UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE GUAYAQUIL

CARRERA DE ELECTRÓNICA Y AUTOMATIZACIÓN

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE IDENTIFICACIÓN DE COMPONENTES ELÉCTRICOS EN MÓDULOS DIDÁCTICOS MEDIANTE VISIÓN ARTIFICIAL

Trabajo de titulación previo a la obtención del

Título de Ingeniero en Electrónica

AUTORES: GUSTAVO ADRIAN BAQUE CATAGUA

DENNIS FERNANDO BENAVIDES PONCE

TUTOR: Ing. LIVINGTON MIRANDA

Guayaquil – Ecuador

2024 - 2025

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros, Gustavo Adrián Baque Catagua con cedula de ciudadanía N°0950139097 y Dennis Fernando Benavides Ponce con cedula de ciudadanía N°1205364571, afirmamos que:

Somos las autores y responsables del presente trabajo, y damos nuestra autorización para que la Universidad Politécnica Salesiana, sin fines de lucro, pueda utilizar, diseminar, replicar o hacer público en su totalidad o en partes este trabajo de titulación.

Guayaquil, 03 de agosto del año 2024.

Atentamente,

Gustavo Adrián Baque Catagua
095013909-7

Dennis Fernando Benavides Ponce
120536457-1

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

Nosotros, Gustavo Adrián Baque Catagua con documento de identificación N°0950139097 y Dennis Fernando Benavides Ponce con documento de identificación N°1205364571, manifestamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto Técnico: "Diseño e implementación de un sistema de identificación de componentes eléctricos en módulos didácticos mediante visión artificial", el cual ha sido desarrollado para optar por el título de: Ingeniero en Electrónica, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

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

Guayaquil, 03 de agosto del año 2024.

Atentamente,

Gustavo Adrián Baque Catagua
095013909-7

Dennis Fernando Benavides Ponce
120536457-1

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Livington Alfredo Miranda Delgado con documento de identificación N°0930635172, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE IDENTIFICACIÓN DE COMPONENTES ELÉCTRICOS EN MÓDULOS DIDÁCTICOS MEDIANTE VISIÓN ARTIFICIAL, realizado por Baque Catagua Gustavo Adrián con documento de identificación N°0950139097 y Benavides Ponce Dennis Fernando con documento de identificación N°1205364571, obteniendo como resultado final el trabajo de titulación bajo la opción de Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Guayaquil, 03 de agosto del año 2024.

Atentamente,

Ing. Livington Alfredo Miranda Delgado
0930635172

DEDICATORIA

Deseo dedicar este trabajo a mi madre, quien ha sido un apoyo incondicional durante toda mi carrera académica. Su constante aliento y sacrificios han sido la motivación detrás de cada uno de mis logros. Agradezco también a mis profesores y amigos, cuyo conocimiento compartido y compañerismo han enriquecido mi formación como ingeniero en electrónica. Además, este proyecto está dedicado a todos aquellos que desafiaron mis capacidades, lo cual me llevo a ser más determinado, disciplinado y me motivaron a alcanzar mis metas.

Dennis Benavides Ponce

Dedico este proyecto a mi familia, por su afecto brindado de manera incondicional y su constante apoyo durante mi formación académica. A mis padres, por su sacrificio y confianza en mí, por ser fuentes de inspiración y motivación en cada etapa de este camino. También dedico este proyecto a mis mentores y profesores, cuyo conocimiento y orientación han sido fundamentales para mi desarrollo como ingeniero en electrónica. Sin ellos, este logro no habría sido posible.

Gustavo Baque Catagua

AGRADECIMIENTO

Agradezco profundamente a mis padres, quienes me han brindado su apoyo de manera incondicional en cada uno de los pasos durante mi formación académica. Su amor, paciencia y sacrificios han sido la fuerza motriz detrás de mis logros. Finalmente, agradezco a todos aquellos que, de una forma u otra, contribuyeron a la culminación de este trabajo. Su apoyo y contribuciones, aunque a veces imperceptibles, han sido vitales para alcanzar este objetivo.

Dennis Benavides Ponce

Deseo agradecer a mis compañeros de clase y amistades, con quienes he compartido innumerables horas de estudio y trabajo. Su colaboración y amistad han sido esenciales para la culminación de este proyecto. También quiero agradecer a mi compañero de investigación por su colaboración y entusiasmo, que hicieron de este proyecto una experiencia desafiante y memorable.

Gustavo Baque Catagua

Resumen

Con el transcurso del tiempo, la inteligencia artificial (IA) ha tomado una relevancia muy importante en múltiples campos, incluida la educación. Este proyecto de titulación tiene como objetivo diseñar e implementar un sistema de identificación de componentes eléctricos en módulos didácticos mediante visión artificial. Con esta iniciativa se pretende automatizar el proceso de identificación de componentes, reduciendo los errores humanos y optimizando el tiempo empleado por estudiantes, técnicos y profesores en los laboratorios de Electrónica y Automatización de la Universidad Politécnica Salesiana sede Guayaquil. A través de la creación de un conjunto de datos diversificado, el etiquetado de imágenes, el desarrollo de un modelo de clasificación y su validación en condiciones reales, se pretende lograr una mejoría importante en la gestión de los módulos didácticos.

*Palabras Clave***:** Inteligencia Artificial, Módulos Didácticos, Precisión, Componentes Eléctricos, Eficiencia, Automatizado, Errores Humanos.

Abstract

Over time, artificial intelligence (AI) has taken on very important relevance in multiple fields, including education. This degree project aims to design and implement an identification system for electrical components in teaching modules using artificial vision. This initiative aims to automate the component identification process, reducing human errors and optimizing the time spent by students, technicians and professors in the Electronics and Automation laboratories of the Salesiana Polytechnic University, Guayaquil. Through the creation of a diversified data set, the labeling of images, the development of a classification model and its validation in real conditions, it is intended to achieve a significant improvement in the management of the teaching modules.

Keywords: Artificial Intelligence, Didactic Modules, Precision, Electrical Components, Efficiency, Automated, Human Errors.

I Contenido

INTRODUCCIÓN 19

II PROBLEMA 20

2.1 Justificación 21

2.2 Delimitación del problema 22

2.3 Objetivo general 23

2.4 Objetivos específicos 23

III FUNDAMENTO TEÓRICO 24

3.1 Inteligencia Artificial 24

3.2 Visión Artificial 25

3.3 Python 26

3.4 PyCharm 26

3.5 Machine learning 27

3.6 Roboflow 28

3.7 Redes Neuronales Convolucionales 28

3.8 Procesamiento de imágenes 29

3.9 Módulos didácticos Industriales 30

3.10 Matriz de confusión 31

3.11 Aprendizaje Supervisado 32

IV MARCO METODOLÓGICO 34

4.1 Preparación y Adquisición de Datos 35
4.1.1 Selección de la Cámara y Configuración del Entorno 35
4.1.2 Creación del Dataset 35
4.1.3 Etiquetado de Imágenes 38
4.1.4 Preparación del Conjunto de Datos 42
4.2 Desarrollo del Software 46
4.2.1 Selección y Configuración del Entorno de Desarrollo 46
4.2.2 Implementación del Modelo de Identificación 47
4.2.3 Integración del Software con la Cámara 50
4.3 Creación de la Interfaz de interacción con el usuario 54
4.3.1 Pantalla principal o de inicio 54
4.3.2 Pantalla de Identificación de componentes 55
4.3.3 Pantalla de base de datos 56
V Análisis de Resultados 57
5.1 Crear un Conjunto de Datos Diversificado con los Objetos a Identificar 57
5.2 Realizar el Etiquetado de Cada Objeto en el Conjunto de Imágenes Según su Categoría 57
5.3 Desarrollar y Entrenar un Modelo de Clasificación 58
5.4 Validar el Sistema en Condiciones Reales 60
CRONOGRAMA 63

PRESUPUESTO 64

CONCLUSIONES 65

RECOMENDACIONES 66

REFERENCIAS BIBLIOGRAFICAS 67

VI Bibliografía 67

ANEXOS 72

Ilustraciones

Ilustración 1: Representación de una Red Neuronal Biológica y una Artificial (Franco, y otros, s.f.). 24

Ilustración 2: Detección de Objetos con Visión Artificial (Gavilán, 2023). 25

Ilustración 3: Entorno de PyCharm..... 27

Ilustración 4: Red Neuronal Convolucional (De la Rosa, 2016). 29

Ilustración 5: Esquema de la descomposición de colores en una imagen en cada plano (Borella, 2022). 30

Ilustración 6: Módulo Didáctico Industrial del Laboratorio de Automatización de la UPS..... 31

Ilustración 7: Matriz de confusión sin normalizar y matriz de confusión normalizada (García, 2023). 32

Ilustración 8: Técnicas del Aprendizaje Supervisado (Larroso, 2023)..... 33

Ilustración 9: Diagrama de flujo del sistema de visión artificial. 34

Ilustración 10: Cámara Web Argomtech Cam50 (Argom tech, s.f.). 35

Ilustración 11: Búsqueda de imágenes de los componentes en sitios web. 36

Ilustración 12: Dataset de imágenes de los componentes eléctricos..... 37

Ilustración 13: Comparación de aplicaciones para etiquetado de imágenes..... 39

Ilustración 14: Plataforma Roboflow, ventana donde se escogerá el tipo de proyecto a realizar. 40

Ilustración 15: Ventana para la creación de las clases en Roboflow. 40

Ilustración 16: Ventana para realizar la carga de imágenes, ya sea de forma individual o en grupo a través de una carpeta. 41

Ilustración 17: Ventana donde se realiza el etiquetado de imágenes..... 42

Ilustración 18: Ventana que muestra los resultados del etiquetado de imágenes. 42

Ilustración 19: Ventana de división de subconjuntos para entrenamiento, validación y prueba. . 43
Ilustración 20: Plataforma Leonardo AI, opción image to image. 44
Ilustración 21: Resultados obtenidos en Leonardo AI para la generación de imágenes con IA… 44
Ilustración 22: Ventana de data aumentada. 45
Ilustración 23: Ventana del resultado que se obtuvo al realizar la data aumentada de imágenes. 46
Ilustración 24: Entorno de Desarrollo IDE PyCharm. 47
Ilustración 25: Modelos de pre-entrenados de YOLO (Github, s.f.). 49
Ilustración 26: Modelo Defectuoso Entrenado con YOLOv8m.pt. 49
Ilustración 27: Modelo valido entrenado desde la plataforma Roboflow. 50
Ilustración 28: Interacción entre la interfaz de la app y la cámara. 51
Ilustración 29: Interacción entre la interfaz y la cámara. 51
Ilustración 30: Matriz de Confusión. 52
Ilustración 31: Matriz de Confusión Normalizada. 53
Ilustración 32: Interfaz Principal de la aplicación de visión artificial. 54
Ilustración 33: Pantalla de identificación de componentes del módulo didáctico. 55
Ilustración 34: Pantalla de base de datos de los componentes. 56
Ilustración 35: Etapa final de etiquetado de imágenes. 58
Ilustración 36: Curva de precisión del modelo. 59
Ilustración 37: Curva obtenida del Recall del modelo. 59
Ilustración 38: Captura realizada desde la interfaz. 60
Ilustración 39: Validación del sistema en los laboratorios de Automatización de la Universidad Politécnica Salesiana sede Guayaquil. 61

Ilustración 40: Validación del sistema en condiciones reales en el laboratorio de Automatización, con los módulos didácticos. 62
Ilustración 41: Validación de la identificación en la parte exterior de los módulos didácticos en los laboratorios de Automatización. 62
Ilustración 42: Cronograma de actividades..... 63

INTRODUCCIÓN

La Inteligencia Artificial (IA) y la visión artificial transforman la manera de interactuar con la tecnología, incluida la educación. Se ha demostrado que estas tecnologías automatizan tareas que antes dependían en gran medida de la intervención humana. La visión artificial, la cual posibilita a las máquinas interpretar y entender el entorno visual, se ha vuelto esencial en varios sectores, desde la manufactura hasta la atención médica. Su aplicación en educación abre nuevas posibilidades para mejorar y explorar procesos de enseñanza y aprendizaje, especialmente en disciplinas técnicas y científicas.

En la Universidad Politécnica Salesiana sede Guayaquil, el uso de módulos didácticos es importante para la capacitación práctica de los estudiantes de Electrónica y Automatización. Estos módulos simulan entornos industriales reales, lo que permite a los estudiantes adquirir habilidades prácticas y enfrentar desafíos en su vida profesional. Sin embargo, la identificación manual de los componentes eléctricos por los que está compuesto un módulo es propensa a errores y consume un tiempo valioso. Este proyecto pretende abordar este problema mediante la planificación y ejecución de un sistema de identificación automatizado con el uso de la visión artificial. El sistema no sólo reducirá los errores humanos, sino que también optimizará el tiempo dedicado por estudiantes y técnicos a la identificación de componentes.

El proyecto se centrará en varias fases clave: creación de un conjunto diversificado de datos, etiquetado de imágenes, desarrollo de un modelo de clasificación utilizando técnicas de aprendizaje automático y validación del sistema en situaciones reales del laboratorio. Esta integración de la visión artificial en los laboratorios de Electrónica y Automatización representa un avance significativo en la modernización de los métodos educativos y puede servir como modelo para implementar tecnologías avanzadas en otros campos educativos.

II PROBLEMA

Actualmente la IA juega un papel crucial en diversos ámbitos, pero en especial en el educativo. Este enfoque otorga a la inteligencia artificial una participación relevante, que complementa el análisis humano e intenta superar la idea tradicional de la inteligencia artificial como tecnología de reemplazo humano, especialmente en educación (Cukurola, Kent, & Luckin, 2019).

La implementación de módulos didácticos son un aporte fundamental en la educación práctica de los alumnos de Ingeniería Electrónica y Automatización en la Universidad Politécnica Salesiana sede Guayaquil. Con el transcurso del tiempo ha quedado evidenciado que los módulos didácticos no solo son un aporte para la educación práctica de los alumnos, sino que además los prepara para afrontar retos que se les presenten en el mundo laboral al ser expuestos a una diversidad de tecnologías y equipos que se utilizan en las industrias (Rivas & Jaramillo, 2020).

No obstante, existen desafíos recurrentes en el uso de módulos didácticos como es la identificación de sus componentes eléctricos de manera rápida y precisa, Este proceso, al ser realizado de forma manual esta propenso a errores, especialmente entre principiantes y técnicos que aún no se familiarizan del todo con los componentes. Una identificación de forma incorrecta conlleva no solo al consumo de tiempo valioso, además puede llevar a errores en la implementación de prácticas y experimentos (Ferro, 2024).

Abordando esta problemática, la visión artificial surge como una solución prometedora. La visión artificial es una de las tecnologías que prometen mejorar los procesos y facilitar el trabajo humano. Se pretende dar solución a problemas actuales como el porcentaje de error en tareas específicas, la baja productividad por sobrecarga de trabajo, el incremento de costos en mano de obra y la inseguridad que puede producir la participación humana en un entorno de trabajo (Carvalho, 2023).

En la UPS sede Guayaquil, se desarrolló un proyecto en el que, mediante visión artificial, se realizó una segmentación de piezas electrónicas en placas electrónicas que busca automatizar tareas diarias en la compañía Multiservicios Electrónicos Politécnicos localizada en Guayaquil, por problemas como el desconocimiento de principiantes o técnicos con respecto a los componentes electrónicos porque ocupaban más tiempo en la identificación visual de

componentes. La implementación de este sistema les permitió automatizar la tarea de identificar componentes e incluso capacitar al personal (Navarrete & Yanse, 2024).

En los Laboratorios de Electrónica y Automatización, el campus Centenario de la Universidad Politécnica Salesiana planifica y controla mantenimientos de equipos haciendo un listado a mano que se tabula en tablas para cada laboratorio, lo que provoca errores por fatiga o cansancio, lo que conlleva a que el personal designado no pueda aprovechar su tiempo en otras tareas de gestión académica.

2.1 Justificación

El desarrollo de un sistema de identificación automatizada de elementos en módulos didácticos mediante la visión artificial es un gran avance tecnológico para brindar una herramienta de apoyo y digitalización de la información en entornos educativos. Actualmente se han desarrollado muchos proyectos donde se ha demostrado el éxito de integrar tecnologías avanzadas en procesos de identificación y clasificación. Por ejemplo, un proyecto reciente desarrollado en la Universidad Politécnica Salesiana sede Guayaquil, demostró que la implementación de visión artificial en la identificación y clasificación de enfermedades de las plantas de tomates redujo los errores en un alto porcentaje, optimizando de manera significativa el tiempo de procesamiento y mejorando la precisión para obtener una prevención y generar un tratamiento a seguir en tiempo real (Valenzuela, 2021).

Un ejemplo relevante es el proyecto donde se implementó un sistema similar para el reconocimiento de personas utilizando un Drone con redes neuronales artificiales. En donde los resultados obtenidos mostraron una efectividad del 80% en la velocidad y precisión de la identificación de personas, lo que llevó a una reducción en la inseguridad y falta de control dentro de la Universidad Politécnica Salesiana sede Quito Campus Sur, ofreciendo una solución al problema mediante un sistema basado en inteligencia artificial (Pazmiño, 2023).

Por otra parte, en el trabajo sobre la identificación de objetos a partir de técnicas de visión por computador y aprendizaje automático, se menciona que lograron detectar barcos cargueros y grúas en el puerto de Sevilla. Además, se plantea contabilizar ciertos objetos para llevar un registro para un posterior análisis de datos que les permita tomar medidas respecto a la información obtenida (Martín et al., 2022).

Por lo tanto, realizar un sistema automatizado de identificación de elementos en módulos didácticos de automatización brindará beneficios como: la reducción de errores humanos, la optimización del tiempo de mantenimiento, la mejora de la eficiencia operativa, la minimización de los costos asociados con el mantenimiento y un análisis con los datos obtenidos para estos procesos.

2.2 Delimitación del problema

Este proyecto se desarrollará en los laboratorios de Electrónica y Automatización de la Universidad Politécnica Salesiana sede Guayaquil, en concreto utilizando los módulos didácticos de formación profesional para los alumnos de Ingeniería Electrónica y Automatización.

OBJETIVOS

2.3 Objetivo general

Diseñar un sistema de identificación automatizada de componentes eléctricos en módulos didácticos del Laboratorio de la Carrera Electrónica y Automatización de la Universidad Politécnica Salesiana sede Guayaquil.

2.4 Objetivos específicos

1. Crear un conjunto de datos diversificados con los objetos a identificar.
2. Realizar el etiquetado de cada objeto en el conjunto de imágenes según su categoría.
3. Desarrollar y entrenar un modelo de clasificación.
4. Validar el sistema en condiciones reales.

III FUNDAMENTO TEÓRICO

3.1 Inteligencia Artificial

La Inteligencia Artificial (IA) se define como la capacidad de las máquinas para realizar tareas que normalmente requieren inteligencia humana, usando algoritmos y aprendiendo de los datos para tomar decisiones. A diferencia de los humanues, los sistemas de IA pueden procesar grandes volúmenes de información sin necesidad de descanso y con menos errores. La habilidad de aprender y decidir ha permitido que la IA asuma muchas tareas previamente reservadas para los humanos, mejorando la eficiencia en diversos ámbitos. Sin embargo, el crecimiento acelerado de la IA también requiere que estemos atentos a las posibles desventajas que pueda conllevar (Rouhiainen, 2018).

En la Ilustración 1, se muestra como las redes neuronales artificiales tratan de imitar el comportamiento del cerebro, por lo cual su estructura es muy similar a las redes neuronales biológicas.

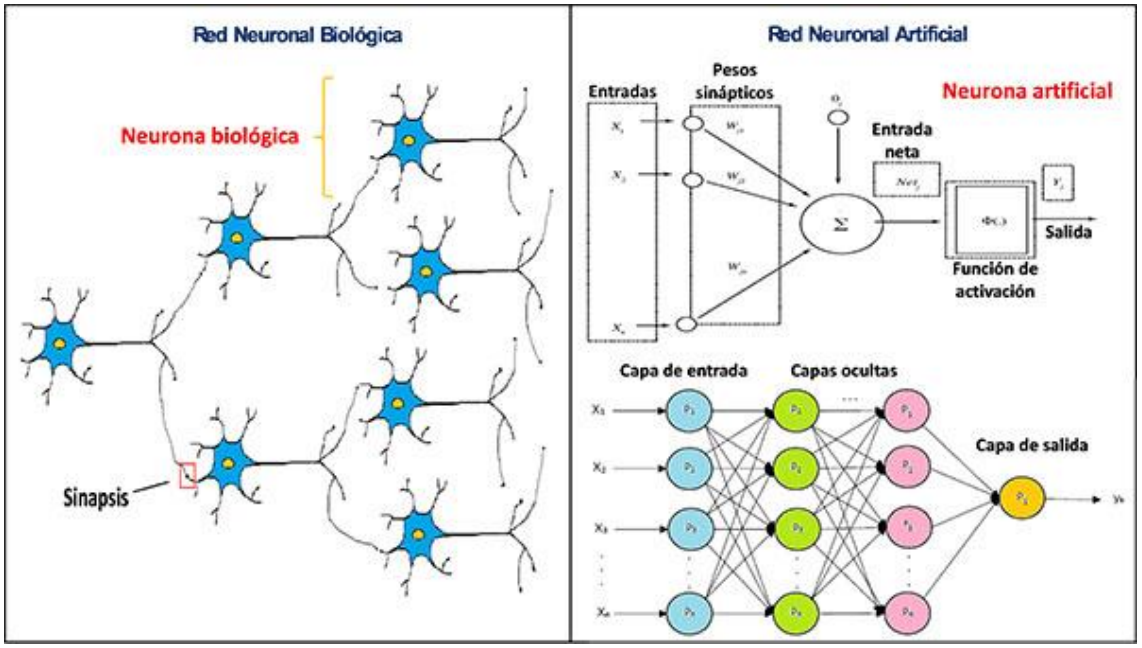

Ilustración **1:** Representación de una Red Neuronal Biológica y una Artificial **(Franco, y otros, s.f.).**

3.2 Visión Artificial

La visión artificial, una rama crucial de la inteligencia artificial, capacita a los sistemas informáticos para interpretar y analizar imágenes y videos, emulando la habilidad humana de percibir y entender el entorno visual. A nivel industrial, esta tecnología juega un papel fundamental en una variedad de aplicaciones, desde el control de calidad automatizado hasta la detección de defectos y la clasificación de productos, impulsando la innovación y la competitividad en diversos sectores industriales (Martin, 2023).

Con la ayuda de cámaras, datos y algoritmos en vez de retinas, nervios ópticos y una corteza visual, es como la visión artificial enseña a las maquinas a ejecutar tareas en tiempos más reducidos. En el caso de un sistema que se entrena para inspeccionar los productos es capaz de superar al humano de forma rápida debido a que puede analizar muchos productos o procesos en segundos mientras va encontrando fallas o problemas minuciosos que no son perceptibles para el ojo humano (Rosquez, s.f.)

En la ilustración 2, se observa el uso de la visión artificial para la detección de objetos en el control y seguridad de los peatones y conductores en la vía pública.

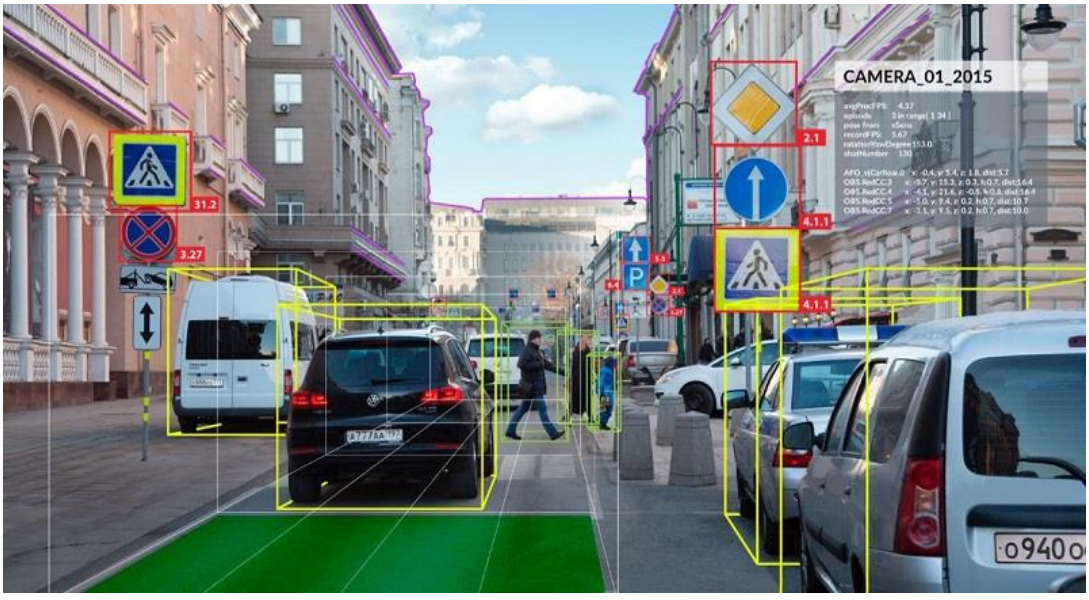

Ilustración **2:** Detección de Objetos con Visión Artificial **(Gavilán, 2023).**

3.3 Python

Python es un lenguaje de programación de código libre, creado por Guido van Rossum en 1991. Se trata de un lenguaje orientado a objetos, sencillo de interpretar. Su flexibilidad como lenguaje de uso general lo hace ideal para satisfacer las demandas de diversas industrias, como el desarrollo de sitio web, el análisis de datos, machine learning, inteligencia artificial, blockchain y los videojuegos (Londoño, 2023).

Python, consta de algoritmos que son utilizados para la visión por computadora y el aprendizaje profundo, esta herramienta es importante para la detección de objetos usando algoritmos como YOLO (You Only Look Once). Es un lenguaje de programación que permite acceder a varias bibliotecas y marcos de trabajo, como TensorFlow, Pytorch y OpenCV, lo cual hacen más fácil la implementación, entrenamiento y optimización de modelos de redes neuronales convolucionales (CNN). Además, Python tiene la facilidad de integrarse de forma eficaz con distintas plataformas como CUDA, logrando aprovechar el poder que tienen las GPU para acelerar la interfaz y el procesamiento de imágenes y videos, haciendo que sea posible la detección rápida y precisa de objetos en diversas aplicaciones (Jurado, 2024).

3.4 PyCharm

Es un entorno de desarrollo integrado (IDE) de JetBrains, está disponible en dos versiones, una gratuita denominada PyCharm Community y otra de pago PyCharm Professional. Ofrece características avanzadas como resaltado de sintaxis, completado de código e inspección para encontrar errores. Su capacidad para navegar entre archivos y refactorizar código facilita el desarrollo, mientras que la integración con sistemas de control de versiones como Git simplifica la gestión de cambios. La versión Professional incluye funciones adicionales para desarrollo web, bases de datos, trabajo remoto y análisis de datos, con soporte para frameworks web, herramientas de bases de datos y Jupyter Notebooks. Además, es altamente personalizable y se integra con tecnologías como Docker y Ansible (Escalante, 2023).

En la ilustración 3, se observa la ventana de PyCharm para crear proyectos.

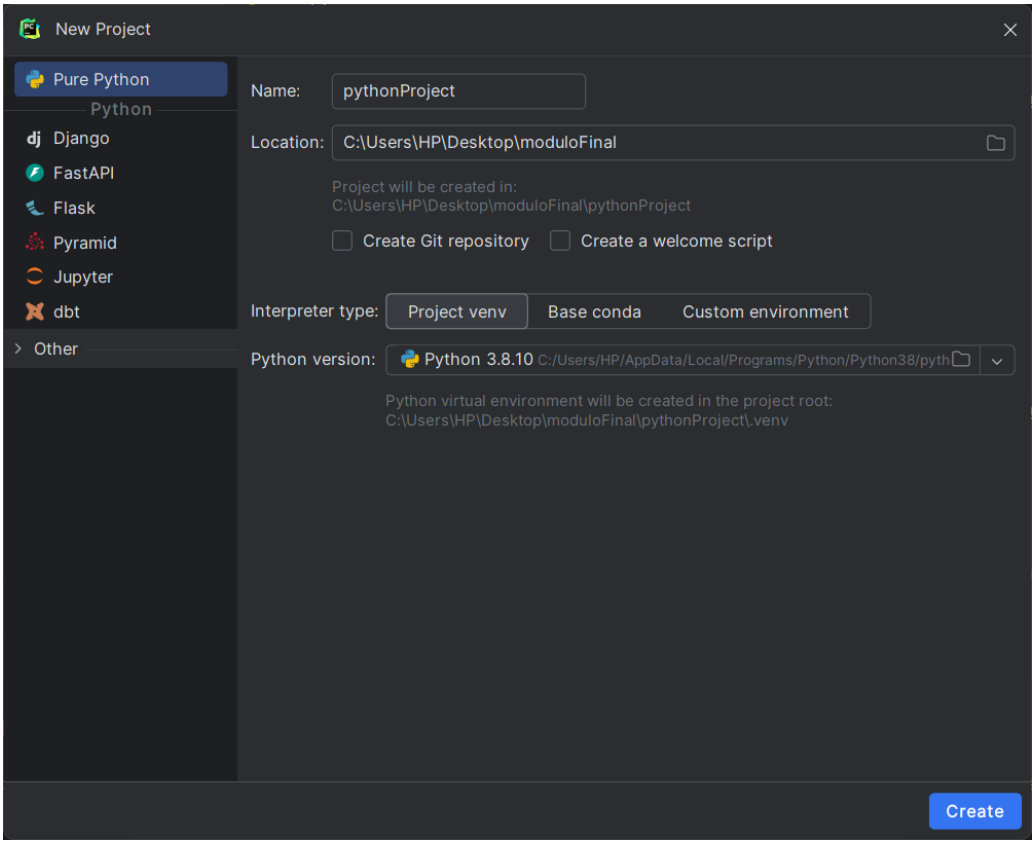

Ilustración 3: Entorno de PyCharm.

3.5 Machine learning

El machine learning, surgido en los años 80, es una rama de la inteligencia artificial que permite a las máquinas aprender autónomamente mediante el análisis de datos y la identificación de patrones, lo que les permite realizar predicciones. Esta tecnología tiene un impacto global creciente, con un mercado valorado en 8.000 millones de dólares en 2021 y una proyección de 117.000 millones para 2027. Su uso es tan común que a menudo no se aprecia que se interactúa con él, como cuando vemos anuncios personalizados en Internet, resultado de un análisis de datos diseñado para influir en diversas decisiones (Gómez, 2022).

Su propósito es crear algoritmos que extraen información de los datos con fines como explicar, clasificar o predecir. Además, analizar la estructura de los datos, adaptarlos a modelos comprensibles y utilizables por ingenieros y expertos en aprendizaje automático en distintos campos laborales (Mancilla, 2020).

3.6 Roboflow

Roboflow proporciona todas las herramientas necesarias para construir y desplegar modelos de visión por ordenador. Permite la integración en cualquier fase del proceso mediante API y SDK, o utilizando su interfaz integral para automatizar todo el proceso, desde la adquisición de imágenes hasta la inferencia. Ya sea para etiquetar datos, entrenar modelos o desplegarlos, Roboflow ofrece los bloques de construcción necesarios para crear soluciones personalizadas de visión por ordenador (Ultralytics, 2023).

3.7 Redes Neuronales Convolucionales

Las redes neuronales convolucionales (CNN) son una arquitectura especializada de redes neuronales diseñada para procesar datos estructurados en cuadrículas, como imágenes. Son extremadamente eficientes para tareas de visión por computador, como el reconocimiento de objetos, gracias a su capacidad para identificar patrones espaciales jerárquicos en los datos de entrada. Las CNN reducen el número de variables de entrada utilizando propiedades espaciales de las imágenes, lo que permite manejar grandes cantidades de píxeles efectivamente. Desde su éxito en la competencia ImageNet en 2012, han evolucionado significativamente, convirtiéndose en una herramienta clave en el aprendizaje automático y la inteligencia artificial (Parada Torralba, 2022).

Se puede observar en la ilustración 4, las distintas capas que puede tener una red neuronal en las cuales aprende a detectar distintas características de una imagen, estas características pueden empezar desde lo más simple, como detectar brillo y bordes, hasta transformarse en algo más complejo como definir objetos de forma singular.

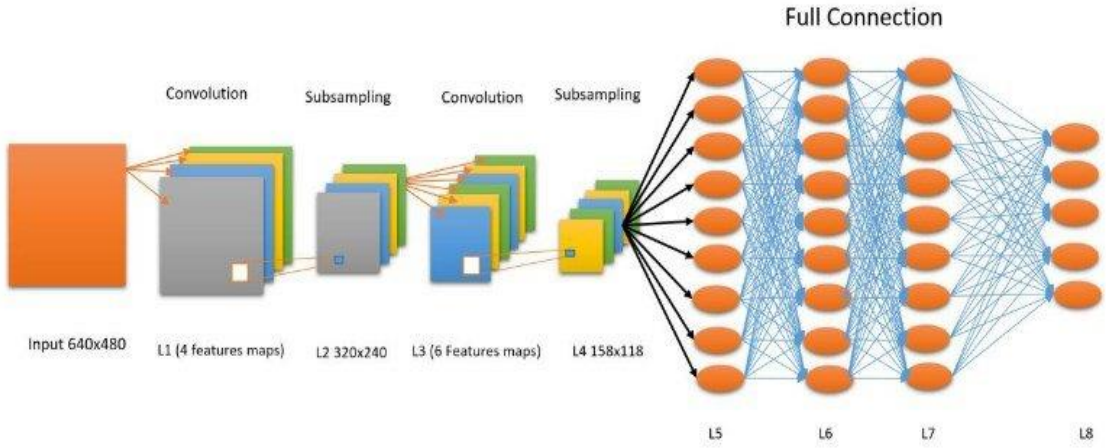

Ilustración 4: Red Neuronal Convolucional **(De la Rosa, 2016).**

3.8 Procesamiento de imágenes

El procesamiento digital de imágenes es una técnica fundamental que involucra la manipulación de imágenes digitales mediante algoritmos computacionales para mejorar su calidad o extraer información relevante. Este proceso incluye etapas como la adquisición de imágenes, donde se capturan mediante sensores, y el preprocesamiento para eliminar fallos y mejorar la imagen. La segmentación es clave, dividiendo la imagen en partes constituyentes para identificar áreas de interés. Las aplicaciones del procesamiento de imágenes abarcan desde la medicina y la biología hasta la astronomía y la arqueología, mejorando la interpretación y análisis de imágenes en diversos campos científicos y técnicos (La Serna Palomino & Román Concha, 2009).

En la ilustración 5, se muestra un ejemplo del esquema de lo que se necesita para que la máquina logre procesar una imagen, como es la descomposición de sus colores.

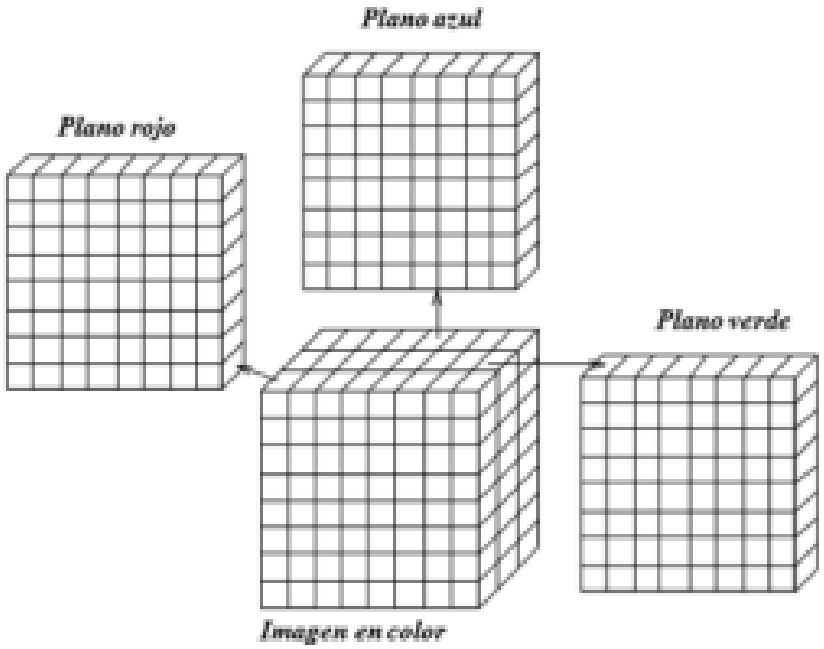

Ilustración 5**:** Esquema de la descomposición de colores en una imagen en cada plano **(Borella, 2022).**

3.9 Módulos didácticos Industriales

Un módulo didáctico industrial es una herramienta educativa esencial diseñada para facilitar el aprendizaje, manejo y desarrollo de proyectos de automatización de procesos mecánicos utilizando Controladores Lógicos Programables (PLCs). Este módulo proporciona a los estudiantes los componentes básicos necesarios para una comprensión y visualización más clara de los conceptos teóricos impartidos en el laboratorio. Además, permite la introducción y prueba de programas desarrollados en lenguajes de programación específicos para PLCs, asegurando el correcto funcionamiento a través de dispositivos tangibles conectados a entradas y salidas digitales y analógicas. Esta interacción práctica mejora significativamente la experiencia educativa, preparando a los estudiantes para enfrentar desafíos reales en entornos industriales (Polanco Herrera & Villaruel Cuadrado, 2009).

En la ilustración 6, se muestra los módulos didácticos que están situados en los laboratorios de automatización, los cuales son usados por los estudiantes de ingeniería Electrónica y Automatización para realizar prácticas constantes.

Ilustración 6: Módulo Didáctico Industrial del Laboratorio de Automatización de la UPS.

3.10 Matriz de confusión

La métrica de precisión solo se muestra cuando se activan las métricas avanzadas. La matriz de confusión se emplea para calcular otras métricas, como precisión y exhaustividad. En esta matriz, cada columna representa las instancias de una clase pronosticada, tal como lo ha determinado IBM Maximo Visual Inspection, mientras que cada fila representa las instancias de una clase real. Cada celda muestra la cantidad de veces que una imagen ha sido clasificada correcta o incorrectamente. La matriz de confusión puede visualizarse como una tabla de valores o un mapa de calor. Esta matriz es útil para identificar si el modelo está confundiendo clases o dejando de identificar ciertas categorías (Barrios Arce, 2019).

Se observa en la ilustración 7, la gráfica de una matriz de confusión en donde la diagonal principal es la que va a indicar el grado de precisión que tiene el modelo entrenado.

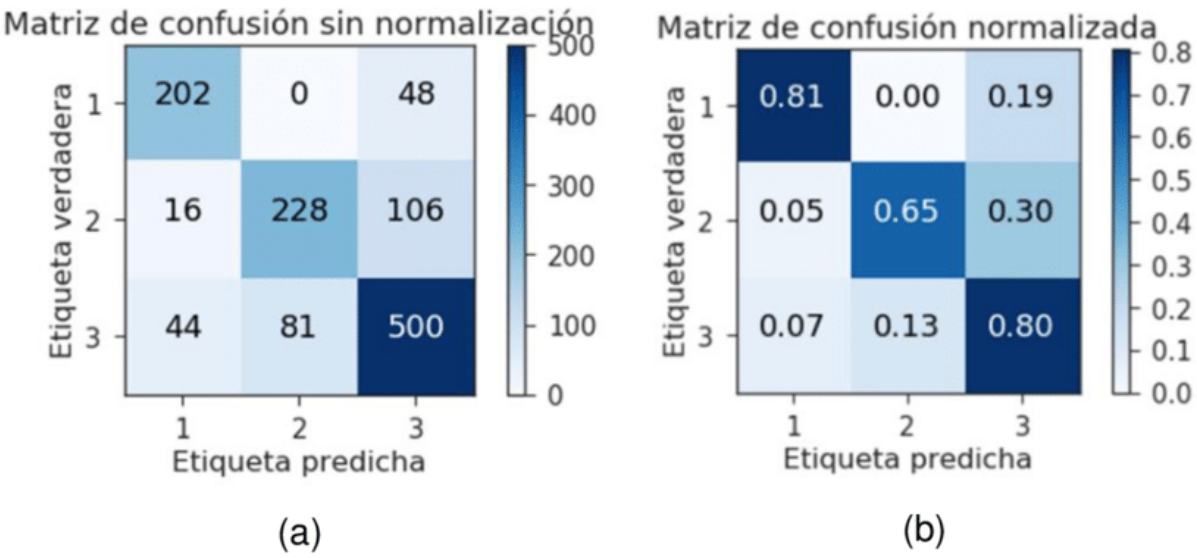

Ilustración **7**: Matriz de confusión sin normalizar y matriz de confusión normalizada **(García, 2023).**

3.11 Aprendizaje Supervisado

El aprendizaje supervisado es una técnica del aprendizaje automático donde el algoritmo aprende a partir de datos con pares de entradas y salidas, ejemplos etiquetados. Durante la fase de entrenamiento, el modelo se nutre de estos ejemplos para aprender a realizar una tarea específica. Los datos utilizados para entrenar el algoritmo consisten en variables de características, también conocidas como "features", que están etiquetadas con una variable objetivo o "target". Este enfoque permite al modelo adquirir la capacidad de ejecutar tareas basándose en ejemplos etiquetados proporcionados durante el entrenamiento (Larroso, 2023).

Se da a conocer en la ilustración 8, las ramas en las que está dividido el aprendizaje automático y sus usos aplicados a la sociedad.

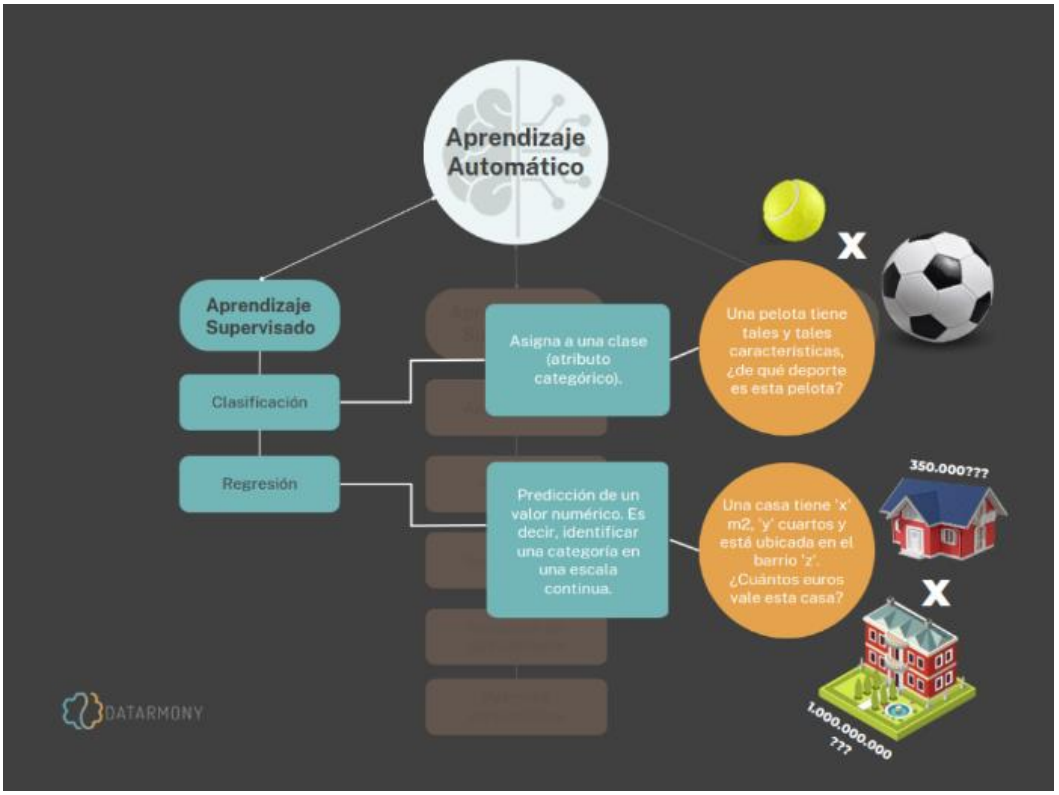

Ilustración 8: Técnicas del Aprendizaje Supervisado **(Larroso, 2023).**

IV MARCO METODOLÓGICO

En esta sección se describirá el proceso detallado para la puesta en marcha del sistema, comenzando por la preparación del software hasta el desarrollo y aplicación de técnicas de segmentación para identificar componentes.

En la ilustración 9, se visualiza el esquema que tendrá el sistema a desarrollar, indicando el paso a paso para la ejecución y el desarrollo del modelo de segmentación de componentes eléctricos con visión artificial:

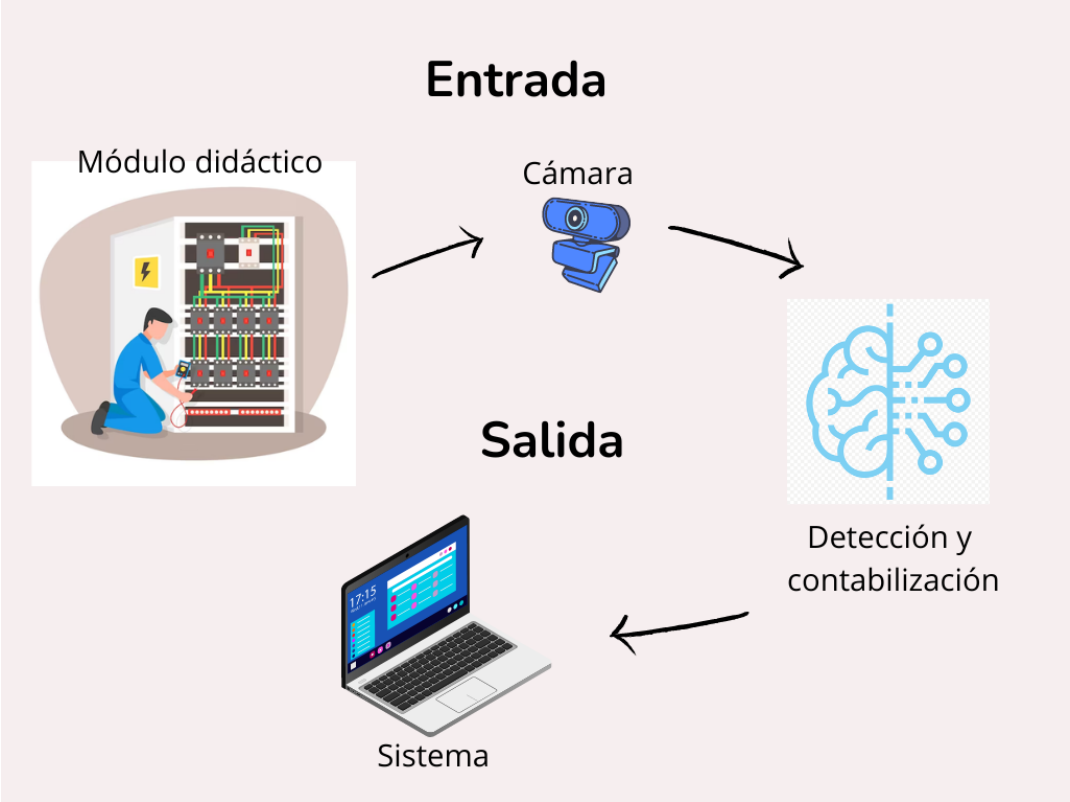

Ilustración 9: Diagrama de flujo del sistema de visión artificial.

La metodología que se empleó se divide en varias etapas clave: preparación y adquisición de datos, desarrollo del software y validación del sistema.

4.1 Preparación y Adquisición de Datos

4.1.1 Selección de la Cámara y Configuración del Entorno

Para este trabajo es fundamental saber elegir la cámara que se va a utilizar, teniendo en cuenta los factores de alta resolución y un ajuste automático de enfoque y exposición. La cámara debe colocarse en un ángulo donde tenga buena visibilidad de los componentes eléctricos en los módulos didácticos.

En cuanto al entorno de captura se debe dar prioridad a una óptima iluminación para evitar las sombras y reflejos que podrían afectar la calidad de la imagen. Es recomendable usar iluminación extra para mejorar la claridad del entorno donde se trabajará.

En la ilustración 10, se observa la cámara que se utilizará para la realizar la detección en tiempo real y que dará vida a la visión artificial para este proyecto.

Ilustración 10**:** Cámara Web Argomtech Cam50 **(Argom tech, s.f.)**.

4.1.2 Creación del Dataset

En este paso, se procede a realizar la búsqueda de imágenes que contengan los componentes que se encuentran integrados en los módulos didácticos, se efectúa la búsqueda utilizando distintos medios (sitios web, manuales técnicos, capturas de los módulos didácticos que están en los laboratorios). Esta carpeta donde se almacenan todas las imágenes encontradas se la

denomina “Dataset”. Este conjunto de imágenes permitirá entrenar el modelo, para esto se creó un Dataset diversificado para ayudar al modelo a captar distintas perspectivas de los objetos.

En la ilustración 11, se muestran las búsquedas en sitios web de cada componente para la creación del Dataset.

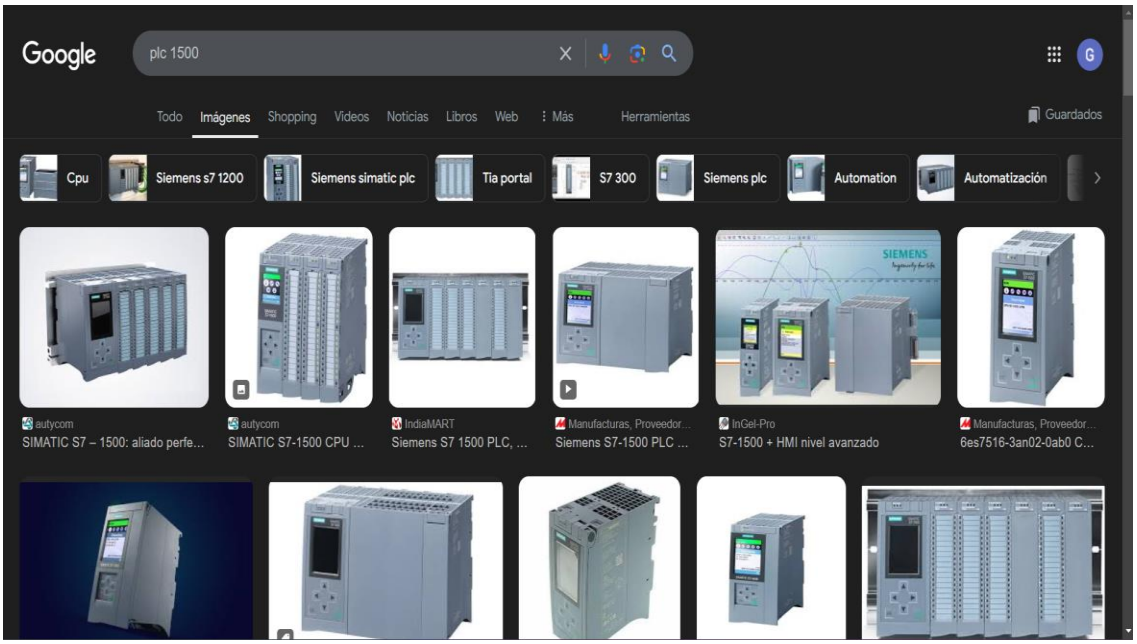

Ilustración 11**:** Búsqueda de imágenes de los componentes en sitios web.

También, se observa la ilustración 12 donde consta la lista de las 30 clases de objetos que se crearon, correspondientes a cada componente que se va a identificar. Cada carpeta consta de 50 imágenes con distintos ángulos que ayudarán y le darán mayor consistencia al modelo.

Clases	Cantidad De Imágenes Recopiladas
SIMATIC S7-1500 Siemens	50
SIMATIC S7-1200 Siemens	50
E/D 32DI Siemens	50
CM 1241 RS-232 Siemens	50
CM 1243-5 Siemens	50
CSM1277 SIMATIC NET Siemens	50
Fuente A. MeanWell	50
Fuente A. PM 1507 Siemens	50
Borneras electricas	50
Selector	50
Parada de emergencia	50
Variador de frecuencia Siemens	50
Router tplink	50
Luz piloto roja	50
Luz piloto verde	50
SCALANCE XB005 Siemens	50
Modulo PM 1207 Siemens	50
Siemens LOGO Power	50
Bornes bananas	50
Potenciometro	50
Pulsador N.A. verde	50
HMI KTP700 Siemens	50
Relay Camsco	50
Relay Schneider	50
Relay Siemens	50
Voltimetro Analogico	50
Voltimetro digital	50
Disyuntor 1P	50
Disyuntor 2P	50
Base fusible	50

Ilustración 12: Dataset de imágenes de los componentes eléctricos.

4.1.3 Etiquetado de Imágenes

Para esta parte del proceso, se realizó un análisis entre distintas aplicaciones que permitían crear el etiquetado de imágenes, entre las opciones están: **Roboflow**, **Amazon AWS Rekognition** y **Google Cloud Vision**. La intención de esta evaluación fue identificar cuál de estas plataformas ofrece las mejores características y funcionalidades para cumplir con los requisitos específicos del proyecto, que incluyen la facilidad de uso, las capacidades de etiquetado y la flexibilidad en el entrenamiento de modelos. Este análisis comparativo que se muestra en la ilustración 13 proporciona una visión clara de las ventajas y desventajas de cada plataforma, haciendo más fácil la elección según las necesidades del proyecto.

Roboflow se destacó por su interfaz gráfica intuitiva, herramientas avanzadas de preprocesamiento y etiquetado personalizado, así como opciones de prueba gratuita que permiten una evaluación inicial sin costo.

En la ilustración 13, se puede observar más detalladamente características de las distintas aplicaciones que se intentaron probar e información relevante del porque se eligió Roboflow, para la realización del etiquetado de imágenes.

Características	Roboflow	Amazon AWS Rekognition	Google Cloud Vision
Tipo de Servicio	Plataforma de etiquetado y preprocesamiento de imágenes, con capacidades de entrenamiento de modelos personalizados.	Servicio de reconocimiento y análisis de imágenes mediante API, con capacidades de etiquetado automático y análisis de contenido.	Servicio de análisis de imágenes y etiquetado automático a través de API, con capacidades de reconocimiento y etiquetado.
Modelos Preentrenados	Ofrece modelos preentrenados para tareas comunes y permite entrenamiento personalizado con tus datos.	Proporciona modelos preentrenados para reconocimiento general, sin opciones para entrenamiento personalizado.	Ofrece modelos preentrenados para una variedad de tareas de análisis y etiquetado, sin opciones para entrenamiento personalizado.
Interfaz de Usuario	Interfaz gráfica intuitiva para etiquetado de imágenes, preprocesamiento y exportación de datos.	Interfaz basada en API y consola de AWS, menos intuitiva para usuarios sin experiencia técnica.	Interfaz gráfica y API, con integración sencilla en el ecosistema de Google Cloud, pero menos enfocada en el etiquetado manual.
Automatización del Etiquetado	Proporciona herramientas de automatización y augmentación de datos, junto con la opción de etiquetado manual.	Ofrece etiquetado automático mediante API con configuraciones para detección de objetos y análisis de imágenes.	Proporciona etiquetado automático con herramientas integradas para la detección y clasificación de objetos.
Exportación de Datos	Exportación flexible en varios formatos (JSON, CSV, TFRecord), fácil integración con otros sistemas.	Exportación limitada a formatos estándar, integración más centrada en la API y menos en formatos personalizados.	Exportación en formatos estándar con integración directa en otros servicios de Google Cloud.
Capacidades de Preprocesamiento	Incluye herramientas de preprocesamiento y augmentación de imágenes (rotación, escalado, recorte, etc.).	Capacidades básicas de preprocesamiento mediante configuración de API, sin herramientas integradas para augmentación.	Capacidades limitadas de preprocesamiento, con enfoque en análisis y etiquetado post-procesamiento.
Escalabilidad	Escalabilidad basada en el uso de recursos en la plataforma con opciones de pago por uso y suscripción.	Alta escalabilidad con infraestructura de AWS, adecuado para grandes volúmenes de datos y alto rendimiento.	Alta escalabilidad con infraestructura de Google Cloud, adecuado para análisis a gran escala y alto rendimiento.
Soporte de Documentación	Documentación completa y recursos de soporte para integración y uso, con tutoriales y ejemplos prácticos.	Amplia documentación y soporte técnico, con recursos de AWS para integración avanzada y solución de problemas.	Documentación extensa y soporte técnico, con recursos de Google Cloud para integración y uso.
Costo	Costos basados en suscripción y uso, con opciones de prueba gratuita para inicialización.	Pago por uso, con una capa gratuita limitada y precios basados en el volumen de imágenes procesadas.	Pago por uso, con una capa gratuita y precios basados en el número de imágenes y características utilizadas.

Ilustración 13**:** Comparación de aplicaciones para etiquetado de imágenes**.**

4.1.3.1 Uso de la plataforma Roboflow

Una vez creado el Dataset se procede a subir a la plataforma de Roboflow en la cual se realiza el etiquetado de forma manual para identificar cada componente en el contexto de su ubicación dentro de la imagen. Esta es una de las etapas más cruciales dentro del proyecto para así poder entrenar el modelo de visión artificial.

En primer lugar, se abrió la plataforma de Roboflow donde se trabajará en la creación del modelo, para este caso se escogió la opción que más se relaciona a este proyecto, que es “Object Detection” tal como se muestra en la ilustración 14.

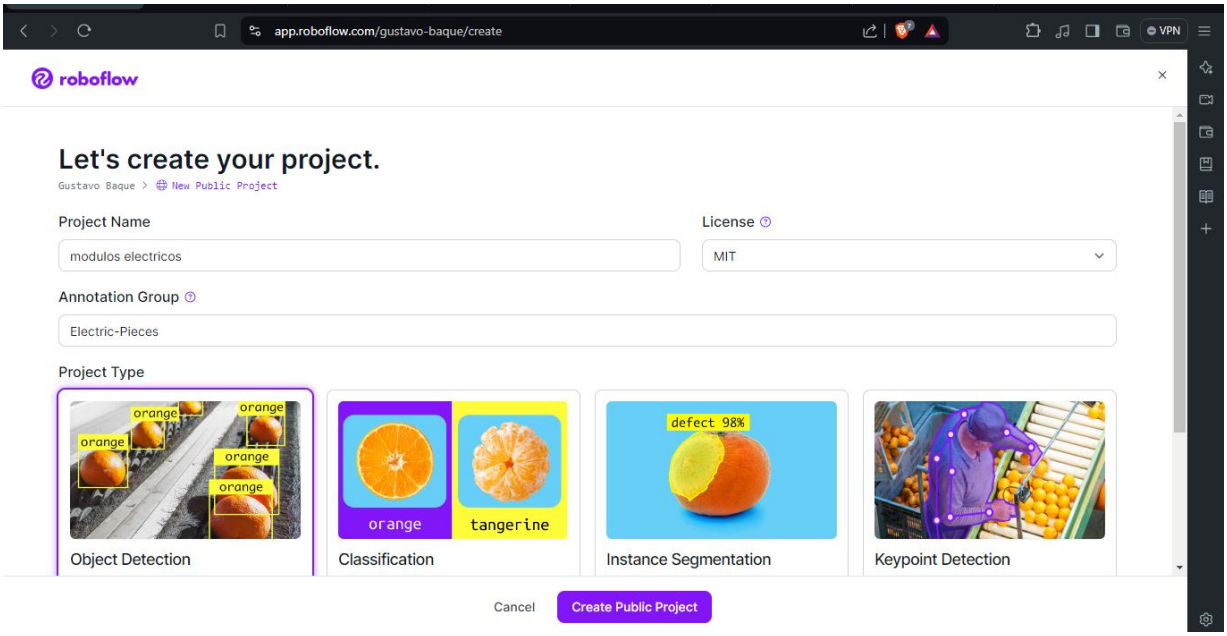

Ilustración 14: Plataforma Roboflow, ventana donde se escogerá el tipo de proyecto a realizar.

A continuación, se procedió a crear las clases que se van a utilizar para la identificación de los componentes, cada clase corresponde a un componente eléctrico de los módulos didácticos. En este caso, se crearon 30 clases que se pueden visualizar en la ilustración 15.

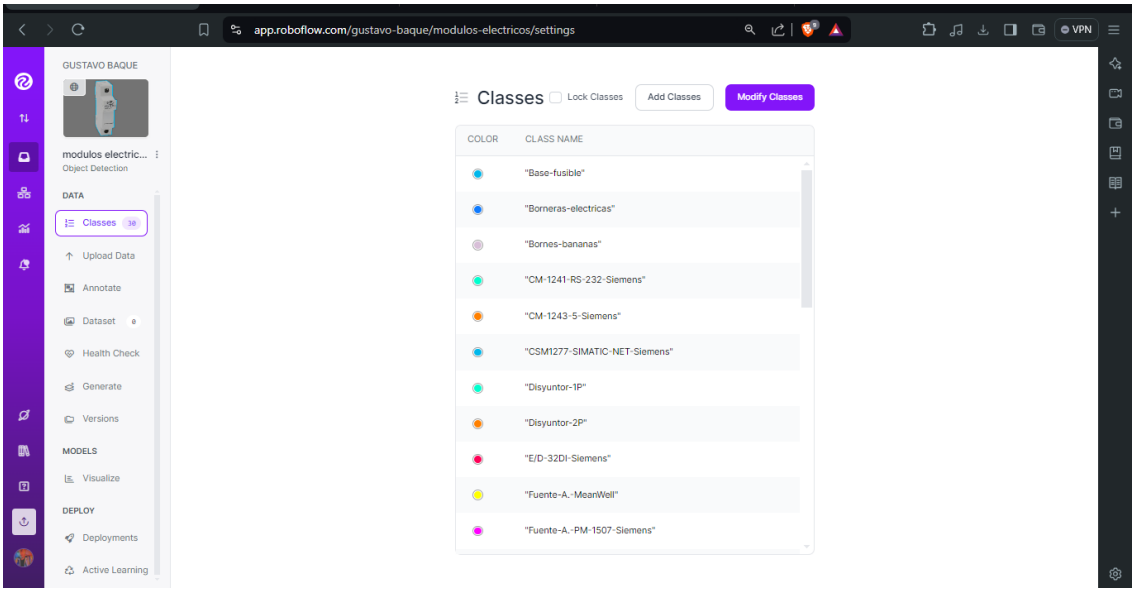

Ilustración 15: Ventana para la creación de las clases en Roboflow.

Luego, se realizó la carga completa del Dataset a la plataforma y se verifico que se hayan cargado las 1500 imágenes que se habían añadido a la carpeta, tal como se puede observar en la ilustración 16.

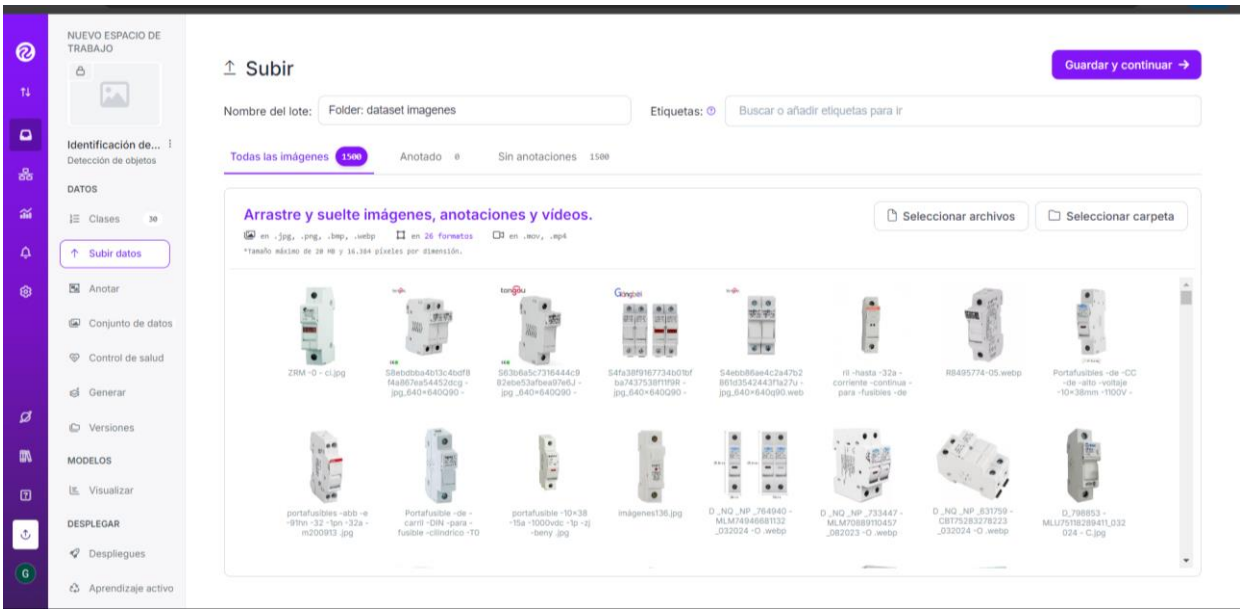

Ilustración 16: Ventana para realizar la carga de imágenes, ya sea de forma individual o en grupo a través de una carpeta.

Después de haber cargado el Dataset, en la ilustración 17 se observa cómo se procede a realizar el etiquetado de las imágenes, para este caso se usó el etiquetado en forma de cajas delimitadoras de cada componente.

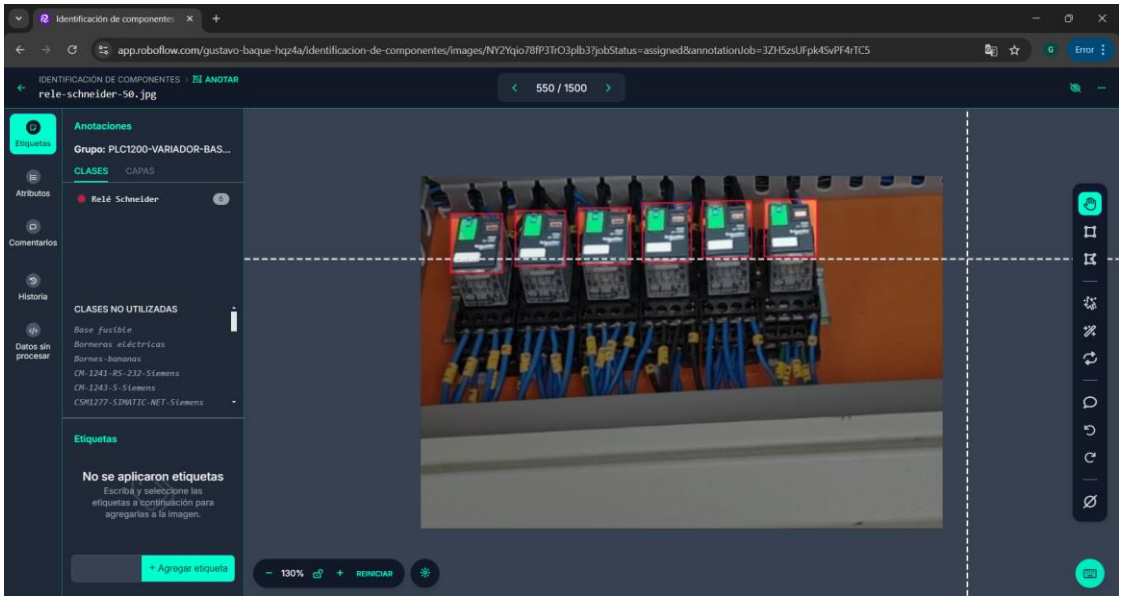

Ilustración 17: Ventana donde se realiza el etiquetado de imágenes.

En la ilustración 18, se muestra el resultado obtenido una vez culminada la parte de etiquetado de las imágenes.

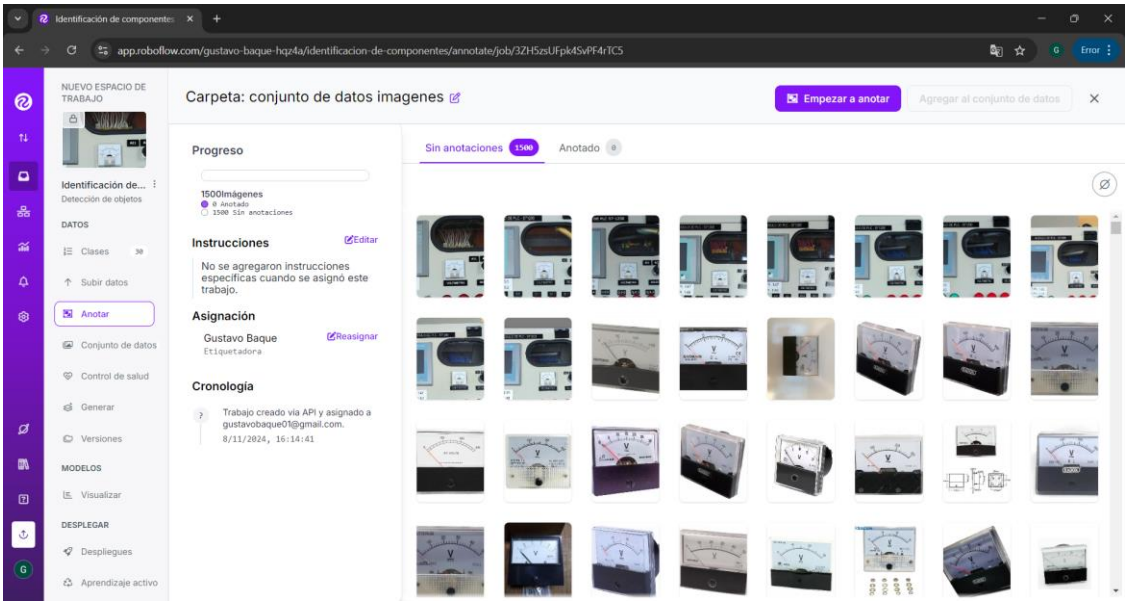

Ilustración 18: Ventana que muestra los resultados del etiquetado de imágenes.

4.1.4 Preparación del Conjunto de Datos

Una vez que se han capturado y etiquetado las imágenes, el conjunto de datos se divide en tres subconjuntos: entrenamiento, validación y prueba. La división realizada comprende, el 70%

para entrenamiento, 20% para validación y 10% para prueba. Los subconjuntos de datos deben ser equilibrados para asegurar una representación adecuada de todas las categorías de componentes.

Se observa, en la ilustración 19, la división realizada de los subconjuntos de datos dando un total de 4200 imágenes usadas para entrenamiento, 300 para validación y 150 para prueba, las cuales corresponden a los porcentajes requeridos por el modelo de visión artificial.

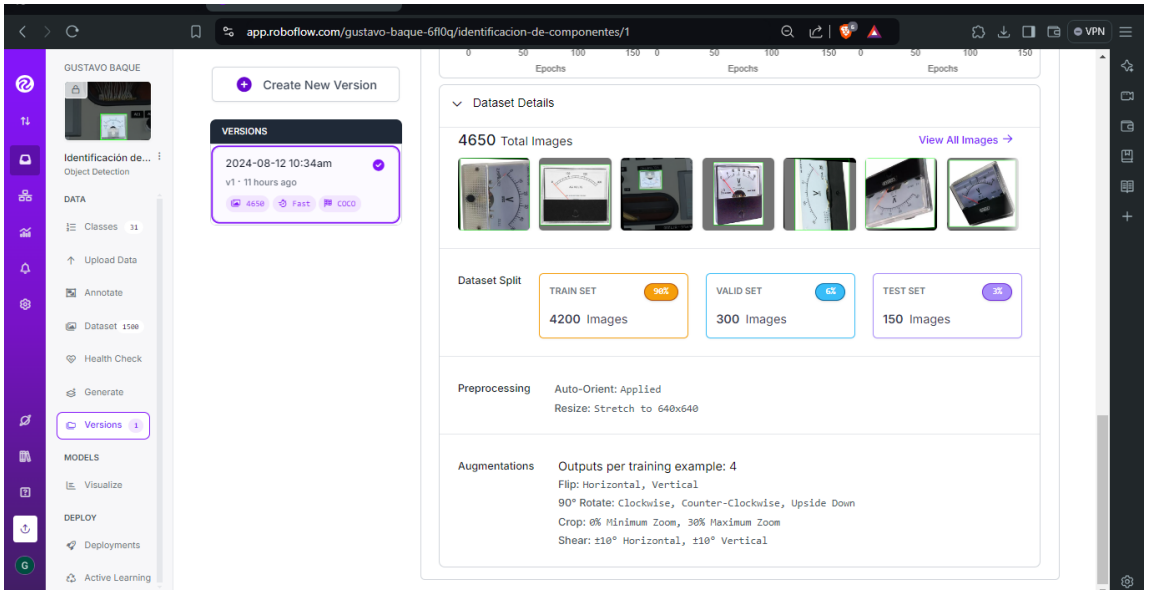

Ilustración 19: Ventana de división de subconjuntos para entrenamiento, validación y prueba.

Además, se realizó un preprocesamiento de las imágenes para ajustar el tamaño, normalizar los colores y aplicar técnicas de aumento de datos como rotación, escalado y recorte para mejorar la robustez del modelo.

Para la data aumentada se utilizaron aplicaciones como:

Leonardo AI: Esta aplicación permitió generar imágenes por medio de inteligencia artificial. Es una buena alternativa por su facilidad de uso y por las distintas herramientas que proporciona.

En la ilustración 20, se observa la plataforma Leonardo AI en la herramienta “image to image”, la cual permite generar nuevas imágenes a partir de imágenes de referencia. Se cargaron 20 imágenes del Módulo CM 1241 RS-232 Siemens y adicionalmente se agregó el prompt el cual es un texto donde se especifica las instrucciones del diseño esperado.

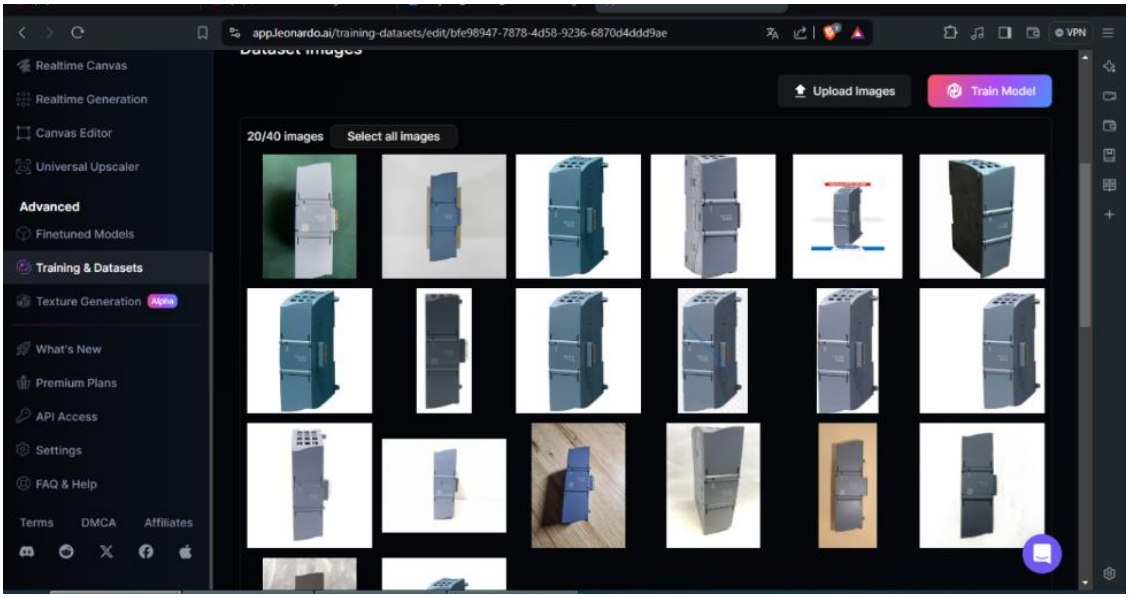

Ilustración 20: Plataforma Leonardo AI**,** opción image to image.

A continuación, se muestra en la ilustración 21 los resultados que se obtuvieron al generar las imágenes, se visualizan dispositivos que no tienen similitud con respecto al módulo antes mencionado ya que las letras estaban distorsionadas y el diseño del dispositivo no era el mismo. Por lo tanto, podemos concluir que esta IA de imágenes no es óptima para aumentar data de objetos con un diseño específico.

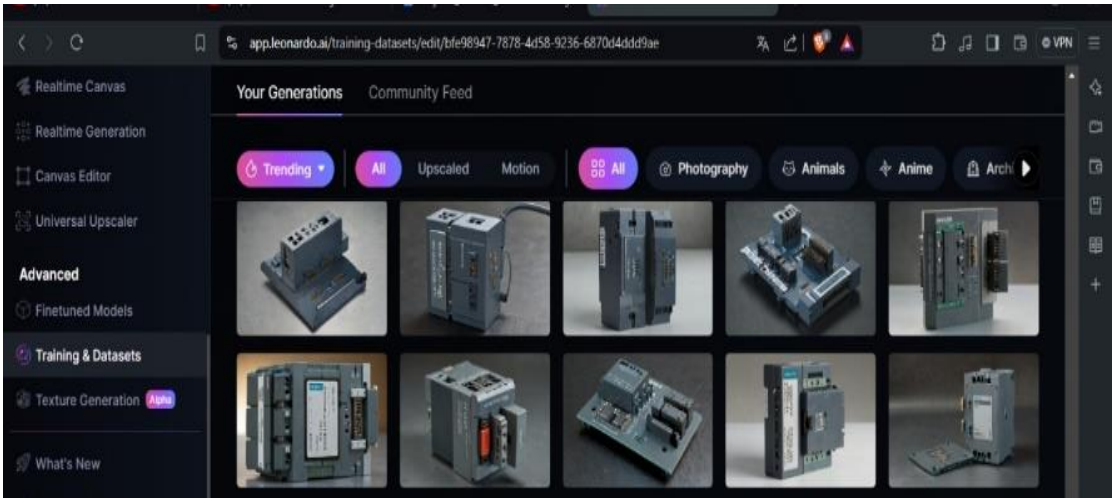

Ilustración 21: Resultados obtenidos en Leonardo AI para la generación de imágenes con IA.

Roboflow: Una vez generado el etiquetado de imágenes, Roboflow ofrece una herramienta que permite aumentar la data de imágenes modificando distintos parámetros entre ellos las

rotaciones, iluminación, zoom. Estas herramientas ayudan a darle una mejor precisión al modelo, ya que se obtiene un Dataset más robusto.

En la ilustración 22 y 23, se visualiza los diferentes parámetros que se utilizaron para la data aumentada de imágenes, en este caso se incrementó el Dataset, dando como resultado un total de 4650 imágenes para el entrenamiento del modelo, ya que esta herramienta permite aumentar las imágenes para el entrenamiento del modelo.

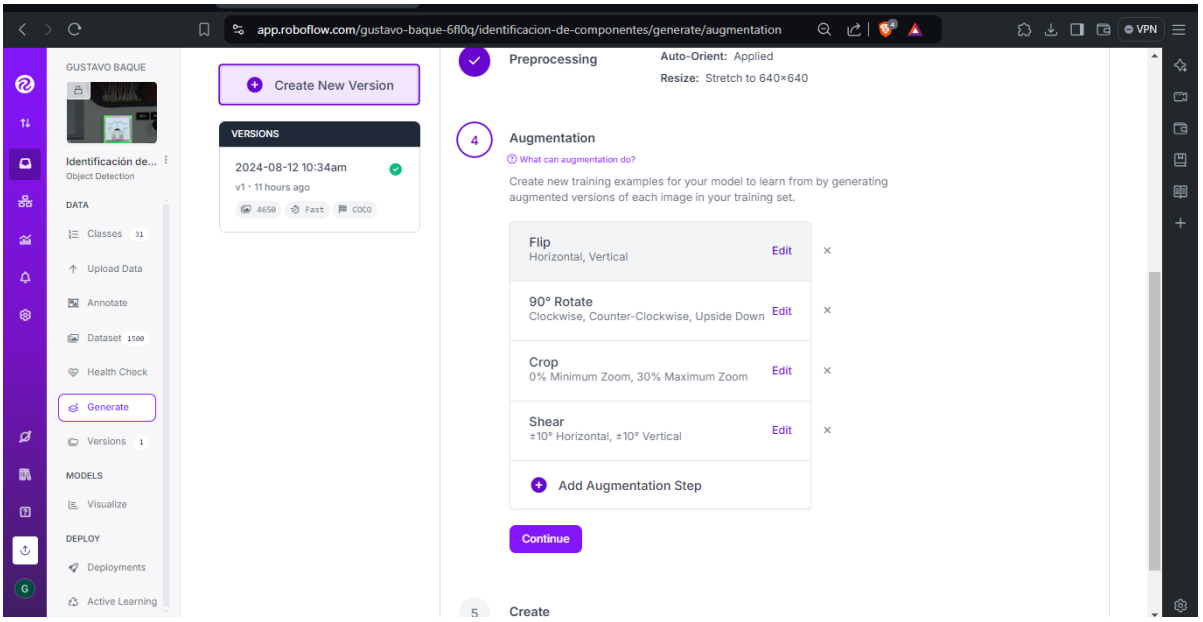

Ilustración 22: Ventana de data aumentada.

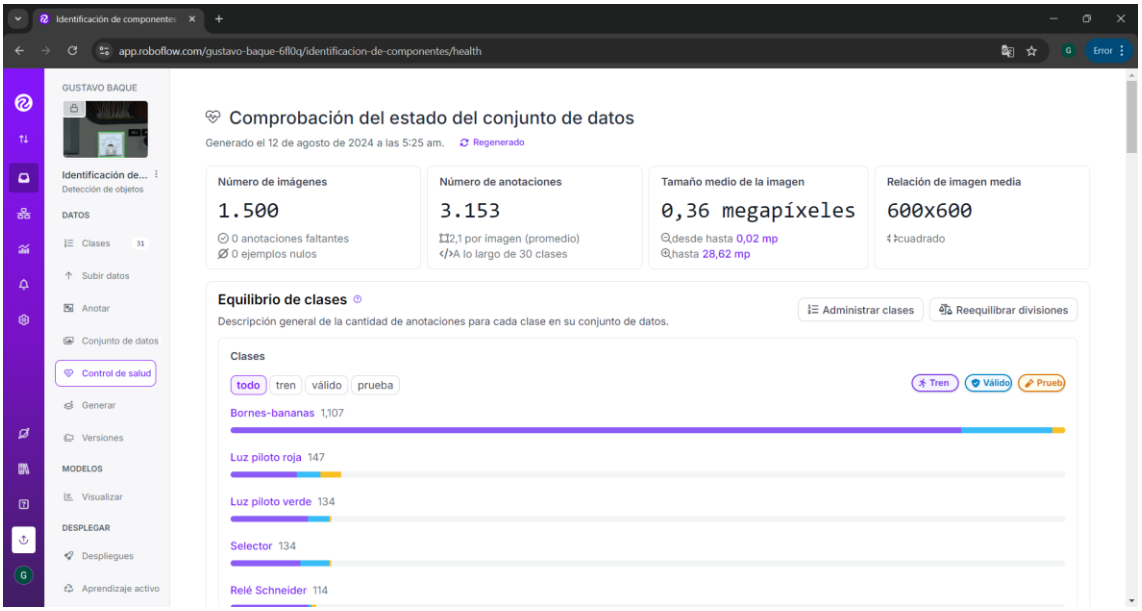

Ilustración 23: Ventana del resultado que se obtuvo al realizar la data aumentada de imágenes.

Luego de probar las distintas herramientas anteriormente mencionadas, la mejor opción para realizar la data aumentada es Roboflow ya que tiene una aplicación integrada que permite realizar “Data Augmentation” según los requerimientos que le pidan, sin distorsionar las imágenes, dando buenos resultados y consistencia al modelo.

4.2 Desarrollo del Software

4.2.1 Selección y Configuración del Entorno de Desarrollo

En este caso, se usó un entorno de desarrollo IDE (Entorno de Desarrollo Integrado) muy popular como es PyCharm debido a su amplia adopción en la comunidad de inteligencia artificial y por qué posee bibliotecas robustas para procesamiento de imágenes y aprendizaje automático, como es el caso de OpenCV, TensorFlow, Keras, entre otras. La configuración del entorno virtual se da por la gestión de las dependencias y para asegurar la consistencia en la ejecución del código.

Se muestra en la ilustración 24, el entorno de desarrollo que se va a utilizar para el entrenamiento del modelo y creación de las interfaces gráficas para la interacción con el usuario.

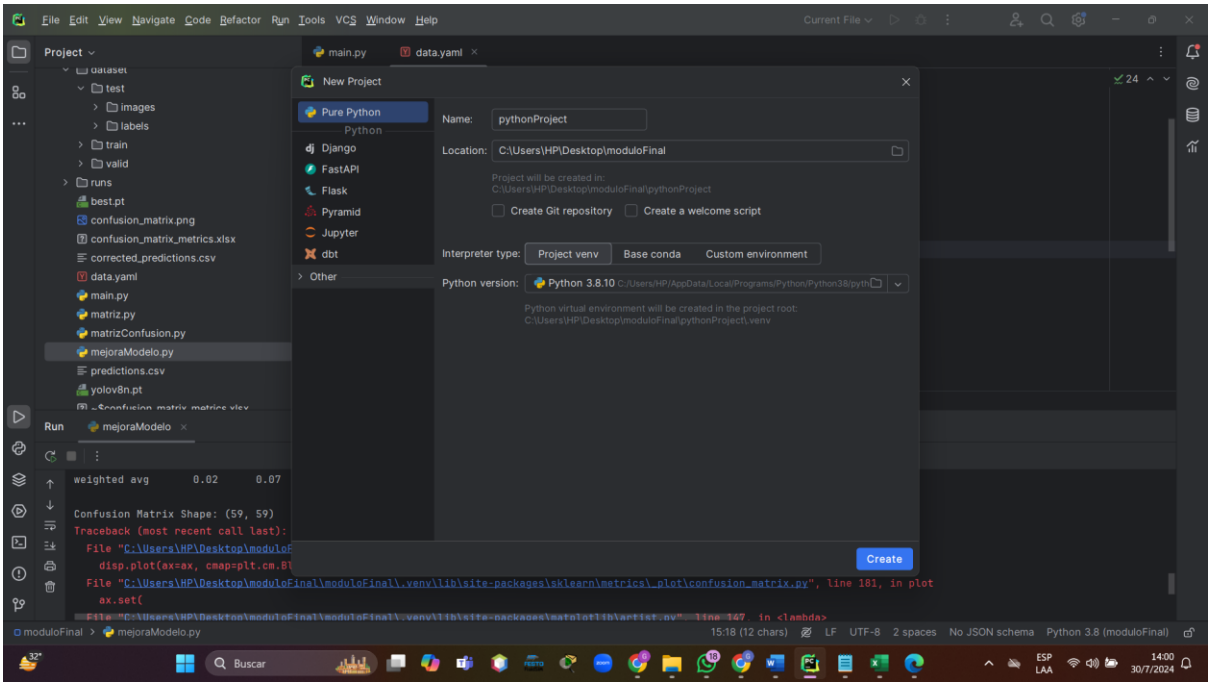

Ilustración 24: Entorno de Desarrollo IDE PyCharm**.**

4.2.2 Implementación del Modelo de Identificación

En este caso, se utilizó una biblioteca de código abierto “Ultralytics” para el entrenamiento, evaluación e implementación del modelo de identificación de componentes, ya que esta librería ofrece una interfaz de fácil manejo para modelos de YOLO, incluyendo YOLOv8 que es la versión mejorada en términos de precisión y eficiencia.

4.2.2.1 Instalación de la biblioteca Ultralytics

Se procedió a instalar la librería Ultralytics, para esto se introduce la línea de código “pip install Ultralytics” en la terminal. para que así se instalen las dependencias necesarias para su funcionamiento.

En la tabla 1, se muestran de forma detallada las dependencias que se instalan al instalar la librería Ultralytics.

Tabla 1:

Dependencias instaladas a través de la librería Ultralytics.

Dependencia	*Versión*	*Función*
'torch'	2.0.0	Entrenamiento e inferencia de modelos.
'opencv-python'	4.8.0	Manipulación y procesamiento de imágenes.
'matplotlib'	3.7.0	Creación de gráficos y visualización de datos.
'numpy'	1.24.3	Operaciones matemáticas y manejo de arrays multidimensionales.
'Pillow'	9.5.0	Manipulación de imágenes en formato PIL.
'tqdm'	4.65.0	Visualización de barras de progreso durante procesos largos.
'requests'	2.31.0	Para hacer solicitudes HTTP, para descargar modelos y datos desde la web.

4.2.2.2 Entrenamiento del modelo

Entre las bibliotecas de visión por computador, se utilizó YOLO (You Only Look Once) versión 8, ya que es una de las arquitecturas de redes neuronales convolucionales más avanzadas y eficaces para la detección de objetos en tiempo real.

YOLO es capaz de realizar detección de objetos a una velocidad muy alta, sin bajar el índice de precisión. YOLOv8 es la última iteración de esta serie y se ha introducido mejoras importantes en términos de precisión, velocidad y eficiencia comparándolas con sus predecesores.

En la ilustración 25, se muestran los diferentes modelos pre-entrenados que YOLO tiene a disposición de los usuarios.

Model	size (pixels)	mAP^{val} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

Ilustración 25: Modelos de pre-entrenados de YOLO **(Github, s.f.)**.

Debido a que es un modelo más robusto en primera instancia se eligió YOLOv8m pero presento dificultad para generar un buen entrenamiento para este proyecto probablemente por la tarjeta gráfica de la laptop que se utilizó. Por lo tanto se eligió el modelo más ligero YOLOv8n, el cual funcionó de manera correcta. En la siguiente ilustración 26 se presentan las gráficas de precisión del modelo antes mencionado.

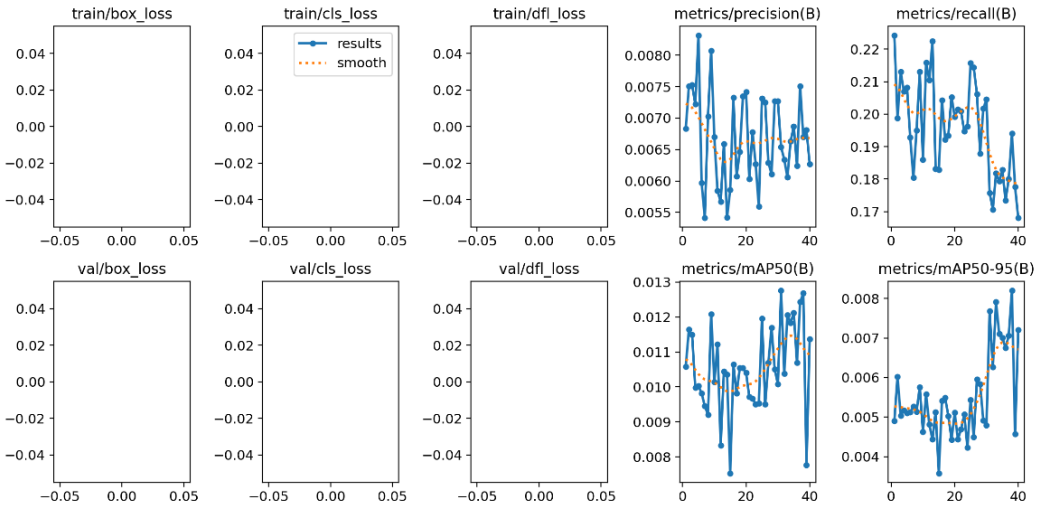

Ilustración 26: Modelo Defectuoso Entrenado con YOLOv8m.pt.

A continuación, en la ilustración 27 se puede observar la diferencia que tiene el modelo al ser entrenado desde la plataforma Roboflow donde se observa un rendimiento más óptimo y ajustado a los resultados esperados.

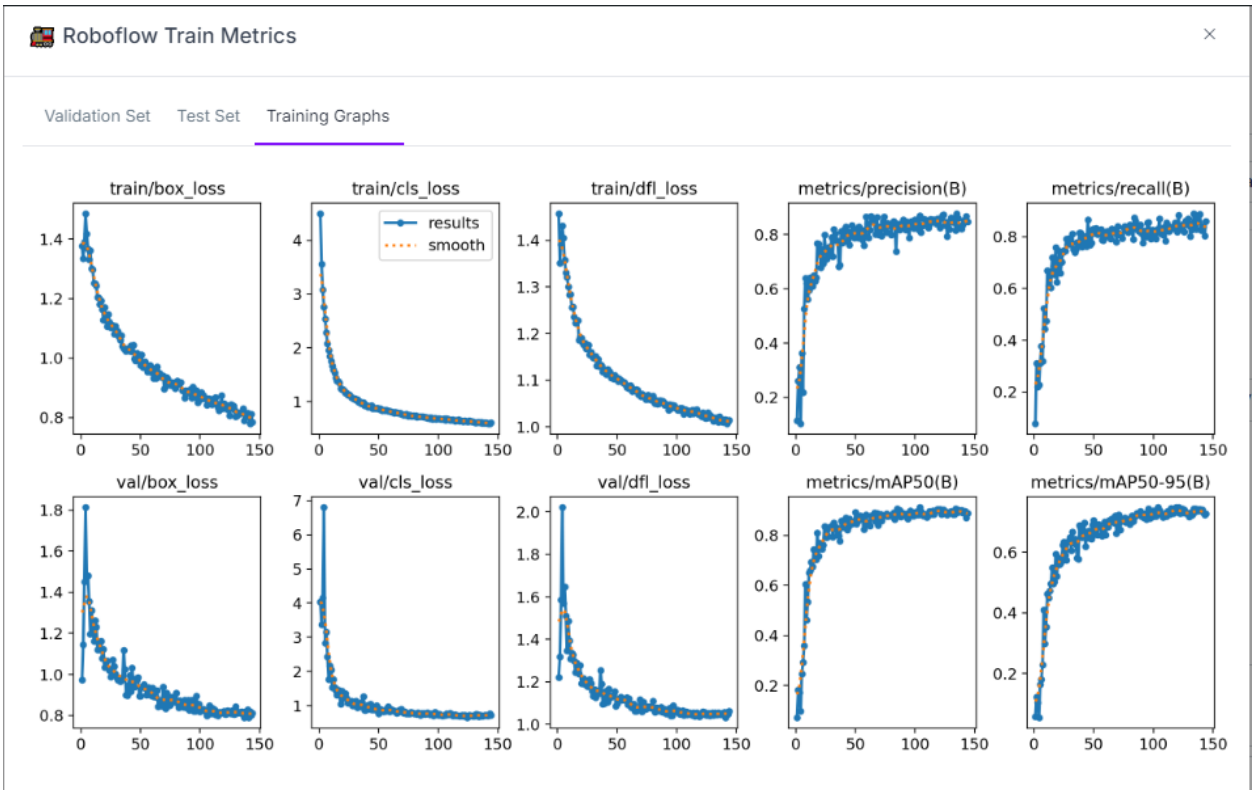

Ilustración 27: Modelo valido entrenado desde la plataforma Roboflow.

4.2.3 Integración del Software con la Cámara

El software desarrollado se integra con la cámara para permitir la captura en tiempo real y el procesamiento de imágenes. En este caso se usará la biblioteca OpenCV la que va a permitir la captura, manipulación de imágenes y YOLOv8 para la detección de objetos. Luego se conecta la cámara al pc para que el sistema la reconozca automáticamente.

La interfaz debe ser intuitiva y permitir ajustes en tiempo real, como cambiar la configuración de la cámara y visualizar los resultados de la clasificación.

En las ilustraciones 28 y 29, se muestra la interacción entre la cámara y la interfaz de usuario, permitiendo realizar la identificación de los componentes en tiempo real.

Ilustración 28: Interacción entre la interfaz de la app y la cámara.

Ilustración 29**:** Interacción entre la interfaz y la cámara.

4.2.3.1 Métricas de evaluación

4.2.3.1.1 Matriz de confusión

Esta información permitió conocer el rendimiento que tiene el modelo de clasificación. Aquí se muestra una forma estructurada de observar las predicciones del modelo frente a los valores reales, para permitir realizar un análisis minucioso de la precisión y los errores presentes en el modelo.

Se observa en la Ilustración 30, la matriz de confusión normal donde se muestran el conteo absoluto de las instancias que fueron clasificadas de forma correcta e incorrecta por el modelo. Aquí se puede visualizar tanto los errores como los aciertos en términos de conteos absolutos, permitiendo comprender directamente el desempeño que tiene el modelo.

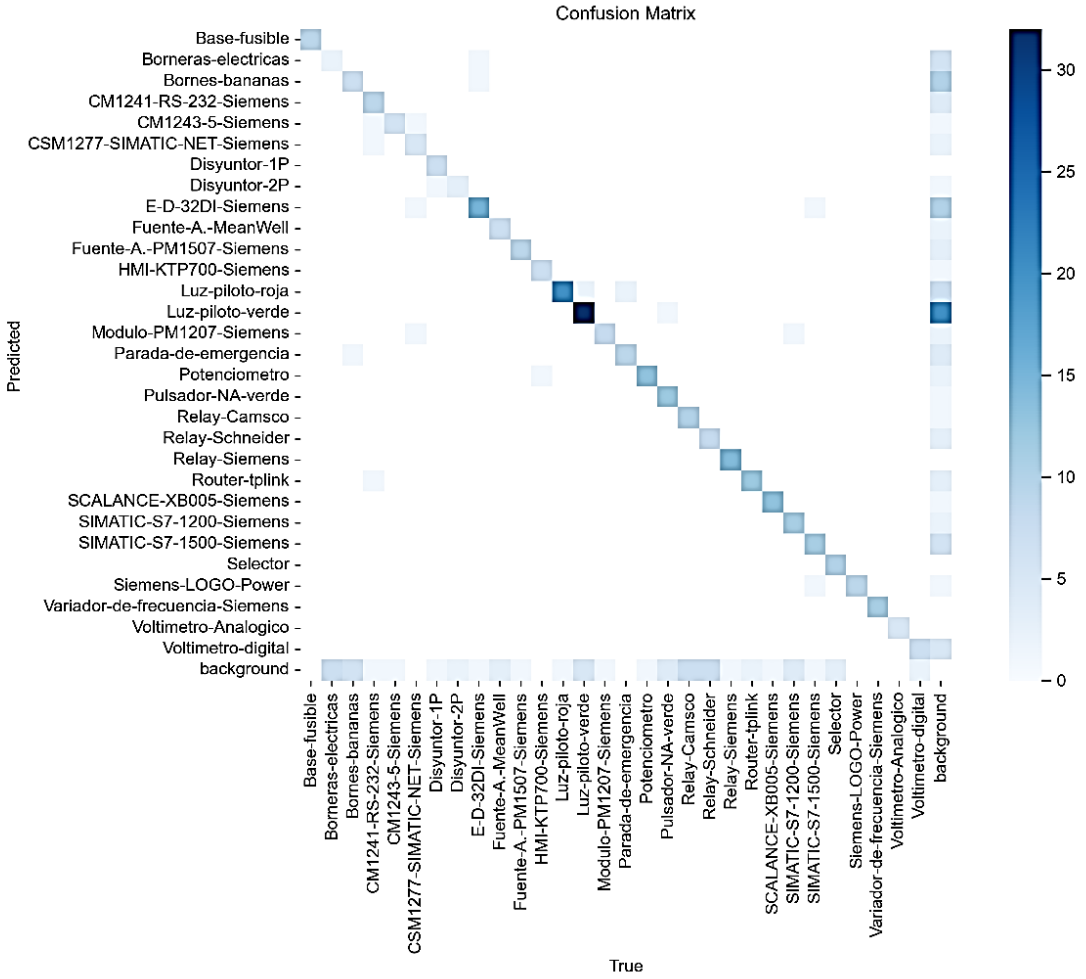

Ilustración 30: Matriz de Confusión.

Mientras que en la ilustración 31, se muestra la matriz de confusión normalizada, donde los conteos son convertidos en proporciones o porcentajes, dando a conocer cuáles son las fracciones de las instancias verdaderas de cada clase que fueron identificadas de forma correcta o incorrecta.

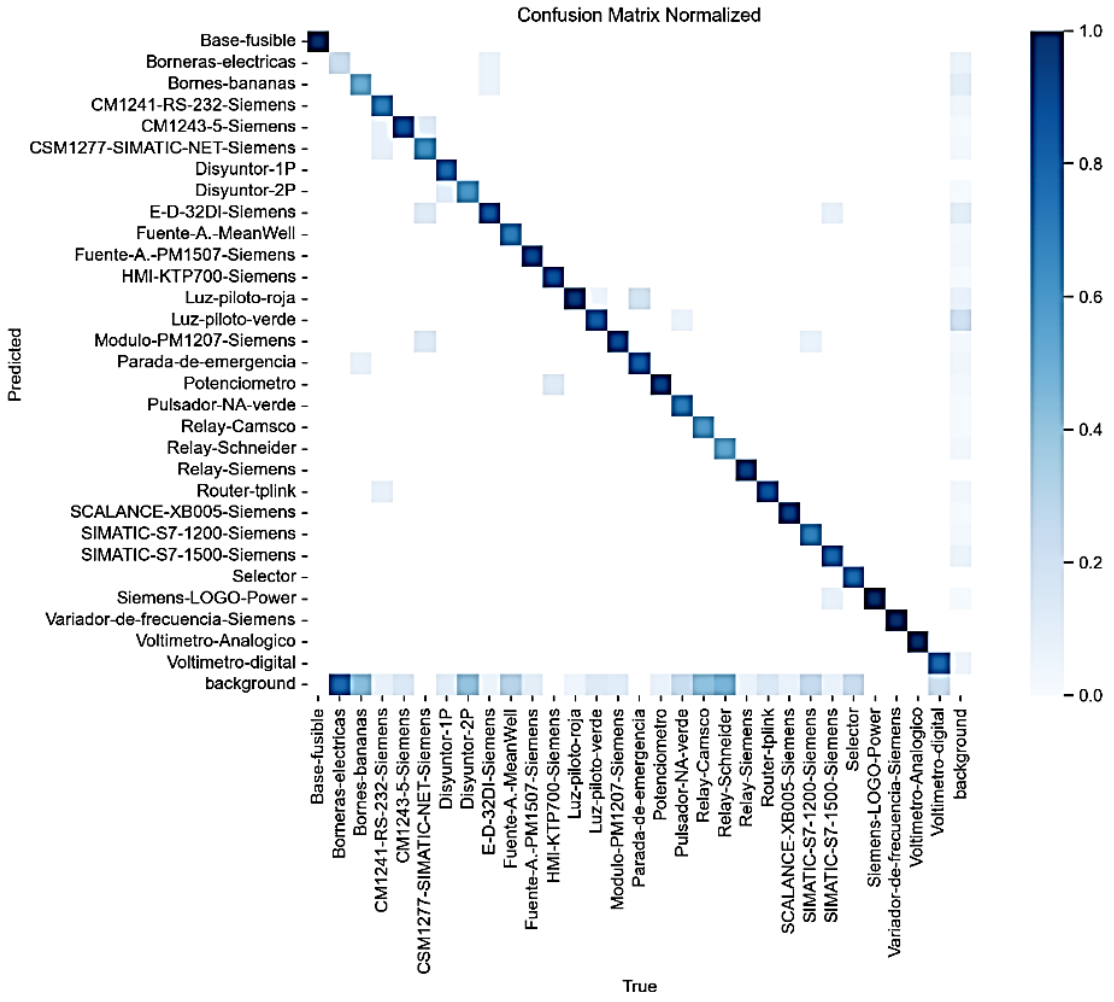

Ilustración 31: Matriz de Confusión Normalizada.

4.3 Creación de la Interfaz de interacción con el usuario

Se realizó una interfaz interactiva y fácil de manejar para el usuario, la cual se compone de varias pantallas:

4.3.1 Pantalla principal o de inicio

Aquí se muestra una ventana en la que el usuario va a poder visualizar las instrucciones para utilizar la aplicación de forma correcta y podrá iniciar la identificación como se observa en la ilustración 32.

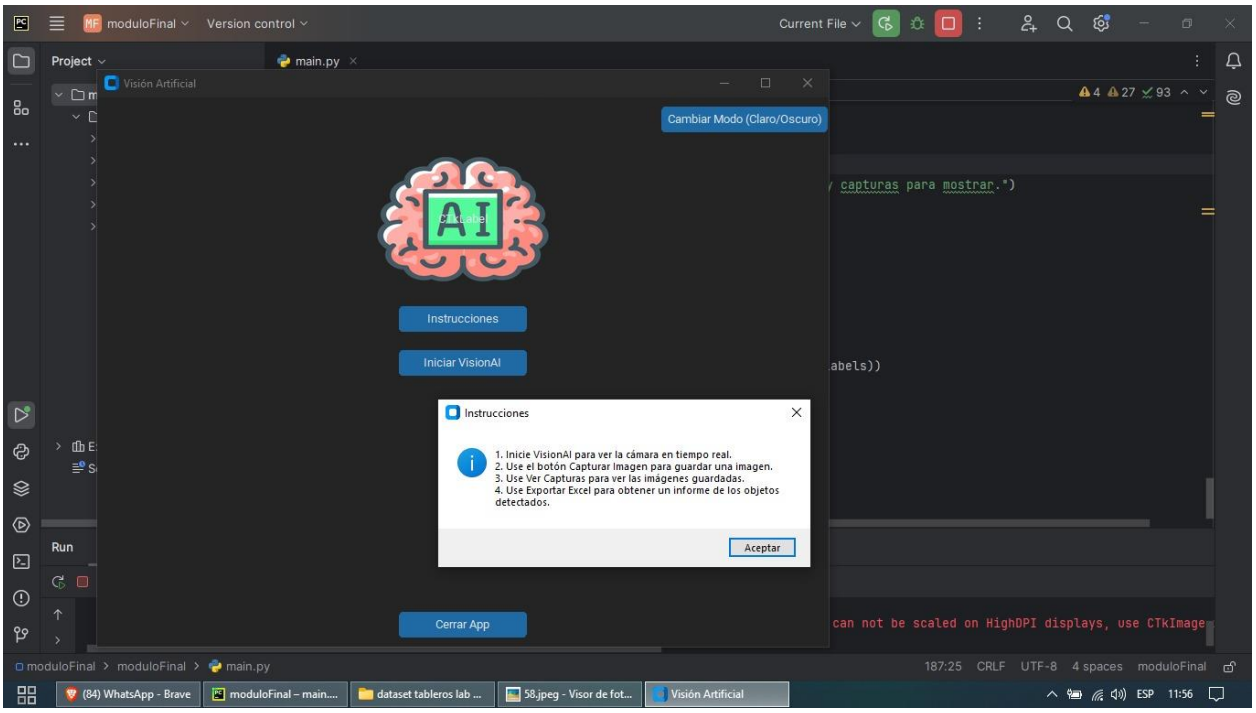

Ilustración 32**:** Interfaz Principal de la aplicación de visión artificial.

4.3.2 Pantalla de Identificación de componentes

En esta pantalla se da por iniciado la identificación de los componentes y se tendrá la opción de realizar capturas a lo que se visualiza con la cámara, también se podrá generar un archivo de Excel en el cual muestre un listado con los componentes que se están identificando y una cotización de los mismos, como se puede visualizar en la ilustración 33.

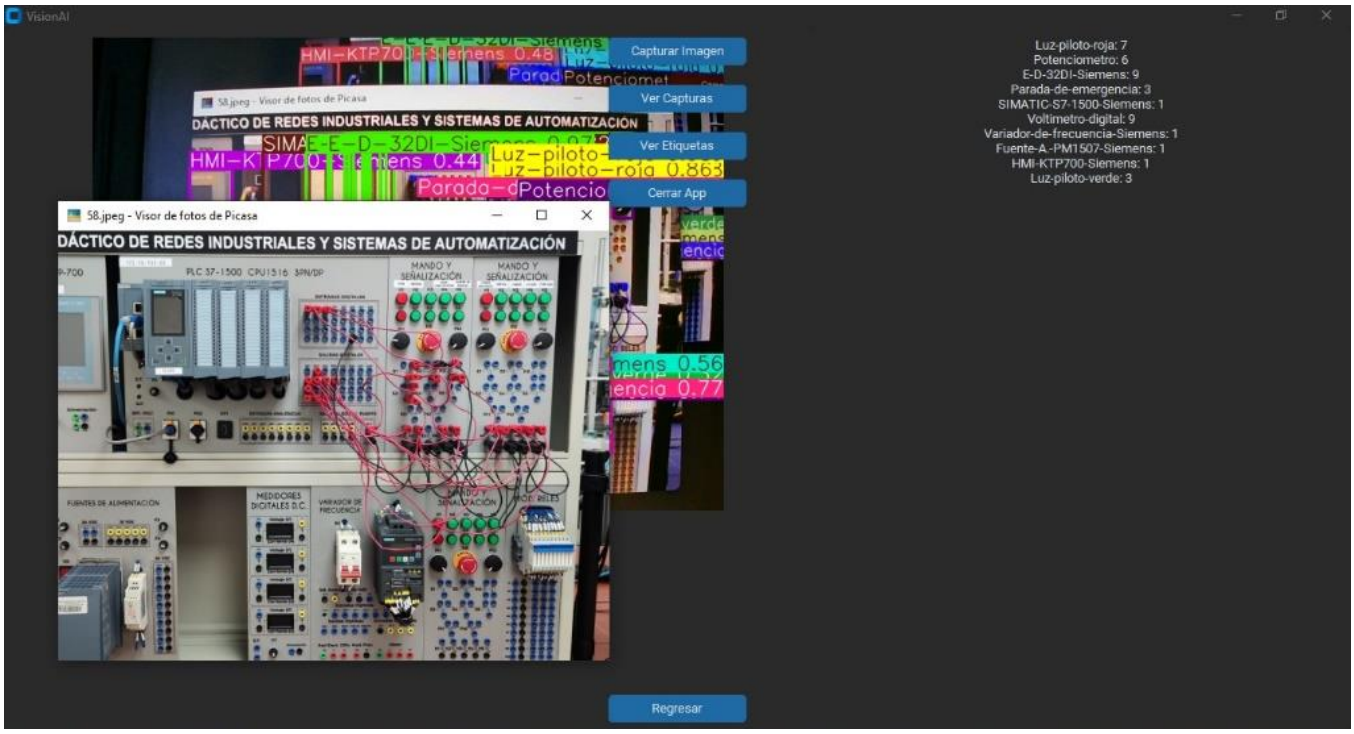

Ilustración 33: Pantalla de identificación de componentes del módulo didáctico.

4.3.3 Pantalla de base de datos

En la ilustración 34, se muestra en una ventana emergente la lista de todos los componentes o clases que fueron registrados para el entrenamiento del modelo.

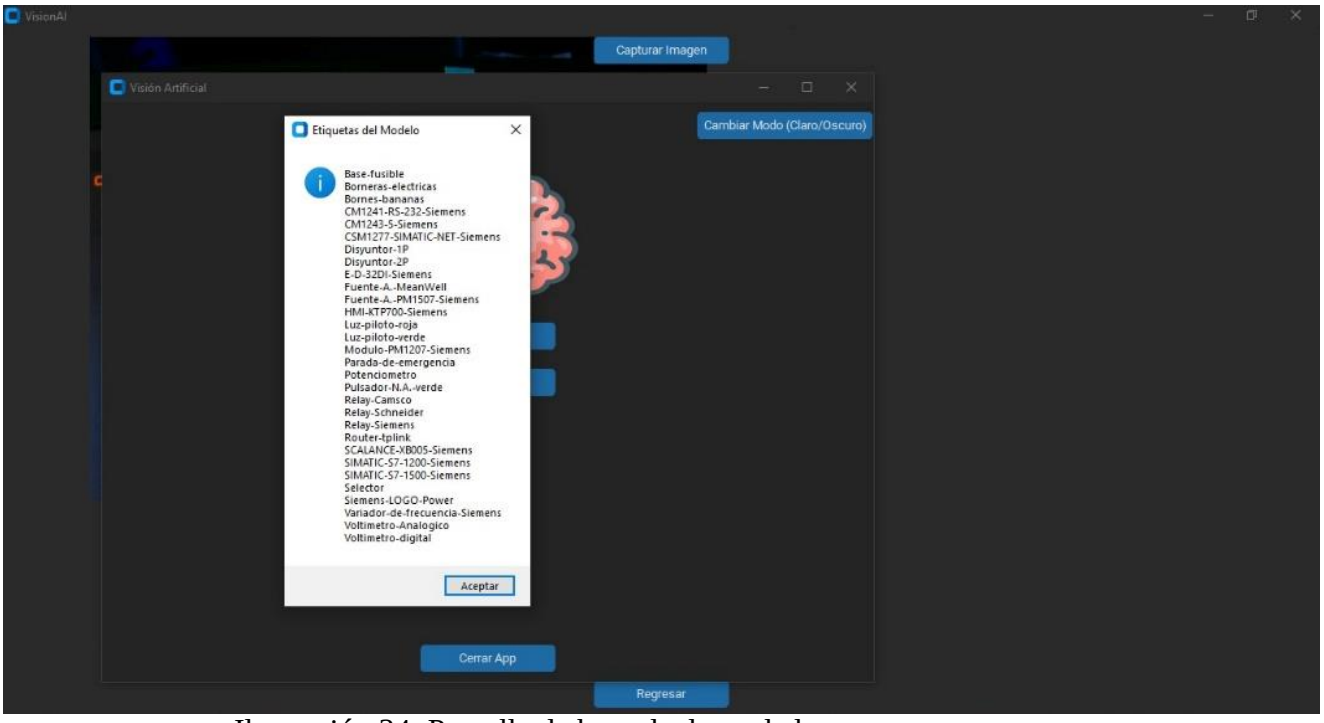

Ilustración 34: Pantalla de base de datos de los componentes**.**

V Análisis de Resultados

5.1 Crear un Conjunto de Datos Diversificado con los Objetos a Identificar

Se logró recopilar datos bastante amplios que incluyen imágenes de los 30 componentes eléctricos de los módulos didácticos. Se utilizaron una diversidad de fuentes de búsqueda para garantizar una variedad de imágenes, donde se toma en cuenta temas como utilizar ángulos de rotación diferentes, condiciones de iluminación y variaciones de fondos. Incluso, se trató de aumentar el Dataset mediante técnicas de "Data Augmentation" con aplicaciones externas como Leonardo AI para tener una amplia gama de imágenes. Se encontraron problemas al generar las imágenes específicamente distorsión en los textos de la marca que presentaban cada una, lo que requería un ajuste adicional para mejorar la calidad del conjunto de datos. Es por ello por lo que se utilizó la misma plataforma de Roboflow que ofrece la herramienta "Data Augmentation" para realizar la expansión del Dataset obtenido originalmente.

5.2 Realizar el Etiquetado de Cada Objeto en el Conjunto de Imágenes Según su Categoría

Esta etapa fue crucial para el correcto funcionamiento del proyecto. Se utilizó Roboflow para realizar el etiquetado, esta aplicación es de un uso bastante intuitivo e interactivo y ofrece una amplia gama de herramientas de etiquetado de imágenes para marcar de forma manual cada componente. Este proceso se realizó con cautela y considerando cada mínimo detalle de calidad en las imágenes para garantizar que el modelo tenga datos de entrenamiento preciso.

En la ilustración 35, se observa los resultados obtenidos luego del proceso de etiquetado de imágenes en Roboflow.

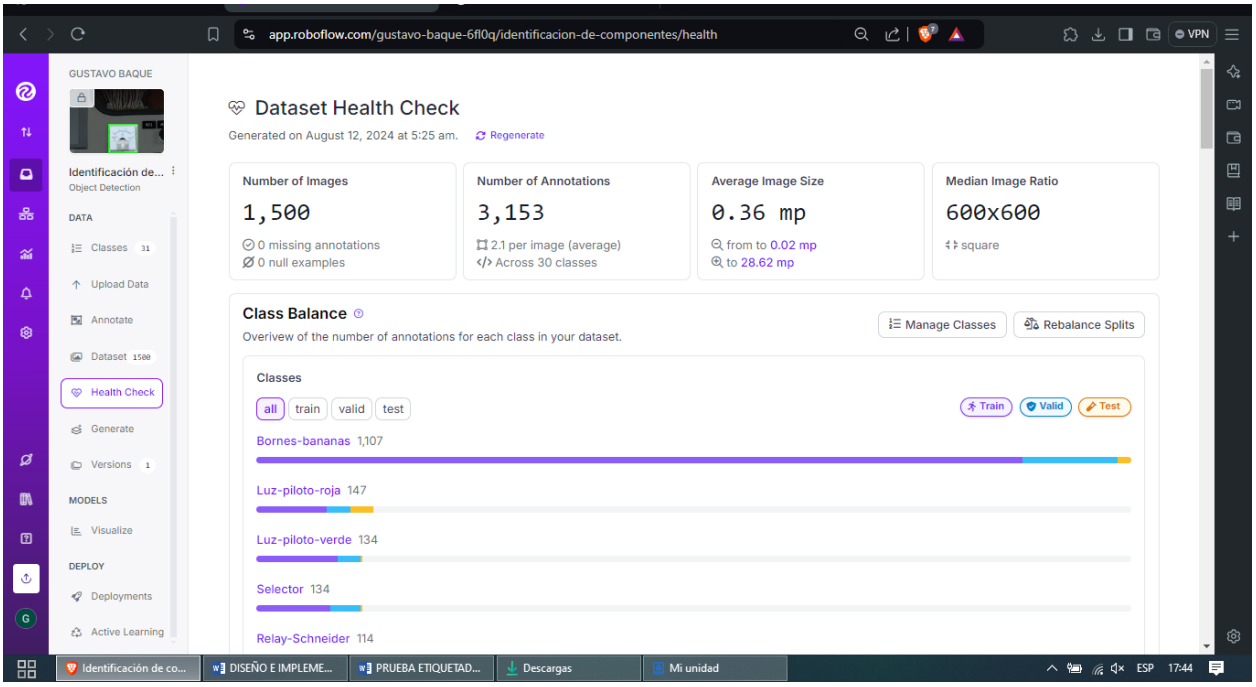

Ilustración 35: Etapa final de etiquetado de imágenes.

5.3 Desarrollar y Entrenar un Modelo de Clasificación

Para el entrenamiento y desarrollo del modelo, se utilizó la biblioteca YOLOv8 en Python. Debido, a su reconocimiento por su gran precisión y eficiencia en la segmentación de objetos en tiempo real. El entrenamiento del modelo fue realizado en múltiples iteraciones, realizando ajustes en los hiperparámetros y haciendo uso de técnicas para la validación que permita optimizar el rendimiento.

En la ilustración 36 y 37, se muestran los resultados del entrenamiento del modelo de identificación de componentes.

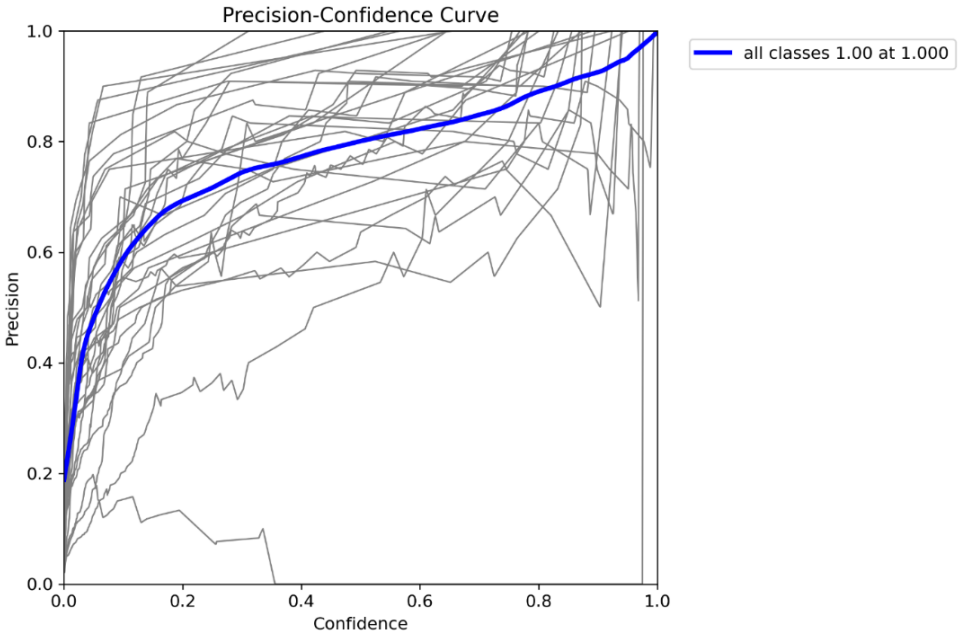

Ilustración 36: Curva de precisión del modelo.

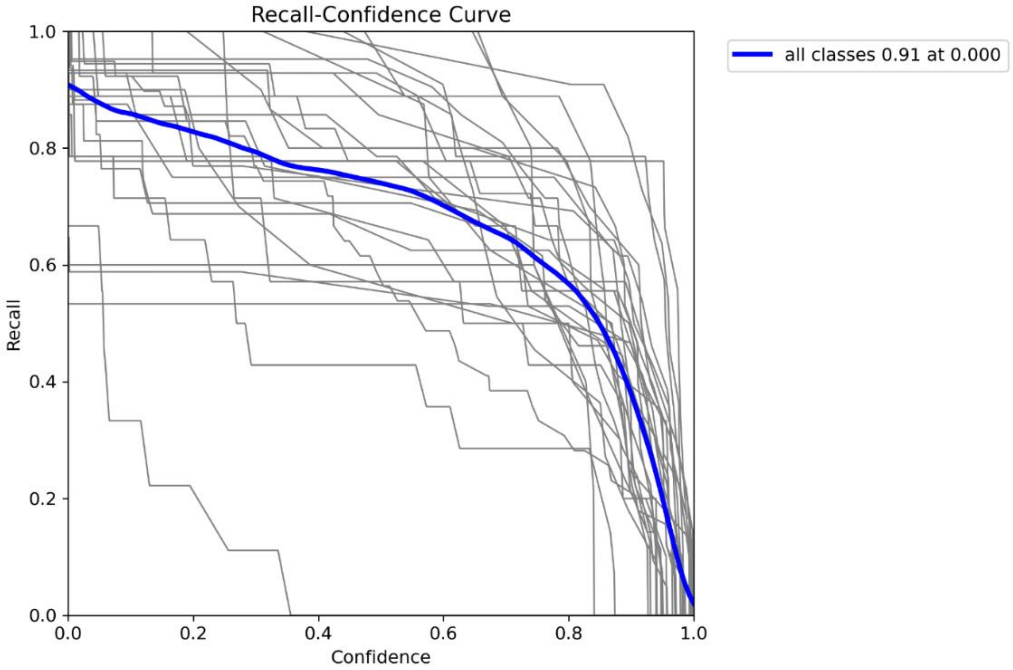

Ilustración 37: Curva obtenida del Recall del modelo.

5.4 Validar el Sistema en Condiciones Reales

Como evidencia final se realizaron varias pruebas en los laboratorios de Automatización de la UPS en los módulos didácticos. Para esto se integró una cámara de 1080p para realizar la captura de las imágenes y el diseño de una interfaz de usuario donde se muestra la segmentación de los componentes en tiempo real, así como una lista de los componentes identificados.

En las ilustraciones 38 y 39, se muestran las pruebas que se realizaron dentro del laboratorio de automatización de la Universidad Politécnica Salesiana sede Guayaquil.

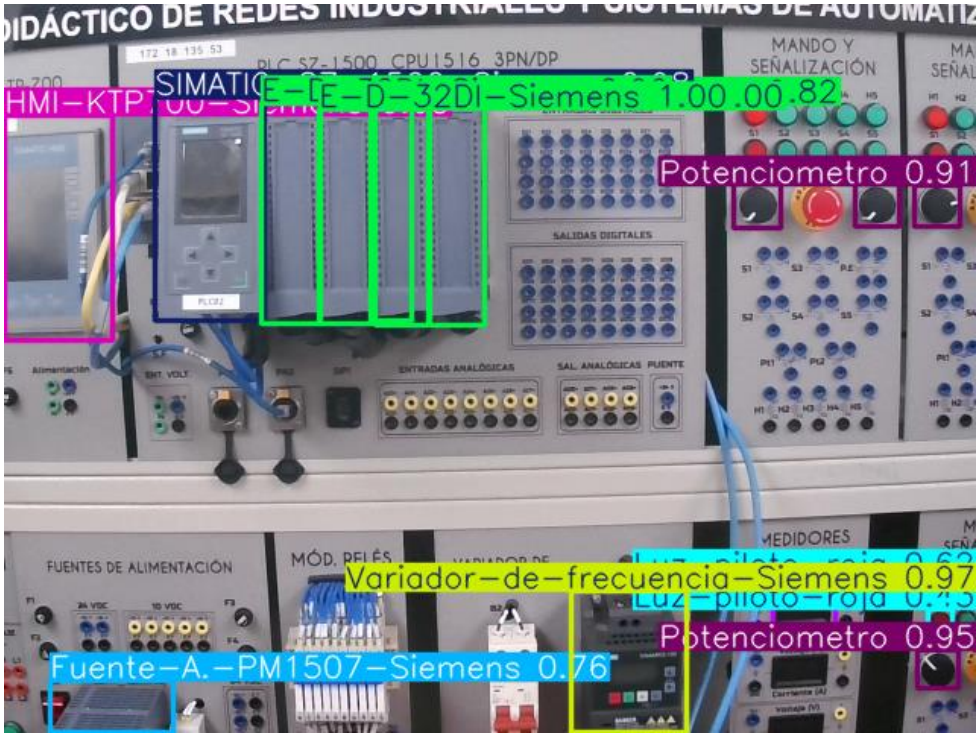

Ilustración 38: Captura realizada desde la interfaz.

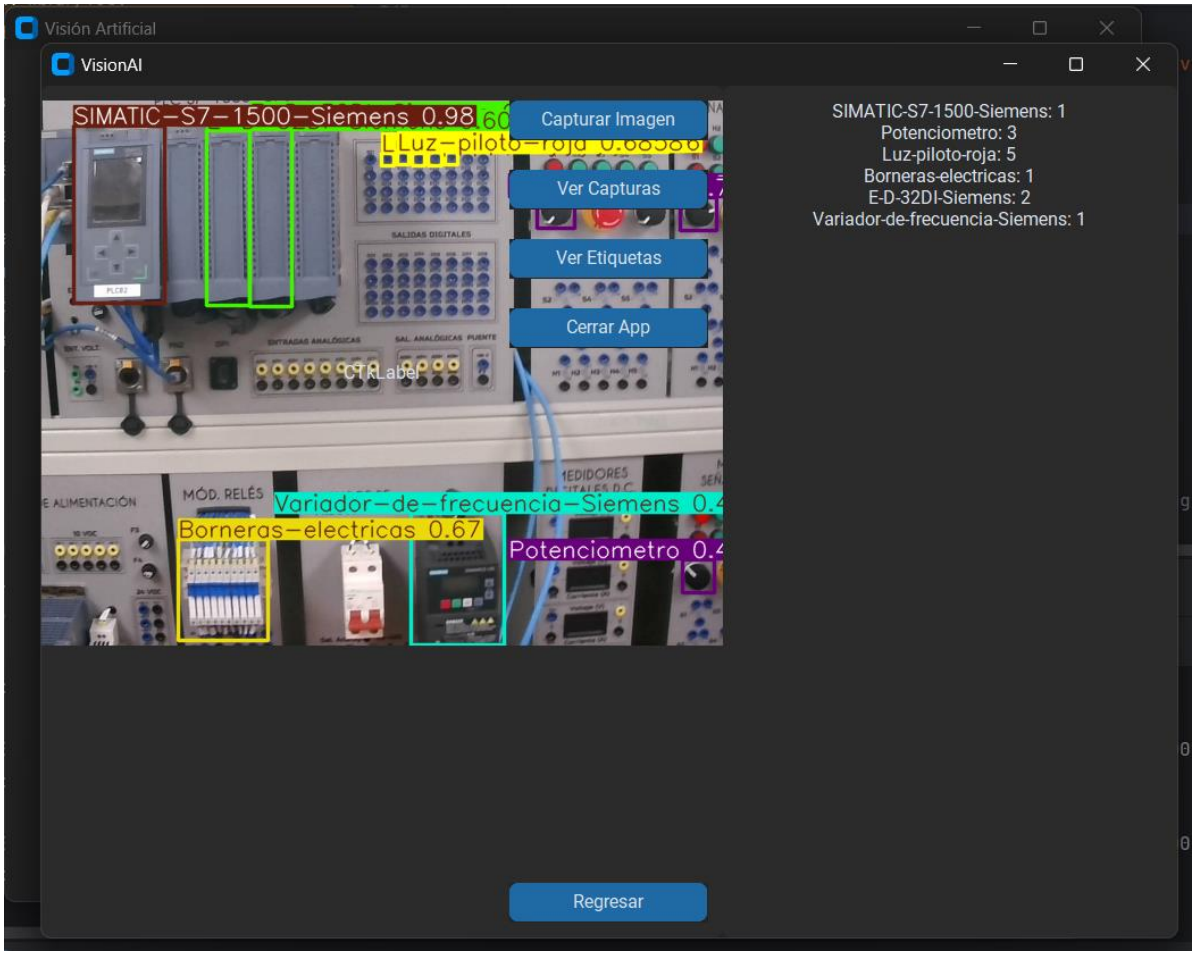

Ilustración 39: Validación del sistema en los laboratorios de Automatización de la Universidad Politécnica Salesiana sede Guayaquil.

Se puede observar, en las ilustraciones 40 y 41, la validación del sistema en condiciones reales en el laboratorio de Automatización. Se puede corroborar la identificación de los componentes tanto en la parte interior como exterior de los módulos didácticos.

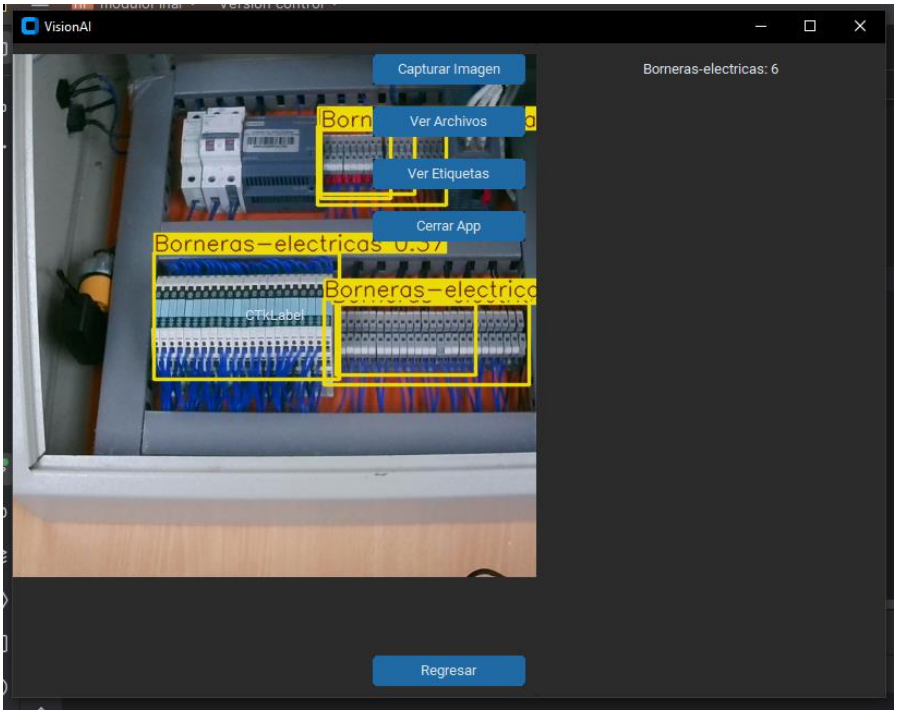

Ilustración 40: Validación del sistema en condiciones reales en el laboratorio de Automatización, con los módulos didácticos.

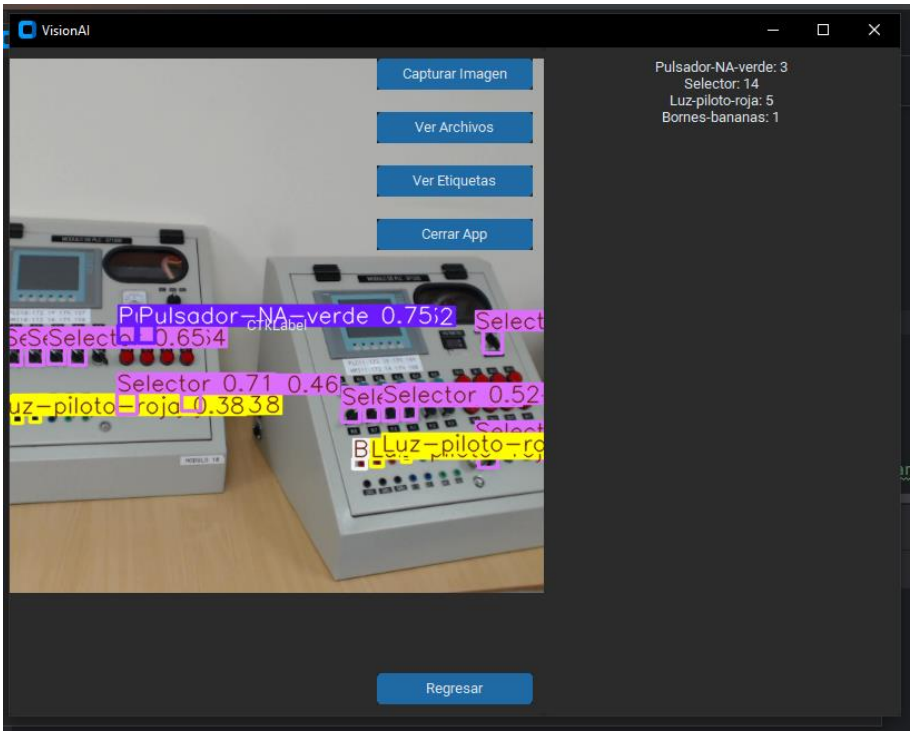

Ilustración **41**: Validación de la identificación en la parte exterior de los módulos didácticos en los laboratorios de Automatización.

CRONOGRAMA

A continuación, en la ilustración 42, se presenta el organigrama considerando el tiempo establecido, desde la semana 1 a la 13.

CRONOGRAMA DE ACTIVIDADES				
ACTIVIDADES	ABRIL	MAYO	JUNIO	JULIO
Fase de Investigación y Planificación				
Revisión de literatura sobre visión artificial, análisis de datos y técnicas de identificación de objetos.	X			
Definición de objetivos de la tesis y formulación de preguntas de investigación.	X			
Desarrollo del marco teórico y metodológico.	X			
Revisión y ajuste del plan de trabajo		X		
Fase de Recolección de Datos				
Diseño y desarrollo del sistema de visión artificial para la identificación de componentes.		X	X	
Recopilación de datos mediante la captura de imágenes de los componentes en condiciones controladas.	X	X		
Preprocesamiento de datos y preparación para el análisis.	X	X		
Fase de Análisis de Datos				
Aplicación de algoritmos de análisis de datos para identificar y clasificar los componentes en las imágenes.		X	X	
Interpretación de los resultados y comparación de diferentes técnicas de identificación.			X	
Redacción del capítulo de resultados y discusión.			X	
Fase de Redacción y Revisión				
Redacción del borrador de la tesis, incluyendo introducción, metodología y resultados.			X	X
Revisión y edición del contenido, incorporando comentarios y sugerencias del asesor y otros revisores.			X	X
Preparación del formato final y corrección de errores gramaticales y ortográficos.			X	X
Fase de Defensa y Entrega				
Preparación de la presentación de la tesis y ensayos para la defensa oral.				X
Defensa de la tesis ante el comité evaluador.				X
Corrección final y entrega de la versión final de la tesis.				X

Ilustración 42: Cronograma de actividades.

PRESUPUESTO

El presupuesto estimado incluirá costos asociados con la adquisición de equipos de visión artificial, software, desarrollo de software, pruebas y validación, así como los costos de personal involucrado en el proyecto. Además, se hará uso de una laptop HP Pavilion Gaming que es de uso personal necesaria para el entrenamiento del modelo. En la tabla 2 se encuentran detallados los gastos del proyecto.

Tabla 2

Presupuesto

CANTIDAD	MATERIALES	P. UNITARIO	P. TOTAL
1	cámara web	$30,00	$30,00
1	Licencia software Roboflow	$0,00	$0,00
1	Licencia software Leonardo AI	$12,00	$12,00
400	Honorarios profesionales	$2,87	$1.148,00
		VALOR TOTAL	$1.190,00

CONCLUSIONES

Se logró crear un conjunto de imágenes diversificadas haciendo uso de sitios web, revistas científicas, manuales o capturas de imágenes que ayudaron a darle un rendimiento óptimo al modelo. Además, se utilizó una herramienta que ofrecía la plataforma de Roboflow denominada "Augmentation" la cual permitió darle más robustez al modelo mejorando la capacidad de identificación del sistema en condiciones reales.

Para el etiquetado de cada objeto, se utilizó una herramienta denominada cajas delimitadoras, la cual permitía encerrar el objeto de una manera precisa para ayudar a separar un componente de otro. El etiquetado preciso permitió al modelo aprender de manera efectiva las características de cada clase de objeto, resultando en una mayor exactitud en las predicciones. La validación de los datos etiquetados mostró que las etiquetas se alinearon correctamente con los objetos identificados en las imágenes.

El desarrollo y entrenamiento del modelo fue desarrollado en PyCharm, utilizando la librería Ultralytics para realizar un entrenamiento adecuado y preciso del modelo. Los resultados fueron satisfactorios como se puede observar en las ilustraciones 30 y 31; las cuales muestran a través de la matriz de confusión la precisión del entrenamiento del modelo de identificación de componentes.

La validación del sistema en condiciones reales confirmó que el modelo funciona correctamente en los laboratorios de Automatización. Los resultados de las pruebas en este escenario mostraron que el modelo mantuvo una alta precisión y confiabilidad en la identificación de objetos. Esta validación subrayó la robustez del sistema y su capacidad para operar con éxito en situaciones reales, cumpliendo con el objetivo de garantizar la aplicabilidad práctica del modelo.

RECOMENDACIONES

Se recomienda la expansión de este proyecto a otros laboratorios y departamentos dentro de la Universidad Politécnica Salesiana para maximizar los beneficios de la identificación automatizada de componentes eléctricos. Implementar este sistema en diferentes contextos educativos podría potenciar el aprendizaje práctico y técnico de un mayor número de estudiantes.

Es crucial mantener y actualizar regularmente el conjunto de datos utilizado para entrenar el modelo de visión artificial, incorporando nuevas imágenes y categorías de componentes eléctricos. Esto asegurará que el sistema se mantenga preciso y relevante, adaptándose a las necesidades cambiantes y a la evolución de la tecnología en el ámbito educativo y profesional.

Se sugiere explorar la integración de otras tecnologías complementarias, como la realidad aumentada (AR) y la realidad virtual (VR), para mejorar aún más la experiencia educativa. Estas tecnologías podrían proporcionar visualizaciones interactivas y en tiempo real de los componentes identificados, facilitando el aprendizaje y la comprensión de los estudiantes.

Finalmente, se recomienda realizar estudios comparativos con otros métodos de identificación automatizada para evaluar la efectividad y eficiencia del sistema propuesto. Estos estudios podrían incluir la comparación con tecnologías emergentes y enfoques alternativos, proporcionando una base sólida para futuras mejoras e innovaciones en el campo de la educación y la automatización.

REFERENCIAS BIBLIOGRAFICAS

VI Bibliografía

Argom tech. (s.f.). *Web Cam Full HD 1080P with Microphone & LEDs CAM50 [Imagen]*. Obtenido de Argom tech: https://www.argomtech.com/products/web-cam-full-hd-1080p-with-microphone-cam50

Barrios Arce, J. I. (26 de Julio de 2019). *La matriz de confusión y sus métricas*. Obtenido de Realth BIG DATA: https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/

Borella, B. (Septiembre de 2022). *Introducción a la visión artificial: procesos y aplicaciones [Imagen]*. Obtenido de Universidad Complutense: https://docta.ucm.es/entities/publication/072ca3fb-540f-4dc9-8a16-0241cb5cd986

Carvalho, L. (18 de Mayo de 2023). *Tecnología Educativa: desafíos e importancia de la integración de sistemas*. Obtenido de SYDLE: https://www.sydle.com/es/blog/tecnologia-educativa-e-integracion-de-sistemas-63518d4d4037f13569393469

Cukurola, M., Kent, C., & Luckin, R. (Junio de 2019). *Artificial Intelligence and Multimodal Data in the Service of Human Decision*. Obtenido de Research Gate: https://www.researchgate.net/publication/333749078_Artificial_intelligence_and_multimodal_data_in_the_service_of_human_decision-making_A_case_study_in_debate_tutoring

De la Rosa, F. (Septiembre de 2016). *ResearchGate*. Obtenido de Red Neuronal Convolucional: https://www.researchgate.net/figure/Red-neuronal-convolucional-4_fig7_308783857

Escalante, M. (3 de Agosto de 2023). *Qué es PyCharm y su comparación con otros IDEs*. Obtenido de abc xperts: https://abcxperts.com/que-es-pycharm-y-su-comparacion-con-otros-ides/

Ferro, C. (20 de Marzo de 2024). *Identificación de componentes de la placa de circuito: una guía completa*. Obtenido de Wevolver: https://www.wevolver.com/article/circuit-board-components-identification-a-comprehensive-guide

Franco, M., Romero, M., Palomar, M., Cobos, J., Álvarez, G., & Hernández, D. (s.f.). *DE NEURONAS BIOLÓGICAS A NEURONAS ARTIFICIALES, EL FASCINANTE MUNDO DE LAS REDES NEURONALES*. Obtenido de [Imagen]: Recuperado de: https://www.uaeh.edu.mx/divulgacion-ciencia/redes-neuronales/

García, F. (Junio de 2023). *a-Matriz-de-confusion-sin-normalizar-b-Matriz-de-confusion-normalizada [Imagen]*. Obtenido de ResearchGate: https://www.researchgate.net/figure/Figura-40-a-Matriz-de-confusion-sin-normalizar-b-Matriz-de-confusion-normalizada_fig11_334821433

Gavilán, I. (14 de 12 de 2023). *Principales aportaciones de la visión artificial [Imagen]*. Obtenido de Arsys: https://www.arsys.es/blog/vision-artificial

Github. (s.f.). *ultralytics [Imagen]*. Obtenido de Github: https://www.google.com/search?q=yolov8&oq=yolo&gs_lcrp=EgZjaHJvbWUqDggAEEUYJxg7GIAEGIoFMg4IABBFGCcYOxiABBiKBTIJCAEQRRg5GIAEMgcIAhAuGIAEMgcIAxAAGIAEMgcIBBAAGIAEMgYIBRBFGD0yBggGEEUYPTIGCAcQRRg90gEIMjE5NmowajeoAgiwAgE&sourceid=chrome&ie=UTF-8

Gómez, E. (8 de Noviembre de 2022). *Diferencia entre machine learning y deep learning*. Obtenido de Universitat Oberta de Catalunya: https://blogs.uoc.edu/informatica/es/machine-learning-vs-deep-learning-diferencias/

IBM. (25 de Junio de 2024). *Métricas*. Obtenido de IBM: https://www.ibm.com/docs/es/mas-cd/maximo-vi/continuous-delivery?topic=configuring-understanding-metrics

Jurado, M. (10 de Abril de 2024). *Mi primer proyecto de AI con Yolo (Detección de Objetos en Tiempo Real)*. Obtenido de CDO LATAM: https://cdo-latam.com/mi-primer-proyecto-de-ai-con-yolo-deteccion-de-objetos-en-tiempo-real/#:~:text=YOLOv8%20es%20un%20nuevo%20modelo,interfaz%20de%20l%C3%ADnea%20de%20comandos.

La Serna Palomino, N., & Román Concha, U. (30 de Dicembre de 2009). *Técnicas de Segmentación en Procesamiento Digital de Imágenes* . Obtenido de Universidad Nacional Mayor de San Marcos: https://revistasinvestigacion.unmsm.edu.pe/index.php/sistem/article/view/3299/2749

Larroso, C. (2 de Febrero de 2023). *Aprendizaje Supervisado en Machine Learning*. Obtenido de Datarmony: https://datarmony-web.ew.r.appspot.com/blog/aprendizaje-supervisado-algoritmos-ejemplos/

Londoño, P. (04 de abril de 2023). *Hubspot*. Obtenido de https://blog.hubspot.es/website/que-es-python

Mancilla, E. (20 de diciembre de 2020). *Invgate*. Obtenido de https://blog.invgate.com/es/machine-learning

Martin, R. (19 de junio de 2023). *INESEM Business School*. Obtenido de https://www.inesem.es/revistadigital/informatica-y-tics/vision-artificial-industrial/

Navarrete, M., & Yanse, J. (2024). *Diseño e implementación de un sistema de segmentación de componentes electrónicos en tarjetas electrónicas mediante visión artificial*. Obtenido de DSpace: https://dspace.ups.edu.ec/handle/123456789/27729

Parada Torralba, P. (14 de Septiembre de 2022). *Qué son las Redes Neuronales Convolucionales*. Obtenido de IEBS: https://www.iebschool.com/blog/redes-neuronales-convolucionales-big-data/

Pazmiño, B. (2023). *Desarrollo de un sistema de identificación de personas con un drone mediante el uso de redes neuronales artificiales y visión artificial*. Obtenido de DSpace: https://dspace.ups.edu.ec/handle/123456789/24197

Polanco Herrera , L. G., & Villaruel Cuadrado, E. J. (2009). *Diseño y construcción de un módulo didáctico con controladores programables; para el laboratorio de automatización industrial de procesos mecánicas de la Facultad de Ingeniería Mecánica*. Obtenido de BIBDIGITAL: https://bibdigital.epn.edu.ec/handle/15000/1616

Rivas, W., & Jaramillo, S. (2020). *Detección de emociones y reconocimiento facial utilizando aprendizaje profundo*. Obtenido de UTMACH: https://repositorio.utmachala.edu.ec/browse?type=subject&order=DESC&rpp=20&value=DETECCI%C3%93N+DE+EMOCION

Rosquez, A. (s.f.). *VISIÓN ARTIFICIAL*. Obtenido de ceupe: https://www.ceupe.com.ve/blog/vision-artificial.html

Rouhiainen, L. (Noviembre de 2018). *Inteligencia Artificial*. Obtenido de cdnstatics2: https://proassetspdlcom.cdnstatics2.com/usuaris/libros_contenido/arxius/40/39307_Inteligencia_artificial.pdf

Ultralytics. (12 de Noviembre de 2023). *Roboflow*. Obtenido de Ultralytics: https://docs.ultralytics.com/es/integrations/roboflow/

Valenzuela, S. (Septiembre de 2021). *Detección y Clasificación de Enfermedades en el Tomate Mediantye Deep Learning y Computer Visión*. Obtenido de SEDICI: https://sedici.unlp.edu.ar/handle/10915/139770?show=full

ANEXOS

Código de Python utilizado

```
import customtkinter as ctk
import cv2
from PIL import Image, ImageTk
import pandas as pd
import os
import tkinter as tk
from tkinter import filedialog, messagebox
from ultralytics import YOLO
import numpy as np

# Configuración del modelo de visión
model = YOLO("best.pt")

# Directorios de almacenamiento
CAPTURAS_DIR = "C:/Users/Dennis/Documents/moduloFinal/moduloFinal/Archivos"
# Cambia esta ruta a la de tu computadora
EXCEL_DIR = "C:/Users/Dennis/Documents/moduloFinal/moduloFinal/Archivos"
# Cambia esta ruta a la de tu computadora
os.makedirs(CAPTURAS_DIR, exist_ok=True)
os.makedirs(EXCEL_DIR, exist_ok=True)

# Lista de precios de los objetos
ITEMS = [
    {"name": "Base-fusible", "price": 1.36},
    {"name": "Borneras-electricas", "price": 12.82},
    {"name": "Bornes-bananas", "price": 1.23},
    {"name": "CM1241-RS-232-Siemens", "price": 316.50},
    {"name": "CM1243-5-Siemens", "price": 1002.00},
    {"name": "CSM1277-SIMATIC-NET-Siemens", "price": 319.00},
    {"name": "Disyuntor-1P", "price": 6.15},
    {"name": "Disyuntor-2P", "price": 15.70},
    {"name": "E-D-32DI-Siemens", "price": 387.00},
    {"name": "Fuente-A.-MeanWell", "price": 122.00},
    {"name": "Fuente-A.-PM1507-Siemens", "price": 414.00},
    {"name": "HMI-KTP700-Siemens", "price": 686.00},
    {"name": "Luz-piloto-roja", "price": 4.82},
    {"name": "Luz-piloto-verde", "price": 4.82},
    {"name": "Modulo-PM1207-Siemens", "price": 90.58},
    {"name": "Parada-de-emergencia", "price": 5.00},
    {"name": "Potenciometro", "price": 63.16},
    {"name": "Pulsador-N.A.-verde", "price": 12.17},
    {"name": "Relay-Camsco", "price": 4.11},
    {"name": "Relay-Schneider", "price": 12.00},
    {"name": "Relay-Siemens", "price": 8.40},
    {"name": "Router-tplink", "price": 20.00},
    {"name": "SCALANCE-XB005-Siemens", "price": 442.00},
    {"name": "SIMATIC-S7-1200-Siemens", "price": 335.00},
    {"name": "SIMATIC-S7-1500-Siemens", "price": 665.00},
    {"name": "Selector", "price": 22.36},
    {"name": "Siemens-LOGO-Power", "price": 101.50},
    {"name": "Variador-de-frecuencia-Siemens", "price": 170.00},
```

```
    {"name": "Voltimetro-Analogico", "price": 9.10},
    {"name": "Voltimetro-digital", "price": 5.60}
]


class App(ctk.CTk):

    def __init__(self):
        super().__init__()
        self.title("Visión Artificial en Modulos Didácticos")
        self.geometry("800x600")
        self.modo_claro = True  # Empezar en modo claro por defecto
        self.setup_ui()

    def setup_ui(self):
        self.switch_theme_button = ctk.CTkButton(self, text="Cambiar Modo
(Claro/Oscuro)", command=self.switch_theme)
        self.switch_theme_button.pack(pady=10, side=tk.TOP, anchor="ne")

        # Cargar el logo usando PIL
        try:
            logo_image = Image.open("logo4.png")  # Asegúrate de tener un
archivo logo.png
            logo_image = logo_image.resize((200, 150), Image.LANCZOS)  #
Ajustar tamaño del logo si es necesario
            self.logo_image_tk = ImageTk.PhotoImage(logo_image)
            self.logo_label = ctk.CTkLabel(self, image=self.logo_image_tk)
            self.logo_label.pack(pady=10)
        except FileNotFoundError:
            messagebox.showerror("Error", "No se encontró el archivo
logo4.png. Asegúrate de que el archivo esté en la ruta correcta.")

        self.instructions_button = ctk.CTkButton(self, text="Instrucciones",
command=self.show_instructions)
        self.instructions_button.pack(pady=10)

        self.start_button = ctk.CTkButton(self, text="Iniciar VisionAI",
command=self.open_vision_screen)
        self.start_button.pack(pady=10)

        self.close_button = ctk.CTkButton(self, text="Cerrar App",
command=self.quit)
        self.close_button.pack(pady=10, side=tk.BOTTOM)

    def switch_theme(self):
        if self.modo_claro:
            ctk.set_appearance_mode("dark")
            self.modo_claro = False
        else:
            ctk.set_appearance_mode("light")
            self.modo_claro = True

    def show_instructions(self):
        instructions = "1. Inicie VisionAI para ver la cámara en tiempo
real.\n\n" \
                       "2. Use el botón Capturar para guardar una imagen y
generar el archivo excel.\n\n" \
```

```
                        "3. Use Ver Archivos para comprobar que han generado
los archivos.\n\n" \
                        "4. Use Ver Etiquetas para visualizar todas las clases
de objetos que puede detectar el modelo.\n\n" \
                        "5. Use Cerrar App para finalizar esta aplicación."
        messagebox.showinfo("Instrucciones", instructions)

    def open_vision_screen(self):
        self.vision_screen = VisionScreen(self)
        self.vision_screen.grab_set()


class VisionScreen(ctk.CTkToplevel):

    def __init__(self, master):
        super().__init__(master)
        self.title("VisionAI")
        self.geometry("800x600")
        self.master = master
        self.modo_claro = master.modo_claro

        self.setup_ui()
        self.start_camera()

    def setup_ui(self):
        self.camera_frame = ctk.CTkFrame(self)
        self.camera_frame.place(relwidth=0.6, relheight=1, relx=0, rely=0)

        self.side_frame = ctk.CTkFrame(self)
        self.side_frame.place(relwidth=0.4, relheight=1, relx=0.6, rely=0)

        self.camera_label = ctk.CTkLabel(self.camera_frame)
        self.camera_label.pack(pady=10)

        self.capture_button = ctk.CTkButton(self, text="Capturar Imagen",
command=self.capture_image)
        self.capture_button.pack(pady=10, side=tk.TOP)

        self.view_captures_button = ctk.CTkButton(self, text="Ver Archivos",
command=self.view_captures)
        self.view_captures_button.pack(pady=10, side=tk.TOP)

        self.view_labels_button = ctk.CTkButton(self, text="Ver Etiquetas",
command=self.view_labels)
        self.view_labels_button.pack(pady=10, side=tk.TOP)

        self.close_button = ctk.CTkButton(self, text="Cerrar App",
command=self.quit)
        self.close_button.pack(pady=10, side=tk.TOP)

        self.back_button = ctk.CTkButton(self, text="Regresar",
command=self.destroy)
        self.back_button.pack(pady=10, side=tk.BOTTOM)

        self.label_text = tk.StringVar()
        self.label_display = ctk.CTkLabel(self.side_frame,
textvariable=self.label_text)
```

```
        self.label_display.pack(pady=10)

    def start_camera(self):
        self.video_capture = cv2.VideoCapture(1)
        self.update_frame()

    def update_frame(self):
        ret, frame = self.video_capture.read()
        if ret:
            frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
            results = model(frame)
            annotated_frame = results[0].plot()  # Draw bounding boxes on the
frame
            self.show_image_in_tkinter(annotated_frame)

            # Update labels
            self.update_labels(results)

        self.after(10, self.update_frame)

    def show_image_in_tkinter(self, frame):
        image = Image.fromarray(frame)
        image_tk = ImageTk.PhotoImage(image)
        self.camera_label.configure(image=image_tk)
        self.camera_label.image = image_tk

    def update_labels(self, results):
        labels = results[0].names
        counts = {}
        for det in results[0].boxes.cls:
            label = labels[int(det.item())]
            counts[label] = counts.get(label, 0) + 1
        self.label_text.set('\n'.join([f"{k}: {v}" for k, v in
counts.items()]))

    def capture_image(self):
        ret, frame = self.video_capture.read()
        if ret:
            frame = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
            results = model(frame)
            annotated_frame = results[0].plot()
            img_path = os.path.join(CAPTURAS_DIR, "captura.png")
            cv2.imwrite(img_path, cv2.cvtColor(annotated_frame,
cv2.COLOR_RGB2BGR))
            self.save_detection_data(results)
            messagebox.showinfo("Captura de Imagen ", f"Imagen capturada y
guardada en {img_path}")

    def save_detection_data(self, results):
        detection_data = []
        labels = results[0].names
        counts = {}
        for det in results[0].boxes.cls:
            label = labels[int(det.item())]
            counts[label] = counts.get(label, 0) + 1

        for label, count in counts.items():
```

```
            item = next((item for item in ITEMS if item["name"] == label),
None)
            if item:
                total_price = count * item["price"]
                detection_data.append({
                    "Elemento detectado": label,
                    "Cantidad": count,
                    "Precio Unitario": item["price"],
                    "Total": total_price
                })

        df = pd.DataFrame(detection_data)
        total_price_sum = df["Total"].sum()
        df.loc[len(df.index)] = ["Precio total", "", "", total_price_sum]

        #total_objects_sum = df["Cantidad"].sum()
        #df.loc[len(df.index)] = ["Total elementos", "", "",
total_objects_sum]

        capture_name =
f"prefactura_{os.path.basename('captura.png').split('.')[0]}.xlsx"
        excel_path = os.path.join(CAPTURAS_DIR, capture_name)
        df.to_excel(excel_path, index=False)
        return df

    def view_captures(self):
        captures = os.listdir(CAPTURAS_DIR)
        if not captures:
            messagebox.showinfo("Ver Archivos", "No hay capturas para
mostrar.")
            return

        self.captures_screen = CapturesScreen(self)
        self.captures_screen.grab_set()

    def view_labels(self):
        labels = [item["name"] for item in ITEMS]
        messagebox.showinfo("Etiquetas del Modelo\n", "\n".join(labels))


class CapturesScreen(ctk.CTkToplevel):

    def __init__(self, master):
        super().__init__(master)
        self.title("Ver Capturas")
        self.geometry("800x600")
        self.master = master
        self.setup_ui()

    def setup_ui(self):
        self.capture_listbox = tk.Listbox(self)
        self.capture_listbox.pack(pady=10, fill=tk.BOTH, expand=True)

        self.load_captures()

        #self.export_button = ctk.CTkButton(self, text="Exportar Excel",
command=self.export_to_excel)
```

```
        #self.export_button.pack(pady=10, side=tk.BOTTOM)

        self.back_button = ctk.CTkButton(self, text="Regresar",
command=self.destroy)
        self.back_button.pack(pady=10, side=tk.BOTTOM)

    def load_captures(self):
        captures = os.listdir(CAPTURAS_DIR)
        for capture in captures:
            self.capture_listbox.insert(tk.END, capture)

    def export_to_excel(self):
        captures = [file for file in os.listdir(CAPTURAS_DIR) if
file.endswith('.png')]
        if not captures:
            messagebox.showinfo("Exportar a Excel", "No hay capturas para
exportar.")
            return

        data = []
        for capture in captures:
            img_path = os.path.join(CAPTURAS_DIR, capture)
            frame = cv2.imread(img_path)
            results = model(frame)
            labels = results[0].names
            counts = {}
            for det in results[0].boxes.cls:
                label = labels[int(det.item())]
                counts[label] = counts.get(label, 0) + 1

            for label, count in counts.items():
                item = next((item for item in ITEMS if item["name"] == label),
None)
                if item:
                    total_price = count * item["price"]
                    data.append({
                        "Elemento detectado": label,
                        "Cantidad": count,
                        "Precio Unitario": item["price"],
                        "Total": total_price
                    })

        df = pd.DataFrame(data)
        total_price_sum = df["Total"].sum()
        df.loc[len(df.index)] = ["Precio total", "", "", total_price_sum]
        excel_path = os.path.join(EXCEL_DIR, "prefactura_tablero.xlsx")
        df.to_excel(excel_path, index=False)
        messagebox.showinfo("Exportar a Excel", f"Datos exportados a
{excel_path}")


if __name__ == "__main__":
    app = App()
    app.mainloop()
```

UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR

SALESIANOS DON BOSCO

Guayaquil, 03 de agosto del 2024

Ing. Orlando Barcia, Msc.

Director de Carrera de Electrónica y Automatización.

De mis consideraciones:

Yo, Livington Miranda Delgado, portador de la cédula de ciudadanía No. 0930635172 tutor de trabajo de titulación **"DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE IDENTIFICACIÓN DE COMPONENTES ELÉCTRICOS EN MÓDULOS DIDÁCTICOS MEDIANTE VISIÓN ARTIFICIAL"**, informo la calificación al Trabajo de Tituación de los estudiantes de la Malla Ajuste: GUSTAVO ADRIAN BAQUE CATAGUA y DENNIS FERNANDO BENAVIDES PONCE.

Criterio	Descripción del criterio	Puntaje	Observaciones
Planteamiento e identificación del problema	Se muestra la importancia del problema y la contribución que se quiere alcanzar con el Proyecto técnico.	15	
Revisión del marco teórico y fuentes de información	Este criterio estable la relación entre la revisión literaria y el problema a abordar en el Proyecto técnico, así como el adecuado nivel de exhaustividad en la revisión de las fuentes de información.	15	
Contenido Metodológico	Se establecen con claridad y de manera estructurada las distintas fases, uso de métodos, herramientas, diseños, recursos, materiales, etc, para el desarrollo del Proyecto técnico y la propuesta de solución.	20	
Funcionalidad	Permite evaluar el nivel de funcionalidad del trabajo desarrollado, tomando en cuenta los objetivos del mismo.	30	
Presentación de Resultados	Se expresan o presentan los resultados alcanzados en el desarrollo del proyecto técnico y cómo se relacionan con el cumplimiento de los objetivos, el impacto y la innovación.	15	
Conclusiones Recomendaciones	Este criterio establece la claridad con que el autor expone su posición y sus ideas respecto a las conclusiones y recomendaciones expresadas.	5	
	PUNTAJE FINAL:	**100**	

Por la atención que se sirva dar a la presente, quedo de usted muy agradecido.

Atentamente,

Ing. Livington Miranda Delgado, Msc.

Docente Tutor.