todos los parámetros establecidos los podemos observar en la Figura 22.

FIGURA 22

Configuración del puerto LAN del router para conexión con el NVR



The image shows a configuration window for the router's LAN settings. The 'Tipo de Conexión a Internet' is set to 'IP estática'. The 'Dirección IP' is 192.168.0.100, 'Submáscara de Red' is 255.255.255.0, and 'Puerta de Enlace' is 192.168.0.1. The 'DNS Primario' is 8.8.8.8 and 'DNS Secundario' is 8.8.4.4 (Opcional). The 'Tamaño MTU' is 1500 bytes. A note at the bottom states '(No cambie a menos que sea necesario.)'

Tipo de Conexión a Internet:	IP estática
Dirección IP:	192.168.0.100
Submáscara de Red:	255.255.255.0
Puerta de Enlace:	192.168.0.1
DNS Primario:	8.8.8.8
DNS Secundario:	8.8.4.4 (Opcional)
Tamaño MTU:	1500 bytes

(No cambie a menos que sea necesario.)

Una vez configurado el puerto LAN del router, se procede a establecer el enlace en el programa HIKLOOK, utilizando la dirección IP asignada en el router (192.168.0.100). Para ello, es fundamental seguir los siguientes pasos:

- Ejecutar el software HikLook en la computadora como se muestra en la Figura 23.
- Ir a la sección de Dispositivos o Administración de dispositivos.
- Seleccionar la opción Agregar Manualmente o + Agregar según la versión del software.

FIGURA 23

Ejecución del software HIKLOOK



Una vez que se realice los pasos anteriores es necesario completar los campos mostrados en la pestaña de configuración con los siguientes datos:

- Modo de conexión: IP/Domain mostrado en la Figura 24.
- Dirección IP: 192.168.0.100 (la asignada en el router), mostrado en la Figura 25.
- Puerto: Usualmente 8000 (puerto predeterminado del NVR), mostrado en la Figura 26.
- Nombre de usuario: Introducir el usuario configurado en el NVR (por defecto suele ser *admin*).
- Contraseña: Introducir la contraseña correspondiente al usuario del NVR.

FIGURA 24

Configuración del software HIKLOOK para conexión de red

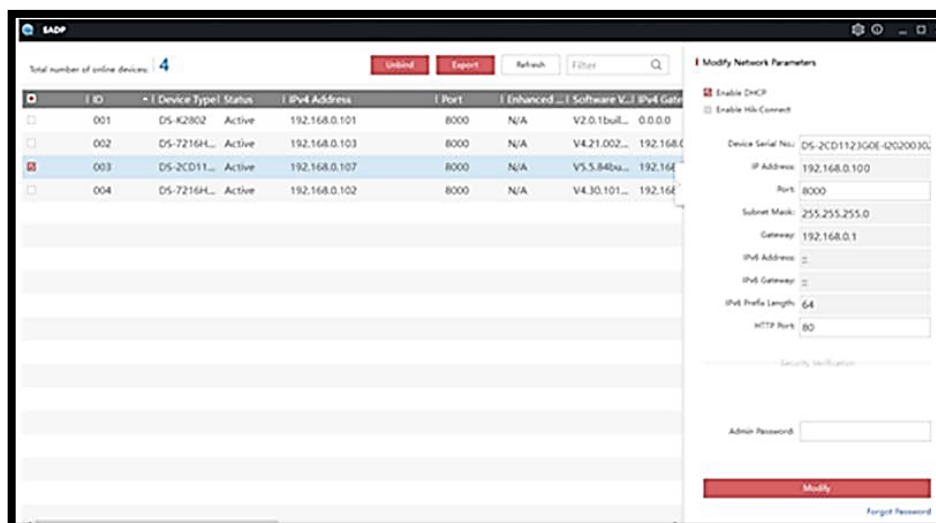
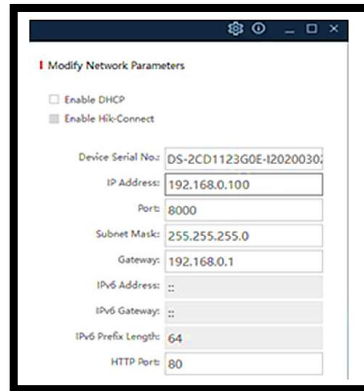


FIGURA 25

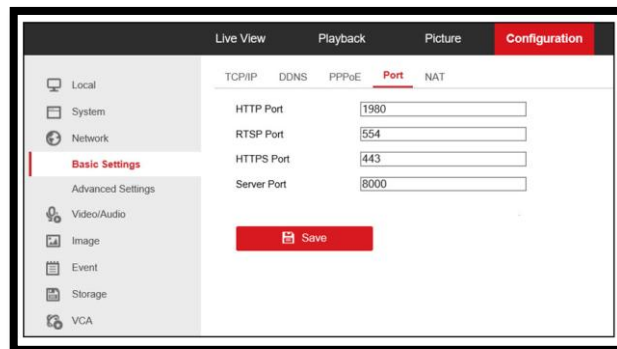
Configuración de parámetros de conexión de red



De la misma manera que en la configuración del router el protocolo de comunicación HTTP deberá ser configurado con el puerto 1980 como se muestra en la Figura 26.

FIGURA 26

Configuración del puerto de administración remota

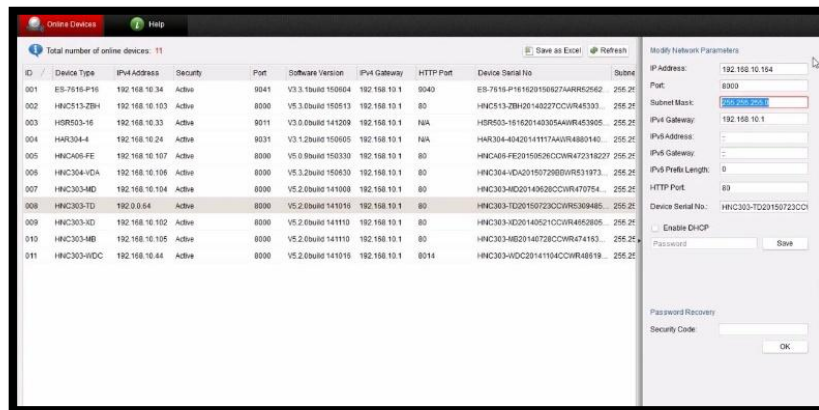


2.7.8. Activación de la cámara

Como se puede observar en la Figura 27, el programa SADP (Search Active Device Protocol) de Hikvision es una herramienta gratuita y muy útil para gestionar dispositivos de la marca, como cámaras IP, DVR y NVR, permite realizar tareas de configuración inicial y mantenimiento de forma rápida y sencilla, SADP detecta automáticamente todos los dispositivos Hikvision conectados en la misma red, también permite configurar la dirección IP, máscara de subred y puerta de enlace de los dispositivos sin necesidad de acceder a ellos directamente, además ayuda a activar dispositivos nuevos configurando una contraseña.

FIGURA 27

Activación de la cámara con SADP

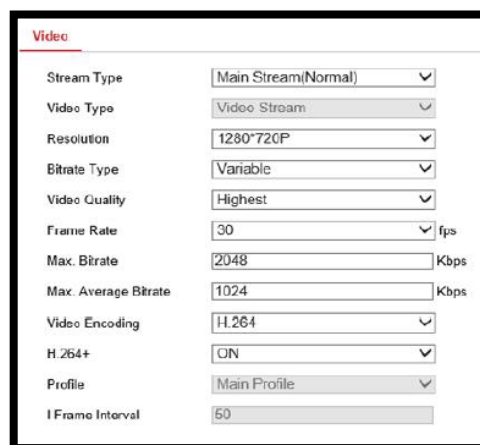


2.7.9. Configuración de los parámetros de tomas de imagen la cámara

La calidad del video por defecto está configurada en el canal principal con una resolución de 1280 x 720 píxeles (720P) como se muestra en la Figura 28, lo que garantiza imágenes de alta calidad, el sistema proporciona una tasa de cuadros de 30 fps (fotogramas por segundo), ofreciendo un video fluido, el flujo de datos que representa la cantidad de bits procesados por segundo se encuentra entre 1024 y 2048 Kbps, ajustándose para equilibrar calidad y eficiencia. La codificación de video utiliza el estándar H.264, que es conocido por su capacidad de ofrecer una excelente calidad de video mientras minimiza el consumo de ancho de banda.

FIGURA 28

Parámetros por defecto de la calidad de imagen

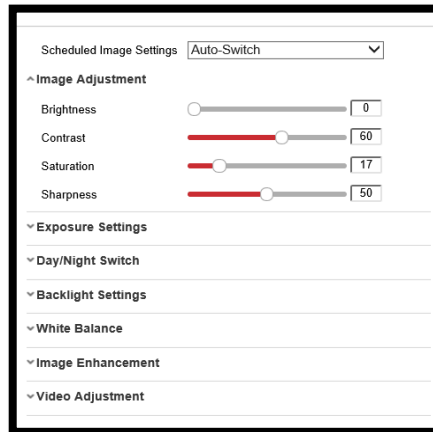


Los valores configurados para la cámara fueron determinados mediante diferentes pruebas in situ, tomando en cuenta las situaciones climáticas presentadas, considerando

la zona de principal interés que es la placa vehicular, esto para garantizar un reconocimiento óptimo a una distancia promedio de 4 metros desde la cámara hasta la zona de toma de imágenes, los ajustes de imagen presentados en la Figura 29, fueron configurados de la siguiente manera: Brillo: 0; Contraste: 60; Saturación: 17; Nitidez: 50

FIGURA 29

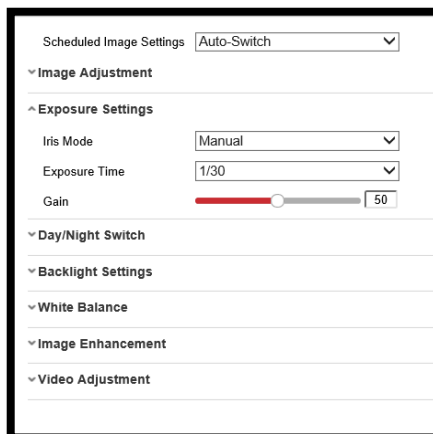
Ajustes de imagen para la captura de video y calidad



La configuración de la velocidad de obturación o tiempo de exposición como se muestra en la Figura 30, es crucial para capturar movimiento y asegurar nitidez en las imágenes., se ha ajustado este parámetro a un tiempo de exposición de 1/30 segundos, lo que permite la captura de imágenes nítidas y claras en tiempo real, además, se ha configurado una ganancia de 50 lo que mejora la visibilidad en condiciones de baja luz, sin comprometer la calidad de la imagen.

FIGURA 30

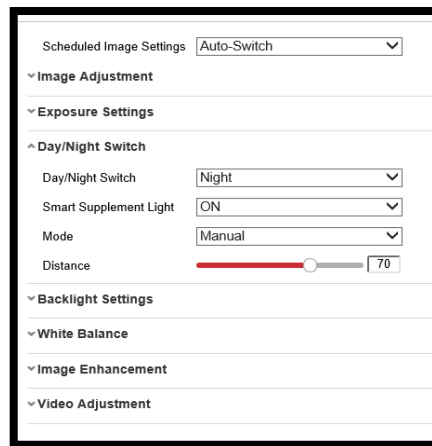
Configuración de la velocidad de obturación



La luz infrarroja de la cámara va a permanecer encendida las 24 horas del día con la capacidad de detectar luz entre los 400 y 700 nm de longitud como se puede observar en la Figura 31, para proporcionar visibilidad en entornos oscuros, además la función de compensación de luces altas se ha configurado con un valor de intensidad o sensibilidad a nivel de 20.

FIGURA 31

Configuración luz infrarroja



2.7.10. Capturas de imagen en tiempo real

Un programa en Python con OpenCV puede ser diseñado para capturar los frames de una cámara que utiliza codificación H.264 a 30 FPS y resolver problemas de control de flujo para evitar retardos o acumulación de frames en la cola de impresión.

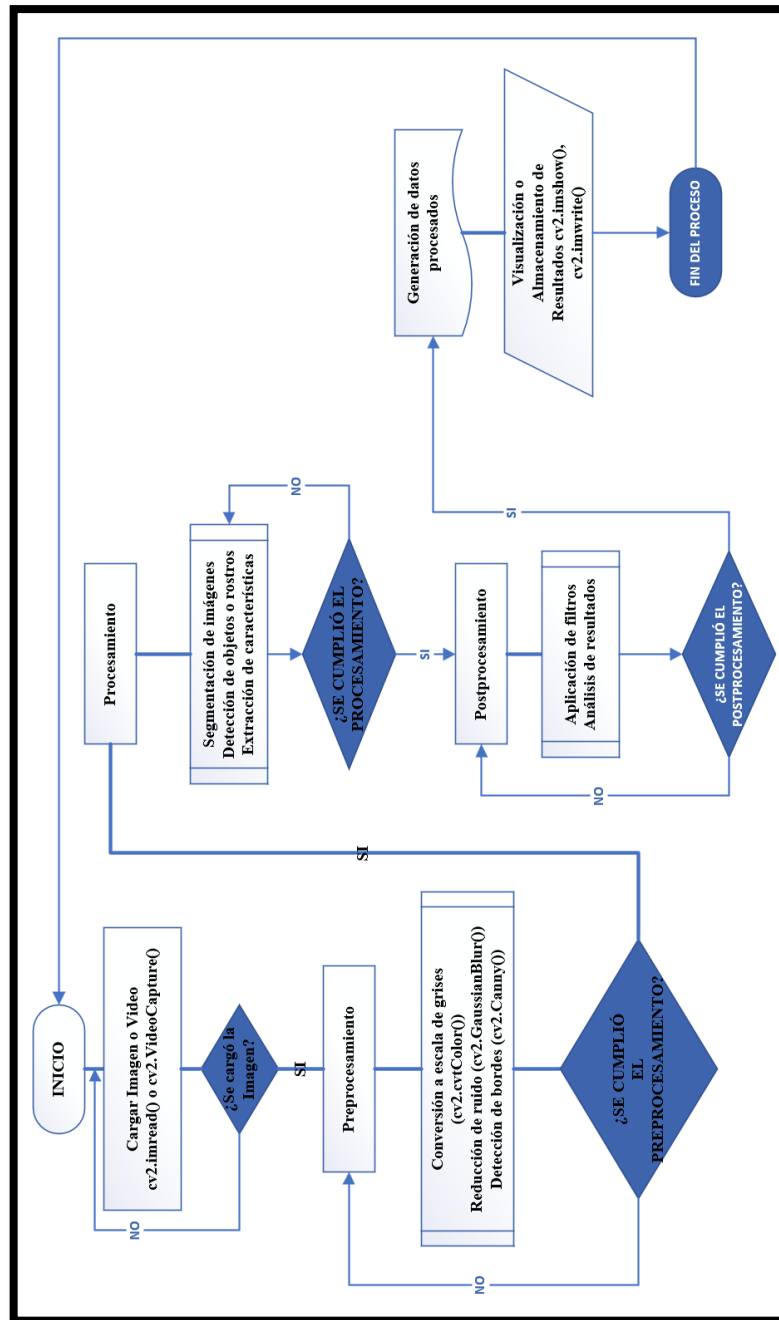
2.7.11. Open CV

En OpenCV, el comando VideoCapture(filename) permite abrir y procesar archivos de video, secuencias de imágenes o incluso transmisiones en vivo, dependiendo del valor ingresado en filename, OpenCV soporta diversos formatos y protocolos gracias a bibliotecas subyacentes como FFmpeg o GStreamer, que permiten manejar una amplia variedad de entradas. Se puede usar patrones para cargar una secuencia de imágenes como frames de un video, en la Figura 32 se muestra el diagrama de flujo del procesamiento de archivos de video mediante OpenCV,

Ejemplo: img_04.jpg cargará imágenes como img_0001.jpg, img_0002.jpg, esto ya depende de la configuración establecida por el diseñador.

FIGURA 32

Diagrama de flujo de funcionamiento de OpenCV



2.7.12. Diseño del programa y red de entrenamiento neuronal

Para el desarrollo de este proyecto se va a utilizar una tarjeta NVIDIA Jetson nano, mediante la red detectNet, la cual permite detectar objetos en imágenes o videos, en este caso, la red se entrena específicamente para reconocer únicamente placas vehiculares en

un video en vivo, para lograr que detectNet detecte únicamente placas vehiculares en un video en vivo, es necesario ajustar y optimizar la red mediante transferencia de aprendizaje con un conjunto de datos específico, a continuación, se describe los pasos necesarios y las herramientas para implementar la red detectNet en el dispositivo NVIDIA Jetson nano.

Preparación del Conjunto de Datos

- Captura imágenes o frames de video en vivo que contengan placas vehiculares en diversas condiciones:
- Diferentes ángulos y distancias.
- Iluminación variada (día/noche).
- Placas con distintas posiciones y tamaños.

Etiquetar manualmente las imágenes usando herramientas como:

- LabelImg (para generar anotaciones en formato Pascal VOC).

Guardar las etiquetas en un formato compatible como:

- Pascal VOC.

Transferencia de Aprendizaje con detectNet

- Utilizar un modelo preentrenado como SSD-MobileNet (optimizado para Jetson).
- Aplicar transferencia de aprendizaje para ajustar el modelo a detectar solo placas vehiculares.

Pasos con PyTorch:

- Cargar un modelo preentrenado en PyTorch.
- Reemplazar la última capa (clasificación) para que solo reconozca la clase "license_plate".
- Entrenar el modelo con tu conjunto de datos etiquetado.
- Exportar el modelo en formato ONNX

Despliegue del Modelo en Jetson

- Utiliza el repositorio Hello AI World para ejecutar el modelo detectNet optimizado en video en vivo.

Se realiza una grabación de la entrada de vehículos al parqueadero durante cuatro semanas, para recolectar un conjunto de imágenes que serán usadas para entrenar la red neuronal, la idea es capturar un número significativo de imágenes representativas de las condiciones de entrada de los vehículos, y usar herramientas de procesamiento de video en OpenCV, como el comando Video Writer, para almacenar estas imágenes en un archivo de video.

En lugar de grabar todo el video en un solo archivo de gran tamaño, se divide la grabación en múltiples archivos como se puede observar en la Figura 33, por ejemplo, uno por día o por hora, esto puede hacerse modificando el nombre del archivo en cada iteración, agregando un sufijo como:

entrada_vehiculos_dia1.avi, entrada_vehiculos_dia2.avi, etc.

Ejemplo del código final:

```
cv2.VideoWriter('vid1.avi', cv2.VideoWriter_fourcc(*'XVID'), 30.0, (1280, 720))
```

FIGURA 33

Obtención de videos



2.7.13. Dividir videos en frames

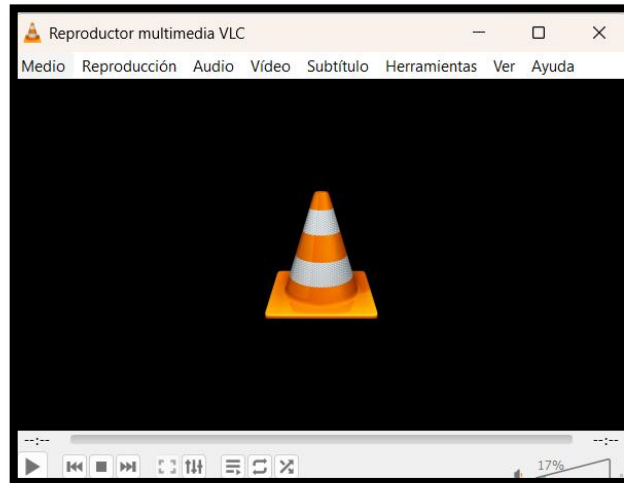
Para dividir un video en sus distintos frames se puede utilizar VLC Media Player, ya que VLC permite extraer frames de un video sin necesidad de escribir código complejo, esta capacidad de extraer frames resulta ser útil en una variedad de aplicaciones, como el análisis de video, la creación de bases de datos de imágenes para entrenamiento de redes neuronales, o simplemente la creación de una secuencia de imágenes a partir de un video.

Pasos para Extraer Frames con VLC Media Player

1. Abrir VLC Media Player como se muestra en la Figura 34.

FIGURA 34

Pantalla principal VLC

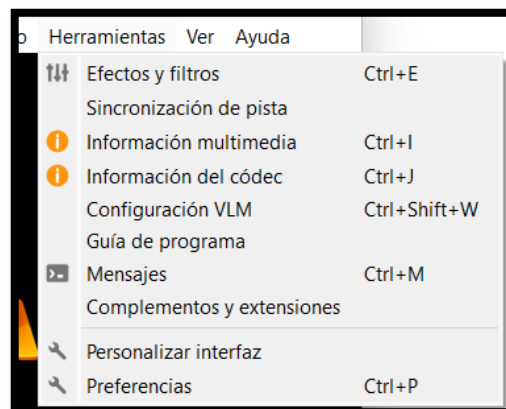


2. Acceder a las Preferencias de VLC

- Abre VLC Media Player.
- Ve al menú Herramientas y selecciona Preferencias (o simplemente presiona Ctrl + P) como se observa en la Figura 35.

FIGURA 35

Menú herramientas VLC



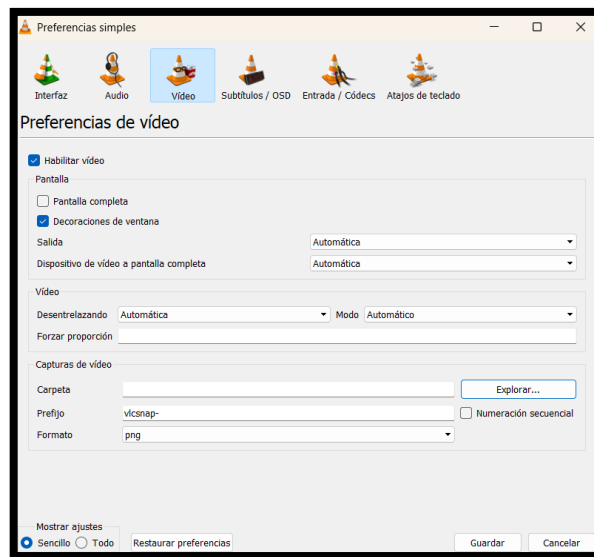
3. Configurar la Carpeta de Salida para los Frames:

- En la ventana de preferencias, selecciona la pestaña "Video".

- En la sección de "Captura de pantalla", busca la opción "explorar o Directorio de capturas", aquí se define la carpeta donde se almacenarán los frames extraídos del video como se puede observar en el icono marcado (Explorar) de la Figura 36.

FIGURA 36

Configuración preferencias de video

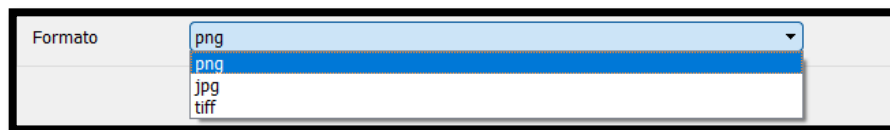


4. Establecer el Formato de Captura de Pantalla:

En el mismo apartado como se observa en la Figura 37, también puedes elegir el formato de archivo para los frames (por ejemplo, PNG, JPEG, TIFF). El formato recomendado para imágenes de alta calidad es PNG.

FIGURA 37

Configuración de capturas de pantalla en formato png.

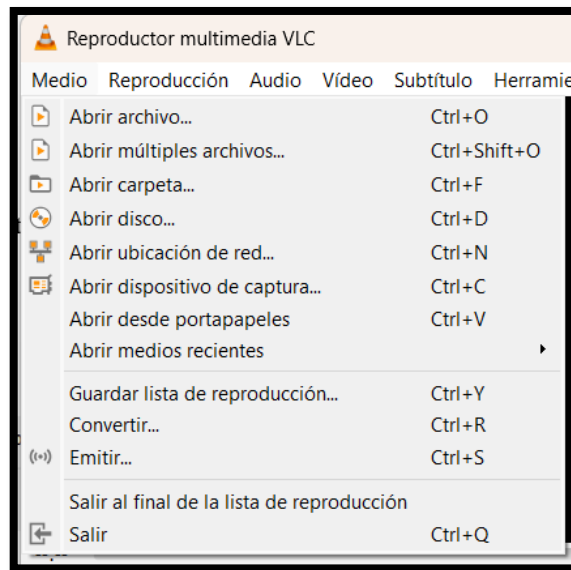


5. Reproducir el Video en VLC:

Como se observa en la Figura 38, se puede abrir el video desde el menú Medio > Abrir archivo o presiona Ctrl + O y seleccionar el video desde la computadora.

FIGURA 38

Reproducción de un video en VLC

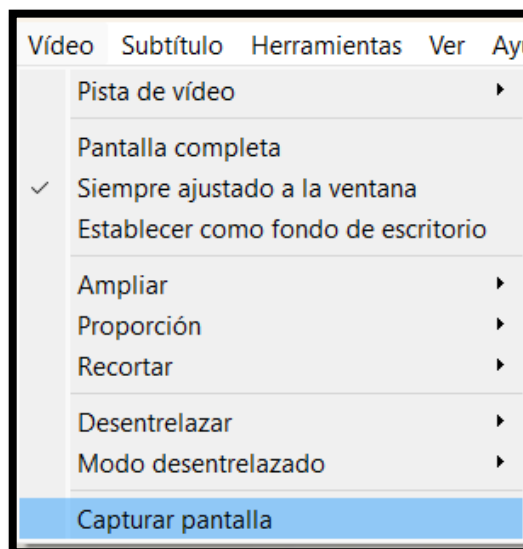


6. Capturar Frames Manualmente:

Mientras el video se reproduce, se puede hacer una captura de pantalla manualmente haciendo clic en el botón "Captura de pantalla" en la barra de herramientas de VLC como se muestra en la Figura 39. o presionar Shift + S (en Windows/Linux) o Cmd + Alt + S (en macOS). Esto guardará el frame actual en la carpeta que se configuró anteriormente.

FIGURA 39

Captura de pantalla manualmente



7. Automatizar la Extracción de Frames con una Función de VLC:

Para extraer frames automáticamente a intervalos regulares, se debe seguir estos pasos:

- Seleccionar Preferencias > Todo (en la esquina inferior izquierda) como se muestra en la Figura 40 y Figura 41.

FIGURA 40

Preferencias en la barra de herramientas

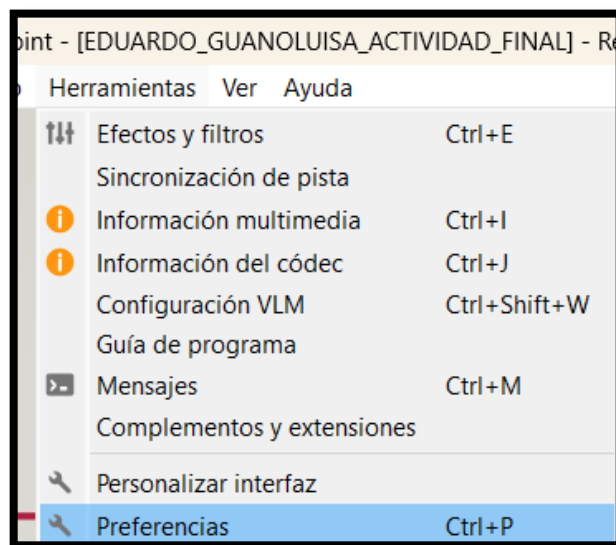
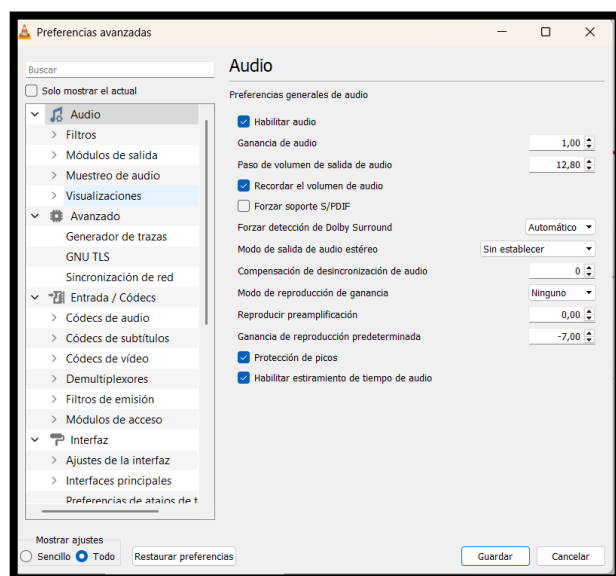


FIGURA 41

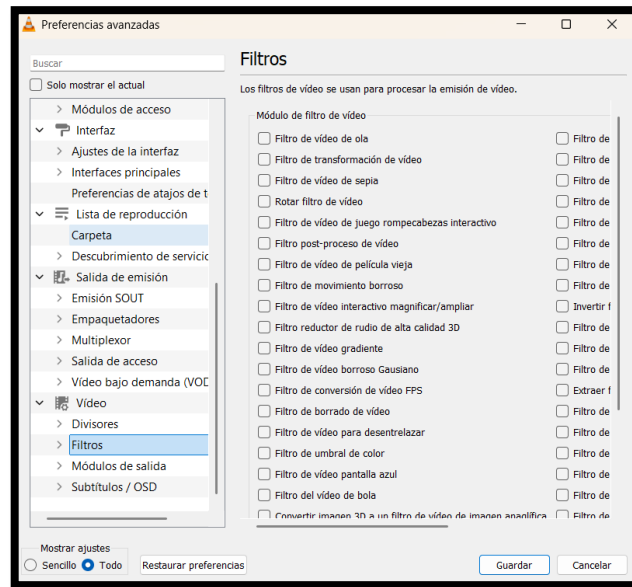
Seleccionar todo en la pestaña de preferencias



- En el menú de preferencias avanzadas, seleccionar "Video" y luego "Filtros" como se observa en la Figura 42.

FIGURA 42

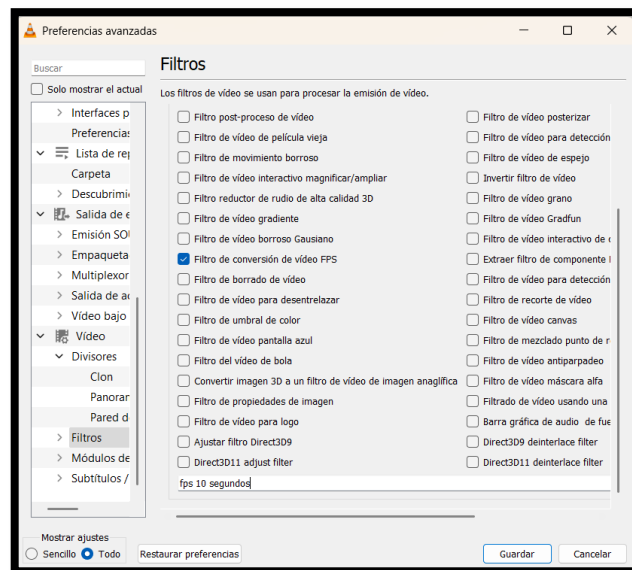
Pestaña video, opción filtros



- Activar el filtro de "Captura de imagen FPS" como se muestra en la Figura 43.

FIGURA 43

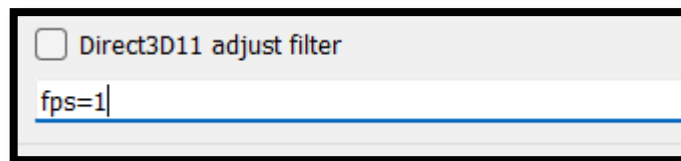
Filtro captura de imagen FPS



- En la sección de "Captura de pantalla FPS", se podrá definir el intervalo entre capturas y otros parámetros de salida.
- Si el video tiene 30 FPS y se configura Captura de pantalla FPS = 1 como se observa en la Figura 44, VLC tomará una captura cada segundo.
- Si se configura Captura de pantalla FPS = 0.5, VLC tomará una captura cada dos segundos

FIGURA 44

Definición del tiempo de captura de pantalla automático



- Después de activar el filtro, VLC comenzará a tomar capturas de pantalla cada vez que el video avance a un intervalo determinado como se puede observar en la Figura 45.
- Las capturas se guardarán automáticamente en el directorio especificado.
- Si se necesita intervalos de tiempo exactos en lugar de fotogramas, ajustar el ratio según la velocidad de cuadros por segundo (FPS) del video.

FIGURA 45

Capturas de pantalla automáticas con VLC



2.7.14. Determinación de las zonas de interés

Para entrenar una red neuronal como DetectNet, es esencial disponer de un conjunto de datos previamente etiquetados que permita identificar y localizar los objetos dentro de las imágenes, estos datos deben seguir un formato estandarizado que facilite la organización y procesamiento de la información durante el entrenamiento del modelo. Uno de los formatos más utilizados para este propósito es Pascal/VOC, un estándar ampliamente adoptado en tareas de visión por computadora, este formato estructura las etiquetas en archivos XML, asociando cada imagen con la información necesaria sobre los objetos detectados, como sus categorías y las coordenadas de los cuadros delimitadores.

Instalación de LabelImg, instalarlo siguiendo estos pasos:

- Si se usa pip, instalar LabelImg con:

```
pip install labelImg
```

Ejecutar este comando en la terminal, esto mostrará la versión de pip instalada en el sistema:

```
pip --version
```

Actualizar pip para asegurarse de tener la versión más reciente:

```
python -m pip install --upgrade pip
```

Instalar un paquete:

Por ejemplo, para instalar LabelImg:

```
pip install labelImg
```

Una vez que hemos confirmado que *pip* está correctamente instalado en el sistema, junto con la herramienta *labeling*, lo ejecutamos con:

```
labelImg
```

Abrir las imágenes:

Una vez abierto LabelImg, seleccionar la carpeta donde están las imágenes que se desea etiquetar.

Seleccionar formato Pascal/VOC:

Antes de empezar a etiquetar, asegurarse de configurar el formato de salida a Pascal/VOC:

- Seleccionar Settings > Output Format.
- Seleccionar PascalVOC.

Una vez etiquetadas las imágenes, se convierte en formato compatible con el modelo, DetectNet utiliza datos en formato TFRecord, para comenzar con el proceso de preparación del archivo de configuración, es necesario abrirlo utilizando el editor de texto, en este caso:

predefined_classes.txt

Una vez abierto el archivo identificar y eliminar todas las clases predefinidas o existentes que no sean relevantes para el proyecto, posteriormente se debe agregar la clase específica que se desea etiquetar, como se puede observar en la Figura 46.

FIGURA 46

Nombre de la clase etiquetada



Para comenzar a trabajar con un nuevo conjunto de datos en LabelImg, es necesario abrir un directorio específico donde se almacenarán las imágenes y sus respectivas etiquetas. A continuación, se detallan los pasos para hacerlo:

- Abrir LabelImg.
- Seleccionar un nuevo directorio:
- En la interfaz principal de LabelImg, localiza el botón o la opción denominada "Open Dir" o "Abrir Directorio" en la barra de herramientas como se muestra en la Figura 47.
- Haz clic en esta opción para abrir un cuadro de diálogo que te permitirá navegar por las carpetas de tu sistema.
- Seleccionar la carpeta de trabajo como se muestra en la Figura 48.

FIGURA 47

Abrir el directorio en LabelImg

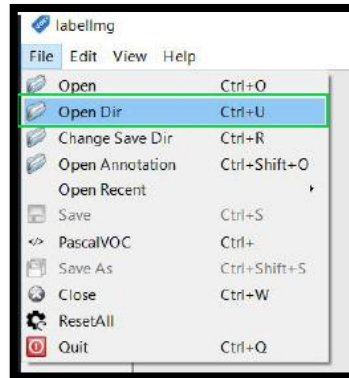
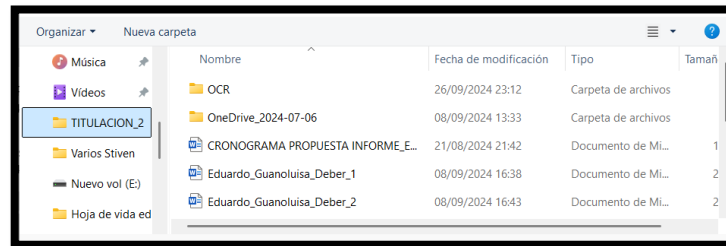


FIGURA 48

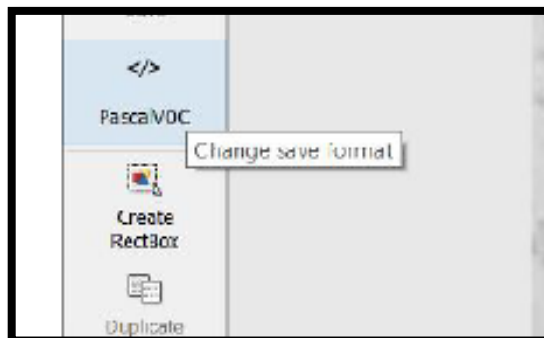
Selección de carpeta de trabajo para el directorio en LabelImg



Verificar que el formato de trabajo se encuentre configurado en Pascal/VOC como se puede observar en la Figura 49. Para evitar tener que cambiar el formato cada vez que se abre la aplicación, revisar si LabelImg guarda esta configuración por defecto, si no lo hace, verificar la documentación para configurar el formato deseado de manera predeterminada.

FIGURA 49

Configuración formato Pascal/VOC



Hacer clic en el botón Create RectBox para comenzar, luego, utilizar el cursor para delimitar el área de interés en la imagen o el documento, una vez definido el recuadro, seleccionar o asignar la etiqueta correspondiente que describa de manera precisa el contenido o la categoría del área seleccionada como se puede observar en la Figura 50 y Figura 51.

FIGURA 50

Etiqueta de la zona de interés

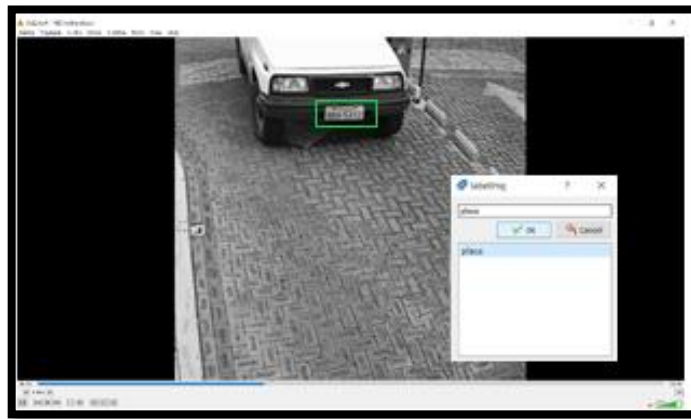
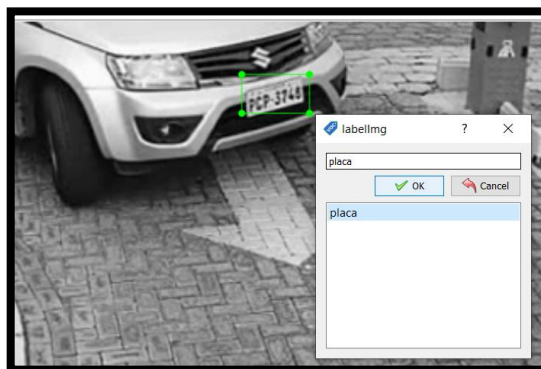


FIGURA 51


Etiqueta de la zona de interés



Guardar y confirmar que el archivo .xml se encuentra en el directorio especificado como se muestra en la Figura 52, esto asegura que los datos han sido generados o guardados correctamente y están disponibles para su uso o validación.

FIGURA 52

Información guardada en el archivo .xml



```
img(1).xml - Notepad
File Edit Format View Help
<annotation>
  <folder>pljpg</folder>
  <filename>img(1).jpg</filename>
  <path>D:\pythonchris\yolo2pascal\pljpg\img(1).jpg</path>
  <source>
    <database>Unknown</database>
  </source>
  <size>
    <width>2720</width>
    <height>1520</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>PLACA</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>1265</xmin>
      <ymin>609</ymin>
      <xmax>1412</xmax>
      <ymax>709</ymax>
    </bndbox>
  </object>
</annotation>
```

Después de completar los pasos anteriores, se repitió el proceso con cada una de las imágenes capturadas. Al finalizar, se etiquetaron un total de 9,600 imágenes, distribuidas de la siguiente manera:

- Entrenamiento: 75% (7,200 imágenes).
- Pruebas (test): 15% (1,440 imágenes).
- Validación: 10% (960 imágenes).

2.7.15. Configuración de NVIDIA Jetson Nano

La configuración de una NVIDIA Jetson Nano implica preparar tanto el hardware como el software para su uso, a continuación, se describe el proceso paso a paso:

1. Preparar el Hardware

- NVIDIA Jetson Nano Developer Kit.
- Tarjeta microSD de al menos 32 GB (se recomienda de clase A1 o superior).
- Fuente de alimentación de 5V/4A o un adaptador micro-USB
- Teclado, ratón y monitor con cable HDMI.
- Acceso a internet (Ethernet o adaptador Wi-Fi compatible).

Montar el dispositivo:

- Insertar la tarjeta microSD en la ranura del Jetson Nano.
- Conectar el teclado, ratón y monitor.
- Conectar la fuente de alimentación.

FIGURA 53

Preparación del HARDWARE



En la Figura 53 se aprecia la disposición operativa de la tarjeta NVIDIA Jetson Nano, la cual está completamente preparada para su funcionamiento, este dispositivo se encuentra ubicado junto al router de conexiones a internet, lo que garantiza una comunicación estable y eficiente mediante un cable Ethernet que conecta directamente la tarjeta a la red. Además, se han conectado los dispositivos periféricos como pantalla, teclado, ratón junto con la fuente de alimentación, de 5V/4A con conector de barril (5.5 mm x 2.1 mm).

2. Preparar el Software

Descargar el Programa Etcher

Abrir el navegador web y acceder al sitio oficial de balenaEtcher para descargar el programa: <https://www.balena.io/etcher/>

3. Descargar la Imagen de la Tarjeta SD

- Acceder al sitio oficial de NVIDIA Jetson Nano: [Jetson Download Center](#).
- Buscar la sección correspondiente al Jetson Nano Developer Kit.
- Descargar la imagen SD más reciente para tu modelo de Jetson Nano.
- El archivo descargado estará comprimido en formato .zip, descomprimir usando herramientas como WinRAR o 7-Zip.

4. Flashear la Imagen en la Tarjeta SD

Insertar una tarjeta microSD (de al menos 32 GB, preferiblemente clase A1) en la computadora utilizando un lector de tarjetas.

Abrir balenaEtcher y seguir estos pasos:

- Haz clic en Flash from file y seleccionar el archivo de imagen .img descargado.
- Haz clic en Select target y elegir la tarjeta microSD.
- Haz clic en Flash! para iniciar el proceso como se muestra en la Figura 54.
- Esperar a que Etcher complete el flasheo, una vez terminado retirar la tarjeta microSD de manera segura como se observa en la Figura 55.

FIGURA 54

Pantalla principal programa ETCHER, opción Flash para completar el proceso

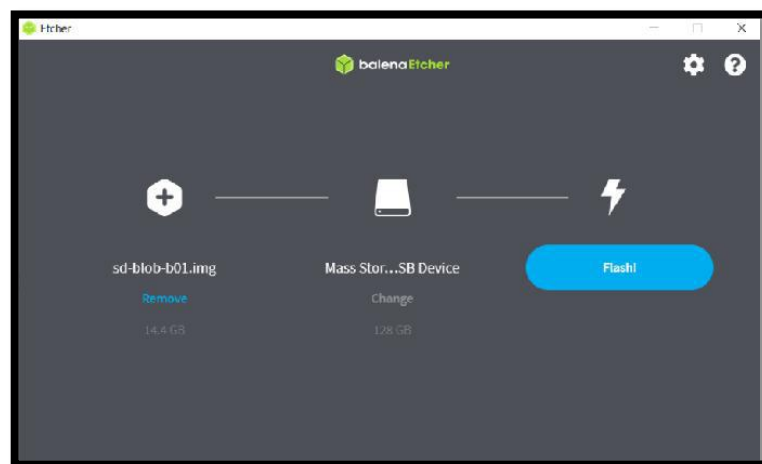
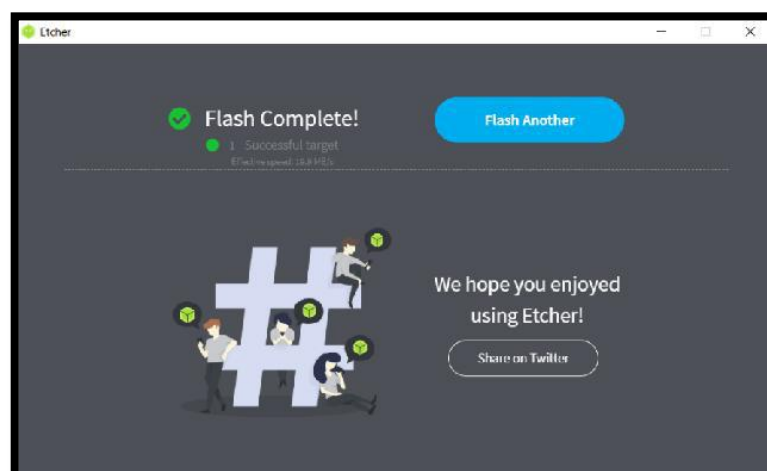


FIGURA 55

Mensaje de Flash completado en el programa ETCHER



2.7.16. Configuración inicial del sistema operativo en la pantalla principal Jetson

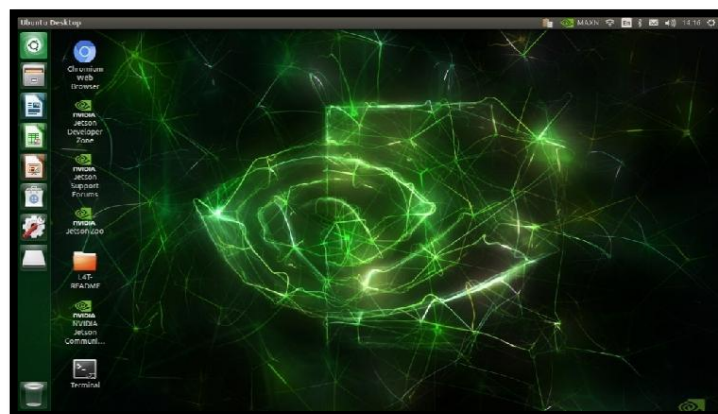
Cuando se arranque la Jetson Nano por primera vez, aparecerá una serie de pasos de configuración:

- Seleccionar el idioma preferido, en este caso se selecciona español y hacer clic en "Siguiete".
- Configurar la distribución del teclado, por facilidad se configura en Español – Latinoamericano.
- Seleccionar la zona horaria acorde a la ubicación en el mapa, Ecuador se encuentra en la zona horaria ECT: UTC-5, en Ecuador no se utiliza el horario de verano, por lo que esta zona horaria permanece fija durante todo el año.
- Crear una cuenta con un nombre de usuario y contraseña personalizados, considerando niveles de dificultad.
- Conexión mediante Ethernet lo cual permite una conexión automática a Internet.
- Seleccionar si se desea habilitar el modo Max-Q (ahorro de energía) o Max-P (máximo rendimiento). En este caso se elige Max-P debido a que se trabaja con una fuente de alimentación estable (5V/4A).

Después de la configuración inicial, el sistema arrancará automáticamente en el escritorio de Ubuntu, se visualizará la interfaz gráfica, que es un entorno basado en LXDE o similar como se puede observar en la Figura 56.

FIGURA 56

Escritorio, pantalla principal de la tarjeta JETSON NANO



2.7.17. Entrenamiento de detectnet

Para continuidad del proyecto es muy importante hacer un entrenamiento personalizado con DetectNet mediante el repositorio de Jetson Inference, lo cual permite crear modelos de detección de objetos adaptados a diferentes necesidades.

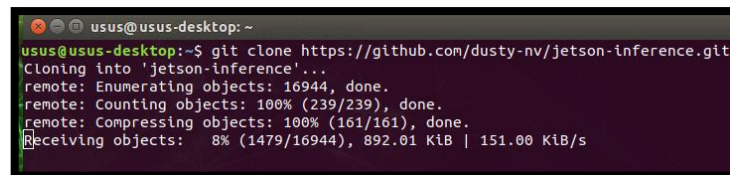
- Abrir un terminal en la pantalla de Jetson Nano.
- Clonar el repositorio oficial de Jetson Inference como se observa en la Figura 57.

Código

```
git clone https://github.com/dusty-nv/jetson-inference.git
```

FIGURA 57

Terminal de Jetson Nano con el comando git clone



```
usus@usus-desktop: ~  
usus@usus-desktop:~$ git clone https://github.com/dusty-nv/jetson-inference.git  
Cloning into 'jetson-inference' ...  
remote: Enumerating objects: 16944, done.  
remote: Counting objects: 100% (239/239), done.  
remote: Compressing objects: 100% (161/161), done.  
Receiving objects: 8% (1479/16944), 892.01 KiB | 151.00 KiB/s
```

Código

```
cd jetson-inference
```

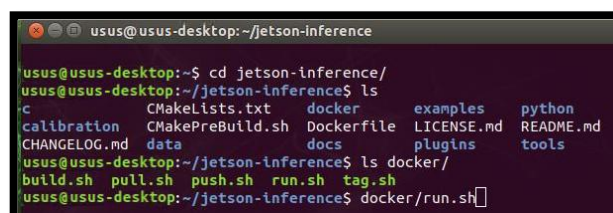
Ejecutar el script de configuración incluido para instalar dependencias y compilar el proyecto, esto configurará un contenedor Docker con todo lo necesario para trabajar con el proyecto como se puede observar en la Figura 58.

Código

```
sudo ./docker/run.sh
```

FIGURA 58

Comando Jetson Nano ejecutado el Docker/run.sh



```
usus@usus-desktop: ~/jetson-inference  
usus@usus-desktop:~$ cd jetson-inference/  
usus@usus-desktop:~/jetson-inference$ ls  
c          CMakeLists.txt    docker      examples  python  
calibration CMakePreBuild.sh Dockerfile  LICENSE.md README.md  
CHANGELOG.md data              docs        plugins   tools  
usus@usus-desktop:~/jetson-inference$ ls docker/  
build.sh pull.sh push.sh run.sh tag.sh  
usus@usus-desktop:~/jetson-inference$ docker/run.sh
```

El Model Downloader de Jetson Inference permite descargar modelos preentrenados que pueden ser utilizados para transfer learning o directamente para tareas de inferencia.

Asegurarse de estar en el directorio raíz del proyecto jetson-inference:

```
cd ~/jetson-inference
```

Ejecutar el script del Model Downloader:

```
./tools/download-models.sh
```

El Model Downloader presenta una lista de modelos preentrenados disponibles. Algunos de ellos incluyen:

Clasificación de Imágenes:

- ResNet-18
- GoogLeNet
- VGG-16

Detección de Objetos (SSD/DetectNet):

- SSD-Mobilenet-V1
- SSD-Inception-V2

Segmentación de Imágenes:

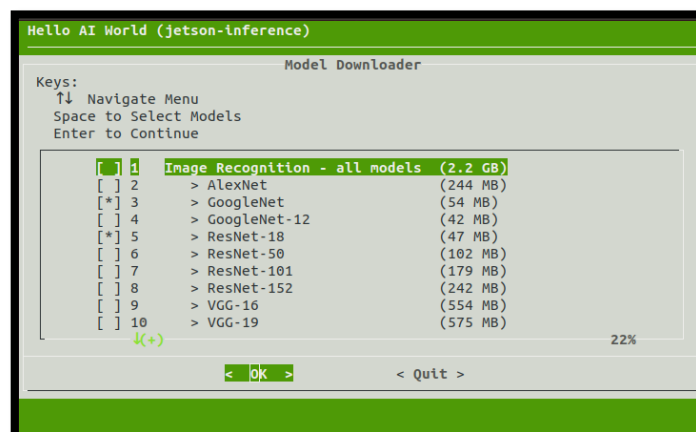
- FCN-ResNet-18
- DeepLab-V3

El script mostrará opciones y permitirá seleccionar los modelos escribiendo el número correspondiente o seleccionando todos los modelos.

- En este caso vamos a seleccionar All models en nuestra interfaz el número “1” como se muestra en la Figura 59.

FIGURA 59

Interfaz de Model Downloader selección all models



Para pegar la carpeta que contiene imágenes y etiquetas (con la distribución de 75% para entrenamiento, 15% para validación y 10% para prueba) en el directorio adecuado dentro de `jetson-inference/python/training/detection/ssd/data`, seguir estos pasos:

2.7.18. Asegurarse de la Estructura de las Carpetas

Primero, verificar que los datos estén organizados correctamente, se deberá tener algo como esto:

```
dataset/  
├── ImageSets/  
│   ├── image1.jpg  
│   ├── image2.jpg  
│   └── ...  
├── Annotations/  
│   ├── image1.xml  
│   ├── image2.xml  
│   └── ...  
└── Main/  
    ├── train.txt  
    ├── val.txt  
    ├── test.txt  
    └── trainval.txt...
```

Donde:

- `ImageSets/` contiene las imágenes.
- `Annotations/` contiene los archivos de etiquetas (por ejemplo, en formato `.xml` para Pascal VOC).
- `Main/` contiene los archivos de texto que definen qué imágenes van a cada conjunto (entrenamiento, validación, prueba).

Copiar los Datos a la Carpeta `data` de Jetson Inference

- Abrir un terminal en Jetson Nano.
- Navegar a la carpeta donde se tiene el dominio del conjunto de datos, si está en el directorio `~/dataset`, usar:

```
cd ~/dataset
```

- Copiar las carpetas de datos a:

```
jetson-inference/python/training/detection/ssd/data.
```

- Asegurarse de que el directorio de destino ya existe:

```
cp -r Images/ Annotations/ Main/ ~/jetson-inference/python/training/detection/ssd/data/
```

- Esto copiará las carpetas Images/, Annotations/ y Main/ al directorio data dentro de jetson-inference.

Una vez copiado el contenido, navegar al directorio de data para verificar que todo esté en su lugar:

```
cd ~/jetson-inference/python/training/detection/ssd/data
```

```
ls
```

- Se debería observar las carpetas ImageSets/, Annotations/, JPEGImages y Main/ en este directorio.
- Annotations/: Contiene los 9,600 archivos de etiquetas en formato .xml como se observa en la Figura 60.

FIGURA 60

Archivos formato .xml



Para distribuir automáticamente las imágenes y etiquetas en proporciones específicas (75% para entrenamiento, 15% para prueba y 10% para validación) utilizando los archivos dentro de la carpeta Main.

La carpeta Main contiene archivos de texto que indican las imágenes disponibles y cómo deben ser distribuidas, esta carpeta denominada Main se encuentra dentro de ImageSets/, esto contiene:

- train.txt: Lista de imágenes para entrenamiento.
- val.txt: Lista de imágenes para validación.
- test.txt: Lista de imágenes para prueba.
- trainval.txt: Lista combinada de imágenes para entrenamiento y validación.
- Usar Python para leer los archivos en Main y distribuir las imágenes y anotaciones en las carpetas correspondientes.

Asegurarse de que los datos estén organizados así:

dataset/

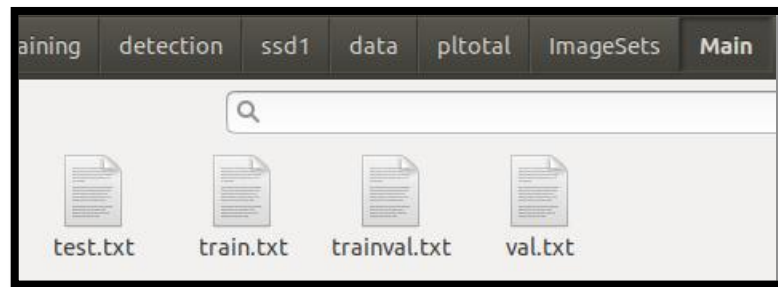
```
|— JPEGImages/      # Contiene las 9,600 imágenes .jpg
|— Labels.txt/      # Contiene las etiquetas con el nombre PLACA
|— Annotations/    # Contendrá los archivos .xml de las etiquetas
|— ImageSets/      # Contiene la carpeta MAIN
  |— Main/          # Contendrá train.txt, val.txt, test.txt
  |— train.txt      # Lista de imágenes para entrenamiento
  |— val.txt        # Lista de imágenes para validación
  |— test.txt       # Lista de imágenes para prueba
  └— trainval.txt   # Lista combinada de entrenamiento y validación
```

Guardar este script en un archivo, por ejemplo, distribute_data.py:

El script creará las siguientes carpetas organizadas como se observa en la Figura 61.

FIGURA 61

Script con las carpetas organizadas



La carpeta llamada JPEGImages/ que contiene 9,600 imágenes en formato .jpg se necesita integrarlas en el flujo de trabajo de entrenamiento de Jetson Inference (en el directorio jetson-inference/python/training/detection/ssd/data), seguir estos pasos para organizar los datos correctamente y distribuirlos para entrenamiento, validación y prueba:

2.7.19. Organizar las Imágenes y las Etiquetas

Primero, asegurarse de que las imágenes en la carpeta JPEGImages/ estén acompañadas de sus etiquetas en formato compatible (generalmente **Pascal VOC**, con archivos .xml para cada imagen), además, verificar de tener los archivos train.txt, val.txt y test.txt en la carpeta Main/, que especifican qué imágenes irán a cada conjunto de datos tal como se puede verificar en la Figura 62.

FIGURA 62

9,600 imágenes en formato jpg



Labels.txt

Contiene el nombre de la etiqueta “PLACA”

- Comenzar el entrenamiento con los datos personalizados:

Código

```
cd ../python/training/detection/ssdpython3 train_ssd.py \  
  --dataset-type=voc \  
  --data=/path/to/my-dataset \  
  --model-dir=models/my-model \  
  --batch-size=4 \  
  --epochs=50 \  
  --pretrained-ssd=models/mobilenet-v1-ssd-mp-0_675.pth
```

Evaluar el Modelo

Después de entrenar, se evalúa el modelo para comprobar su precisión.

Código

```
python3 eval_ssd.py \  
  --dataset-type=voc \  
  --data=/path/to/my-dataset \  
  --model-dir=models/my-model \  
  --trained-model=models/my-model/mb1-ssd-Epoch-50-Loss-2.000.pth
```

Desplegar el Modelo

Exportar el modelo entrenado a formato ONNX:

Código

```
python3 export_onnx.py \  
  --model-dir=models/my-model \  
  --trained-model=models/my-model/mb1-ssd-Epoch-50-Loss-2.000.pth
```

- Usar DeepStream o el código de Jetson Inference para realizar la inferencia en la Jetson Nano.

2.7.20. Entrenamiento personalizado de la red neuronal

Abrir el directorio `jetson-inference/python/training/detection/ssd` en la Jetson Nano, siguiendo estos pasos:

- Comprobar de estar en el directorio raíz del proyecto `jetson-inference`.
- Navegar al directorio `ssd` donde se encuentran los scripts de entrenamiento.
- Verificar que está en el directorio correcto, debería ver archivos como:
- `train_ssd.py`
- `eval_ssd.py`
- `export_onnx.py`
- Otros archivos relacionados con la detección SSD.

Para el entrenamiento del modelo SSD en la Jetson Nano ejecutar el comando de entrenamiento neuronal, este comando entrenará un modelo SSD utilizando los datos en el directorio `data/pltotal`, especificando el número de épocas, el tamaño de lote y otros parámetros mediante el comando de entrenamiento:

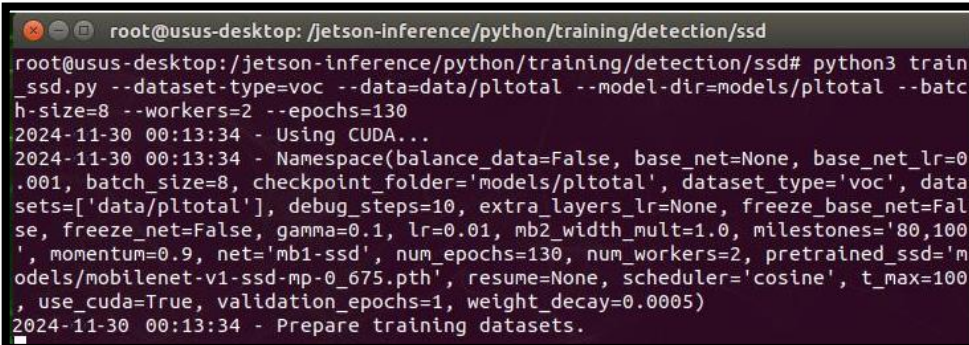
```
python3 python/training/detection/ssd/train_ssd.py --dataset-type=voc --
data=data/pltotal --model-dir=models/pltotal --batch-size=8 --workers=2 --
epochs=130
```

Monitorear el Progreso

Una vez que ejecutado el comando el proceso de entrenamiento comenzará, durante el entrenamiento el terminal mostrará información sobre la pérdida (loss), el progreso de las épocas y otros detalles importantes como observamos en la figura 63.

FIGURA 63

Jetson Nano ejecutando la red de entrenamiento neuronal



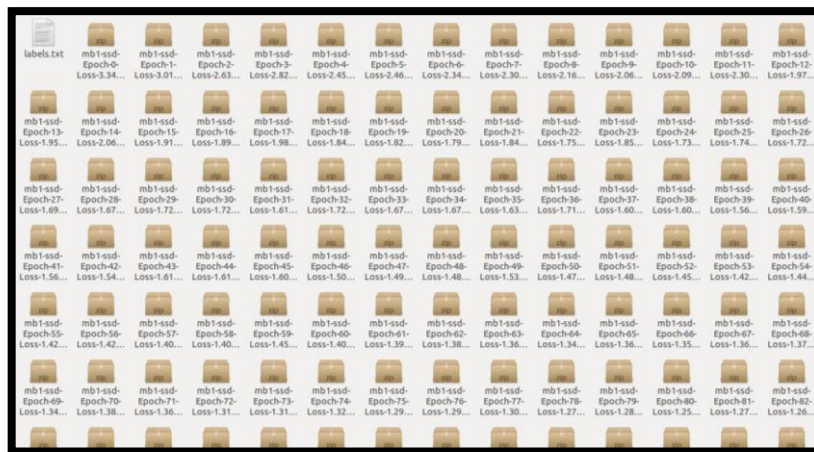
```
root@usus-desktop: /jetson-inference/python/training/detection/ssd
root@usus-desktop: /jetson-inference/python/training/detection/ssd# python3 train
_ssd.py --dataset-type=voc --data=data/pltotal --model-dir=models/pltotal --batc
h-size=8 --workers=2 --epochs=130
2024-11-30 00:13:34 - Using CUDA...
2024-11-30 00:13:34 - Namespace(balance_data=False, base_net=None, base_net_lr=0
.001, batch_size=8, checkpoint_folder='models/pltotal', dataset_type='voc', data
sets=['data/pltotal'], debug_steps=10, extra_layers_lr=None, freeze_base_net=Fal
se, freeze_net=False, gamma=0.1, lr=0.01, mb2_width_mult=1.0, milestones='80,100
', momentum=0.9, net='mb1-ssd', num_epochs=130, num_workers=2, pretrained_ssd='m
odels/mobilenet-v1-ssd-mp-0_675.pth', resume=None, scheduler='cosine', t_max=100
, use_cuda=True, validation_epochs=1, weight_decay=0.0005)
2024-11-30 00:13:34 - Prepare training datasets.
```

Durante el entrenamiento de un modelo SSD en Jetson Inference, se genera un archivo para cada época entrenada como se puede ver en la Figura 64, y estos archivos se almacenan en el directorio que se haya especificado con la opción `--model-dir`, en el caso los modelos entrenados se guardarán en:

`jetson-inference/python/training/detection/ssd/models/pltotal`

FIGURA 64

Archivos de salida al finalizar el entrenamiento de la red neuronal



Convertir el modelo a formato ONNX esto es un paso importante para poder desplegarlo en otras plataformas o realizar inferencias optimizadas, para lo cual se ha utilizado el comando:

```
python3 onnx_export.py --model-dir=models/pltotal
```

Resultado del Comando de Conversión

El resultado indica que la época 105 del entrenamiento tuvo las menores pérdidas, con un valor de 1.2275, esto significa que el modelo entrenado en la época 105 fue el que logró el mejor rendimiento en cuanto a la minimización de la función de pérdida, lo que generalmente se traduce en un mejor desempeño en tareas de detección de objetos.

Al ejecutar el comando, el modelo de la época con la menor pérdida (en este caso, la época 105) fue convertido correctamente a formato ONNX, el comando que ejecutado probablemente generó un archivo en el directorio `models/pltotal` similar a:

```
models/pltotal/
├── model_epoch_105.pth
```

|— model_epoch_105.onnx

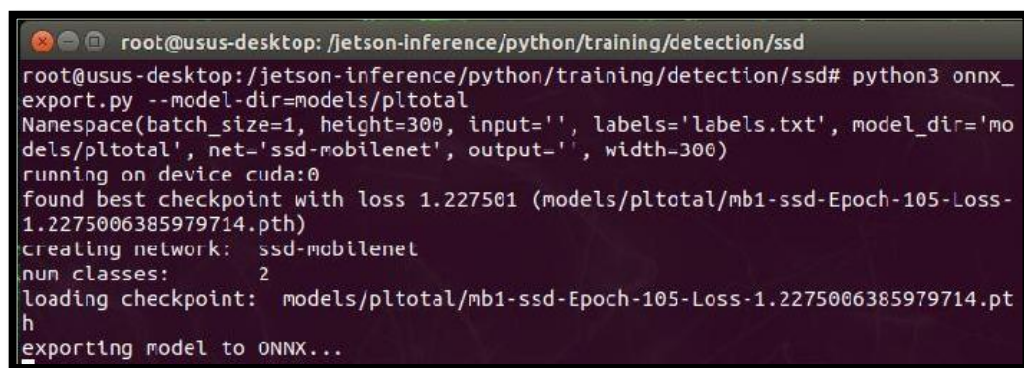
Para comprobar de que la conversión fue exitosa, verificar si el archivo ONNX fue generado correctamente en el directorio especificado como se puede observar en la Figura 65, navegar hasta el directorio:

```
ls models/pltotal
```

Se podrá ver el archivo model_epoch_105.onnx junto al archivo .pth original.

FIGURA 65

Ejecución de la conversión a un modelo onnx



```
root@usus-desktop: /jetson-inference/python/training/detection/ssd
root@usus-desktop: /jetson-inference/python/training/detection/ssd# python3 onnx_
export.py --model-dir=models/pltotal
Namespace(batch_size=1, height=300, input='', labels='labels.txt', model_dir='mo
dels/pltotal', net='ssd-mobilenet', output='', width=300)
running on device cuda:0
found best checkpoint with loss 1.227501 (models/pltotal/mb1-ssd-Epoch-105-Loss-
1.2275006385979714.pth)
creating network: ssd-mobilenet
num classes: 2
loading checkpoint: models/pltotal/mb1-ssd-Epoch-105-Loss-1.2275006385979714.pt
h
exporting model to ONNX...
```

Para probar el Modelo Entrenado sobre un video verificar de tener las dependencias necesarias instaladas, como OpenCV para leer el video y mostrar los resultados, si no se ha guardado el script se puede crear un script en Python para cargar el modelo entrenado y hacer inferencias cuadro por cuadro en el video, este código en un archivo Python, infer_video.py, debe guardar y luego ejecutar desde la terminal de Jetson Nano.

El script procesa el video, aplica el modelo entrenado y mostrará las detecciones en tiempo real o guardará el video procesado en un archivo de salida (output_video.mp4). Si se tiene un modelo ONNX y se quiere aprovechar la aceleración de TensorRT, convertir el modelo ONNX a TensorRT para mejorar el desempeño en la inferencia, esto es particularmente útil cuando se procesa videos en tiempo real como se muestra en la Figura 66.

FIGURA 66

Aplicación de la red neuronal sobre un video en marcha



2.7.21. Base de datos obtenidos del prototipo

El diseño e implementación de las bases de datos se llevó a cabo utilizando SQLite3, lo cual genera un archivo gestionado por su propia biblioteca, esta aplicación incluye mecanismos de bloqueo para evitar que la información almacenada en la base de datos se dañe, pierda. SQLite3 se integra fácilmente con Python a través del módulo estándar sqlite3, aquí se detalla los pasos de manera clara para crear un nuevo archivo de base de datos en DB Browser for SQLite:

Abrir el programa DB Browser for SQLite

- Asegurarse de que DB Browser for SQLite esté instalado.
- Si no está descargado se lo puede descargar ingresando al link: sqlitebrowser.org.

Crear un nuevo archivo de base de datos

- Hacer clic en la opción "Nuevo archivo de base de datos"
- Asignar un nombre significativo al archivo (por ejemplo, base_de_datos_prototitpo_cbvision.db).
- Clic en Guardar.

Crear una tabla en la base de datos

- Una vez creado el archivo, aparecerá la ventana principal del programa.
- En la parte superior, seleccionar la pestaña "Diseñar la base de datos".
- Hacer clic en el botón "Crear tabla".
- Escribir un nombre para la tabla.
- Agregar columnas para la tabla.

- Especificar el nombre de cada columna.
- Seleccionar el tipo de datos.
- Definir si la columna es clave primaria, no nula, o si debe tener un valor predeterminado.
- Cuando se termine, hacer clic en Aceptar.

Guardar la base de datos

- Hacer clic en Archivo; Guardar base de datos o presionar (Ctrl + S).

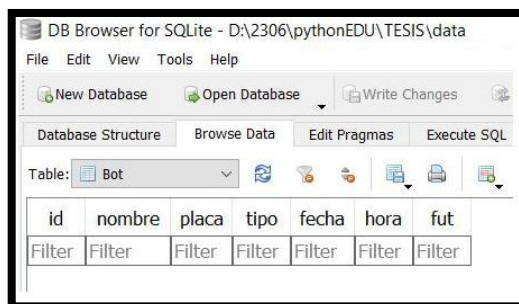
Insertar o visualizar datos (opcional)

Para agregar datos:

- Ir a la pestaña "Examinar datos".
- Seleccionar la tabla que se acaba de crear como se muestra en la Figura 67.
- Hacer clic en "Nuevo registro" e ingresar los datos en las columnas.

FIGURA 67

Nuevo archivo en el programa DB Browser para SQLite y campos requeridos en la base de datos temporal



2.7.22. Subprocesos de ejecución de la programación para el prototipo

La ejecución de todos los procesos en la programación del prototipo requiere un alto coste computacional, se optó por dividir el proceso general en tres subprocesos, esta estrategia permitió distribuir la carga de trabajo de manera más eficiente, asegurando un rendimiento óptimo y facilitando la gestión de las distintas tareas.

Subproceso 1: Abarca la captura del video, el análisis a través de una red neuronal, la identificación y recorte de la región de interés dentro del video procesado aplicando el uso de las siguientes librerías:

- future, os, sys, y time: Manejo de funciones del sistema, control del tiempo y compatibilidad con versiones futuras de Python.
- threading: Gestión de procesos concurrentes.
- jetson.inference y jetson.utils: Funciones especializadas para el procesamiento de datos.
- numpy: Manipulación eficiente de arreglos y datos numéricos.
- cv2: Procesamiento de imágenes y videos.

Subproceso 2: Gestiona una base de datos temporal para almacenar información procesada de manera transitoria, mediante la aplicación de las siguientes librerías:

- sqlite3: Gestión de bases de datos ligeras y embebidas.
- datetime: Manejo de fechas y horas.

Subproceso 3: Conlleva varias funcionalidades primordiales del sistema, Interfaz Gráfica, OCR, Base de Datos Permanente y Control GPIO, para estas tareas, se utilizaron las siguientes librerías:

- tkinter.
- pandas.
- datetime.
- SQLite3.
- Jetson.GPIO.
- pytesseract.
- PIL.
- cv2

Cada subproceso opera de manera sincrónica, aunque de manera autónoma, lo que significa que cada uno funciona en su propia terminal mientras se sincronizan para completar el ciclo de funcionamiento completo del prototipo. Antes de utilizar este comando, es necesario importar al script:

```
import subprocess
```


subprocess.run

2.7.23. Diseño de interfaz de usuario

Como se puede observar en la Figura 68 la interfaz de usuario establece un equilibrio entre diferentes elementos, como la legibilidad del texto, la selección de tipografías adecuadas, el uso de simbolismos claros, una paleta de colores armoniosa y texturas intuitivas, durante el proceso de diseño, se buscó un equilibrio integral entre diversos aspectos fundamentales:

- Legibilidad: Claridad del texto.
- Tipografía: Funcionalidad y estética.
- Simbolismo: Gráficos con significados entendibles.
- Color y Contraste.

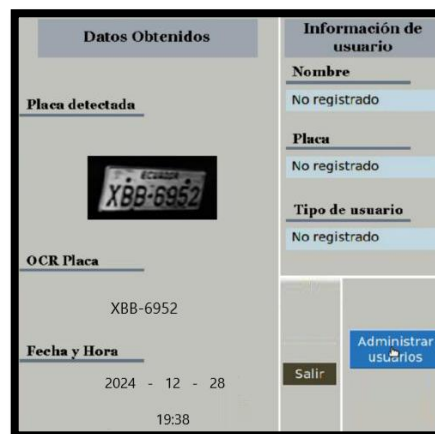
Además, se implementaron indicadores visuales para guiar a los usuarios a través de la interfaz de manera clara y eficiente.

- Botones: Diseñados llamativos.
- Indicadores de estado: Uso de colores y símbolos específicos para representar acciones.

Este enfoque equilibrado garantiza una interfaz visualmente coherente, funcional y accesible para una amplia variedad de usuarios, alineándose con los estándares ergonómicos y de diseño moderno como podemos observar en la Figura 68.

FIGURA 68

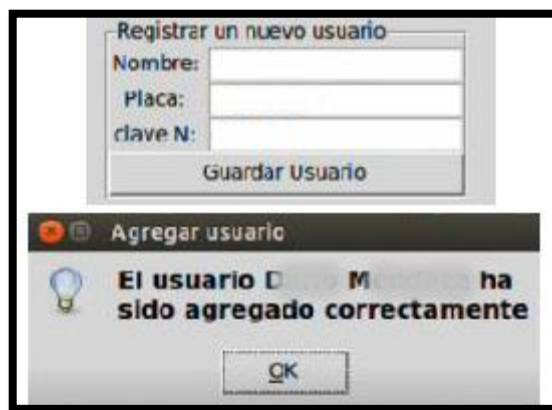
Indicadores de la interface al ingreso del aplicativo



En el proceso de registro de usuario implementa una notificación que proporciona retroalimentación inmediata al usuario al ingresar un nuevo registro, este mensaje, como se observa en la Figura 69, el mensaje que aparece tiene como objetivo principal informar al usuario sobre el estado del registro, ya sea para confirmar que la operación se realizó con éxito, alertar sobre un error o proporcionar información adicional, por ejemplo “*El usuario XXX XXX ha sido agregado correctamente*”.

FIGURA 69

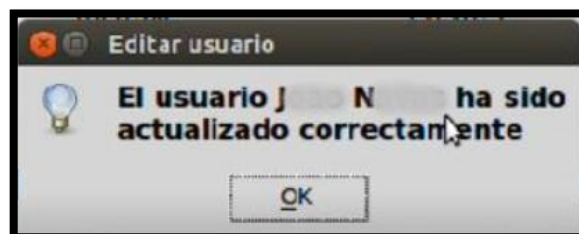
Mensaje al agregar un nuevo usuario a la base de datos.



La edición de usuarios, emite un mensaje de alerta al usuario sobre el estado de la operación, este mensaje como se muestra en la Figura 70, en caso de una edición exitosa, el sistema despliega textos claros como "*El usuario XXX XXX ha sido actualizado correctamente*", si ocurre un problema, el mensaje indica claramente la naturaleza del error, como "*No se pudo guardar los cambios. Verifique los datos ingresados*".

FIGURA 70

Mensaje al ingresar un usuario nuevo.



Para registro de un nuevo usuario, se despliega un mensaje de notificación, como se muestra en la Figura 71, con el objetivo de informar al usuario sobre la situación y guiarlo

hacia la corrección, en caso de que algún campo este vacío se desplegara un mensaje “*Todos los campos son requeridos*”.

FIGURA 71

Mensaje cuando hace falta ingresar datos de los nuevos usuarios



Si no se ha realizado la selección correspondiente, se despliega un mensaje de notificación, como se muestra en la Figura 72, para alertar al usuario sobre la situación y evitar acciones erróneas.

FIGURA 72

Mensaje de error al no marcar o seleccionar un usuario para eliminar o editar



En la primera ventana de la interfaz de usuario, destinada al inicio de sesión, se implementa un sistema de notificación que se activa cuando un usuario introduce una contraseña o usuario incorrectos como se observa en la Figura 73, este mensaje tiene como objetivo informar de manera inmediata sobre el error, guiar al usuario para corregirlo y garantizar la seguridad del sistema al gestionar adecuadamente los intentos fallidos.

FIGURA 73

Mensaje de error al ingresar credenciales de acceso incorrectos



De igual manera en la primera ventana de la interfaz de usuario, diseñada para el inicio de sesión, el sistema detecta si el usuario ha dejado en blanco los campos de *Usuario* y/o *Contraseña*, cuando ocurre esta situación se despliega un mensaje de notificación, como se muestra en la Figura 74.

FIGURA 74

Mensaje de error cuando el usuario o contraseña no son ingresados



CAPÍTULO 3. RESULTADOS Y DISCUSIÓN

3.1. Pruebas de funcionamiento y validación del sistema de vigilancia.

A continuación, se detallan las pruebas realizadas con el objetivo de evaluar la fiabilidad del prototipo al analizar en tiempo real los automóviles que intentan ingresar a la zona controlada de garaje, permitiendo comprobar la eficiencia y robustez del sistema bajo diversas condiciones operativas.

- Se probó el sistema durante el día y la noche y bajo diferentes condiciones de iluminación.
- Se realizaron pruebas hasta con 4 vehículos ingresando en secuencia al parqueadero.

Para el desarrollo de las pruebas de funcionamiento, se consideraron las placas vehiculares y personas a cargo de los vehículos que se presentan en la Tabla 1, estas placas fueron seleccionadas acorde a los vehículos existentes, abarcando una amplia variedad de escenarios que reflejan diferentes casos de uso, es fundamental tener en cuenta el tipo de usuario asociado a cada placa, ya que esta diferenciación permite evaluar la eficacia del sistema al interactuar con las bases de datos, tanto permanente como temporal.

TABLA 1

Registro de usuarios para las pruebas del prototipo

NOMBRE/CARGO	PLACA	DEFINICIÓN
JOSÉ JÁCOME	PCH-3934	JEFE DE AGENCIA
BYRON VELASTEGUI	TBH-5336	JEFE TÉCNICO
MILTON TENORIO	IBL-0226	CALL CENTER
CRISTIAN FAJARDO	TBJ-9236	JEFE ZONAL
EDWIN LAGUAQUIZA	PCO-183	PERSONAL TÉCNICO
VÍCTOR GUERRA	TBA-7104	PERSONAL TÉCNICO
EDUARDO GUANOLUISA	UBP-606	PERSONAL TÉCNICO
PATRICIO PONCE	PDH-7921	PERSONAL RECAUDADOR
JORGE LUMBI	XBA-8867	PERSONAL TÉCNICO


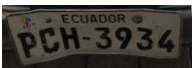

PAUL LAGUAQUIZA	IBM-970	PERSONAL TÉCNICO
HUGO CRUZ	HBD-5197	PERSONAL TÉCNICO
WALTER GUANO	PCP-2830	PERSONAL TÉCNICO
GRUPO 1	ABM-2494	EDWIN/JORGE
GRUPO 2	ABI-9026	HUGO/WALTER
GRUPO 3	AAN-652	EDUARDO/PAUL
RECAUDACIÓN	PCA-7506	PATRICIO/VÍCTOR


En la empresa CBVISION, se han identificado dos periodos de mayor afluencia vehicular, los cuales están directamente relacionados con las actividades diarias del personal administrativo y técnico, durante la mañana el horario de mayor movimiento vehicular ocurre entre las 8:00 y las 8:30 horas, en este intervalo se registra el ingreso del personal administrativo, así como la salida de los técnicos hacia sus respectivas asignaciones en campo, este flujo temprano refleja el inicio de las operaciones diarias de la empresa, por otro lado, en la tarde el horario pico se extiende desde las 17:30 hasta las 19:00 horas o mucho más, durante este periodo los técnicos regresan de sus labores en campo

A continuación, en la Tabla 2 se presentan los resultados recopilados de análisis del primer día a partir de las pruebas realizadas con nuestro sistema, para garantizar la confiabilidad y precisión de los datos obtenidos, se llevaron a cabo mediciones en diferentes días y con los diferentes vehículos registrados, este enfoque permitió analizar el comportamiento del sistema en una amplia gama de escenarios, asegurando que la información recopilada sea representativa y consistente.

TABLA 2

Resultados obtenidos en la base de datos del día 12/12/2024 por la mañana

Fecha	Clima	Hora	Placa	OCR	PORCENTAJE	
					LETRAS	NÚMEROS
12/12/2024	8.7°	7:50		TBH-5336	100%	100%
12/12/2024	8.7°	7:53		PCH-3934	100%	100%
12/12/2024	8.7°	7:54		TBJ-9236	100%	100%

12/12/2024	8.7°	7:55		IBL-0226	100%	100%
12/12/2024	8.7°	7:55		UBP-606	100%	100%
12/12/2024	8.9°	7:56		PCU-183	66.7%	100%
12/12/2024	8.9°	7:56		TBA-7104	100%	100%
12/12/2024	8.9°	7:57		PCP-2836	100%	66.7%
12/12/2024	8.9°	7:57		IBM-970	100%	100%
12/12/2024	8.9°	7:58		XBA-8867	100%	100%
12/12/2024	8.9°	7:58		HBD-5197	100%	100%
12/12/2024	8.9°	8:00		PDH-7921	100%	100%

En el primer día de muestreo y funcionamiento del prototipo, se identificó que dos de las placas presentadas no fueron reconocidas al 100%, este inconveniente se atribuye al deterioro visible de dichas placas, el cual puede ser causado por diversos factores, como desgaste por el tiempo, exposición prolongada a condiciones climáticas adversas, acumulación de suciedad, o daños físicos.

TABLA 3


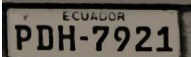
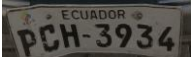

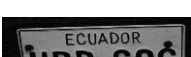

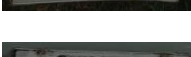



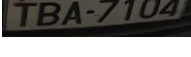
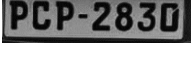
Resultados ingreso de vehículos empresariales en el horario de la tarde el día 12/12/2024

Fecha	Clima	Hora	Placa	OCR	PORCENTAJE	
					LETRAS	NÚMEROS
12/12/2024	16.0°	17:59		AAN-652	100%	100%
12/12/2024	15.0°	18:50		ABI-9026	100%	100%
12/12/2024	13.8°	19:53		ABM-2494	100%	100%
12/12/2024	12.5°	21:47		PCA-7506	100%	100%

Durante el mismo día de análisis, al momento del retorno de los vehículos de trabajo a su parqueadero al finalizar la jornada laboral, el sistema realizó un nuevo muestreo de datos, al igual que en el muestreo llevado a cabo en la mañana, los resultados obtenidos demostraron un 100% de precisión en la identificación de las placas vehiculares como se muestra en la Tabla 3.

TABLA 4

Datos obtenidos en la mañana del día 13/12/2025 con vehículos personales

Fecha	Clima	Hora	Placa	OCR	PORCENTAJE	
					LETRAS	NÚMEROS
13/12/2024	5.4°	7:50		HBD-5197	100%	100%
13/12/2024	5.4°	7:50		PDH-7921	100%	100%
13/12/2024	5.5°	7:51		PCH-3934	100%	100%
13/12/2024	5.5°	7:51		TBJ-9236	100%	100%
13/12/2024	5.7°	7:52		UBP-606	100%	100%
13/12/2024	5.7°	7:52		IBL-0226	100%	100%
13/12/2024	5.8°	7:54		PCU-183	66.7%	100%
13/12/2024	5.8°	7:54		TBH-5336	100%	100%
13/12/2024	5.9°	7:56		IBM-970	100%	100%
13/12/2024	5.9°	7:56		TBA-7104	100%	100%
13/12/2024	5.9°	7:57		PCP-2836	100%	66.7%
13/12/2024	5.9°	7:58		XBA-8867	100%	100%








Como se puede observar en la Tabla 4, el reconocimiento de las mismas dos placas que no fueron identificadas al 100% en días anteriores persiste como un problema recurrente, este comportamiento indica la necesidad de implementar ajustes específicos en el sistema para resolver esta limitación y garantizar una fiabilidad total en el reconocimiento de las placas.

En primera instancia, se recomienda realizar ajustes en la ubicación y el enfoque de la cámara, además, como parte de las acciones correctivas, se ha decidido realizar un entrenamiento específico de la red neuronal, tomando en cuenta únicamente las dos placas problemáticas, este enfoque permitirá que el modelo se ajuste a las características particulares de estas placas, como posibles imperfecciones, desgastes o patrones únicos que puedan estar dificultando su correcta identificación.

TABLA 5

Enfoque y entrenamiento de la red neuronal con las placas vehiculares que presentan inconvenientes (primer vehículo)



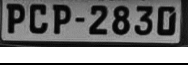
N°	Fecha	Clima	Hora	Placa	OCR	PORCENTAJE	
						LETRAS	NÚMEROS
1	14/12/2024	3.4°	7:56		PCU-183	66.7%	100%
2	14/12/2024	18.4°	13:37		PCU-183	66.7%	100%
3	14/12/2024	15.5°	17:52		PCU-183	66.7%	100%
4	15/12/2024	7.5°	7:51		PCU-183	66.7%	100%
5	15/12/2024	19.7°	13:40		PCU-183	66.7%	100%
6	15/12/2024	13.3°	18:00		PCU-183	66.7%	100%
7	16/12/2024	8.3°	7:48		PCU-183	66.7%	100%
8	16/12/2024	20.8°	14:02		PCU-183	66.7%	100%








9	16/12/2024	12.4°	19:00		PCO-183	100%	100%
10	17/12/2024	5.1°	7:42		PCO-183	100%	100%
11	17/12/2024	13.0°	17:57		PCO-183	100%	100%
12	18/12/2024	20.9°	13:58		PCO-183	100%	100%
13	18/12/2024	12.4°	18:13		PCO-183	100%	100%
14	20/12/2024	21.0°	13:56		PCO-183	100%	100%
15	23/12/2024	2.5°	7:33		PCO-183	100%	100%

Como se puede observar en la Tabla 5, después de realizar varias pruebas al primer vehículo con el objetivo de mejorar la fiabilidad y eficiencia de nuestro sistema de seguridad, se lograron resultados positivos posterior a 8 pruebas erróneas realizadas previo a los ajustes correctos, las mismas que reflejan un avance significativo en el rendimiento del sistema, estas pruebas fueron diseñadas para identificar y resolver cualquier inconveniente relacionado con el reconocimiento óptico de caracteres (OCR) en las placas vehiculares.

TABLA 6

Enfoque y entrenamiento de la red neuronal con las placas vehiculares que presentan inconvenientes (segundo vehículo)

N°	Fecha	Clima	Hora	Placa	OCR	PORCENTAJE	
						LETRAS	NÚMEROS
1	14/12/2024	3.4°	7:58		PCP-2830	100%	100%
2	14/12/2024	18.4°	13:40		PCP-2830	100%	100%
3	14/12/2024	15.5°	17:56		PCP-2830	100%	100%



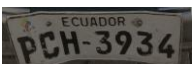
4	15/12/2024	7.5°	7:53		PCP-2830	100%	100%
5	15/12/2024	19.7°	13:58		PCP-2830	100%	100%
6	15/12/2024	13.3°	18:10		PCP-2830	100%	100%
7	16/12/2024	8.3°	7:50		PCP-2830	100%	100%
8	16/12/2024	20.8°	14:03		PCP-2830	100%	100%
9	16/12/2024	12.4°	19:01		PCP-2830	100%	100%
10	17/12/2024	5.1°	7:44		PCP-2830	100%	100%

En el caso del segundo vehículo en análisis, los resultados presentados en la Tabla 6 evidencian que las modificaciones y ajustes implementados en el sistema dieron resultados favorables desde las primeras pruebas realizadas, estas pruebas abarcaron diversos días y escenarios, asegurando un análisis integral del rendimiento del sistema frente a diferentes condiciones operativas.

Para validar el funcionamiento del prototipo se valida en los siguientes días de trabajo con resultados positivos en todos los casos, como podemos observar en la tabla, aquí podemos evidenciar que con los cambios realizados este sistema presenta un 100% de efectividad.

TABLA 7

Análisis general de los datos obtenidos el día 08/01/2025 luego del entrenamiento y modificaciones


Fecha	Clima	Hora	Placa	OCR	PORCENTAJE	
					LETRAS	NÚMEROS
08/01/2025	6.7°	7:51		TBA-7104	100%	100%
08/01/2025	6.7°	7:51		TBJ-9236	100%	100%
08/01/2025	6.7°	7:51		PCH-3934	100%	100%

08/01/2025	6.8°	7:53		IBM-970	100%	100%
08/01/2025	6.8°	7:53		PCP-2830	100%	100%
08/01/2025	6.8°	7:53		UBP-606	100%	100%
08/01/2025	6.9°	7:57		PDH-7921	100%	100%
08/01/2025	6.9°	7:57		TBH-5336	100%	100%
08/01/2025	6.9°	7:58		XBA-8867	100%	100%
08/01/2025	6.9°	7:58		IBL-0226	100%	100%
08/01/2025	7.0°	8:00		HBD-5197	100%	100%
08/01/2025	7.0°	8:03		PCO-183	100%	100%
08/01/2025	12.6°	10:13		ABK-4901	100%	100%
08/01/2025	14.2°	12:18		TBI-6504	100%	100%
08/01/2025	21.0°	14:27		PWK-784	100%	100%

Como se puede observar en la Tabla 7, el día 8 de enero se obtuvo un 100% de precisión en los datos recopilados por el sistema, incluso ante el ingreso de tres vehículos que no estaban registrados previamente en nuestra base de datos, este resultado evidencia la capacidad del sistema para procesar y reconocer de manera eficiente placas vehiculares desconocidas, diferenciándolas correctamente de las registradas en el sistema.

TABLA 8

Registro ingreso de vehículos empresariales posterior a modificaciones

Fecha	Clima	Hora	Placa	OCR	PORCENTAJE	
					LETRAS	NÚMEROS
08/01/2025	16.0°	17:59		PCA-7506	100%	100%


08/01/2025	15.8°	18:30		ABI-9026	100%	100%
08/01/2025	15.8°	19:43		ABM-2494	100%	100%
08/01/2025	10.5°	23:57		AAN-652	100%	100%

De igual manera, se puede observar en la Tabla 8 que el sistema opera de manera óptima durante el ingreso de los vehículos empresariales a la zona de parqueo, logrando registrar los datos con un 100% de precisión en el reconocimiento de caracteres.

En los días posteriores de pruebas, el sistema continuó mostrando un 100% de efectividad en el reconocimiento de caracteres de todos los vehículos que ingresaron a los patios, este resultado consistente valida la robustez y estabilidad del sistema frente a diferentes escenarios operativos, hasta la fecha de las pruebas, no se ha evidenciado el ingreso de vehículos no registrados en la base de datos, con la excepción del caso puntual ocurrido el día 8 de enero, cuando ingresaron tres vehículos no registrados que fueron correctamente identificados por el sistema.

TABLA 9

Datos obtenidos el último día de pruebas del prototipo 12/01/2025

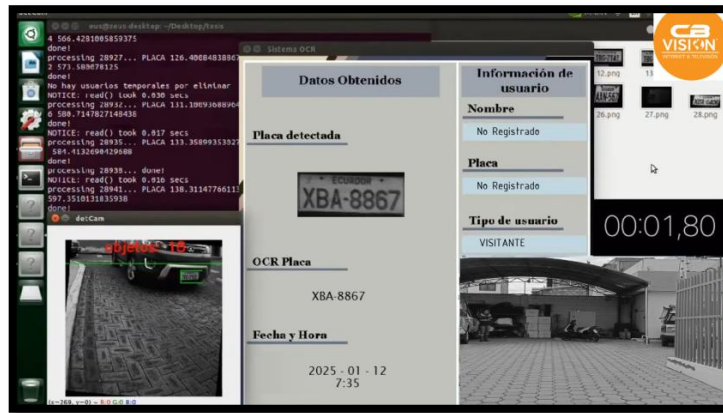
Fecha	Clima	Hora	Placa	OCR	PORCENTAJE	
					LETRAS	NÚMEROS
12/01/2025	7.6°	7:31		TBA-7104	100%	100%
12/01/2025	7.6°	7:31		TBJ-9236	100%	100%
12/01/2025	7.9°	7:32		PCH-3934	100%	100%
12/01/2025	7.9°	7:32		IBM-970	100%	100%
12/01/2025	7.9°	7:33		PCP-2830	100%	100%
12/01/2025	7.9°	7:33		UBP-606	100%	100%

12/01/2025	7.9°	7:33		PDH-7921	100%	100%
12/01/2025	8.1°	7:35		TBH-5336	100%	100%
12/01/2025	8.1°	7:35		XBA-8867	100%	100%
12/01/2025	8.2°	7:38		IBL-0226	100%	100%
12/01/2025	8.2°	7:38		HBD-5197	100%	100%
12/01/2025	8.2°	7:38		PCO-183	100%	100%
12/01/2025	16.3°	17:59		PCA-7506	100%	100%
12/01/2025	16.3°	18:00		ABI-9026	100%	100%
12/01/2025	16.2°	18:00		ABM-2494	100%	100%
12/01/2025	16.2°	18:01		AAN-652	100%	100%

Como se puede observar en la Tabla 9, los resultados obtenidos hasta el último día de análisis reflejan un 100% de efectividad en el reconocimiento de caracteres y el registro de datos por parte del sistema, es importante aclarar que la información relacionada con el clima, presentada en los resultados de análisis, fue ingresada manualmente, estos datos corresponden al clima actual en la ciudad a las horas específicas en que se realizaron las pruebas y se recopilaron de fuentes externas confiables, la decisión de registrar esta información manualmente se debe a las dificultades técnicas encontradas para automatizar la obtención de este parámetro mediante programación directa dentro del sistema.

FIGURA 75

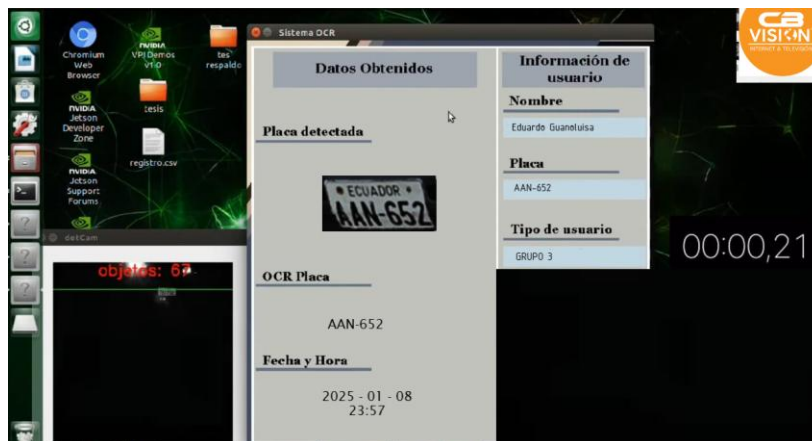
Muestra de funcionamiento del prototipo desde el interior del vehículo durante el día



Como se puede observar en la Figura 75, se muestra el funcionamiento del prototipo en diversos casos de prueba, en este caso, se presenta la prueba realizada el 12 de enero de 2025 a las 7:35 a.m., en la cual se grabó desde el interior del vehículo mientras se llevaba a cabo el proceso de funcionamiento normal del prototipo y corroborar que el sistema tiene una respuesta confiable. Además, se midió el tiempo de respuesta del sistema para procesar la información capturada, con los resultados obtenidos, se verifica que el tiempo de respuesta se mantuvo por debajo de los 2 segundos, lo que demuestra un rendimiento eficiente y rápido del sistema en condiciones de operación real. En la parte superior derecha de la figura, se puede observar el registro de usuarios, donde se asigna al vehículo como visitante debido a que no está registrado en la base de datos.

FIGURA 76

Muestra de funcionamiento del prototipo desde el interior del vehículo durante la noche



En la Figura 76 se muestra el proceso de funcionamiento del prototipo durante la prueba realizada el 8 de enero de 2025, mientras un vehículo de la agencia retornaba al finalizar la jornada laboral, en este caso, se puede observar que el vehículo está registrado en la base de datos y ha sido asignado al GRUPO 3, a cargo de Eduardo Guanoluisa. El sistema registra la hora de ingreso a las 23:57, a pesar de las condiciones de baja iluminación, la cámara logra captar con precisión la zona de interés y obtener los resultados esperados, lo que evidencia la efectividad del sistema en entornos con poca luz, el sistema automático de iluminación no se encendió ese día debido a un fallo en el sistema eléctrico de la sucursal y la estación de energía se destinó en su totalidad a cubrir la demanda del sistema de vigilancia.

En la Tabla 10 se presenta un ejemplo ilustrativo de cómo se realiza el registro en la base de datos del ingreso de vehículos a los patios de la empresa, cabe recordar que en apartados anteriores se abordaron aspectos relacionados con la confidencialidad y la protección de la información, debido a estas consideraciones, se ha optado por omitir las tablas específicas que contienen los datos completos de los registros.

TABLA 10

Ejemplo demostrativo de la base de datos

Fecha	Descripción	Hora	Placa	OCR	RESPONSABLE JEFE DE GRUPO
12/12/2024	GRUPO 3	17:59		AAN-652	Eduardo Guanoluisa
14/12/2024	GRUPO 2	18:50		ABI-9026	Hugo Cruz
18/12/2024	GRUPO 1	19:53		ABM-2494	Edwin Laguaquiza
05/01/2025	RECAUDACIÓN	21:47		PCA-7506	Patricio Ponce
08/01/2025	NO REGISTRADO	12:18		TBI-6504	VISITANTE

3.1.1. Pruebas de fiabilidad y confiabilidad del sistema con vehículos no registrados

El día 18 de enero de 2025, se realizó el ingreso de diez vehículos ajenos a la empresa en las instalaciones del garaje, para asegurar la fiabilidad de los resultados, cada uno de estos vehículos efectuó tres ingresos en distintos horarios a lo largo del día, de esta manera, se buscó evaluar la precisión del prototipo en la detección y gestión de datos en diferentes escenarios, en la Tabla 11, se presentan los resultados obtenidos, los cuales permiten corroborar el correcto funcionamiento del sistema bajo estas condiciones controladas de prueba.

TABLA 11

Resultados de las pruebas con vehículos no registrados

Fecha	Clima	Hora	Placa	OCR	PORCENTAJE	
					LETRAS	NÚMEROS
18/12/2024	6.1°	6:38		PTC-150	66.7%	100%
18/12/2024	7.0°	7:10		PPB-1460	100%	100%
18/12/2024	7.6°	7:42		CBM-871	100%	100%
18/12/2024	9.6°	7:53		IAI-0152	100%	100%
18/12/2024	9.7°	7:59		PDU-093	100%	100%
18/12/2024	9.9°	8:07		TBA-4782	100%	100%
18/12/2024	10.1°	8:40		PXE-482	100%	100%
18/12/2024	10.2°	8:54		XBW-184	100%	100%
18/12/2024	10.3°	9:10		PGC-101	100%	100%
18/12/2024	10.3°	9:11		PFH-6692	100%	100%
18/12/2024	10.8°	10:12		PDU-093	100%	100%
18/12/2024	10.8°	10:15		TBA-4782	100%	100%

18/12/2024	10.8°	10:18		IAI-0152	100%	100%
18/12/2024	10.9°	10:20		PXE-482	100%	100%
18/12/2024	10.9°	10:30		CBM-871	100%	100%
18/12/2024	10.9°	10:35		XBW-184	100%	100%
18/12/2024	10.9°	10:40		PPB-1460	100%	100%
18/12/2024	10.9°	10:42		PGC-101	100%	100%
18/12/2024	10.9°	10:48		PTC-150	66.7%	100%
18/12/2024	10.9°	10:50		PFH-6692	100%	100%
18/12/2024	16.2°	14:12		XBW-184	100%	100%
18/12/2024	15.3°	15:20		CBM-871	100%	100%
18/12/2024	13.8°	16:37		PPB-1460	100%	100%
18/12/2024	13.5°	16:50		PXE-482	100%	100%
18/12/2024	11.4°	18:12		PGC-101	100%	100%
18/12/2024	11.2°	18:56		IAI-0152	100%	100%
18/12/2024	10.8°	19:13		PTC-150	66.7%	100%
18/12/2024	10.1°	20:03		TBA-4782	100%	100%
18/12/2024	9.2°	21:06		PFH-6692	100%	100%
18/12/2024	9.1°	21:10		PDU-093	100%	100%

Como se puede observar en la Tabla 11, se presentan los datos obtenidos de la prueba realizada con vehículos no registrados en la base de datos, los resultados indican que, de los diez vehículos analizados, nueve alcanzaron un 100 % de aciertos en el

reconocimiento, mientras que un vehículo presentó un 66.7 % de precisión en el OCR de texto durante las tres pruebas realizadas.

A continuación, en la Tabla 12 se muestra el ejemplo demostrativo de la base de datos obtenida de nuestro prototipo y como se registraron estos ingresos a los patios de la empresa.

TABLA 12

Ejemplo demostrativo de la base de datos con vehículos no registrados


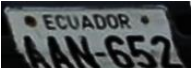

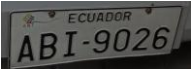



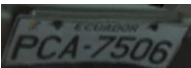

Fecha	Descripción	Hora	Placa	OCR	RESPONSABLE JEFE DE GRUPO
18/01/2025	NO REGISTRADO	6:38		PTC-150	VISITANTE
18/01/2025	NO REGISTRADO	7:10		PPB-1460	VISITANTE
18/01/2025	NO REGISTRADO	7:42		CBM-871	VISITANTE
18/01/2025	NO REGISTRADO	7:53		IAI-0152	VISITANTE
18/01/2025	NO REGISTRADO	7:59		PDU-093	VISITANTE
18/01/2025	NO REGISTRADO	8:07		TBA-4782	VISITANTE
18/01/2025	NO REGISTRADO	8:40		PXE-482	VISITANTE
18/01/2025	NO REGISTRADO	8:54		XBW-184	VISITANTE
18/01/2025	NO REGISTRADO	9:10		PGC-101	VISITANTE
18/01/2025	NO REGISTRADO	9:11		PFH-6692	VISITANTE

Adicionalmente, el día 19 de enero de 2025 se llevó a cabo una nueva prueba con un total de nueve vehículos: cinco no registrados en la base de datos y cuatro pertenecientes a la

empresa, el objetivo de esta prueba fue verificar la integridad y correcto funcionamiento de la base de datos, así como analizar el proceso de registro y corroborar la precisión de los datos obtenidos, las pruebas se realizaron en distintos horarios a lo largo del día para evaluar el desempeño del sistema en diferentes condiciones, los resultados obtenidos se presentan en la Tabla 13.

TABLA 13

Prueba con vehículos registrados y no registrados

Fecha	Descripción	Hora	Placa	OCR	RESPONSABLE JEFE DE GRUPO
19/01/2025	NO REGISTRADO	8:07		TBA-4782	VISITANTE
19/01/2025	GRUPO 3	9:03		AAN-652	Eduardo Guanoluisa
19/01/2025	NO REGISTRADO	10:54		XBW-184	VISITANTE
19/01/2025	GRUPO 2	11:25		ABI-9026	Hugo Cruz
19/01/2025	NO REGISTRADO	12:11		PFH-6692	VISITANTE
19/01/2025	GRUPO 1	12:53		ABM-2494	Edwin Laguaquiza
19/01/2025	NO REGISTRADO	14:10		PGC-101	VISITANTE
19/01/2025	RECAUDACIÓN	14:47		PCA-7506	Patricio Ponce
19/01/2025	NO REGISTRADO	17:00		PDU-093	VISITANTE

Como se puede observar en la Tabla 13, el OCR aplicado a los vehículos registrados, junto con la verificación en la base de datos mostró una efectividad del 100 %, esto confirma el correcto funcionamiento del prototipo y su capacidad para registrar con

precisión los vehículos que ingresan al garaje de la empresa, este nivel de precisión garantiza que el sistema proporcione información confiable a los encargados de la sucursal de la empresa, lo que resulta fundamental para la toma de decisiones en casos fortuitos o situaciones imprevistas presentadas cuando no haya personal de seguridad en su cumplimiento del turno respectivo.

A continuación, se describen un conjunto de ventajas y limitaciones observadas durante la puesta en marcha del prototipo implementado.

3.1.2. Ventajas del prototipo

Cuando se ejecuta el software el prototipo es capaz de realizar múltiples tareas para el reconocimiento y gestión vehicular dentro de la empresa CBVISION, sucursal Salcedo, estas funciones incluyen:

- Reconocimiento automático de placas vehiculares: Mediante la red neuronal implementada.
- Procesamiento y tratamiento de imágenes: Las imágenes capturadas son procesadas de manera automática con OCR.
- Comparación en tiempo real con la base de datos.
- Está diseñado con una interfaz HMI para la interacción con los operadores.
- Eliminación de procesos manuales en el control de acceso vehicular.
- Comparación instantánea de los datos obtenidos con la base de datos permanente y temporal.
- Identificación inmediata de vehículos autorizados y no autorizados.
- Adaptación a distintos horarios y condiciones de luz gracias a la luminaria de compensación con temporizador y sensor de movimiento.
- Posibilidad de expandir la base de datos para incluir más usuarios y vehículos.
- Minimización de fallos en el control de acceso debido a fatiga o descuidos de operadores humanos.
- Compatibilidad con diferentes tipos de cámaras y sensores para mejorar el reconocimiento.
- Posibilidad de implementar alertas visuales o sonoras en caso de acceso no autorizado.

- Identificación de placas de manera rápida y eficiente mediante una red neuronal y OCR.

3.1.3. Limitaciones del prototipo

Para garantizar el óptimo funcionamiento del prototipo, es fundamental considerar los siguientes aspectos:

- Cumplimiento normativo: Placas vehiculares que cumplen con el Reglamento a la Ley de Transporte Terrestre, Tránsito y Seguridad Vial.
- Condicionante de validez: Placas deterioradas, repintadas o con daños visibles.
- Ubicación de la placa: Una placa mal posicionada en el vehículo.
- Iluminación: Iluminación insuficiente o excesiva.
- Condiciones Climáticas Extremas: Lluvia intensa, niebla densa, polvo o suciedad acumulada en las placas.
- Velocidad del Vehículo: Velocidad superior a la esperada.
- Interferencias en la Captura de Imágenes: Obstáculos, vehículos estacionados o cruce de peatones.

CONCLUSIONES

En el desarrollo de esta investigación, se cumplieron los objetivos planteados, logrando diseñar e implementar un sistema de seguridad basado en inteligencia artificial para el reconocimiento de placas vehiculares en la Empresa CBVISION sucursal Salcedo.

- La revisión del estado del arte, permite identificar las tecnologías más utilizadas en sistemas de seguridad basados en inteligencia artificial y entrenamiento de redes neuronales, permitiendo establecer los parámetros de diseño adecuados para el desarrollo del prototipo, considerando las condiciones operativas y los requisitos de precisión, además brindó la información suficiente para la selección de nuestra tarjeta NVIDIA, y aplicación de algoritmos de programación.
- El desarrollo del prototipo conlleva en su principal funcionamiento el uso de sensores y cámaras para la captura de imágenes de las placas vehiculares, permitiendo la recolección y análisis de datos en tiempo real, la implementación de este sistema consideró variables de entrada como la iluminación, el ángulo de captura y la resolución de la imagen, optimizando la detección y procesamiento de información.
- La implementación del sistema de iluminación automática ayuda a tener un mejor enfoque de las zonas de interés bajo diferentes condiciones climatológicas, garantizando un desempeño óptimo del prototipo en condiciones nocturnas o de baja visibilidad, aplicando la técnica de iluminación directa es decir emitir la luz en la misma dirección de enfoque de la cámara, permitiendo de esta manera mejorar la captación de imágenes y brindar mayor fiabilidad al prototipo.
- El entrenamiento de la red neuronal se realizó mediante técnicas de deep learning para el reconocimiento de caracteres en las placas de los vehículos, el entrenamiento de la red neuronal tardó un tiempo de 54 horas en ejecutarse, mostrando un correcto funcionamiento desde sus primeras pruebas, brindando confiabilidad y calidad al prototipo.
- En los días de prueba del prototipo se pudo observar que las placas deterioradas o alteradas representan un punto negativo para el correcto funcionamiento del sistema, ya que, genera lecturas de caracteres incorrectas, en horarios nocturnos

imposibilita la captura de la imagen, el infrarrojo de la cámara no reconoce la zona de interés pese a tener la iluminación externa del sistema automático.

- Para evitar que el sistema de procesamiento se haga demorado, se implementó un sensor que ayuda a detectar la zona de interés (placa) y así se evite de estar teniendo capturas de imágenes incorrectas, una vez que el vehículo sobrepasa la línea referencial se toma la imagen de muestra y comienza su desarrollo.
- En pruebas realizadas de rapidez de funcionamiento del sistema se pudo evidenciar que el sistema en horarios del día es capaz de procesar una imagen en 2 segundos y registrarla en la base de datos. En horarios nocturnos el prototipo se tarda alrededor de 5 segundos en hacer su proceso de toma de imagen y extracción del texto.

RECOMENDACIONES

- El prototipo opera a partir de la captura de video en tiempo real, por lo que es fundamental que el lente de la cámara se mantenga en óptimas condiciones, libre de suciedad, polvo u otras impurezas que puedan afectar la calidad de la imagen y, por ende, la precisión del reconocimiento de placas.
- Para garantizar un desempeño óptimo, las placas vehiculares deben cumplir con el Reglamento a la Ley de Transporte Terrestre, Tránsito y Seguridad Vial, la modificación o alteración de sus características, como repintados, deterioro o cambios en el material, pueden afectar la correcta identificación mediante el sistema OCR.
- Se debe realizar un mantenimiento periódico de la cámara donde se deberá limpiar regularmente el lente con un paño adecuado para evitar acumulación de polvo o manchas y evitar rayones en el lente.
- Verificar el enfoque y alineación de la cámara para garantizar una captura clara de las placas, ya que de esto depende el correcto desempeño del prototipo.
- Instalar la cámara en un ángulo adecuado para capturar de manera óptima las placas, evitando obstáculos o elementos que interfieran, además se debe ajustar la altura y orientación según el tipo de vehículos que acceden a la zona de interés.
- Mantener la información de los vehículos registrada en el sistema actualizada para evitar inconsistencias, además se deberá revisar periódicamente los registros históricos para detectar posibles anomalías o accesos no autorizados.

REFERENCIAS

- Agbeyangi, A., Alashiri, O., & Otunuga, A. (2020). *Identificación automática del número de placas del vehículo usando Raspberry Pi*. Obtenido de <https://ieeexplore.ieee.org/document/9077648/authors#authors>
- Ayerdi, A. (25 de 10 de 2024). *Que es OCR y para que se utiliza*. Obtenido de <https://start.docuware.com/es/blog/tecnologia-ocr-docuware>
- Berenguer, J. (09 de 2023). *PROGRAMACIÓN DE UNA RED NEURONAL Y AJUSTE DE SUS PARÁMETROS*. Obtenido de https://oa.upm.es/75938/1/TFG_JUAN_BERENGUER_TRIANA.pdf
- Cáceres, P. A. (07 de 07 de 2021). *IMPLEMENTACIÓN DE UN SISTEMA INTELIGENTE PARA LA IDENTIFICACIÓN VEHICULAR*. Obtenido de <https://repositorio.uta.edu.ec/server/api/core/bitstreams/51e6e2ac-df51-4d0b-b0cd-350ec45c0333/content>
- Caiafa, F. (06 de 2020). *Que es la inteligencia artificial*. Obtenido de <https://ri.conicet.gov.ar/handle/11336/110093>
- Cañadas, R. (04 de 11 de 2021). *Redes neuronales artificiales*. Obtenido de <https://abdatum.com/tecnologia/redes-neuronales-artificiales>
- Cardenas Hidalgo, P., Flores Vargas, J. A., Lopez Zavaleta, J., & Martinez Moreno, P. (06 de 2009). *DISEÑO DE SISTEMA DE RECONOCIMIENTO DE PLACAS UTILIZANDO MATLAB*. Obtenido de <https://tesis.ipn.mx/bitstream/handle/123456789/7376/ice%20237.pdf?sequence=1&isAllowed=y>
- Casillas, L., Gilbert, M., & Perez, O. (15 de 02 de 2021). *Bases de datos en MySQL*. Obtenido de https://d1wqtxts1xzle7.cloudfront.net/54167148/MYSQL_MANUAL-libre.pdf?1502992016=&response-content-disposition=inline%3B+filename%3DBases_de_datos_en_MySQL.pdf&Expires=1738734771&Signature=UiK2jv4lpyT9Y48su3ajJOCFLfVTeow45u4R92lzE8D3ZgXSinYmVW~kjmM12cyFHv
- Cervilla, D. (04 de 11 de 2017). *Sistemas de iluminación para aplicaciones de visión artificial*. Obtenido de <https://bcnvision.es/blog-vision-artificial/iluminacion-vision-artificial2/>
- Duran Gomez, M. (2009). RECONOCIMIENTO DE CARACTERES ÓPTICOS OCR POR MEDIO DE CORRELACIÓN Y REDES NEURONALES. *UPR*, 10-40.

- Gomez, R. (12 de 10 de 2024). *Inteligencia Artificial*. Obtenido de <https://www.comoves.unam.mx/numeros/articulo/2/la-inteligencia-artificial-hacia-donde-nos-lleva>
- Guerra Cantero, L. M., & Vanegas Lago, E. (2014). Software libre para la restauración y transcripción de imágenes archivísticas. *3ciencias*, 21-33. Obtenido de <https://3ciencias.com/wp-content/uploads/2014/02/DOCLUX-OCR.pdf>
- Hercó. (17 de 11 de 2021). *Programador de horario digital*. Obtenido de <https://suministrosherco.com/blog/programadores-horarios-que-son-y-ventajas?srltid=AfmBOoqnkCVDGxF-ScHwfaFNgjJuuFBZhA8xE-mnSXTF5hYCBml4Da6A>
- HIKVISION. (10 de 2023). *Camara IP tipo domo 2 MP*. Obtenido de <https://hikvisioncolombia.com/producto/camara-tipo-domo-ip-2-mp-ir-30-m-ds2cd1121i28/>
- Laura, A. (03 de 2023). *Vision Artificial y sus componentes*. Obtenido de <https://repositorio.uta.edu.ec/server/api/core/bitstreams/704dfca1-b41e-4933-a2dc-1d333d4c1d91/content>
- León, R., Huaman, J., & Castro, J. (19 de 01 de 2023). *DESARROLLO DE UN SISTEMA DE DETECCIÓN DE PLACAS DE AUTOS*. Obtenido de <https://www.iiis.org/CDs2023/CD2023Spring//papers/CB206TN.pdf>
- Loaiza, L., & Ibañez, E. (19 de 02 de 2024). *DESARROLLO DE UN PROTOTIPO DE SISTEMA DE DETECCIÓN DE ETIQUETAS IMPRESAS EN SACOS DE BALANCEADO MEDIANTE REDES NEURONALES*. Obtenido de <https://dspace.ups.edu.ec/bitstream/123456789/27420/1/UPS-GT004975.pdf>
- MAZ. (12 de 11 de 2023). *Rele de 11 pies*. Obtenido de <https://mazcr.com/es/rees-electromagneticos/1478-rele-11-pines-planos-bobina-12-vac-5a.html>
- NVIDIA. (2021). *Modulo Jetson Nano*. Obtenido de <https://www.nvidia.com/es-es/autonomous-machines/embedded-systems/jetson-nano/product-development/>
- Pech, F., Tass, N., Domínguez, B., Guzmán, J., Coop, R., & Rejón, C. (23 de 11 de 2021). *Identificación y Reconocimiento de Placas de Automóviles por OCR*. Obtenido de https://www.researchgate.net/profile/Fernando-Pech-May-2/publication/373012372_LibroENC10Marzo2023/links/64d3bb9dc80b930269fb6c6b/LibroENC10Marzo2023.pdf#page=406
- Pinto Vizcarra, C. A. (29 de 06 de 2019). *SISTEMA DE IDENTIFICACIÓN DE VEHÍCULOS UTILIZANDO REDES NEURONALES ARTIFICIALES*. Obtenido de <https://repositorio.ucsm.edu.pe/bitstream/handle/20.500.12920/9577/4A.0322.I M.pdf?sequence=1>

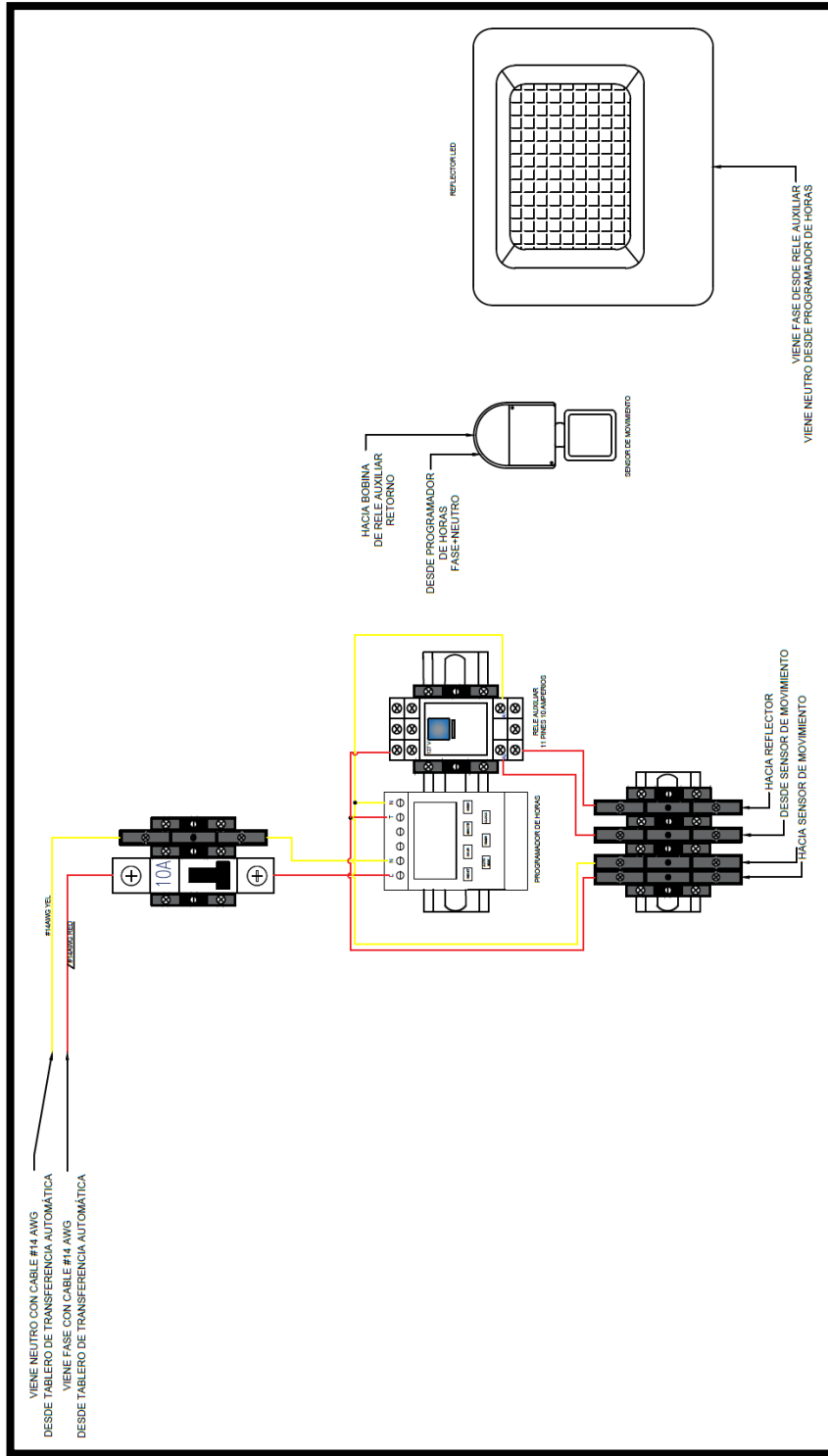
- REPSOL. (06 de 07 de 2024). *Inteligencia Artificial*. Obtenido de <https://www.repsol.com/es/energia-futuro/tecnologia-innovacion/inteligencia-artificial/index.cshtml>
- Rodrigo, J. A. (05 de 2021). *Redes neuronales con Python*. Obtenido de <https://cienciadedatos.net/documentos/py35-redes-neuronales-python>
- Roldán Cordero, D. (2023). APLICACIÓN DE TÉCNICAS DE MACHINE LEARNING PARA LA MEJORA DEL RECONOCIMIENTO ÓPTICO DE CARACTERES (OCR) SOBRE SUPERFICIES INDUSTRIALES. *UCrea*, 10-25.
- Rubio, O. (2020). PROCESAMIENTO DIGITAL DE IMÁGENES. *vistronica*.
- S&P. (05 de 09 de 2024). *Sensor de movimiento*. Obtenido de <https://www.solerpalau.com/es-es/blog/sensores-movimiento/>
- Steren. (05 de 09 de 2024). *Sensor de movimiento*. Obtenido de <https://www.steren.com.ec/sensor-de-movimiento-pir-para-intemperie.html>
- TECH, T. T. (24 de 08 de 2024). *Sistemas de seguridad: Tipos y casos de uso*. Obtenido de <https://pyv.technology/blog/sistemas-de-seguridad-tipos-y-casos-de-uso-guia-completa/>
- Tecnología, G. d. (20 de 10 de 2021). *Programador horario digital*. Obtenido de <https://genteca.com.ve/programador-horario-digital-de-bornera-para-equipos-electricos-en-general-220v.r.aspx>
- Vieites, A. G. (7 de 09 de 2014). *Seguridad en Equipos Informaticos*. Obtenido de https://www.google.com.ec/books/edition/MF0486_3_Seguridad_en_Equipos_Inform%C3%A1ti/o6W6EAAAQBAJ?hl=es&gbpv=1&dq=Los+sistemas+de+seguridad+son+conjuntos+de+tecnolog%C3%ADas,+procedimientos+y+estrategias+dise%C3%B1ados+para+proteger+personas,+bienes+e+inf
- Wong, E. (01 de 06 de 2023). *Que es una direccion IP estatica y dinamica*. Obtenido de <https://www.godaddy.com/resources/es/tecnologia/que-es-direccion-ip-estatica-y-dinamica>
- Zainal, A., Atul, C., Prashengit, D., & Shahadat, H. (21 de 12 de 2017). *Sistema de reconocimiento de placas de licencia basado en propiedades de contorno y modelo de aprendizaje profundo*. Obtenido de <https://ieeexplore.ieee.org/document/8289029/authors#authors>

ANEXOS

ANEXO 1

FIGURA 77

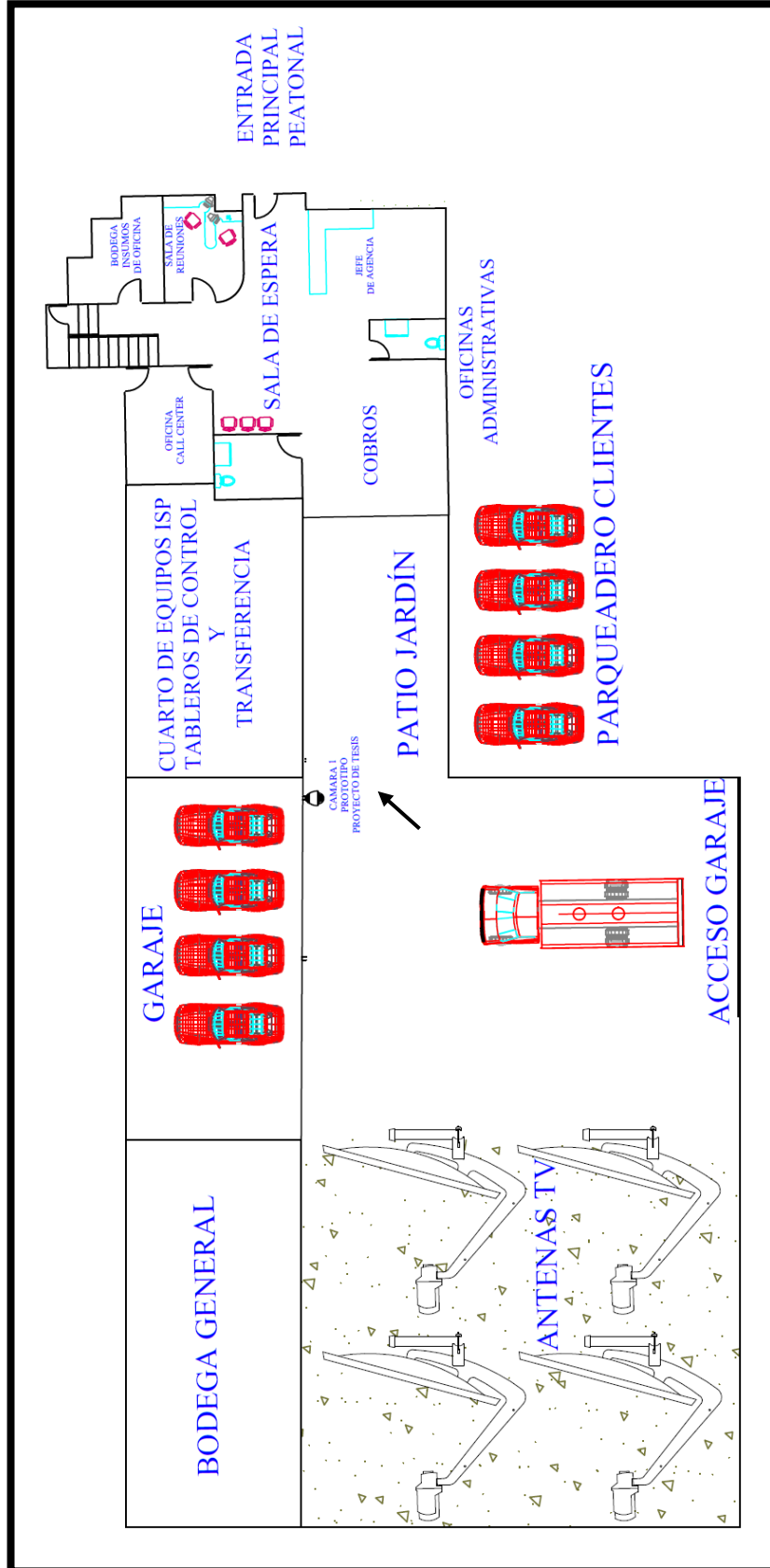
Representación gráfica del sistema de iluminación automático



ANEXO 2

FIGURA 78

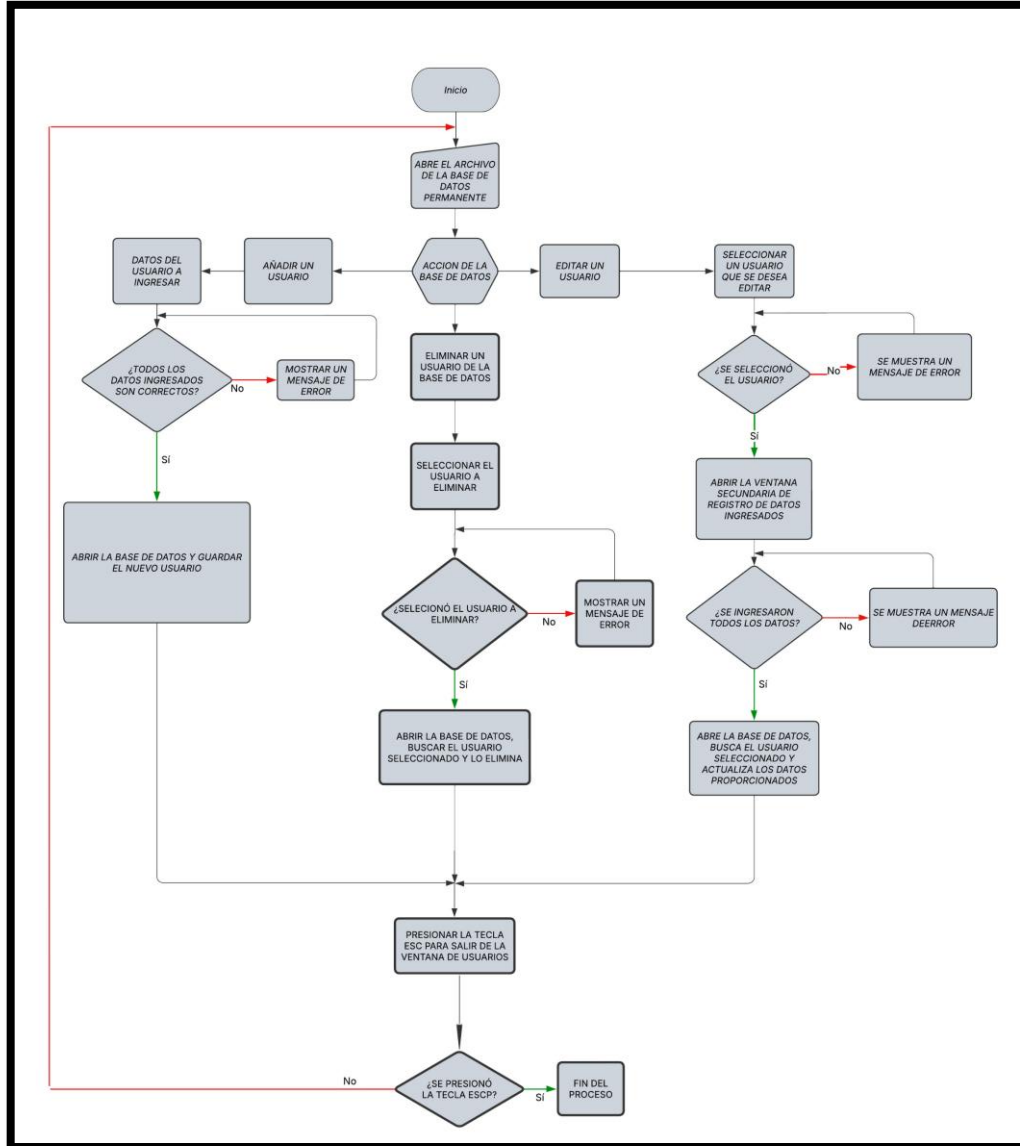
Diseño en AutoCAD de la zona donde se ha implementado el proyecto



ANEXO 3

FIGURA 79

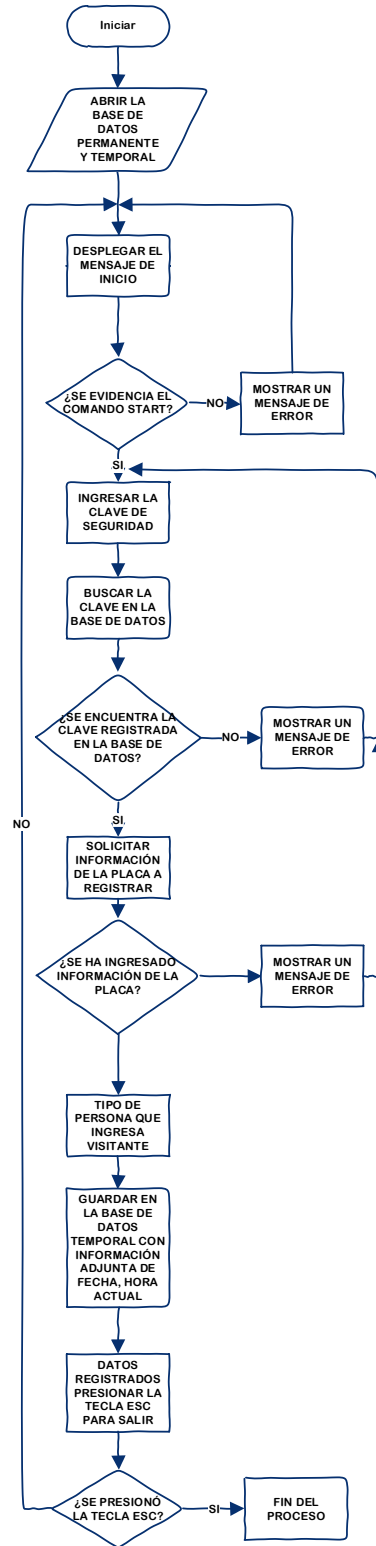
Diagrama de flujo de la base de datos permanente



ANEXO 4

FIGURA 80

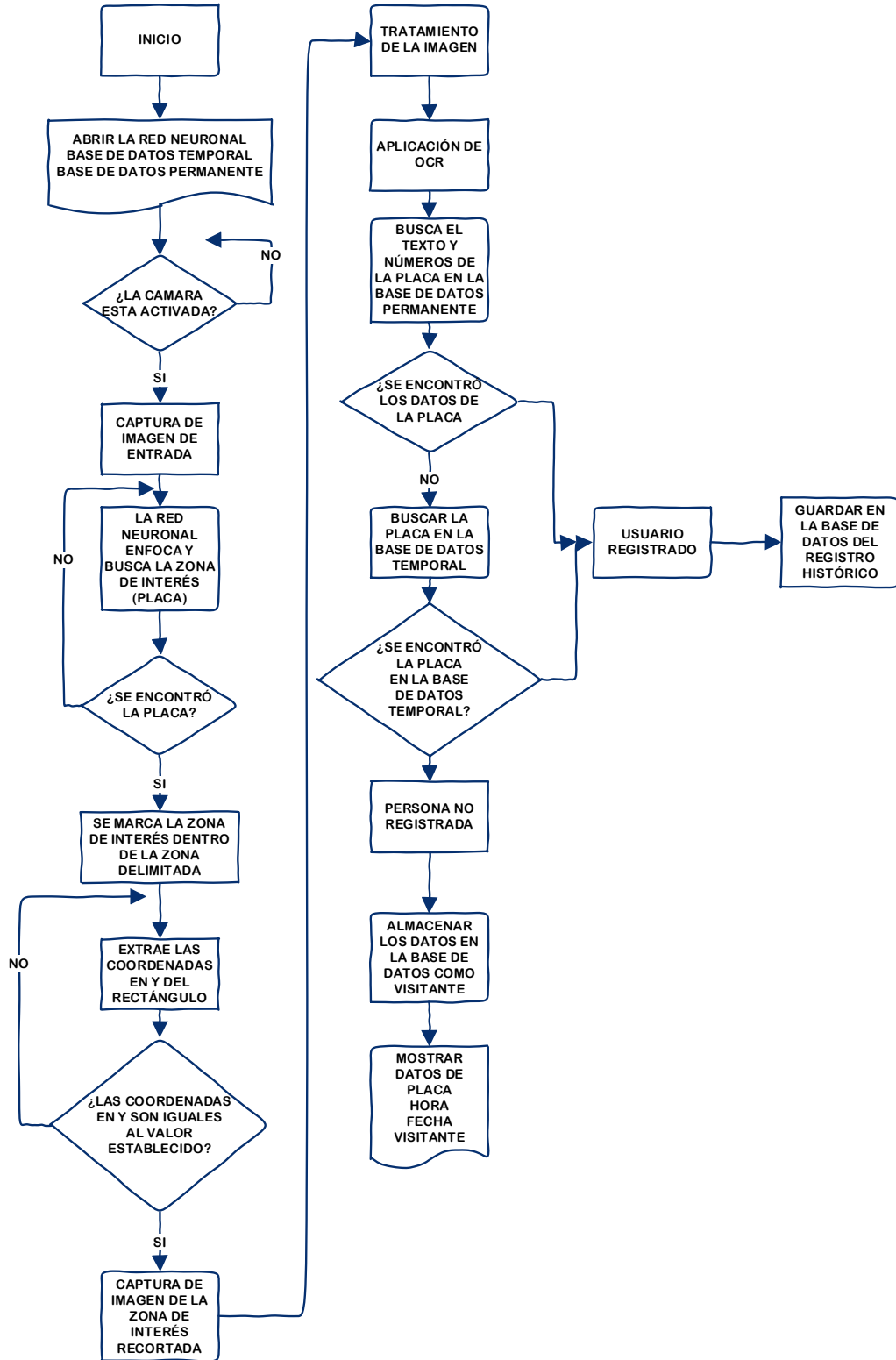
Diagrama de flujo de la base de datos temporal



ANEXO 5

FIGURA 81

Diagrama de flujo del funcionamiento total del prototipo



ANEXO 6

FIGURA 82

Carta de validación por parte de la empresa



Salcedo, 07 de febrero de 2025

Ing. Alicia Andrade Vera, Msc.

COORDINADORA DEL PROGRAMA MAESTRÍA EN ELECTRÓNICA Y
AUTOMATIZACIÓN

Presente.-

De mis consideraciones:

Por medio de la presente notificamos a usted que el señor, Edwin Eduardo Guanoluisa Huertas, con cedula 0502931744, cumplió con éxitos la implementación de un prototipo de seguridad con inteligencia artificial para monitorear el ingreso de vehículos a nuestra empresa CBVISION de la sucursal salcedo, el mismo que ha sido de gran ayuda en temas de seguridad y se encuentra totalmente operativo.

Es todo en cuanto puedo decir en honor a la verdad, el señor Edwin Eduardo Guanoluisa Huertas puede hacer uso del presente certificado para fines pertinentes, sin que afecte directa o indirectamente a la empresa.

Atentamente.

Ing. Luis Fernando Marin Molina

Jefe de agencia

CBVISION Salcedo

ANEXO 7

CÓDIGO GENERAL DE PROGRAMACIÓN DEL PROTOTIPO

procesamiento de imágenes en tiempo real con OpenCV y NumPy, mientras maneja múltiples hilos para mejorar la eficiencia.

```
from __future__ import print_function
import os
import sys
import time
import threading
import jetson.inference
import jetson.utils
import numpy as np
import cv2

net = jetson.inference.detectNet(argv=["--model=pl8b/ssd-mobilenet.onnx",
                                     "--labels=pl8b/labels.txt",
                                     "--input-blob=input_0", "--output-cvg=scores", "--output-bbox=boxes"],
                                threshold=0.4)

dispW=1280
dispH=720
flip=2
font=cv2.FONT_HERSHEY_SIMPLEX

# función similar a cv.VideoCapture
class FreshestFrame(threading.Thread):
    def __init__(self, capture, name='FreshestFrame'):
        self.capture = capture
        assert self.capture.isOpened()

# esto permite que read() se bloquee hasta que haya un nuevo frame o captura de imagen
self.cond = threading.Condition()

# esto nos permite detener el hilo
self.running = False

# manteniendo el frame más nuevo alrededor
self.frame = None

# pasar un número de secuencia permite que read() NO se bloquee
# si el número disponible actualmente es exactamente el que se solicita
self.latestnum = 0

self.callback = None

super().__init__(name=name)
self.start()

def start(self):
    self.running = True
    super().start()

def release(self, timeout=None):
    self.running = False
    self.join(timeout=timeout)
    self.capture.release()
```

```

def run(self):
    counter = 0
    while self.running:
        # bloque para frame nuevo
        (rv, img) = self.capture.read()

        assert rv
        counter += 1

        # guardar el nuevo frame obtenido
        with self.cond: # lock the condition for this operation
            self.frame = img if rv else None
            self.latestnum = counter
            self.cond.notify_all()

        if self.callback:
            self.callback(img)

def read(self, wait=True, seqnumber=None, timeout=None):

    with self.cond:
        if wait:
            if seqnumber is None:
                seqnumber = self.latestnum+1
            if seqnumber < 1:
                seqnumber = 1

            rv = self.cond.wait_for(lambda: self.latestnum >= seqnumber,
timeout=timeout)

            if not rv:
                return (self.latestnum, self.frame)

        return (self.latestnum, self.frame)

def main():
    cap = cv2.VideoCapture("rtsp://admin:Clsg1002@192.168.0.107:554/streaming/channels/0101 -
-input-codec=h264")
    global objetos
    objetos = 0
    cap.set(cv2.CAP_PROP_FRAME_WIDTH, dispW)
    cap.set(cv2.CAP_PROP_FRAME_HEIGHT, dispH)
    cap.set(cv2.CAP_PROP_FPS, 30)

    fresh = FreshestFrame(cap)

    # control de la cámara sin que se haga lento el proceso
    def callback(img):
        img = img[0:10, 0:10]
    fresh.callback = callback

    # bucle principal
    # obtener el frame más reciente, pero nunca el mismo dos veces (cnt aumenta)
    # ver read() para más detalles
    cnt = 0

    while True:

```

```

#tratamiento de la imagen,
# Rotación
M = cv2.getRotationMatrix2D((ancho//2,alto//2),85,1)
img = cv2.warpAffine(img,M,(ancho,alto))
ancho1 = img1.shape[1] #columnas
alto1 = img1.shape[0] # filas
# Rotación
M1 = cv2.getRotationMatrix2D((ancho1//2,alto1//2),85,1)
img1 = cv2.warpAffine(img1,M1,(ancho1,alto1))
frame=cv2.cvtColor(img,cv2.COLOR_BGR2RGBA).astype(np.float32)
frame=jetson.utils.cudaFromNumpy(frame)
detections=net.Detect(frame, 700, 700)
cv2.line(img,(0,105),(1200,105),(0,255,0),2)

#deteccion del zona de interes, placa dentro del sensor marcado en la programacion
for detect in detections:
    #print(detect)
    ID=detect.ClassID
    top=detect.Top
    left=detect.Left
    bottom=detect.Bottom
    right=detect.Right
    item=net.GetClassDesc(ID)
    print(item,top,left,bottom,right)

cv2.rectangle(img,(int(left),int(top)),(int(right+10),int(bottom+10)),(0,255,0),2)
mid = (top + bottom) /2
if 100+5 > mid > 100-5:
    objetos += 1
    imageOut = img1[int(top):int(bottom+10), int(left):int(right+10)]
    print("numero de autos="+ str(objetos))
    cv2.imwrite('img_tem/placa.png',imageOut)
    cv2.imwrite('img_tem/placa1.png',imageOut)

text = f'objetos: {objetos}'
cv2.putText(img, text, (175, 50), cv2.FONT_HERSHEY_SIMPLEX, 2, (0, 0, 255), 5)

imgmost = cv2.resize(img, (300, 300), fx=0, fy=0, interpolation=cv2.INTER_CUBIC)
cv2.imshow('detCam',imgmost)
#cv2.imshow('detCam',img)

key = cv2.waitKey(1)
if key == 27:
    break

print("done!")

fresh.release()
cap.release()
cv2.destroyAllWindows()
if __name__ == '__main__':
    main()

#extraccion de texto y letras de la imagen
# GUI
import tkinter as tk
from tkinter import ttk
from tkinter import *

```

```

from PIL import Image
from PIL import ImageTk
from tkinter import messagebox
# Video
import cv2
# Time
import datetime
# Database
import sqlite3
# OCR
import pytesseract
# librería GPIO
import Jetson.GPIO as GPIO
# Data Analytics
import pandas as pd
# Pin Definition
led_pin = 7
# Set up the GPIO channel
GPIO.setmode(GPIO.BOARD)
GPIO.setup(led_pin, GPIO.OUT, initial=GPIO.LOW)

# primera ventana
class Login:
    def __init__(self, window):
        # Window Definition
        self.wind = window
        self.wind.title("Sistema de acceso")
        self.wind.geometry("680x680+400+50")
        self.wind.resizable(False, False)
        # CB Image
        self.bg = ImageTk.PhotoImage(file="IMAGES/f1.jpg")
        self.bg_image = Label(self.wind, image=self.bg).place(x=0, y=0, relwidth=1, relheight=1)

        # cuadro de acceso
        Frame_login = Frame(self.wind, bg="white")
        Frame_login.place(x=295, y=295, height=215, width=320)

        # información de acceso y configuracion texto mostrado en la ventana de interfaz
        lbl_user = Label(Frame_login, text="Usuario", font=("futura", 15, "bold"), fg="gray",
        bg="white").place(x=0,
        self.txt_pass = Entry(Frame_login, font=("times new roman", 15), show="*", bg="lightgray")
        self.txt_pass.place(x=30, y=130, width=350, height=35)

        # Login Button
        Login_btn = Button(self.wind, command=self.login_function, text="Siguiente", cursor="hand2",
        fg="black",
        bg="#F49819", font=("futura", 15)).place(x=400, y=480)

        # Error Messages
        def login_function(self):
            if self.txt_pass.get() == "" or self.txt_user.get() == "":
                messagebox.showerror("Error", "Todos los campos son requeridos", parent=self.wind)

            elif self.txt_user.get() != "Admin2021" or self.txt_pass.get() != "Clsg1002":
                messagebox.showerror("Error", "Usuario o clave incorrectos", parent=self.wind)

            else:
                self.new_window()

```

```

# segunda ventana
class Video(ttk.Frame):
    def __init__(self, window):
        # definición de la ventana sistema OCR
        self.wind = window
        self.wind.title("Sistema OCR")
        self.wind.geometry("680x680+400+50")
        self.wind.resizable(False, False)

        # Logo en la ventana de OCR e interfaz
        self.bg = ImageTk.PhotoImage(file="IMAGES/PF2.jpg")
        self.bg_image = Label(self.wind, image=self.bg).place(x=0, y=0, relwidth=1, relheight=1)

        # OCR en tiempo Real
        self.pocr = Label(self.wind, text="", bg="#C7C7C0", fg='black', font=("futura", 18))
        self.pocr.place(x=170, y=450)

        # Informacion de usuario
        # Nombre
        self.nom = Label(self.wind, text="", bg="#CADADD", fg='black', font=("futura", 15))
        self.nom.place(x=435, y=130)
        # Placa
        self.plac = Label(self.wind, text="", bg="#CADADD", fg='black', font=("futura", 15))
        self.plac.place(x=435, y=233)
        # Tipo
        self.nca = Label(self.wind, text="", bg="#CADADD", fg='black', font=("futura", 15))
        self.nca.place(x=435, y=343)

        # Lectura de la imagen
        def elegir_imagen():
            # Base de Datos permanente
            conexion = sqlite3.connect('database/database.db')
            cursor = conexion.cursor()

            # Base de datos temporal
            con = sqlite3.connect('database/datatem.db')
            cursorObj = con.cursor()

            # Imagen de ruta
            path_image = 'img_tem/placa.png'

            # OCR Variables
            global text
            global num

            # Validacion
            if len(path_image) > 0:
                # tratamiento y lectura de la imagen, obtencion de coordenadas x, y
                image = cv2.imread(path_image)
                # Resize
                imgr = cv2.resize(image, (200, 100), fx=0, fy=0, interpolation=cv2.INTER_CUBIC)
                #imgr1 = cv2.resize(image, (200, 100), fx=0, fy=0, interpolation=cv2.INTER_CUBIC)
                # conversión de color de la imagen para mejores resultados
                img1 = cv2.cvtColor(imgr, cv2.COLOR_BGR2GRAY)
                imgr1 = cv2.cvtColor(imgr, cv2.COLOR_BGR2GRAY)
                # Filtro gaussiano
                blur = cv2.GaussianBlur(img1, (5, 5), 10)

```

```

        # guardar las imagenes
        cv2.imwrite('img_tem/pl1.png', pl1)
        cv2.imwrite('img_tem/pl2.png', pl2)
        # OCR de la primera imagen (letras)
        custom_config1 = r'--oem 3 --psm 10 -c
tessedit_char_whitelist=ABCDEFGHIJKLMNOPQRSTUVWXYZ'
        text = pytesseract.image_to_string(pl1, config=custom_config1)
        # OCR de la segunda imagen (numeros)
        custom_config = r'--oem 3 --psm 10 -c tessedit_char_whitelist=0123456789'
        numb = pytesseract.image_to_string(pl2, config=custom_config)
        # sobrescribir imagen principal
        cv2.rectangle(pl, (0, 0), (450, 230), (0, 0, 0), -1)
        cv2.imwrite('img_tem/placa.png', pl)
    else:
        # asignación de valores estaticos
        numb = '00'
        text = 'pk'
        # lectura de las dos partes de la imagen
        let = cv2.imread('img_tem/pl1.png')
        num = cv2.imread('img_tem/pl2.png')
        # OCR de la primera imagen (letras)
        custom_config2 = r'--oem 3 --psm 10 -c
tessedit_char_whitelist=ABCDEFGHIJKLMNOPQRSTUVWXYZ'
        text1 = pytesseract.image_to_string(let, config=custom_config2)
        # OCR de la segunda imagen (numeros)
        custom_config3 = r'--oem 3 --psm 10 -c tessedit_char_whitelist=0123456789'
        numb1 = pytesseract.image_to_string(num, config=custom_config3)
        # mostrar el resultado final del OCR en el GUI
        global PLAC1
        PLAC1 = str(text1.strip() + "-" + numb1.strip())
        self.pocr['text'] = '{}'.format(PLAC1)

    # lectura de la imagen
    imagemost = cv2.imread('img_tem/placa1.png')
    # cambiar tamaño
    imagemostr = cv2.resize(imagemost, (200, 100), fx=0, fy=0,
interpolation=cv2.INTER_CUBIC)
    # conversión del color
    imageToShow = cv2.cvtColor(imagemostr, cv2.COLOR_BGR2RGB)
    # mostrar imagen en el GUI
    im = Image.fromarray(imageToShow)
    img = ImageTk.PhotoImage(image=im)
    lblInputImage.configure(image=img)
    lblInputImage.image = img
    # actualizar después de xx milisegundos para continuar el proceso
    lblInputImage.after(500, elegir_imagen)

    # tiempo y fecha (real)
    time = datetime.datetime.now()
    fechaLabel = Label(self.wind, text=time.strftime("%Y - %m - %d\n%H:%M"))
    # mostrar tiempo y fecha en GUI
    fechaLabel.config(bg='#C7C7C0', font=("futura", 15))
    fechaLabel.place(x=135, y=590)
    # Fecha y hora como variables individuales
    f = time.strftime("%Y-%m-%d")
    h = time.strftime("%H:%M")

    # Concatenación de letras y números (para comparar)

```

```

PLAC = str(text.strip() + "-" + numb.strip())

# Buscar usuario por placa en base de datos permanente
sql = ("\"SELECT nombre, placa, clave FROM Usuarios WHERE placa = '%s'\" % (PLAC))
cursor.execute(sql)
usuario = cursor.fetchone()

# Buscar usuario por placa en base de datos temporal
sql2 = ("\"SELECT nombre, placa, tipo, fecha, hora, fut FROM Bot WHERE placa = '%s'\" %
(PLAC))
cursorObj.execute(sql2)
usuariotemp = cursorObj.fetchone()

# si se encuentra el usuario en la base de datos permanente
if str(usuario) != 'None':
    # User data
    nombreb = usuario[0]
    placab = usuario[1]
    claveb = 'Grupo 1; Grupo 2; Grupo 3; Recaudación'

# mostrar el dato de usuario en GUI
self.nom['text'] = '{}'.format(nombreb)
self.plac['text'] = '{}'.format(placab)
self.nca['text'] = '{}'.format(claveb)
# Guarda la información del usuario en un registro general.
path = '/home/zeus/Desktop/registro.csv'
df1 = pd.read_csv(path)
SeriesA = df1['Nombre']
SeriesB = df1['Placa']
SeriesC = df1['Tipo']
SeriesD = df1['Fecha']
SeriesE = df1['Hora']
A = pd.Series(nombreb)
B = pd.Series(placab)
C = pd.Series('Residente')
D = pd.Series(f)
E = pd.Series(h)
SeriesA = SeriesA.append(A)
SeriesB = SeriesB.append(B)
SeriesC = SeriesC.append(C)
SeriesD = SeriesD.append(D)
SeriesE = SeriesE.append(E)
df2 = pd.DataFrame(
    {"Nombre": SeriesA, "Placa": SeriesB, "Tipo": SeriesC, "Fecha": SeriesD, "Hora":
SeriesE})
df2.to_csv(path, index=False)

# Lugar donde se muestra la imagen.
lblInputImage = Label(self.wind)
lblInputImage.place(x=120, y=230)

elegir_imagen()

# Nueva Ventana
def new_window(self):
    self.newWindow = Toplevel(self.wind)
    self.app = Usuarios(self.newWindow)

```



```

# Tabla de datos
self.tree = ttk.Treeview(self.wind, height=22, columns=("placa", "clave"))
self.tree.grid(row=5, column=0, columnspan=3)
self.tree.heading("#0", text='Nombre', anchor=CENTER)
self.tree.heading("placa", text='Placa', anchor=CENTER)
self.tree.heading("clave", text='Clave', anchor=CENTER)

# Barra de desplazamiento
Scrollbar = ttk.Scrollbar(self.wind, orient=tk.VERTICAL, command=self.tree.yview)
Scrollbar.grid(row=5, column=3, sticky="nse")
self.tree.configure(yscrollcommand=Scrollbar.set)

# Mostrar información en la tabla
self.get_usuarios()

def run_query(self, query, parameters=()):
    with sqlite3.connect(self.db_name) as conn:
        cursor = conn.cursor()
        result = cursor.execute(query, parameters)
        conn.commit()
    return result

def get_usuarios(self):
# Añadir nuevos usuarios
def add_usuarios(self):
    if self.validation():
        # Escribir la información en la tabla
        query = "INSERT INTO Usuarios VALUES (NULL, ?, ?, ?)"
        parameters = (self.name.get(), self.placa.get(), self.clave.get())
        self.run_query(query, parameters)
        messagebox.showinfo("Agregar usuario",
            'El usuario {} ha sido agregado correctamente'.format(self.name.get()),
            parent=self.wind)
        self.name.delete(0, END)
        self.placa.delete(0, END)
        self.clave.delete(0, END)

# Mensaje de error
else:
    messagebox.showerror("Error",
        'Todos los campos son requeridos'.format(self.name.get()), parent=self.wind)
self.get_usuarios()

# Borrar usuarios
def delete_usuario(self):
    try:
        self.tree.item(self.tree.selection())['text'][0]
    except IndexError as e:
        messagebox.showerror("Error", 'Por favor seleccione un usuario', parent=self.wind)
    return

# obtener información de usuarios
nombre = self.tree.item(self.tree.selection())['text']
placa = self.tree.item(self.tree.selection())['values'][0]
clave = self.tree.item(self.tree.selection())['values'][1]

# Editar usuarios
def edit_usuario(self):

```

```

try:
    self.tree.item(self.tree.selection())['text'][0]
except IndexError as e:
    messagebox.showerror("Error", 'Por favor seleccione un usuario', parent=self.wind)
    return

# obtener información de usuarios
nombrev = self.tree.item(self.tree.selection())['text']
placav = self.tree.item(self.tree.selection())['values'][0]
clavev = self.tree.item(self.tree.selection())['values'][1]
self.edit_wind = Toplevel()
self.edit_wind.title = 'Editar Usuario'

# placa anterior
Label(self.edit_wind, text='Placa anterior').grid(row=2, column=1)
Entry(self.edit_wind, textvariable=StringVar(self.edit_wind, value=placav), state='readonly').grid \
    (row=2, column=2)

# Nueva placa
Label(self.edit_wind, text='Placa nueva').grid(row=3, column=1)
placan = Entry(self.edit_wind)
placan.grid(row=3, column=2)

# Boton de actualizar
Button(self.edit_wind, text='Actualizar',
        command=lambda: self.edit_records(nombren.get(), nombrev, placan.get(), placav, claven.get(),
        clavev)).grid(row=6, column=2, sticky=W)

# actualizar datos de usuario
def edit_records(self, nombren, nombrev, placan, placav, claven, clavev):
    query = 'UPDATE Usuarios SET nombre = ?, placa = ?, clave=? WHERE nombre = ? and placa=?
and clave=?'
    parameters = (nombren, placan, claven, nombrev, placav, clavev)
    self.run_query(query, parameters)
    self.edit_wind.destroy()
    messagebox.showinfo("Editar usuario", 'El usuario { } ha sido actualizado
correctamente'.format(nombren),
        parent=self.wind)
    self.get_usuarios()

window = Tk()
application = Login(window)
window.mainloop()

#Subprocess para mejor rendimiento del sistema
import subprocess

scripts_paths = ("GUIFRC2.py", "opendec2.py")

ps = [subprocess.Popen(["python3", script]) for script in scripts_paths]
exit_codes = [p.wait() for p in ps]

if not any(exit_codes):
    print("Todos los procesos terminaron con éxito")
else:
    print("Algunos procesos terminaron de forma inesperada.")

```