



INSPECCIÓN DE SUBESTACIONES ELÉCTRICAS: YOLOv5 EN LA IDENTIFICACIÓN DE PUNTOS CALIENTES MEDIANTE IMÁGENES TÉRMICAS

ELECTRIC SUBSTATION INSPECTION: YOLOv5 IN HOTSPOT DETECTION THROUGH THERMAL IMAGING

Daniel A. Pérez-Aguilar^{1,*3} , Jair M. Pérez-Aguilar¹ ,
 Andy P. Pérez-Aguilar¹ , Redy H. Risco-Ramos² ,
 Manuel E. Malpica-Rodríguez³ 

Recibido: 11-10-2023, Recibido tras revisión: 27-11-2023, Aceptado: 06-12-2023, Publicado: 01-01-2024

Resumen

Las subestaciones son instalaciones clave dentro de un sistema eléctrico; las fallas intempestivas tienden a causar baja calidad y efectos negativos del suministro eléctrico. Un indicador temprano de posibles fallas en los equipos eléctricos es la aparición de puntos calientes; por lo que su detección y posterior corrección programada evita incurrir en fallas mayores y paradas de operación innecesarias. En esta investigación se realizaron 64 experimentos del algoritmo YOLOv5, con la finalidad de proponer un mecanismo automatizado de visión por computadora para la detección de puntos calientes en imágenes térmicas de subestaciones eléctricas. Los mejores resultados muestran un valor mAP de 81,99 %, los cuales se obtuvieron con el algoritmo YOLOv5m y la aplicación de transfer learning. Estos resultados dejan una base para profundizar y mejorar el desempeño del algoritmo, variando otros hiperparámetros a los considerados en el presente estudio.

Palabras clave: aprendizaje por transferencia, detección de objetos, imágenes térmicas, puntos calientes, subestaciones eléctricas, YOLOv5

Abstract

Substations are key facilities within an electrical system, untimely failures tend to cause low quality and negative effects on the electrical supply. An early indicator of potential electrical equipment failure is the appearance of hot spots; therefore, its detection and subsequent programmed correction avoids incurring in major failures and unnecessary operation stops. In this research, 64 experiments of the YOLOv5 algorithm were carried out, with the purpose of proposing an automated computer vision mechanism for the detection of hot spots in thermal images of electrical substations. The best results show a mAP value of 81.99 %, which were obtained with the YOLOv5m algorithm and the transfer learning application. These results leave a basis to deepen and improve the performance of the algorithm by varying other hyperparameters to those considered in this study.

Keywords: Electrical substations, Hot spots, Object detection, Thermal images, Transfer learning, YOLOv5

^{1,*}Facultad de Ingeniería, Universidad Tecnológica del Perú, Perú. Autor para correspondencia ✉: daniel.perez@upn.pe.

²Facultad de Ingeniería, Universidad de Piura, Perú.

³Facultad de Ingeniería, Universidad Privada del Norte, Perú.

Forma sugerida de citación: Pérez-Aguilar, D. A.; Pérez-Aguilar, J. M.; Pérez-Aguilar, A. P.; Risco-Ramos, R. H. y Malpica-Rodríguez, M. E. "Inspección de subestaciones eléctricas: YOLOv5 en la identificación de puntos calientes mediante imágenes térmicas," *Ingenius, Revista de Ciencia y Tecnología*, N.º 31, pp. 43-54, 2024. DOI: <https://doi.org/10.17163/ings.n31.2024.04>.

1. Introducción

Las subestaciones eléctricas son instalaciones compuestas por equipos necesarios para ejecutar transformaciones de tensión, frecuencia, cantidad de fases o unión de circuitos; este tipo de equipamiento suele situarse cerca de centrales de generación de energía o en zonas no urbanas [1, 2]. Para garantizar una adecuada calidad del suministro eléctrico, se requiere que este tipo de equipos se mantengan en funcionamiento la mayor parte del tiempo de su vida útil [3, 4]. Es común que las fallas en las subestaciones eléctricas vengan precedidas por la aparición de lo que los especialistas denominan puntos calientes; los cuales son elementos que presentan una temperatura por encima de su operación normal, comparado con otro equipo de similares características y condiciones de funcionamiento [5].

Normalmente, los puntos calientes se producen por aumento de la resistencia eléctrica en un circuito; el mismo que puede provenir a causa de suciedad, falsos contactos, sobrecorrientes, desbalance de carga, pérdida de aislamiento, etc. Un punto caliente no es apreciable a simple vista, por lo que se requiere del uso de técnicas especiales, como la termografía infrarroja; sin embargo, el análisis debe realizarse de manera manual por un experto o especialista en el área, lo cual implica una considerable inversión de tiempo, además de posibles errores humanos en el diagnóstico [6].

Esta es una técnica no invasiva para la medición de la temperatura en diversos objetos, la cual no requiere de contacto físico directo con los mismos. Para lograr esta medición, se emplea la captura de la radiación infrarroja del espectro electromagnético utilizando cámaras termográficas, las cuales generan imágenes en el espectro infrarrojo [7].

La inteligencia artificial (IA) es una disciplina que ha tenido gran repercusión tecnológica en los últimos años; siendo aplicable en diversos sectores, entre los cuales se encuentra el eléctrico [8]. La combinación de termografía y técnicas de IA abarcan una amplia gama de aplicaciones. Estas incluyen el mantenimiento predictivo en entornos industriales, evaluación de estructuras edificadas, identificación de pérdidas energéticas, detección de puentes térmicos, localización de emisiones gaseosas, identificación de áreas con temperaturas elevadas, investigaciones médicas, pronósticos meteorológicos, aplicaciones militares, entre otras [9].

La detección de objetos forma parte de la IA y visión por computadora, buscando detectar uno o varios objetos, ya sea en formato de imagen o video. En la detección de objetos se encuentran algoritmos clasificados en dos grandes grupos (i) Extracción manual (handcrafted features) y (ii) Extracción automática (basada en deep learning o aprendizaje profundo); esta última se divide en algoritmos de detección de objetos de una y dos etapas, siendo la primera la que presenta detecciones en un menor tiempo (ver Figura 1), [10].

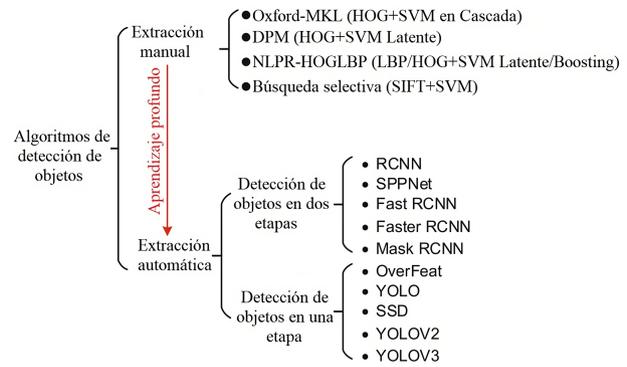


Figura 1. Algoritmos de detección de objetos [11]

YOLO (abreviatura de “*You Only Look Once*”) es uno de los principales algoritmos de detección de objetos. En comparación a otros algoritmos como SSD o Faster R-CNN, se emplea para la detección y reconocimiento en tiempo real de múltiples objetos. YOLO aborda la detección como un problema de regresión y proporciona probabilidades asociadas a cada clase detectada en una única ejecución del algoritmo. Las ventajas clave de YOLO incluyen (i) rapidez, posibilitando la detección en tiempo real; (ii) precisión destacada debido a su baja tasa de errores; y (iii) notable capacidad de aprendizaje [12].

YOLOv5 representa la quinta generación del algoritmo YOLO, siendo un detector de una sola etapa, y una de las opciones más viables si se desea realizar detección de objetos en tiempo real (FPS) [13]. La Figura 2 muestra la arquitectura de YOLOv5, en la que se aprecian las capas personalizadas del algoritmo, compuestas principalmente por convoluciones y maxpooling; además, en la Figura 3, se aprecia su pseudocódigo.

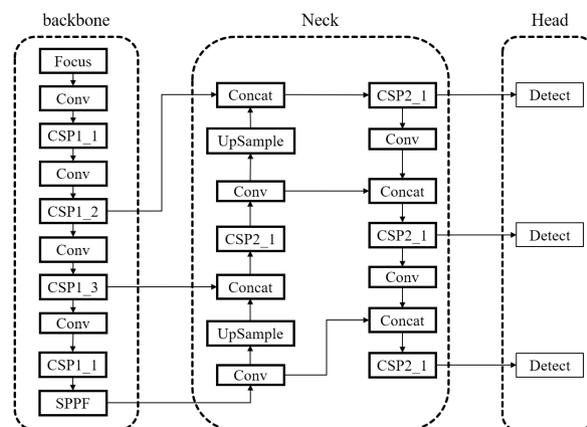


Figura 2. Arquitectura de YOLOv5 [14]

Matemáticamente (véase la ecuación (1)), el modelo predice las coordenadas b_x, b_y, b_w, b_h de las cajas delimitadoras y la confianza C , de que hay un objeto en cada celda de una cuadrícula $S \times S$, donde cada celda se encarga de detectar los objetos presentes en su interior.

Estas predicciones se obtienen aplicando una función sigmoide (σ) a la salida de una red neuronal, donde W representa los pesos de la red neuronal, $f(x)$ la entrada, y b el sesgo. El modelo realiza estas predicciones para B cajas, lo que le permite detectar múltiples objetos en una sola pasada, haciéndolo eficiente para la detección en tiempo real. YOLOv5 emplea principalmente tres técnicas: (i) utilización de bloques residuales, (ii) regresión de bounding box, e (iii) intersección sobre unión (IOU) [15]. La combinación de estas tres técnicas genera resultados como los ilustrados en la Figura 4.

$$B \cdot (b_x, b_y, b_w, b_n, C) = \sigma(W \cdot f(x) + b) \quad (1)$$

Pseudocode - YOLOv5	
Input:	Number class; Class name;
1.	Load images and pre-process data
2.	Define the model architecture
3.	- Backbone network (e.g., CSPNet, GAMMAttention, SPPFCSPC)
4.	- Neck network (e.g., YOLOv5Neck)
5.	- Detection head (e.g., YOLOv5Head)
6.	- Loss function (e.g., Focal Loss)
7.	Train the model
8.	- Compute loss on mini-batch of images
9.	- Compute gradients and update weights using optimizer (e.g., Adam)
10.	Prediction
11.	- Apply non-maximum suppression to remove overlapping predictions
12.	- Output final detection results (bounding boxes, class probabilities, confidence scores)

Figura 3. Pseudocódigo-YOLOv5 [14]

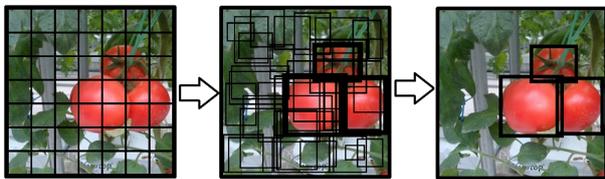


Figura 4. YOLOv5-Combinación de técnicas [15]

A continuación, se presentan algunos estudios previos del tema en cuestión.

Con el propósito de garantizar la seguridad en sistemas de energía eléctrica, se investigó una red neuronal convolucional (CNN) basada en el algoritmo de detección de objetos YOLO. Se realizaron predicciones referentes a las coordenadas, el ángulo de orientación y la clasificación de cada componente del equipo. Los resultados experimentales indicaron que el enfoque es resistente al ruido, logrando un nivel de precisión del 93,7 % mediante el empleo de una unidad de procesamiento gráfico (GPU) durante la fase de entrenamiento [16].

Un equipo de potencia es una parte importante del sistema de energía, además del foco de operación y mantenimiento. Li [17] menciona que la tecnología de detección de anomalías por infrarrojos es un medio eficaz para detectar fallas en equipos eléctricos debido a su seguridad, simplicidad e intuición. Se planteó la aplicación de YOLOv3 mediante un conjunto de imágenes infrarrojas recopiladas en campo, obteniéndose un valor mAP (*Mean Average Precision*) de 34,63 % y una tasa de recuperación de 21 %.

Greco et al. [18] señalan que las fallas suelen aparecer como puntos calientes en la superficie de paneles fotovoltaicos, por lo que se estudió el problema de detección de puntos calientes utilizando YOLO, demostrándose que el algoritmo en cuestión presenta una capacidad eficaz y eficiente para llevar a cabo la segmentación de paneles en una imagen. Se efectuó una evaluación cuantitativa, y se realizó una comparación con enfoques previamente establecidos para la detección de paneles fotovoltaicos. Los resultados experimentales obtenidos corroboraron la robustez y eficacia de YOLO.

El uso de CNN requiere una alta capacidad computacional y de memoria. Nguyen et al. [19] propusieron un método de detección de objetos basado en una CNN y YOLO; para ello, se consideró el formato de etiquetado PASCAL VOC. El modelo alcanzó un valor mAP de 64,16 %.

Un estudio de tres algoritmos de detección de objetos en imágenes [12] indicó que: (i) Single Shot Detector o SSD [20] obtiene un rendimiento bajo en comparación a Faster R-CNN al detectar objetos pequeños, presentando la desventaja que requiere de un set de datos grande, además de técnicas de data augmentation para el entrenamiento, lo cual es computacionalmente costoso y requiere de mayor tiempo de ejecución. (ii) Faster R-CNN [21] es más preciso, pero la complejidad del algoritmo requiere de un tiempo de entrenamiento alto, siendo significativamente más lento que YOLO y requiere analizar muchas veces cada imagen, a diferencia de YOLO. Finalmente, se indica que (iii) YOLO [22] presenta un modelo optimizado y más eficiente que SSD y Faster R-CNN. Versiones más recientes del algoritmo ofrecen una latencia baja, mejoras en el entrenamiento y tiempo de ejecución. Señalaron además que YOLO permite trabajar en tiempo real y obtener mejores precisiones, más aún si se usan técnicas de transfer learning [23].

En relación con técnicas tradicionales de detección de objetos, algunas de las más utilizadas son el agrupamiento de K -means, máquinas de vectores soporte, sistemas difusos, histogramas de gradientes orientados (HOG), entre otras [24]; las cuales obtienen precisiones superiores al 80 %. Sin embargo, el objetivo de este artículo no es profundizar en técnicas tradicionales, sino brindar alternativas basadas en algoritmos modernos y específicos para la detección de objetos tales

como YOLOv5. A pesar de ello, la Tabla 1 (elaborada sobre la base de [24–26], presenta una comparativa entre técnicas tradicionales y YOLOv5, destacando

la superioridad de YOLOv5 en aspectos clave para la detección de objetos.

Tabla 1. Comparativa de YOLOv5 vs. técnicas tradicionales

Característica	Técnicas tradicionales	YOLOv5
Precisión	Variable y dependiente de la calidad de características manuales o parámetros específicos.	Alta precisión, especialmente en la detección de múltiples objetos y de diferentes tamaños en tiempo real.
Manejo de superposición de objetos	Pueden presentar dificultades al lidiar con la superposición de objetos.	Buen manejo de superposición de objetos debido a su capacidad para predecir múltiples cajas delimitadoras.
Adaptabilidad a diferentes tamaños de objetos	Requiere ajustes manuales para adaptarse a diferentes tamaños de objetos.	Escalable y puede adaptarse a objetos de diferentes tamaños.
Capacidad de generalización	Menor capacidad de generalización a nuevas situaciones sin ajustes manuales.	Capacidad de generalizar patrones y características de manera automática, y en escenarios distintos.
Entrenamiento End-to-End	Proceso de entrenamiento más complejo que implica múltiples etapas y ajustes.	Entrenamiento end-to-end, lo que facilita la implementación y reduce la necesidad de etapas intermedias manuales.
Rendimiento en tiempo real	Rendimiento variable, puede ser lento dependiendo de la técnica y la cantidad de características manuales.	Diseñado específicamente para lograr detección en tiempo real, lo que lo hace eficiente para aplicaciones prácticas.
Manejo de grandes conjuntos de datos	Algunas técnicas tradicionales pueden tener limitaciones al manejar grandes conjuntos de datos.	Eficiente en el manejo de grandes conjuntos de datos, beneficiándose de la capacidad de las GPU para acelerar operaciones.

Finalmente, mencionar que, YOLO presenta diversas aplicaciones, no obstante, el análisis de la literatura demuestra que este tipo de técnicas no ha sido empleado en la identificación de puntos calientes en imágenes termográficas de subestaciones eléctricas, lo cual representa un aspecto innovador. Es por ello por lo que el presente estudio es continuación de una investigación publicada previamente por los autores en esta prestigiosa revista [27]. Se busca analizar el funcionamiento de cuatro versiones del algoritmo YOLOv5, divididas en 64 experimentos y aplicados a la detección de puntos calientes, para ello, se incluye el uso de un conjunto imágenes térmicas de subestaciones eléctricas capturadas por una empresa de distribución de energía en el norte peruano.

El documento inicia con una descripción del algoritmo y metodología utilizada; a continuación, se presentan los resultados, y finalmente, se plasman las conclusiones correspondientes.

2. Materiales y métodos

La investigación partió desde la captura de imágenes térmicas, el etiquetado de las mismas, la realización de un preprocesamiento y la correspondiente división del set de datos. Posteriormente, se descargaron los pesos previamente entrenados de las versiones YOLOv5s (*small*), YOLOv5m (*medium*), YOLOv5l (*large*) y YOLOv5x (*extra-large*). Finalmente, se realizaron 64 experimentos y se analizaron mediante diferentes métricas. El proceso de implementación completo se detalla en los siguientes puntos y se resume en la Figura 5.

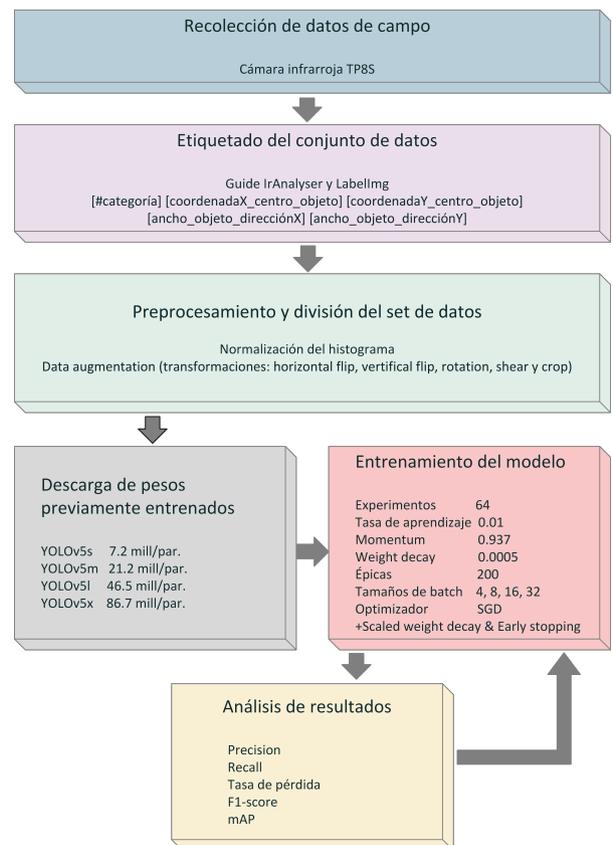


Figura 5. Pasos para la implementación de YOLOv5

2.1. Recolección de datos de campo

La metodología inició por la recolección de datos de campo mediante la captura de imágenes térmicas. Se utilizó la cámara infrarroja TP8S, que incluye un rango

espectral de 8-14 μm , detector FPA (384x288 píxeles, 35 μm), campo de visión de $22^\circ \times 16^\circ / 35 \text{ mm}$, enfoque electrónico automático, Sensibilidad térmica de 0.08 $^\circ\text{C}$ a 130 $^\circ\text{C}$, y un zoom continuo de $\times 1$ a $\times 10$.

La fuente de datos empleada en este estudio consta de un conjunto de 815 imágenes termográficas, las cuales fueron adquiridas por expertos de una entidad dedicada a la distribución de energía eléctrica en la región septentrional de Perú. La Figura 6 exhibe un ejemplo del proceso de generación de una imagen termográfica de una subestación eléctrica a partir de la radiación en el espectro electromagnético.

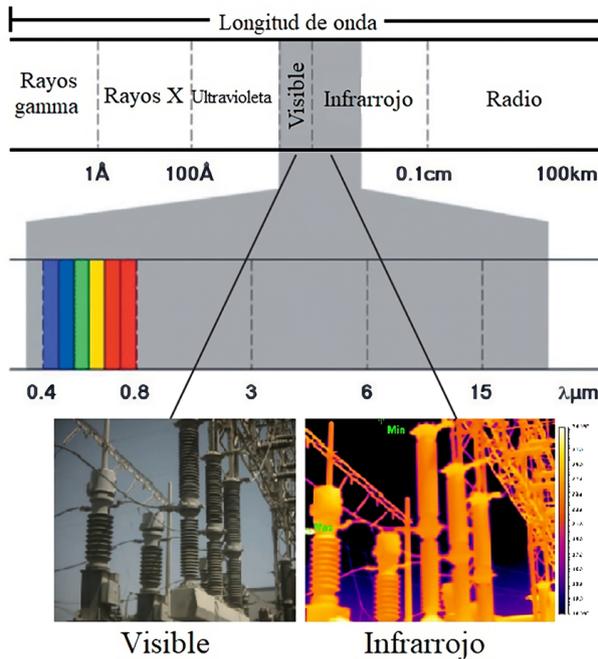


Figura 6. Espectro electromagnético de una subestación eléctrica

Estas imágenes presentan una resolución de 384×288 píxeles y fueron categorizadas en dos clases principales: líneas eléctricas, y subestaciones eléctricas (ver Tabla 2).

Tabla 2. Distribución del conjunto de datos original

Clases	Total
Línea	342
Subestación	473

2.2. Etiquetado del conjunto de datos

El etiquetado de este conjunto de datos fue realizado por un especialista en detección de puntos calientes en imágenes termográficas utilizando las herramientas *Guide IrAnalyser* y *LabelImg*. Cada punto caliente identificado se asoció con un archivo de texto correspondiente a la imagen subyacente, en el que se registraron

la clase del objeto y sus coordenadas, siguiendo un formato específico: [número de categoría] [coordenada X del centro del objeto] [coordenada Y del centro del objeto] [ancho del objeto en dirección X] [ancho del objeto en dirección Y]. En la Figura 7 se ilustra el formato empleado para representar dos puntos calientes; este formato se repite en cada fila de acuerdo con el número de puntos calientes identificados y etiquetados.

```
Sub_station (13).txt
1 0.459635 0.362847 0.111979 0.315972
1 0.826823 0.336806 0.117188 0.340278
```

Figura 7. Archivo de texto – 2 puntos calientes

2.3. Preprocesamiento y división del set de datos

Como parte del preprocesamiento de las imágenes, se realizó la normalización del histograma, con la finalidad de ajustar la distribución de intensidades de píxeles de cada imagen térmica para mejorar sus características.

Inicialmente, se contó con un total de 815 imágenes; de las cuales, 342 representaban imágenes de líneas eléctricas, y 473 subestaciones eléctricas, tal como se detalla en la Tabla 2. No obstante, luego de realizar el etiquetado de imágenes con puntos calientes, la cantidad de imágenes se redujo a 138 en total. De estas, 116 imágenes correspondían a subestaciones, y 22 a líneas eléctricas; por lo cual, se delimitó el presente estudio únicamente al análisis de subestaciones eléctricas.

A continuación, las imágenes se cargaron en la plataforma *Roboflow* y, a través de la técnica de división de tipo *"hold-out"*, se generaron tres conjuntos de datos: (i) un conjunto de 81 imágenes destinado al entrenamiento, (ii) un conjunto de 23 imágenes para validación, y (iii) un conjunto de 12 imágenes para pruebas.

Además, con la finalidad de incrementar el número de imágenes y hacer el entrenamiento más eficaz, se aplicó *data augmentation*, mediante las transformaciones: *horizontal flip*, *vertical flip*, *rotation*, *shear* y *crop*, obteniéndose un set de datos con 278 imágenes: (i) 243 imágenes para entrenamiento, (ii) 23 para validación y (iii) 12 para pruebas.

Andrew Yan-Tak Ng, director del laboratorio de Inteligencia Artificial de la Universidad de Stanford, en un artículo publicado en *Spectrum*, revista editada por IEEE, señala lo siguiente: “En diversas industrias, contar con millones de datos para entrenar modelos de inteligencia artificial es complicado; por lo que, disponer de una cantidad pequeña de imágenes realmente buenas o de calidad, puede ser útil para construir sistemas de inspección de defectos. Sumado a esto, la

precisión incrementa si se trabaja con los pesos de modelos previamente entrenados” [28].

2.4. Descarga de pesos previamente entrenados

Buscando mejorar el desempeño de la solución, se procedió a descargar los pesos preentrenados de los cuatro algoritmos pertenecientes a YOLOv5. Estos pesos preentrenados se aplicaron a las capas convolucionales del detector, lo cual contribuyó significativamente a incrementar los porcentajes de precisión y reducir los tiempos de duración del proceso de entrenamiento. La Tabla 3 proporciona información detallada sobre los pesos de preentrenamiento asociados a cada uno de los algoritmos, así como la cantidad de parámetros contenidos en cada uno de ellos. Se destaca que YOLOv5s se presenta como el algoritmo de menor complejidad, mientras que YOLOv5x es el más sofisticado en términos de parámetros y complejidad.

Tabla 3. Peso y parámetros de cada algoritmo

Algoritmo	Pesos entrenados (MB)	Parámetros (millones)
YOLOv5s	14.1	7.2
YOLOv5m	40.8	21.2
YOLOv5l	89.3	46.5
YOLOv5x	166	86.7

2.5. Entrenamiento del modelo

Se realizaron 64 experimentos según el detalle de la Tabla 4, los cuales fueron ejecutados en la plataforma Google Colaboratory (Colab), cuya versión gratuita ofrece un procesador Intel Xeon con 2.30 GHz, acelerador de GPU NVIDIA Tesla K80, memoria RAM de 13 GB, y espacio en disco de 40 GB.

Tabla 4. Peso y parámetros de cada algoritmo

Exp.	Algoritmo	Batch size	Data augmentation	Transfer learning
E1	YOLOv5s	4	No	No
E2	YOLOv5s	8	No	No
E3	YOLOv5s	16	No	No
E4	YOLOv5s	32	No	No
E5	YOLOv5s	4	No	Sí
E6	YOLOv5s	8	No	Sí
E7	YOLOv5s	16	No	Sí
E8	YOLOv5s	32	No	Sí
E9	YOLOv5s	4	Sí	No
E10	YOLOv5s	8	Sí	No
E11	YOLOv5s	16	Sí	No
E12	YOLOv5s	32	Sí	No
E13	YOLOv5s	4	Sí	Sí
E14	YOLOv5s	8	Sí	Sí

Exp.	Algoritmo	Batch size	Data augmentation	Transfer learning
E15	YOLOv5s	16	Sí	Sí
E16	YOLOv5s	32	Sí	Sí
E17	YOLOv5m	4	No	No
E18	YOLOv5m	8	No	No
E19	YOLOv5m	16	No	No
E20	YOLOv5m	32	No	No
E21	YOLOv5m	4	No	Sí
E22	YOLOv5m	8	No	Sí
E23	YOLOv5m	16	No	Sí
E24	YOLOv5m	32	No	Sí
E25	YOLOv5m	4	Sí	No
E26	YOLOv5m	8	Sí	No
E27	YOLOv5m	16	Sí	No
E28	YOLOv5m	32	Sí	No
E29	YOLOv5m	4	Sí	Sí
E30	YOLOv5m	8	Sí	Sí
E31	YOLOv5m	16	Sí	Sí
E32	YOLOv5m	32	Sí	Sí
E33	YOLOv5l	4	No	No
E34	YOLOv5l	8	No	No
E35	YOLOv5l	16	No	No
E36	YOLOv5l	32	No	No
E37	YOLOv5l	4	No	Sí
E38	YOLOv5l	8	No	Sí
E39	YOLOv5l	16	No	Sí
E40	YOLOv5l	32	No	Sí
E41	YOLOv5l	4	Sí	No
E42	YOLOv5l	8	Sí	No
E43	YOLOv5l	16	Sí	No
E44	YOLOv5l	32	Sí	No
E45	YOLOv5l	4	Sí	Sí
E46	YOLOv5l	8	Sí	Sí
E47	YOLOv5l	16	Sí	Sí
E48	YOLOv5l	32	Sí	Sí
E49	YOLOv5xl	4	No	No
E50	YOLOv5xl	8	No	No
E51	YOLOv5xl	16	No	No
E52	YOLOv5xl	32	No	No
E53	YOLOv5xl	4	No	Sí
E54	YOLOv5xl	8	No	Sí
E55	YOLOv5xl	16	No	Sí
E56	YOLOv5xl	32	No	Sí
E57	YOLOv5xl	4	Sí	No
E58	YOLOv5xl	8	Sí	No
E59	YOLOv5xl	16	Sí	No
E60	YOLOv5xl	32	Sí	No
E61	YOLOv5xl	4	Sí	Sí
E62	YOLOv5xl	8	Sí	Sí
E63	YOLOv5xl	16	Sí	Sí
E64	YOLOv5xl	32	Sí	Sí

Los hiperparámetros utilizados fueron una tasa de aprendizaje de 0.01, momentum de 0.937, weight decay de 0.0005, 200 épocas y 4 tamaños de batch, además del optimizador SGD.

Con la finalidad de evitar el overfitting (sobreajuste del modelo), se incluyó la regularización *Scaled weight decay* y la técnica de *Early stopping*, configurada con un patience de 100; es decir, el modelo detiene el entrenamiento si no observa mejoras en las últimas 100 épocas. Además, los pesos de las mejores épocas se almacenan en cada ejecución y se analizan los resultados de las curvas de mAP, Precision, Recall, Loss. El flujo seguido para el entrenamiento de YOLOv5 se plasma en la Figura 8.

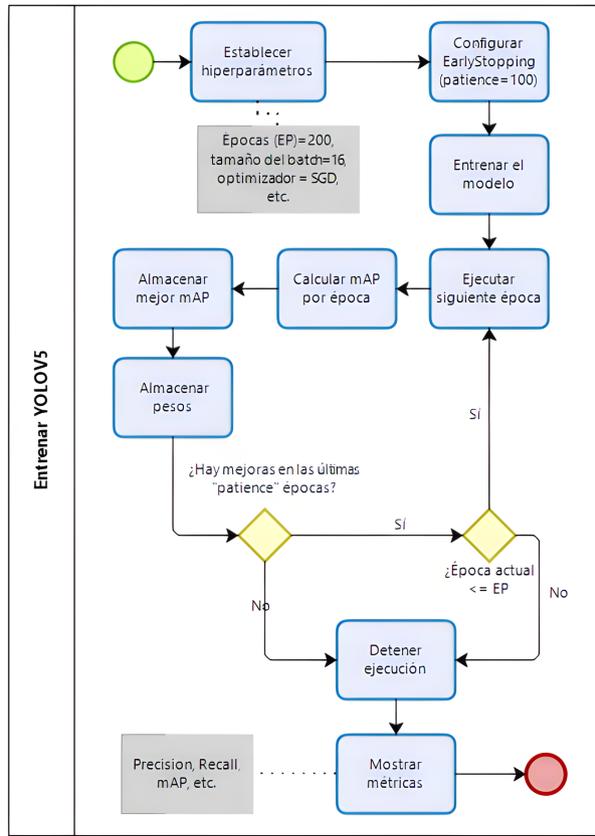


Figura 8. Diagrama de flujo del entrenamiento de YOLOv5

2.6. Análisis de resultados

Luego del entrenamiento de cada modelo, se analizaron métricas como *Precision* (2), *Recall* (3), F1-score (4), tasa de pérdida, y mAP (5), el cual se calcula en función del *Average Precision* (6). Siendo TP el conjunto de verdaderos positivos, FP los falsos positivos, FN los falsos negativos y N el número de clases.

En diversas investigaciones se señala que métricas como F1-score y mAP (*Mean Average Precision*) son adecuadas al momento de comparar modelos [12], [29].

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1_{SCORE} = 2 * \frac{precision * recall}{precision + recall} \quad (4)$$

$$mAP = \frac{1}{n} \sum_{k=1}^{k=n} AP_k \quad (5)$$

$$AP = \sum_{k=0}^{k=n-1} [Recalls(k) - Recalls(k + 1) * Precisions(k)] \quad (6)$$

3. Resultados y discusión

A continuación, se presentan los resultados obtenidos por cada una de las cuatro variantes de YOLOv5.

3.1. YOLOv5s

YOLOv5 *small* obtuvo mejores resultados (mAP = 69.42 % y F1-score = 68.23 %) al entrenar el modelo con el experimento 5, es decir, con un tamaño de batch de 8, 171 épocas, aplicando transfer learning y sin incorporar data augmentation (ver Tabla 5).

Tabla 5. YOLOv5 small – resultados del entrenamiento

Exp.	Precision	Recall	F1-score	mAP
E1	47.16 %	67.57 %	55.55 %	50.26 %
E2	75.97 %	51.35 %	61.28 %	59.77 %
E3	59.97 %	64.86 %	62.32 %	65.08 %
E4	55.95 %	75.68 %	64.34 %	64.70 %
E5	60.40 %	78.38 %	68.23 %	69.42 %
E6	59.17 %	78.33 %	67.42 %	65.44 %
E7	79.98 %	53.99 %	64.47 %	68.99 %
E8	57.57 %	70.27 %	63.29 %	57.24 %
E9	80.00 %	54.05 %	64.51 %	64.65 %
E10	57.43 %	72.97 %	64.27 %	64.96 %
E11	56.39 %	59.46 %	57.89 %	53.07 %
E12	55.26 %	56.76 %	56.00 %	52.85 %
E13	67.72 %	56.70 %	61.72 %	56.30 %
E14	57.12 %	64.80 %	60.72 %	59.94 %
E15	58.31 %	75.68 %	65.87 %	58.82 %
E16	58.47 %	64.86 %	61.50 %	60.52 %

3.2. YOLOv5m

YOLOv5 *medium* obtiene resultados superiores a YOLOv5s (mAP = 81.99 % y F1-score = 78.57 %) al entrenar el modelo con el experimento 22, es decir, con un tamaño de batch de 16, 139 épocas, aplicando transfer learning y sin incorporar data augmentation (ver Tabla 6).

Tabla 6. YOLOv5 medium – resultados del entrenamiento

Exp.	Precision	Recall	F1-score	mAP
E17	59.01 %	70.27 %	64.15 %	50.26 %
E18	86.13 %	51.35 %	64.34 %	59.77 %
E19	70.27 %	70.26 %	70.27 %	65.08 %
E20	65.71 %	62.16 %	63.89 %	64.70 %
E21	70.72 %	78.38 %	74.35 %	69.42 %
E22	70.21 %	89.18 %	78.57 %	65.44 %
E23	74.34 %	78.31 %	76.28 %	68.99 %
E24	74.28 %	70.24 %	72.20 %	57.24 %
E25	84.05 %	45.95 %	59.41 %	64.65 %
E26	88.62 %	43.24 %	58.12 %	64.96 %
E27	58.95 %	62.16 %	60.51 %	53.07 %
E28	49.97 %	75.68 %	60.19 %	52.85 %
E29	91.60 %	59.46 %	72.11 %	56.30 %
E30	64.82 %	64.86 %	64.84 %	59.94 %
E31	69.42 %	67.50 %	68.45 %	58.82 %
E32	81.42 %	59.46 %	68.73 %	60.52 %

3.3. YOLOv5l

YOLOv5 large obtuvo resultados similares a YOLOv5m (mAP = 81.88 % y F1-score = 80.51 %) al entrenar el modelo con el experimento 37, es decir, con un tamaño de batch de 8, 180 épocas, aplicando transfer learning y sin incorporar data augmentation (ver Tabla 7).

Tabla 7. YOLOv5 large – resultados del entrenamiento

Exp.	Precision	Recall	F1-score	mAP
E33	80.70 %	45.95 %	58.55 %	55.38 %
E34	60.97 %	67.57 %	64.10 %	62.77 %
E35	55.97 %	75.68 %	64.35 %	68.42 %
E36	61.86 %	70.13 %	65.74 %	63.07 %
E37	89.79 %	72.97 %	80.51 %	81.88 %
E38	73.80 %	83.78 %	78.48 %	79.98 %
E39	68.11 %	81.08 %	74.03 %	78.10 %
E40	89.18 %	67.57 %	76.88 %	80.56 %
E41	57.14 %	64.86 %	60.76 %	61.48 %
E42	73.04 %	51.27 %	60.25 %	59.81 %
E43	52.99 %	70.27 %	60.42 %	63.03 %
E44	49.12 %	75.65 %	59.56 %	65.17 %
E45	73.53 %	67.56 %	70.42 %	71.16 %
E46	80.47 %	78.38 %	79.41 %	77.68 %
E47	82.20 %	64.86 %	72.51 %	71.97 %
E48	69.22 %	72.95 %	71.04 %	70.16 %

3.4. YOLOv5xl

YOLOv5 extra-large obtuvo resultados ligeramente más bajos a YOLOv5m y YOLOv5l (mAP = 79.25 % y F1-score = 76.92 %) al entrenar el modelo con el experimento 56, es decir, con un tamaño de batch de 32, 178 épocas, aplicando *transfer learning* y sin incorporar *augmentation* (ver Tabla 8). Siendo el único caso en el que los mejores resultados se presentaron con un tamaño de batch relativamente superior a los otros tres algoritmos.

Tabla 8. YOLOv5 extra-large – resultados del entrenamiento

Exp.	Precision	Recall	F1-score	mAP
E49	51.32 %	51.28 %	51.30 %	55.38 %
E50	53.29 %	64.86 %	58.51 %	62.77 %
E51	62.42 %	67.57 %	64.89 %	68.42 %
E52	67.51 %	67.57 %	67.54 %	63.07 %
E53	77.04 %	72.56 %	74.73 %	81.88 %
E54	72.09 %	83.70 %	77.46 %	79.98 %
E55	69.76 %	81.06 %	74.99 %	78.10 %
E56	73.17 %	81.08 %	76.92 %	80.56 %
E57	57.12 %	64.86 %	60.74 %	61.48 %
E58	49.99 %	64.86 %	56.47 %	59.81 %
E59	56.51 %	70.27 %	62.64 %	63.03 %
E60	77.26 %	45.91 %	57.59 %	65.17 %
E61	70.96 %	59.45 %	64.70 %	71.16 %
E62	76.66 %	62.16 %	68.65 %	77.68 %
E63	71.42 %	81.03 %	75.92 %	71.97 %
E64	73.78 %	83.78 %	78.46 %	70.16 %

3.5. YOLOv5 – Mejores resultados

Se elaboró un ranking de los 64 experimentos, considerando a *Mean Average Precision* (mAP) como factor de ordenamiento (ver Tabla 9).

Tabla 9. Detalle de los experimentos realizados

Exp.	Ranking	Loss	mAP	Mejor época
E1	63	0.05484	50.26 %	157
E2	52	0.05137	59.77 %	153
E3	31	0.04212	65.08 %	167
E4	34	0.04340	64.70 %	143
E5	22	0.04060	69.42 %	171
E6	29	0.03988	65.44 %	190
E7	23	0.04106	68.99 %	146
E8	56	0.04953	57.24 %	193
E9	35	0.04657	64.65 %	115
E10	32	0.03710	64.96 %	168
E11	61	0.03740	53.07 %	134
E12	62	0.03750	52.85 %	185
E13	57	0.03718	56.30 %	76
E14	50	0.03360	59.94 %	74
E15	54	0.02952	58.82 %	127
E16	49	0.03107	60.52 %	200
E17	44	0.05050	62.57 %	160
E18	25	0.05133	68.49 %	139
E19	33	0.04590	64.73 %	194
E20	60	0.05628	53.40 %	175
E21	17	0.04042	72.87 %	107
E22	1	0.04541	81.99 %	139
E23	12	0.04426	75.69 %	182
E24	11	0.04362	75.97 %	154
E25	45	0.05280	61.87 %	86
E26	46	0.04952	61.65 %	86
E27	42	0.05051	62.80 %	88
E28	39	0.04908	63.09 %	63
E29	16	0.05071	73.20 %	108
E30	28	0.04914	66.17 %	69
E31	27	0.04743	68.35 %	37
E32	20	0.04665	70.96 %	63
E33	58	0.05487	55.38 %	154
E34	43	0.04841	62.77 %	153
E35	26	0.04331	68.42 %	182
E36	40	0.03910	63.07 %	123
E37	2	0.01974	81.88 %	180
E38	4	0.01602	79.98 %	128
E39	7	0.01615	78.10 %	125
E40	3	0.01396	80.56 %	160
E41	47	0.04325	61.48 %	136
E42	51	0.03757	59.81 %	162
E43	41	0.03571	63.03 %	60
E44	30	0.03288	65.17 %	126
E45	19	0.01628	71.16 %	182
E46	8	0.01431	77.68 %	124
E47	18	0.01702	71.97 %	43
E48	21	0.01196	70.16 %	165
E49	64	0.06242	44.01 %	184
E50	53	0.04919	58.92 %	198
E51	37	0.04517	63.48 %	198
E52	14	0.04052	74.29 %	196
E53	9	0.02470	77.62 %	50
E54	6	0.01478	78.78 %	178
E55	13	0.01468	75.57 %	87
E56	5	0.01247	79.25 %	178
E57	48	0.04357	61.42 %	101
E58	59	0.04474	54.13 %	181
E59	36	0.03493	64.20 %	156
E60	55	0.03677	58.66 %	98
E61	38	0.01463	63.36 %	165
E62	24	0.01390	68.92 %	89
E63	15	0.01015	73.42 %	109
E64	10	0.01134	77.01 %	92

Los resultados más destacados de cada uno de los algoritmos se documentaron en la Tabla 10, y se presentan de manera gráfica en la Figura 9 (mAP), Figura 10 (precision), Figura 11 (recall), Figura 12 (tasa de pérdida durante el entrenamiento) y Figura 13 (tasa

de pérdida durante la validación). En los cuatro escenarios evaluados, se observa que el desempeño más sobresaliente se logró mediante el empleo de transfer learning, prescindiendo del uso de data augmentation.

De los 64 experimentos llevados a cabo, el modelo más sobresaliente surgió al utilizar YOLOV5m y ser entrenado durante un total de 139 épocas, con un tamaño de batch de 8, sin recurrir al empleo de *data augmentation*, y haciendo uso de la técnica de *transfer learning*.

Tabla 10. Mejores resultados por algoritmo

Algoritmo	Experimento	mAP	Mejor época	Ranking
YOLOv5s	E5	69.42 %	171	22
YOLOv5m	E22	81.99 %	139	1
YOLOv5l	E37	81.88 %	180	2
YOLOv5xl	E56	79.25 %	178	5

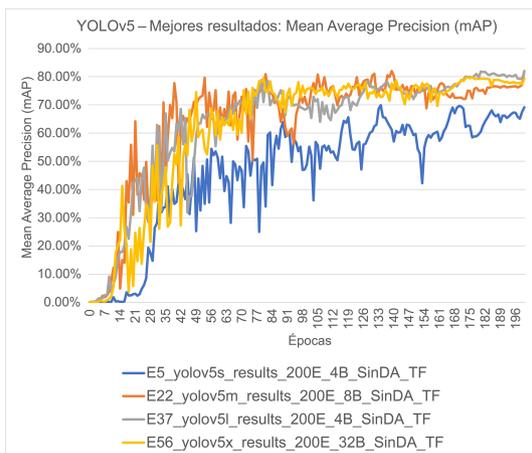


Figura 9. YOLOv5 – Mejores resultados: mAP – Entrenamiento del modelo

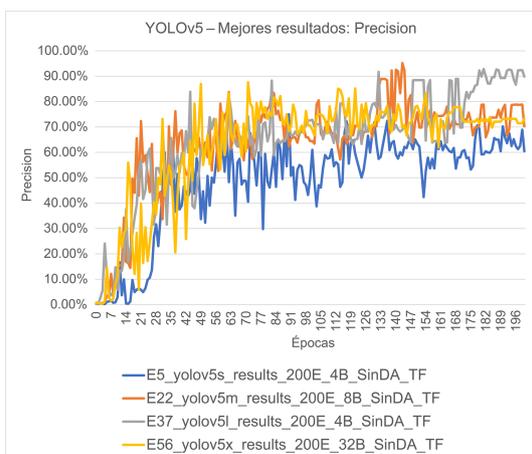


Figura 10. YOLOv5 – Mejores resultados: Precision – Entrenamiento del modelo

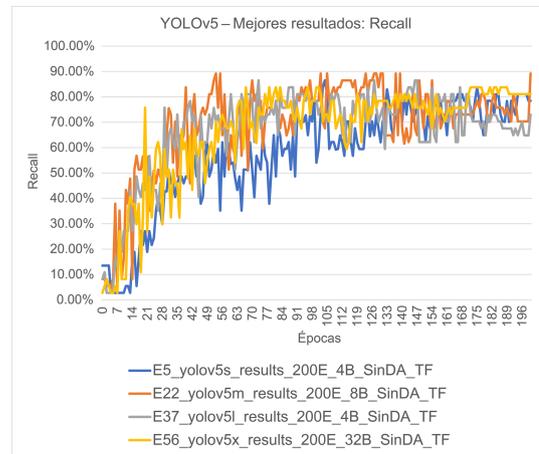


Figura 11. YOLOv5 – Mejores resultados: Recall – Entrenamiento del modelo

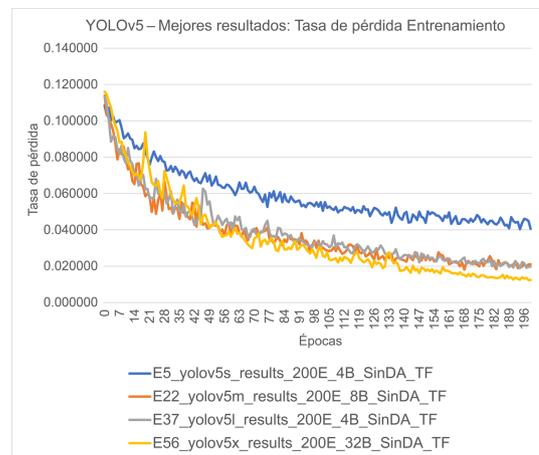


Figura 12. YOLOv5 – Mejores resultados: Tasa de pérdida – Entrenamiento del modelo

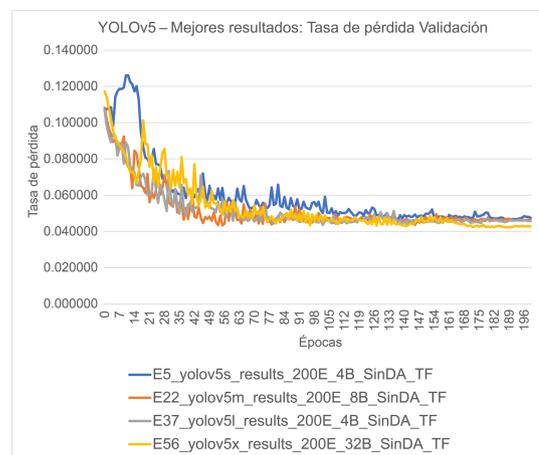


Figura 13. YOLOv5 – Mejores resultados: Tasa de pérdida – Validación del modelo

La descripción visual de la propuesta resultante de esta investigación se ilustra en la Figura 14. El primer paso consiste en introducir la base de datos

de imágenes termográficas, a continuación, se procede con el entrenamiento del modelo utilizando el algoritmo YOLOv5m, y finalmente, se obtiene la detección de puntos calientes en las imágenes de subestaciones eléctricas.

Algunas predicciones del modelo y sus porcentajes correspondientes se muestran en la Figura 15. Estos resultados se obtuvieron con imágenes del conjunto de datos de pruebas, las cuales no fueron incluidas en el entrenamiento ni en la validación del modelo de manera previa.

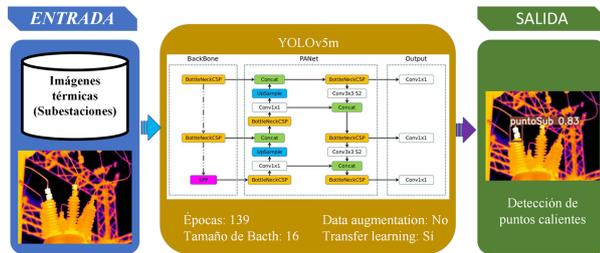


Figura 14. Modelo propuesto

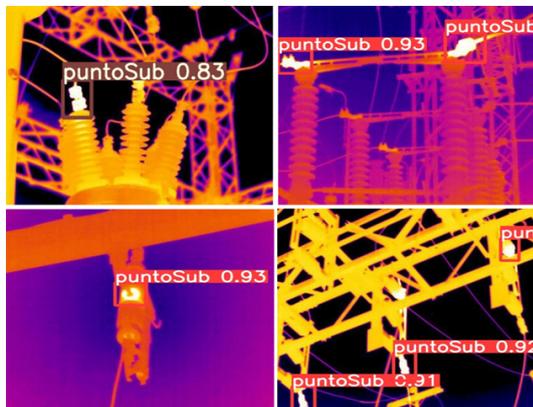


Figura 15. YOLOv5 medium – Detección de puntos calientes

4. Conclusiones

El objetivo principal de esta investigación fue explorar nuevas perspectivas en contraposición a las técnicas convencionales de inteligencia artificial, poniendo un énfasis particular en el algoritmo de detección de objetos YOLOv5, debido a su notoria eficiencia en los procesos de entrenamiento.

Se analizaron 4 versiones del algoritmo YOLOv5, divididos en 64 experimentos y entrenados con un conjunto de imágenes térmicas de subestaciones eléctricas. Los resultados muestran una clara tendencia a mejorar las precisiones si se entrenan con un mayor número de épocas, o se consideran valores distintos a los utilizados en esta investigación, tales como optimizadores, hiperparámetros, entre otros.

La inclusión de técnicas de *data augmentation* ejerce un impacto adverso en la precisión de los modelos en todos los escenarios evaluados. Por otro lado, el uso de la estrategia de *transfer learning*, incluyendo la adquisición de pesos preentrenados para las capas convolucionales, se evidenció como un factor que contribuye a mejorar el rendimiento. En cuanto al tamaño del batch, se observó que los resultados óptimos se obtuvieron en el rango de valores entre 4 y 32. Esta elección se fundamenta en la limitación de tamaño inherente al conjunto de datos actual. Sin embargo, es importante señalar que este valor podría variar en investigaciones futuras que involucren conjuntos de datos más extensos de imágenes termográficas.

Las técnicas *Scaled weight decay* y *Early stopping* contribuyeron a evitar el *Overfitting*. Para ello, se consideró la detención temprana del entrenamiento, cuando no mostraba mejoría en la ejecución de un número determinado de épocas.

Los resultados de la presente investigación sientan un precedente valioso para la futura profundización en la aplicación de algoritmos similares en el ámbito de la detección de puntos calientes en el sector eléctrico. Por lo que, como dirección para investigaciones futuras; se recomienda considerar la inclusión de modelos con otras variantes del algoritmo YOLOv5, como YOLOv5n6, YOLOv5s6, YOLOv5m6, YOLOv5l6, YOLOv5x6, o bien, evaluar alternativas como R-CNN y Faster R-CNN, entre otras. Basándonos en los resultados obtenidos en este estudio, se anticipa que estas exploraciones podrían conducir a rendimientos superiores.

Por último, se sugiere analizar la posibilidad de integrar el modelo desarrollado en una cámara termográfica, lo que permitiría la generación de alertas en tiempo real al momento de capturar imágenes en campo.

Referencias

- [1] S. Hussain, J. Hernández Fernández, A. K. Al-Ali, and A. Shikfa, "Vulnerabilities and countermeasures in electrical substations," *International Journal of Critical Infrastructure Protection*, vol. 33, p. 100406, 2021. [Online]. Available: <https://doi.org/10.1016/J.IJCIP.2020.100406>
- [2] Y.-E. Bouffard-Vercelli and B. André, "Future architectures of electrical substations," in *2021 Petroleum and Chemical Industry Conference Europe (PCIC Europe)*, 2021. [Online]. Available: <https://doi.org/10.23919/PCICEurope50407.2021.9805424>
- [3] W. Pavon, E. Inga, S. Simani, and M. Nonato, "A review on optimal control for the smart grid electrical substation enhancing transition

- stability,” *Energies*, vol. 14, no. 24, 2021. [Online]. Available: <https://doi.org/10.3390/en14248451>
- [4] M. Lin, L. Fu, F. Zeng, G. Yang, and M. Sun, “Design of distributed substation high voltage electrical equipment online monitoring system based on image segmentation technology,” *Journal of Physics: Conference Series*, vol. 2143, no. 1, p. 012001, dec 2021. [Online]. Available: <https://dx.doi.org/10.1088/1742-6596/2143/1/012001>
- [5] M. A. Haq, D. Kurniawan Danu, Syafii, and Muhardika, “Mitigation of the potential for sudden high-temperature hotspots on substation equipment,” in *2023 4th International Conference on High Voltage Engineering and Power Systems (ICHVEPS)*, 2023, pp. 194–198. [Online]. Available: <https://doi.org/10.1109/ICHVEPS58902.2023.10257349>
- [6] S. Y. Lee and S. S. Teoh, “A survey on infrared thermography based automatic electrical fault diagnosis techniques,” in *10th International Conference on Robotics, Vision, Signal Processing and Power Applications*, M. A. M. Zawawi, S. S. Teoh, N. B. Abdullah, and M. I. S. Mohd Sazali, Eds. Singapore: Springer Singapore, 2019, pp. 537–542. [Online]. Available: https://doi.org/10.1007/978-981-13-6447-1_68
- [7] F. Ciampa, P. Mahmoodi, F. Pinto, and M. Meo, “Recent advances in active infrared thermography for non-destructive testing of aerospace components,” *Sensors*, vol. 18, no. 2, 2018. [Online]. Available: <https://doi.org/10.3390/s18020609>
- [8] M. Haenlein and A. Kaplan, “A brief history of artificial intelligence: On the past, present, and future of artificial intelligence,” *California Management Review*, vol. 61, no. 4, pp. 5–14, 2019. [Online]. Available: <https://doi.org/10.1177/0008125619864925>
- [9] A. Ghahramani, G. Castro, S. A. Karvigh, and B. Becerik-Gerber, “Towards unsupervised learning of thermal comfort using infrared thermography,” *Applied Energy*, vol. 211, pp. 41–49, 2018. [Online]. Available: <https://doi.org/10.1016/j.apenergy.2017.11.021>
- [10] Y. J. Wai, Z. bin Mohd Yussof, S. I. bin Salim, and L. K. Chuan, “Fixed point implementation of Tiny-YOLO-v2 using OpenCL on FPGA,” *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 10, 2018. [Online]. Available: <http://dx.doi.org/10.14569/IJACSA.2018.091062>
- [11] Y. Xiao, Z. Tian, J. Yu, Y. Zhang, S. Liu, S. Du, and X. Lan, “A review of object detection based on deep learning,” *Multimedia Tools and Applications*, vol. 79, no. 33, pp. 23 729–23 791, Sep 2020. [Online]. Available: <https://doi.org/10.1007/s11042-020-08976-6>
- [12] S. Srivastava, A. V. Divekar, C. Anilkumar, I. Naik, V. Kulkarni, and V. Pattabiraman, “Comparative analysis of deep learning image detection algorithms,” *Journal of Big Data*, vol. 8, no. 1, p. 66, May 2021. [Online]. Available: <https://doi.org/10.1186/s40537-021-00434-w>
- [13] D. Dlužnevskij, P. Stefanovic, and S. Ramanauškaite, “Investigation of YOLOv5 efficiency in iphone supported systems,” *Baltic Journal of Modern Computing*, vol. 9, no. 3, pp. 333–344, 2021. [Online]. Available: <https://doi.org/10.22364/bjmc.2021.9.3.07>
- [14] Z. Ma, Y. Wan, J. Liu, R. An, and L. Wu, “A kind of water surface multi-scale object detection method based on improved YOLOv5 network,” *Mathematics*, vol. 11, no. 13, 2023. [Online]. Available: <https://doi.org/10.3390/math11132936>
- [15] G. Liu, J. C. Nouaze, P. L. Touko Mbouembe, and J. H. Kim, “YOLO-Tomato: A robust algorithm for tomato detection based on YOLOv3,” *Sensors*, vol. 20, no. 7, 2020. [Online]. Available: <https://doi.org/10.3390/s20072145>
- [16] X. Gong, Q. Yao, M. Wang, and Y. Lin, “A deep learning approach for oriented electrical equipment detection in thermal images,” *IEEE Access*, vol. 6, pp. 41 590–41 597, 2018. [Online]. Available: <https://doi.org/10.1109/ACCESS.2018.2859048>
- [17] X. Li, “Design of infrared anomaly detection for power equipment based on YOLOv3,” in *2019 IEEE 3rd Conference on Energy Internet and Energy System Integration (EI2)*, 2019, pp. 2291–2294. [Online]. Available: <https://doi.org/10.1109/EI247390.2019.9061852>
- [18] A. Greco, C. Pironti, A. Saggese, M. Vento, and V. Vigilante, “A deep learning based approach for detecting panels in photovoltaic plants,” in *Proceedings of the 3rd International Conference on Applications of Intelligent Systems*, ser. APPIS 2020. New York, NY, USA: Association for Computing Machinery, 2020. [Online]. Available: <https://doi.org/10.1145/3378184.3378185>
- [19] D. T. Nguyen, T. N. Nguyen, H. Kim, and H.-J. Lee, “A high-throughput and power-efficient FPGA implementation of YOLO CNN for object detection,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 8, pp. 1861–1873, 2019. [Online]. Available: <https://doi.org/10.1109/TVLSI.2019.2905242>

- [20] D. Fan, D. Liu, W. Chi, X. Liu, and Y. Li, "Improved ssd-based multi-scale pedestrian detection algorithm," in *Advances in 3D Image and Graphics Representation, Analysis, Computing and Information Technology*, R. Kountchev, S. Patnaik, J. Shi, and M. N. Favorskaya, Eds. Singapore: Springer Singapore, 2020, pp. 109–118. [Online]. Available: https://doi.org/10.1007/978-981-15-3867-4_14
- [21] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017. [Online]. Available: <https://doi.org/10.1109/TPAMI.2016.2577031>
- [22] W. Chen, H. Huang, S. Peng, C. Zhou, and C. Zhang, "YOLO-face: a real-time face detector," *The Visual Computer*, vol. 37, no. 4, pp. 805–813, Apr 2021. [Online]. Available: <https://doi.org/10.1007/s00371-020-01831-7>
- [23] S. A. Sánchez, H. J. Romero, and A. D. Morales, "A review: Comparison of performance metrics of pretrained models for object detection using the tensorflow framework," *IOP Conference Series: Materials Science and Engineering*, vol. 844, no. 1, p. 012024, may 2020. [Online]. Available: <https://dx.doi.org/10.1088/1757-899X/844/1/012024>
- [24] J. Xue, F. Cheng, Y. Li, Y. Song, and T. Mao, "Detection of farmland obstacles based on an improved YOLOv5s algorithm by using CIoU and anchor box scale clustering," *Sensors*, vol. 22, no. 5, 2022. [Online]. Available: <https://doi.org/10.3390/s22051790>
- [25] A. Li, S. Sun, Z. Zhang, M. Feng, C. Wu, and W. Li, "A multi-scale traffic object detection algorithm for road scenes based on improved YOLOv5," *Electronics*, vol. 12, no. 4, 2023. [Online]. Available: <https://doi.org/10.3390/electronics12040878>
- [26] J. Shi, J. Yang, and Y. Zhang, "Research on steel surface defect detection based on YOLOv5 with attention mechanism," *Electronics*, vol. 11, no. 22, 2022. [Online]. Available: <https://doi.org/10.3390/electronics11223735>
- [27] D. A. Pérez-Aguilar, R. H. Risco-Ramos, and L. Casaverde-Pacherrez, "Transfer learning en la clasificación binaria de imágenes térmicas," *INGENIUS*, no. 26, pp. 71–86, 2021. [Online]. Available: <https://doi.org/10.17163/ings.n26.2021.07>
- [28] A. Yan-Tak Ng. (2022) Unbiggen AI. IEEE Spectrum. IEEE Spectrum. [Online]. Available: <https://bit.ly/3RNNvsvr>
- [29] R. Padilla, W. L. Passos, T. L. B. Días, S. L. Netto, and E. A. B. da Silva, "A comparative analysis of object detection metrics with a companion open-source toolkit," *Electronics*, vol. 10, no. 3, 2021. [Online]. Available: <https://doi.org/10.3390/electronics10030279>