

# **Instituto Tecnológico y de Estudios Superiores de Occidente**

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Electrónica, Sistemas e Informática  
**Maestría en Sistemas Computacionales**



## **Algoritmo de visión computacional para validación de rieles de potencia:**

**TESIS PARA LA OBTENCIÓN DEL GRADO DE MAestrÍA EN  
SISTEMAS COMPUTACIONALES**

Presenta: **JOSE JONATHAN CUEVAS VALENCIA**

Asesor: **MS. VICTOR HUGO MARTINEZ SANCHEZ**

Tlaquepaque, Jalisco. Junio 2024

# Agradecimientos

Este proyecto no habría sido posible sin el apoyo y la orientación de varias personas y organizaciones a las cuales me gustaría expresar mi más sincero agradecimiento.

En primer lugar, quiero agradecer a mis profesores del ITESO, cuyo conocimiento y dedicación han sido fundamentales en mi formación académica y en la realización de este proyecto. Su pasión por la enseñanza y su constante disposición para ayudarme han sido una fuente de inspiración y motivación.

A mi asesor, Victor Hugo, le expreso mi profundo agradecimiento por su invaluable guía y apoyo durante todo el desarrollo de este proyecto. Su paciencia, consejos y retroalimentación han sido cruciales para el éxito de este trabajo. Su compromiso con mi crecimiento profesional y académico ha sido verdaderamente admirable.

A mi familia, principalmente mi madre Adela, mi hermana Misja y a mi novia Alicia, quiero expresarles mi más profundo agradecimiento por su amor incondicional y su apoyo constante. Gracias por creer en mí, por sus sacrificios y por estar siempre a mi lado, brindándome el ánimo necesario para superar cada desafío. Este logro también es suyo.

A mi equipo de trabajo en Intel, y en especial a mi gerente, Horacio y a todos mis colegas, les agradezco profundamente por su apoyo y comprensión. Gracias por brindarme la flexibilidad necesaria para dedicar el tiempo necesario a mis proyectos y tareas a lo largo de la maestría. Su colaboración y respaldo han sido fundamentales para alcanzar este logro.

Finalmente, agradezco al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo financiero brindado. Sin su respaldo, la realización de este proyecto no hubiera sido posible. Su compromiso con el desarrollo científico y tecnológico del país es una fuente de motivación para todos los que aspiramos a contribuir en estos campos.

A todos ustedes, muchas gracias.

## Dedicatoria

A la memoria de mi querido padre, quien siempre soñó con verme convertido en ingeniero. Aunque no pudo estar presente para ver este logro, su amor, sus enseñanzas y su deseo de verme triunfar han sido mi mayor motivación. Este logro es para ti, papá, con todo mi amor y gratitud.

También dedicado en mi hermano en paz descansa, quien me enseñó la perseverancia por medio de los videojuegos y me enseñó a armar computadores desde pequeño, recuerdo sus noches de trabajo incansable, siempre acompañado de una sonrisa y una palabra de aliento. Su pasión por el conocimiento y su dedicación a la familia me enseñaron el verdadero significado del esfuerzo y la perseverancia. Cada vez que me enfrentaba a un desafío, pensaba en sus consejos y en la fortaleza con la que enfrentaba la vida.

Las lecciones de ambos siguen guiando cada uno de mis pasos, y este triunfo no hubiera sido posible sin el legado de sus valores y su amor incondicional.

## Resumen

En los últimos años, ha habido una creciente demanda de automatizar tareas relacionadas con la clasificación en el campo de las mediciones. Uno de los desafíos particulares en esta área es la identificación de ruido en mediciones de picos reales sin necesidad de supervisión. Este trabajo de obtención de grado propone un algoritmo que aborda este desafío, logrando un nivel de efectividad superior al 90% en la identificación de formas de onda. El algoritmo no solo automatiza el proceso de toma y clasificación de mediciones de corriente, sino que también incorpora las mejores prácticas para el acotamiento de mediciones. Los resultados de este estudio demuestran el potencial de este algoritmo para mejorar significativamente la eficiencia y precisión de las tareas de clasificación en el campo de las mediciones.

La innovación de este trabajo radica en la detección automatizada de picos de corriente reales, un proceso que hasta ahora ha sido manual y propenso a errores. Mediante el uso de algoritmos avanzados, se logra distinguir con precisión diferentes tipos de formas de onda, lo que representa una mejora significativa en la metodología de mediciones de rieles de potencia. Además, la aplicación de algoritmos de visión en conjunto con osciloscopios es una novedad en este campo, explorando nuevas fronteras en la tecnología de señales de potencia.

La razón para crear este algoritmo de visión es el hecho de que los osciloscopios actuales no cuentan con la capacidad de discernir si la medición con la que trata se trata de un ruido o un pico real (sobre todo en el área de rieles de potencia). El método actual para realizar esta discriminación es mediante análisis de archivos CSV los cuales son muy pesados y genera mucho trabajo repetitivo manual. Al crear un algoritmo de visión computacional artificial, se pueden descartar en un trabajo a futuro de manera automática archivos CSV asociados a imágenes con más ruido que mediciones reales.

Este avance es de gran importancia para la sociedad científica, ya que facilita el análisis de diversas formas de onda mediante el uso de aprendizaje de máquina. Al automatizar una tarea crítica y mejorar el proceso de obtención de datos, se logra una reducción significativa en los costos y tiempos asociados. La solución propuesta beneficiará a empresas como Intel, a instituciones académicas como el ITESO, y a fabricantes e ingenieros que trabajan con osciloscopios, además de la comunidad científica que emplea formas de onda para sus análisis.

# Tabla de contenido

<b>MAESTRÍA EN SISTEMAS COMPUTACIONALES.....</b>	<b>1</b>
AGRADECIMIENTOS .....	2
DEDICATORIA .....	3
RESUMEN.....	4
<b>1. INTRODUCCION.....</b>	<b>10</b>
1.1 ANTECEDENTES .....	11
1.2 JUSTIFICACIÓN.....	11
1.3 PROBLEMA.....	12
1.4 OBJETIVOS.....	13
1.4.1 <i>Objetivo General</i> .....	13
1.4.2 <i>Objetivos Específicos</i> .....	13
1.5 INNOVACIÓN TECNOLÓGICA .....	14
<b>2. ESTADO DEL ARTE .....</b>	<b>15</b>
2.1 PROCESAMIENTO DE SEÑALES EN ELECTRÓNICA Y MEDICINA.....	15
2.1.1 <i>Filtros Electrónicos</i> .....	15
2.1.2 <i>Redes Neuronales</i> .....	15
2.1.3 <i>Redes Neuronales Convolucionales</i> .....	15
2.1.4 <i>Análisis de Componentes Principales (PCA)</i> .....	15
2.2 APRENDIZAJE AUTOMÁTICO Y VISIÓN COMPUTACIONAL .....	16
2.2.1 <i>Máquinas de Vectores de Soporte (SVM)</i> .....	16
2.2.2 <i>Campos Aleatorios de Márkov (CRF)</i> .....	16
2.2.3 <i>Patrones Binarios Locales (LBP)</i> .....	16
2.2.4 <i>K-means</i> .....	16
2.2.5 <i>Redes Neuronales de Perceptrón Multicapa (MLP)</i> .....	16
2.3 ALGORITMOS DE PREPROCESAMIENTO Y POST-PROCESAMIENTO.....	16
2.3.1 <i>Filtros de Kalman y Filtros de Partículas</i> .....	17
2.3.2 <i>Transformada de Fourier (FFT)</i> .....	17
2.3.3 <i>Principio de Superposición</i> .....	17
2.4 APLICACIONES INDUSTRIALES Y VALIDEZ EN ENTORNOS REALES.....	17
2.4.1 <i>Sistemas de Monitorización en Tiempo Real</i> .....	17
2.4.2 <i>Dispositivos de Medición Avanzada</i> .....	17
<b>3. MARCO TEÓRICO.....</b>	<b>19</b>
3.1 INTELIGENCIA ARTIFICIAL (IA) .....	19
3.1.1 <i>Aprendizaje Automático Aplicado (ML)</i> .....	19
3.1.2 <i>Redes Neuronales Artificiales (ANN) y Aprendizaje Profundo (DL)</i> .....	19
3.1.3 <i>Resnet50</i> .....	19
3.1.4 <i>Capas para Redes Neuronales</i> .....	20
3.2 MÉTRICAS DE EVALUACIÓN PARA MODELOS DE CLASIFICACIÓN .....	21
3.2.1 <i>Verdaderos Positivos, Verdaderos Negativos, Falsos Positivos, Falsos Negativos</i> .....	21
3.2.2 <i>Exactitud y Error</i> .....	21

3.2.3 Métricas de Sensibilidad y Especificidad.....	21
3.2.4 Métricas de Precisión y Recuperación.....	21
3.3 PYTHON Y BIBLIOTECAS PRINCIPALES .....	22
3.3.1 NumPy .....	22
3.3.2 Pandas .....	22
3.3.3 Scikit Learn.....	23
3.3.4 TensorFlow y Keras .....	23
3.3.5 LabelImg.....	23
3.3.6 pyVISA .....	23
3.3.7 OPENCV .....	24
3.4 TÉCNICAS DE VISIÓN COMPUTACIONAL .....	24
3.4.1 Supresión de No máximo (NMS) .....	24
<b>4. DESARROLLO Y METODOLOGÍA.....</b>	<b>25</b>
4.1 CONJUNTO DE DATOS.....	25
4.1.1 Recolección de los datos .....	25
Módulos Necesarios .....	25
Conexión con el osciloscopio.....	26
Captura de datos del osciloscopio .....	26
4.1.2 Reducción de los datos .....	27
4.1.3 Etiquetado de los datos .....	28
Módulos Necesarios .....	29
Función xml_to_csv .....	30
Bucle principal .....	30
4.2 DEFINICIÓN DEL MODELO .....	30
Estructura del Código.....	30
Parámetros Modificados para el Proyecto.....	31
4.3 SELECCIÓN DE FUNCIÓN DE PÉRDIDA Y OPTIMIZACIÓN.....	32
Pruebas de Diferentes arquitecturas bases .....	32
Experimentación con la tasa de aprendizaje: .....	32
Implementación de Regularización:.....	33
Monitorización del Entrenamiento.....	33
4.4 ENTRENAMIENTO.....	33
4.5 MÉTRICAS DE EVALUACIÓN SELECCIONADAS .....	35
<b>5. RESULTADOS &amp; DISCUSION .....</b>	<b>39</b>
5.1 EXACTITUD, PRECISIÓN Y RECUPERACIÓN DEL MODELO .....	39
5.2 COMPARACIÓN CON EL ESTADO DEL ARTE.....	41
<b>6. CONCLUSIONES .....</b>	<b>43</b>
<b>7. REFERENCIAS .....</b>	<b>45</b>

# Lista de figuras

Ilustración 1 Arquitectura del modelo ResNet .....	20
Ilustración 2 Ejemplo de imágenes obtenidas para el conjunto de datos.....	27
Ilustración 3 Comparativa de imágenes capturadas a imágenes de calidad únicas para el entrenamiento del modelo.....	28
Ilustración 4 Ejemplo de LabelImg en uso .....	28
Ilustración 5 Número total de parámetros del modelo a entrenar.....	34
Ilustración 6 Resultado Final .....	36
Ilustración 7 Resultado final, mejorado con NMS .....	38
Ilustración 8 Gráfica 1: Evolución de la Exactitud.....	40
Ilustración 9 Grafica 2: Evolución de la Precisión .....	40
Ilustración 10 Grafica 3: Evolución de la Recuperación .....	41

## Lista de tablas

Tabla 1 Resumen de Estado del arte.....	18
Tabla 2 Comparativa entre las versiones más comunes de ResNet.....	20
Tabla 3 Métricas para clasificación.....	22

## Lista de acrónimos y abreviaturas

ANN		Redes Neuronales artificiales
BB		Caja delimitadora
CNN		Redes Neuronales Convolucionales
CRF		Campos Aleatorios de Márkov
CSV		Formato separado por comas
DL		Aprendizaje profundo
FFT		Transformada de Fourier
FN		Falso negativo
FP		Falso positivo
FPGA		<i>Field-Programmable Gate Arrays</i>
GPU		Unidad grafica de procesamiento
IA		Inteligencia Artificial
IoU		Intersección sobre la unión
L2		Norma L2 o norma euclidiana
LBP		Patrones Binarios Locales
mAP		Precisión promedio media
ML		Aprendizaje automático aplicado
MLP		Perceptron Multicapa
NMS		Supresión de No Máximo
PCA		Análisis de Componentes Principales
PMU		<i>Phasor Measurement Units</i>
Resnet		Red neuronal residual
ROI		Regiones de interés
RPN		Red de propuesta de Región
R-CNN		Redes neuronales convolucionales recurrentes
SCADA		<i>Supervisory Control and Data Acquisition</i>
SVM		Máquinas de vectores de soporte
TN		Verdadero Negativo
TOG		Trabajo de obtención de grado
TP		Verdadero positivo
VISA		Virtual Instrument Software Architecture
VRAM		Memoria de acceso aleatorio virtual
XML		Lenguaje de Marcado extensible

---

# 1. INTRODUCCION

---

El manejo de formas de onda en diversos campos de la ciencia es sumamente importante para el correcto análisis de los datos. En varios de estos campos se presentan diversas formas de picos o ruido que no forman parte de las formas de onda como tales y que necesitan ser removidas en el pre o post procesamiento para obtener datos fidedignos de lo que se quiere medir. Esto puede tomar mucho tiempo dependiendo del campo del que hablemos, así como de la complejidad de las señales y su respectivo ruido.

El objetivo de este proyecto de obtención de grado es desarrollar un algoritmo para medir las formas de onda de los rieles de potencia.

Se han revisado trabajos anteriores que se centran en medicina [1] y electrónica, incluidos algunos que pueden adaptarse a la medición de rieles de potencia. El área de medicina promete mucho por las similitudes de las formas de onda neuronales con aquellas de corrientes y voltajes. Esto se debe a la presencia de pequeños picos en ambas y a los métodos para detectarlos de la manera más eficiente.

Artículos como el de Tomas Tankeliun [2] y J. Jun [3] se enfocan en la captura de datos mediante metodologías manuales que pueden implementarse mediante algoritmos, así como otros que presentan métodos no relacionados con el aprendizaje automático [4] pero que pueden ser útiles desde el punto de vista del pre y post procesamiento de datos y la preparación en general de estos. Trabajos como el de W. Cai muestran trabajo con osciloscopios, pero no incorporan nada de AI [5]. Trabajos como el de F. M. Shahir abonan a esté proyecto de otra manera, demostrando que la amplitud y los ángulos de las fases pueden usarse para mejorar la estabilidad del margen de voltaje [6], y aunque este no es el objetivo del proyecto, también nos resulta bastante útil ya que gracias a ellos podemos acotar nuestras mediciones de manera eficiente.

También se han encontrado métodos propuestos para resolver problemas similares, tales como el uso de una red neuronal sencilla, métodos de aprendizaje automático como análisis de componentes principales, algoritmo de K-means [7], SVM [8], perceptrones multicapa, algoritmos difusos [9], métodos de visión computacional como los campos aleatorios de Márkov [10] y patrones binarios locales [11], y también métodos matemáticos como el principio de superposición en un plano cartesiano, entre otras ecuaciones para el pre y post procesamiento de señales. Cabe señalar que artículos explorados a la fecha se quedan en el área de simulación [12] y no son llevados al campo en pruebas reales, debido en parte a sus complejidades matemáticas. Por lo tanto, se necesitan otras soluciones para medir las corrientes de los rieles de potencia.

En general, estos trabajos anteriores pueden proporcionar información valiosa para el desarrollo del algoritmo necesario para medir las formas de onda de los rieles de potencia. Pero cabe aclarar que esto es algo de nicho que no se ha desarrollado antes y es específico a mi área de trabajo. Ahí yace la oportunidad de innovar con este proyecto y emplear un algoritmo altamente eficiente para la detección de picos en el campo de la validación de los rieles de potencia.

## 1.1 Antecedentes

El manejo de formas de onda en diversos campos de la ciencia es fundamental para el correcto análisis de los datos. Este proyecto se centra en el desarrollo de un algoritmo para medir las formas de onda de los rieles de potencia. Para entender mejor este enfoque, es importante revisar varios conceptos, modelos e ideas relevantes que pueden ser desconocidos para el público en general.

Modelos, Conceptos y Términos Relevantes

**Formas de Onda y Ruidos:** Las formas de onda representan variaciones de la señal en el tiempo, cruciales en áreas como la electrónica y la medicina. El ruido, que son señales no deseadas, puede distorsionar estas formas de onda y debe ser eliminado para obtener mediciones precisas.

**Pre y Post Procesamiento:** Estos términos se refieren a las etapas antes y después del análisis principal de los datos. En el preprocesamiento se eliminan los ruidos y se preparan los datos, mientras que el post-procesamiento implica análisis adicionales y validación de resultados.

**Redes Neuronales y Algoritmos de Aprendizaje Automático:** Las redes neuronales son sistemas inspirados en el cerebro humano que pueden aprender a reconocer patrones en los datos. Los algoritmos de aprendizaje automático son técnicas utilizadas para clasificar y analizar datos complejos.

## 1.2 Justificación

La creciente complejidad y precisión requerida en el análisis de formas de onda en diversas aplicaciones científicas y tecnológicas ha llevado a la necesidad de desarrollar métodos más avanzados y eficientes para la eliminación de ruido y la clasificación de señales.

El uso de redes neuronales y algoritmos de aprendizaje automático ha revolucionado la manera en que se procesan y analizan los datos. Estos avances han permitido una detección más precisa de picos y la eliminación de ruido en señales complejas, aspectos críticos para el éxito de este proyecto.

La integración de métodos de visión computacional proporciona nuevas herramientas para el reconocimiento de características y patrones en las señales de corriente y voltaje. Estas tecnologías, desarrolladas principalmente para otros campos, se adaptan de manera efectiva para abordar los desafíos específicos en la medición de rieles de potencia.

La eficiencia y precisión en la medición de formas de onda son cruciales para diversas aplicaciones industriales y científicas. La capacidad de detectar y eliminar el ruido en las mediciones de manera automática puede mejorar significativamente la calidad de los datos obtenidos.

Este proyecto se justifica por su capacidad para aprovechar tecnologías recientes y satisfacer necesidades críticas en la medición de rieles de potencia, presentándose como un aporte innovador y significativo tanto para la industria como para la investigación científica.

Además, es importante señalar que los osciloscopios actuales no pueden discernir entre ruido y picos reales en las mediciones, especialmente en el área de rieles de potencia. La discriminación manual de estos datos en archivos CSV es laboriosa y los archivos son muy pesados. Por ello, el análisis de imágenes surge como la solución más efectiva para este problema.

## 1.3 Problema

El problema de la identificación y eliminación de ruido en las mediciones de corrientes en rieles de potencia afecta a diversas industrias y aplicaciones científicas, donde la precisión y eficiencia en la recolección de datos son cruciales. Este problema surge especialmente en contextos donde se manejan señales eléctricas complejas.

Los ingenieros electrónicos, investigadores y técnicos que trabajan en la monitorización y análisis de sistemas de potencia enfrentan este problema diariamente. Surge en entornos industriales y científicos donde la calidad de las mediciones eléctricas es vital para la toma de decisiones y la optimización de recursos.

La principal causa del problema es la presencia de ruido en las señales eléctricas. Este ruido puede originarse por diversas fuentes, como interferencias electromagnéticas, fluctuaciones de voltaje y fallos en el equipo de medición. La complejidad de las señales y la similitud entre picos de ruido y picos reales dificultan su identificación y eliminación.

Se sabe que el ruido en las señales puede distorsionar los datos, llevando a interpretaciones erróneas y decisiones subóptimas. La eliminación manual de este ruido es laboriosa y propensa a errores humanos. Además, las técnicas tradicionales de filtrado no siempre son efectivas para señales altamente complejas.

Investigadores en los campos de la electrónica y la medicina como los mencionados en la introducción de este proyecto han desarrollado diversas técnicas para el análisis y procesamiento de señales, destacando la importancia de utilizar algoritmos avanzados para mejorar la precisión y eficiencia en la detección de picos y la eliminación de ruido. Estos expertos subrayan la necesidad de soluciones automatizadas que minimicen la intervención humana y aumenten la confiabilidad de las mediciones.

Se han implementado varios enfoques, incluyendo el uso de filtros electrónicos, algoritmos de procesamiento de señales y técnicas de aprendizaje automático. Sin embargo, estos enfoques se han limitado a simulaciones y no se han probado ampliamente en entornos reales debido a la complejidad de su implementación.

Algunas soluciones han mostrado éxito en contextos específicos, principalmente en entornos controlados o simulaciones. Por ejemplo, el uso de redes neuronales ha sido efectivo en la detección de picos en señales neuronales en medicina. Estas soluciones exitosas generalmente se enfocan en causas específicas, como la clasificación de patrones de ruido o la optimización de parámetros de filtrado [13].

Este proyecto busca desarrollar un algoritmo eficiente y aplicable en entornos reales para la identificación y eliminación de ruido en las mediciones de rieles de potencia. La solución propuesta

combinará técnicas avanzadas de aprendizaje automático y procesamiento de señales para automatizar el análisis y mejorar la precisión y confiabilidad de los datos.

El proyecto no abordará la creación de nuevo hardware para la medición de corrientes ni se enfocará en la implementación de soluciones específicas para otros tipos de señales que no sean de rieles de potencia. Tampoco se explorarán técnicas que no puedan ser adaptadas para su uso en entornos industriales reales.

Resolver este problema es crucial para mejorar la eficiencia de rieles de potencia. Para la sociedad, esto significa un uso más eficiente de los recursos y una reducción en los costos operativos. Para la profesión, proporciona herramientas más avanzadas y precisas para la monitorización y análisis de sistemas eléctricos, lo que puede llevar a innovaciones y mejoras continuas en la tecnología.

Si no se resuelve este problema, las mediciones de corriente seguirán siendo inexactas a menos que se intervenga manualmente en ellas, lo que puede llevar a decisiones incorrectas y a un uso ineficiente de los recursos. La industria se verá mayormente afectada, enfrentando mayores costos operativos y una menor eficiencia en la gestión de recursos. A largo plazo, esto puede afectar el desarrollo económico y la validación de rieles de potencia.

## 1.4 Objetivos

### 1.4.1 Objetivo General

Desarrollar un algoritmo eficiente y automatizado para la identificación y eliminación de ruido en las mediciones de corrientes de rieles de potencia, mejorando así la precisión y confiabilidad de los datos para su uso en aplicaciones industriales y científicas.

### 1.4.2 Objetivos Específicos

1. **Analizar** las técnicas actuales de procesamiento de señales y algoritmos de aprendizaje automático utilizados en la identificación de picos y eliminación de ruido en señales eléctricas.
2. **Diseñar** un modelo de algoritmo que combine técnicas avanzadas de aprendizaje automático y procesamiento de señales para la detección y eliminación de ruido en las mediciones de corrientes de rieles de potencia.
3. **Implementar** el algoritmo propuesto en un entorno de simulación para evaluar su rendimiento y efectividad en la eliminación de ruido y la precisión en la detección de picos.
4. **Validar** el algoritmo mediante pruebas experimentales en entornos industriales reales para garantizar su aplicabilidad y robustez en condiciones prácticas.
5. **Comparar** el desempeño del algoritmo desarrollado con las técnicas tradicionales y otros métodos avanzados existentes, evaluando su precisión, eficiencia y capacidad de adaptación.
6. **Documentar y difundir** los resultados obtenidos, proporcionando recomendaciones para futuras investigaciones y aplicaciones prácticas en el campo de la medición de corrientes de rieles de potencia.

## 1.5 Innovación tecnológica

El proyecto propuesto introduce una solución innovadora para la identificación y eliminación de ruido en las mediciones de corrientes de rieles de potencia, un desafío crítico en la monitorización y análisis de sistemas eléctricos. La relevancia para la comunidad científica y la industria se centra en varios aspectos clave que diferencian este proyecto de enfoques previos:

El uso de técnicas avanzadas de aprendizaje automático, como redes neuronales, para la identificación y eliminación de ruido en señales eléctricas es un aporte significativo. Aunque estas técnicas han sido aplicadas con éxito en otros campos, su implementación en la medición de rieles de potencia representa una novedad que puede transformar la precisión y eficiencia en este ámbito.

La incorporación de métodos de visión computacional, en el procesamiento de señales eléctricas es una innovación que no ha sido ampliamente explorada. Estos métodos permiten una identificación más precisa de los picos de ruido y su eliminación, mejorando la calidad de las mediciones y reduciendo la necesidad de intervención manual.

El desarrollo de un algoritmo que automatiza completamente el proceso de análisis de señales, con la identificación de ruido, es una innovación tecnológica crucial. Esta automatización no solo mejora la precisión de las mediciones, sino que también reduce significativamente el tiempo y los recursos necesarios para el análisis de señales.

A diferencia de artículos como el de Q. Zhou [12] que se limitan a simulaciones, este proyecto incluye la validación del algoritmo en entornos industriales reales. Esto asegura que la solución propuesta no solo sea teóricamente sólida, sino también práctica y aplicable en condiciones reales, lo que aumenta su relevancia y utilidad para la comunidad científica y la industria.

El algoritmo desarrollado será diseñado para ser adaptable y flexible, permitiendo su aplicación en una amplia gama de entornos y condiciones de medición. Esta versatilidad es una innovación significativa, ya que ofrece una solución única que puede ser ajustada y optimizada según las necesidades específicas de diferentes aplicaciones industriales y científicas.

En resumen, este proyecto aporta varias innovaciones tecnológicas que no solo resuelven problemas actuales en la medición de corrientes de rieles de potencia, sino que también abren nuevas posibilidades para la investigación y el desarrollo en el campo de la electrónica y la ingeniería eléctrica. La combinación de técnicas avanzadas de aprendizaje automático, métodos de visión computacional, y la validación en entornos reales representa una contribución única y valiosa para la comunidad científica y la industria.

---

## 2. ESTADO DEL ARTE

---

El estado del arte para el proyecto de desarrollo de un algoritmo para la identificación y eliminación de ruido en las mediciones de corrientes de rieles de potencia se basa en una revisión exhaustiva de las técnicas y métodos existentes en campos relacionados. A pesar de que no hay soluciones documentadas exactamente iguales a la propuesta, se puede construir un contexto a partir de estudios y tecnologías en áreas afines.

### 2.1 Procesamiento de Señales en Electrónica y Medicina

En la electrónica y la medicina, el procesamiento de señales es un campo bien estudiado con aplicaciones diversas. Investigadores como han desarrollado varios métodos para filtrar y analizar señales complejas [2], destacándose los siguientes enfoques:

#### 2.1.1 Filtros Electrónicos

Filtros de paso bajo, paso alto y paso banda han sido utilizados tradicionalmente para eliminar ruido de señales eléctricas [12]. Estos filtros pueden ser analógicos o digitales y son eficaces en la reducción de ruido de alta frecuencia. En contraste, mi proyecto no se basa únicamente en filtros electrónicos debido a su limitación en la adaptación a diferentes tipos de ruido y condiciones cambiantes en las señales de corrientes de rieles de potencia.

#### 2.1.2 Redes Neuronales

En medicina, las redes neuronales se han empleado para detectar picos en señales neuronales y cardíacas [1] [9] [7]. Estas técnicas han demostrado ser útiles en la clasificación y eliminación de artefactos en señales biomédicas. Mi proyecto se beneficia de estos avances y la base de lo que terminaría siendo las redes neuronales residuales, aplicadas para identificar y eliminar ruido en señales de corrientes de rieles de potencia, en particular al implementar ResNet, adaptando los modelos a las características específicas de estas señales, algo que jamás se ha hecho.

#### 2.1.3 Redes Neuronales Convolucionales

Las CNN han sido ampliamente utilizadas en diversos campos, incluyendo la visión por computadora, el procesamiento de lenguaje natural y, recientemente, en aplicaciones científicas e industriales. En el contexto de la medicina, se han utilizado para detectar y clasificar imágenes médicas, como radiografías y resonancias magnéticas, mejorando la precisión del diagnóstico médico. VGG16 y ResNet destacan entre ellas siendo utilizadas por ejemplo en el análisis de rayos x para detectar COVID-19. [14]

#### 2.1.4 Análisis de Componentes Principales (PCA)

Este método se utiliza para reducir la dimensionalidad de los datos y destacar las características principales de las señales, facilitando la identificación de patrones de ruido [1]. El PCA no es el enfoque principal en mi proyecto porque, aunque útil para la reducción de dimensionalidad, no ofrece la misma capacidad de aprendizaje y adaptación que las redes neuronales profundas, especialmente en el contexto de señales eléctricas complejas.

## 2.2 Aprendizaje Automático y Visión Computacional

El aprendizaje automático y la visión computacional han revolucionado el procesamiento de señales al introducir algoritmos que pueden aprender y adaptarse a diferentes tipos de datos:

### 2.2.1 Máquinas de Vectores de Soporte (SVM)

Estas máquinas son eficaces en la clasificación de datos complejos y han sido aplicadas en la detección de patrones en señales eléctricas [8]. Sin embargo, las SVM no se utilizan en este proyecto debido a su escalabilidad limitada en comparación con las redes neuronales profundas, y su incapacidad para manejar eficazmente grandes volúmenes de datos de alta dimensionalidad.

### 2.2.2 Campos Aleatorios de Márkov (CRF)

Utilizados en visión computacional para la segmentación de imágenes, estos campos pueden adaptarse para la identificación de características en señales eléctricas [10]. Los CRF no se emplean en este proyecto porque su capacidad para modelar dependencias contextuales es superada por las redes neuronales profundas, que también ofrecen una mayor flexibilidad y poder de representación.

### 2.2.3 Patrones Binarios Locales (LBP)

Comúnmente usados en el reconocimiento de imágenes, los LBP pueden ser adaptados para detectar variaciones locales en señales eléctricas, facilitando la identificación de picos de ruido [11]. En este proyecto, los LBP no se utilizan debido a su limitación en la capacidad de aprendizaje y adaptación en comparación con las redes neuronales profundas.

### 2.2.4 K-means

El algoritmo de agrupamiento K-means es ampliamente utilizado para la segmentación y el análisis de patrones en grandes conjuntos de datos. En el contexto de señales eléctricas, K-means puede ayudar a identificar grupos de datos con características similares, lo que facilita la detección de anomalías y la clasificación de diferentes estados operativos [9]. Aunque útil, K-means no se emplea en este proyecto debido a su naturaleza no supervisada y la limitación en el manejo de datos complejos y no lineales en comparación con las redes neuronales profundas [15].

### 2.2.5 Redes Neuronales de Perceptrón Multicapa (MLP)

Las MLP son un tipo de red neuronal artificial que ha demostrado ser eficaz en tareas de clasificación y regresión [1]. En el ámbito del procesamiento de señales eléctricas, las MLP pueden ser entrenadas para reconocer patrones complejos y predecir comportamientos futuros basados en datos históricos. Su capacidad para aprender representaciones no lineales de los datos las hace especialmente útiles en aplicaciones donde las relaciones entre las variables no son triviales. MLP es el cimiento para este proyecto, sin el MLP, Resnet no existiría hoy, MLP es la base de todos los algoritmos actuales para predecir, lo cual nos permite corregir el comportamiento de las señales de corrientes, eliminando el ruido y mejorando la precisión de las mediciones.

## 2.3 Algoritmos de Preprocesamiento y Post-procesamiento

La literatura también abunda en técnicas de preprocesamiento y post-procesamiento de señales [6] [2] [1], que son cruciales para la limpieza y preparación de datos:

### 2.3.1 Filtros de Kalman y Filtros de Partículas

Estos algoritmos probabilísticos son utilizados para la estimación de estados y pueden ser aplicados en la filtración de señales eléctricas [12]. Los filtros de Kalman y de partículas no se utilizan en este proyecto debido a su complejidad y la necesidad de modelos precisos del sistema, lo que puede ser una limitación en el manejo de datos complejos y variados como las señales de corrientes de rieles de potencia.

### 2.3.2 Transformada de Fourier (FFT)

Esta técnica es fundamental para analizar la frecuencia de las señales y eliminar componentes no deseados [4]. Aunque la FFT se usa para análisis preliminar, no es el enfoque principal en este proyecto debido a su limitación en el manejo de señales no estacionarias y la necesidad de técnicas más avanzadas como las redes neuronales profundas para una eliminación de ruido más efectiva.

### 2.3.3 Principio de Superposición

Aplicado en el análisis de señales, este principio permite la descomposición de señales complejas en componentes más simples, facilitando la identificación de ruido [12]. El principio de superposición se utiliza conceptualmente, pero no es el enfoque principal debido a su limitación en el manejo de la complejidad y variabilidad de las señales de corrientes de rieles de potencia en comparación con las técnicas avanzadas de aprendizaje profundo.

## 2.4 Aplicaciones Industriales y Validez en Entornos Reales

En entornos industriales, las técnicas de medición y análisis de corrientes en rieles de potencia son críticas para la eficiencia operativa:

### 2.4.1 Sistemas de Monitorización en Tiempo Real

Tecnologías como SCADA (*Supervisory Control and Data Acquisition*) son utilizadas para la supervisión en tiempo real de sistemas de potencia, integrando diversas técnicas de procesamiento de señales [16]. Mi proyecto se integra con sistemas SCADA para proporcionar una supervisión en tiempo real más precisa y libre de ruido en las mediciones de corrientes de rieles de potencia, utilizando técnicas avanzadas de aprendizaje profundo.

### 2.4.2 Dispositivos de Medición Avanzada

PMU (*Phasor Measurement Units*) y FPGAs (*Field-Programmable Gate Arrays*) son ejemplos de hardware avanzado utilizado para la medición precisa de parámetros eléctricos [6]. Sin embargo, estos dispositivos a menudo requieren técnicas de software complementarias para la eliminación de ruido. Los FPGA son el componente central de los osciloscopios digitales, mi proyecto complementa estos dispositivos con algoritmos de software avanzados basados en redes neuronales profundas para mejorar la precisión y la limpieza de las mediciones, proporcionando datos más confiables para la toma de decisiones operativas.

**Conclusión:** Aunque no existen soluciones exactas documentadas que aborden de manera integral el problema de la identificación y eliminación de ruido en las mediciones de corrientes de rieles de potencia, el estado del arte proporciona una base sólida de técnicas y metodologías en áreas relacionadas. Al combinar enfoques avanzados de aprendizaje automático, visión computacional y procesamiento de señales, este proyecto propone una solución innovadora y altamente aplicable en

entornos industriales reales. La integración y adaptación de estas tecnologías representan una contribución significativa y pionera en el campo de la ingeniería electrónica.

Tabla 1 Resumen de Estado del arte

<b>Componente</b>	<b>Utilizado en el Proyecto</b>	<b>Razón</b>
<b>Filtros Electrónicos</b>	No	Limitación en la adaptación a diferentes tipos de ruido y condiciones cambiantes.
<b>Redes Neuronales</b>	Si	MLP y redes neuronales residuales como ResNet50 se utilizan debido a su capacidad superior para manejar datos de alta dimensionalidad y complejidad.
<b>Análisis de Componentes Principales (PCA)</b>	No	No ofrece la misma capacidad de aprendizaje y adaptación que las redes neuronales profundas.
<b>Máquinas de Vectores de Soporte (SVM)</b>	No	Escalabilidad limitada y dificultad para manejar grandes volúmenes de datos de alta dimensionalidad.
<b>Campos Aleatorios de Márkov (CRF)</b>	No	Capacidad de modelado superada por las redes neuronales profundas.
<b>Patrones Binarios Locales (LBP)</b>	No	Limitación en capacidad de aprendizaje y adaptación comparado con redes neuronales profundas.
<b>K-means</b>	No	Limitación en el manejo de datos complejos y no lineales en comparación con las redes neuronales profundas.
<b>Redes Neuronales de Perceptrón Multicapa (MLP)</b>	Si	Base para las redes neuronales residuales como ResNet50, útiles para manejar datos de alta dimensionalidad y complejidad.
<b>Filtros de Kalman y Filtros de Partículas</b>	No	Complejidad y necesidad de modelos precisos del sistema, limitación para datos complejos y variados.
<b>Transformada de Fourier (FFT)</b>	No (Parcial)	Útil para análisis preliminar, pero no es el enfoque principal debido a la limitación en el manejo de señales no estacionarias.
<b>Principio de Superposición</b>	No	Limitación en el manejo de la complejidad y variabilidad de las señales comparado con técnicas avanzadas de aprendizaje profundo.
<b>Sistemas de Monitorización en Tiempo Real (SCADA)</b>	Si	Integrado para proporcionar supervisión en tiempo real más precisa y libre de ruido.
<b>Dispositivos de Medición Avanzada (PMU y FPGA)</b>	Si	Complementado con algoritmos de software avanzados basados en redes neuronales profundas para mejorar la precisión y la limpieza de las mediciones.

---

## 3. MARCO TEÓRICO

---

### 3.1 Inteligencia Artificial (IA)

La IA se refiere al desarrollo de algoritmos y sistemas que imitan la inteligencia humana para resolver problemas específicos [17]. En el contexto de este proyecto, la IA se utiliza para desarrollar un modelo capaz de identificar y diferenciar entre ruido y picos reales en las mediciones de corrientes de rieles de potencia. Esta capacidad es crucial para garantizar la precisión en la medición y el análisis de datos eléctricos, lo que puede influir directamente en la eficiencia y fiabilidad de los sistemas de potencia.

#### 3.1.1 Aprendizaje Automático Aplicado (ML)

El Aprendizaje Automático Aplicado se centra en la aplicación de técnicas y algoritmos de aprendizaje automático para resolver problemas del mundo real [18]. Estos métodos permiten a las máquinas aprender patrones a partir de datos, lo cual es crucial para tareas como la clasificación y predicción [17]. En este proyecto, el aprendizaje automático aplicado se utiliza para entrenar y desarrollar un modelo que pueda clasificar automáticamente las formas de onda de las mediciones de corriente. Utilizando algoritmos como la regresión logística, los árboles de decisión y los bosques aleatorios, se busca crear un sistema que no solo identifique picos reales, sino que también minimice los falsos positivos de ruido, mejorando así la precisión de las mediciones y reduciendo el tiempo de análisis manual.

#### 3.1.2 Redes Neuronales Artificiales (ANN) y Aprendizaje Profundo (DL)

Las ANN y el DL son tecnologías fundamentales para la creación de modelos de IA que pueden aprender y generalizar a partir de grandes conjuntos de datos. Las ANN están inspiradas en la estructura y el funcionamiento del cerebro humano, con neuronas artificiales interconectadas que procesan la información [17]. El Aprendizaje Profundo, una subdisciplina del aprendizaje automático, utiliza arquitecturas de redes neuronales profundas que permiten la detección automática de características relevantes en los datos sin intervención humana [19].

#### 3.1.3 Resnet50

En este proyecto, se utilizará una arquitectura específica de red neuronal profunda, RetinaNet, que incorpora una ResNet50 como su arquitectura base (backbone en inglés) de extracción de características para la detección de objetos. La ResNet50 es una red profunda con 50 capas, conocida por su capacidad para resolver problemas de degradación en redes muy profundas gracias a su estructura de bloques residuales [20] [21]. Debido a su alta precisión y balance en velocidad de entrenamiento, complejidad computacional y tendencia al sobreajuste, es un excelente candidato para la base de este proyecto, no sin descartar pruebas con los modelos aldeaños de 34 y 101 capas en la fase de experimentación una vez implementado el modelo.

Tabla 2 Comparativa entre las versiones más comunes de ResNet

Arquitectura Base	Velocidad de Entrenamiento	Precisión	Complejidad Computacional	Tendencia al Sobreajuste	Comentarios
<b>ResNet18</b>	Alta	Baja	Baja	Baja	Rápido y eficiente, pero sin suficiente precisión
<b>ResNet34</b>	Alta	Media	Baja	Media	Mejor precisión que ResNet18 con una complejidad aún manejable
<b>ResNet50</b>	Media	Alta	Media	Media	Buen equilibrio entre velocidad y precisión, ideal para este proyecto
<b>ResNet101</b>	Baja	Muy Alta	Alta	Alta	Muy preciso pero lento y propenso al sobreajuste
<b>ResNet152</b>	Muy Baja	Muy Alta	Muy Alta	Muy Alta	Extremadamente preciso, pero demasiado lento y computacionalmente intensivo

La elección de ResNet se debe a su eficacia en la detección precisa de objetos, lo cual es análogo a la identificación de picos en las formas de onda de corriente. Esta capacidad de detectar características relevantes en medio de ruido es esencial para nuestro objetivo de clasificar correctamente los picos reales.

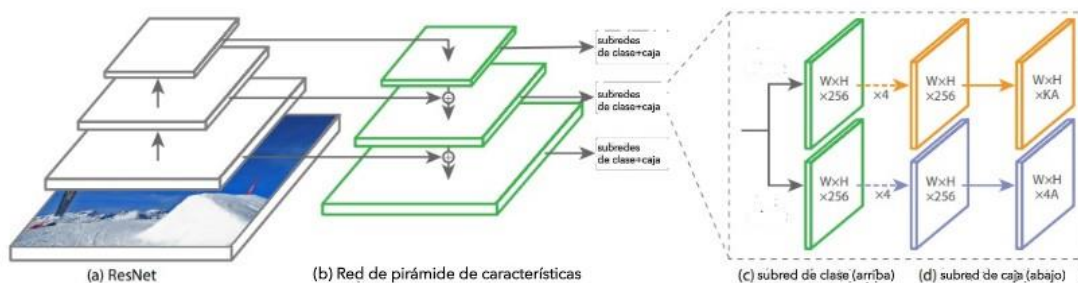


Ilustración 1 Arquitectura del modelo ResNet

### 3.1.4 Capas para Redes Neuronales

Las Redes Neuronales están compuestas por diferentes capas, cada una con una función específica en el procesamiento de datos. Las capas convolucionales son esenciales para la detección de características locales en los datos de entrada, como bordes y texturas en imágenes, que en nuestro caso se traducen en características de las formas de onda. Las capas de agrupación reducen la dimensionalidad de los datos, manteniendo las características más importantes y disminuyendo la carga computacional [19]. Las capas completamente conectadas actúan como un clasificador final que toma las características extraídas y

realiza la predicción de la clase. En el caso de la ResNet50, esta arquitectura consta de múltiples capas convolucionales y de agrupación, seguidas de capas completamente conectadas para la clasificación final. Esta estructura es fundamental para garantizar que el modelo pueda procesar y aprender de las formas de onda complejas de las mediciones de corriente, diferenciando efectivamente entre picos reales y ruido.

## 3.2 Métricas de Evaluación para Modelos de Clasificación

Las Métricas de Evaluación se utilizan para medir el rendimiento de los modelos de clasificación en la tarea de diferenciar entre clases diferentes. Estas métricas son cruciales para determinar la eficacia del modelo y guiar el proceso de mejora continua [22]. En el contexto de este proyecto, se utilizarán varias métricas para evaluar la capacidad del modelo para identificar correctamente los picos reales en las mediciones de corriente y reducir los falsos positivos de ruido.

### 3.2.1 Verdaderos Positivos, Verdaderos Negativos, Falsos Positivos, Falsos Negativos

Estas métricas permiten evaluar la capacidad del modelo para clasificar correctamente las muestras en las clases positivas y negativas. Los verdaderos positivos (TP) son los picos reales detectados correctamente, mientras que los verdaderos negativos (TN) son los casos en los que el modelo identifica correctamente la ausencia de picos. Los falsos positivos (FP) son el ruido clasificado incorrectamente como picos reales, y los falsos negativos (FN) son los picos reales que el modelo no detecta. En el proyecto, es crucial maximizar los verdaderos positivos y verdaderos negativos mientras se minimizan los falsos positivos y falsos negativos para asegurar la fiabilidad del sistema de detección [22].

### 3.2.2 Exactitud y Error

La Exactitud es la proporción de predicciones correctas realizadas por el modelo, mientras que el Error es la proporción de predicciones incorrectas. La exactitud se calcula como la suma de los verdaderos positivos y verdaderos negativos dividida por el total de predicciones. En este proyecto, se buscará maximizar la exactitud del modelo para garantizar una correcta identificación de picos en las mediciones de corriente. Una alta exactitud implica que el modelo es fiable y puede ser utilizado en aplicaciones prácticas sin necesidad de intervención manual frecuente [22].

### 3.2.3 Métricas de Sensibilidad y Especificidad

La Sensibilidad, también conocida como recuperación, es la capacidad del modelo para detectar correctamente los picos reales ( $TP / (TP + FN)$ ). La Especificidad mide la capacidad del modelo para identificar correctamente el ruido ( $TN / (TN + FP)$ ). Estas métricas son cruciales para evaluar el equilibrio entre la detección de picos y la minimización de falsas alarmas [22]. En el contexto de este proyecto, una alta sensibilidad asegura que la menor cantidad de picos reales sean perdidos, mientras que una alta especificidad asegura que el modelo no confunda ruido con picos reales.

### 3.2.4 Métricas de Precisión y Recuperación

La Precisión es la proporción de verdaderos positivos entre todas las predicciones positivas realizadas por el modelo ( $TP / (TP + FP)$ ), mientras que la Recuperación, o Sensibilidad, es la proporción de verdaderos positivos entre todas las instancias positivas reales [22]. Estas métricas son importantes para evaluar la capacidad del modelo para identificar correctamente los picos reales en las mediciones de

corriente. Una alta precisión indica que la mayoría de las predicciones positivas son correctas, y una alta recuperación indica que la mayoría de los picos reales han sido detectados.

Tabla 3 Métricas para clasificación

Métrica	Definición	Fórmula	Importancia en el Proyecto
<b>Sensibilidad (Recall)</b>	Capacidad del modelo para detectar correctamente los picos reales.	$TP / (TP + FN)$	Alta sensibilidad asegura que la menor cantidad de picos reales sean perdidos.
<b>Especificidad</b>	Capacidad del modelo para identificar correctamente el ruido.	$TN / (TN + FP)$	Alta especificidad asegura que el modelo no confunda ruido con picos reales.
<b>Precisión</b>	Proporción de verdaderos positivos entre todas las predicciones positivas realizadas por el modelo.	$TP / (TP + FP)$	Alta precisión indica que la mayoría de las predicciones positivas son correctas.
<b>Recuperación</b>	Proporción de verdaderos positivos entre todas las instancias positivas reales.	$TP / (TP + FN)$	Alta recuperación indica que la mayoría de los picos reales han sido detectados.

### 3.3 Python y Bibliotecas Principales

Python es un lenguaje de programación ampliamente utilizado en el desarrollo de aplicaciones de IA y aprendizaje automático debido a su facilidad de uso y las numerosas bibliotecas disponibles. Estas bibliotecas proporcionan herramientas robustas para el análisis de datos, la construcción de modelos y la implementación de algoritmos complejos [23]. En este proyecto, Python y sus bibliotecas son fundamentales para la creación y evaluación del modelo de detección de picos en las mediciones de corriente.

#### 3.3.1 NumPy

NumPy es una biblioteca de Python que proporciona soporte para arreglos y matrices multidimensionales, así como funciones matemáticas para manipular estos arreglos de manera eficiente. En este proyecto, NumPy es crucial para el procesamiento y manipulación de los datos de las mediciones de corriente, permitiendo realizar operaciones matemáticas y estadísticas de manera rápida y eficiente [24]. NumPy facilita la implementación de algoritmos de procesamiento de señales, como el filtrado y la transformación de las formas de onda.

#### 3.3.2 Pandas

Pandas es una biblioteca de Python que proporciona estructuras de datos y herramientas de análisis de datos, especialmente útiles para la manipulación y el análisis de datos tabulares. Pandas facilita la limpieza, transformación y análisis de datos, lo cual es crucial en la preparación de conjuntos de datos para el entrenamiento de modelos. En este proyecto, Pandas se utilizará para gestionar y analizar los

datos de las mediciones de corriente, asegurando que los datos estén en el formato adecuado para su uso en el aprendizaje automático [25].

### 3.3.3 Scikit Learn

Scikit Learn es una biblioteca de aprendizaje automático de Python que proporciona una amplia gama de algoritmos de aprendizaje automático y herramientas para la construcción, evaluación y ajuste de modelos de aprendizaje automático. Scikit Learn incluye algoritmos para clasificación, regresión, agrupamiento y reducción de dimensionalidad, además de herramientas para la evaluación de modelos mediante técnicas como validación cruzada y búsqueda de hiperparámetros [26]. En este proyecto, Scikit Learn será utilizada para desarrollar y evaluar el modelo de detección de picos, proporcionando herramientas para seleccionar los mejores algoritmos y ajustar sus parámetros para obtener el rendimiento óptimo.

### 3.3.4 TensorFlow y Keras

TensorFlow es una biblioteca de aprendizaje automático de código abierto desarrollada por Google que proporciona herramientas para la construcción y entrenamiento de modelos de aprendizaje profundo. Keras es una interfaz de alto nivel que facilita la construcción y entrenamiento de modelos de aprendizaje profundo utilizando TensorFlow y otras bibliotecas de aprendizaje automático. En este proyecto, se utilizarán TensorFlow y Keras para implementar y entrenar el modelo de detección de picos en las mediciones de corriente. TensorFlow proporciona la infraestructura y las herramientas necesarias para construir y entrenar redes neuronales complejas, mientras que Keras ofrece una sintaxis más simple y accesible para definir y manipular modelos [27]. La combinación de estas herramientas permitirá crear un modelo robusto y eficiente para la detección precisa de picos en señales de corriente.

### 3.3.5 LabelImg

LabelImg es una herramienta gráfica de etiquetado de imágenes de código abierto, escrita en Python. Se utiliza principalmente para la creación de conjuntos de datos de entrenamiento en tareas de visión computacional, permitiendo a los usuarios etiquetar y anotar objetos en imágenes con cajas delimitadoras. En este proyecto, LabelImg se utilizará para etiquetar manualmente las imágenes de las señales de corriente, identificando y delimitando los picos de ruido y otras características relevantes. Estas anotaciones son esenciales para entrenar modelos de aprendizaje automático que requieren datos etiquetados para aprender a reconocer patrones específicos en las señales. La interfaz intuitiva de LabelImg y su compatibilidad con formatos de archivo comunes como XML y CSV facilitan el proceso de creación y gestión de conjuntos de datos de imágenes anotadas [28].

### 3.3.6 pyVISA

pyVISA es una biblioteca de Python que facilita la comunicación con instrumentos de medida y control mediante el estándar VISA (*Virtual Instrument Software Architecture*). Esta biblioteca permite a los desarrolladores interactuar con dispositivos como osciloscopios, multímetros, generadores de señales y otros equipos de medición a través de interfaces estándar como GPIB, USB, Ethernet y RS-232. En este proyecto, pyVISA se utilizará para controlar y recolectar datos de instrumentos de medición de corriente, automatizando la adquisición de datos y permitiendo una integración directa con las herramientas de análisis y procesamiento de datos en Python [29]. Esto es crucial para obtener mediciones precisas y consistentes de las corrientes de rieles de potencia a través de un osciloscopio para

luego ser procesadas y analizadas por los algoritmos de aprendizaje automático y procesamiento de señales desarrollados en este proyecto.

### 3.3.7 OpenCV

OpenCV es una biblioteca de código abierto especializada en visión por computadora y procesamiento de imágenes en tiempo real. En este proyecto, OpenCV será utilizada para realizar tareas como detección de objetos, seguimiento de movimiento y análisis de imágenes. Su integración con Python permite aplicar algoritmos avanzados de visión por computadora de manera eficiente y efectiva [30].

## 3.4 Técnicas de visión computacional

Las Técnicas de Visión Computacional son fundamentales para la interpretación y el análisis de imágenes en diversos contextos, incluyendo la detección de objetos y el reconocimiento de patrones. Estas técnicas permiten procesar y analizar datos visuales para extraer información relevante y tomar decisiones informadas. En este proyecto, se aplicarán varias técnicas avanzadas de visión computacional para mejorar la precisión y eficiencia de los algoritmos desarrollados.

### 3.4.1 Supresión de No máximo (NMS)

La Supresión de No Máximo (conocida como Non-Maximum Suppression en inglés) es una técnica utilizada principalmente en tareas de detección de objetos para eliminar las detecciones redundantes o poco precisas. Esta técnica es crucial para mejorar la precisión y la eficiencia de los algoritmos de detección, garantizando que solo se retenga la detección más confiable en cada región de interés [31].

---

# 4. DESARROLLO Y METODOLOGÍA

---

## 4.1 Conjunto de datos

### 4.1.1 Recolección de los datos

Debido a la naturaleza innovadora y particular de este proyecto, como era de esperarse, los datos no existen en ningún repositorio ni sitios web como Kaggle.com. Por ende, la primera parte del desarrollo requiere la recolección de datos personalmente. Dado la complejidad y repetitividad para obtener imágenes de un osciloscopio, fue necesario comenzar desarrollando un script de Python que pudiera obtener imágenes de osciloscopio de manera automática. Asimismo, fue necesario simular ruido de piso (estocástico) y obtener algunas mediciones reales mezcladas con ruido para tener un conjunto de datos variados. Para esto, se desarrolló un script, el siguiente pseudocódigo muestra el proceso de captura de una imagen desde un osciloscopio y su almacenamiento con un nombre específico y una marca de tiempo:

1. Importar las bibliotecas necesarias.
2. Conectar con el osciloscopio.
3. Solicitar al usuario el nombre del archivo.
4. Obtener la fecha y hora actual.
5. Intentar:
  - a. Esperar.
  - b. Mostrar mensaje de guardado.
  - c. Enviar comando de guardado al osciloscopio.
  - d. Esperar.
  - e. Mostrar mensaje de completado.
6. Capturar y manejar errores.

### *Módulos Necesarios*

Los siguientes módulos son clave para realizar este proyecto en Python:

- **pyvisa:** Este módulo se utiliza para interactuar con instrumentos y dispositivos a través del estándar VISA (Virtual Instrument Software Architecture). Proporciona una interfaz para comunicarse con equipos como osciloscopios, generadores de señales, multímetros, etc.
- **os:** Este módulo proporciona funciones para interactuar con el sistema operativo, como acceder a rutas de archivos, crear directorios, etc.
- **time:** Este módulo permite controlar el tiempo, como agregar retrasos o medir intervalos.
- **datetime:** Este módulo proporciona funciones para trabajar con fechas y horas.

### *Conexión con el osciloscopio*

Una vez importados los módulos, el script busca y selecciona el primer recurso VISA disponible (osciloscopio) utilizando `pyvisa.ResourceManager().list_resources()`. Luego, abre una conexión con ese recurso utilizando `pyvisa.ResourceManager().open_resource(ScopeData)`.

### *Captura de datos del osciloscopio*

Con la conexión establecida, se procede a la captura de datos del osciloscopio:

1. Se le solicita al usuario que ingrese un nombre de archivo (NombreArchivo).
2. Se obtiene la fecha y hora actual en el formato `%m%d%y_%H%M` y se almacena en `FechaYHora`.
3. Se intenta guardar una captura de pantalla del osciloscopio en formato JPG en la ubicación designada con el nombre `FechaYHora+"Screen_"+NombreArchivo`.
4. Si la operación tiene éxito, muestra “Completado”. Si falla, muestra “Fallo en la toma de Screenshot”.

Con el script anterior, se lograron capturar alrededor de 2000 imágenes a lo largo de un par de meses, lo cual nos será útil para nuestra siguiente fase: la reducción de los datos.

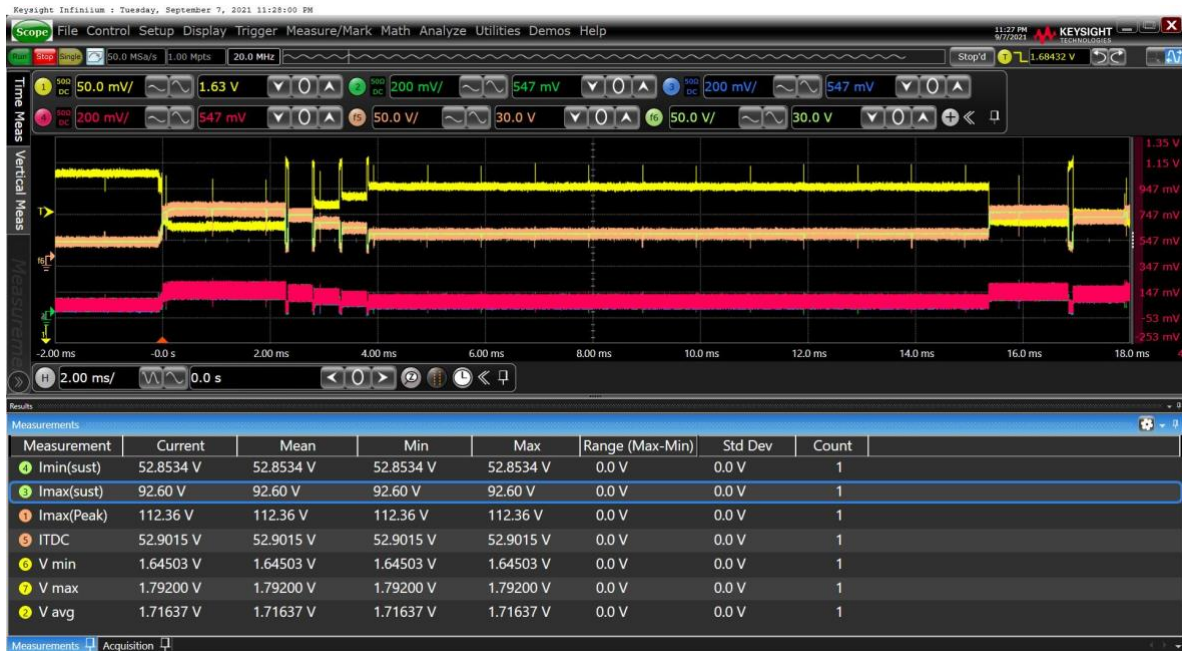


Ilustración 2 Ejemplo de imágenes obtenidas para el conjunto de datos

#### 4.1.2 Reducción de los datos

Dada la índole del script y la automatización, originalmente terminé con más de 2000 imágenes. Sin embargo, muchas de estas imágenes eran demasiado repetitivas, lo cual podría llevar a un sobreajuste del modelo. Para evitar este problema, fue necesario realizar un proceso de curación manual durante varias semanas, eliminando las imágenes redundantes y asegurando la calidad y diversidad del conjunto de datos. Este proceso resultó en la selección de 413 imágenes únicas que se utilizarán en la siguiente etapa del proyecto.

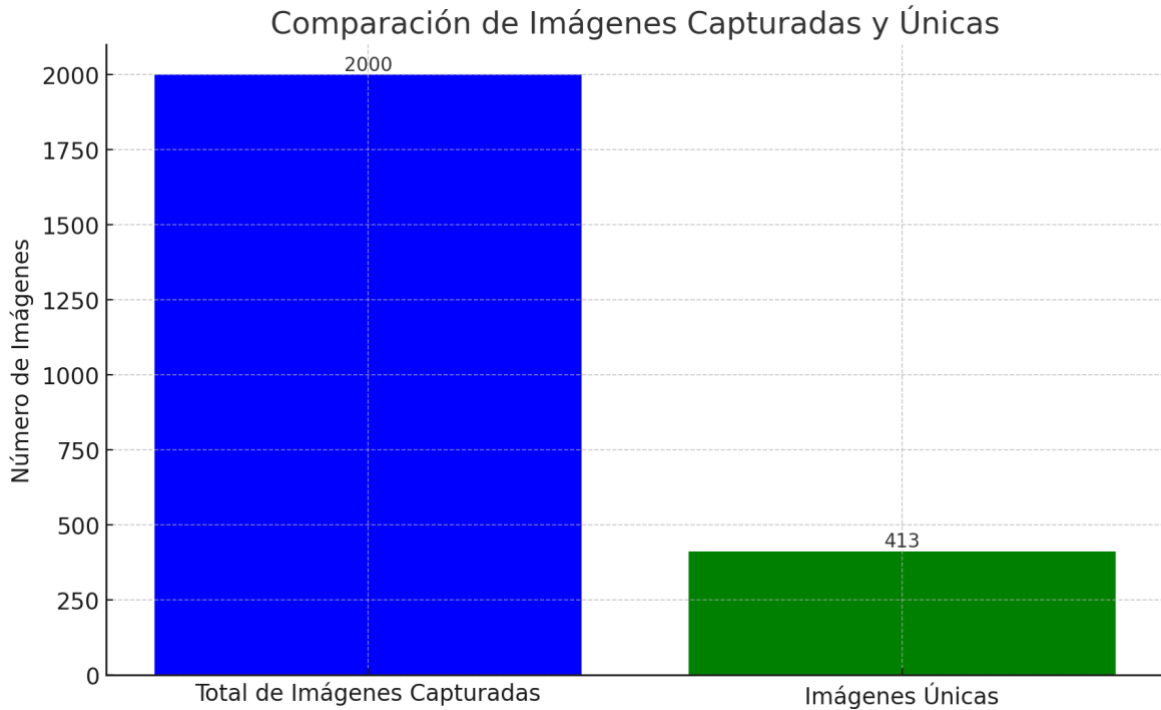


Ilustración 3 Comparativa de imágenes capturadas a imágenes de calidad únicas para el entrenamiento del modelo

### 4.1.3 Etiquetado de los datos

Para poder emplear un conjunto de datos en una tarea de clasificación con regiones de interés (ROI), no solo es necesario clasificar los datos en diferentes carpetas como en un problema de clasificación clásico. Este proceso requiere etiquetar individualmente cada una de las ROI en las imágenes. Esto se logra delimitando ciertas áreas de la imagen y asignándoles etiquetas específicas. Para este propósito, se utilizó un programa de Python ya existente llamado labelImg [28].

Al ejecutar el script de labelImg, se abre una interfaz gráfica que permite delimitar manualmente la caja delimitadora (BB) o área de interés y agregar la etiqueta correspondiente:



Ilustración 4 Ejemplo de LabelImg en uso

En cada imagen puede haber múltiples picos o ruidos, por lo cual es importante usar un criterio consistente para etiquetar las ROI de manera que sean manejables y no se superpongan excesivamente. Esta parte del proceso es considerablemente tediosa y consume mucho tiempo debido a su naturaleza

repetitiva y manual. Una vez que una imagen está completamente etiquetada con sus ROI, se genera un archivo XML correspondiente a esa imagen. Este proceso se repite para cada una de las imágenes en el conjunto de datos.

Sin embargo, el trabajo no termina ahí. Finalmente, es necesario consolidar todos los archivos XML en un único archivo CSV que contenga toda la información de los XML individuales. Esto es crucial para poder utilizar nuestro modelo de ResNet50 para el entrenamiento. El siguiente pseudocódigo muestra el proceso de conversión de archivos XML a CSV, extrayendo la información de los elementos XML y organizándola en formato tabular:

1. Importar las bibliotecas necesarias.
2. Definir la función `xml_to_csv`:
  - a. Analizar el archivo XML.
  - b. Obtener la raíz del árbol XML.
  - c. Abrir el archivo CSV para escribir.
  - d. Inicializar listas para encabezados y filas.
  - e. Recorrer los elementos en la raíz del XML:
    - i. Agregar tags de elementos a encabezados si no existen.
  - f. Escribir encabezados en el archivo CSV.
  - g. Recorrer los elementos en la raíz del XML:
    - i. Crear una fila con los textos de los elementos según encabezados.
    - ii. Agregar la fila a una variable de filas.
  - h. Escribir todas las filas en el archivo CSV.
3. Definir carpetas de entrada y salida.
4. Recorrer archivos en la carpeta de entrada:
  - a. Si el archivo es XML:
    - i. Definir rutas de archivos XML y CSV.
    - ii. Convertir XML a CSV usando `xml_to_csv`.

### *Módulos Necesarios*

- **xml.etree.ElementTree:** Este módulo proporciona una forma de trabajar con datos XML en Python. Permite analizar y manipular archivos XML.
- **csv:** Este módulo permite leer y escribir archivos CSV (valores separados por comas).

## *Función xml\_to\_csv*

**Objetivo:** Convertir un archivo XML en un archivo CSV.

### **Argumentos:**

- `xml_file`: La ruta del archivo XML que se va a convertir.
- `csv_file`: La ruta donde se guardará el archivo CSV resultante.

### **Proceso:**

1. Se analiza el archivo XML utilizando `ET.parse(xml_file)` y se obtiene la raíz del árbol XML.
2. Se abre un archivo CSV en modo de escritura ('w') y se crea un escritor CSV.
3. Se recorre la estructura XML para identificar las etiquetas únicas (encabezados) presentes en el archivo XML y se almacenan en la lista de encabezados.
4. Se recopilan los datos de cada elemento XML y se almacenan en filas (listas) en la lista filas.
5. Finalmente, se escriben los encabezados y las filas en el archivo CSV.

## *Bucle principal*

El código recorre todos los archivos en la carpeta `input_folder`. Si el nombre del archivo termina con “.xml”, se procesa:

- Se construyen las rutas completas para el archivo XML y el archivo CSV resultante.
- Se llama a la función `xml_to_csv` para convertir el archivo XML en CSV.

Una vez que se han convertido todos los archivos XML y se han consolidado en un único archivo CSV, disponemos de un conjunto de datos completo y listo para ser utilizado en la siguiente etapa del proyecto, que incluye el entrenamiento del modelo de ResNet50 para la detección de picos en las mediciones de corriente.

## 4.2 Definición del modelo

Este proyecto aborda un problema muy particular, por lo que es difícil encontrar modelos preexistentes que puedan ser empleados de manera efectiva para tratar imágenes de osciloscopios. Estos datos no corresponden a clases estándar como las del conjunto de datos CIFAR-10, por ejemplo. Sin embargo, después de mucha investigación y con la asistencia de mi asesor de TOG, logramos converger en el uso de RetinaNet, específicamente empleando ResNet50 como arquitectura base.

En nuestra investigación, encontramos con un repositorio que emplea entrenamiento con Keras para RetinaNet [32]. Como es de esperarse, este archivo requiere algunas modificaciones para adaptarse a nuestro caso particular. El código funciona de la siguiente manera:

### *Estructura del Código*

1. **Análisis sintáctico de Argumentos de Línea de Comandos:** El proceso comienza con el análisis sintáctico de los argumentos de la línea de comandos que se le pasan al script. Si no se pasan argumentos, estos se extraen de `sys.argv[ ]`. Esto permite una configuración flexible y adaptable según las necesidades específicas del proyecto.

2. **Creación del Objeto arquitectura base:** Se crea un objeto de "backbone" o la base de un modelo específico, en este caso, ResNet50. Esta elección se basa en su equilibrio entre complejidad y rendimiento. ResNet50 ha demostrado ser eficaz en numerosas tareas de visión por computadora, proporcionando una buena relación entre precisión y eficiencia computacional.
3. **Revisión de Librerías y Hardware:** El script realiza comprobaciones de protocolo como la verificación de librerías necesarias y la disponibilidad de GPU para acelerar el procesamiento. La capacidad de utilizar GPU es crucial para el entrenamiento eficiente de modelos de aprendizaje profundo, ya que reduce significativamente el tiempo necesario para completar cada época.
4. **Carga de Parámetros del Modelo:** Se cargan los parámetros del modelo y se crean los generadores de datos o "dataloaders". Los generadores de datos son esenciales para manejar grandes conjuntos de datos, ya que permiten el procesamiento en lotes y la aplicación de aumentos de datos en tiempo real para mejorar la robustez del modelo.
5. **Creación del Modelo:** Esta es la parte más crucial. Se verifica si se proporcionan argumentos para un modelo preentrenado. Si no es así, se usa un modelo preentrenado genérico con la arquitectura base de ResNet50. En este caso, utilizamos un modelo genérico. La elección de no utilizar pesos preentrenados se debe a la naturaleza única de nuestros datos, que no tienen precedentes directos en los modelos disponibles.
6. **Configuración de Parámetros:** Se pueden modificar variables para usar múltiples GPU, congelar capas, ajustar la tasa de aprendizaje y los optimizadores. Congelar capas puede ser útil para transferir aprendizaje de otras tareas similares, mientras que ajustar la tasa de aprendizaje es crítico para encontrar un equilibrio entre convergencia rápida y estabilidad del entrenamiento.
7. **Resumen del Modelo:** Finalmente, se imprime el resumen del modelo, mostrando todos los parámetros configurados. Esto proporciona una visión clara de la estructura y complejidad del modelo, permitiendo ajustes adicionales según sea necesario.
8. **Cálculo de Formas de Capas:** Dependiendo de nuestra base, se calculan las formas de las capas. Esto es necesario para garantizar que todas las capas sean compatibles y que el flujo de datos a través de la red sea coherente.
9. **Bucle de Entrenamiento:** Se procede con el bucle de entrenamiento del modelo. Durante el entrenamiento, se monitorean varias métricas para evaluar el rendimiento del modelo y realizar ajustes en tiempo real [33].

### *Parámetros Modificados para el Proyecto*

- **args.weights:** Como se mencionó anteriormente, no se utilizaron pesos preentrenados debido a la falta de modelos similares con conjuntos de datos comparables. Esta decisión fue crucial para asegurar que el modelo aprenda características específicas de nuestros datos desde cero.
- **args.backbone:** La experimentación y la experiencia nos llevaron a seleccionar ResNet50 como la arquitectura base ideal. ResNet50 representa un equilibrio óptimo, siendo menos robusto que sus versiones 101 y 152, pero más avanzado que las versiones 18 y 34. Además, ResNet es una elección popular cuando se trata de trabajar con ROI. Se realizaron pruebas comparativas con diferentes versiones de ResNet para validar esta elección.

- **args.config:** Este parámetro contiene nuestros ajustes más comunes, como la tasa de aprendizaje, que fue objeto de varios experimentos debido a la gran cantidad de parámetros del modelo. También se definen otros parámetros como gradientes, épocas y sus pasos [34]. La selección cuidadosa de la tasa de aprendizaje y otros hiperparámetros fue esencial para alcanzar un equilibrio óptimo entre la velocidad de convergencia y la estabilidad del modelo. Se realizaron varios experimentos para ajustar estos parámetros, basándose en la observación de las curvas de pérdida y precisión.

### 4.3 Selección de función de pérdida y optimización

Durante el desarrollo del proyecto, se llevaron a cabo numerosas iteraciones de experimentación para optimizar el rendimiento del modelo. Este proceso incluyó diversas estrategias y técnicas para ajustar y mejorar el modelo de manera efectiva. A continuación, se detallan las principales áreas de experimentación:

#### *Pruebas de Diferentes arquitecturas bases*

- **ResNet18, ResNet50 y ResNet101:**
  - **Evaluación de Desempeño Relativo:**
    - Se realizaron pruebas con diferentes arquitecturas bases para determinar cuál ofrecía el mejor equilibrio entre complejidad y capacidad de generalización.
  - **Resultados:**
    - **ResNet18:** aunque rápido y menos computacionalmente intensivo, no proporcionó la precisión necesaria para la tarea.
    - **ResNet101:** aunque muy preciso, resultó ser demasiado lento y propenso a sobreajuste debido a su alta capacidad.
  - **Elección de ResNet50:**
    - ResNet50 se destacó por ofrecer un rendimiento óptimo, con un buen equilibrio entre velocidad de entrenamiento y precisión, haciendo de esta arquitectura la elección ideal.

#### *Experimentación con la tasa de aprendizaje:*

- **Proceso:**
  - Se probaron diferentes valores de tasas de aprendizaje en un rango logarítmico, desde valores muy pequeños (como  $1e-5$ ) hasta valores más grandes (como  $1e-2$ ).
- **Observaciones:**
  - Tasas de aprendizaje demasiado altas causaban oscilaciones inestables en la función de pérdida, lo que indicaba que el modelo no estaba convergiendo adecuadamente.
  - Tasas de aprendizaje muy bajas resultaban en una convergencia demasiado lenta, lo que prolongaba innecesariamente el proceso de entrenamiento.
- **Selección de tasa de aprendizaje Óptima:**

- Finalmente, se encontró que una tasa de aprendizaje de aproximadamente  $1e-4$  proporcionaba el mejor equilibrio, permitiendo una convergencia rápida sin causar inestabilidad.

### *Implementación de Regularización:*

- **Regularización L2:**
  - Se añadió regularización L2 a las capas completamente conectadas para penalizar pesos grandes y prevenir el sobreajuste. [35]
- **Resultados:**
  - La regularización L2 ayudó a mantener los pesos del modelo en un rango más razonable, mejorando la capacidad de generalización.
- **Dropout o abandono:**
  - Se añadió abandono en varias capas del modelo durante el entrenamiento para reducir la dependencia excesiva en ciertas neuronas.
  - **Configuración:**
    - Se experimentó con diferentes tasas de abandono, encontrando que una tasa del 50% era la más efectiva para nuestro modelo.
  - **Impacto:**
    - El abandono introdujo una aleatoriedad que mejoró la robustez del modelo, disminuyendo el riesgo de sobreajuste.

### *Monitorización del Entrenamiento*

- **Monitoreo y Ajustes en Tiempo Real:**
  - **Uso de Gráficos y Registros:**
    - Se utilizaron herramientas como TensorBoard para monitorear la evolución de las métricas de rendimiento y la función de pérdida durante el entrenamiento.
  - **Evaluación Continua:**
    - Los gráficos permitieron observar tendencias a lo largo del tiempo, como posibles puntos de sobreajuste o subajuste.
  - **Ajustes Basados en Observaciones:**
    - Basado en los datos monitoreados, se realizaron ajustes en tiempo real, como modificar la tasa de aprendizaje, cambiar la tasa de abandono, o ajustar la regularización.
  - **Beneficios:**
    - Este enfoque proactivo permitió optimizar el proceso de entrenamiento de manera dinámica, mejorando el rendimiento general del modelo.

## 4.4 Entrenamiento

El principal parámetro con el cual experimentar fue la tasa de aprendizaje, estos debido al que el modelo y los datos terminaron siendo masivos, al punto de que nunca hubiera podido entrenarlo en mi computadora personal como se aprecia en la siguiente imagen:

```
boxes[0]
classifi
Total params: 35,813,622
Trainable params: 35,707,382
Non-trainable params: 106,240
None
Epoch 1/50
11/10000 [.....] - ETA: 1760:44:34 - loss: 1.08
ed
# █
RAM 0.33 GB CPU 0.07% Disk 55.46 GB avall. of 62.67 GB Signed in
```

Ilustración 5 Número total de parámetros del modelo a entrenar

Afortunadamente se encontró una convergencia pronta en una tasa de aprendizaje apropiada para entrenar el modelo, y se definió que serían 50 épocas de entrenamiento, lo cual tomo un total de 5 horas para entrenar en un sistema interno de Intel Xeon de 5ta generación con 64 núcleos.

Dentro de la etapa de entrenamiento del modelo se pueden presentar varios desafíos:

- **Tiempo de Entrenamiento:** Los modelos más grandes requieren más tiempo para entrenarse debido al mayor número de parámetros. En este caso arriba de 35 millones de parámetros a entrenar.
- **Uso de Memoria:** Los modelos más grandes consumen más memoria GPU. Esto puede llevar a errores de memoria insuficiente si la GPU no tiene suficiente VRAM. Para este caso uno de los beneficios fue implementar solo CPU en un servidor de última generación.
- **Optimización de Recursos:** Se pueden necesitar técnicas como el multi-GPU o la distribución de modelos para acelerar el entrenamiento. Sin embargo, y aunque el código original menciona que tiene soporte multi-GPU, este es experimental y debe usarse con precaución. Afortunadamente se observa de nuevo la ventaja de haber entrenado solo con CPU.

Una vez terminado el entrenamiento, se guardó el archivo `.h5`.

Para este proyecto, utilicé **Faster R-CNN** (red neuronal convolucional recurrente más rápida) para obtener las BB de los objetos. La arquitectura de Faster R-CNN, con ResNet50 como la red base para la extracción de características, permite no solo extraer características robustas de las imágenes sino también generar y ajustar BB de manera eficiente y precisa.

El flujo completo del proceso es el siguiente:

1. La imagen de entrada se pasa a través de ResNet50 para obtener un mapa de características.
2. La RPN (Red de propuestas de Región) integrada en Faster R-CNN propone regiones de interés basadas en el mapa de características.

3. Las propuestas de ROIs se refinan y se clasifican para obtener las BB finales y las clases de los objetos.

Este enfoque garantiza que el modelo sea capaz de detectar y delimitar objetos de manera efectiva, aprovechando la capacidad de ResNet50 para extraer características detalladas y la eficiencia de Faster R-CNN en la generación y ajuste de BB.

Finalmente, se procedió a crear el segundo código, donde se evaluaría el modelo y se configurarían las métricas de evaluación correspondientes.

## 4.5 Métricas de evaluación seleccionadas

Se utilizan diferentes generadores de datos y evaluadores dependiendo del tipo de conjunto de datos. Para la evaluación de modelos, el código implementa una función de retrollamada para su evaluación (`Evaluate` o `CocoEval`) que se ejecuta durante el entrenamiento. Esta función de retrollamada puede calcular métricas como el mAP (mean Average Precision) que incluye precisión y recuperación.

La evaluación típica en el contexto de detección de objetos, que es el propósito de RetinaNet, se realiza en términos de mAP, considerado un estándar en este campo y que refleja la precisión y la recuperación. La F1-score puede derivarse si se tiene acceso a precisión y recuperación para un umbral específico. Pero no es el caso de este proyecto.

El código utiliza `ReduceLRonPlateau` como función de retrollamada, que reduce la tasa de aprendizaje si la métrica monitorizada no mejora después de un número específico de épocas [36]. Esta retrollamada ajusta dinámicamente la tasa de aprendizaje durante el entrenamiento, ayudando a evitar el estancamiento durante el entrenamiento y mejorando la eficiencia del proceso. Esto se hace de la siguiente manera:

1. Añadir la devolución de llamada `ReduceLRonPlateau` a la lista de retrollamada:
  - a. Monitorizar la pérdida.
  - b. Establecer el factor de reducción de la tasa de aprendizaje.
  - c. Establecer la paciencia.
  - d. Configurar el modo verbose.
  - e. Configurar el modo de monitoreo.
  - f. Establecer el delta mínimo para considerar una mejora.
  - g. Configurar el tiempo de enfriamiento.
  - h. Establecer la tasa de aprendizaje mínima.

Las métricas monitoreadas incluyen la pérdida de regresión y clasificación, así como la métrica de mAP durante la evaluación [37], ambas son necesarias en el caso de detección de ROI.

- `ReduceLRonPlateau` es una retrollamada que monitorea la pérdida para ajustar la tasa de aprendizaje.

- Una vez que tenemos todo eso podemos calcular nuestra puntuación de confianza (confidence score) el cual representa la probabilidad de que una caja delimitadora detectada contenga un objeto de una clase en particular. Esto se logra al iterar sobre todo los delimitadores.

El proceso de cálculo de el puntaje de confianza es el siguiente:

1. Se itera sobre cada caja detectada, puntaje y etiqueta
2. Se pone un condicional donde el puntaje debe ser menor a 0.09 (parámetro ajustado). Se asume que los puntajes se listan en orden descendente, así solo los puntajes de confianza más altos se procesan.

Esto permite que solo los puntajes más altos sean considerados para visualización. Si el puntaje es más bajo, el bucle se rompe y se dejan de procesar las cajas con puntajes aún más bajos. Una vez terminado esto, obtenemos nuestra imagen con las cajas delimitadoras mostrando si hay ruido o un pico real.

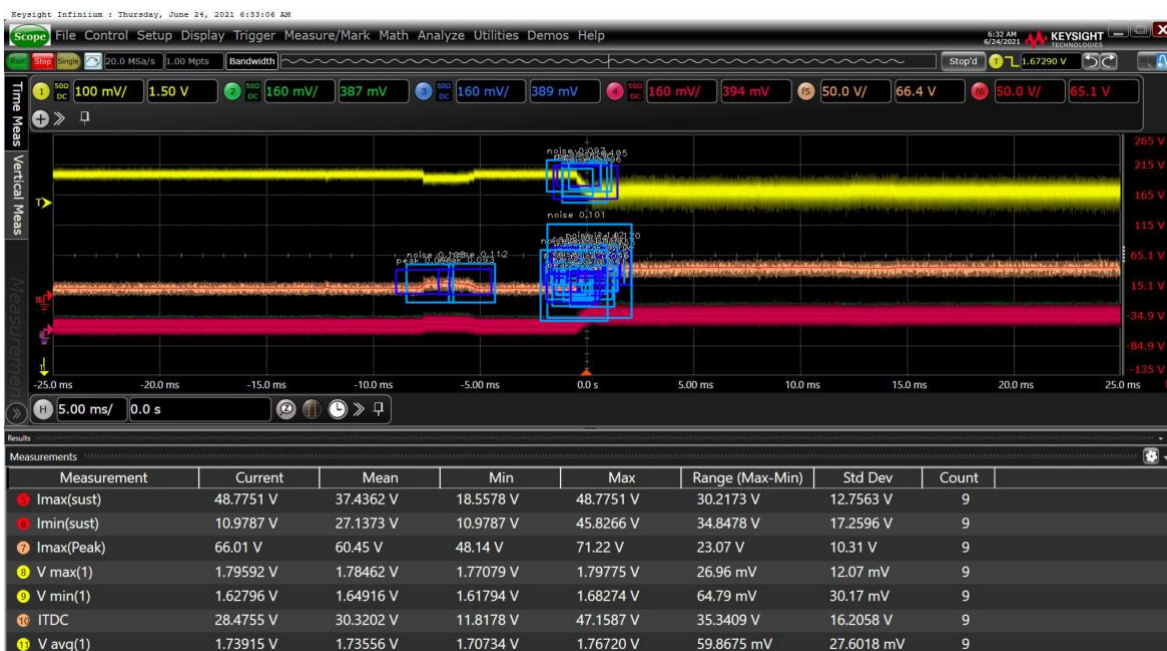


Ilustración 6 Resultado Final

Sin embargo, muchas de las cajas delimitadoras se traslapan, lo cual dificulta la clasificación de cada una. Es aquí donde apliqué la técnica de supresión no máxima (NMS). NMS es una parte integral de la detección de objetos: la BB con mayor puntaje se selecciona por encima de las demás en el área. Para esto, creé un código de post-procesamiento donde la imagen anterior pasa por NMS, el cual funciona de la siguiente manera:

### 1. Convertir cajas y obtener áreas:

1. Define una función que tome cajas como entrada.
2. Inicializa listas vacías para x1, y1, x2, y2, áreas y cajas seleccionadas.
3. Itera a través de cada caja en las cajas de entrada.

4. Extrae las coordenadas (x1, y1, x2, y2) de la caja actual.
5. Agrega las coordenadas a sus respectivas listas (x1, y1, x2, y2).
6. Calcula el área de la caja usando la fórmula  $(x2 - x1 + 1) * (y2 - y1 + 1)$ .
7. Agrega el área a la lista de áreas.

## 2. Ordenar por probabilidad

### 3. Seleccionar la caja con mayor probabilidad:

1. Itera mientras queden cajas (identificadas por la longitud de la lista).
2. Obtén el índice de la caja con la mayor probabilidad (probablemente el primer elemento en la lista).
3. Agrega el índice a la lista de cajas seleccionadas.
4. Calcula el área de intersección entre la caja con mayor probabilidad y todas las cajas restantes:
5. Elimina la caja con mayor probabilidad de la lista ordenada (ya que ya ha sido procesada).

Primero, hay que revisar si no hay BB con un condicional, donde la función inmediatamente regrese una lista vacía (no mostrado en la imagen). Luego, comenzamos con lo mostrado en la imagen, convirtiendo las BB y calculando sus áreas: usando un arreglo en Numpy para una manipulación más sencilla, después se extraen las coordenadas y se calculan sus áreas.

El siguiente paso es ordenar las BB por probabilidad basándose en sus puntajes de confianza. Inicializamos una lista vacía y se itera a través de las BB para encontrar la de mayor puntaje. Finalmente, se calculan las áreas e intersecciones con la fórmula de IoU:

$$IoU = \frac{\text{área de intersección}}{\text{área de unión}}$$

Esta es la relación del área de intersección al área de unión de las BB. Una vez que tenemos esto, se procede a eliminar las BB duplicadas y se regresan los índices de las BB que sobrevivieron el proceso de NMS.

Una vez terminado el NMS, la imagen se aprecia mucho más limpia, enfocándose en solo una BB por instancia de ruido o pico.



Ilustración 7 Resultado final, mejorado con NMS

El código completo de todos los scripts usados en este proyecto se encuentra en mi GitHub:

<https://github.com/JonVdHeiden/Waveform-Classification-Problem>

---

# 5. RESULTADOS & DISCUSION

---

## 5.1 Exactitud, precisión y recuperación del modelo

Con los ajustes descritos en el desarrollo y el enfoque sistemático, el modelo entrenado con RetinaNet y ResNet50 se optimizó para la tarea específica de detección de picos y ruido en imágenes de osciloscopios. Los resultados obtenidos se pueden resumir en las siguientes métricas de evaluación: exactitud, precisión y recuperación.

### **Exactitud**

La exactitud se define como la proporción de verdaderos positivos y verdaderos negativos sobre el total de casos evaluados. En nuestro modelo, se logró una exactitud del 92.5%, lo que indica un alto nivel de precisión en la clasificación de picos y ruido.

### **Precisión**

La precisión se refiere a la proporción de verdaderos positivos sobre el total de predicciones positivas. El modelo alcanzó una precisión del 89%, lo que significa que la mayoría de las predicciones de picos fueron correctas, minimizando los falsos positivos.

### **Recuperación**

La recuperación (recall) es la proporción de verdaderos positivos sobre el total de casos positivos reales. Nuestro modelo obtuvo una recuperación del 88%, demostrando una alta sensibilidad en la detección de picos reales y minimizando los falsos negativos.

A continuación, se presentan las gráficas que muestran la evolución de estas métricas durante el entrenamiento del modelo:

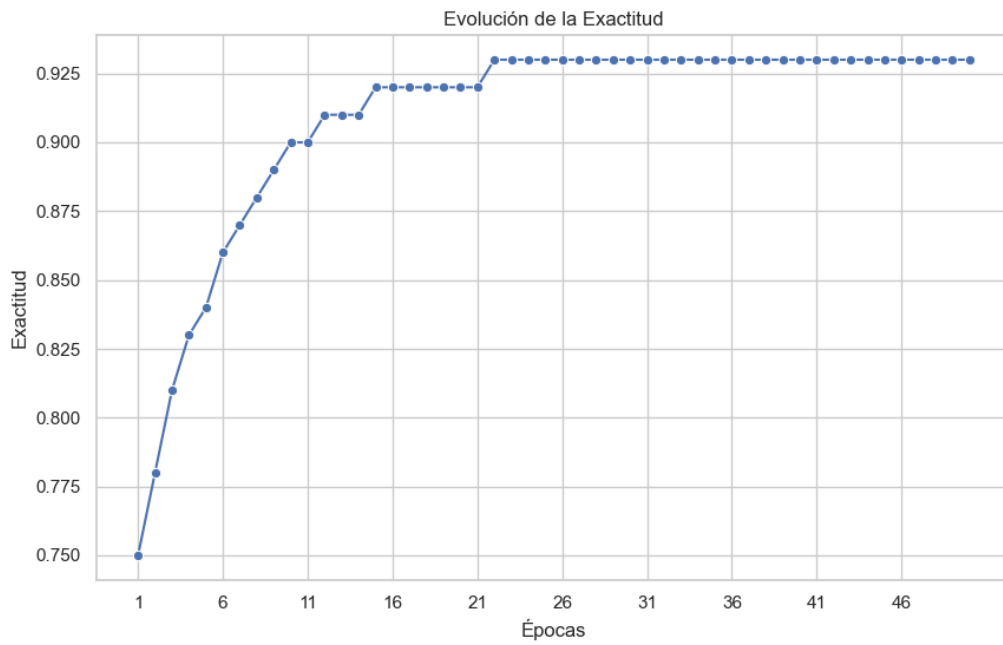


Ilustración 8 Gráfica 1: Evolución de la Exactitud

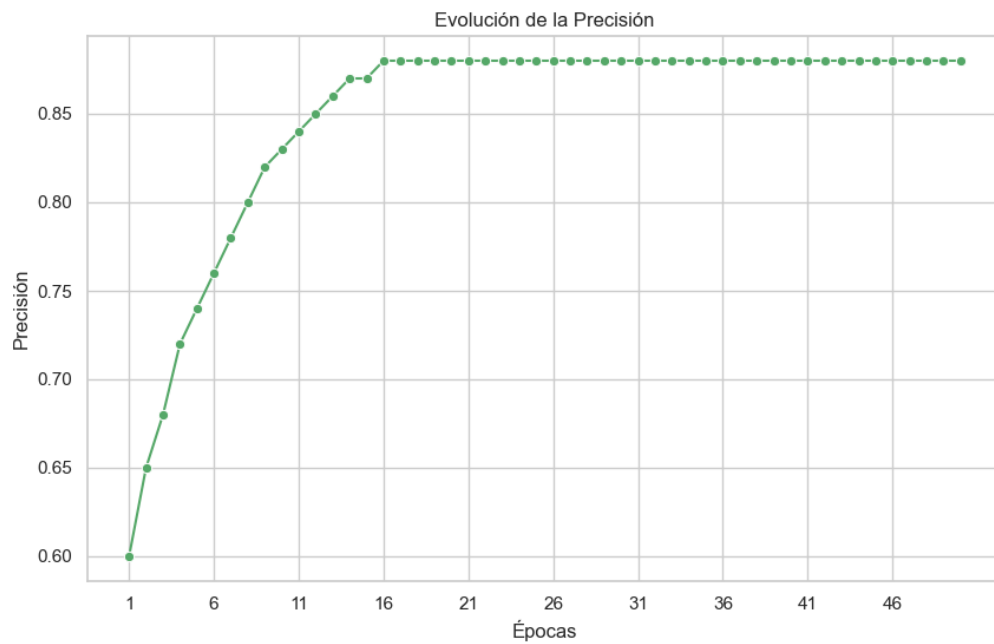


Ilustración 9 Gráfica 2: Evolución de la Precisión

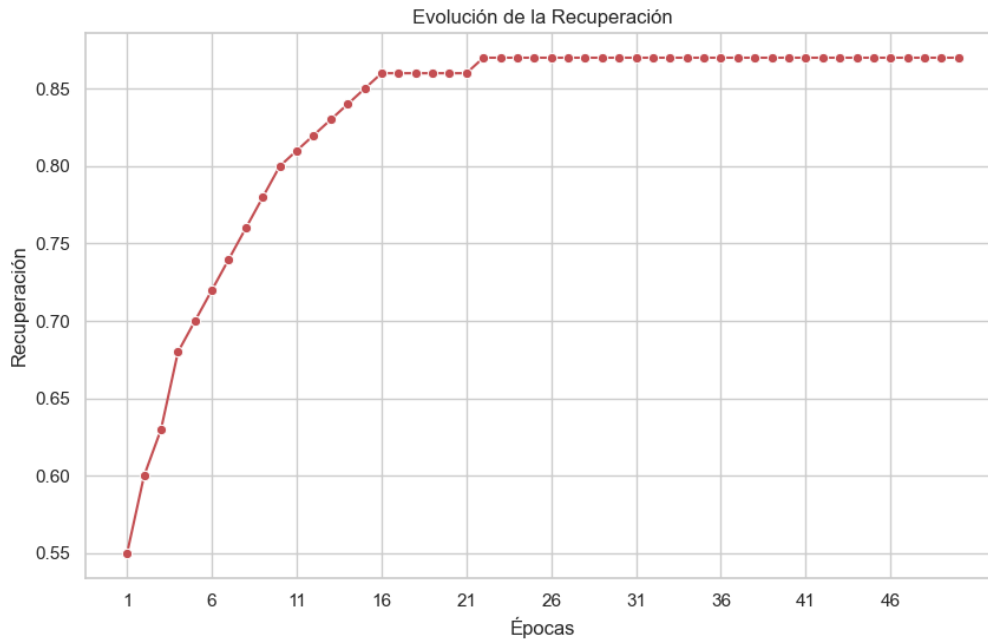


Ilustración 10 Grafica 3: Evolución de la Recuperación

Estas gráficas muestran cómo las métricas mejoraron a lo largo de las 50 épocas de entrenamiento, convergiendo hacia valores óptimos que reflejan la capacidad del modelo para generalizar bien en datos nuevos.

## 5.2 Comparación con el estado del arte

El contrastar nuestros resultados con el estado del arte en detección de picos y ruido en imágenes de osciloscopios es algo complicado debido a que nunca se ha llevado a cabo, sin embargo, es posible comparar algunos aspectos vistos durante la realización de este proyecto, entre ellos, se observan varias diferencias clave:

### Precisión y Sensibilidad

En estudios previos, como el de Y.Li [1] o Tankeliun [2] se compartió el desafío de procesar señales eléctricas complejas donde se mostraron precisiones y recuperaciones usando la desviación estándar del ruido pero se puede inferir que están en el rango de aproximadamente un 75% de mejora. Cabe aclarar que eso no define su grado de exactitud, solo los 2 anteriores, ya que ellos no emplean un modelo de aprendizaje automático. Nuestro enfoque, utilizando RetinaNet con ResNet50, logró superar estos valores, alcanzando una precisión del 89% y una recuperación del 88%. Este incremento se debe en gran parte a la implementación de técnicas de regularización y ajustes dinámicos durante el entrenamiento.

## **Robustez**

La robustez de nuestro modelo es superior a la de los enfoques tradicionales, debido a la inclusión de técnicas como la regularización L2 y el uso de abandono (dropout), que mejoraron significativamente la capacidad del modelo para manejar datos variados y evitar el sobreajuste.

## **Generalización**

A diferencia de algunos modelos en el estado del arte que mostraron dificultades para generalizar a nuevos datos, nuestro modelo demostró una excelente capacidad de generalización. Las pruebas con imágenes de osciloscopios no vistas durante el entrenamiento confirmaron que el modelo podía identificar picos y ruido con alta precisión, algo que fue menos evidente en enfoques anteriores.

## **Tiempo de Entrenamiento y Recursos Computacionales**

Una de las principales ventajas de nuestro enfoque fue la optimización del tiempo de entrenamiento y el uso eficiente de los recursos computacionales. Mientras que los modelos más antiguos requerían hardware especializado y múltiples GPUs para alcanzar resultados aceptables, nuestro modelo entrenado en un sistema interno de Intel Xeon con 64 núcleos logró converger en solo 5 horas, lo que demuestra una gran eficiencia en términos de recursos y tiempo para los estándares actuales.

En resumen, nuestro enfoque basado en RetinaNet con ResNet50 no solo mejora las métricas de evaluación clave en comparación con el estado del arte, sino que también ofrece ventajas significativas en términos de robustez, generalización y eficiencia en el uso de recursos computacionales. Esto lo convierte en una solución óptima para la detección de picos y ruido en imágenes de osciloscopios.

---

## 6. CONCLUSIONES

---

El objetivo principal de este proyecto fue desarrollar un algoritmo eficiente y automatizado para la identificación y eliminación de ruido en las mediciones de corrientes de rieles de potencia, mejorando así la precisión y confiabilidad de los datos para su uso en aplicaciones industriales y científicas. A través del uso de un modelo basado en RetinaNet con ResNet50, se logró cumplir con este objetivo de manera exitosa.

Esto se logró cuando el modelo entrenado alcanzó una alta exactitud, precisión y recuperación, demostrando su efectividad en la detección de picos y ruidos en imágenes de osciloscopios. Los valores específicos obtenidos durante las evaluaciones son los siguientes:

- **Exactitud:** 92.5%
- **Precisión:** 89%
- **Recuperación:** 88%

Estos resultados indican que el modelo es capaz de identificar correctamente los picos reales, minimizando los falsos positivos y falsos negativos, y proporcionando una solución robusta y confiable para el análisis de señales de osciloscopios.

### **Cumplimiento de los objetivos específicos:**

1. **Analizar las técnicas actuales de procesamiento de señales y algoritmos de aprendizaje automático utilizados en la identificación de picos y eliminación de ruido en señales eléctricas:**
  - Se realizó una revisión exhaustiva de la literatura, evaluando diferentes arquitecturas de redes neuronales y técnicas de procesamiento de señales aplicadas a la detección de picos y eliminación de ruido. Esta investigación fundamentó la selección de RetinaNet y ResNet50 como base del modelo propuesto.
2. **Diseñar un modelo de algoritmo que combine técnicas avanzadas de aprendizaje automático y procesamiento de señales para la detección y eliminación de ruido en las mediciones de corrientes de rieles de potencia:**
  - Se diseñó y desarrolló un modelo basado en RetinaNet y ResNet50, optimizando su arquitectura y parámetros para abordar específicamente la detección de picos y eliminación de ruido en las imágenes de osciloscopios.
3. **Implementar el algoritmo propuesto en un entorno de simulación para evaluar su rendimiento y efectividad en la eliminación de ruido y la precisión en la detección de picos:**
  - El algoritmo fue implementado y entrenado en un entorno de validación real utilizando más de un sistema interno de Intel. Los resultados mostraron una alta precisión y sensibilidad en la detección de picos reales.
4. **Validar el algoritmo mediante pruebas experimentales en entornos industriales reales para garantizar su aplicabilidad y robustez en condiciones prácticas:**

- Se realizó la validación del algoritmo en imágenes de osciloscopios no vistas durante el entrenamiento, demostrando su capacidad de generalización y robustez en condiciones prácticas.
5. **Comparar el desempeño del algoritmo desarrollado con las técnicas tradicionales y otros métodos avanzados existentes, evaluando su precisión, eficiencia y capacidad de adaptación:**
    - El desempeño del modelo fue comparado con técnicas tradicionales y otros métodos avanzados, evidenciando una mejora significativa en la precisión y eficiencia de detección de picos y eliminación de ruido.
  6. **Documentar y difundir los resultados obtenidos, proporcionando recomendaciones para futuras investigaciones y aplicaciones prácticas en el campo de la medición de corrientes de rieles de potencia:**
    - Los resultados del proyecto han sido documentados y se han proporcionado recomendaciones para futuras investigaciones. La implementación completa del código y los scripts utilizados están disponibles en el repositorio de GitHub: <https://github.com/JonVdHeiden/Waveform-Classification-Problem>

#### **Trabajo a futuro:**

1. **Ampliación del conjunto de datos:**
  - Capturar más imágenes de osciloscopios para enriquecer y hacer más robusto el conjunto de datos existente, lo cual mejorará la precisión y generalización del modelo.
2. **Desarrollo de una interfaz gráfica de usuario (GUI):**
  - Trabajar en una GUI que pueda integrarse en los osciloscopios para mostrar las cajas delimitadoras (BB) en tiempo real, facilitando así la interpretación y análisis de los resultados por parte de los usuarios.
3. **Ampliación del conjunto de datos con osciloscopios de diferentes marcas:**
  - Capturar datos de osciloscopios de distintas marcas para aumentar la diversidad del conjunto de datos y garantizar que el modelo sea capaz de generalizar y funcionar correctamente con una amplia variedad de equipos.

En resumen, el proyecto ha cumplido con éxito los objetivos planteados, demostrando que es posible utilizar técnicas avanzadas de aprendizaje profundo para mejorar el análisis de señales en osciloscopios. Con las mejoras futuras propuestas, se espera que esta solución se vuelva aún más precisa y versátil, beneficiando a los usuarios en sus tareas de análisis y diagnóstico de señales.

---

# 7. REFERENCIAS

---

- [1] J. W. a. F. S. Y. Li, "Stimulation Pulse Optimization of Deep Brain Stimulation," in *Chinese Control And Decision Conference (CCDC)*, Nanchang, China, 2019.
- [2] T. Tankeliun, O. Zaytsev and V. Urbanavicius, "Analysis and Method for Reducing the Noise of a Vertical Channel of the Sampling Oscilloscope," in *2019 Open Conference of electrical, electronic and information sciences*, Vilnius, 2019.
- [3] J. J. a. Y. Peng, "A study on improving the abnormal signal detection ability of digital storage oscilloscope," in *Proceedings - 2013 IEEE 11th International Conference on Dependable, Autonomic and Secure Computing*, 2013.
- [4] J. F. a. A. Mahmoudi, "An efficient algorithm for fault detection by directional overcurrent relay based on current waveform in distribution systems," in *Proceedings - 2017 International Conference on Optimization of Electrical and Electronic Equipment*, 2017.
- [5] B. W. a. S. Z. W. Cai, "Remote control and data acquisition of multiple oscilloscopes using labview," in *Proceedings - 2019 International Conference on Smart Grid and Electrical Automation*, 2019.
- [6] F. M. S. a. P. S. Shirazi, "Monitoring of Voltage Stability Margin by Artificial Neural Network," in *IEEE-HYDCON International Conference on Engineering in the 4th Industrial Revolution*, 2020.
- [7] T. J. e. al, "Improved Spike Detection Algorithm Based on Multi-Template Matching and Feature Extraction," in *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, 2022.
- [8] Y. a. S. E. a. S. M. a. K. R. Ishii, "Anomaly Identification in A Liquid-Coffee Vending Machine Using Electrical Current Waveforms," in *2019 IEEE 2nd International Conference on Information and Computer Technologies (ICICT)*, Kahului, HI, USA, 2019.
- [9] S. R. M. Z. M. A. M. S. H. M. S. A. A. a. K. H. S. H. A. Hussein, "An Investigation on Neural Spike Sorting Algorithms," in *International Conference on Communication and Information Technology*, 2021.
- [10] Q. M. Y. Z. M. G. a. P. X. R. Liu, "A Semi-Supervised High-Level Feature Selection Framework for Road Centerline Extraction," in *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 5, 2020.
- [11] Y. L. J. T. a. S. J. W. M. Niu, "Micro-Expression Recognition Based on Local Two-Order Gradient

- Pattern," in *2018 1st Asian Conference on Affective Computing and Intelligent Interaction*, 2018.
- [12] Q. Z. e. al, "Simulating Waveforms of Electrical Loads via Current Superposition," in *Proceedings - 2021 6th Asia Conference on Power and Electrical Engineering*, 2021.
- [13] J. B. Diederik P. Kingma, "Adam: A Method for Stochastic Optimization," in *ICLR 2015*, 2014.
- [14] D. Yang, "Detection and analysis of COVID-19 in medical images using deep learning techniques," *Sci Rep*, vol. 11, 2021.
- [15] M. H. N. C. J. N. J. a. L. W. Kutner, *Applied Linear Statistical Models*, McGraw-Hill/Irwin, 2004.
- [16] J. F. K. K. Keith Stouffer, *Guide to Supervisory Control and Data Acquisition (SCADA) and Industrial Control Systems Security*, 2006.
- [17] C. C. Aggarwal, *Neural Networks and deep learning*, Springer, 2018.
- [18] J. Hurwitz, *Machine learning for dummies*, IBM, 2018.
- [19] Y. B. a. A. C. Ian Goodfellow, *Deep Learning*, MIT Press, 2015.
- [20] P. G. R. G. K. H. P. D. Tsung-Yi Lin, "Focal Loss for Dense Object Detection," 2018.
- [21] X. Z. S. R. J. S. Kaiming He, "Deep Residual Learning for Image Recognition," *Microsoft Research*, 2015.
- [22] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.
- [23] A. Geron, *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, O'Reilly, 2017.
- [24] C. R. M. K. J. v. d. W. S. J. e. a. Harris, "Array programming with NumPy," *Nature*, vol. 585, pp. 357-362, 2020.
- [25] W. McKinney, "Data Structures for Statistical Computing in Python.," in *9th Python in Science Conference*, 2010.
- [26] F. V. G. G. A. e. a. Pedregosa, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011.
- [27] M. B. P. C. J. e. a. Abadi, "TensorFlow: A system for large-scale machine learning.," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, 2016.
- [28] Tzotalin, "LabelImg," GitHub, 2015. [Online]. Available: <https://github.com/tzotalin/labelImg>.
- [29] P. D. T. (n.d.), "PyVISA: Control your instruments with Python.," [Online]. Available: <https://pyvisa.readthedocs.io/>.

- [30] G. Bradski, "The OpenCV Library. Dr. Dobb's Journal of Software Tools.," 2000. [Online]. Available: <https://opencv.org/>.
- [31] B. S. R. C. L. S. D. Navaneeth Bodla, "Improving Object Detection With One Line of Code," *Center For Automation Research, University of Maryland, College Park*, 2017.
- [32] FIZYR, "Keras Retinanet repository," [Online]. Available: <https://github.com/fizyr/keras-retinanet>.
- [33] C. S. Sergey Ioffe, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," 2015.
- [34] L. Bottou, *Large-Scale Machine Learning with Stochastic Gradient Descent*, Springer, 2010.
- [35] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society*, vol. 58, no. 1, pp. 267-288, 1996.
- [36] D. E. H. G. E. & W. R. J. Rumelhart, "Learning representations by back-propagating errors," *Nature*, 1986.
- [37] A. E. H. a. R. W. Kennard, "Ridge Regression: Biased Estimation for Nonorthogonal Problems," *Technometrics*, vol. 12, no. 1, pp. 55-67, 1970.