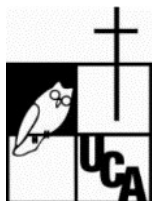


UNIVERSIDAD CENTROAMERICANA JOSÉ SIMEÓN CAÑAS
UNIVERSIDAD DON BOSCO



DETECCIÓN DE DEFECTOS EN AISLADORES ELÉCTRICOS
EN LÍNEAS DE TRANSMISIÓN UTILIZANDO REDES
NEURONALES CONVOLUCIONALES EN EL SALVADOR

TRABAJO DE INVESTIGACIÓN PREPARADO PARA EL
DECANATO DE POSTGRADOS Y EL CENTRO DE ESTUDIOS DE
POSTGRADOS UDB

PARA OPTAR AL GRADO DE:
MAESTRO EN GERENCIA DE MANTENIMIENTO INDUSTRIAL

POR:
AMADEO JOSÉ PORTILLO ROMERO
CÉSAR MARTÍNEZ SARAVIA
ROGELIO ANTONIO ALVARENGA SEGOVIA

SEPTIEMBRE 2025
ANTIGUO CUSCATLÁN, EL SALVADOR, C.A.

Rectores

Andreu Oliva de la Esperanza, S.J.

Mario Rafael Olmos Argueta, SDB.

Secretarias Generales

Lidia Gabriela Bolaños Teodoro

Yesenia Xiomara Martínez Oviedo

Decana de Postgrados UCA

Nelly Arely Chévez Reynosa

Decano de Facultad de ingeniería UDB

Moisés Roberto Guerra Menjívar

Directores de la Maestría en Gerencia de Mantenimiento Industrial

Diana Carolina Cruz UCA

José Luis Martínez UDB

Director del trabajo de graduación

Dr. Héctor Enrique Cañas Sánchez

AGRADECIMIENTOS

Quiero agradecer a mi amada esposa Cristina, que me ha apoyado durante todo este viaje, que me ha impulsado a no rendirme. A mi hijo Rafael, quién es la razón por la cual quiero ser mejor cada día. A mis padres y mi hermana que me han apoyado para lograr este objetivo. A mis compañeros, César y Rogelio, que ha sido un camino largo para nosotros. A nuestro director de tesis, Dr. Héctor Enrique Cañas, por guiarnos en este proceso. Finalmente, a Dios por todo el camino que he recorrido para estar acá.

Amadeo José Portillo Romero

En primer lugar, manifiesto mi profundo agradecimiento a mi prometida, **Brenda Leiva**, por su constante apoyo, paciencia y comprensión. De igual manera, a mi familia: a mis padres, **César Wilfredo Martínez y Eufemia Saravia de Martínez**, y a mi hermana **Alida**, quienes, con su respaldo incondicional, confianza y ánimo permanente me acompañaron en cada etapa de este camino académico. A mis compañeros y colegas de investigación, **Rogelio Alvarenga y Amadeo Portillo**, por su colaboración y el compromiso compartido durante el desarrollo de este trabajo. Al **Dr. Héctor Enrique Cañas** por su orientación cercana, su disposición para resolver nuestras dudas y, especialmente, por la generosidad al compartir sus conocimientos y experiencia, lo cual resultó fundamental para la culminación de este proyecto.

Finalmente, expreso mi gratitud a todas las personas que, aportaron su apoyo, motivación y acompañamiento, contribuyendo de manera significativa a la realización y conclusión de este esfuerzo académico.

César Martínez Saravia

Quiero expresar mi más profundo agradecimiento a mi novia, **Sara Araujo**, por haber sido mi mayor apoyo a lo largo de este proceso. Gracias por tu paciencia, comprensión y motivación constante en los momentos en que las cosas parecían más difíciles. Tus palabras de aliento, tu compañía y tu confianza en mí fueron una fuente de inspiración que me impulsó a seguir adelante hasta concluir esta etapa.

De manera muy especial, quiero dedicar este logro a mi mamá, **Evelyn De Alvarenga**, quien, aunque ya no está físicamente conmigo, dejó en mí las enseñanzas, valores y principios que me han guiado en cada paso de mi vida. Su ejemplo de esfuerzo, amor y dedicación sigue siendo mi mayor motivación y me ha permitido convertirme en la persona que soy hoy.

Asimismo, extiendo mi agradecimiento a mis colegas de tesis, **César Martínez** y **Amadeo Portillo**, por el compromiso, la colaboración y el trabajo en equipo que hicieron posible el desarrollo de este proyecto.

Finalmente, mi reconocimiento y gratitud al **Dr. Héctor Cañas**, nuestro asesor de tesis, por su guía, dedicación y valiosas recomendaciones a lo largo de todo el proceso. Su experiencia y acompañamiento fueron fundamentales para orientar nuestro trabajo en la dirección correcta y alcanzar los resultados propuestos.

Rogelio Antonio Alvarenga Segovia

RESUMEN

Los aisladores eléctricos son componentes esenciales dentro del sistema de transmisión de energía eléctrica en El Salvador. Las distintas condiciones ambientales y la variedad de materiales utilizados pueden provocar diferentes tipos de defectos en estos elementos, lo que compromete tanto la seguridad operativa como la confiabilidad de la red. Por ello, resulta fundamental contar con métodos efectivos de detección de defectos que complementen las rutinas de inspección y mantenimiento.

Este trabajo propone el uso de redes neuronales convolucionales (CNN) como herramienta para mejorar la identificación de defectos en aisladores eléctricos y, en consecuencia, optimizar la confiabilidad del sistema actual. Para ello, se ha utilizado como referencia la metodología CRISP-DM (*Cross-Industry Standard Process for Data Mining*), esta metodología al ser una de las más importantes en proyectos de ciencia de datos y minería de datos, ha sido un marco de trabajo ideal para estructurar este trabajo. CRISP-DM, esta metodología se compone de seis fases principales que guían el proceso de construcción de modelos de visión por computadora aplicado en los aisladores eléctricos. Se emplearon técnicas de detección de objetos, YOLOv8 (*You Only Look Once*), para localizar los aisladores en las imágenes y reconocer los defectos presentes mediante el entrenamiento del modelo.

El modelo fue entrenado para aprender directamente de las imágenes, identificando patrones característicos de aisladores defectuosos. Su desempeño se evaluó utilizando métricas de precisión, recall y mAP@0.5 (*mean Average Precision a un umbral de 0.5*), alcanzando un desempeño global del 89,7 %. Además, se desarrolló una interfaz que permite cargar imágenes de aisladores, procesarlas y mostrar la presencia de defectos con un nivel de certeza entre 0,49 y 0,74, mientras que la detección de los aisladores en sí alcanza un nivel de confianza de 0,83.

El enfoque basado en CNNs demuestra ser una herramienta eficaz para la detección automatizada de defectos, ya que permite reconocer patrones de daño específicos en los aisladores, contribuyendo a mejorar la seguridad, la confiabilidad operativa y la eficiencia en los tiempos de mantenimiento.

ÍNDICE

RESUMEN.....	i
ÍNDICE DE TABLAS	4
ÍNDICE DE FIGURAS	5
SIGLAS.....	7
CAPÍTULO 1 INTRODUCCIÓN.....	8
1.1 Antecedentes	9
1.2 Planteamiento del problema	13
1.3 Preguntas de investigación	14
1.4 Justificantes de la investigación	14
1.5 Objetivos	15
1.5.1 Objetivo general	15
1.5.2 Objetivos específicos.....	15
1.6 Alcance.....	15
1.7 Estado del arte	15
CAPÍTULO 2 MARCO TEÓRICO	19
2.1 Tipos de aisladores eléctricos y defectos.....	20
2.1.1 Materiales y diseño de aisladores eléctricos	20
2.1.2 Defectos comunes de los aisladores eléctricos	22
2.1.3 Repercusiones de los aisladores eléctricos defectuosos	23
2.2 Métodos convencionales de detección de defectos en aisladores eléctricos	24
2.2.1 Inspección visual	25
2.2.2 Termografía infrarroja	25
2.3 Aprendizaje profundo para la detección de objetos	26
2.4 Aplicaciones en la detección de objetos.....	27
2.5 Redes neuronales.....	28

2.5.1	Redes neuronales convolucionales (CNNs).....	28
2.5.2	Conceptos básicos y arquitectura.....	29
2.5.3	Componentes claves y capas	29
2.5.4	Arquitecturas populares de CNNs	30
2.6	Arquitectura YOLO (You Only Look Once)	32
2.6.1	Descripción de la arquitectura	32
2.6.2	Visión general y evolución	33
2.6.3	Características clave y ventajas	34
2.6.4	Arquitectura YOLOV8	35
2.6.5	Diseño del modelo y capas	35
2.7	Metodologías para minería de datos.....	36
2.7.1	CRISP-DM	38
CAPÍTULO 3 MATERIALES Y MÉTODOS.....		41
3.1	Implementación metodología CRISP-DM	42
3.1.1	Entendimiento del negocio	42
3.1.2	Entendimiento de los datos.....	43
3.1.3	Preparación de los datos	44
3.1.4	Construcción del modelo	44
3.1.5	Prueba y evaluación del modelo	45
3.1.6	Despliegue del modelo	46
3.2	Recopilación y procesamiento de datos	46
3.2.1	Adquisición y anotación de imágenes	46
3.3	Implementación y configuración de modelos de detección de objetos	47
3.3.1	Selección de un modelo apropiado	47
3.3.2	Personalización y entrenamiento de modelos.....	47
3.4	Métodos para la evaluación de las CNN implementadas	50
3.4.1	Función de pérdida y optimización.....	50

3.4.2	Entrenamiento y parada anticipada.....	50
3.5	Detalles de implementación	51
3.5.1	Requisitos de hardware y software	51
3.5.2	Implementación del código y bibliotecas	51
CAPÍTULO 4 RESULTADOS Y DISCUSIÓN		53
4.1	Evaluación de la aplicación de redes convolucionales.....	54
4.2	Discusión de resultados	56
4.3	Análisis económico costo-beneficio.....	62
CAPÍTULO 5 CONCLUSIONES Y RECOMENDACIONES		63
5.1	CONCLUSIONES	64
5.2	RECOMENDACIONES	66
REFERENCIAS BIBLIOGRÁFICAS		67
ANEXOS.....		71
ANEXO A. CODIGO DEL ALGORITMO UTILIZADO		72

ÍNDICE DE TABLAS

Tabla 1.1 Comparación de los diferentes métodos de inspección de aisladores de transmisión. Adaptado de Banerjee (2018).....	11
Tabla 1.2 Trabajos relacionados a la de detección de defectos en aisladores basados en algoritmos de aprendizaje profundo. Fuente: (Yue Li et al., 2022)	17
Tabla 2.1 Ventajas y desventajas de la inspección visual. Fuente: (Han et al., 2009)	25
Tabla 2.2 Ventajas y desventajas de la termografía infrarroja. Fuente: (Han et al., 2009)	25
Tabla 2.3 Evolución de YOLO Fuente: (Terven et al., 2023).....	34
Tabla 3.1 Cotización de equipo de cómputo para entrenamiento del modelo	51
Tabla 4.1 Descripción de funcionalidades de la interfaz	54
Tabla 4.2 Comparación de costos análisis de imágenes.....	62

ÍNDICE DE FIGURAS

Figura 1.1 Inspección manual realizada tradicionalmente de lado izquierdo vs inspección con UAV de lado derecho.	10
Figura 1.2 Tendencia de producción de investigaciones basadas en modelos de aprendizaje profundo enfocados en aisladores, del año 2012 al 2022.	16
Figura 2.1 Diferentes tipos de aisladores utilizados en sistemas de transmisión de energía (INMR, 2022).	21
Figura 2.2 Arquitectura de Redes Neuronales Convolucionales (CNN) (Dhakal, 2024).	29
Figura 2.3 Arquitectura LeNet-5 (Fuente: https://goo.su/3BwOK)	30
Figura 2.4 Arquitectura AlexNet (Fuente: https://goo.su/NRYz2Lt).....	30
Figura 2.5 Arquitectura VGGNet (Fuente: https://goo.su/dSMm).....	31
Figura 2.6 Arquitectura Inception (Fuente: https://goo.su/1pzFWG).....	31
Figura 2.7 Arquitectura ResNet (Fuente: https://goo.su/V1rwReh).....	32
Figura 2.8 Arquitectura YOLOv8 (Fuente: https://goo.su/869w9).....	36
Figura 2.9 Evolución de los procesos y minería de datos (Plotnikova et al., 2020)	38
Figura 2.10 Elementos de la metodología (Sharda et al., 2018)	39
Figura 3.1 Flujo actual del proceso de inspección de líneas de transmisión (Fuente: Elaboración propia).....	43
Figura 3.2 Arquitectura de la solución implementada (Fuente: Elaboración propia).	45
Figura 3.3 Imágenes ingresadas en el sistema (Fuente: Elaboración propia).	47
Figura 3.4 Distribución del dataset en conjuntos de entrenamiento, validación y prueba.	48
Figura 3.5 Interfaz del proceso de entrenamiento del modelo de detección de defectos en aisladores eléctricos.	48
Figura 3.6 Desempeño del modelo durante el entrenamiento.....	49
Figura 4.1 Interfaz de procesamiento de imágenes en el sistema.	54
Figura 4.2 Ventana información del sistema.....	55
Figura 4.3 Ventana de configuración de rutas del sistema.....	55
Figura 4.4 Interfaz de procesamiento de imágenes en ejecución.....	56
Figura 4.5 Imagen cargada.....	57
Figura 4.6 Resultado del análisis con inteligencia artificial.....	57
Figura 4.7 Detección de aislador polimérico contaminado.....	58
Figura 4.8 Detección de aislador flameado.....	59

Figura 4.9 Detección de aislador quebrado.....	60
Figura 4.10 Generación de reporte PDF	61

SIGLAS

ADALINE	ADaptative LINear Element	Red neuronal adaptativa
AES		Applied Energy Services
BERT	Bidirectional Encoder Representations from Transformers	Representaciones de codificador bidireccional de transformadores
CAESS		Compañía de Alumbrado Eléctrico de San Salvador, S.A. de C.V.
CLESA		Compañía de luz eléctrica salvadoreña
CNN	Convolutional Neural Network	Redes Neuronales Convolucionales
CPU	Central Processing Unit	Unidad Central de Procesamiento
CRISP-DM	Cross Industry Standard Process for Data Mining	Proceso Estándar Inter-Industrias para Minería de Datos
EEO		Empresa electrificadora de oriente
EPDM		Monómero de etileno propileno dieno
GPT	Generative Pre-trained Transformer	Transformador Generativo Preentrenado
GPU	Graphics Processing Unit	Unidad de Procesamiento Gráfico
IA	Artificial Intelligence	Inteligencia artificial
IDE	Integrated Development Environment	Desarrollo integrado
ILSVRC	ImageNet Large Scale Visual Recognition Challenge	Desafío de reconocimiento visual a gran escala
KDD	Knowledge Discovery in Databases	Descubrimiento de Conocimiento en Bases de Datos
LSTM	Long Short-Term Memory	Memoria a corto-largo plazo
RAM	Random Access Memory	Memoria de Acceso Aleatorio
R-CNN	Region-based Convolutional Neural Network	Redes Neuronales Convolucionales basadas en región
ReLU	Rectified Linear Unit	Unidad lineal rectificada
SF6		Azufre hexafluoruro
SSD	Single Shot MultiBox Detector	Detector de disparo único multicaja
TANH		Tangente hiperbólica
UAV	Unmanned Aerial Vehicle	Vehículos aéreos no tripulados
YOLO	You Only Look Once	Solo miras una vez

CAPÍTULO 1 INTRODUCCIÓN

1.1 Antecedentes

Los sistemas eléctricos de potencia juegan un papel fundamental en el desarrollo y funcionamiento de las sociedades modernas a nivel global. Estos sistemas son esenciales para suministrar electricidad de manera constante y confiable a diversos mercados, entre estos, zonas residenciales, centros comerciales, hospitales y áreas industriales. La integridad y eficiencia de los sistemas eléctricos de potencia son, por lo tanto, cruciales para el bienestar y la productividad de la población (Frizzo Stefenon et al., 2020).

Un sistema eléctrico de potencia se compone de tres procesos principales: generación, transmisión y distribución de electricidad (Nandhini y Prajith, 2023). La generación se refiere al proceso de producir energía eléctrica a partir de diversas fuentes, mientras que la transmisión implica el transporte de esta energía desde las plantas generadoras hasta los centros de distribución. Finalmente, la distribución se encarga de entregar la electricidad a los consumidores finales.

Las líneas de transmisión, en particular, desempeñan un rol crítico como medio de transporte de la energía eléctrica desde las plantas generadoras hasta los puntos de distribución. Estas líneas están conformadas por diversos elementos, entre los cuales se destacan los aisladores eléctricos. La función principal de los aisladores eléctricos es la de mantener el sistema adecuadamente aislado para prevenir defectos y asegurar un suministro eléctrico seguro y confiable (Frizzo Stefenon et al., 2020).

Las inspecciones en las líneas de transmisión de potencia tienen como objetivo buscar defectos en los aisladores eléctricos, estos defectos pueden provocar pérdidas en el suministro de energía, daños en la vida útil de las líneas de transmisión o en caso más serios, interrupción completa del servicio de energía eléctrica (Wang et al., 2021). Desde sus inicios, se ha conocido la importancia de tener a los aisladores en una rutina de inspección periódica, dicha inspección se realizaba de forma manual, es decir una persona previamente capacitada debía subir a las líneas de alta tensión a inspeccionar los aisladores uno por uno (Huang et al., 2017). Esta forma de inspeccionar era confiable y precisa pero aparte supone un trabajo mucho más laborioso, desgastante, tardado y costoso. Además, su principal punto en contra es el alto riesgo al cual se ven expuestos los trabajadores y por muy capacitados que estén y con el equipo de protección adecuado, se exponen a recibir una descarga eléctrica, que, en líneas de alta tensión, podría provocar accidentes mortales.

En las líneas de transmisión, los aisladores están expuestos a ciertas condiciones medioambientales lo cual facilita la acumulación de contaminación en su superficie, esta se disuelve con la humedad formando electrolitos, aumentando el deterioro del aislador e incrementando la posibilidad de un efecto de arco eléctrico.

Cabe resaltar que, aproximadamente el 82% de los defectos en líneas de transmisión se deben a la acumulación de contaminantes que podrían ser previamente detectadas (Banerjee, 2018). Con el

avance de la tecnología se han ido diseñando nuevas formas de ejecutar esta labor como se muestra en la Figura 1.1.

Los vehículos aéreos no tripulados (UAV), se han ido usando progresivamente para mejorar la eficiencia de la inspección y reducir la carga laboral del personal técnico y su exposición a riesgos. Los primeros usos de los UAV fueron por medio de un técnico que lo dirigía con una pantalla en mano, esto aparte de tomar mucho tiempo, tiene la desventaja que se puede dar un mal diagnóstico debido a factores como la resolución de la pantalla o problemas de iluminación (Liu et al., 2023).

Hoy en día, los avances de la inteligencia artificial (IA), específicamente, el aprendizaje profundo se posiciona como una técnica de vanguardia en el ámbito de la detección de defectos en aisladores. Entre otros usos, el aprendizaje profundo se ha utilizado para la detección de objetos, reconstrucción de imágenes de alta resolución, de empañamiento, de imágenes y la detección efectiva de defectos en general. La evolución en la transmisión de datos de los UAV empieza por el envío de información a través de una red de comunicación móvil inalámbrica a un servidor de alta potencia informática, integrado con IA.

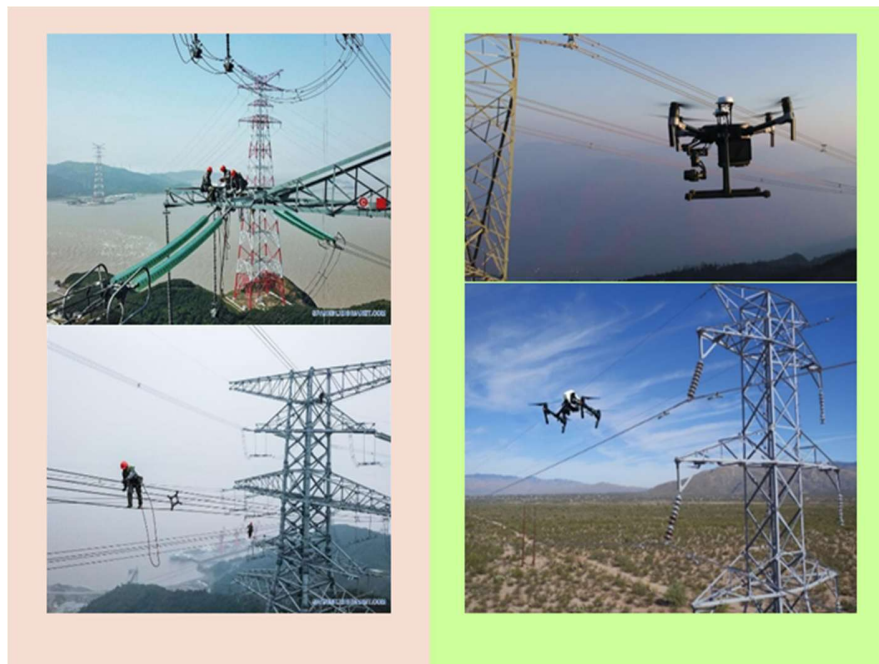


Figura 1.1 Inspección manual realizada tradicionalmente de lado izquierdo vs inspección con UAV de lado derecho.

La inspección con UAV representa distintos retos, entre ellos, la distribución geográfica de las líneas de transmisión que pueden ser en bosques, ríos, lagos, ciudades, y en cada una presenta un reto en cuanto al fondo de las imágenes capturadas por los UAV. Entre otros retos, es tener en cuenta los diferentes materiales de los cuales están fabricados los aisladores (vidrio, cerámica, entre otros), sus

diferentes formas, texturas y contornos hacen que el trabajo de detección de defectos se vuelva aún más complejo (Liu et al., 2023).

Entre otros métodos de inspección, se tienen los dispositivos informáticos portátiles integrados con IA que no requieren enviar datos a un servidor externo para ser analizados, por lo que no dependen de una red inalámbrica, y no se sacrifica la velocidad de detección (Gouda et al., 2022). En la Tabla 1.1, se muestra una comparación de los diferentes métodos de inspección de líneas de transmisión (Liu et al., 2023).

Tabla 1.1 Comparación de los diferentes métodos de inspección de aisladores de transmisión. Adaptado de Banerjee (2018).

Método de inspección	Ventaja	Desventaja
Inspección por técnico experto	Confiable y precisa	Peligro para la vida Consume mucho tiempo
Inspección por helicóptero	Efectiva respecto al tiempo Confiable	Alto costo
Inspección con UAV	Segura Efectiva en cuanto a costo y tiempo	Procesamiento manual tedioso (5-6GB)
Inspección automática	Sin peligro para la vida No requiere intervención humana Efectiva en cuanto a costo y tiempo	Sigue en desarrollo La precisión aún no es tan buena como la de un técnico experto.

En El Salvador, la empresa AES representa una gran parte del sistema de distribución del país, posee y administra las líneas de media tensión y es responsable de la entrega de la energía al usuario final, ya sea hogares o industrias.

Con una inversión que supera los 250.000 dólares, AES El Salvador ha adquirido 12 UAV de última generación y ha capacitado a 19 pilotos para monitorear la red de distribución de energía. La utilización de esta tecnología permite una evaluación más precisa del estado de la infraestructura eléctrica, la predicción de los fallos de funcionamiento de la red y la garantía de un suministro de energía eficiente y confiable a los usuarios. AES ha revelado que, a partir del año 2017, ha inspeccionado más de 900 kilómetros de líneas eléctricas utilizando esta innovación (AES El Salvador, 2021). Además, invierte en el desarrollo herramientas de software para el análisis de imágenes capturadas con UAV, lo que facilita la adopción de la IA para detectar posibles fallos en los equipos eléctricos y mejorar la eficiencia de las operaciones de mantenimiento. Estos avances

tecnológicos han posicionado a AES como líder en la implementación de soluciones innovadoras para el sector eléctrico, destacándose por su compromiso con la mejora continua y la excelencia operativa (AES El Salvador, 2021).

1.2 Planteamiento del problema

Los aisladores eléctricos son componentes esenciales en las líneas de transmisión de energía y representan el 5% del costo de capital de estas líneas y el 50% de los costos de mantenimiento por inspección y reemplazo. Además, son la causa de aproximadamente el 70% de los defectos que se producen en una línea de transmisión de energía. Las condiciones climáticas de El Salvador, como en áreas costeras pueden acelerar el deterioro de los materiales debido a la presencia de cloruros en el ambiente. (Palacios & Sánchez, 2025)

Compararon aisladores con 10 y 50 años de servicio y revelaron que los aisladores más antiguos presentan una reducción del 89.3% en su factor de calidad en comparación con los más recientes. Este factor de calidad evalúa tanto la capacidad mecánica como eléctrica de los aisladores.

Es innegable que los métodos y materiales para fabricar aisladores han ido mejorado significativamente con el tiempo. Por ejemplo, los aisladores producidos en el año 2005 mostraron una mejora del 90.95% en su factor de calidad en comparación con los fabricados en 1975 (Sanyal et al., 2020).

Aproximadamente el 82% de los defectos en los aisladores se producen en aquellos con acumulación de contaminación, lo cual puede detectarse con anticipación. Actualmente, la inspección manual, es realizada por un técnico especializado que sube a las líneas de transmisión para verificar el estado de los aisladores siendo uno de los métodos más utilizados en la actualidad. Aunque este enfoque es confiable y efectivo, representa riesgos para la seguridad del trabajador y requiere mucho tiempo. Para aisladores que operan a 33 kV, se recomienda un mantenimiento preventivo cada 8 a 18 meses o dependiendo a las condiciones medioambientales a las que están expuestos (Janaka, Lucas, & Fernando, 2010).

Considerando que la acumulación de contaminación en los aisladores eléctricos representa una de las principales causas de deterioro en su desempeño, y sumado a los recientes avances en inteligencia artificial, surge la necesidad de implementar técnicas de aprendizaje profundo, en particular redes neuronales convolucionales (CNNs), para entrenar un algoritmo predictivo que permita la detección temprana de defectos en dichos componentes. Este enfoque busca incrementar la confiabilidad y la disponibilidad de las líneas de transmisión, al mismo tiempo que reduce los costos asociados al mantenimiento. El algoritmo propuesto será validado utilizando bases de datos de dominio público internacionales, con el fin de comparar la precisión obtenida frente a la inspección manual tradicional.

1.3 Preguntas de investigación

- ¿Son necesarias las CNNs para la detección de defectos en aisladores en líneas de distribución en El Salvador?
- ¿Cómo detectar de forma exacta y precisa defectos en aisladores eléctricos utilizando las CNNs en líneas de distribución en El Salvador?
- ¿Se obtendrá una mejora en el rendimiento del proceso actual de detección de defectos en aisladores eléctricos utilizando las CNNs en líneas de distribución en El Salvador?

1.4 Justificantes de la investigación

La elección de este tema de investigación surge de una oportunidad de mejora identificada en el monitoreo y mantenimiento de aisladores eléctricos en las líneas de distribución en El Salvador. Históricamente, no se ha realizado una evaluación exhaustiva sobre cómo optimizar la detección temprana de defectos en estos componentes críticos. La aplicación de las CNNs se considera una estrategia proactiva para incrementar la confiabilidad y la disponibilidad de las líneas de transmisión, reduciendo al mismo tiempo los costos asociados al mantenimiento.

Actualmente, el personal de mantenimiento utiliza UAV para capturar imágenes de aisladores eléctricos en líneas de distribución. Una vez que las imágenes son tomadas, se descargan manualmente de la memoria del UAV y se procede a clasificar los defectos visualmente, lo que puede ser un proceso tedioso y propenso a errores humanos. Este método manual de inspección no solo consume tiempo, sino que también puede pasar por alto defectos incipientes debido a la limitación de recursos y la variabilidad en la evaluación humana. Al implementar un sistema de detección automatizada basado en las CNNs, las imágenes capturadas por los UAV podrían analizarse de manera rápida y precisa, identificando problemas potenciales en los aisladores sin la intervención humana directa. Esto no solo mejoraría la eficiencia del proceso de mantenimiento, sino que también aumentaría la precisión y la confiabilidad de la detección de defectos, permitiendo una respuesta más ágil y efectiva para garantizar la estabilidad del sistema eléctrico.

Esta investigación aportará nuevas soluciones al sector eléctrico salvadoreño, enfocándose en el uso de redes neuronales para aumentar la confiabilidad y el rendimiento de las líneas de transmisión. La implementación de este sistema servirá como una herramienta para el personal de mantenimiento, proponiendo un método más eficiente de gestión de defectos, que permitirá reducir los costos operativos y mejorar la continuidad del servicio eléctrico. Los ahorros generados mediante la optimización del monitoreo se podrían destinar a la evaluación y desarrollo de nuevos proyectos en otras áreas críticas del sistema eléctrico, a la inversión en tecnologías de mantenimiento predictivo, y a la capacitación de los equipos de trabajo en técnicas avanzadas de detección de defectos.

1.5 Objetivos

1.5.1 Objetivo general

Proponer un sistema de mantenimiento predictivo para la detección de defectos en aisladores eléctricos en líneas de distribución utilizando CNNs en El Salvador.

1.5.2 Objetivos específicos

- Caracterizar los defectos generados en los aisladores eléctricos en líneas de distribución y las herramientas utilizadas para su identificación.
- Entrenar las CNNs para la detección de defectos en aisladores eléctricos en líneas de distribución para su aplicación en El Salvador.
- Evaluar los modelos basados en CNNs entrenados para detectar defectos en aisladores eléctricos, utilizando imágenes de un repositorio abierto.

1.6 Alcance

Se propone entrenar un modelo predictivo basado en CNNs para la detección de defectos en aisladores eléctricos, a partir de una base de imágenes de aisladores de dominio público. El objetivo principal de la investigación es generar un modelo de detección precisa y confiable de defectos en aisladores sin necesidad de la intervención humana, con el fin de reducir el riesgo al cual están expuestos actualmente los técnicos que inspeccionan las líneas de distribución.

La implementación de este modelo en la práctica para la inspección de aisladores en líneas de distribución estará sujeta a la decisión de una empresa de distribución de energía eléctrica como AES de ponerlo a prueba y además de contar con el equipo de UAV necesario.

Esta investigación pretende ser beneficiosa para las empresas de distribución de energía eléctrica en El Salvador, ya que les permitirá ofrecer un servicio más confiable a partir de la detección temprana de defectos en aisladores eléctricos de sus líneas de distribución, y reducir el riesgo al cual están expuestos sus técnicos de campo que se encargan de ejecutar dichas inspecciones de forma manual.

1.7 Estado del arte

Hace poco más de una década, en el año 2012, hubo un aumento en la adopción de algoritmos predictivos basados en aprendizaje profundo para la detección de defectos en aisladores de líneas de distribución demostrando una notable eficacia y eficiencia en el proceso. El año 2022, en particular,

fue testigo de un aumento notable de la producción académica en torno a estos modelos de aprendizaje profundo, con un total de 2442 publicaciones en las bases de datos de la Web de la Ciencia, una cifra muy similar a la del año 2021, confirmando el auge de este campo de investigación, la Figura 1.2, muestra la evolución en la producción de estas investigaciones. (Yue Li et al., 2022)

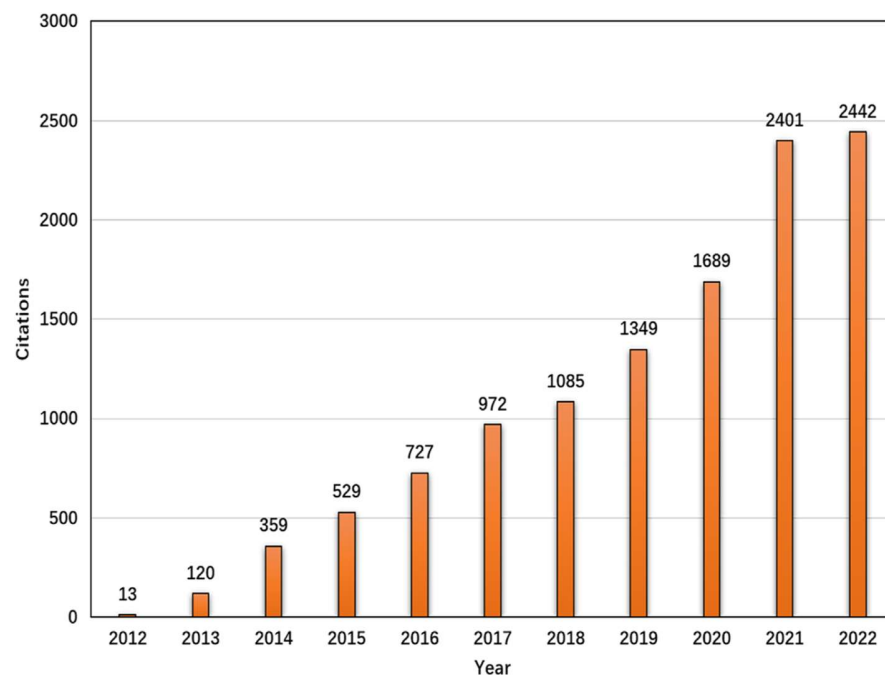


Figura 1.2 Tendencia de producción de investigaciones basadas en modelos de aprendizaje profundo enfocados en aisladores, del año 2012 al 2022. (Yue Li et al., 2022)

En comparación con métodos convencionales, los enfoques basados en aprendizaje profundo presentan un mayor grado de adaptabilidad y flexibilidad a la hora de detectar anomalías de aislamiento en una amplia gama de entornos y circunstancias. Algunos investigadores se centraron en analizar la ubicación precisa de estos defectos, mientras que otros se concentraron únicamente en examinar la ubicación de los aisladores. En la Tabla 1.2 se presenta una comparativa de lo que fue desarrollado en este campo de investigación entre el año 2019 al 2023, en ella se observan algunos métodos de aprendizaje basados en defectos de aisladores y como estos cubren o no diferentes etapas de procesamiento, los distintos pasos que normalmente forman en el flujo de trabajo en la detección de defectos en aisladores eléctricos.

Tabla 1.2 Trabajos relacionados a la de detección de defectos en aisladores basados en algoritmos de aprendizaje profundo. Fuente: (Yue Li et al., 2022)

Autor	Año	Método principal	Aumento de datos	Detección de aislador	Ubicación de defecto
Sampedro et al. [8]	2019	CNN	☑	☑	☑
Jiang et al. [54]	2019	SSD	☑	☑	☑
Zhang et al. [55]	2019	SWT	☑		☑
Miao et al. [621]	2019	SSD	☑	☑	
Wang et al. [1631]	2019	SCA	☑		☑
Han et al. [59]	2019	CNN	☑	☑	☑
Tao et al. [30]	2020	CNN	☑	☑	☑
Bet et al. [7]	2020	Faster R-CNN	☑	☑	☑
Zhao et al. [22]	2020	AVSCNet			☑
Wang et al. [22]	2020	ResNeSt	☑	☑	☑
Bet al. [58]	2020	Faster R-CNN	☑		☑
Liu et al. [61]	2020	RetinaNet		☑	☑
Liu et al. [60]	2021	YOLOv3	☑	☑	
Wen et al. [56]	2021	Exact R-CNN	☑	☑	☑
Zhao et al. [10]	2021	Faster R-CNN		☑	☑
Zhang et al. [37]	2022	Yolov5	☑	☑	☑

Lu et al. [38]	2022	SCRDet	☑	☑	☑
Yang et al. [42]	2022	YOLOv3	☑	☑	☑
Dai et AL [46]	2022	YOLOX	☑	☑	☑
Souza et al. [41]	2023	Hybrid- YOLO	☑	☑	☑

CAPÍTULO 2 MARCO TEÓRICO

2.1 Tipos de aisladores eléctricos y defectos

La red de transmisión de energía eléctrica constituye un sistema altamente complejo, que opera como una gran máquina interconectada, extendiéndose a lo largo de vastas distancias desde los centros de generación hasta los usuarios finales. Este sistema está conformado por diversos componentes —entre ellos generadores, transformadores, líneas de transmisión y dispositivos de protección— los cuales, debido a las altas potencias involucradas y al nivel tecnológico requerido, suelen presentar dimensiones considerables y elevados costos de adquisición, instalación y mantenimiento.(Frizzo Stefenon et al., 2020)

En este contexto, la preservación de la integridad de dichos activos resulta crucial, no solo por razones económicas, sino también por la necesidad de garantizar la seguridad operativa del sistema y proteger tanto al personal técnico como a la población en general. El aseguramiento de un funcionamiento continuo, seguro y confiable del sistema de transmisión depende, en gran medida, de la correcta selección, instalación y mantenimiento de los dispositivos de protección.

Entre estos dispositivos, los aisladores eléctricos desempeñan un rol fundamental dentro de las líneas de distribución. Su función principal es aislar los conductores de alta tensión, evitando fugas de corriente y proporcionar soporte mecánico a los conductores contra el viento, hielo, lluvia, etc. La adecuada gestión del mantenimiento de los aisladores es, por tanto, esencial para la confiabilidad del sistema de distribución y la mitigación de riesgos operativos.(Monemi et al., 2024)

2.1.1 Materiales y diseño de aisladores eléctricos

Los sistemas de transmisión de energía dependen de los aisladores para separar los conductores eléctricos de sus estructuras de soporte, evitando fugas eléctricas y garantizando una operación segura. La elección del material y el diseño del aislador tiene un impacto significativo en el rendimiento y la confiabilidad general del sistema de transmisión de energía.

Esta sección ofrece una visión detallada de los diversos materiales y diseños de aisladores comúnmente utilizados en sistemas de transmisión de energía. Este análisis servirá como base para el desarrollo de un sistema de detección y reconocimiento de defectos en aisladores basado en CNNs profundas.



Figura 2.1 Diferentes tipos de aisladores utilizados en sistemas de transmisión de energía (INMR, 2022).

Los aisladores cerámicos, compuestos predominantemente de porcelana o vidrio como en la figura 2.1, se han utilizado ampliamente en sistemas de transmisión de energía debido a su superior capacidad eléctrica y propiedades mecánicas. Los aisladores de porcelana se fabrican combinando caolín, feldespato y cuarzo, cocidos a altas temperaturas para producir un denso, desafiante y material resistente a la humedad. Los aisladores de vidrio están hechos de vidrio templado o endurecido, ofreciendo alta resistencia mecánica y resistencia a la degradación medioambiental. Ambos aisladores de porcelana y vidrio demuestran alta resistencia dieléctrica, confiabilidad y larga duración vida útil, lo que los hace adecuados para líneas de transmisión de energía de alta tensión.

El aislador polimérico, también conocido como aislante compuesto, comprende caucho de silicona, caucho de monómero de etileno propileno dieno (EPDM) y varillas de fibra de vidrio el aislante de la carcasa, generalmente de caucho de silicona, proporciona excelentes propiedades hidrófobas y oposición a la radiación ultravioleta y la estabilidad ambiental. La varilla de fibra de vidrio imparte alta fortaleza mecánica y resistencia a fuerzas de flexión y tracción. Debido a su peso ligero naturaleza, alta solidez mecánica y a la degradación ambiental, los aisladores de polimérico han ganado popularidad en los últimos años y se están volviendo más frecuentes en líneas de distribución y transmisión.

Los aisladores con aislamiento de gas, emplean gas comprimido, como azufre hexafluoruro (SF₆), para proporcionar aislamiento eléctrico. SF₆ es un gas inerte y no tóxico con propiedades

dieléctricas excepcionales, lo que lo hace adecuado para aislamiento de aplicaciones de alta tensión. Los aisladores aislados en gas se utilizan comúnmente en equipo de maniobra de alta tensión, subestaciones aisladas en gas y líneas aisladas en gas debido a su diseño compacto, baja requisitos de mantenimiento y excelente rendimiento dieléctrico. Sin embargo, las preocupaciones sobre el impacto ambiental de las fugas de SF₆ y el desarrollo de alternativas los aisladores ecológicos con aislamiento de gas siguen siendo temas de investigación en curso.(Xing et al., 2022)

Los aisladores híbridos, combinan los beneficios de aisladores cerámicos y poliméricos mediante la utilización de un núcleo cerámico rodeado por un alojamiento de polímero. Este diseño ofrece la alta resistencia dieléctrica y confiabilidad de la cerámica de los aisladores, junto con las propiedades hidrofóbicas, livianas y ambientales. Resistencia de aisladores poliméricos se espera que los aisladores híbridos ganen más tracción a medida que ofrecen un rendimiento mejorado y requisitos de mantenimiento reducidos en comparación con aisladores tradicionales.(Saleem & Akbar, 2022)

2.1.2 Defectos comunes de los aisladores eléctricos

Los aisladores garantizan el aislamiento eléctrico y mantienen la estabilidad y confiabilidad en el suministro eléctrico sistemas de transmisión. Sin embargo, los factores medioambientales y las tensiones operativas hacen que los aisladores sean susceptibles a diversos defectos. Estos defectos pueden comprometer el rendimiento de los aisladores y, si no se abordan con prontitud, pueden tener consecuencias catastróficas. Esta sección analiza los defectos comunes de los aisladores que se deben considerar al desarrollar un sistema basado en CNNs profundas para los defectos en aisladores y reconocimiento.

La falta de inspecciones a tiempo en los aisladores, en especial la contaminación, puede llevar a defectos y a interrupciones en el sistema eléctrico. (Monemi et al., 2024)

Los defectos más comunes que se pueden detectar en los aisladores eléctricos pueden ser las siguientes:

- Envejecimiento natural: debido a que algunos están hechos de goma de silicona, EPDM y resinas epoxi, están destinados a envejecer con el tiempo debido a su exposición con el ambiente, afectando sus propiedades dieléctricas, térmicas y mecánicas.
- Estrés eléctrico y ambiental: los aisladores experimentan degradación debido a la exposición a estrés eléctrico (descargas de corona, arcos de banda seca) y estrés ambiental (exposición al sol, vientos, polvos, lluvias, etc), provocando perforaciones en las cubiertas del aislador.
- Hidrofobicidad: exponer el aislador a condiciones de humedad y a la formación de bandas secas pueden distorsionar el campo eléctrico y causar descargas superficiales. Estas descargas pueden dañar la hidrofobicidad de la superficie al consumir la fina capa polimérica.

- Erosión: la exposición al ambiente puede llevar a la erosión de la superficie del polímero, formando grietas a través del aislador.
- Daño por efecto corona: ocurre cuando el campo eléctrico alrededor del aislador es lo suficiente intenso para ionizar el aire circundante, causando una descarga visible o audible. Este puede ocurrir en la superficie del aislador, en defectos del mismo y en las uniones con otros materiales(Saleem & Akbar, 2022)

2.1.3 Repercusiones de los aisladores eléctricos defectuosos

Los aisladores desempeñan un papel fundamental en los sistemas de transmisión eléctrica, ya que garantizan la estabilidad y seguridad del flujo de energía. Sin embargo, cuando presentan defectos, pueden generar múltiples problemas que afectan el rendimiento, la confiabilidad y la seguridad del sistema. En esta sección, se analizan los principales efectos negativos de los aisladores defectuosos, incluyendo la pérdida de energía, la disminución de la eficiencia, el daño a los equipos, los riesgos para la seguridad y el impacto ambiental.

Pérdida de energía y disminución de la eficiencia: Uno de los propósitos clave de los aisladores en las líneas de transmisión es evitar fugas de corriente y asegurar que la energía eléctrica se transporte de manera eficiente desde las centrales generadoras hasta los usuarios finales. Cuando un aislador está defectuoso, aumenta la corriente de fuga, lo que se traduce en pérdidas significativas de energía. Esto no solo reduce la eficiencia del sistema, sino que también incrementa los costos operativos para las compañías eléctricas, costos que eventualmente pueden trasladarse a los consumidores.(Monemi et al., 2024)

Daño a los equipos e interrupciones en el servicio: Los aisladores defectuosos pueden comprometer tanto la estabilidad mecánica como la integridad eléctrica de las líneas de transmisión. En algunos casos, pueden provocar descargas eléctricas, donde la electricidad atraviesa el aire y genera un cortocircuito. Estos eventos pueden causar daños graves en los aisladores, conductores y otros componentes de la red eléctrica, lo que puede derivar en defectos del sistema, interrupciones del servicio y cortes de energía que afectan a usuarios residenciales, comerciales e industriales.

Riesgos para la seguridad: El mal funcionamiento de un aislador no solo compromete la operatividad del sistema eléctrico, sino que también puede representar un peligro para las personas. Por ejemplo, si un aislador roto permite que un conductor energizado caiga al suelo, puede haber riesgo de electrocución o incluso de incendios en la vegetación cercana. Asimismo, las descargas eléctricas provocados por aisladores en mal estado pueden generar explosiones o incendios, poniendo en peligro tanto a los trabajadores de las empresas eléctricas como a la población en general.

Impacto medioambiental: Los aisladores defectuosos también pueden tener consecuencias negativas en el medio ambiente. La corriente de fuga puede generar calentamiento localizado, lo que en algunos casos puede liberar gases tóxicos o subproductos peligrosos que afectan la calidad del aire y la salud pública. Además, el desprendimiento de materiales aislantes, como vidrio, porcelana o compuestos sintéticos, puede contribuir a la contaminación y la acumulación de residuos en el entorno.(Monemi et al., 2024)

Dado que los aisladores defectuosos pueden generar pérdidas de energía, reducir la eficiencia del sistema, dañar equipos, provocar interrupciones del servicio, representar un riesgo para la seguridad y afectar el medio ambiente, es fundamental desarrollar métodos eficaces para su detección.

2.2 Métodos convencionales de detección de defectos en aisladores eléctricos

La detección de defectos en aisladores constituye un aspecto crítico en el mantenimiento de los sistemas eléctricos, ya que su deterioro puede derivar en interrupciones del servicio y costos elevados de reparación y mantenimiento. En este contexto, la identificación temprana de defectos ha evolucionado significativamente, permitiendo mejorar la eficiencia operativa, la seguridad del personal y la optimización de recursos económicos.

Tradicionalmente, la inspección de defectos en aisladores se realizaba mediante revisiones programadas por técnicos o personal de mantenimiento, lo cual presentaba diversas limitaciones, tales como la subjetividad en la evaluación, el riesgo inherente a la labor en altura y la falta de inmediatez en la detección de anomalías. No obstante, los avances tecnológicos han permitido el desarrollo de metodologías más eficientes y seguras para la detección y diagnóstico de defectos en aisladores.

En particular, en China se ha implementado el uso de UAV equipados con cámaras de alta resolución para la inspección de líneas de transmisión, permitiendo la obtención y el análisis inmediato de imágenes. Para el procesamiento y evaluación de estas imágenes, se han desarrollado diversos métodos, entre los que destacan la siguientes: segmentación por umbral y corrosión morfológica, la segmentación basada en entropía máxima y el análisis de textura.

Si bien estos algoritmos han demostrado ser efectivos en condiciones óptimas de iluminación y calidad de imagen, su desempeño se ve afectado por factores medioambientales adversos, lo que puede comprometer la precisión en la detección de defectos. Por tanto, resulta fundamental el desarrollo de estrategias que permitan mitigar dichas limitaciones y mejorar la confiabilidad de los sistemas de inspección automatizada.

2.2.1 Inspección visual

La inspección visual es el método más utilizado para la detección de defectos en aisladores. Consiste en examinar físicamente los aisladores en busca de signos de daño, como grietas, desprendimientos, decoloración o contaminación. Esta inspección puede realizarse desde el suelo utilizando binoculares, telescopios o cámaras, o bien mediante una evaluación más cercana por parte del personal técnico, quienes pueden acceder a las estructuras con escaleras, camiones con canastilla o incluso UAV.

Tabla 2.1 Ventajas y desventajas de la inspección visual. Fuente: (Han et al., 2009)

Ventajas y desventajas en inspección visual	
Ventajas	1- Bajo costo y fácil implementación 2- No requiere equipo sofisticado ni capacitación especializada
Desventajas	1- Subjetivo y propenso a errores humanos 2- Requiere mucho tiempo y mano de obra 3- Efectividad limitada para detectar defectos internos o defectos menores no visibles

2.2.2 Termografía infrarroja

La termografía infrarroja es un método de inspección sin contacto que permite detectar patrones térmicos en los aisladores. Los aisladores defectuosos suelen presentar distribuciones anómalas de temperatura debido al aumento de la corriente de fuga o a defectos internos. Mediante la captura de imágenes infrarrojas, los técnicos pueden identificar puntos calientes, los cuales pueden indicar posibles defectos o áreas que requieren una evaluación más detallada.

Tabla 2.2 Ventajas y desventajas de la termografía infrarroja. Fuente: (Han et al., 2009)

Ventajas y desventajas en termografía infrarroja	
Ventajas	1- Método sin contacto y no destructivo 2- Capaz de detectar defectos ocultos o internos 3- Proporciona información en tiempo real
Desventajas	1- Requiere equipo especializado y capacitación 2- Puede verse afectado por factores externos, como la temperatura ambiente y la radiación solar 3- No siempre detecta todos los tipos de defectos

2.3 Aprendizaje profundo para la detección de objetos

El aprendizaje profundo es un subcampo del aprendizaje automático que ha evolucionado significativamente en los últimos años, revolucionando áreas como la visión por computadora, el procesamiento del lenguaje natural y el reconocimiento de voz (LeCun et al., 2015).

Las bases del aprendizaje profundo se establecieron con los primeros modelos de redes neuronales artificiales, los cuales han sido perfeccionados a lo largo del tiempo. Los avances recientes en hardware, algoritmos de optimización y acceso a grandes volúmenes de datos han permitido un desarrollo sin precedentes en esta área (Schmidhuber, 2015).

El perceptrón, introducido por Frank Rosenblatt, marcó un hito al implementar conceptos fundamentales como los pesos y las funciones de activación. En paralelo, Widrow y Hoff desarrollaron el modelo ADALINE, que utilizaba el algoritmo cuadrados mínimos medios para la actualización de pesos. Aunque estos modelos tenían limitaciones, sentaron las bases para las redes neuronales actuales (Goodfellow et al., 2016).

El desarrollo del perceptrón multicapa y la introducción del algoritmo de retro propagación fueron avances clave en la década de 1980. Este algoritmo, propuesto por Rumelhart permitió el entrenamiento eficiente de redes más profundas y contribuyó al renacimiento del interés en las redes neuronales (LeCun et al., 2015)

Las CNNs han sido fundamentales en la evolución del aprendizaje profundo, especialmente en tareas de visión por computadora. En 2012, Krizhevsky revolucionaron el campo con AlexNet, una arquitectura que demostró la capacidad de las CNNs para la clasificación de imágenes a gran escala. Desde entonces, modelos como ResNet y EfficientNet han optimizado la profundidad y eficiencia de las redes convolucionales (Tan & Le, 2019).

Las redes neuronales recurrentes han sido ampliamente utilizadas en tareas de procesamiento de secuencias, como el reconocimiento de voz y el análisis de texto. Sin embargo, su limitación principal ha sido el problema del gradiente descendiente, lo que llevó al desarrollo de la arquitectura Long Short-Term Memory (LSTM). Modelos más recientes, como los transformadores, han superado muchas de estas limitaciones y han mejorado significativamente el rendimiento en tareas de modelado del lenguaje. A partir de la década de 2010, el auge del aprendizaje profundo ha sido impulsado por la disponibilidad de grandes volúmenes de datos, la mejora de hardware con unidades de procesamiento gráfico y el refinamiento de los algoritmos de optimización. En 2017, Google presentó un transformador, una arquitectura que revolucionó el procesamiento del lenguaje natural y dio lugar a modelos como BERT y GPT (Dosovitskiy et al., 2020).

2.4 Aplicaciones en la detección de objetos

La detección de objetos basada en aprendizaje profundo desempeña un papel crucial en el desarrollo de vehículos autónomos. En los automóviles sin conductor, la identificación y localización en tiempo real de otros vehículos, peatones y obstáculos son esenciales para una navegación segura. Las CNNs y arquitecturas como YOLO (You Only Look Once) han mejorado significativamente la detección de objetos en este campo, permitiendo un reconocimiento más rápido y preciso (Redmon, Farhadi, 2018).

El aprendizaje profundo ha revolucionado el análisis de videos en vigilancia y seguridad. Las técnicas de detección de objetos facilitan la identificación y el seguimiento de personas, vehículos y otros elementos de interés, mejorando el monitoreo de seguridad y permitiendo alertas en tiempo real ante actividades sospechosas (Zhao et al., 2019). Métodos avanzados como Faster R-CNN y Single Shot MultiBox Detector (SSD) se han implementado ampliamente en sistemas de seguridad modernos (Liu et al., 2016)

Las técnicas de detección de objetos han sido adoptadas en la imagenología médica para detectar y localizar enfermedades, tumores y otras condiciones en diversas modalidades, como radiografías, resonancias magnéticas y tomografías computarizadas. Los modelos de aprendizaje profundo, en particular las CNNs, han mejorado la precisión diagnóstica, facilitando la detección temprana y la planificación de tratamientos personalizados (Litjens et al., 2017). Los desarrollos recientes en el análisis de imágenes médicas con aprendizaje profundo han superado a los métodos tradicionales, ofreciendo mayor sensibilidad y especificidad (Esteve et al., 2017).

En el sector agrícola, la detección de objetos ayuda en la identificación y el conteo de cultivos, la detección de plagas y enfermedades, y la evaluación de la salud de las plantas. Modelos de aprendizaje profundo como Faster R-CNN y YOLO se han utilizado para automatizar estos procesos, contribuyendo a una mejor gestión de los cultivos y a la predicción del rendimiento (Kamilaris & Prenafeta-Boldú, 2018). La integración de imágenes obtenidas mediante UAV con modelos de aprendizaje profundo ha optimizado aún más el monitoreo agrícola a gran escala (McCool et al., 2017).

La detección de objetos basada en aprendizaje profundo se utiliza cada vez más en el comercio minorista y el comercio electrónico para el reconocimiento automatizado de productos en imágenes y videos. Esto permite una gestión eficiente del inventario y mejora la experiencia del cliente mediante la búsqueda visual y los sistemas de recomendación (Borji et al., 2019). Sistemas como Amazon Go emplean modelos de aprendizaje profundo para compras sin cajero, mejorando la eficiencia y la experiencia del usuario (Wankhede et al., 2018).

En la industria manufacturera, la detección de objetos se emplea ampliamente en el control de calidad y la detección de defectos. Los modelos de aprendizaje profundo pueden identificar defectos en los productos con alta precisión, asegurando que solo los artículos de calidad lleguen al consumidor (Du et al., 2016). La inspección visual automatizada con CNNs se ha convertido en un enfoque estándar en las líneas de producción modernas (Chen et al., 2021).

En el campo de la robótica, la detección de objetos es fundamental para el reconocimiento, la manipulación y la navegación de objetos. Los modelos de aprendizaje profundo permiten a los robots percibir y comprender su entorno, facilitando interacciones complejas y la toma de decisiones precisas. Las CNN y las arquitecturas basadas en transformadores han contribuido a mejoras significativas en la percepción y la capacidad de agarre de los robots (Levine et al., 2018).

La detección de objetos basada en aprendizaje profundo ha sido implementada con éxito en múltiples dominios, mejorando la precisión y la eficiencia en diversas tareas. Los avances en modelos de aprendizaje profundo, particularmente en CNNs y el marco YOLO, han permitido la detección de objetos en tiempo real, lo que hace que estas técnicas sean ideales para aplicaciones que requieren una toma de decisiones rápida y adaptable (Redmon, Farhadi, 2018).

2.5 Redes neuronales

2.5.1 Redes neuronales convolucionales (CNNs)

Las CNNs son modelos avanzados de aprendizaje profundo diseñados específicamente para tareas de visión computacional, como el reconocimiento de imágenes y la detección de objetos. Su funcionamiento se inspira en el procesamiento visual humano, y están compuestas por capas convolucionales, de agrupamiento (pooling) y completamente conectadas, que trabajan en conjunto para analizar y procesar imágenes. Las capas convolucionales utilizan filtros para identificar patrones locales en las imágenes, mientras que las capas de agrupamiento reducen las dimensiones espaciales, lo que disminuye la complejidad computacional y mejora la invariancia ante traslaciones. Posteriormente, las capas completamente conectadas realizan razonamiento de alto nivel y clasificación basada en las características extraídas. Las CNNs han logrado un notable éxito en diversos campos, convirtiéndose en una de las opciones más populares para tareas de aprendizaje automático relacionadas con imágenes (LeCun et al., 2015).

2.5.2 Conceptos básicos y arquitectura

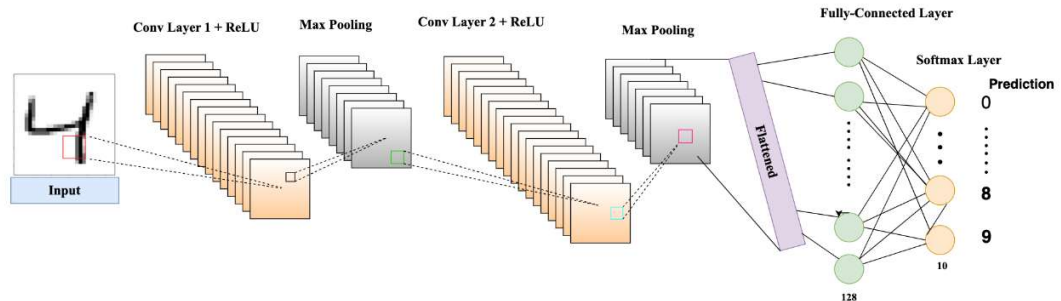


Figura 2.2 Arquitectura de Redes Neuronales Convolucionales (CNN) (Dhakal, 2024).

Las CNNs están diseñadas para aprender y extraer automáticamente características jerárquicas de las imágenes de entrada mediante capas interconectadas, como se muestra en la figura 2.2. La arquitectura básica de una CNN consta de una capa de entrada, múltiples capas alternadas de convolución y agrupamiento (pooling), y una o más capas completamente conectadas seguidas de una capa de salida. La capa de entrada toma los datos de la imagen en bruto, mientras que la capa de salida proporciona probabilidades de clase u otros resultados deseados basados en las características extraídas a lo largo de la red (LeCun et al., 2015).

2.5.3 Componentes claves y capas

Los componentes claves de una CNN se describen como sigue:

- **Capas de convolución:** Realizan operaciones de convolución utilizando filtros (kernels) para aprender y detectar patrones o características locales dentro de las imágenes de entrada. Los filtros se deslizan sobre la imagen de entrada, generando un conjunto de mapas de características.
- **Funciones de activación:** Introducen no linealidad en la red, permitiéndole aprender patrones más complejos. Las funciones de activación comunes incluyen ReLU, sigmoide y tanh.
- **Capas de agrupamiento (pooling):** Reducen las dimensiones espaciales de los mapas de características, disminuyendo el número de parámetros y la complejidad computacional, mejorando también la invariancia a la traducción. Los métodos de agrupamiento estándar son el max pooling y el average pooling.
- **Capas completamente conectadas:** Aplanan la salida de las capas de convolución y agrupamiento en un solo vector, que luego es procesado por una o más capas completamente conectadas para tareas de razonamiento de alto nivel y clasificación.

- Capa de salida: La capa final en una CNN, que generalmente utiliza una función de activación softmax para proporcionar probabilidades para cada clase en un problema de clasificación multiclase.

2.5.4 Arquitecturas populares de CNNs

Existen diversas arquitecturas de CNNs que han logrado resultados de vanguardia en diversas tareas de visión por computadora. Algunas arquitecturas populares de CNNs incluyen:

LeNet-5: Una de las primeras arquitecturas de CNN diseñadas por Yann LeCun en 1998, utilizada principalmente para el reconocimiento de dígitos escritos a mano. (Figura 2.3)

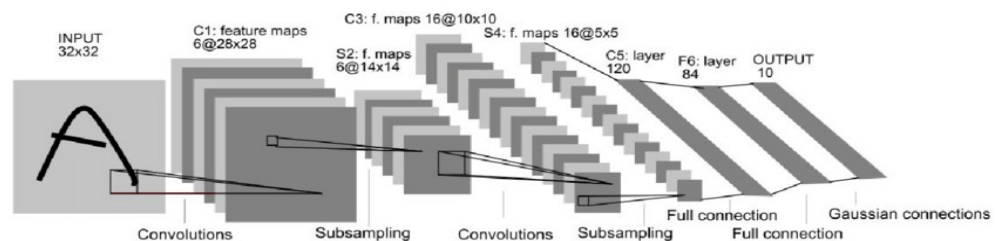


Figura 2.3 Arquitectura LeNet-5 (Fuente: <https://goo.su/3BwOK>)

AlexNet: Desarrollada por Alex Krizhevsky, Ilya Sutskever y Geoffrey Hinton en 2012, superó significativamente a otros modelos en el ImageNet Large Scale Visual Recognition Challenge (ILSVRC) y generó un gran interés en el aprendizaje profundo. (Figura 2.4)

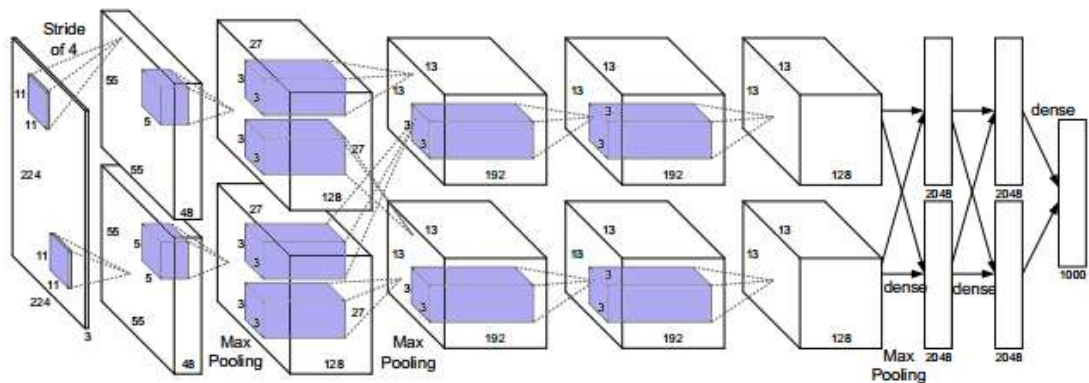


Figura 2.4 Arquitectura AlexNet (Fuente: <https://goo.su/NRYz2Lt>).

VGGNet: Creada por el Visual Geometry Group de la Universidad de Oxford en 2014, conocida por su arquitectura más profunda y filtros convolucionales pequeños. (Figura 2.5)

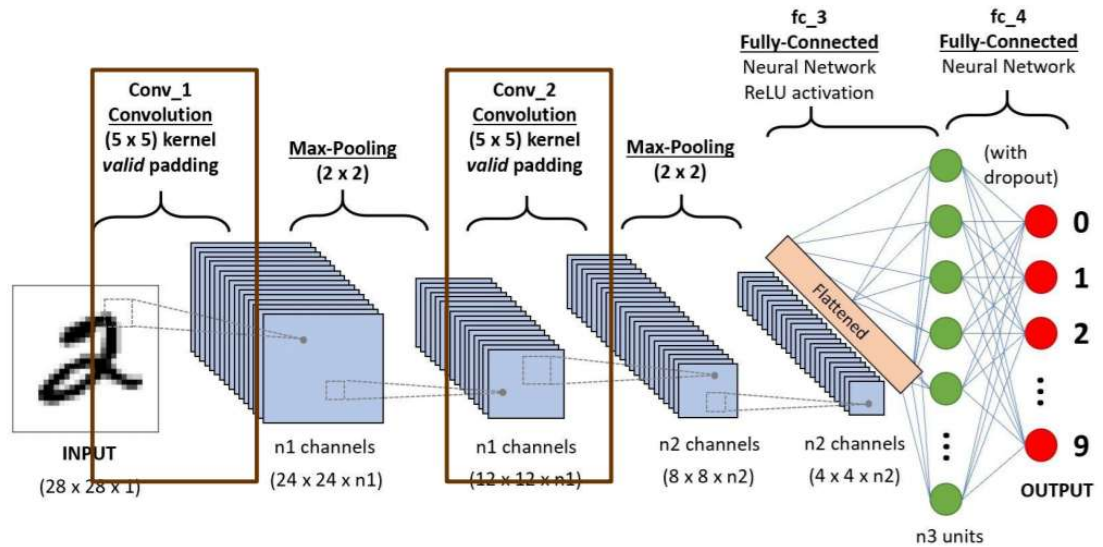


Figura 2.5 Arquitectura VGGNet (Fuente: <https://goo.su/dSMm>).

Inception (GoogLeNet): Introducida por los investigadores de Google en 2014, cuenta con el módulo Inception, que ayuda a reducir el número de parámetros y la complejidad computacional. (Figura 2.6)

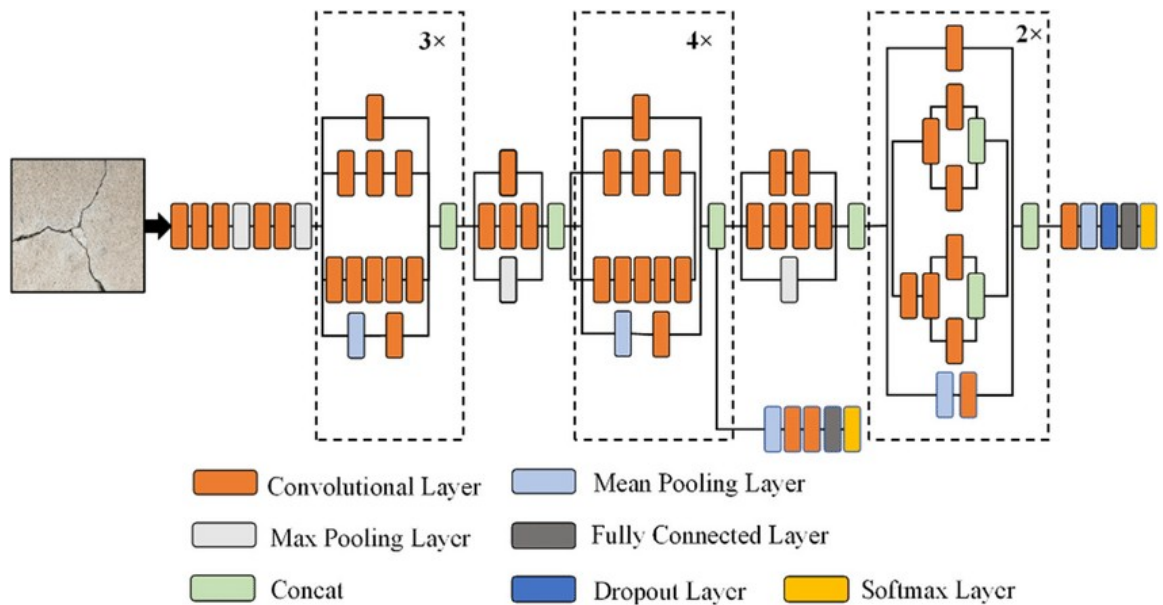


Figura 2.6 Arquitectura Inception (Fuente: <https://goo.su/1pzFWG>)

ResNet: Desarrollada por Microsoft Research en 2015, introdujo conexiones residuales (conexiones de salto) para permitir el entrenamiento de redes profundas, abordando de manera efectiva el problema del gradiente desvanecido. (Figura 2.7)

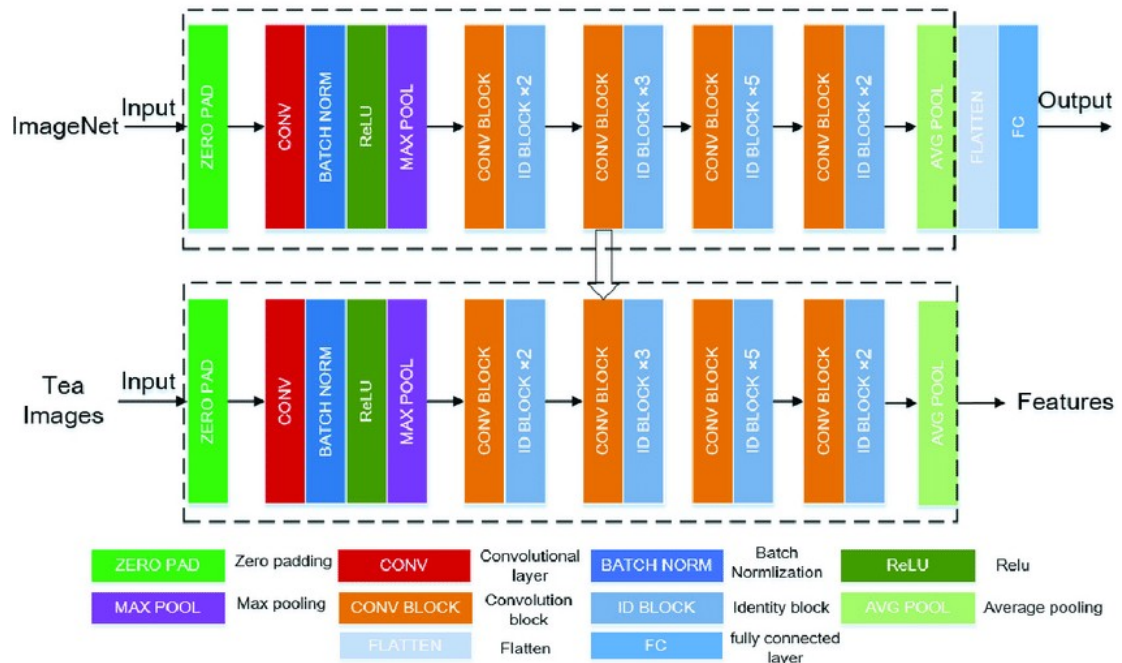


Figura 2.7 Arquitectura ResNet (Fuente: <https://goo.su/V1rwReh>)

2.6 Arquitectura YOLO (You Only Look Once)

2.6.1 Descripción de la arquitectura

YOLO (You Only Look Once) es una arquitectura popular de detección de objetos en el campo de la visión por computadora. Fue introducido por primera vez por Joseph Redmon, Santosh Divvala, Ross Girshick y Ali Farhadi en su artículo de 2016 titulado "You Only Look Once: Unified, Real-Time Object Detection". La principal ventaja de YOLO sobre otros métodos de detección de objetos es su capacidad para lograr un rendimiento en tiempo real manteniendo un alto nivel de precisión (Redmon, Farhadi, 2018)

La arquitectura YOLO se basa en la idea de dividir una imagen de entrada en una cuadrícula (por ejemplo, 7x7, 13x13 o 19x19). Cada celda de la cuadrícula es responsable de predecir un número fijo de cuadros delimitadores alrededor de los objetos, junto con las probabilidades de clase asociadas a cada objeto detectado. El modelo luego combina estas predicciones para generar la salida final, que consiste en los objetos detectados, sus ubicaciones y sus etiquetas de clase.

La arquitectura de YOLO comprende una única CNN que procesa la imagen de entrada y predice simultáneamente los cuadros delimitadores y las probabilidades de clase. Esto difiere de los métodos tradicionales de detección de objetos, que a menudo utilizan redes separadas para la localización de objetos y la clasificación o requieren múltiples pasadas a través de la imagen (Redmon, Farhadi, 2018).

2.6.2 Visión general y evolución

YOLO (You Only Look Once) es un sistema de detección de objetos en tiempo real que utiliza una única red neuronal convolucional para identificar objetos en imágenes. Desde su introducción en 2015, ha experimentado varias iteraciones, cada una mejorando en precisión y eficiencia (Redmon, Farhadi, 2018).

YOLOv1 (2015): Introdujo el concepto de un modelo unificado para la detección de objetos, dividiendo la imagen en una cuadrícula y prediciendo cuadros delimitadores y probabilidades de clase para cada celda (Redmon et al., 2016).

YOLOv2 (2016): Mejoró la precisión y velocidad al incorporar normalización por lotes, anclajes para predicción de cuadros y entrenamiento multiescala, permitiendo la detección de más de 9000 categorías de objetos (Redmon et al., 2016).

YOLOv3 (2018): Utilizó una red base más profunda (Darknet-53) y predicciones multiescala para mejorar la exactitud, manteniendo la velocidad en tiempo real (Redmon, Farhadi, 2018).

YOLOv4 (2020): Introdujo mejoras arquitectónicas y de entrenamiento, como la red base CSPNet y la función de activación Mish, logrando un equilibrio óptimo entre velocidad y precisión (Bochkovskiy et al., 2020).

YOLOv5 (2020): Aunque no es una versión oficial, incorporó mejoras como el uso de la red base EfficientDet y ofreció diferentes tamaños de modelo para equilibrar velocidad y precisión (Bochkovskiy et al., 2020).

YOLOv6 (2022): Desarrollado por Meituan para aplicaciones industriales, enfocándose en eficiencia y rendimiento en entornos reales (Li et al., 2022).

YOLOv7 (2022): Introdujo técnicas avanzadas de entrenamiento y mejoras en la arquitectura, estableciendo nuevos estándares en detección en tiempo real (Wang et al., 2023).

YOLOv8 (2023): Añadió funcionalidades como segmentación y seguimiento de objetos, consolidándose como una solución integral para tareas de visión por computadora (Terven et al., 2023).

Tabla 2.3 Evolución de YOLO Fuente: (Terven et al., 2023).

Versión	Año	Principal contribuyente	Mejoras y características clave
YOLOv1	2016	Joseph Redmon	- Primera versión de YOLO - Detección de objetos en tiempo real - Modelo unificado para detección y clasificación
YOLOv2	2017	Joseph Redmon	- Mayor precisión - Mejor recuperación y localización - Introducción de cajas anda (anchor boxes) - Uso de normalización por lotes (batch normalization)
YOLOv3	2018	Joseph Redmon	- Más preciso y más rápido - Uso de predicciones en múltiples escalas - Mejor detección de objetos pequeños - Arquitectura de backbone Darknet-53
YOLOv4	2020	Alexey Bochkovskiy	- Precisión de última generación - Optimizado para velocidad y eficiencia - Introducción de CSPNet, PANet y SAM - Mejoras en la aumentación de datos (Mosaic)
YOLOv5	2020	Glenn Jocher	- Más fácil de entrenar y desplegar - Implementación en PyTorch - Arquitectura de modelo mejorada - Mayor velocidad y precisión
YOLOv6	2022	Meituan	- Enfocado en aplicaciones industriales - Mayor eficiencia en el entrenamiento - Arquitectura más rápida y ligera
YOLOv7	2022	Wang et al.	- Optimizado para rendimiento en tiempo real - Más preciso y eficiente que versiones anteriores - Nuevas técnicas de entrenamiento y escalado del modelo
YOLOv8	2023	Ultralytics	- Soporta detección de objetos, segmentación y seguimiento - Mejor precisión y adaptabilidad - Opciones mejoradas de escalado del modelo

2.6.3 Características clave y ventajas

Velocidad: El enfoque de una sola pasada de YOLO lo hace significativamente más rápido que muchos otros marcos de detección de objetos, lo que lo hace adecuado para aplicaciones en tiempo real (Redmon et al., 2016).

Aprendizaje de extremo a extremo: Al unificar el proceso de propuesta de regiones y clasificación, YOLO simplifica el proceso de entrenamiento y permite el aprendizaje simultáneo de detección y localización de objetos (Redmon, Farhadi, 2018).

Conciencia del contexto: YOLO procesa la imagen completa de una sola vez, capturando implícitamente información contextual sobre los objetos y su entorno. Esto resulta en una mejor comprensión de la escena y una detección de objetos más precisa (Bochkovskiy et al., 2020).

Evolución y adaptabilidad: Las diferentes versiones de YOLO demuestran la adaptabilidad y evolución del marco, que ha mejorado constantemente en precisión y velocidad a lo largo del tiempo (Hussain, 2023).

2.6.4 Arquitectura YOLOv8

YOLOv8 es la última versión de la serie de modelos YOLO, desarrollada por Ultralytics, que amplía las capacidades de detección de objetos al incorporar tareas como la segmentación de instancias y la estimación de poses. Esta versión introduce mejoras arquitectónicas significativas que optimizan tanto la precisión como la eficiencia del modelo (Sohan et al., 2024).

2.6.5 Diseño del modelo y capas

Backbone: YOLOv8 utiliza una versión mejorada de CSPDarknet como su red troncal. CSPDarknet optimiza la extracción de características mediante una partición eficiente de mapas de características, reduciendo la redundancia computacional sin comprometer la precisión (Wang et al., 2020).

Neck: En lugar de la arquitectura tradicional, YOLOv8 incorpora el módulo C2f (Cross-Stage Partial Fusion), que mejora la fusión de características y la detección de objetos a múltiples escalas. Este diseño permite una mejor propagación de la información y una mayor eficiencia en el uso de la memoria del modelo (Qu et al., 2025)

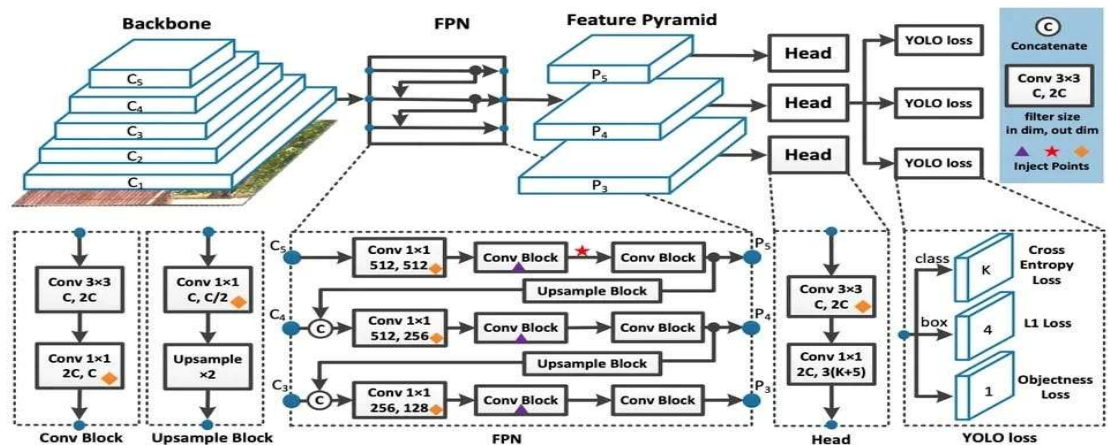


Figura 2.8 Arquitectura YOLOv8 (Fuente: <https://goo.su/869w9>)

Head: La cabeza del modelo está diseñada para predecir cuadros delimitadores, puntuaciones de confianza y probabilidades de clase (Figura 2.8). Además, YOLOv8 introduce cabezas adicionales para tareas como la segmentación de instancias y la estimación de poses, ampliando su aplicabilidad en diversas tareas de visión por computadora (Qadri et al., 2023)

Funciones de Activación: El modelo emplea funciones de activación avanzadas como SiLU (Swish Linear Unit) para mejorar la capacidad de aprendizaje y la precisión en las predicciones. La función SiLU ha demostrado mejorar la estabilidad del entrenamiento en redes profundas al mitigar problemas como la saturación de gradientes (Elfwing et al., 2018).

Escalado del Modelo: Al igual que versiones anteriores, YOLOv8 ofrece múltiples tamaños de modelo (p. ej., nano, pequeño, mediano, grande y extragrande) para equilibrar la velocidad y la precisión según las necesidades específicas de la aplicación. Esta escalabilidad permite su implementación en hardware diverso, desde dispositivos móviles hasta servidores de alto rendimiento (Qu et al., 2025).

2.7 Metodologías para minería de datos

A finales del siglo pasado, con el creciente uso de técnicas de minería de datos en la industria para extraer conocimiento, algunas empresas e instituciones reconocieron la necesidad de colaborar para identificar buenas prácticas y errores comunes a partir de sus experiencias pasadas. Con el apoyo financiero de la Unión Europea, estas entidades se unieron con este propósito. La diversidad de los datos ha aumentado considerablemente en términos de origen, formato y modalidades, y, a su vez, la variedad de técnicas provenientes de áreas como el aprendizaje automático, la gestión de datos, la visualización y la inferencia causal. (Martínez-Plumed et al., 2019).

Fayyad, Piatetsky-Shapiro y Smyth definen el “Knowledge Discovery in Databases” (KDD) como el proceso global de descubrimiento de conocimiento a partir de los datos, que incluye el almacenamiento y acceso de los mismos, la escalabilidad de los algoritmos para trabajar con grandes volúmenes de datos de manera eficiente, la interpretación y visualización de los resultados, y la interacción entre humanos y máquinas. Por su parte, la minería de datos se considera un paso dentro de este proceso, en el cual los datos previamente procesados se transforman en patrones que pueden generar conocimiento útil y aplicable. (Martínez-Plumed et al., 2019)

El KDD consta de nueve pasos principales:

1. Entendimiento de la aplicación
2. Creación de un set de datos
3. Limpieza de los datos
4. Caracterización de los datos
5. Determinar el objetivo de la minería de datos.
6. Escoger un algoritmo de minería
7. Minería de datos
8. Interpretación de datos
9. Uso del conocimiento adquirido

El proceso KDD se volvió la base para muchas otras metodologías con todas las brechas y deficiencias que tenía el modelo original. Todo eso no hizo más que aumentar el contexto de su aplicación y cambiar por completo su enfoque, brindándole pasos complementarios que le dan importancia al punto de vista humano y resaltando las decisiones que deben ser hechas por el usuario en el transcurso del proceso de minería de datos. En la Figura 2.9 se muestra la evolución y alternativas del proceso KDD como lo fue el proceso Caberna (cita) que cambiaba en cantidad de pasos y dando relevancia al detallado, procesamiento y descubrimiento de datos. (Plotnikova et al., 2020).

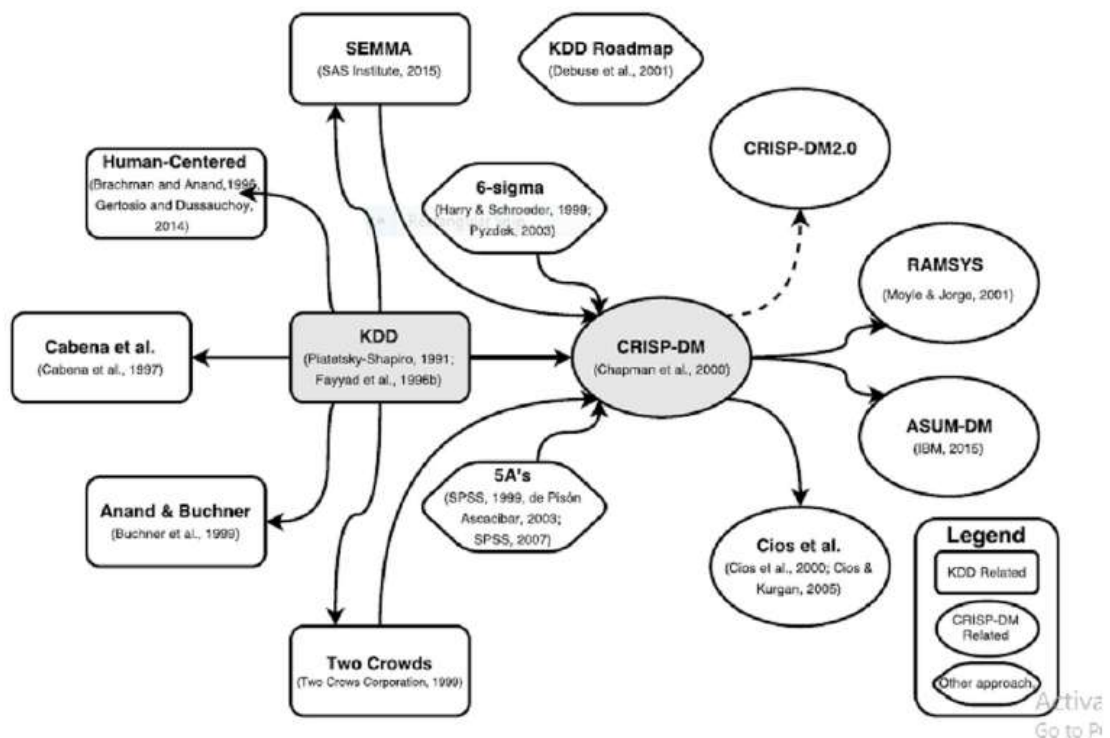


Figura 2.9 Evolución de los procesos y minería de datos (Plotnikova et al., 2020)

2.7.1 CRISP-DM

En los años 2000 se introdujo la primera versión de “Cross Industry Standard Process for Data Mining” CRISP-DM como una alternativa al KDD (Martínez-Plumed et al., 2019). El modelo CRISP-DM es un marco genérico para el proceso de minería de datos que ofrece una visión integral del ciclo de vida de los proyectos de minería de datos. Este modelo ha ganado gran popularidad tanto en la industria como en la academia, siendo considerado por diversas encuestas y estudios como el estándar de facto para los proyectos de descubrimiento de conocimiento y minería de datos. Entre sus principales ventajas se incluyen la reducción de costos y tiempos, así como la minimización de los requisitos de conocimiento previos para llevar a cabo proyectos de minería de datos. Además, la implementación del modelo facilita la capacitación, la transferencia de conocimientos, la documentación y la captura de mejores prácticas. (Plotnikova et al., 2020).

A lo largo de los años, CRISP-DM ha consolidado y ampliado el modelo original (KDD), y aunque está basado en KDD, se distingue por su enfoque iterativo, que consiste en seis fases, las cuales se muestran en la Figura 2.10, que se ejecutan de manera cíclica. Esta iteración continua marca la principal diferencia con el modelo KDD inicial, que asume una ejecución secuencial de sus pasos. Al igual que KDD, CRISP-DM tiene como objetivo proporcionar directrices claras para llevar a cabo

procesos de minería de datos sobre grandes volúmenes de datos, lo que lo convierte en una herramienta eficaz para la práctica del análisis y la explotación de información en diversos sectores.

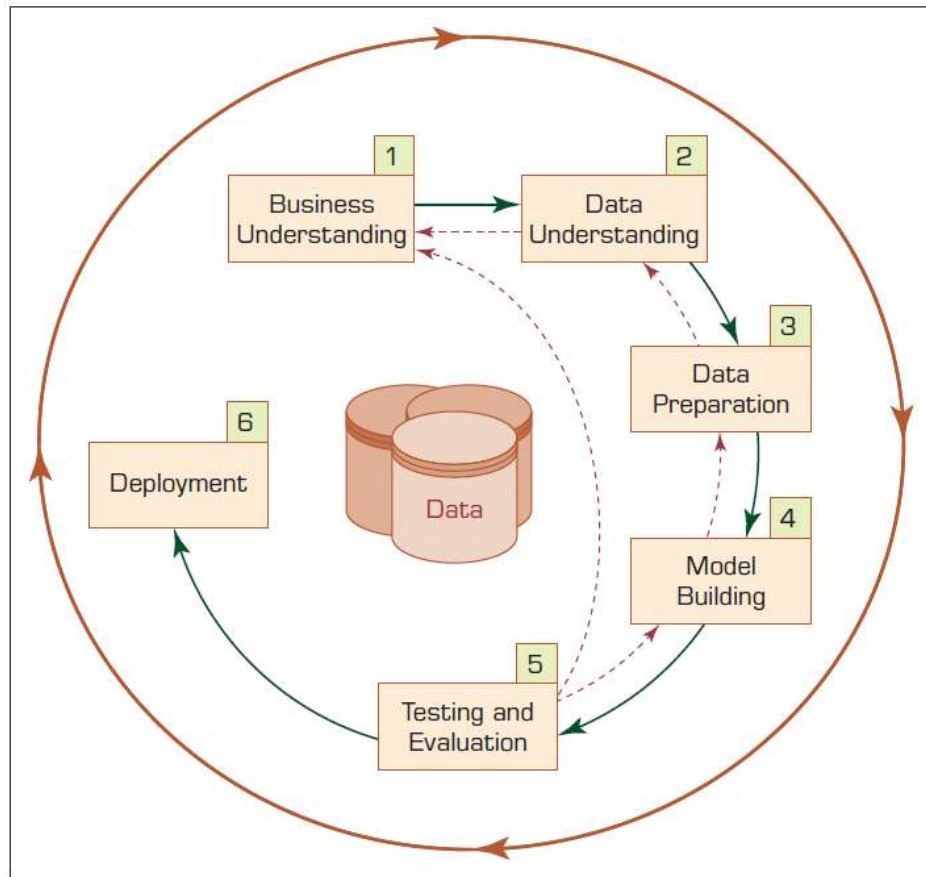


Figura 2.10 Elementos de la metodología (Sharda et al., 2018)

Paso 1. Entendimiento del negocio: El éxito de un estudio de minería de datos depende de entender claramente su propósito y definir objetivos específicos. Luego, se elabora un plan que asigna responsabilidades y establece un presupuesto inicial.

Paso 2. Entendimiento de la información: La minería de datos comienza con una identificación precisa de los datos relevantes según los objetivos comerciales. Esto incluye comprender las fuentes de datos y las variables necesarias, así como utilizar técnicas estadísticas para analizarlas. Las fuentes de datos incluyen información demográfica, sociográfica, transacciones comerciales, y en la actualidad, también redes sociales y datos generados por máquinas. Los datos pueden ser cuantitativos o cualitativos, y es esencial realizar un preprocesamiento de los datos antes de aplicar algoritmos de minería.

Paso 3. Preparación de la información: El preprocesamiento de datos es crucial para preparar los datos para el análisis en minería de datos. Representa la fase más demandante en términos de tiempo y esfuerzo, debido a datos incompletos, el ruido y la inconsistencia que suelen tener los datos reales.

Paso 4. Construcción del modelo: En la fase de modelado, se aplican diversas técnicas a los datos preparados para resolver una necesidad comercial específica. Dado que no existe un único método óptimo, es crucial probar varios modelos y ajustar los parámetros para obtener los mejores resultados. Algunos métodos pueden requerir modificaciones en los datos, lo que puede implicar volver a la etapa de preparación.

Paso 5. Pruebas y evaluación: En la fase de evaluación, se mide qué tan bien los modelos cumplen con los objetivos comerciales y si es necesario desarrollar más modelos. Los resultados pueden revelar hallazgos adicionales que también aporten valor. La interpretación de estos patrones de conocimiento depende de la colaboración entre analistas de datos y tomadores de decisiones, utilizando herramientas de visualización y tabulación para facilitar el entendimiento.

Paso 6. Implementación: El proyecto de minería de datos no termina con el desarrollo de los modelos; estos deben ser presentados de forma comprensible para el usuario final y, en algunos casos, implementados a nivel empresarial. La fase de implementación puede incluir también el mantenimiento de los modelos, ya que los datos y el negocio cambian constantemente, lo que puede hacer que los modelos se vuelvan obsoletos. Es necesario un plan de monitoreo y mantenimiento para garantizar que los resultados sigan siendo válidos y útiles.

De esta manera, los fundamentos sobre las arquitecturas de detección de objetos y los enfoques metodológicos de minería de datos permiten contar con una base sólida para el desarrollo de la propuesta. Estos elementos no solo facilitan la comprensión del funcionamiento de modelos como YOLOv8, sino que también orientan el proceso de preparación, análisis y validación de la información, lo que resulta clave para aplicar las técnicas de aprendizaje profundo en la detección de defectos en aisladores eléctricos.

CAPÍTULO 3 MATERIALES Y MÉTODOS

3.1 Implementación metodología CRISP-DM

3.1.1 Entendimiento del negocio

El presente proyecto surge de la necesidad de mejorar el mantenimiento predictivo en los aisladores eléctricos en las líneas de distribución de energía eléctrica dentro de AES El Salvador, una empresa clave en el sector eléctrico del país. Esta compañía gestiona un amplio portafolio de infraestructuras, entre las que se encuentran líneas de distribución de alta tensión. Para asegurar su correcto funcionamiento, se llevan a cabo inspecciones aéreas mediante UAV, los cuales capturan imágenes detalladas del estado de los componentes eléctricos.

Actualmente, la evaluación de estas imágenes se realiza de manera manual por parte de especialistas en mantenimiento siguiendo el procedimiento descrito en la figura 3.1, quienes analizan cada fotografía con el fin de identificar posibles anomalías en los aisladores y otros elementos de la red. Sin embargo, este proceso demanda una cantidad considerable de tiempo y recursos, lo que reduce la eficiencia operativa y la capacidad de respuesta ante defectos.

Por ello, este proyecto busca alinearse con los objetivos estratégicos de AES El Salvador, promoviendo el uso de tecnologías avanzadas para optimizar los procesos de inspección, reducir costos operativos y mejorar la confiabilidad de la infraestructura eléctrica.

Con el propósito de entender a profundidad la operativa actual del mantenimiento de las líneas de transmisión en AES El Salvador, se realizaron reuniones con expertos del área técnica. A través de estas sesiones, se logró recopilar información clave sobre la metodología empleada para la captura, almacenamiento y análisis de imágenes utilizadas en las inspecciones.

Uno de los principales desafíos identificados fue la dependencia del análisis manual, lo que implica un alto consumo de tiempo en la generación de reportes y diagnóstico de defectos. Adicionalmente, se constató que las imágenes obtenidas no solo son útiles para detectar defectos en los aisladores, sino que también permiten evaluar otros factores que influyen en la fiabilidad del sistema eléctrico, tales como:

- Condiciones del terreno donde se encuentran las estructuras de soporte.
- Crecimiento de vegetación en áreas cercanas a las líneas de transmisión.
- Presencia de corrosión en estructuras metálicas y herrajes.
- Estado de los pararrayos, conductores y sistemas de puesta a tierra.

Aunque el enfoque principal de este proyecto es la detección automatizada de defectos en aisladores eléctricos, el análisis realizado reveló que la aplicación de algoritmos de visión artificial podría extenderse a otros aspectos del mantenimiento de la red de transmisión. Esto permitiría a AES

El Salvador optimizar sus procesos, reduciendo los tiempos de revisión, mejorando la capacidad de respuesta y maximizando la eficiencia en la gestión de la infraestructura eléctrica.



Figura 3.1 Flujo actual del proceso de inspección de líneas de transmisión (Fuente: Elaboración propia)

3.1.2 Entendimiento de los datos

En esta etapa, el objetivo principal fue analizar en profundidad las imágenes obtenidas durante las inspecciones de mantenimiento de las líneas de transmisión de energía, las cuales fueron capturadas mediante UAV, las cuales se tomaron de un repositorio de datos públicos. Estas imágenes corresponden a diversas torres de transmisión y sirven como insumo clave para detectar posibles defectos en la red eléctrica.

Para garantizar una comprensión detallada, se evaluaron aspectos fundamentales de los datos, como el formato y la resolución de las imágenes, las condiciones de iluminación y el ángulo de captura. También se examinaron las características específicas de los elementos presentes en las imágenes, tales como torres, cables, aisladores y otros componentes eléctricos. Durante este proceso, se identificaron desafíos significativos, incluyendo variaciones en las condiciones climáticas y la presencia de elementos ajenos al análisis, lo que podría afectar la precisión del procesamiento.

Las imágenes utilizadas en este proyecto provienen de bases de datos en línea de acceso libre, lo que permitió su uso sin restricciones para el entrenamiento y evaluación del modelo.

Para mantener la consistencia del conjunto de datos y garantizar condiciones óptimas de visibilidad, la captura de imágenes mediante UAV se tiene que realizar en condiciones climáticas estables y sin presencia de lluvia o niebla, ya que la humedad y las precipitaciones pueden afectar tanto la calidad de la imagen como la seguridad operativa del equipo.

Asimismo, las capturas deben programarse en horarios con iluminación natural favorable, preferiblemente en las primeras horas de la mañana o en la tarde, evitando el mediodía, cuando la incidencia solar directa puede generar sombras intensas y sobreexposición.

3.1.3 Preparación de los datos

La preparación de los datos es una etapa fundamental en el desarrollo de sistemas de visión por computadora, ya que garantiza que el modelo pueda aprender de manera efectiva a partir de ejemplos representativos. En este proyecto, se llevó a cabo una preparación meticulosa del conjunto de datos utilizando la plataforma Roboflow, lo que facilitó la organización, anotación y optimización de las imágenes de aisladores eléctricos.

Recopilación de imágenes:

Se recopilaron un total de 2,150 imágenes de aisladores eléctricos en diversas condiciones operativas. Estas imágenes incluían tanto aisladores en buen estado como aquellos que presentaban diferentes tipos de defectos. Las imágenes fueron obtenidas de inspecciones de campo y bases de datos públicas, asegurando una variedad de escenarios y condiciones ambientales.

Anotación de imágenes:

Una vez cargadas las imágenes en Roboflow, se procedió a su anotación manual utilizando las herramientas de etiquetado proporcionadas por la plataforma. Se definieron las siguientes clases para la anotación, basadas en las etiquetas disponibles en el proyecto de Roboflow:

- Glassdirty: Aislador de vidrio con suciedad superficial.
- Glassloss: Aislador de vidrio con pérdida de material.
- Polymer: Aislador de material polimérico en buen estado.
- Polymerdirty: Aislador polimérico con suciedad.
- Two glass: Aislador compuesto por dos elementos de vidrio.
- Broken disc insulator: Aislador de disco roto.
- Pollution-flashover: Aislador con descarga por contaminación.
- Snow: Aislador cubierto de nieve.

Estas clases permitieron una categorización detallada de los diferentes tipos de defectos y condiciones presentes en los aisladores, lo que es esencial para el entrenamiento efectivo del modelo de detección.

3.1.4 Construcción del modelo

Para la construcción del modelo de detección de defectos en aisladores eléctricos, se utilizó la arquitectura YOLOv8, una de las más recientes y eficientes para tareas de detección de objetos. YOLOv8 se destaca por su velocidad y precisión, lo cual lo hace adecuado para aplicaciones en tiempo real y con recursos limitados.

El conjunto de datos anotado se exportó desde la plataforma Roboflow en formato compatible con YOLOv8. Luego, se configuró el entorno de entrenamiento utilizando SublimeText como entorno

de desarrollo integrado (IDE) y se seleccionaron los hiperparámetros más adecuados, como el número de épocas, tamaño de imagen, tasa de aprendizaje y el modelo base (YOLOv8n, versión liviana).

El entrenamiento del modelo se realizó desde cero, es decir, sin utilizar pesos previamente entrenados, con el objetivo de lograr que el modelo aprenda directamente a partir de las imágenes específicas de aisladores eléctricos. Este proceso permitió que el modelo se adaptara a las características particulares del conjunto de datos, incluyendo defectos como suciedad, pérdida de material, roturas y descargas por contaminación.

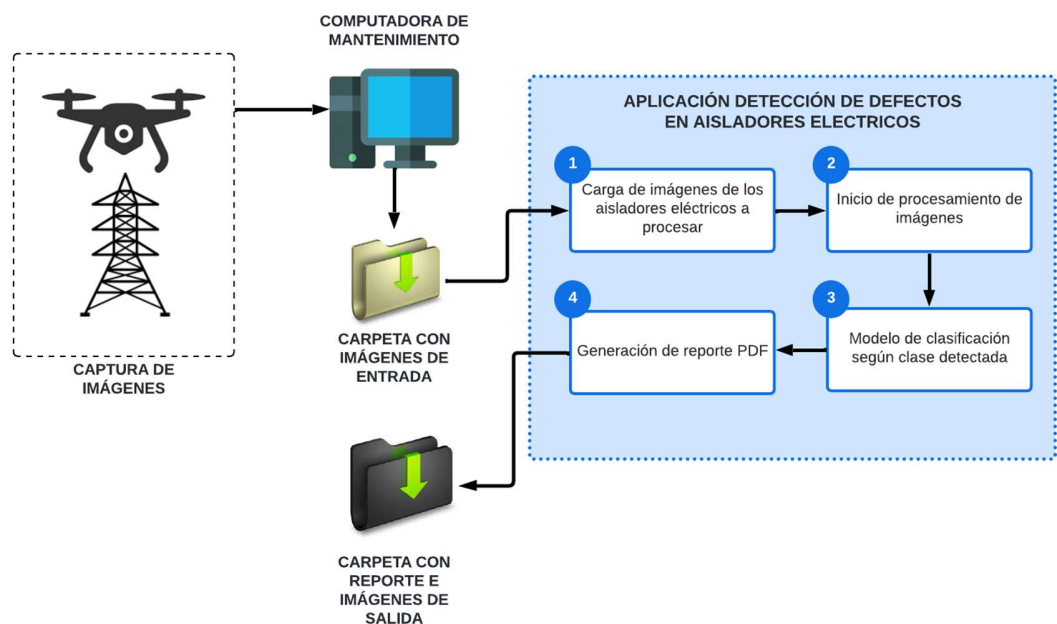


Figura 3.2 Arquitectura de la solución implementada (Fuente: Elaboración propia).

3.1.5 Prueba y evaluación del modelo

Una vez finalizado el entrenamiento, se evaluó el modelo utilizando subconjuntos de validación y prueba previamente definidos durante la preparación de datos. La evaluación se centró en medir la precisión del modelo en la detección de cada una de las clases de defectos de aisladores.

Las métricas principales utilizadas fueron:

- Precisión (Precision): porcentaje de predicciones correctas sobre el total de predicciones realizadas por el modelo.
- Recall: proporción de verdaderos positivos detectados sobre el total de elementos reales de una clase.
- mAP@0.5 (mean Average Precision a umbral 0.5): promedio de precisión considerando que se acepta una detección como correcta si la superposición (IoU) entre predicción y etiqueta real es mayor al 50%.

El modelo alcanzó una mAP@0.5 de 89.7%, lo que indica un buen desempeño en la identificación de objetos. Algunas clases como Glassdirty y Polymerdirty presentaron altos valores de precisión, mientras que clases más complejas como Pollution-flashover o Broken disc insulator mostraron un rendimiento ligeramente menor, debido a la similitud visual entre defectos y condiciones ambientales.

3.1.6 Despliegue del modelo

Luego de la evaluación, se procedió al despliegue del modelo para su integración en una aplicación práctica. El modelo fue exportado en formato YOLOv8 (.pt), lo que permitió su uso en una aplicación desarrollada en Python con la biblioteca Tkinter para la interfaz gráfica de usuario.

Esta aplicación permite al usuario cargar una imagen desde su computadora y visualizar los resultados del modelo, que incluyen las clases detectadas y la probabilidad de confianza para cada una. Además, la aplicación cuenta con una opción para generar un reporte en PDF, que incluye la imagen original, la imagen con detecciones, las métricas de evaluación y los datos generales del proyecto.

El despliegue se pensó con el objetivo de ser usado como herramienta de apoyo en inspecciones técnicas de aisladores eléctricos, ofreciendo una solución automatizada que puede ayudar a reducir tiempos de diagnóstico y mejorar la toma de decisiones en campo.

3.2 Recopilación y procesamiento de datos

3.2.1 Adquisición y anotación de imágenes

Este proceso se repitió para cada imagen, asegurando que todas las instancias relevantes fueran correctamente anotadas, tal como se muestra en la Figura 3.2.

División del conjunto de datos:

Finalmente, el conjunto de datos se dividió en tres subconjuntos:

1. Entrenamiento: 78.76% de las imágenes, utilizadas para entrenar el modelo.
2. Validación: 14.18% de las imágenes, utilizadas para ajustar los hiperparámetros y prevenir el sobreajuste.
3. Prueba: 7.06% de las imágenes, utilizadas para evaluar el rendimiento final del modelo.

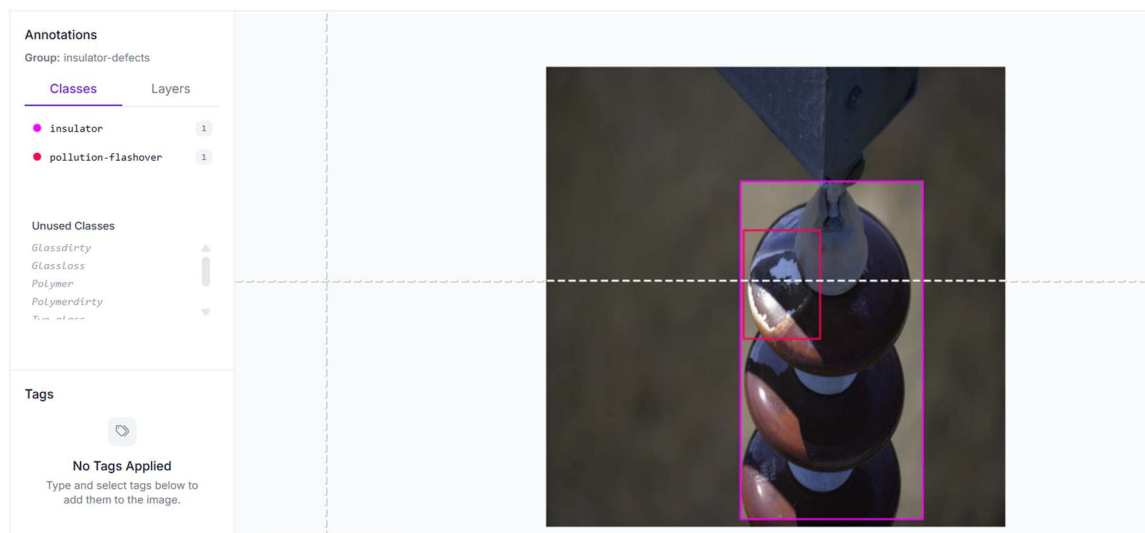


Figura 3.3 Imágenes ingresadas en el sistema (Fuente: Elaboración propia).

3.3 Implementación y configuración de modelos de detección de objetos

3.3.1 Selección de un modelo apropiado

La selección del modelo se basó en un análisis de varias arquitecturas populares de detección de objetos como SSD, Faster R-CNN y YOLO. Entre estas, YOLOv8 fue elegido por su balance entre rendimiento y facilidad de implementación. A diferencia de otras versiones anteriores, YOLOv8 presenta una arquitectura completamente nueva, sin anclas, lo que mejora la precisión y reduce la complejidad del entrenamiento.

Además, YOLOv8 incluye modelos preconfigurados en distintas escalas (nano, small, medium, large) que permiten ajustarse a diferentes capacidades de hardware. Para este proyecto, se utilizó la variante YOLOv8n (nano), adecuada para su despliegue en entornos con recursos limitados.

3.3.2 Personalización y entrenamiento de modelos

Luego de seleccionar YOLOv8, se procedió a personalizarlo para el conjunto de datos de aisladores eléctricos. Se definieron 8 clases correspondientes a diferentes condiciones y defectos de los aisladores. El modelo se entrenó desde cero, cargando las imágenes anotadas en formato YOLO a través de Roboflow, como se muestra en la figura 3.3

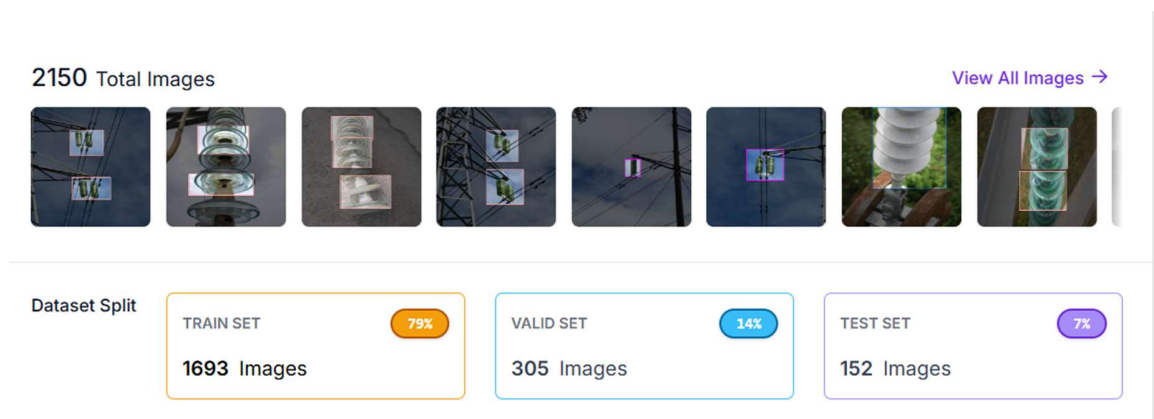


Figura 3.4 Distribución del dataset en conjuntos de entrenamiento, validación y prueba.

Cuando se tiene todo el etiquetado de las imágenes se procede a entrenar el modelo dentro de la aplicación de RoboFlow (Figura 3.4)

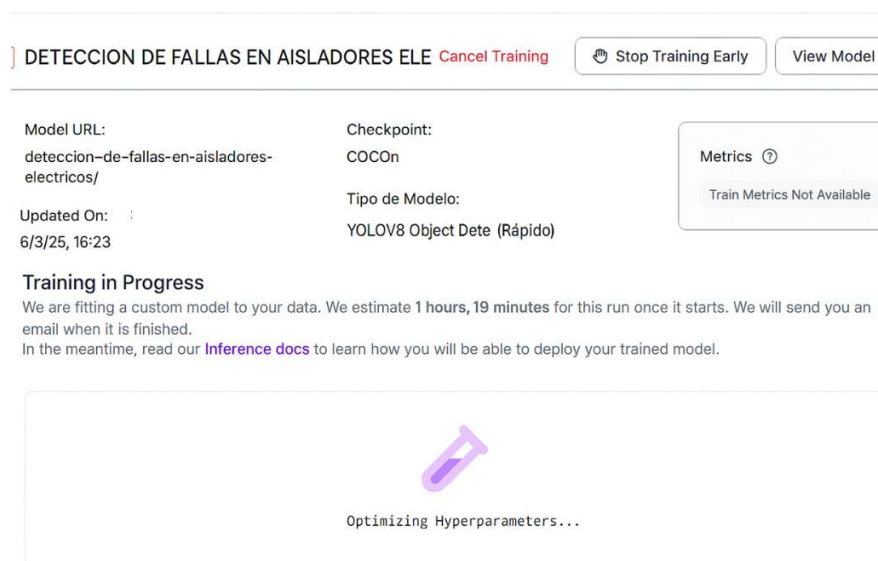


Figura 3.5 Interfaz del proceso de entrenamiento del modelo de detección de defectos en aisladores eléctricos.



Figura 3.6 Desempeño del modelo durante el entrenamiento.

Para evaluar el desempeño del modelo entrenado de detección de defectos en aisladores eléctricos, se analizaron diferentes métricas a lo largo de las **épocas** (ciclos de entrenamiento). En la figura 3.5 se presentan cuatro gráficos que muestran el comportamiento del modelo durante el proceso de entrenamiento: **precisión media (mAP)** y las **funciones de pérdida (loss)** correspondientes a cajas (box), clases (class) y objetos (object).

1. Precisión Media (mAP)

- En la parte superior del gráfico se muestra la evolución de dos métricas:
 - mAP (mean Average Precision):** Representada con la línea morada fuerte, esta métrica resume la precisión del modelo en la detección de objetos. En este caso, se observa una mejora progresiva desde aproximadamente 0.20 hasta alcanzar valores cercanos a 0.44 al final de las 14 épocas.
 - mAP@50:95:** Línea morada clara, mide la precisión promedio en diferentes umbrales de intersección sobre unión (IoU) desde 0.50 hasta 0.95. Aunque más baja que la mAP general, muestra una tendencia ascendente, lo que indica una mejora del modelo en detecciones más exigentes.

Estas métricas son fundamentales para evaluar la calidad del modelo en tareas de detección de objetos, como identificar defectos en los aisladores.

2. Box Loss (Pérdida de Caja)

- Esta pérdida evalúa qué tan bien el modelo predice la ubicación (caja delimitadora) de los objetos. Se observa que al inicio el valor era alto (~ 2.2), pero disminuye gradualmente hasta estabilizarse cerca de 1.85, indicando que el modelo ha aprendido a localizar los objetos con mayor precisión.

3. Class Loss (Pérdida de Clasificación)

- Evalúa la precisión con la que el modelo clasifica los objetos detectados. Comenzó con una pérdida mayor a 3.5, pero disminuyó rápidamente en las primeras épocas y se estabilizó en valores cercanos a 1.5. Esta reducción indica una mejora en la capacidad del modelo para distinguir entre clases (por ejemplo, diferentes tipos de defectos).

4. Object Loss (Pérdida de Objeto)

- Indica la confianza del modelo en que realmente hay un objeto en una determinada caja. Inició en valores cercanos a 2.35 y se redujo hasta estabilizarse alrededor de 1.85. Una baja pérdida en este aspecto es deseable, ya que significa que el modelo está seguro de sus detecciones.

3.4 Métodos para la evaluación de las CNN implementadas

3.4.1 Función de pérdida y optimización

Durante el entrenamiento, el modelo minimiza una función de pérdida compuesta, que incluye tres componentes:

- Pérdida de clasificación: mide el error al predecir la clase correcta del objeto.
- Pérdida de localización: penaliza las diferencias entre la caja predicha y la real.
- Pérdida de confianza (objeto/no objeto): mide qué tan seguro está el modelo de que hay un objeto en esa posición.

3.4.2 Entrenamiento y parada anticipada

Para evitar el sobreajuste, se aplicó la técnica de parada anticipada (early stopping), que detiene el entrenamiento cuando la pérdida en el conjunto de validación deja de mejorar después de un número determinado de épocas. En este caso, se definió un umbral de paciencia de 20 épocas.

Además, se monitorizó la $mAP@0.5$, que representa el promedio de precisión cuando el umbral de intersección sobre unión (IoU) es de al menos 50%. Este valor alcanzó un pico de 89.7%, indicando que el modelo tenía un rendimiento estable y preciso.

3.5 Detalles de implementación

3.5.1 Requisitos de hardware y software

El entrenamiento y prueba del modelo se realizaron en un entorno mixto: una parte en Google Colab para aprovechar recursos en la nube y otra localmente en una computadora personal. Las especificaciones utilizadas fueron:

- Procesador: AMD 7ma generación
- RAM: 16 GB
- GPU: NVIDIA RTX 3050 (6 GB VRAM)
- Sistema operativo: Windows 11
- Entorno de desarrollo local: Sublime Text 3 (como editor principal)
- Entorno de desarrollo en la nube: Google Colab (para entrenamiento acelerado por GPU)

Tabla 3.1 Cotización de equipo de cómputo para entrenamiento del modelo

Componente	Modelo sugerido	Precio aproximado (USD)
Procesador (CPU)	AMD Ryzen 7 5700X (8 núcleos, 4.6 GHz)	\$320.00
Tarjeta gráfica (GPU)	NVIDIA RTX 4070 Ti 12 GB GDDR6X	\$1,100.00
Memoria RAM	32 GB DDR4 3200 MHz (2x16GB)	\$160.00
Almacenamiento SSD	NVMe M.2 1TB PCIe 4.0	\$130.00
Placa base	B550 (para Ryzen)	\$180.00
Fuente de poder	750W 80+ Gold	\$140.00
Gabinete	ATX con ventilación	\$100.00
Monitor	24" 1080p IPS	\$200.00
Sistema operativo	Windows 11 Pro	\$150.00
Otros	Teclado + mouse	\$80.00
Total		\$2,560.00

3.5.2 Implementación del código y bibliotecas

La implementación del sistema se realizó en lenguaje Python 3 utilizando Sublime Text como IDE (interface development environment). El modelo previamente entrenado fue cargado en formato .pt mediante la librería ultralytics, lo cual permitió realizar inferencias sobre imágenes de aisladores eléctricos.

Se desarrolló una interfaz gráfica de usuario (GUI) con Tkinter, que permite al usuario cargar imágenes desde su ordenador, ejecutar el modelo y visualizar los resultados. También se integró la

generación automática de reportes en PDF que incluyen imágenes procesadas, resultados y análisis del modelo.

Las principales bibliotecas utilizadas fueron:

- **Tkinter:** Para la creación de la interfaz gráfica, incluyendo botones, menús y áreas de visualización.
- **PIL (Pillow):** Para manipulación y visualización de imágenes dentro de la GUI (Image, imageTk).
- **Ultralytics.YOLO:** Para cargar y ejecutar el modelo entrenado en tiempo real.
- **Os, glob, sys:** Para la gestión de archivos y rutas.
- **Matplotlib.pyplot y numpy:** Para la generación de gráficas como matriz de confusión y curvas de pérdida o precisión.
- **Datetime:** Para agregar fecha y hora automática a los reportes generados.
- **Reportlab.pdfgen.canvas y reportlab.platypus.Image:** Para crear reportes PDF personalizados que incluyen imágenes, métricas y gráficas.

CAPÍTULO 4 RESULTADOS Y DISCUSIÓN

4.1 Evaluación de la aplicación de redes convolucionales

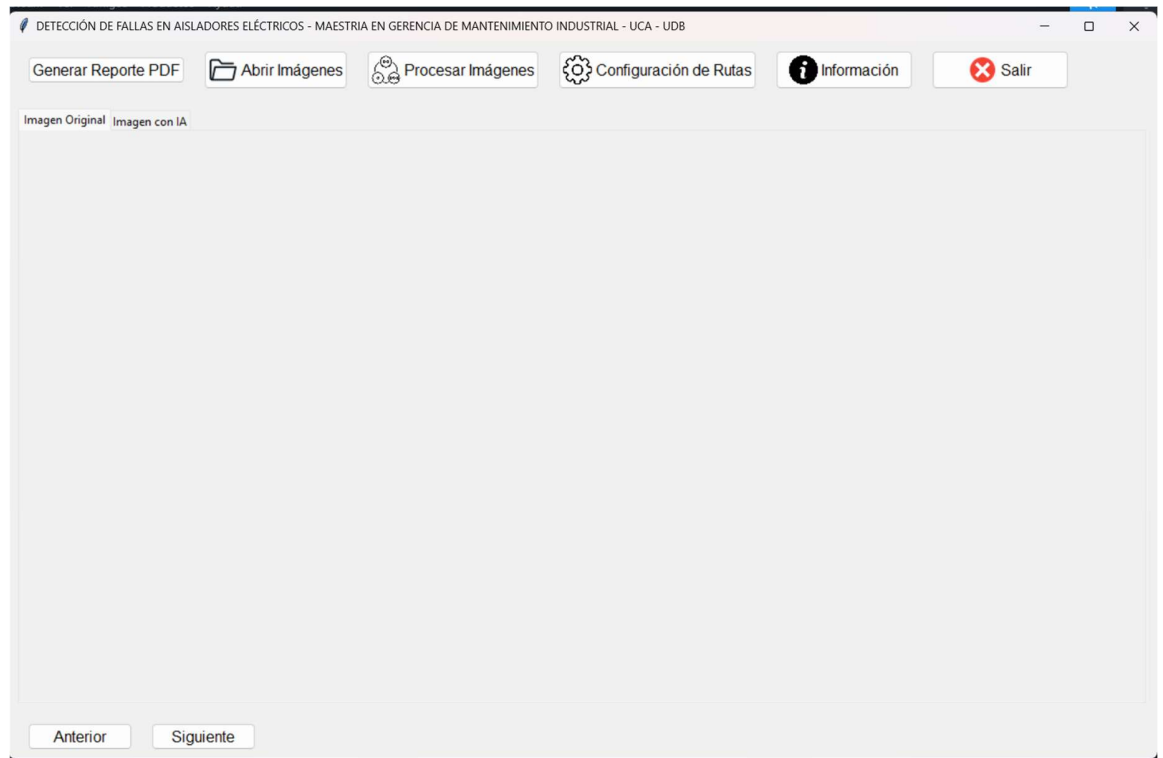


Figura 4.1 Interfaz de procesamiento de imágenes en el sistema.

Tabla 4.1 Descripción de funcionalidades de la interfaz

Botón/Elemento	Función	Descripción
Generar Reporte PDF	Exportación	Genera un reporte en formato PDF con resultado de la imagen de inicio de la interfaz
Abrir Imágenes	Carga de datos	Permite seleccionar y cargar las imágenes de aisladores a analizar
Procesar Imágenes	Procesamiento	Ejecuta el algoritmo de detección de defectos sobre las imágenes cargadas
Configuración de rutas	Configuración	Establece las rutas de almacenamiento de archivos e imágenes y a su vez cargar el archivo entrenado del sistema
Información	Datos	Información sobre integrantes de la tesis
Salir	Navegación	Cierra la aplicación

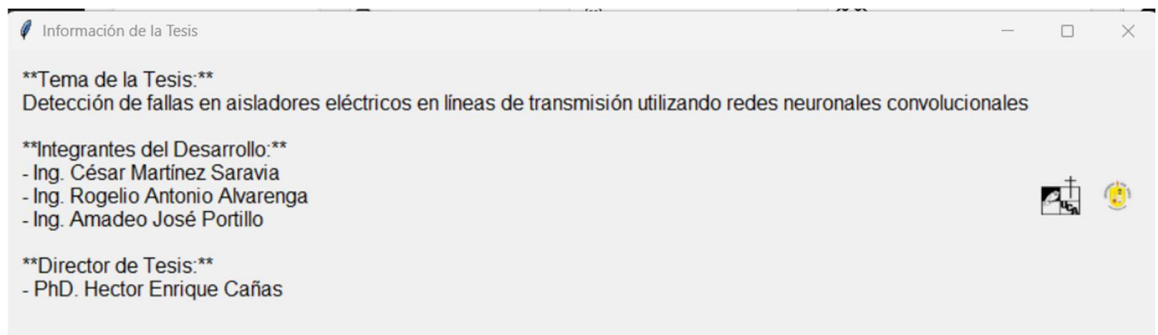


Figura 4.2 Ventana información del sistema

La ventana de información del sistema presenta los datos generales del proyecto de tesis, incluyendo el título completo de la investigación, los nombres de los tres integrantes del equipo de desarrollo y el director de tesis. Esta funcionalidad permite al usuario acceder rápidamente a los créditos y detalles académicos del proyecto, proporcionando transparencia sobre la autoría y dirección del trabajo de investigación

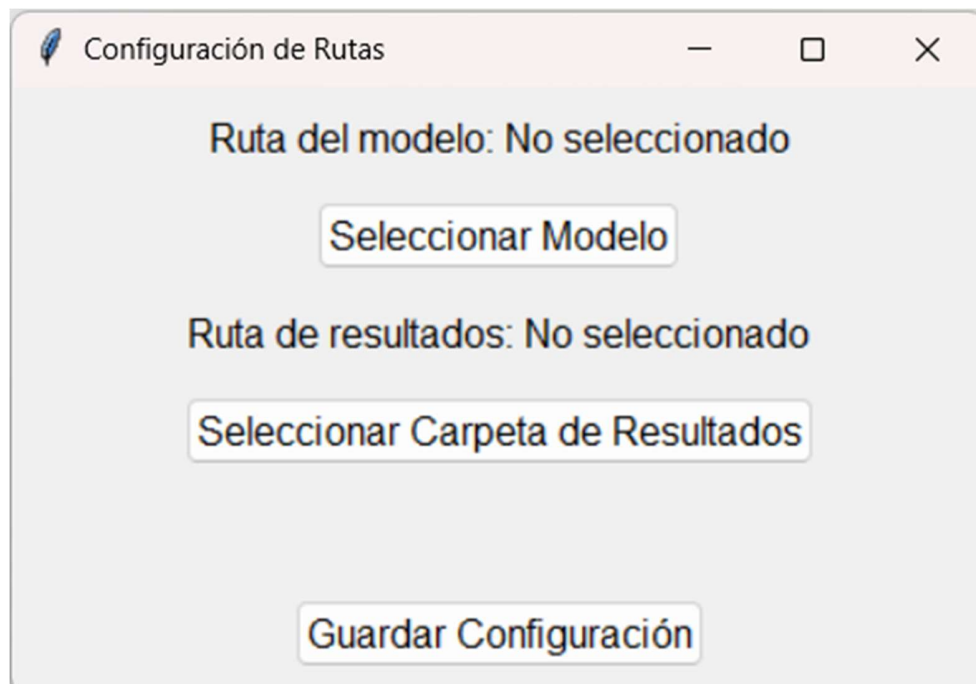


Figura 4.3 Ventana de configuración de rutas del sistema

Esta ventana permite al usuario configurar las rutas de acceso necesarias para el funcionamiento del sistema. La interfaz presenta dos secciones principales: la selección del modelo de red neuronal entrenado y la configuración del directorio donde se almacenarán los resultados del análisis. El botón 'Guardar Configuración' permite confirmar y almacenar los ajustes realizados para su uso posterior en el procesamiento de imágenes.

4.2 Discusión de resultados

En cuanto a los valores de certeza obtenidos (0.40 – 0.84), estos porcentajes corresponden a detecciones de defectos específicos en condiciones reales de imágenes capturadas en campo. Estos niveles pueden parecer bajos en comparación con métricas controladas, resultan competitivos frente a métodos convencionales, como la inspección visual, donde la variabilidad humana puede situarse incluso por debajo del 50 % de acierto en defectos incipientes y requiere de una expertiz por parte del operador para identificarlo. Además, el modelo alcanzó un nivel de confianza de **0.83 en la detección de los aisladores**, lo cual evidencia su solidez en la localización del objeto principal antes de clasificar el defecto. Estos resultados pueden mejorarse mediante estrategias de aumento de datos, entrenamiento incremental y refinamiento del modelo en escenarios locales.

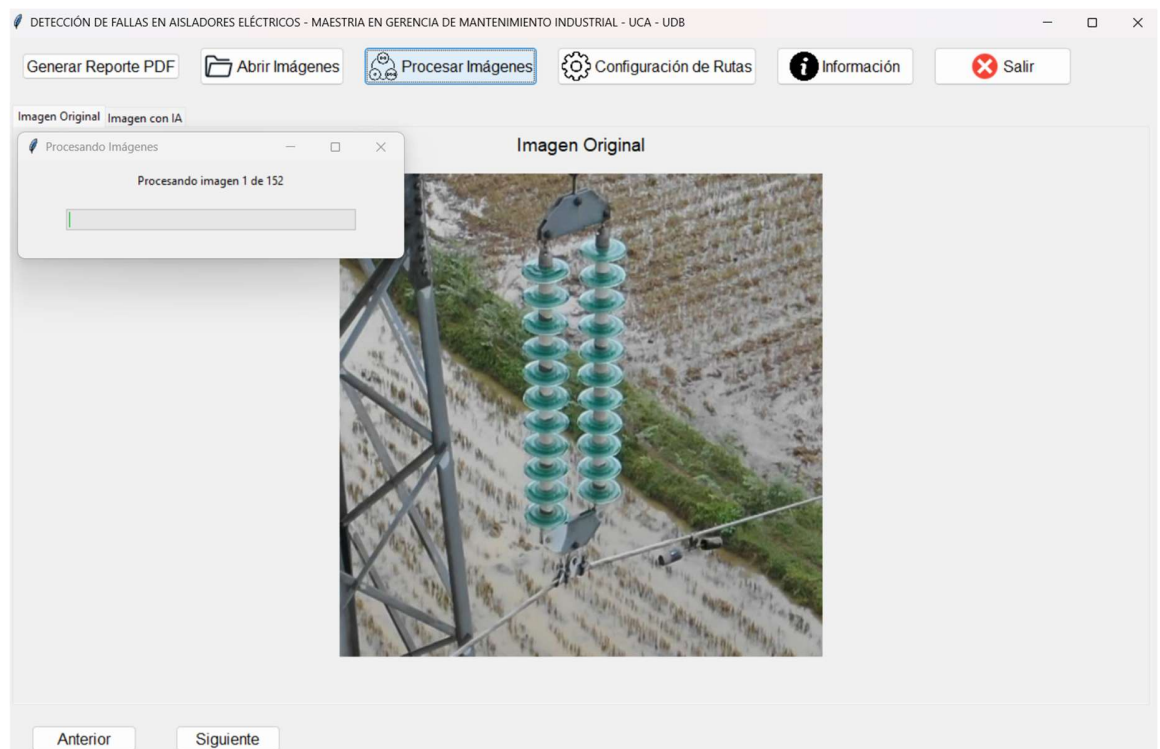


Figura 4.4 Interfaz de procesamiento de imágenes en ejecución

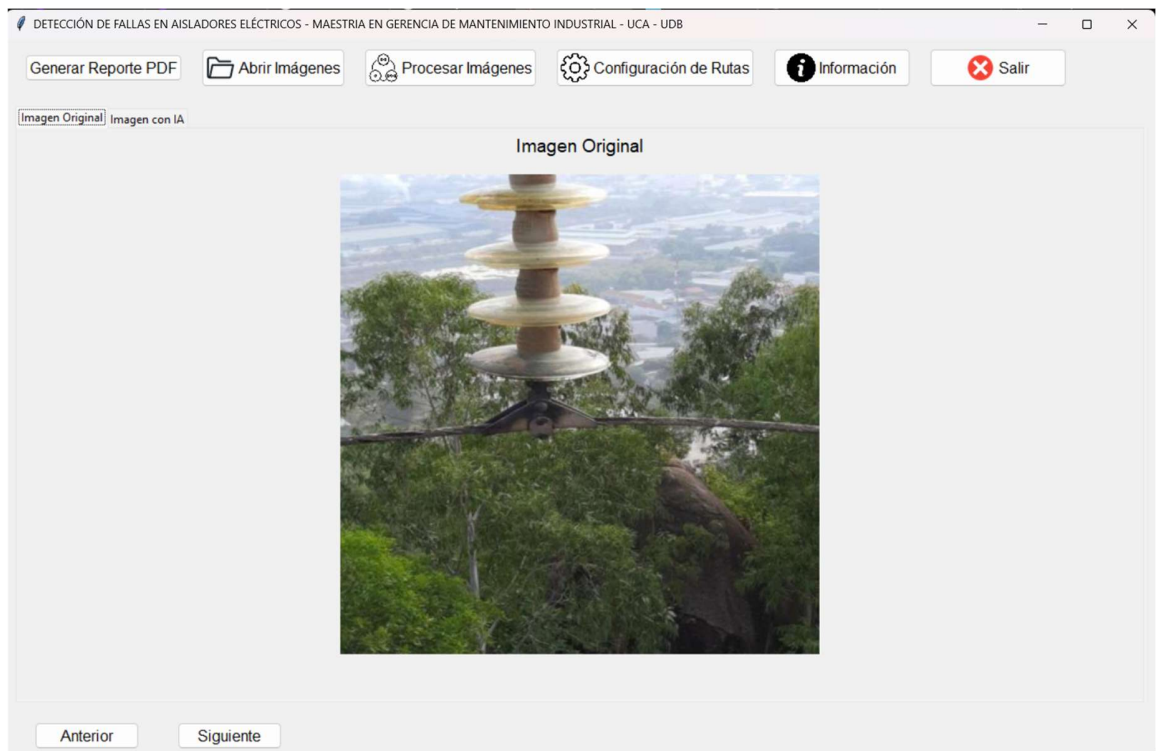


Figura 4.5 Imagen cargada

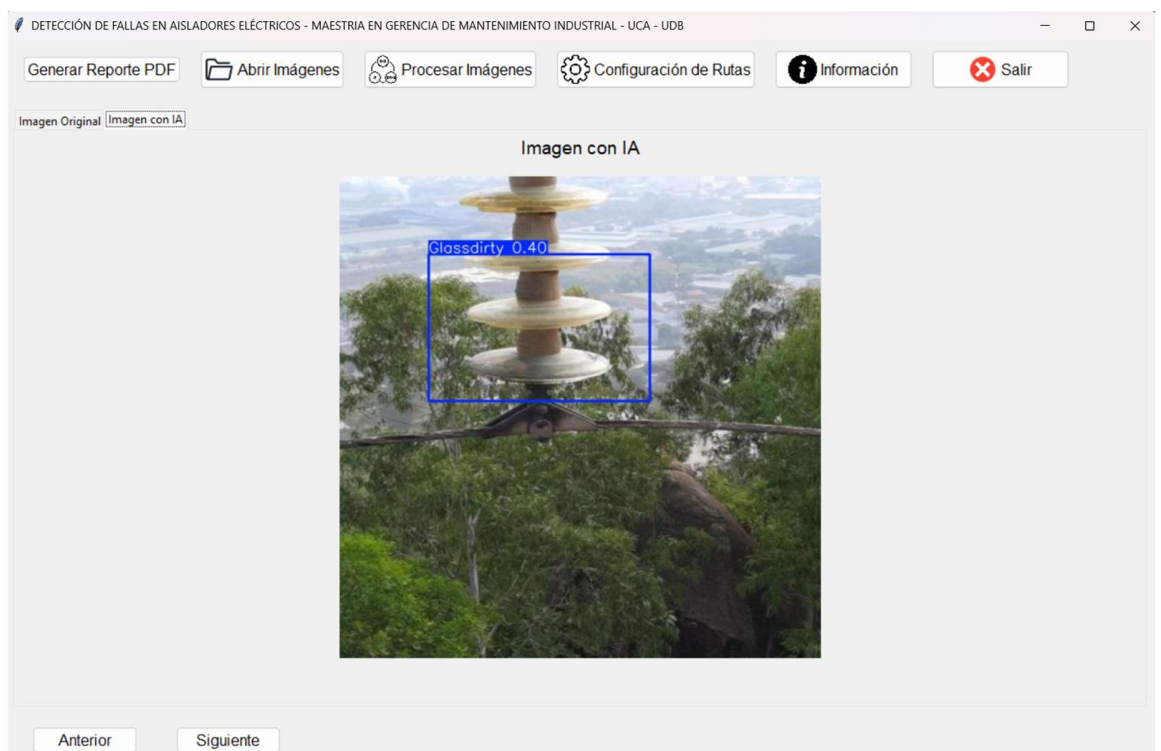


Figura 4.6 Resultado del análisis con inteligencia artificial.

La interfaz presenta el resultado del procesamiento mediante inteligencia artificial, mostrando la imagen procesada en la pestaña 'Imagen con IA'. El sistema ha detectado y delimitado un aislador eléctrico mediante un recuadro azul, identificándolo con la etiqueta 'Glassdirty' y un valor de confianza de 0.40. Esta detección indica que el algoritmo ha clasificado el aislador como sucio o contaminado con un nivel de certeza del 40%.

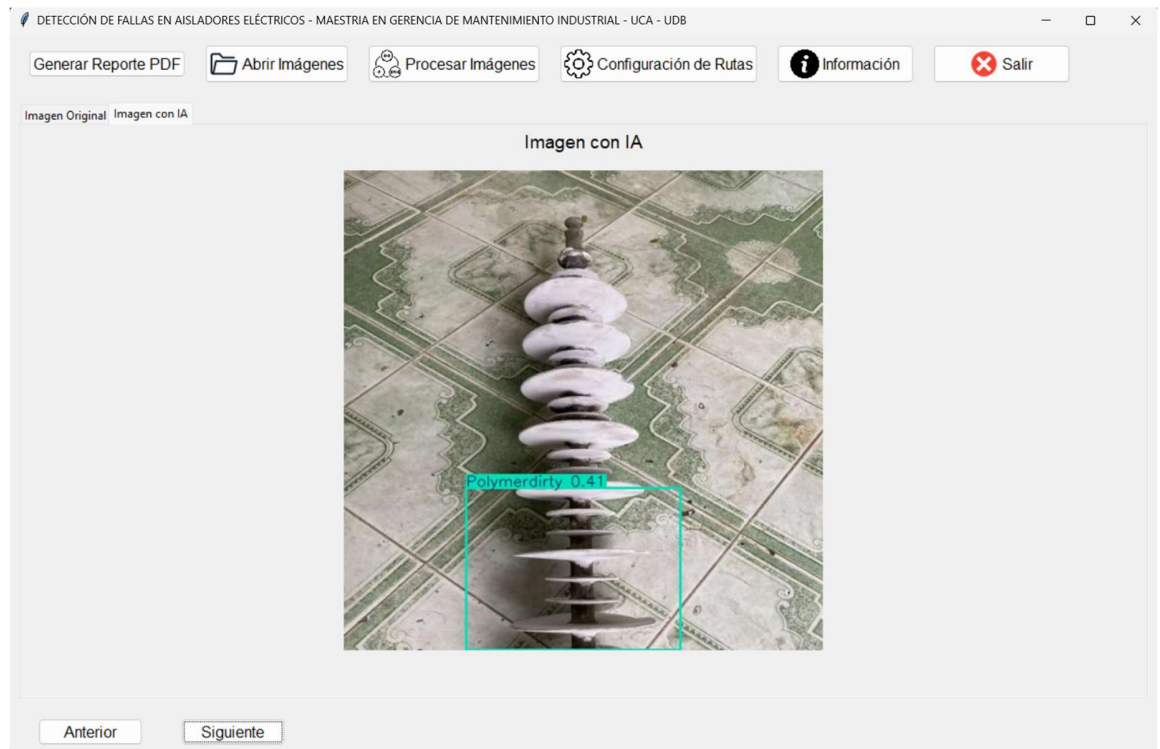


Figura 4.7 Detección de aislador polimérico contaminado

La interfaz presenta el resultado del análisis con inteligencia artificial para un aislador polimérico instalado en una torre de transmisión. El sistema ha detectado y clasificado el objeto como 'Polymerdirty' con un nivel de confianza de 0.41, delimitándolo mediante un recuadro celeste. Esta detección indica que el algoritmo ha identificado correctamente un aislador polimérico con presencia de contaminación o suciedad, evidenciando la capacidad del modelo para distinguir entre el estado limpio y contaminado de los aisladores.

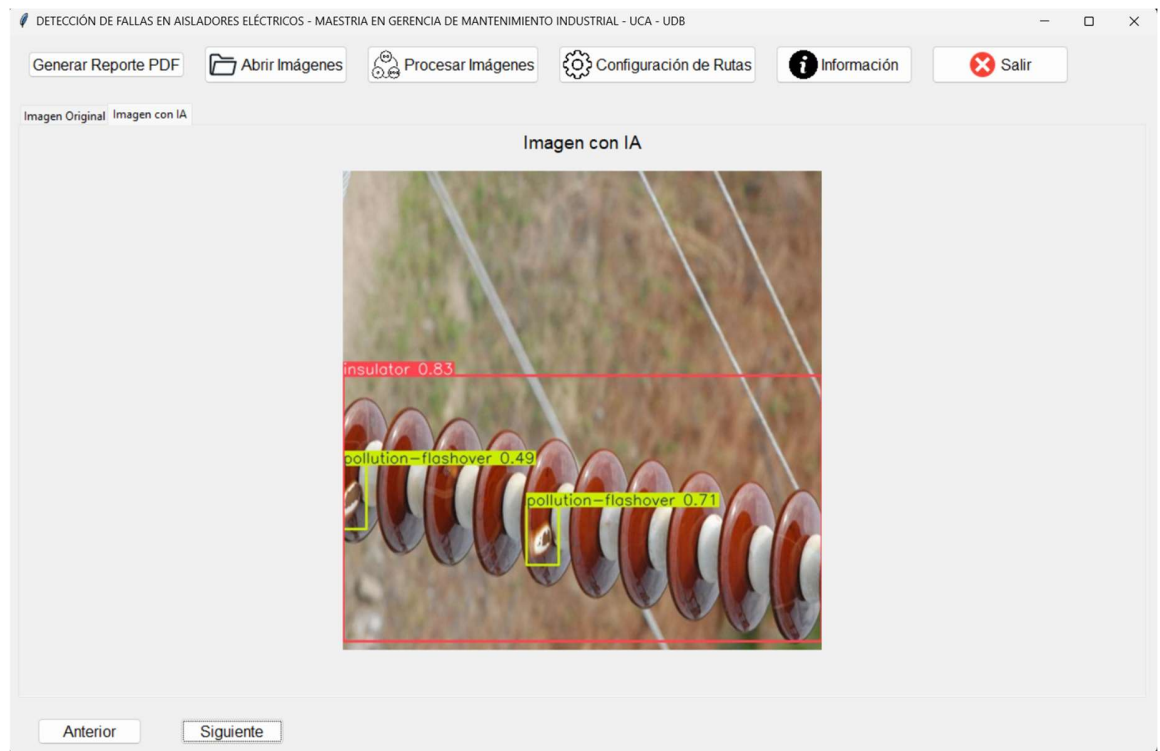


Figura 4.8 Detección de aislador flameado

La interfaz presenta el resultado del análisis con inteligencia artificial aplicado a una cadena de aisladores eléctricos de líneas de transmisión de alta tensión. El sistema ha detectado y clasificado múltiples anomalías en la imagen: un aislador defectuoso identificado como 'insulator' con un nivel de confianza de 0.83 delimitado por una línea roja, y dos instancias de contaminación por flameo clasificadas como 'pollution-flashover' significado que es un aislador flameado con niveles de confianza de 0.49 y 0.74 respectivamente, delimitadas mediante recuadros amarillos. Esta detección demuestra la capacidad del modelo para identificar diferentes tipos de defectos en aisladores cerámicos, incluyendo defectos estructurales y contaminación superficial que puede provocar flameos o descargas eléctricas no deseadas.

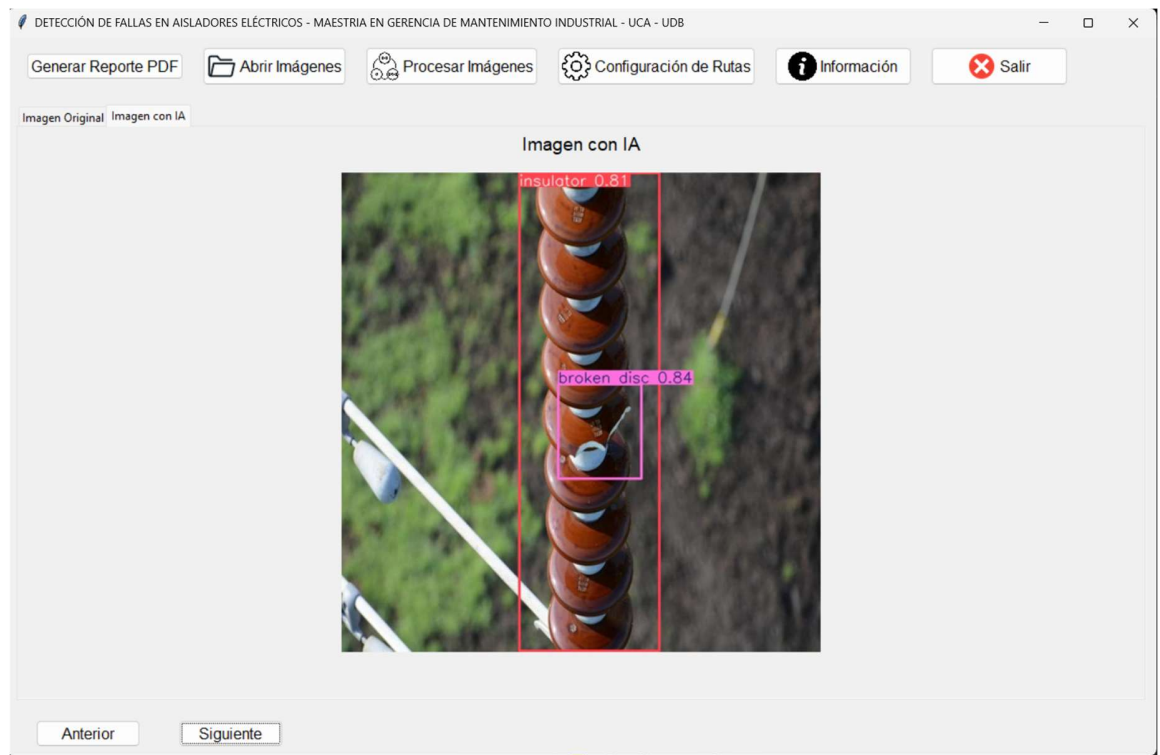
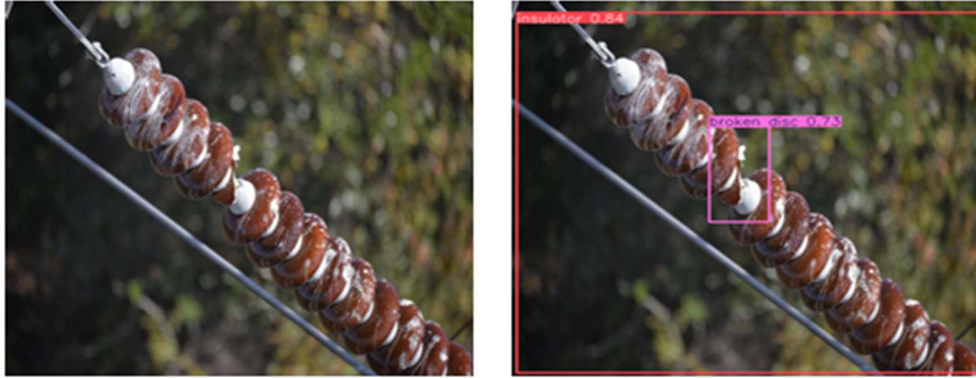


Figura 4.9 Detección de aislador quebrado

La interfaz presenta el resultado del análisis con inteligencia artificial aplicado a una cadena de aisladores cerámicos de líneas de transmisión eléctrica. El sistema ha detectado y clasificado dos elementos principales: la cadena completa de aisladores identificada como 'insulator' con un nivel de confianza de 0.81 delimitada por un recuadro rojo, y dentro de esta, un disco específico clasificado como 'broken_disc' con un nivel de confianza de 0.84 delimitado mediante un recuadro magenta. Esta detección demuestra la capacidad del modelo para identificar no solo la presencia de aisladores cerámicos tipo disco, sino también para detectar componentes individuales dañados o fracturados dentro de la cadena.

Reporte de Detección de Fallas en Aisladores Eléctricos

Fecha y hora: 2025-07-06_19-10-45



Resultados de detección (aproximado):

Clase: insulator, Confianza: 0.84

Clase: broken disc, Confianza: 0.73

Tema: Detección de fallas en aisladores eléctricos en líneas de transmisión
Desarrollado por: César Martínez, Rogelio Alvarenga, Amadeo Portillo
Director: PhD. Hector Enrique Cañas
Maestría en Gerencia de Mantenimiento Industrial - UCA/UDB

Figura 4.10 Generación de reporte PDF

Esta característica demuestra la capacidad del sistema para no solo detectar y clasificar aisladores eléctricos, sino también para documentar automáticamente los resultados del análisis en un formato de reporte profesional, facilitando la gestión documental y el seguimiento de las inspecciones de mantenimiento de la infraestructura eléctrica. El algoritmo evidencia su integración completa en un flujo de trabajo de mantenimiento predictivo, desde la detección hasta la generación de reportes técnicos.

4.3 Análisis económico costo-beneficio

Actualmente, cada jornada de inspección con UAV puede generar entre 1,000 y 3,000 fotografías, que deben ser revisadas manualmente por técnicos especializados en mantenimiento. El costo de un técnico ronda de \$4.00 a \$15.00 USD por hora, dependiendo de la experiencia. Dado que el análisis de una jornada puede requerir entre 4 y 8 horas, para filtrar las imágenes, generar un reporte, el costo estimado por jornada dedicada a la revisión de imágenes oscila entre \$32 y \$120 USD al día.

Con el sistema propuesto basado CNNs, las fotografías capturadas por UAV se cargan en la aplicación y se procesan automáticamente en segundos, generando un reporte estructurado. Esto reduce la intervención humana a una revisión final breve de los resultados, estimada en 30 a 40 minutos por jornada, equivalente a un costo de \$4.00 a \$15.00 USD según el técnico asignado.

Tabla 4.2 Comparación de costos análisis de imágenes

Método	Tiempo de jornada	Costo por jornada	Costo anual (100 jornadas)
Análisis manual de imágenes UAV	8 h	\$32.00-\$120.00	\$3,200.00-\$12,000.00
Análisis automático	30-40 min	\$2.00-\$10.00	\$200.00-\$1,000.00

CAPÍTULO 5 CONCLUSIONES Y RECOMENDACIONES

5.1 CONCLUSIONES

La investigación permitió comprobar que las redes neuronales convolucionales representan una herramienta eficaz para apoyar la detección de defectos en aisladores eléctricos de líneas de transmisión en El Salvador. Al entrenar un modelo con imágenes de distintos tipos de defectos, fue posible reconocer patrones característicos de aisladores contaminados, fracturados o deteriorados, con un nivel de precisión suficiente para plantear su aplicación como apoyo en los procesos de mantenimiento. Esto demuestra que la inteligencia artificial puede complementar de manera práctica las labores técnicas que hoy dependen en gran medida de la inspección manual.

El uso de algoritmos de visión computacional, como la arquitectura YOLO, ofrece ventajas significativas frente a métodos tradicionales. La reducción en tiempos de análisis y la capacidad de procesar grandes volúmenes de imágenes aportan eficiencia y disminuyen la probabilidad de error humano en las inspecciones. Aun así, los resultados ponen de manifiesto que la calidad de las bases de datos y la variedad de escenarios de entrenamiento son determinantes en el desempeño del modelo, lo que obliga a seguir fortaleciendo los conjuntos de imágenes empleados.

Desde la perspectiva del sector eléctrico, la propuesta constituye un primer paso hacia la integración de sistemas de diagnóstico automático dentro de los planes de mantenimiento predictivo. Si bien todavía es necesario validar el modelo en condiciones reales de operación, el desarrollo evidencia el potencial de la inteligencia artificial para aumentar la confiabilidad de la red, reducir costos asociados a inspecciones manuales y, sobre todo, minimizar la exposición de los técnicos a riesgos en campo.

Si bien el modelo de detección basado en CNNs mostró resultados prometedores, es importante reconocer ciertas limitaciones. En primer lugar, el dataset utilizado puede introducir sesgos, ya que no contempla de manera uniforme todas las condiciones climáticas y de iluminación en las que se encuentran los aisladores en campo. Asimismo, la variedad de materiales y diseños de aisladores (vidrio, porcelana, polímero) puede afectar el desempeño del modelo en contextos distintos a los considerados en el entrenamiento.

Otra limitación se relaciona con la calidad de las imágenes capturadas por UAV, pues la distancia, ángulo de enfoque y resolución influyen directamente en la precisión del algoritmo. Finalmente, la detección de ciertos defectos poco comunes o incipientes puede no alcanzar los niveles

de confiabilidad esperados, lo que implica la necesidad de un entrenamiento continuo con nuevos datos que reflejen escenarios más variados.

5.2 RECOMENDACIONES

Ampliar y diversificar la base de datos de imágenes, incorporando condiciones propias de la red salvadoreña, como ambientes costeros, zonas de alta contaminación y variaciones de iluminación.

Realizar pruebas piloto del modelo en entornos reales de operación, en conjunto con personal de mantenimiento, para medir su desempeño y ajustar parámetros de detección según la práctica en campo.

Explorar el uso de técnicas de aprendizaje transferido y de optimización de hiperparámetros, con el fin de mejorar la precisión y robustez del modelo frente a escenarios poco representados en el entrenamiento inicial.

Desarrollar una interfaz de usuario sencilla e intuitiva que facilite la implementación del sistema como herramienta de apoyo al personal técnico, reduciendo la necesidad de conocimientos avanzados en inteligencia artificial.

Considerar en futuras investigaciones la aplicación del sistema en líneas de transmisión, a fin de evaluar el desempeño del modelo en aisladores en líneas de transmisión, ampliando así el alcance y la generalización de la propuesta inicial orientada a la red de distribución de AES El Salvador.

Implementar técnicas avanzadas de filtrado y preprocesamiento de imágenes en las etapas iniciales del sistema, con el propósito de mejorar la calidad visual de los datos antes del análisis por el modelo. Se sugiere para futuras investigaciones incorporar filtros adaptativos, corrección de iluminación y reducción de ruido, así como el desarrollo de un módulo automatizado que permita la descarga, clasificación y actualización continua de las imágenes. Esto contribuiría a optimizar el rendimiento del modelo de detección y facilitar la ampliación dinámica de la base de datos de entrenamiento.

Elaborar un presupuesto técnico y operativo que contemple los costos asociados a la adquisición o arrendamiento de UAV, mantenimiento preventivo, baterías, licencias de software y seguros, así como los gastos relacionados con la capacitación del personal encargado de la operación y análisis de datos.

REFERENCIAS BIBLIOGRÁFICAS

- Banerjee, A. (2018). Insulator Fault Detection using Image Processing.
- Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.
- Borji, A., Cheng, M.-M., Hou, Q., Jiang, H., & Li, J. (2019). Salient object detection: A survey. *Computational visual media*, 5, 117–150.
- Chen, Y., Ding, Y., Zhao, F., Zhang, E., Wu, Z., & Shao, L. (2021). Surface defect detection methods for industrial products: A review. *Applied Sciences*, 11, 7657.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., . . . others. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.
- Du, X., Cai, Y., Wang, S., & Zhang, L. (2016). Overview of deep learning. *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, (págs. 159–164).
- Elfwing, S., Uchibe, E., & Doya, K. (2018). Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 107, 3–11.
- Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *nature*, 542, 115–118.
- Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning* (Vol. 1). MIT press Cambridge.
- Hussain, M. (2023). YOLO-v1 to YOLO-v8, the rise of YOLO and its complementary nature toward digital manufacturing and industrial defect detection. *Machines*, 11, 677.
- Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and electronics in agriculture*, 147, 70–90.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, 521, 436–444.
- Levine, S., Pastor, P., Krizhevsky, A., Ibarz, J., & Quillen, D. (2018). Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. *The International journal of robotics research*, 37, 421–436.

- Li, C., Li, L., Jiang, H., Weng, K., Geng, Y., Li, L., . . . others. (2022). YOLOv6: A single-stage object detection framework for industrial applications. *arXiv preprint arXiv:2209.02976*.
- Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A., Ciompi, F., Ghafoorian, M., . . . Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical image analysis*, 42, 60–88.
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, (págs. 21–37).
- Martínez-Plumed, F., Contreras-Ochando, L., Ferri, C., Hernández-Orallo, J., Kull, M., Lachiche, N., . . . Flach, P. (2019). CRISP-DM twenty years later: From data mining processes to data science trajectories. *IEEE transactions on knowledge and data engineering*, 33, 3048–3061.
- McCool, C., Perez, T., & Upcroft, B. (2017). Mixtures of lightweight deep convolutional neural networks: Applied to agricultural robotics. *IEEE Robotics and Automation Letters*, 2, 1344–1351.
- Plotnikova, V., Dumas, M., & Milani, F. (2020). Adaptations of data mining methodologies: A systematic literature review. *PeerJ Computer Science*, 6, e267.
- Qadri, S. A., Huang, N.-F., Wani, T. M., & Bhat, S. A. (2023). Plant disease detection and segmentation using end-to-end yolov8: A comprehensive approach. *2023 IEEE 13th International Conference on Control System, Computing and Engineering (ICCSCE)*, (págs. 155–160).
- Qu, J., Li, Q., Pan, J., Sun, M., Lu, X., Zhou, Y., & Zhu, H. (2025). SS-YOLOv8: small-size object detection algorithm based on improved YOLOv8 for UAV imagery. *Multimedia Systems*, 31, 1–17.
- Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE conference on computer vision and pattern recognition*, (págs. 779–788).

- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61, 85–117.
- Sohan, M., Sai Ram, T., Reddy, R., & Venkata, C. (2024). A review on yolov8 and its advancements. *International Conference on Data Intelligence and Cognitive Informatics*, (págs. 529–545).
- Tan, M., & Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. *International conference on machine learning*, (págs. 6105–6114).
- Terven, J., Córdova-Esparza, D.-M., & Romero-González, J.-A. (2023). A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas. *Machine learning and knowledge extraction*, 5, 1680–1716.
- Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2023). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, (págs. 7464–7475).
- Wang, C.-Y., Liao, H.-Y. M., Wu, Y.-H., Chen, P.-Y., Hsieh, J.-W., & Yeh, I.-H. (2020). CSPNet: A new backbone that can enhance learning capability of CNN. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, (págs. 390–391).
- Wankhede, K., Wukkadada, B., & Nadar, V. (2018). Just walk-out technology and its challenges: A case of Amazon Go. *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, (págs. 254–257).
- Zhao, Z.-Q., Zheng, P., Xu, S.-t., & Wu, X. (2019). Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30, 3212–3232.
- Yue Liu, Decheng Liu, Xinbo Huang, Chenjing Li. (2022) Insulator defect detection with deep learning : A survey. *The Institution of Engineering and Thecnology (IET)*, (págs. 3542 – 3543).

ANEXOS

ANEXO A. CODIGO DEL ALGORITMO UTILIZADO

```
import tkinter as tk
from tkinter import filedialog, messagebox, ttk
from PIL import Image, ImageTk
from ultralytics import YOLO
import os
import glob
import sys
import matplotlib.pyplot as plt
import numpy as np
from datetime import datetime
from reportlab.lib.pagesizes import letter
from reportlab.pdfgen import canvas
from reportlab.platypus import Image as RLImage

# Inicializar variables globales
MODEL_PATH = "
OUTPUT_FOLDER = "
model = None
selected_image_paths = []
current_index = 0

# Crear la ventana principal
root = tk.Tk()
root.title("DETECCIÓN DE DEFECTOS EN AISLADORES ELÉCTRICOS - MAESTRIA EN
GERENCIA DE MANTENIMIENTO INDUSTRIAL - UCA - UDB")

# Configuración de la ventana
root.geometry("1200x750")
root.configure(bg='#ECF0F1')

# Ruta a la carpeta de Descargas
if getattr(sys, 'frozen', False):
```

```

    # Si estamos en un ejecutable generado
    base_path = sys._MEIPASS
else:
    # Si estamos ejecutando como script
    base_path = os.path.dirname(os.path.abspath(__file__))

icons_path = os.path.join(base_path, 'TESIS')

# Función para cargar un icono
def load_icon(filename):
    return ImageTk.PhotoImage(Image.open(os.path.join(icons_path, filename)).resize((30, 30),
Image.LANCZOS))

# Cargar iconos
open_icon = load_icon('open_image.png')
settings_icon = load_icon('settings.png')
exit_icon = load_icon('exit.png')
process_icon = load_icon('process_image.png')
info_icon = load_icon('info.png') # el icono info.png
udb_logo = load_icon('udb_logo.png') # logo UDB
uca_logo = load_icon('uca_logo.png') # logo UCA
pdf_icon = load_icon('pdf.png') if os.path.exists(os.path.join(icons_path, 'pdf.png')) else None

# frame para los botones
menu_frame = ttk.Frame(root, padding="10")
menu_frame.pack(side=tk.TOP, fill=tk.X)

# frame para la imagen y los botones
main_frame = ttk.Frame(root, padding="10")
main_frame.pack(fill=tk.BOTH, expand=True)

# Crear un frame para la pestaña de imágenes
def create_image_frame(tab):
    frame = ttk.Frame(tab)

```

```

frame.pack(fill=tk.BOTH, expand=True)
return frame

# Función para cargar el modelo
def load_model():
    global model
    try:
        model = YOLO(MODEL_PATH)
        print("Modelo cargado exitosamente.")
    except Exception as e:
        messagebox.showerror("Error", f"No se pudo cargar el modelo: {e}")
        root.destroy() # Cerrar la aplicación si no se carga el modelo

# Función para cargar imágenes
def load_images():
    global selected_image_paths
    global current_index
    if not MODEL_PATH or not OUTPUT_FOLDER:
        messagebox.showerror("Error", "Por favor, configure las rutas antes de seleccionar imágenes.")
    return

file_paths = filedialog.askopenfilenames(filetypes=[("Image Files", "*.jpg;*.jpeg;*.png")])
if file_paths:
    selected_image_paths = list(file_paths)
    current_index = 0
    show_image(original_tab, 'Imagen Original')
    show_image(processed_tab, 'Imagen con IA')

def show_image(tab, label_text):
    tab_frame = tab.winfo_children()
    if not tab_frame:
        # Si no hay widgets en el frame, crear uno nuevo
        tab_frame = create_image_frame(tab)

```

```

else:
    tab_frame = tab_frame[0]

for widget in tab_frame.winfo_children():
    widget.destroy()

image_path = selected_image_paths[current_index] if tab == original_tab else
get_processed_image_path()
if image_path:
    try:
        image = Image.open(image_path)
        image.thumbnail((500, 500))
        tk_image = ImageTk.PhotoImage(image)

        label = ttk.Label(tab_frame, text=label_text, font=('Helvetica', 14))
        label.pack(pady=5)

        img_label = ttk.Label(tab_frame, image=tk_image)
        img_label.image = tk_image
        img_label.pack(pady=10)

    except Exception as e:
        messagebox.showerror("Error", f'Ocurrió un error al mostrar la imagen: {e}')

def get_processed_image_path():
    if not selected_image_paths:
        return None

    base_name = os.path.basename(selected_image_paths[current_index])
    pattern = os.path.join(OUTPUT_FOLDER, 'predict*' + base_name)
    matching_files = glob.glob(pattern)
    if matching_files:
        return matching_files[0] # Devolver el primer archivo encontrado

```

```

return None

# Función para procesar las imágenes
def process_images():
    if not selected_image_paths:
        messagebox.showerror("Error", "Por favor, seleccione imágenes antes de procesar.")
        return

    # Crear ventana de progreso
    progress_window = tk.Toplevel(root)
    progress_window.title("Procesando Imágenes")
    progress_window.geometry("400x100")

    progress_label = ttk.Label(progress_window, text="Procesando...")
    progress_label.pack(pady=10)

    progress_bar = ttk.Progressbar(progress_window, length=300, mode='determinate')
    progress_bar.pack(pady=10)
    progress_bar['maximum'] = len(selected_image_paths)

    try:
        for i, image_path in enumerate(selected_image_paths):
            # Actualizar progreso
            progress_label.config(text=f"Procesando {i+1} de {len(selected_image_paths)} imagen")
            progress_bar['value'] = i + 1
            progress_window.update()

            temp_image_path = os.path.join(OUTPUT_FOLDER, 'temp',
os.path.basename(image_path))
            os.makedirs(os.path.dirname(temp_image_path), exist_ok=True)

            image = Image.open(image_path)
            image.save(temp_image_path)

```

```

        results = model.predict(temp_image_path, save=True, project=OUTPUT_FOLDER)

    progress_window.destroy()
    show_image(processed_tab, 'Imagen con IA')
    messagebox.showinfo("Éxito", "Todas las imágenes han sido procesadas")

except Exception as e:
    progress_window.destroy()
    messagebox.showerror("Error", f'Error al procesar: {e}')

def generate_pdf_report():
    if not selected_image_paths:
        messagebox.showerror("Error", "No hay imágenes procesadas para generar el informe.")
        return

    try:
        now = datetime.now().strftime("%Y-%m-%d_%H-%M-%S")
        pdf_path = os.path.join(OUTPUT_FOLDER, f'Reporte_Deteccion_{now}.pdf')
        c = canvas.Canvas(pdf_path, pagesize=letter)

        c.setFont("Helvetica-Bold", 14)
        c.drawString(30, 750, "Reporte de Detección de Defectos en Aisladores Eléctricos")
        c.setFont("Helvetica", 10)
        c.drawString(30, 735, f'Fecha y hora: {now}')

        # Imagen original y con IA
        original = selected_image_paths[current_index]
        procesada = get_processed_image_path()
        if original: c.drawImage(original, 30, 500, width=250, height=200)
        if procesada: c.drawImage(procesada, 300, 500, width=250, height=200)

        c.setFont("Helvetica", 12)
        c.drawString(30, 470, "Resultados de detección (aproximado):")

```

```

if model:
    results = model.predict(original)
    y = 455
    for r in results:
        for box in r.bboxes:
            cls = model.names[int(box.cls)]
            conf = float(box.conf)
            c.drawString(40, y, f'Clase: {cls}, Confianza: {conf:.2f}')
            y -= 15

# Datos del proyecto
c.setFont("Helvetica", 10)
c.drawString(30, 100, "Tema: Detección de defectos en aisladores eléctricos en líneas de
transmisión")
c.drawString(30, 85, "Desarrollado por: César Martínez, Rogelio Alvarenga, Amadeo
Portillo")
c.drawString(30, 70, "Director: PhD. Hector Enrique Cañas")
c.drawString(30, 55, "Maestría en Gerencia de Mantenimiento Industrial - UCA/UDB")

c.save()
messagebox.showinfo("Éxito", f'Reporte generado exitosamente en: {pdf_path}')

except Exception as e:
    messagebox.showerror("Error", f'No se pudo generar el PDF: {e}')

# botón para generar PDF
btn_pdf = ttk.Button(menu_frame, text="Generar Reporte PDF", image=pdf_icon,
compound=tk.LEFT, command=generate_pdf_report)
btn_pdf.pack(side=tk.LEFT, padx=10)

# Función para abrir la ventana de configuración
def open_config_window():
    config_window = tk.Toplevel(root)
    config_window.title("Configuración de Rutas")

```

```

config_window.geometry("400x250")

def select_model():
    global MODEL_PATH
    MODEL_PATH = filedialog.askopenfilename(filetypes=[("Model Files", "*.pt")])
    model_label.config(text=f'Ruta del modelo: {os.path.basename(MODEL_PATH)} if
MODEL_PATH else 'No seleccionado}')
    if MODEL_PATH:
        load_model()
        update_status_label()

def select_output_folder():
    global OUTPUT_FOLDER
    OUTPUT_FOLDER = filedialog.askdirectory()
    output_folder_label.config(text=f'Ruta de resultados:
{os.path.basename(OUTPUT_FOLDER)} if OUTPUT_FOLDER else 'No seleccionado}')
    update_status_label()

def update_status_label():
    status = "Modelo y carpeta de resultados configurados correctamente." if MODEL_PATH
and OUTPUT_FOLDER else "Por favor, configure ambas rutas."
    status_label.config(text=status)

def save_config():
    if not MODEL_PATH or not OUTPUT_FOLDER:
        messagebox.showerror("Error", "Por favor, configure ambas rutas antes de guardar.")
        return
    messagebox.showinfo("Éxito", "Rutas configuradas correctamente")
    config_window.destroy()

model_label = ttk.Label(config_window, text="Ruta del modelo: No seleccionado",
font=('Helvetica', 12))
model_label.pack(pady=10)

```

```

        select_model_btn = ttk.Button(config_window, text="Seleccionar Modelo",
command=select_model)
        select_model_btn.pack(pady=5)

        output_folder_label = ttk.Label(config_window, text="Ruta de resultados: No seleccionado",
font=('Helvetica', 12))
        output_folder_label.pack(pady=10)
        select_output_folder_btn = ttk.Button(config_window, text="Seleccionar Carpeta de
Resultados", command=select_output_folder)
        select_output_folder_btn.pack(pady=5)

        status_label = ttk.Label(config_window, text="", font=('Helvetica', 10))
        status_label.pack(pady=10)

        save_btn = ttk.Button(config_window, text="Guardar Configuración",
command=save_config)
        save_btn.pack(pady=10)

        exit_btn = ttk.Button(config_window, text="Salir", command=config_window.destroy)
        exit_btn.pack(pady=5)

# Función para mostrar la información de la tesis
def show_info():
    info_window = tk.Toplevel(root)
    info_window.title("Información de la Tesis")

    # Crear un frame en el Toplevel
    frame = ttk.Frame(info_window, padding="10")
    frame.pack(fill=tk.BOTH, expand=True)

    # Etiqueta para el texto de la información
    info_text = ""
    **Tema de la Tesis:**

```

Detección de defectos en aisladores eléctricos en líneas de transmisión utilizando redes neuronales convolucionales

****Integrantes del Desarrollo:****

- Ing. César Martínez Saravia
- Ing. Rogelio Antonio Alvarenga
- Ing. Amadeo José Portillo

****Director de Tesis:****

- PhD. Hector Enrique Cañas

""

```
info_label = tk.Label(frame, text=info_text, justify=tk.LEFT, font=('Helvetica', 12))
info_label.pack(side=tk.LEFT, fill=tk.BOTH, expand=True)
```

Función para cargar logos

```
def load_logos():
```

```
    udb_label = tk.Label(frame, image=udb_logo)
    udb_label.pack(side=tk.RIGHT, padx=5, pady=5)
```

```
    uca_label = tk.Label(frame, image=uca_logo)
    uca_label.pack(side=tk.RIGHT, padx=5, pady=5)
```

```
load_logos()
```

Ajuste automático el tamaño de la ventana

```
info_window.update_idletasks()
```

```
info_window.geometry(f'{info_window.winfo_reqwidth()}x{info_window.winfo_reqheight()}')
```

Crear pestañas

```
tab_control = ttk.Notebook(main_frame)
```

```
original_tab = ttk.Frame(tab_control)
```

```

processed_tab = ttk.Frame(tab_control)
info_tab = ttk.Frame(tab_control)

tab_control.add(original_tab, text='Imagen Original')
tab_control.add(processed_tab, text='Imagen con IA')
#tab_control.add(info_tab, text='Información')
tab_control.pack(expand=1, fill='both')

# Crear botones en el menú
btn_open = ttk.Button(menu_frame, text="Abrir Imágenes", image=open_icon,
compound=tk.LEFT, command=load_images)
btn_open.pack(side=tk.LEFT, padx=10)

btn_process = ttk.Button(menu_frame, text="Procesar Imágenes", image=process_icon,
compound=tk.LEFT, command=process_images)
btn_process.pack(side=tk.LEFT, padx=10)

btn_config = ttk.Button(menu_frame, text="Configuración de Rutas", image=settings_icon,
compound=tk.LEFT, command=open_config_window)
btn_config.pack(side=tk.LEFT, padx=10)

btn_info = ttk.Button(menu_frame, text="Información", image=info_icon, compound=tk.LEFT,
command=show_info)
btn_info.pack(side=tk.LEFT, padx=10)

btn_exit = ttk.Button(menu_frame, text="Salir", image=exit_icon, compound=tk.LEFT,
command=root.quit)
btn_exit.pack(side=tk.LEFT, padx=10)

# Etiquetas para los botones de navegación
def show_next_image():
    global current_index
    if current_index < len(selected_image_paths) - 1:
        current_index += 1

```

```

        show_image(original_tab, 'Imagen Original')
        show_image(processed_tab, 'Imagen con IA')

def show_previous_image():
    global current_index
    if current_index > 0:
        current_index -= 1
        show_image(original_tab, 'Imagen Original')
        show_image(processed_tab, 'Imagen con IA')

nav_frame = ttk.Frame(root, padding="10")
nav_frame.pack(side=tk.BOTTOM, fill=tk.X)

prev_button = ttk.Button(nav_frame, text="Anterior", command=show_previous_image)
prev_button.pack(side=tk.LEFT, padx=20)

next_button = ttk.Button(nav_frame, text="Siguiente", command=show_next_image)
next_button.pack(side=tk.LEFT, padx=20)

# Estilo para los botones
style = ttk.Style()
style.configure('TButton', font=('Helvetica', 12), foreground='black')

def update_navigation_buttons():
    """Actualizar estado de botones de navegación"""
    if len(selected_image_paths) <= 1:
        prev_button.config(state='disabled')
        next_button.config(state='disabled')
    else:
        prev_button.config(state='normal' if current_index > 0 else 'disabled')
        next_button.config(state='normal' if current_index < len(selected_image_paths) - 1 else
'disabled')

# Agregar contador de imágenes
if selected_image_paths:

```

```
        image_counter.config(text=f"Imagen {current_index + 1} de  
{len(selected_image_paths)}")
```

```
# Contador al frame de navegación
```

```
image_counter = ttk.Label(nav_frame, text="")
```

```
image_counter.pack(side=tk.LEFT, padx=20)
```

```
def setup_keyboard_shortcuts():
```

```
    """Configurar atajos de teclado"""
```

```
    root.bind('<Control-o>', lambda e: load_images())
```

```
    root.bind('<Control-p>', lambda e: process_images())
```

```
    root.bind('<Left>', lambda e: show_previous_image())
```

```
    root.bind('<Right>', lambda e: show_next_image())
```

```
    root.bind('<F1>', lambda e: show_info())
```

```
    root.bind('<Escape>', lambda e: root.quit())
```

```
# Llamar después de crear la ventana principal
```

```
setup_keyboard_shortcuts()
```

```
root.mainloop()
```