



UNIVERSIDAD  
DE PIURA

FACULTAD DE INGENIERÍA

**Diseño de un clasificador para el mantenimiento predictivo  
de una red de distribución eléctrica utilizando Deep  
Learning**

Tesis para optar el Grado de  
Doctor en Ingeniería con mención en Automatización, Control y Optimización de  
Procesos

**Daniel Alexis Perez Aguilar**

Asesor(es):  
Dr. Ing. César Alberto Chinguel Arrese

Piura, diciembre de 2022



### Declaración Jurada de Originalidad del Trabajo Final

Yo, **Daniel Alexis Pérez Aguilar**, egresado del Programa Académico/ programa de posgrado de **Doctorado en ingeniería, mención Automatización, Control y Optimización de Procesos** de la Facultad de **Ingeniería** de la Universidad de Piura, identificado(a) con DNI **71132678**

Declaro bajo juramento que:

1. Soy autor del trabajo final titulado:  
**“Diseño de un clasificador para el mantenimiento predictivo de una red de distribución eléctrica utilizando Deep Learning”**  
El mismo que presento bajo la modalidad de **Tesis**.<sup>1</sup> para optar el (Título profesional/Grado Académico<sup>2</sup>) de **Grado de Doctor**.
2. Que el trabajo **no** se realizó en coautoría de otros alumnos de la Universidad de Piura.
3. La asesoría del trabajo estuvo a cargo de:
  - **Dr. Ing. César Alberto Chinguel Arrese** \_\_\_\_\_, identificado con DNI N° **45037058**
4. El texto de mi trabajo final respeta y no vulnera los derechos de terceros o de ser el caso derechos de los coautores, incluidos los derechos de propiedad intelectual, datos personales, entre otros. En tal sentido, el texto de mi trabajo final no ha sido plagiado total ni parcialmente, para la cual he respetado las normas internacionales de citas y referencias de las fuentes consultadas.
5. El texto del trabajo final que presento no ha sido publicado ni presentado antes en cualquier medio electrónico o físico.
6. La investigación, los resultados, datos, conclusiones y demás información presentada que atribuyo a mi autoría son veraces.
7. Declaro que mi trabajo final cumple con todas las normas de la Universidad de Piura.

El incumplimiento de lo declarado da lugar a responsabilidad del declarante, en consecuencia; a través del presente documento asumo frente a terceros, la Universidad de Piura y/o la Administración Pública toda responsabilidad que pueda derivarse por el trabajo final presentado. Lo señalado incluye responsabilidad pecuniaria incluido el pago de multas u otros por los daños y perjuicios que se ocasionen.

Fecha: Piura, 20 de marzo del 2023.



.....  
Firma del autor optante<sup>3</sup>

<sup>1</sup> Indicar si es tesis, trabajo de investigación, trabajo académico o trabajo de suficiencia profesional.

<sup>2</sup> Grado de Bachiller, Título de profesional, Grado de Maestro o Grado de Doctor.

<sup>3</sup> Idéntica a DNI, no se admite digital salvo certificado.



*A mi familia, por el apoyo constante.*

*A mi pequeña Sofía, la razón de mi existencia.*



### **Agradecimientos**

D. Pérez reconoce el apoyo financiero del Proyecto Concytec – Banco Mundial "Mejoramiento y Ampliación de los Servicios del Sistema Nacional de Ciencia Tecnología e Innovación Tecnológica" 8682-PE, a través de su unidad ejecutora ProCiencia [Contrato Nro. 06-2018-FONDECYT/BM], para esta investigación denominada "Diseño de un clasificador para el mantenimiento predictivo de una red de distribución eléctrica utilizando Deep Learning", ejecutada como parte del programa de Doctorado en Ingeniería con mención en Automatización, Control, y Optimización de Procesos, desarrollado en el Laboratorio de Sistemas Automáticos de Control de la Universidad de Piura, Perú.





## Resumen

La presente tesis doctoral busca proponer alternativas frente a técnicas tradicionales de inteligencia artificial aplicadas al sector eléctrico, con la finalidad de contribuir con el mantenimiento predictivo de equipos eléctricos, específicamente líneas y subestaciones. La metodología plantea la clasificación binaria de imágenes térmicas mediante la comparación de 11 arquitecturas de transfer learning (VGG16, AlexNet, MobileNet v2, VGG19, ResNet, DenseNet, ShuffleNet, GoogLeNet, Wide ResNet, MNASNet y ResNeXt). Se desarrolló una estructura base en la que se fueron realizando pruebas con cada una de ellas, utilizando un set de datos de 815 imágenes, además de entrenar a los modelos con dos técnicas: en primer lugar, la separación hold-out (60-20-20) y a continuación una validación cruzada con 5-folds, para posteriormente estudiar el rendimiento de cada arquitectura a través del método estadístico del test de Friedman. Los resultados mostraron que la arquitectura VGG16 obtuvo mejores prestaciones, obteniendo un valor de F1-score del 95.11%. Adicionalmente, como aporte a la investigación, se realizó la detección de objetos con el mismo set de datos, para lo cual, de las 815 imágenes del set de datos original (342 líneas y 473 subestaciones eléctricas), se obtuvieron únicamente 138 imágenes con puntos calientes (22 líneas y 116 subestaciones), por lo que se decidió trabajar exclusivamente con las subestaciones eléctricas, a las cuales se les aplicó técnicas de data augmentation y transfer learning. Se analizó y aplicó el algoritmo YOLO y sus distintas variaciones (1 experimento con YOLOv4, 16 experimentos con YOLOv5 small, 16 con YOLOv5 medium, 16 con YOLOv5 large y 16 con YOLOv5 extra-large), obteniendo porcentajes de precisión de 81.99% con YOLOv5m. Ambos resultados son un claro indicio de que el uso de técnicas no tradicionales de inteligencia artificial representa una alternativa fiable en la clasificación y detección de objetos en imágenes térmicas aplicadas al sector eléctrico.



## Tabla de contenido

Introducción	17
Capítulo 1 Estado del arte	23
1.1 Mantenimiento predictivo	23
1.1.1 Definiciones	23
1.1.2 Ventajas y desventajas	25
1.2 Equipos eléctricos	26
1.2.1 Líneas eléctricas	26
1.2.2 Subestaciones eléctricas	26
1.3 Termografía	27
1.3.1 Cámaras termográficas	28
1.3.2 Aplicaciones de la termografía	29
1.4 Inteligencia artificial	31
1.4.1 Machine Learning	32
1.4.2 Deep Learning	34
1.5 Investigaciones relevantes	43
Capítulo 2 Metodología	59
2.1 Área de estudio	59
2.2 Justificación	59
2.3 Objetivos	62
2.3.1 Objetivo general	62
2.3.2 Objetivos específicos	62
2.4 Hipótesis	62
2.5 Propuesta de solución	62
Capítulo 3 Modelo	65
3.1 Set de datos	65
3.2 Arquitecturas de transfer learning	66

3.2.1	AlexNet	69
3.2.2	VGG16	71
3.2.3	VGG19	73
3.2.4	ResNet	75
3.2.5	DenseNet	77
3.2.6	GoogLeNet	79
3.2.7	MobileNet v2	81
3.2.8	ResNeXt	83
3.2.9	Wide ResNet	85
3.2.10	MNASNet	88
3.2.11	ShuffleNet	90
Capítulo 4 Resultados		95
Conclusiones		107
a.	Conclusiones	107
b.	Conclusiones: aporte adicional (ver Apéndice 01)	107
c.	Conclusión final	109
d.	Trabajos futuros	109
Glosario		111
Referencias bibliográficas		113
Apéndices		127
Apéndice 01: Aporte adicional - Detección de puntos calientes		127
Apéndice 02: Artículo en revista WoS		182
Apéndice 03: Artículo en revista Latindex		183
Apéndice 04: Artículo en revista Scopus		184
Apéndice 05: Artículo en revista SciELO		185
Apéndice 06: Artículo en congreso Scopus (1)		186
Apéndice 07: Artículo en congreso Scopus (2)		187
Apéndice 08: Artículo en congreso Scopus (3)		188

## Lista de figuras

Figura 1. Estructura de una red neuronal artificial tradicional.	17
Figura 2. Técnica de separación de datos tipo hold-out.	18
Figura 3. Técnica de validación cruzada con k-folds.	19
Figura 4. Imagen térmica de un equipo de alta tensión.	20
Figura 5. Requisitos para generar un plan de mantenimiento predictivo.	24
Figura 6. Implementación del mantenimiento predictivo.	25
Figura 7. Ejemplo de línea de transmisión.	26
Figura 8. Ejemplo de subestación eléctrica.	27
Figura 9. Espectro electromagnético.	27
Figura 10. Ejemplo de termografía de un vehículo.	28
Figura 11. Cámara termográfica.	29
Figura 12. Envoltorio térmico.	30
Figura 13. Imagen térmica - sobrecalentamiento eléctrico.	30
Figura 14. Componentes del Machine Learning.	32
Figura 15. Algoritmos de Machine Learning.	33
Figura 16. Red neuronal artificial.	34
Figura 17. Técnicas de Deep Learning.	35
Figura 18. Red neuronal clásica.	36
Figura 19. Convolutional Neural Networks.	36
Figura 20. Etapas de la construcción de una CNN.	37
Figura 21. Recurrent Neural Networks.	38
Figura 22. Generative Adversarial Networks.	38
Figura 23. Self-Organizing Maps.	39
Figura 24. Boltzmann Machines.	39
Figura 25. Deep Reinforcement Learning.	40
Figura 26. Autoencoders.	41
Figura 27. Backpropagation.	41
Figura 28. Gradient Descent,	42
Figura 29. Inteligencia artificial, Machine Learning y Deep Learning.	42
Figura 30. (A) Algoritmo JSEG; (B) Algoritmo de segmentación espacial.	44
Figura 31. Arquitectura neuronal convolucional YOLO, basada en Alexnet.	44
Figura 32. Puntos calientes detectados después de la segmentación JSEG.	45
Figura 33. Resultados del entrenamiento – Accuracy.	45
Figura 34. Modelo estándar de MLP.	46
Figura 35. Diagrama para la detección de defectos / sin defectos.	47
Figura 36. Pruebas de rendimiento	47

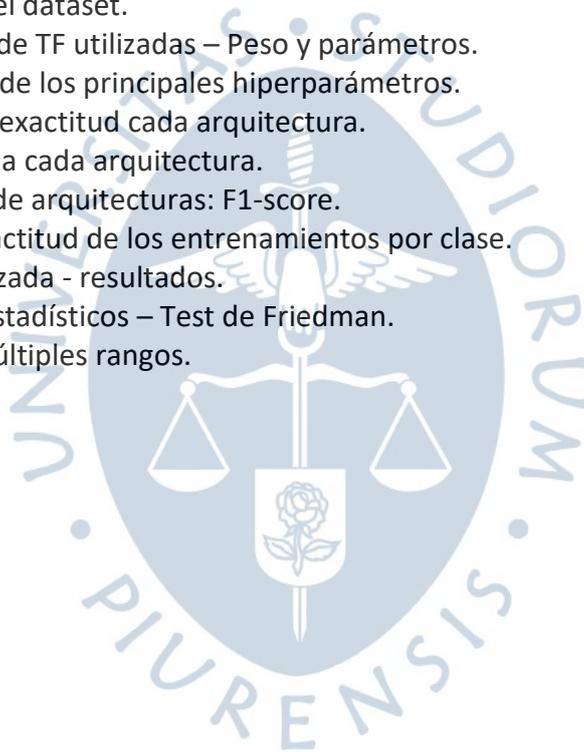
Figura 37. Imagen térmica de muestra con posibles defectos esperados.	48
Figura 38. Comparación de rendimiento de MLP y MLP con corte de gráfico.	48
Figura 39. Rendimiento de los algoritmos en el set de datos de prueba.	49
Figura 40. Clasificación utilizando el análisis discriminante.	50
Figura 41. Imágenes infrarrojas originales.	50
Figura 42. K-means: extracción de características.	51
Figura 43. Resultados del clustering.	51
Figura 44. Configuración experimental para monitoreo de condición con IRT.	52
Figura 45. IRT en el mantenimiento preventivo en transformadores eléctricos.	53
Figura 46. Imágenes grises térmicas antes y después del procesamiento de imágenes.	54
Figura 47. Equipo eléctrico (a) Nivel I (b) Nivel II y (c) Nivel III.	54
Figura 48. Flujo para encontrar la configuración óptima de parámetros.	55
Figura 49. Curva ROC.	56
Figura 50. Evolución del coeficiente de electrificación nacional peruano.	59
Figura 51. Duración promedio de las interrupciones (SAIDI) en Perú.	60
Figura 52. Frecuencia promedio de interrupciones (SAIFI) en Perú.	60
Figura 53. SAIDI y SAIFI por regiones	60
Figura 54. Proceso de detección de fallos.	62
Figura 55. Propuesta de solución.	63
Figura 56. Metodología general del proyecto.	63
Figura 57. Imagen típica de equipos de líneas de transmisión y subestaciones eléctricas.	65
Figura 58. Distribución del dataset.	66
Figura 59. Arquitectura base.	67
Figura 60. Imágenes con data augmentation.	68
Figura 61. Arquitectura AlexNet.	69
Figura 62. AlexNet: Accuracy en Entrenamiento y Validación.	69
Figura 63. AlexNet: Tasa de pérdida en Entrenamiento y Validación.	70
Figura 64. AlexNet: Matriz de confusión.	70
Figura 65. AlexNet: Curva ROC.	70
Figura 66. Arquitectura VGG16.	71
Figura 67. VGG16: Accuracy en Entrenamiento y Validación.	71
Figura 68. VGG16: Tasa de pérdida en Entrenamiento y Validación.	72
Figura 69. VGG16: Matriz de confusión.	72
Figura 70. VGG16: Curva ROC.	72
Figura 71. Arquitectura VGG19.	73
Figura 72. VGG19: Accuracy en Entrenamiento y Validación.	73
Figura 73. VGG19: Tasa de pérdida en Entrenamiento y Validación.	74
Figura 74. VGG19: Matriz de confusión.	74
Figura 75. VGG19: Curva ROC.	74
Figura 76. Bloque de la arquitectura ResNet.	75
Figura 77. ResNet: Accuracy en Entrenamiento y Validación.	75
Figura 78. ResNet: Tasa de pérdida en Entrenamiento y Validación.	76
Figura 79. ResNet: Matriz de confusión.	76
Figura 80. ResNet: Curva ROC.	76
Figura 81. Bloque de la arquitectura DenseNet.	77
Figura 82. DenseNet: Accuracy en Entrenamiento y Validación.	77
Figura 83. DenseNet: Tasa de pérdida en Entrenamiento y Validación.	78

Figura 84. DenseNet: Matriz de confusión.	78
Figura 85. DenseNet: Curva ROC.	78
Figura 86. Módulo inception con reducciones de dimensión.	79
Figura 87. GoogLeNet: Accuracy en Entrenamiento y Validación.	79
Figura 88. GoogLeNet: Tasa de pérdida en Entrenamiento y Validación.	80
Figura 89. GoogLeNet: Matriz de confusión.	80
Figura 90. GoogLeNet: Curva ROC.	80
Figura 91. Bloques de la arquitectura MobileNet v2.	81
Figura 92. MobileNet v2: Accuracy en Entrenamiento y Validación.	81
Figura 93. MobileNet v2: Tasa de pérdida en Entrenamiento y Validación.	82
Figura 94. MobileNet v2: Matriz de confusión.	82
Figura 95. MobileNet v2: Curva ROC.	82
Figura 96. Bloque de la arquitectura ResNeXt.	83
Figura 97. ResNeXt: Accuracy en Entrenamiento y Validación.	83
Figura 98. ResNeXt: Tasa de pérdida en Entrenamiento y Validación.	84
Figura 99. ResNeXt: Matriz de confusión.	84
Figura 100. ResNeXt: Curva ROC.	85
Figura 101. Bloque wide- dropout.	85
Figura 102. Wide ResNet: Accuracy en Entrenamiento y Validación.	86
Figura 103. Wide ResNet: Tasa de pérdida en Entrenamiento y Validación.	86
Figura 104. Wide ResNet: Matriz de confusión.	87
Figura 105. Wide ResNet: Curva ROC.	87
Figura 106. Capa convolucional de MNASNet.	88
Figura 107. MNASNet: Accuracy en Entrenamiento y Validación.	88
Figura 108. MNASNet: Tasa de pérdida en Entrenamiento y Validación.	89
Figura 109. MNASNet: Matriz de confusión.	89
Figura 110. MNASNet: Curva ROC.	89
Figura 111. Unidad ShuffleNet.	90
Figura 112. ShuffleNet: Accuracy en Entrenamiento y Validación.	90
Figura 113. ShuffleNet: Tasa de pérdida en Entrenamiento y Validación.	91
Figura 114. ShuffleNet: Matriz de confusión.	91
Figura 115. ShuffleNet: Curva ROC.	91
Figura 116. Validación cruzada con 5-folds.	95
Figura 117. Gráfico de residuos.	96
Figura 118. Gráfico de probabilidad normal.	97
Figura 119. Gráfico de medias – Fisher LSD.	98
Figura 120. Gráfico de medias – Tukey HSD.	98
Figura 121. Gráfico de medianas.	99
Figura 122. Propuesta de arquitectura de transfer learning.	100
Figura 123. Accuracy: entrenamiento y validación del modelo.	101
Figura 124. Matriz de confusión.	101
Figura 125. Curva ROC.	102
Figura 126. Predicciones del modelo con VGG16.	102
Figura 127. Resumen de resultados.	105



## Lista de tablas

Tabla 1. Principales investigaciones – Clasificación de imágenes térmicas.	56
Tabla 2. Distribución del dataset.	66
Tabla 3. Arquitecturas de TF utilizadas – Peso y parámetros.	66
Tabla 4. Configuración de los principales hiperparámetros.	68
Tabla 5. Porcentaje de exactitud cada arquitectura.	92
Tabla 6. Tasa de pérdida cada arquitectura.	93
Tabla 7. Comparación de arquitecturas: F1-score.	93
Tabla 8. Duración y exactitud de los entrenamientos por clase.	94
Tabla 9. Validación cruzada - resultados.	96
Tabla 10. Resultados estadísticos – Test de Friedman.	97
Tabla 11. Prueba de múltiples rangos.	99

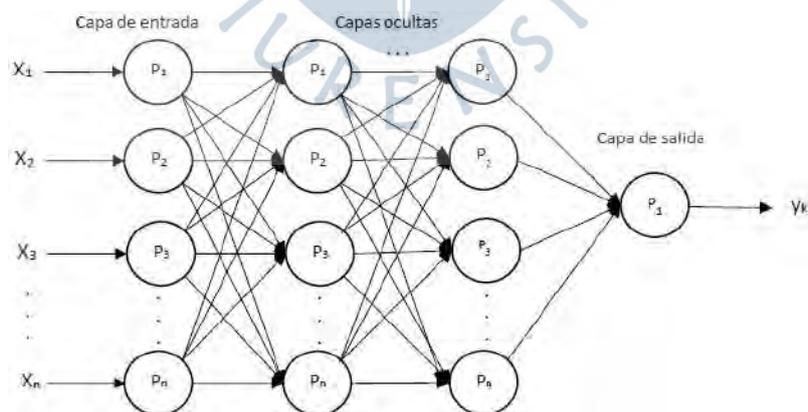




## Introducción

La inteligencia artificial (IA) es considerada como una disciplina en auge en los últimos años, debido a que ha contribuido en la mejora de procesos en diversos sectores, como por ejemplo en el industrial. Algunas de sus aplicaciones más relevantes incluyen la visión por computador, reconocimiento de voz, análisis de textos, procesamiento de imágenes o videos, entre muchos otros (Haenlein & Kaplan, 2019). La inteligencia artificial no es un concepto reciente, pues viene tomando presencia desde hace varias décadas, sin embargo, lo que variado estos últimos años es el incremento de potencia computacional, facilitándose considerablemente el procesamiento de modelos y set de datos complejos (Flasiński & Flasiński, 2016). El objetivo principal de la IA es emular las capacidades y/o comportamientos de los seres humanos, para lo cual se vale de un gran conjunto de algoritmos, técnicas y arquitecturas (Huang & Rust, 2018). Al igual que en muchas otras disciplinas, la IA se encuentra compuesta por otros elementos, tales como el Aprendizaje automático o Machine Learning (ML) y el Aprendizaje profundo o Deep Learning (DL) (Aung et al., 2020).

**Figura 1.** Estructura de una red neuronal artificial tradicional.

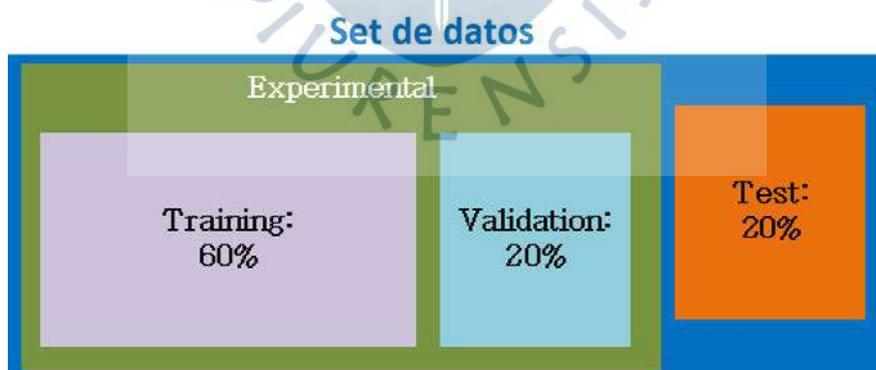


Tomado de “Comparative study between multiclass support vector machines, neural networks and self-organized neuro-fuzzy inference system for classification problems” (Galindo et al., 2020).

El aprendizaje automático (ML) busca la generalización del comportamiento de los conjuntos de datos de entrada, o visto de otra manera, la obtención de predicciones de comportamientos basados en el análisis de patrones en grandes conjuntos de datos (Carvalho et al., 2019). En este mismo sentido, el aprendizaje profundo (DL) forma parte del aprendizaje automático, siendo su objetivo principal que los sistemas informáticos intenten imitar automáticamente los comportamientos y razonamientos de los seres humanos (Wan et al., 2020), teniendo como elemento clave a las ANN o redes neuronales artificiales (**Figura 1**), las cuales buscan simular el proceso sináptico de las neuronas del cerebro humano (Golodov et al., 2017). Asimismo, dentro de las redes neuronales artificiales, encontramos a las CNN o Redes Neuronales Convolucionales, las cuales son mayormente utilizadas en problemas de clasificación de imágenes debido a su alta complejidad (Brunetti et al., 2018).

Si hablamos de técnicas de separación del set de datos, una de las más utilizadas es hold-out (Yildiz et al., 2019), la cual consiste en dividir el set de datos original en sub conjuntos de entrenamiento, validación y pruebas (60%, 20% y 20% respectivamente), tal como se aprecia en la **Figura 2**. Sin embargo, en el ámbito de la investigación esta técnica suele complementarse con otras como la validación cruzada, debido a que esta última reduce la variabilidad que la primera puede acarrear, para ello se utiliza un conjunto de k-folds (Y. Jung, 2018) y la consecuente división del set de datos de manera aleatoria en k subconjuntos, de los cuales k-1 son utilizados en el entrenamiento del modelo, y uno para la validación del mismo (**Figura 3**). Además, si requerimos realizar comparaciones entre distintos modelos (Pacheco et al., 2016), es sumamente recomendable aplicar técnicas estadísticas paramétricas como ANOVA, o no paramétricas, tales como el Test de Friedman (Zimmerman & Zumbo, 2010).

**Figura 2.** Técnica de separación de datos tipo hold-out.

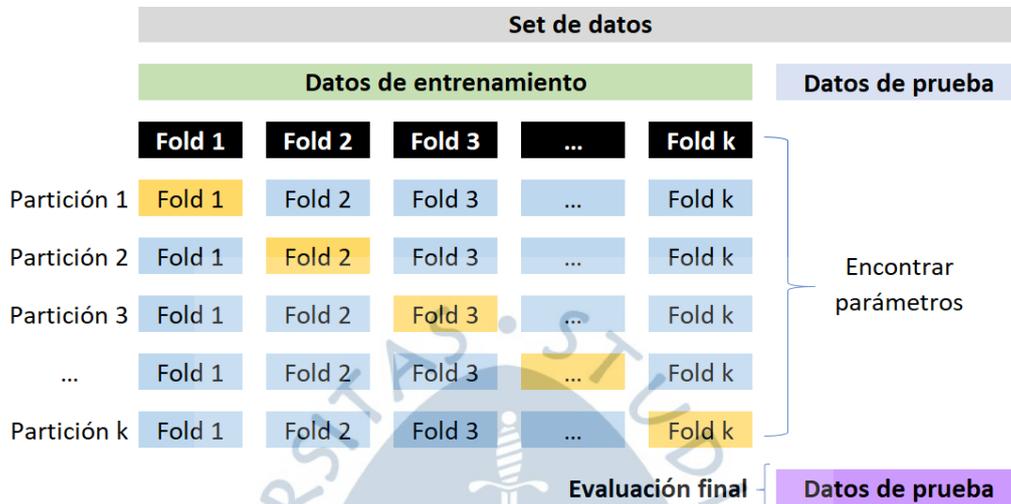


Nota. Elaboración propia a partir de “*Classification and comparison via neural networks*” (Yildiz et al., 2019).

A continuación, se describen algunas aplicaciones de la IA en el sector eléctrico, como por ejemplo en la automatización de la clasificación de imágenes térmicas (Lile & Yiqun, 2018), las cuales son capturadas con el apoyo de equipos especializados: las cámaras infrarrojas. Es aquí donde la termografía infrarroja toma relevancia, ya que esta permite

obtener la radiación infrarroja a través del espectro electromagnético, evitando el contacto directo con el equipo u objeto a estudiar, por lo que no es necesario que estos dejen de funcionar para ser analizados (Dragomir et al., 2017). En la Figura 4 se aprecia una imagen térmica de un equipo eléctrico de alta tensión, la cual fue capturada a distancia.

**Figura 3.** Técnica de validación cruzada con *k*-folds.



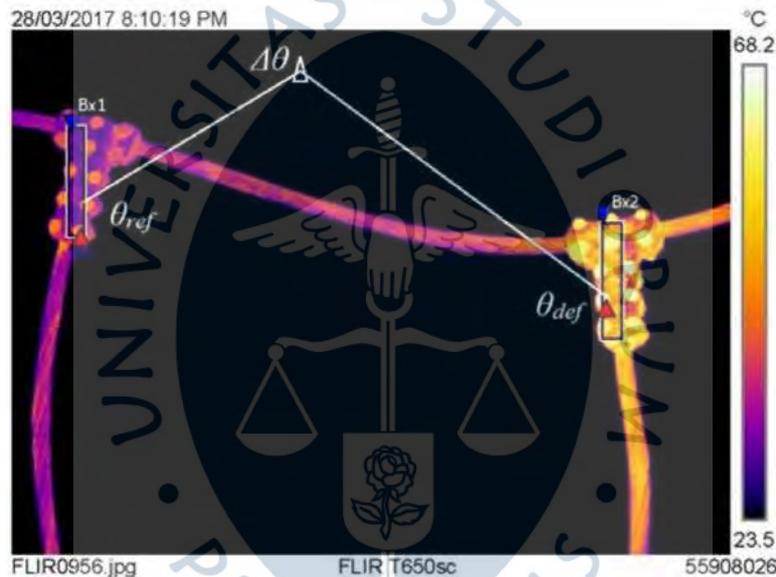
Nota. Elaboración propia a partir de “Multiple predicting *K*-fold cross-validation for model selection” (Y. Jung, 2018).

Una investigación describe un sistema informático de reconocimiento de imágenes termográficas en una red de distribución eléctrica (Fambrini et al., 2018), en la cual se incluyeron el algoritmo de segmentación JSEG en combinación con una CNN (W. Yang et al., 2018). Del mismo modo que una investigación desarrollada en China, en la cual se aplican técnicas de visión por computador y el uso de imágenes térmicas para analizar perturbaciones eléctricas en edificaciones, para ello, entrenaron un MLP (MultiLayer Perceptron o perceptrón multicapa) (Ullah et al., 2017) utilizado para dar solución a problemas no separables linealmente (Ramchoun et al., 2016). Otro estudio considera un enfoque semiautomático con la finalidad de analizar las condiciones térmicas de determinadas instalaciones eléctricas de un edificio (Nazmul Huda et al., 2012), para lo cual se apoyan en imágenes infrarrojas, un MLP y el análisis de componentes principales o PCA (Yuan et al., 2016); mientras que en (Zou & Huang, 2015), se estudia el diagnóstico inteligente mediante la clasificación de las condiciones de equipos eléctricos, para ello utilizan el algoritmo K-means, entrenado con un conjunto de imágenes térmicas, logrando clasificar clusters o grupos con características similares (Yu et al., 2018).

Estas investigaciones entrenan determinados modelos de manera tradicional, sin embargo, en la actualidad se cuenta con distintas técnicas que contribuyen a mejorar la calidad de los entrenamientos y reducir los tiempos de ejecución, una de ellas es el Transfer Learning (TF), el cual está incluido como parte del Deep Learning. Su funcionamiento principal radica en no entrenar una red neuronal desde cero, sino utilizar una red pre entrenada o

entrenada previamente con otros set de datos; en otras palabras, busca reusar los pesos, y en algunos casos la arquitectura, de un modelo entrenado con set de datos de entrada de tamaño considerable (como por ejemplo ImageNet, una base de datos con alrededor de 14 millones de imágenes (Ebrahim et al., 2019)), y aplicarlos a escenarios distintos con nuevos set de datos, con lo cual se logra clasificar de manera más rápida y se utiliza una menor carga computacional (Phan et al., 2020). Aquí entra a tallar el concepto de fine-tuning de un modelo, el cual es un paradigma del TF que intenta adaptar estos pesos a nuevos escenarios o dominios de aplicación (Alshalali & Josyula, 2018); es decir que, se toman los pesos del modelo previamente entrenado, y se van realizando variaciones a distintos parámetros, como la tasa de aprendizaje, buscando obtener rendimientos superiores en las predicciones (Vrbancic & Podgorelec, 2020).

**Figura 4.** Imagen térmica de un equipo de alta tensión.



Tomado de “*Considerations regarding infrared thermal stresses monitoring of electrical equipment*” (Dragomir et al., 2017).

La revisión de la literatura muestra que el transfer learning viene siendo ampliamente utilizado en sectores como (i) agroindustrial (Cengil & Cinar, 2019), mediante la comparación de algunas arquitecturas de transfer learning como VGG16, DenseNet, ResNet, GoogleNet, y AlexNet, con el objetivo de clasificar un set de datos que contiene imágenes de flores, siendo la arquitectura VGG16 con los pesos previamente entrenados la que obtuvo rendimientos superiores a las demás. (iii) En el sector alimenticio (Rajayogi et al., 2019), donde se señala que las redes neuronales convolucionales son unas de las técnicas de clasificación de imágenes más utilizadas, por lo que realizan clasificación de alimentos con el objetivo de obtener un estilo de vida más saludable; por lo cual consideran a las arquitecturas previamente entrenadas :ResNet, VGG16, InceptionV3, y VGG19. O (i) en el sector salud, mediante el apoyo en la detección de patologías en imágenes neurológicas (Kaur & Gandhi, 2019),

detección de objetos en imágenes de Rayos X (Galvez et al., 2019) o cervicales (Xue et al., 2020), entre otros.

Al analizar el uso de técnicas de transfer learning e imágenes térmicas en el sector eléctrico, nos encontramos con una escasez de investigaciones, siendo algunas de las más resaltantes: (shao et al., 2020), en la que se estudia un mecanismo de clasificación de imágenes térmicas de los sistemas de rodamientos de rotores; para ello consideran el uso de una CNN y la aplicación de transfer learning, sin embargo, no se detalla la arquitectura de transfer learning utilizada. En otro caso, se automatiza el monitoreo del estado distintos tipos de maquinaria y/o equipo industrial, para lo cual consideran el uso de una red neuronal convolucional y un set de datos de imágenes térmicas, señalando que una desventaja de las CNN es el requisito de contar con un set de datos de tamaño considerable, por lo que optan por el análisis y aplicación de VGG16 (Janssens et al., 2018).

La información indicada en párrafos previos muestra la necesidad de contar con estudios científicos de la aplicación de transfer learning en el sector eléctrico, ya que no se evidenciaron investigaciones que apliquen diversas arquitecturas con un set de datos de imágenes térmicas y realicen la correspondiente comparación de las mismas. Es por este motivo que, la presente tesis doctoral busca aplicar el uso de transfer learning en la clasificación de imágenes térmicas de líneas y subestaciones eléctricas, además de proponer alternativas al uso de técnicas tradicional de inteligencia artificial como SVM, MLP, entre otras.

La tesis doctoral se distribuye de la siguiente manera: (i) Capítulo 1, estado del arte, en donde se describen conceptos clave e investigaciones relevantes, (ii) Capítulo 2, Metodología, incluyendo el área de estudio, la justificación, objetivos, hipótesis y propuesta de solución, (iii) Capítulo 3, describe modelo, set de datos y arquitecturas de transfer learning utilizadas, (iv) Capítulo 4, los resultados luego entrenar, validar y probar los modelos, (v) Capítulo 5, las conclusiones y recomendaciones del estudio. Adicionalmente, en la sección Anexos se ubican los papers publicados por el autor, además de un aporte complementario a la tesis, el cual tiene que ver con la detección de puntos calientes en imágenes térmicas con la aplicación del algoritmo YOLO.



## Capítulo 1 Estado del arte

### 1.1 Mantenimiento predictivo

#### 1.1.1 Definiciones

El mantenimiento predictivo es considerado como un conjunto de tareas o técnicas ejecutadas con la finalidad de detectar posibles fallos o defectos en la maquinaria, principalmente en temas industriales; esta detección permite conocer de antemano y en etapas tempranas, problemas que a futuro pueden provocar grandes pérdidas de tiempo y dinero debido a los tiempos muertos o paradas de emergencia. El mantenimiento predictivo busca mantener los equipos industriales operando con un nivel de servicio adecuado, por lo que este análisis se realiza, aunque los equipos no hayan presentado anomalías (Kanawaday & Sane, 2018).

Dentro del mantenimiento predictivo se incluyen variables físicas y de estado (desgaste) de las máquinas, basándose en la medición y un continuo monitoreo de las condiciones y parámetros operativos de los equipos o maquinaria. En la literatura, se recomienda establecer y gestionar valores o indicadores de alarma antes de que suceda algún inconveniente, los cuales son denominados como valores de pre-alarma (Cachada et al., 2018). Se puede afirmar entonces, que el mantenimiento predictivo realiza una evaluación constante de los equipos, aunque estos no presenten síntomas de problemas, por lo que recomienda o propone intervenir en ellos o no, de acuerdo al resultado de esta evaluación, generando ahorros monetarios.

En relación a su historia, esta técnica se introdujo a la industria en el siglo XX, con la finalidad de optimizar los activos de las empresas, considerándose como un mecanismo que ayuda a determinar el momento preciso en que estos requieren tareas de mantenimiento (Selcuk, 2017). Desde ese momento, su función principal fue buscar la optimización y disponibilidad de los equipos industriales a un bajo coste, basándose en la categorización de fallos o problemas potenciales.

Para aplicar este tipo de mantenimiento, no se requiere que los equipos o maquinaria produzcan síntomas o señales de existencia de un fallo, representados en ruido, temperatura, vibración, ultrasonido, amperaje, partículas de desgaste, entre otros; con lo cual, el

mantenimiento predictivo pueda aumentar el estado operacional y el nivel de fiabilidad de la maquinaria, a su vez que reduzca los costos de mantenimiento y producción. (De Benedetti et al., 2018)

Para elaborar y ejecutar un plan de mantenimiento predictivo, se requiere comprender los distintos tipos de fallos y efectos provocados en la maquinaria, además de tener en cuenta las ventajas, limitaciones, justificación económica y equipo de profesionales altamente calificados en esta técnica de mantenimiento; sin embargo, también se hace necesario un cambio en la cultura organizacional, buscando que el mantenimiento predictivo sobresalga en relación al correctivo (Ver **Figura 5**) (Q. Liu et al., 2019).

**Figura 5.** Requisitos para generar un plan de mantenimiento predictivo.



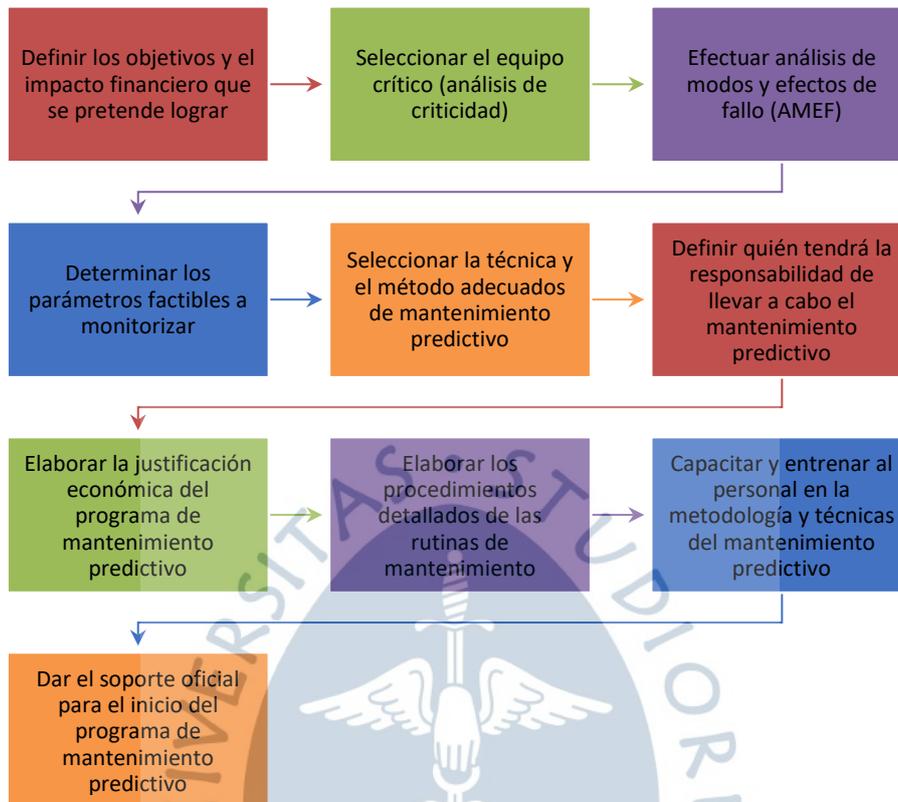
Nota. Elaboración propia a partir de *"Single-machine-based joint optimization of predictive maintenance planning and production scheduling"* (Q. Liu et al., 2019).

La **Figura 6** muestra las etapas para una correcta implantación del mantenimiento predictivo (Kadechkar et al., 2019).

Algunas técnicas utilizadas para el descubrimiento o localización de fallos o defectos en la maquinaria son (D. Jung et al., 2017):

- Análisis tradicional mediante el uso de los sentidos humanos.
- Análisis de vibraciones.
- Uso de ultrasonidos.
- Análisis de Lubricantes.
- Análisis de Máquinas Alternativas.
- Termografía:
- Etc.

**Figura 6. Implementación del mantenimiento predictivo.**



Nota. Elaboración propia a partir de “*Vibration analysis for iot enabled predictive maintenance*” (D. Jung et al., 2017).

### 1.1.2 Ventajas y desventajas

El mantenimiento predictivo presenta una serie de aspectos positivos, sin embargo, también se deben considerar las limitaciones del mismo (Sakib & Wuest, 2018). Algunas de las ventajas del mantenimiento predictivo son:

- Detección de fallas en etapas tempranas.
- Reducción de tiempos muertos, debido a la anticipación de los fallos.
- Realización de inspecciones a la maquinaria en funcionamiento u operación mediante equipos especializados, evitando que la máquina se detenga y/o desmonte.
- Mejora el conocimiento sobre el funcionamiento de la maquinaria.
- Mejora la fiabilidad de las inspecciones.
- Reducción de costos y tiempo de mantenimiento.
- Mayor disponibilidad de los equipos y maquinaria.
- Disminución de pérdidas de materia prima provocadas por paradas innecesarias, no planificadas, o por consecuencia de los re-arranques.
- Disminución de las intervenciones de operarios a los equipos.
- Reducción de costos en repuestos de partes de la maquinaria.

- Reducción de costos en reparaciones y/o mano de obra.
- Incremento de seguridad y reducción de accidentes.

Mientras que sus desventajas se detallan a continuación:

- Inversión inicial elevada, ya que se requieren equipos o tecnología costosa.
- Profesionales altamente calificados en esta técnica de mantenimiento.

## 1.2 Equipos eléctricos

Los equipos eléctricos en los que se centra la investigación son las líneas de transmisión y las subestaciones, los cuales se detallan a continuación.

### 1.2.1 Líneas eléctricas

Las líneas eléctricas se definen como los medios físicos a través de los cuales se efectúa tanto la transmisión, así como la distribución de la energía eléctrica (Ebrahimi et al., 2018).

Según su función, las líneas eléctricas se clasifican en líneas de distribución y líneas de transmisión; las primeras van desde las subestaciones en dirección a los centros de consumo, generalmente ubicados en zonas urbanas (domicilios, empresas industriales, alumbrado público, etc.), mientras que las segundas son utilizadas en el transporte de energía en grandes distancias, por lo que son un mecanismo de unión entre las redes de distribución y las centrales generadoras de energía. Una línea de transmisión está compuesta por elementos como aisladores, conductores, cables de guarda, estructuras de soporte, entre otros. La **Figura 7** muestra un ejemplo de una línea de transmisión (S. Karimi et al., 2018).

**Figura 7.** Ejemplo de línea de transmisión.



Tomado de “Se licitarán líneas de transmisión por US\$/100 millones” (Redacción EC, 2018).

### 1.2.2 Subestaciones eléctricas

Son equipos que tienen por finalidad la realización de transformaciones de frecuencia, tensión, número de fases, o conexión de circuitos, por lo que, generalmente se ubican cerca de las centrales generadoras de energía o en zonas no urbanas (Iglesias-Urkia et al., 2019). La **Figura 8** muestra un ejemplo de una subestación eléctrica.

Las subestaciones se clasifican en equipos de maniobra y de transformación, las primeras se encargan de conectar circuitos y ejecutar sus maniobras, es decir que la tensión

no es transformada; mientras que las segundas tienen por objetivo realizar la transformación de la energía, para lo cual necesitan utilizar transformadores (Lima et al., 2018).

**Figura 8.** Ejemplo de subestación eléctrica.

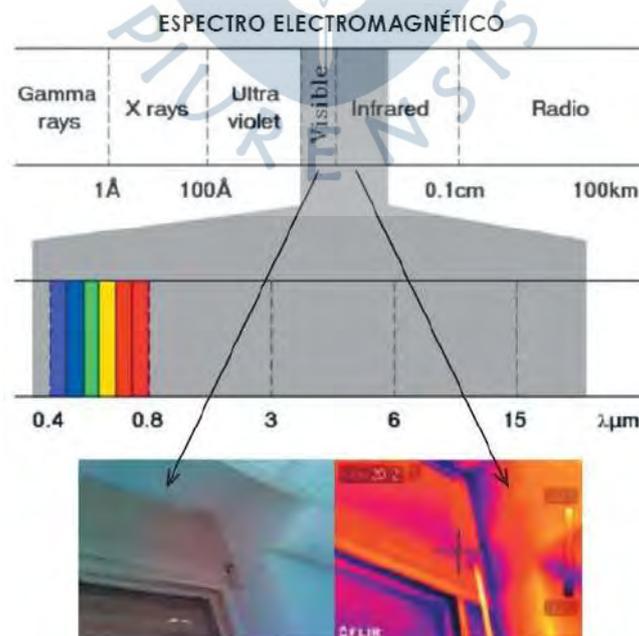


Tomado de "Subestaciones Eléctricas" (Redondo, 2019).

### 1.3 Termografía

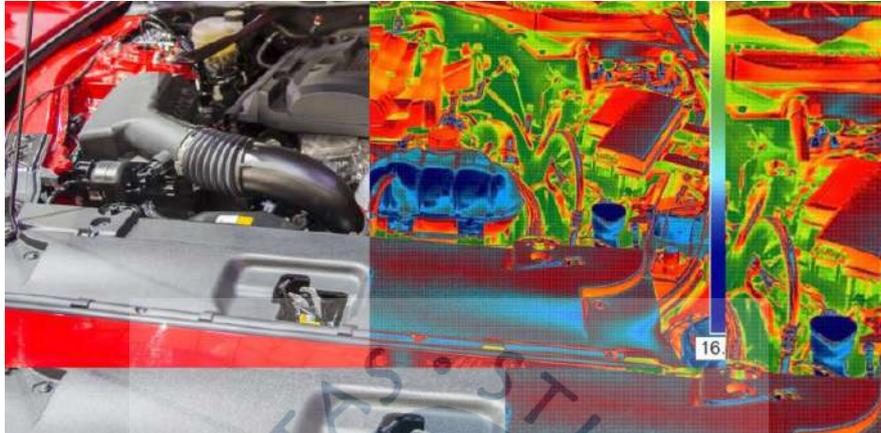
Técnica utilizada para la determinación de temperatura en distintos objetos sin la necesidad de tener contacto físico con los mismos (Wang et al., 2018); para ello capta la radiación infrarroja del espectro electromagnético mediante el uso de cámaras térmicas, las cuales generan imágenes infrarrojas o radiométricas, en las que cada píxel representa un valor de medición de la radiación en relación a la temperatura captada (Ciampa et al., 2018). En la **Figura 9** se observa el proceso de la generación de una imagen térmica a partir del espectro electromagnético, mientras que, en la **Figura 10** se muestra la imagen térmica de un vehículo.

**Figura 9.** Espectro electromagnético.



Tomado de *“Image processing based quantitative damage evaluation in composites with long pulse thermography”* (Wang et al., 2018).

**Figura 10.** Ejemplo de termografía de un vehículo.



Tomado de *“Soluciones punteras en termografía, al servicio de la Industria Automotriz”* (Bezares, 2019).

Con este tipo de imágenes se pueden realizar análisis más detallados con la finalidad de conocer el estado térmico del equipo, fugas de energía, entre otros (Lucchi, 2018), como por ejemplo: quemaduras de fusibles, bornas sobrecalentadas, conexiones inadecuadas, aislamientos ineficientes, entre otros (R. Yang & He, 2016). Las utilidades de la termografía son diversas, algunas de las cuales se detallan a continuación (Ghahramani et al., 2018):

- Mantenimiento predictivo industrial.
- Análisis de edificaciones.
- Análisis de pérdidas energéticas.
- Detección de puente térmico.
- Detección de gases.
- Detección de puntos calientes.
- Estudios médicos.
- Meteorología.
- Aspectos de seguridad o temas militares.
- Etc.

### 1.3.1 Cámaras termográficas

La termografía no sería posible sin el uso de instrumentos que permitan la generación de imágenes térmicas: las cámaras termográficas, las cuales permiten observar la radiación que emite determinado objeto mediante el uso del espectro electromagnético; este tipo de cámaras trabajan con longitudes de onda del infrarrojo término, es decir  $3 \mu\text{m} - 14 \mu\text{m}$ . Para realizar este proceso, se capta la radiación infrarroja emitida por un objeto mediante un detector de radiación, mostrando una serie de colores que representan las señales de la

radiación de onda larga reflejadas en un patrón de comportamiento térmico (Yakunin et al., 2019). En la **Figura 11** se observa el funcionamiento de una cámara termográfica.

**Figura 11.** Cámara termográfica.



Tomado de “Introducción a las cámaras termográficas” (De máquinas y herramientas, 2015).

Un análisis adecuado requiere de profesionales calificados en el uso de cámaras termográficas, además del software necesario para su uso. Es necesario que las cámaras utilizadas presenten una alta resolución, lo cual contribuye al incremento de la precisión y fiabilidad de la información (Procházka et al., 2017). Cuando se trabaja con imágenes térmicas se deben seguir algunas recomendaciones (Szajewska, 2017):

- Encuadre: La imagen debe tener un encuadre correcto, ya que se busca que el objeto capturado brinde la mayor información posible.
- Enfoque: La imagen debe ser enfocada con la mayor precisión posible, con lo cual se disminuye el riesgo de presentar errores en las medidas.
- Radiación: Los niveles de radiación deben ser los adecuados, evitando la que se genera por exposición directa al sol, por lo que se recomienda que las capturas sean realizadas cuando el sol no incide directamente.

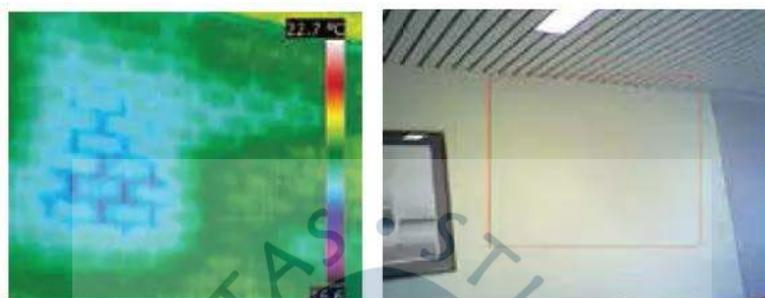
### 1.3.2 Aplicaciones de la termografía

Las aplicaciones de la termografía son muy diversas, a continuación se exponen las más relevantes (Netzelmann et al., 2016):

- En relación a la envolvente térmica, podemos encontrar las siguientes aplicaciones:
  - o Análisis del estado de zonas con puentes térmicos. En la **Figura 12** se aprecia un ejemplo de la envolvente térmica, en la cual se evidencia una falta de aislamiento.

- Estudio de fugas de aire, ya que generalmente, en estas zonas se suelen presentar pérdidas de energía, lo que a su vez genera problemas con los sistemas de ventilación.
- Análisis de zonas con presencia de niveles altos de humedad, los cuales puede acarrear problemas en edificaciones.
- Análisis de edificaciones con continuas pérdidas de calefacción.

**Figura 12.** *Envolvente térmica.*



Tomado de “*Induction thermography: principle, applications and first steps towards standardisation*” (Netzelmann et al., 2016).

- En relación a las instalaciones de sistemas, podemos encontrar las siguientes aplicaciones:
  - Estudio de problemas o fallos en las instalaciones eléctricas o en alguno de sus componentes. En la **Figura 13** se aprecia un ejemplo una imagen térmica con presencia de sobrecalentamiento eléctrico.
  - Análisis de problemas en instalaciones mecánicas. Ejemplo: climatización, sistemas contra incendio, agua caliente, etc.
  - Estudio del estado de elementos como motores o equipos de bombeo, detectando posibles recalentamientos, complicaciones en la lubricación, etc.; evitando que estos problemas escalen a un nivel más complejo e incurran en costos de reparación más elevados.
  - Verificación del estado de paneles fotovoltaicos y mantenimiento de los mismos.

**Figura 13.** *Imagen térmica - sobrecalentamiento eléctrico.*



Tomado de *“Induction thermography: principle, applications and first steps towards standardisation”* (Netzelmann et al., 2016).

#### 1.4 Inteligencia artificial

La Inteligencia Artificial (IA) es considerada como la combinación o conjunto de algoritmos utilizados para la creación de máquinas que simulen las capacidades del cerebro humano; este término ha incrementado su popularidad en los últimos años debido a las enormes cantidades de datos, algoritmos más especializados y avanzados, además de las continuas mejoras tecnológicas y computacionales, sin embargo, fue adoptado en el año 1956 (Flasiński & Flasiński, 2016).

Algunos expertos como Peter Norvig y Stuart Russell, señalan que existen distintos tipos de IA, los cuales se indican a continuación (Huang & Rust, 2018):

- Sistemas que piensan como humanos, lo cuales buscan automatizar una serie de actividades, entre las que se encuentran resolución de problemas, toma de decisiones o el aprendizaje. Ejemplo: Redes neuronales artificiales.
- Sistemas que actúan como humanos, es decir, computadoras que buscan realizar acciones como si lo hiciera un ser humano. Ejemplo: robots.
- Sistemas que piensan racionalmente; que tienen como objetivo emular el pensamiento de los humanos, incluyendo la lógica racional. Ejemplo: sistemas expertos que buscan razonar, percibir o actuar como las personas.
- Sistemas que actúan racionalmente; son los sistemas que tienen como meta, imitar de manera racional el comportamiento de los seres humanos. Ejemplo: agentes inteligentes.

La IA está presente en gran parte de nuestros quehaceres diarios, como en las redes sociales, teléfonos celulares, etc., lo cual viene impulsado por el denominado Big Data, con lo que, si se trabaja de manera adecuada, se pueden lograr incrementos considerables en la calidad de los sistemas de inteligencia artificial. Algunas aplicaciones de la IA son:

- Asistentes personales como los chatbots que interactúan con los seres humanos a través de la sugerencia de servicios, productos, etc., lo cual viene dado en base a los historiales de búsqueda de cada persona.
- Temas financieros, como la búsqueda de patrones, predicciones en la detección de fraude, ayuda personalizada a clientes, etc.
- En la educación, mediante la sugerencia de cursos, creación de ofertas personalizadas, optimización de aprendizaje en niños o personas con problemas físicos o mentales, etc.
- En el ámbito comercial, por ejemplo, a través de pronósticos o predicciones de ventas.
- En temas climáticos, con la ayuda de drones que sean entrenados para sembrar árboles con la finalidad de combatir o erradicar la deforestación, detección de fugas

en barcos que cargues combustibles, reducción del consumo energético con edificios inteligentes, etc.

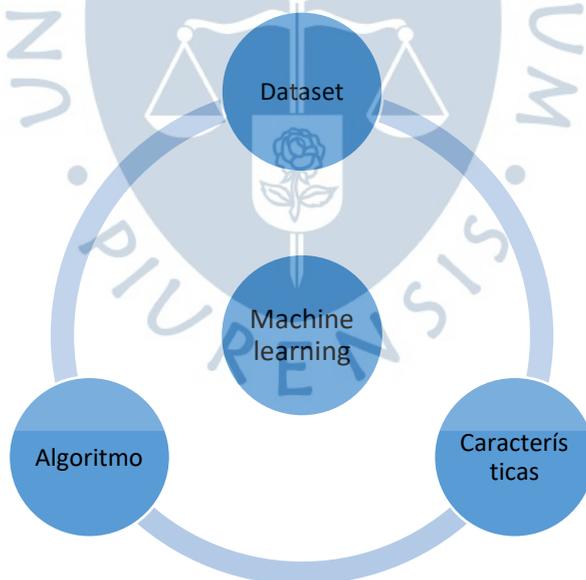
- En el ámbito agrícola, a través de diversas plataformas que permitan realizar análisis predictivos con la finalidad de incrementar la producción agrícola, advertir o pronosticar la presencia de plagas o enfermedades, etc.
- En temas de transporte o aspectos logísticos, evitando atascos, colisiones, reducción del tráfico en horas punta, vehículos inteligentes, etc.
- En el sector salud, como sistema de apoyo al diagnóstico médico mediante la clasificación de imágenes, síntomas, etc.
- Entre muchos otros.

#### 1.4.1 Machine Learning

El aprendizaje automático o machine learning (ML), es un subconjunto o división de la inteligencia artificial, centrado en que las computadoras aprendan sin la necesidad de ser programadas para la realización de determinadas tareas, es decir, busca que los algoritmos aprendan en base a algunos set de datos y puedan realizar predicciones (Zenisek et al., 2019).

El ML empezó a cobrar importancia en la década de los 80, incluyendo tres componentes fundamentales (Carvalho et al., 2019), tal como se aprecia en la **Figura 14**.

**Figura 14.** Componentes del Machine Learning.



Nota. Elaboración propia a partir de “*A systematic literature review of machine learning methods applied to predictive maintenance*” (Carvalho et al., 2019),

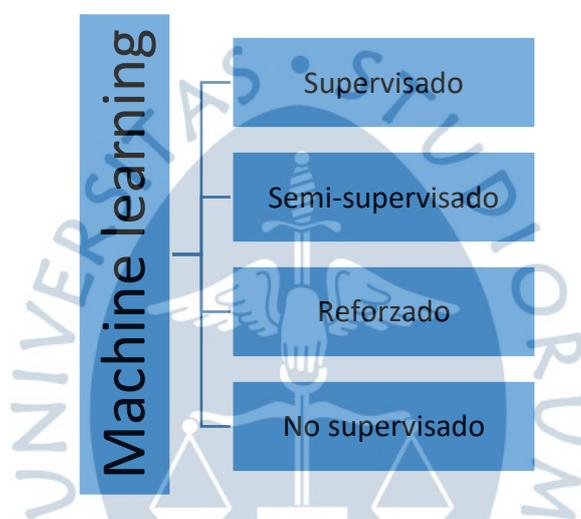
- Conjunto de datos (set de datos): representan los datos de entrada al modelo, pudiendo ser imágenes, video, texto, números, etc.
- Características: representan a los datos o características relevantes que el modelo debe utilizar o prestar atención para solucionar la tarea especificada.

- Algoritmo: Es el centro del modelo, por lo que existe gran variedad de los mismos, pudiendo resolver una misma tarea mediante el uso de distintos algoritmos, ya que presentan precisiones o velocidades diferentes y acorde al caso de aplicación.

En base a estos elementos, se puede dar solución al mismo problema incorporando distintos algoritmos, y dependiendo de estos, variará la precisión y/o velocidad en obtener un resultado, los cuales pueden ser diferentes.

En función del mecanismo o forma de aprendizaje, los algoritmos de machine learning se dividen en cuatro grupos, los cuales son: aprendizaje supervisado, semi-supervisado, reforzado y no supervisado, tal como se muestra en la ver **Figura 15** (Fatima & Pasha, 2017).

**Figura 15.** Algoritmos de Machine Learning.



Nota. Elaboración propia a partir de "Survey of Machine Learning Algorithms for Disease Diagnostic" (Fatima & Pasha, 2017).

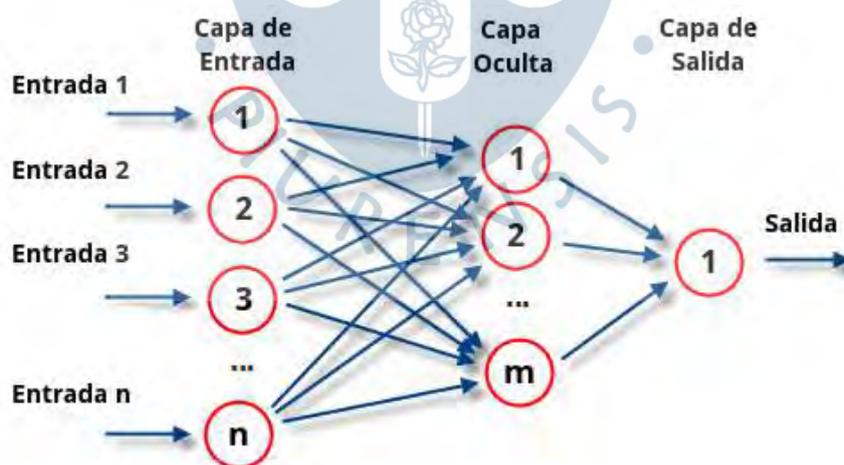
- Aprendizaje supervisado: Involucra un conjunto de datos de entrenamiento previamente etiquetados por un especialista, utilizado generalmente en clasificación y regresión para detección de idioma, filtrado de spam, visión por computadora, etc. Algoritmos utilizados: Máquinas de vectores soporte, Vecinos K-más cercanos, Regresión logística, Ingenuo Bayes, Árbol de decisión, Regresiones lineales y polinomiales.
- Aprendizaje semi-supervisado: Los datos de entrenamiento contienen una combinación de ejemplos etiquetados y no etiquetados, es decir que el programador del sistema conoce los patrones en los datos, sin embargo, el modelo debe ser capaz de encontrarlos por sí mismo.
- Aprendizaje reforzado: Mecanismo similar a la forma de aprender de los seres humanos, es decir mediante la prueba, en el que no se requiere una supervisión constante. Algunos ejemplos son los videojuegos, ya que incluyen entornos con enormes cantidades de datos, en los que los puntajes son un aspecto de recompensa en el entrenamiento.

- Aprendizaje no supervisado: Los datos de entrenamiento no están etiquetados, por lo que el sistema debe buscar patrones mediante la agrupación por características similares. Algoritmos utilizados: agrupamiento K-means, Descomposición de valor singular (SVD), cambio medio, análisis de componentes principales (PCA), FP-crecimiento, DBSCAN, asignación de Dirichlet latente (LDA), o el análisis semántico latente.

#### 1.4.2 Deep Learning

El Deep learning o aprendizaje profundo, es una clase o tipo de algoritmos pertenecientes al Machine Learning, por lo que se basan en el estudio de la estructura del cerebro humano. Estos algoritmos se basan en redes neuronales artificiales (ANN) complejas de varias capas, en las cuales el grado de abstracción incrementa de manera gradual acorde a las transformaciones no lineales aplicadas al set de datos de entrada; para lograr este cometido, las redes neuronales van enviando la información hacia las demás capas mediante los canales de conexión establecidos, en los que cada uno tiene un valor asociado. Las neuronas tienen asignado un único valor numérico denominado bias o sesgo, el cual se agrega a la suma ponderada de las entradas que llegan a la neurona en la que se debe aplicar una determinada función de activación. El resultado de esta función establece si la neurona se activa o no; este proceso se repite hasta llegar a la última capa, en la que se generan las salidas del modelo (Wan et al., 2020). En la **Figura 16** se aprecia la estructura de una red neuronal artificial.

**Figura 16.** Red neuronal artificial.



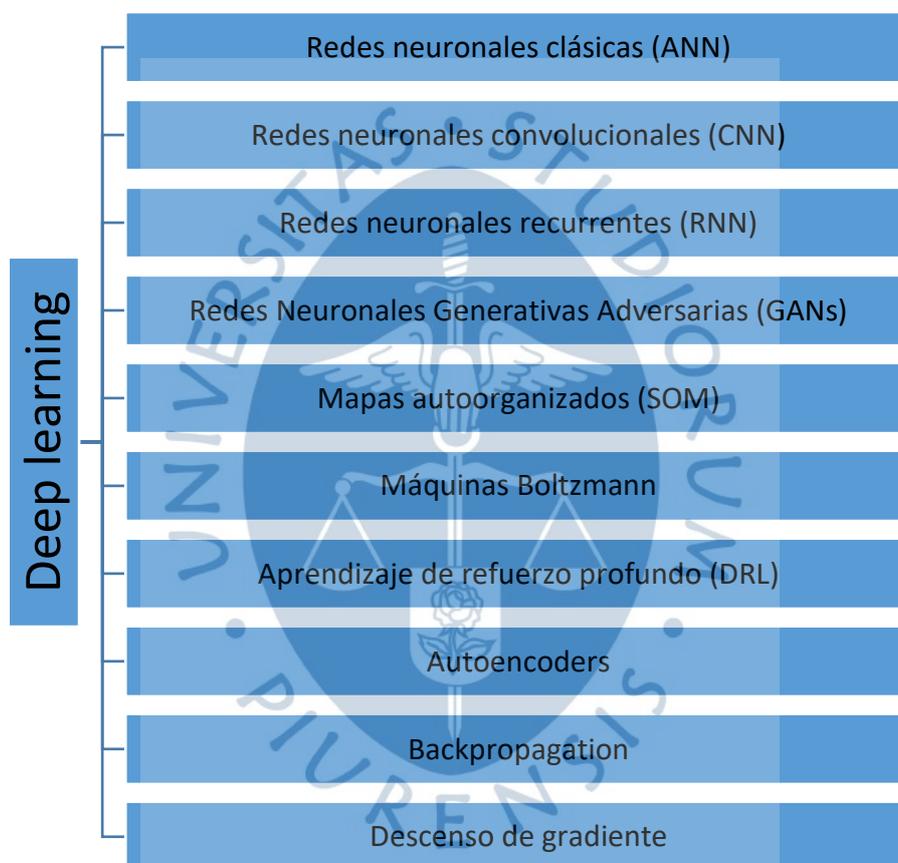
Nota. Elaboración propia a partir de “*Qué son las redes neuronales y sus funciones*” (ATRIA Innovation, 2019).

Las redes neuronales artificiales utilizan algunas técnicas de aprendizaje automático para resolver problemas del mundo real, recurriendo a redes neuronales que simulan la toma de decisiones humanas. Para entrenar las redes neuronales se requieren cantidades masivas de datos de entrenamiento, ya que se debe tener en cuenta una gran cantidad de parámetros

para que la solución sea precisa. Los algoritmos de aprendizaje profundo son bastante populares en la actualidad, sin embargo, en realidad no hay un umbral bien definido entre algoritmos profundos y no tan profundos (Golodov et al., 2017).

Algunas de las aplicaciones del Deep learning son la clasificación de imágenes, monitoreo de condiciones climáticas, mejor comprensión y apoyo en diagnóstico de enfermedades, detección de distintos tipos de fraudes, análisis de textos, etc. (Briese et al., 2020). Las técnicas de DL más conocidas son las siguientes (Garcia-Garcia et al., 2018) (Brunetti et al., 2018), las cuales se aprecian en la **Figura 17**:

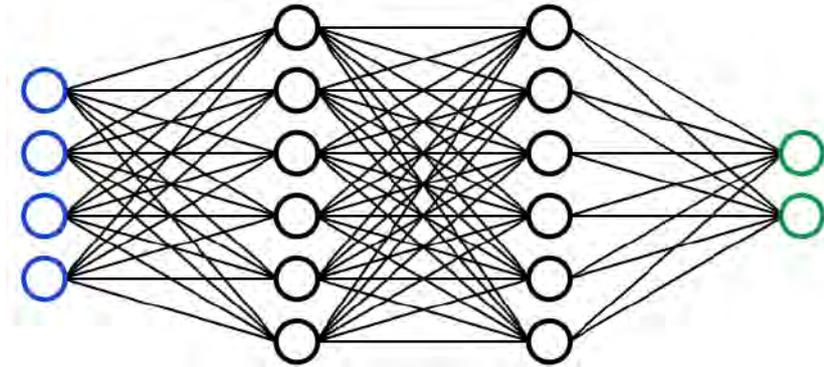
**Figura 17.** Técnicas de Deep Learning.



Nota. Elaboración propia a partir de "Computer vision and deep learning techniques for pedestrian detection and tracking: A survey" (Brunetti et al., 2018).

- Redes neuronales clásicas: También conocido como redes neuronales completamente conectadas, a menudo se identifica por sus perceptrones multicapa, donde las neuronas están conectadas a la capa continua (ver **Figura 18**) lo cual implica la adaptación del modelo en entradas de datos binarios fundamentales. Funciones utilizadas: Función lineal y función no lineal (Curva sigmoidea, tangente hiperbólica - tanh, Unidad lineal rectificadora - ReLU).

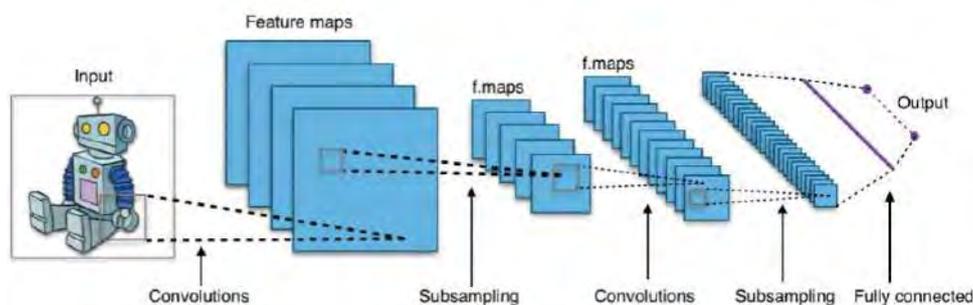
**Figura 18.** Red neuronal clásica.



Tomado de “*Neural Networks From Scratch*” (Agrahari, 2020).

- Redes neuronales convolucionales: Las CNN son un tipo sofisticado de una red neuronal artificial, elaboradas para solucionar problemas con altos grados de complejidad. Debido a esto son consideradas como uno de los tipos más eficientes en entrenamiento de sets de datos de imágenes (ver **Figura 19**). Una red neuronal convolucional presenta cuatro organizaciones diferentes:
  - o Se compone de una sola capa de entrada, que generalmente es una disposición bidimensional de neuronas para analizar datos de imágenes primarias, que es similar a la de los píxeles de las fotos.
  - o Algunas CNN consisten en una capa de neuronas de salida unidimensional que procesa imágenes en sus entradas, a través de las capas convolucionales conectadas dispersas.
  - o Las CNN también tienen la presencia de una tercera capa conocida como capa de muestreo para limitar el número de neuronas involucradas en las capas de red correspondientes.
  - o En general, las CNN tienen capas conectadas simples o múltiples que conectan el muestreo a las capas de salida.

**Figura 19.** Convolutional Neural Networks.

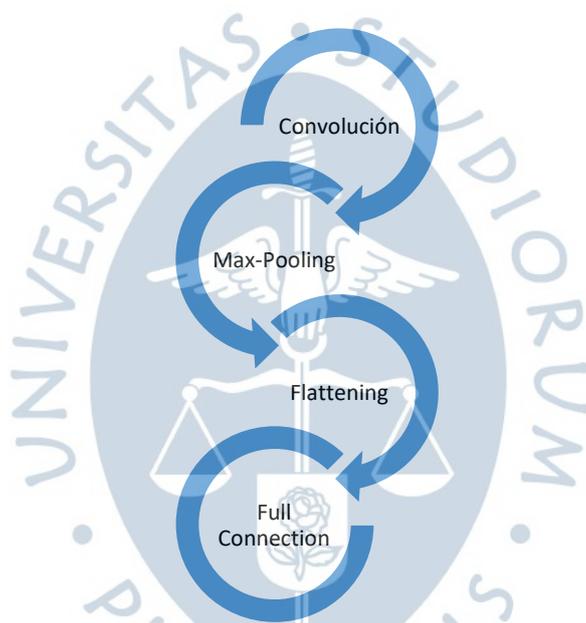


Tomado de “*Introducing Convolutional Neural Networks in Deep Learning*” (Kone, 2019).

Existen cuatro etapas involucradas en la construcción de la CNN como se aprecia en la **Figura 20**.

- Convolución: el proceso deriva mapas de características de los datos de entrada, seguidos de una función aplicada a estos mapas.
- Max-Pooling: ayuda a CNN a detectar una imagen en función de las modificaciones dadas.
- Aplanamiento: en esta etapa, los datos generados se aplanan para que un CNN los analice.
- Conexión completa: a menudo se describe como una capa oculta que compila la función de pérdida para un modelo.

**Figura 20.** Etapas de la construcción de una CNN.

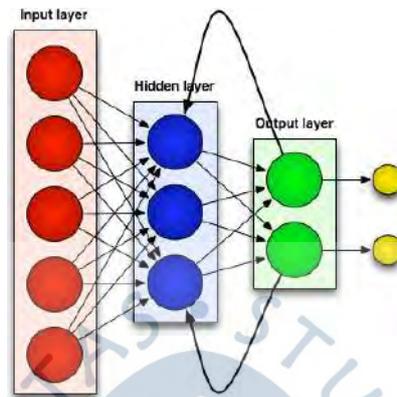


Nota. Elaboración propia a partir de *“Introducing Convolutional Neural Networks in Deep Learning”* (Kone, 2019).

- Redes neuronales recurrentes (RNN): Los RNN se diseñaron por primera vez para ayudar a predecir secuencias. Dichas redes funcionan completamente en secuencias de datos de longitud de entrada variable. RNN pone el conocimiento obtenido de su estado anterior como un valor de entrada para la predicción actual, por lo tanto, puede ayudar a lograr una memoria a corto plazo en una red, lo que lleva a la gestión efectiva de cambios en el precio de acciones u otros sistemas de datos basados en el tiempo. Existen dos tipos generales de diseños RNN que ayudan a analizar los problemas, estos son:
  - LSTM: Útil en la predicción de datos en secuencias de tiempo, usando memoria (ver **Figura 21**). Tiene tres puertas: Entrada, Salida y Olvido.

- RNNs cerrados: también son útiles en la predicción de datos de secuencias de tiempo a través de la memoria. Tiene dos puertas: Actualizar y Restablecer.

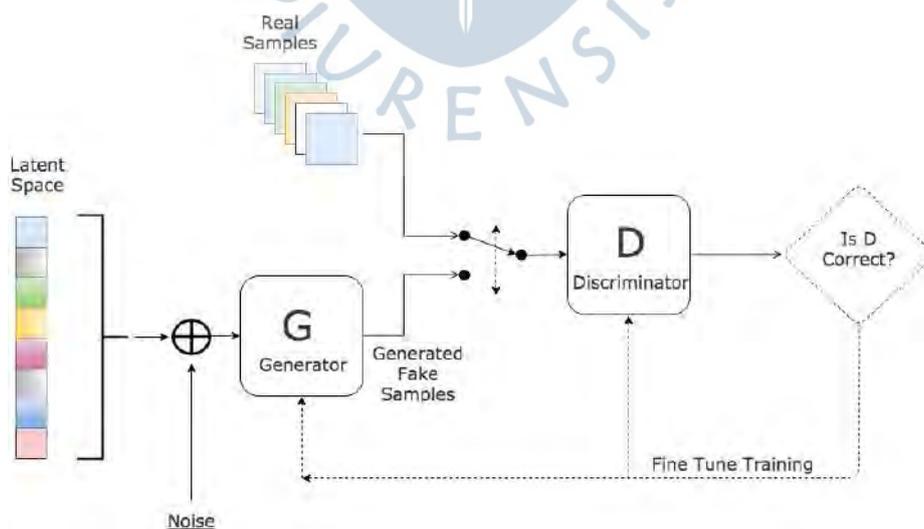
**Figura 21.** *Recurrent Neural Networks.*



Tomado de “*Understanding Recurrent Neural Networks: The Preferred Neural Network for Time-Series Data*” (Roell, 2017).

- Redes Neuronales Generativas Adversarias (GANs): Son una mezcla de 2 técnicas de aprendizaje profundo, integrando un generador y un discriminador. De esta manera, la red neuronal produce generadores, y el discriminador realiza el entendimiento de los datos correctos e incorrectos para poder tomar una decisión. Ambas redes son competitivas, ya que el Generador sigue produciendo datos artificiales idénticos a los datos reales, y el Discriminador detecta continuamente datos reales e irreales. En la **Figura 22** se observa una GAN.

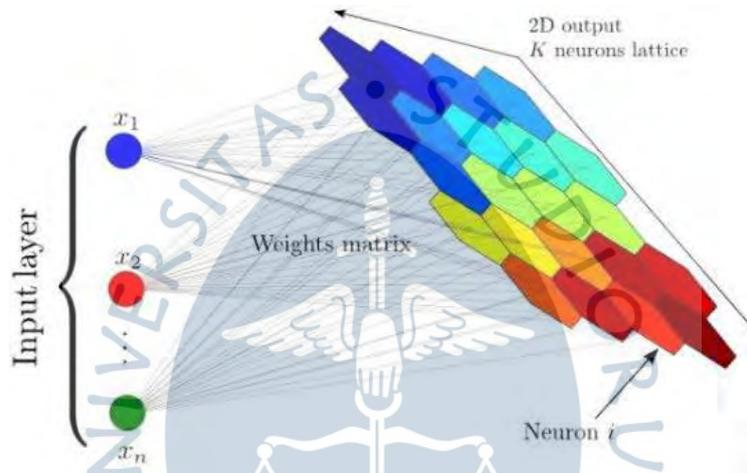
**Figura 22.** *Generative Adversarial Networks.*



Tomado de “*Comparative Study on Generative Adversarial Networks*” (Hitawala & Cheriton, 2018).

- Mapas auto-organizados (SOM): Operan con la ayuda de datos no supervisados que reducen el número de variables aleatorias en un modelo; la dimensión de salida se establece como un modelo bidimensional, debido a que cada proceso de sinapsis se conecta con sus nodos de entrada, así como también a los de salida. A medida que cada punto de datos compite por la representación de su modelo, el SOM actualiza el peso de los nodos más cercanos o las mejores unidades de coincidencia (BMU). Según la proximidad de una UMB, el valor de los pesos cambia; como los pesos se consideran como una característica del nodo en sí misma, el valor representa la ubicación del nodo en la red. En la **Figura 23** se apreciar un ejemplo de un mapa auto-organizado.

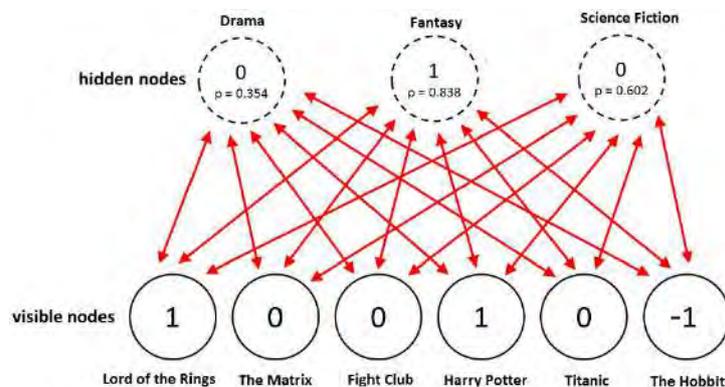
**Figura 23.** *Self-Organizing Maps.*



Tomado de “*Self Organizing Maps (SOM) in NetLogo*” (Sancho Caparrini, 2021).

- Máquinas Boltzmann: Este modelo de red no viene con ninguna dirección predefinida y, por lo tanto, tiene sus nodos conectados en una disposición circular; es por ello que, esta técnica de aprendizaje profundo se utiliza para producir parámetros modelo (ver **Figura 24**). A diferencia de todos los modelos de red deterministas anteriores, el modelo de máquinas de Boltzmann se conoce como estocástico.

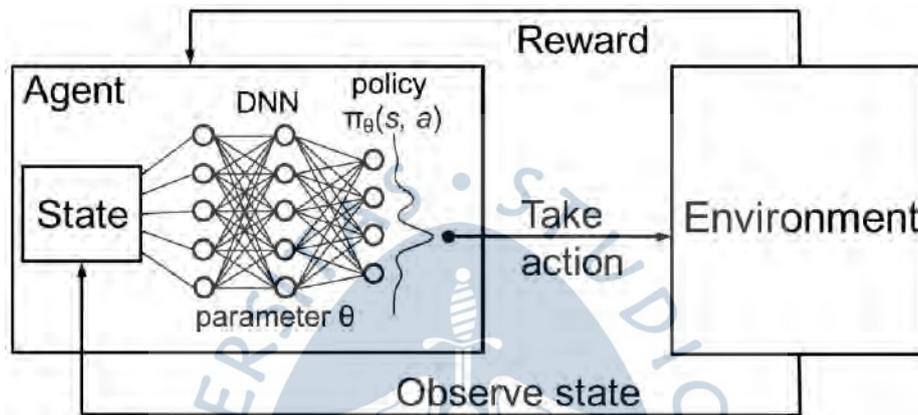
**Figura 24.** *Boltzmann Machines.*



Tomado de “*Deep Learning meets Physics: Restricted Boltzmann Machines*” (Oppermann, 2018).

- Aprendizaje de refuerzo profundo (DRL): El aprendizaje de refuerzo se refiere al proceso en el que un agente interactúa con un entorno para modificar su estado; el agente puede observar y tomar acciones, por lo que, ayuda a una red a alcanzar su objetivo al interactuar con la situación. En el DRL, existe una capa de entrada, diversas capas ocultas y una capa de salida, donde el estado del entorno es la capa de entrada en sí. El modelo funciona en los intentos continuos de predecir la recompensa futura de cada acción tomada en el estado dado de la situación (ver **Figura 25**).

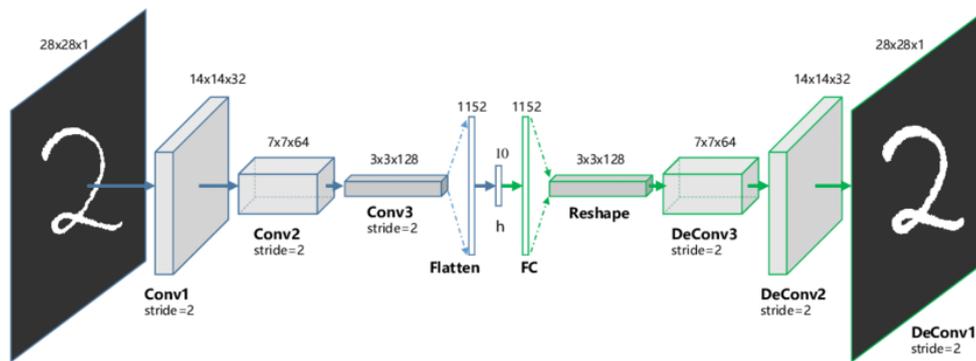
**Figura 25.** *Deep Reinforcement Learning.*



Tomado de "Deep Reinforcement Learning: Value Functions, DQN, Actor-Critic method, Back-propagation through stochastic functions" (Vijayan, 2020).

- Autoencoders: Es uno de los tipos más utilizados de técnicas de aprendizaje profundo; este modelo funciona automáticamente en función de sus entradas, antes de tomar una función de activación y decodificación de salida final. Tal formación de cuellos de botella conduce a producir categorías menores de datos y a aprovechar la mayoría de las estructuras de datos inherentes (ver
- 
- **Figura 26**). Los tipos de autoencoders son:
  - o Sparse: donde las capas ocultas superan en número a la capa de entrada para que tenga lugar el enfoque de generalización para reducir el sobreajuste. Limita la función de pérdida y evita que el autoencoder use en exceso todos sus nodos.
  - o Denoising: una versión modificada de las entradas se transforma en 0 al azar.
  - o Contractive: adición de un factor de penalización a la función de pérdida para limitar el sobreajuste y la copia de datos, en caso de que la capa oculta supere a la capa de entrada.
  - o Stacked: Para un autoencoder, una vez otra capa oculta se agrega, que conduce a dos etapas de codificar a la de una fase de decodificación.

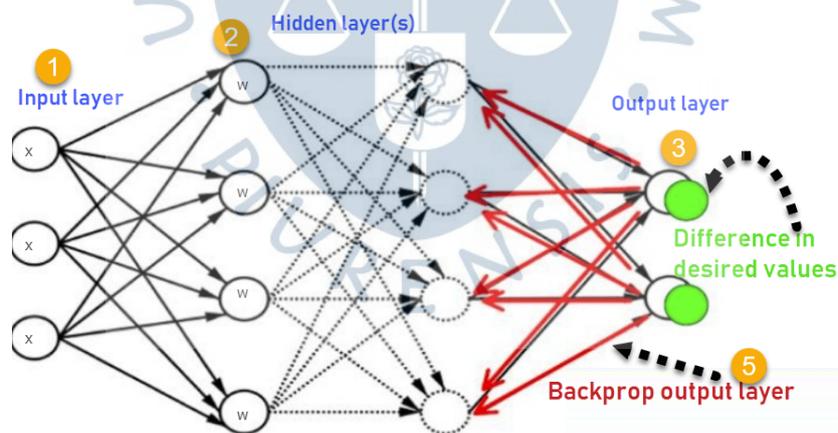
Figura 26. Autoencoders.



Tomado de “Deep Clustering with Convolutional Autoencoders” (Guo et al., 2017).

- Backpropagation: La propagación, por otro lado, se refiere a la transmisión de datos en una dirección dada a través de un canal dedicado, por lo que el Backpropagation es el mecanismo central para que las redes neuronales aprendan sobre cualquier error en la predicción de datos. Todo el sistema puede funcionar de acuerdo con la propagación de la señal en la dirección de avance en el momento de la decisión, y envía de regreso cualquier dato relacionado con las deficiencias en la red. Primero, la red analiza los parámetros y decide los datos, en segundo lugar, se compensa con una función de pérdida, y en tercer lugar, el error identificado se vuelve a propagar para autoajustar los parámetros incorrectos, tal como se aprecia en la **Figura 27**.

Figura 27. Backpropagation.

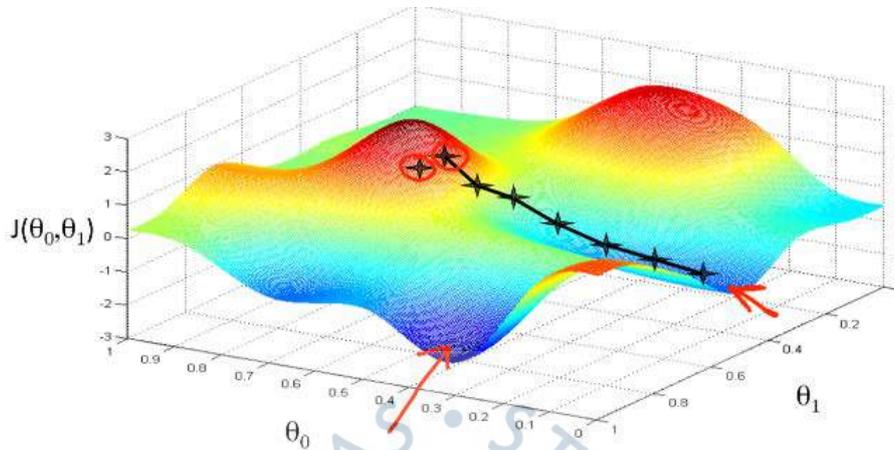


Tomado de “Back Propagation Neural Network: What is Backpropagation Algorithm in Machine Learning?” (Johnson, 2022).

- Descenso de gradiente: En el contexto matemático, gradiente se refiere a una pendiente que tiene un ángulo medible y puede representarse en una relación entre variables. En esta técnica de aprendizaje profundo, la relación entre el error producido en la red neuronal con el de los parámetros de datos se puede representar como "x" e "y". Como las variables son dinámicas en una red neuronal, el error puede

aumentarse o disminuirse con pequeños cambios, por lo que, el objetivo de este método es encontrar la solución óptima (ver **Figura 28**).

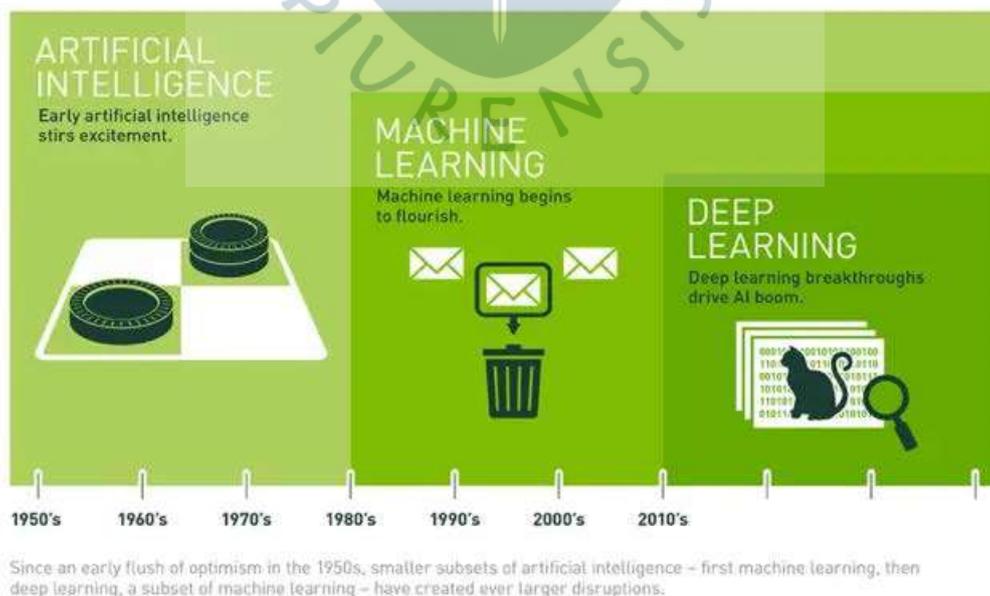
**Figura 28.** *Gradient Descent,*



Tomado de “*Gradient Descent: All You Need to Know*” (Suryansh, 2018).

Entonces, la Inteligencia Artificial (IA) se define como una máquina que intenta imitar las funciones cognitivas de los humanos, Machine Learning (ML) es una rama de IA, la cual intenta generalizar comportamientos mediante el análisis de conjuntos de datos, y Deep Learning o DL forma parte del Machine Learning, este busca que un sistema informático imite o se asemeje al raciocinio y comportamiento de los seres humanos de manera automática, centrándose en el uso de ANN o redes neuronales artificiales, las cuales representan el funcionamiento de las redes neuronales humanas mediante el proceso de sinapsis (Woschank et al., 2020). Estos 3 conceptos se muestran en la **Figura 29**.

**Figura 29.** *Inteligencia artificial, Machine Learning y Deep Learning.*



Tomado de “*Diferencia entre AI, Machine Learning y Deep Learning*” (Copeland, 2021).

## 1.5 Investigaciones relevantes

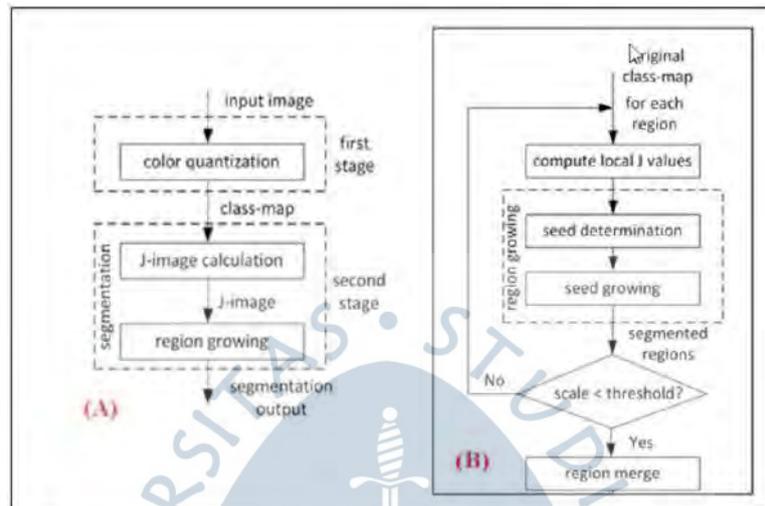
Diversas investigaciones demuestran que el estudio de técnicas de inteligencia artificial es aplicable en algunos sectores como el eléctrico, tal como se demostró en (Fambrini et al., 2018), proponiéndose un sistema de reconocimiento automático basado en Inteligencia Artificial para imágenes térmicas de redes de distribución eléctrica; se indica que la termografía infrarroja se usa generalmente para realizar inspecciones en líneas de distribución de energía eléctrica, con la asistencia de un operador humano, que generalmente es responsable de operar todo el equipo, seleccionar los puntos más calientes de la imagen (correspondientes a los lugares que necesitan mantenimiento), hacer informes y coordinar con el equipo técnico, que hará las reparaciones. El sistema de diagnóstico automático propuesto tuvo como objetivo reemplazar la operación de inspección manual utilizando algoritmos de procesamiento de imágenes, debido a esto, se entrenó y probó un antiguo método de segmentación para imágenes térmicas conocido como JSEG y una red neuronal de aprendizaje profundo, la cual es responsable de reconocer los elementos segmentados.

El algoritmo implementado, probado y descrito en este trabajo, se conoce como JSEG el cual significa segmentación J, porque produce imágenes J, una especie de imagen transformada. La idea principal de JSEG fue dividir el proceso de segmentación en dos etapas procesadas de manera independiente: cuantización de color y segmentación espacial. La segmentación no supervisada de color y textura consiste en considerar la disposición espacial de los píxeles utilizando una técnica de crecimiento regional basada en operadores morfológicos, por lo que se define un modo de homogeneidad con píxeles agrupados en la región segmentada. Por otro lado, debe considerarse diferentes escalas de imágenes. El algoritmo JSEG segmenta las imágenes en pseudocolor producido por las cámaras de visión térmica, sin ajustar los parámetros manuales para cada imagen y simplifica la textura y el color. La segmentación con este algoritmo pasa por tres sub-etapas, que se denominan: cuantificación del espacio de color, tasa de aciertos en regiones y combinación de regiones con colores similares. En la **Figura 30** se aprecia los pasos que sigue el algoritmo JSEG y la segmentación espacial.

La reducción de la duración es uno de los principales problemas del algoritmo JSEG. Para resolver este problema, se seleccionó una implementación de GPU CUDA JSEG modificada, basada en (Zimmerman & Zumbo, 2010). El resultado fue un algoritmo rápido y robusto. El tiempo de segmentación fue de 73 ms (tamaño de cuadrícula 9x9), utilizando la implementación de la placa de video NVIDIA GeForce GTX1080 GPU y 970 ms ejecutándose en la CPU para el mismo tamaño de cuadrícula. El tiempo de segmentación de JSEG en una cuadrícula de 65x65 fue de 2.456 ms en promedio. Para reconocer los elementos sobrecalentados que fueron segmentados por la etapa JSEG anterior, se implementó un clasificador de tipo supervisado basado en una red convolucional de aprendizaje profundo. Este clasificador también funciona en la GPU y fue desarrollado en lenguaje C y CUDA, y está

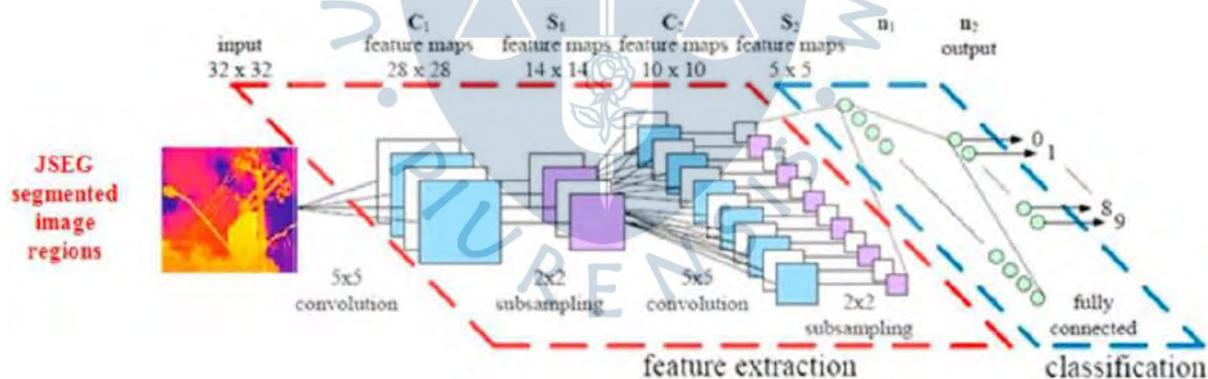
basado en el algoritmo YOLO, tal como se aprecia en la **Figura 31** (Dragomir et al., 2017; Lile & Yiqun, 2018). La entrada de este clasificador es la imagen segmentada del algoritmo JSEG y la salida es una región en el cuadro y una etiqueta que representa la clase de la imagen clasificada.

**Figura 30.** (A) Algoritmo JSEG; (B) Algoritmo de segmentación espacial.



Tomado de “GPU Cuda JsEG segmentation Algorithm associated with Deep Learning Classifier for Electrical Network Images Identification” (Fambrini et al., 2018).

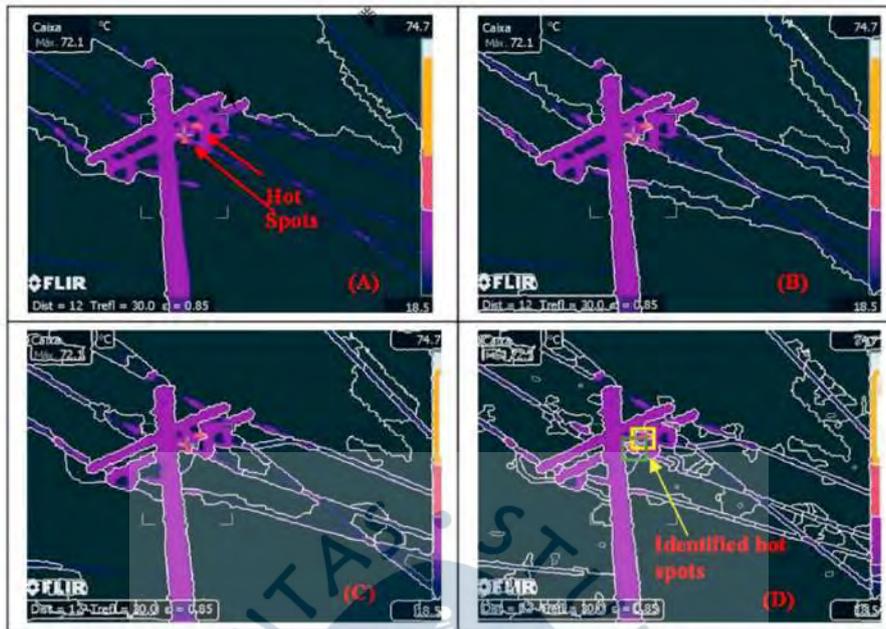
**Figura 31.** Arquitectura neuronal convolucional YOLO, basada en Alexnet.



Tomado de “Relative power of the wilcoxon test, the friedman test, and repeated-measures ANOVA on ranks” (Zimmerman & Zumbo, 2010).

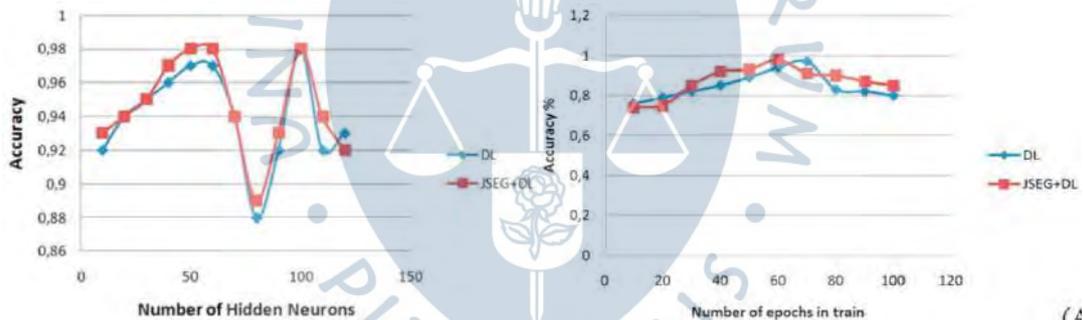
Los resultados obtenidos muestran que el sistema propuesto logra detectar puntos calientes en las imágenes térmicas, además de obtener una precisión mayor al 90% en el entrenamiento de la red neuronal mediante el uso de Deep Learning (ver **Figura 32** y **Figura 33**).

**Figura 32.** Puntos calientes detectados después de la segmentación JSEG.



Tomado de "Relative power of the wilcoxon test, the friedman test, and repeated-measures ANOVA on ranks" (Zimmerman & Zumbo, 2010).

**Figura 33.** Resultados del entrenamiento – Accuracy.

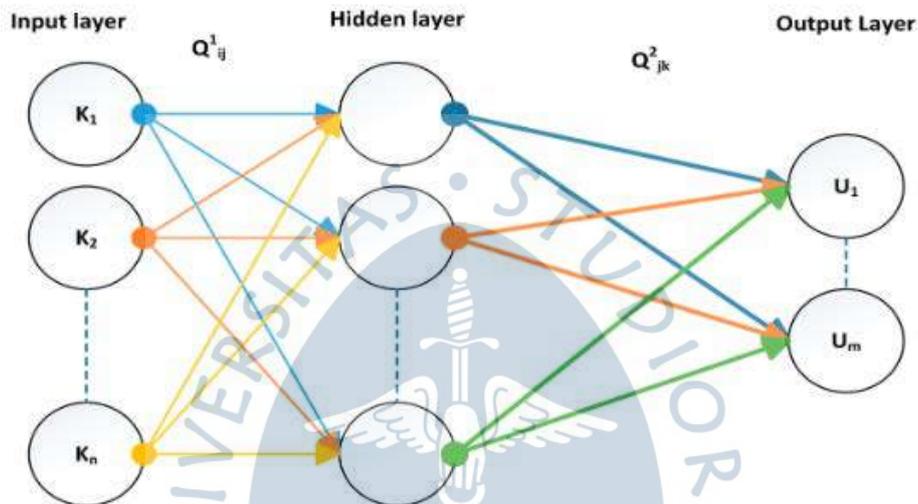


Tomado de "Relative power of the wilcoxon test, the friedman test, and repeated-measures ANOVA on ranks" (Zimmerman & Zumbo, 2010).

El Departamento de Tecnología de Chongqing, China (Ullah et al., 2017) realizó una investigación en la que se indica que las perturbaciones y daños en los equipos son causadas principalmente por grietas de aislamiento, uniones de terminales, cargas irregulares, relés defectuosos, entre otros, los cuales se evidencian mediante el incremento de la temperatura interna de los equipos y/o instrumentos eléctricos, resultando en perturbaciones inesperadas y daños potenciales a los equipos de energía. Por lo tanto, las medidas de prevención iniciales de anomalías térmicas en las herramientas eléctricas son esenciales para evitar fallas en los equipos de potencia. En base a estas evidencias, incluyen mecanismos de prevención de sus subestaciones eléctricas apoyados en técnicas de visión por computador a través del estudio y análisis de imágenes térmicas, con lo cual no se hace necesaria la detención de la planta ya que estas imágenes pueden captarse inclusive cuando la maquinaria se encuentra en

funcionamiento. Se utilizaron un total de 150 imágenes térmicas de diferentes equipos eléctricos en 10 subestaciones diferentes en condiciones de funcionamiento, utilizando 300 puntos de acceso diferentes. Para ello se utiliza el perceptrón multicapa (MLP) con la finalidad de clasificar las condiciones térmicas de los componentes de las subestaciones de energía en dos clases: defectuosas y sin defectos, además se calcula un total de once características, que son datos estadísticos de primer y de segundo orden, a partir de las imágenes de muestra térmica. La **Figura 34** muestra la estructura estándar de un MLP.

**Figura 34.** Modelo estándar de MLP.

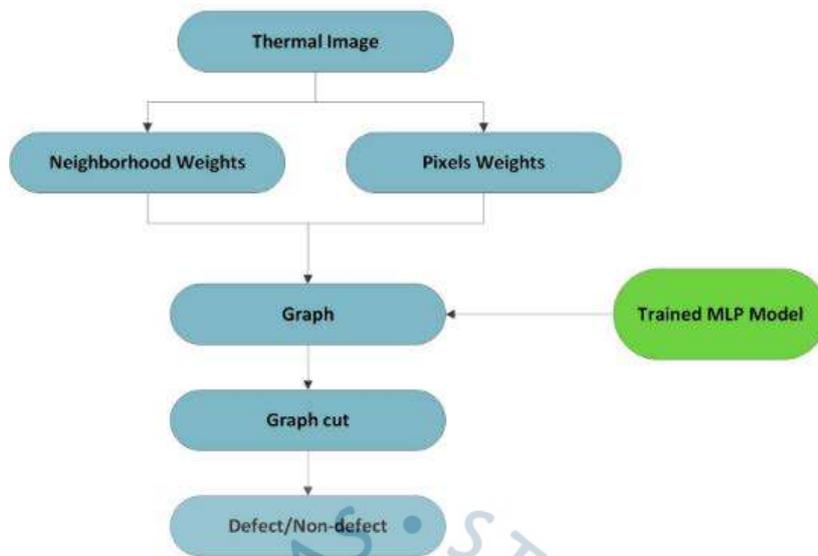


Tomado de “*Predictive Maintenance of Power Substation Equipment by Infrared Thermography Using a Machine-Learning Approach*” (Ullah et al., 2017).

Un gráfico es una combinación de nodos y aristas (arcos) que representan un fenómeno. El corte de gráfico es una técnica para segmentar un gráfico, basado en los pesos de los nodos. Los cortes de gráficos tienen muchas aplicaciones en diferentes campos, por lo que en la investigación se integraron gráficos y cortes de gráficos en un modelo entrenado por MLP para aumentar el rendimiento del clasificador, o visto desde otras palabras, para ayudar al clasificador a aumentar el rendimiento de la clasificación. Los cortes de gráficos son conocidos por minimizar las energías en el problema de la clase binaria. Dado que el problema en la configuración experimental es defectuoso y sin defecto, se integraron cortes de gráficos en la configuración de MLP para aumentar el rendimiento de clasificación, tal como se muestra en el diagrama de la **Figura 35**.

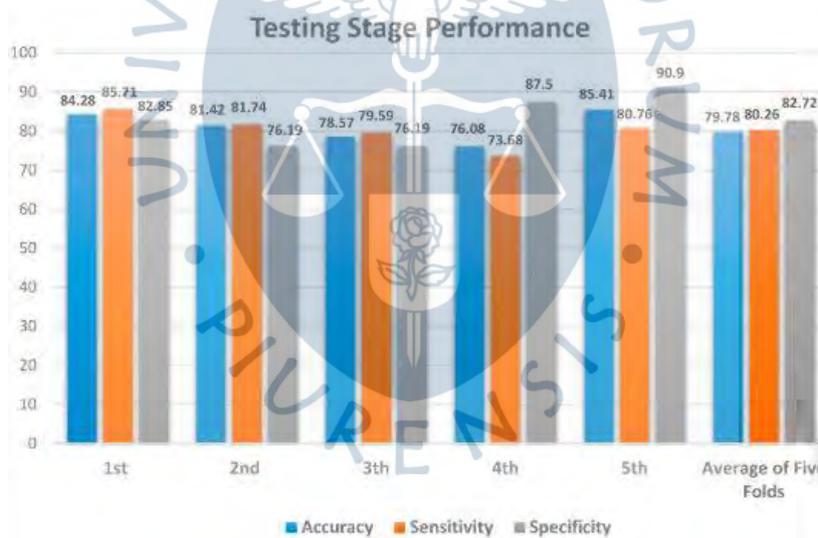
El rendimiento de MLP mostró una precisión inicial de 79.78%, aumentando este valor a un 84% con corte de gráfico (**Figura 36**, **Figura 37**, y **Figura 38**). Se indica además que, con los resultados obtenidos, el Departamento de Tecnología de Chongqing puede organizar las acciones recomendadas y así ahorrar costos en reparaciones e interrupciones, lo cual puede desempeñar un papel importante en la inspección rápida y confiable para evitar la falla del equipo de la subestación eléctrica, lo que evitará la falla de todo el sistema.

Figura 35. Diagrama para la detección de defectos / sin defectos.



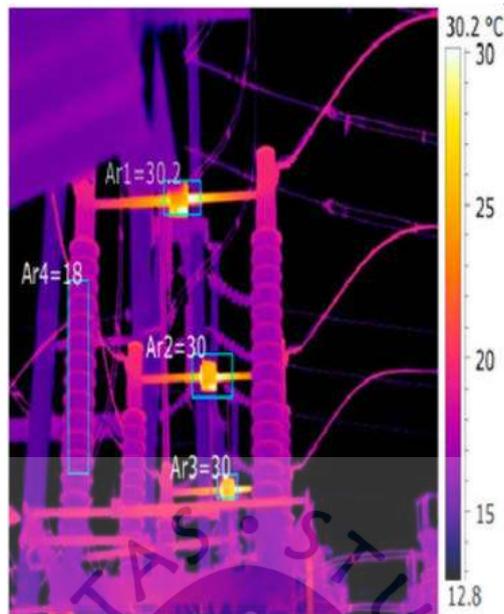
Tomado de “Predictive Maintenance of Power Substation Equipment by Infrared Thermography Using a Machine-Learning Approach” (Ullah et al., 2017).

Figura 36. Pruebas de rendimiento



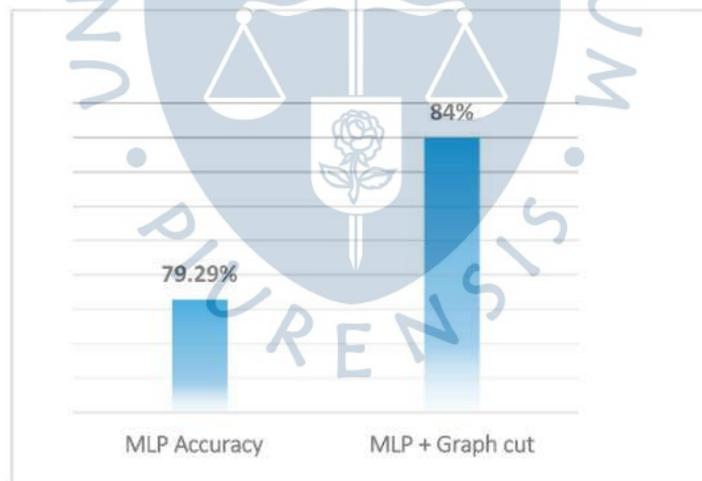
Tomado de “Predictive Maintenance of Power Substation Equipment by Infrared Thermography Using a Machine-Learning Approach” (Ullah et al., 2017).

**Figura 37.** Imagen térmica de muestra con posibles defectos esperados.



Tomado de “*Predictive Maintenance of Power Substation Equipment by Infrared Thermography Using a Machine-Learning Approach*” (Ullah et al., 2017).

**Figura 38.** Comparación de rendimiento de MLP y MLP con corte de gráfico.



Tomado de “*Predictive Maintenance of Power Substation Equipment by Infrared Thermography Using a Machine-Learning Approach*” (Ullah et al., 2017).

Otra investigación indica que mantener la fiabilidad de la instalación eléctrica se ha convertido en parte de las prácticas de eficiencia energética en la construcción (Nazmul Huda et al., 2012). La degradación de las instalaciones eléctricas puede causar sobrecalentamiento, lo que puede provocar fallas posteriores de los equipos que potencialmente pueden resultar en posibles lesiones, cortes de energía no planificados, además de un inminente peligro de incendio. La eficiencia de un sistema eléctrico disminuye antes de la falla, por lo que se gasta

energía generando calor y causando pérdidas de energía innecesarias, por lo que se requiere una prevención temprana para evitar esta situación mediante el monitoreo de la confiabilidad de las instalaciones eléctricas a través de prácticas de auditoría energética. El artículo analiza un enfoque semiautomático con el cual busca analizar las condiciones térmicas de las instalaciones eléctricas dentro de un edificio ubicado en Malasia. Inicialmente, las regiones de interés de las imágenes se segmentan manualmente; luego, las características estadísticas del histograma de primer orden y las características de la matriz de coincidencia de nivel de grises, así como las diferencias de los parámetros de características entre las regiones activas y de referencia, se extraen de las regiones segmentadas. El análisis de componentes principales se aplica para mejorar la selección de características y en la etapa final, la condición de los equipos eléctricos se clasificó utilizando una red neuronal de perceptrón multicapa (MLP).

Considerando el análisis PCA o análisis de componentes principales, se seleccionaron 15 características como entradas de la red neuronal. Los rendimientos de las redes de perceptrón multicapa se compararon y probaron con varios algoritmos de entrenamiento. La precisión de clasificación de las redes MLP también se comparó con el clasificador de análisis discriminante y se descubrió que la red de perceptrón multicapa que utiliza el algoritmo Levenberg-Marquardt (LM) ofrece el mejor rendimiento en las pruebas realizadas, obteniendo un 78.5% de precisión (**Figura 39** y **Figura 40**). Cabe resaltar que el algoritmo LM o método de mínimos cuadrados amortiguados es incluido para solucionar problemas de mínimos cuadrados no lineales. Finalmente, sugiere considerar un sistema de diagnóstico automático con extracción de características adecuada para mejorar el rendimiento del sistema.

**Figura 39.** Rendimiento de los algoritmos en el set de datos de prueba.

Classification performance for testing data set.

Training algorithm	No. of hidden nodes	Acc <sub>overall</sub> (%)	Acc <sub>normal</sub> (%)	Acc <sub>warning</sub> (%)	Acc <sub>critical</sub> (%)
LM	9	78.5	90.7	59.7	86.8
BFG	19	76.5	81.8	72.6	74.0
RP	21	77.0	83.6	68.5	84.1
SCG	14	78.0	85.5	70.3	78.0
CGB	29	76.5	73.5	79.5	77.1
CGF	39	77.5	89.4	67.1	79.6
CGP	29	78.0	83.8	74.1	76.5
OSS	10	78.0	89.4	66.3	82.4
BR	10	76.0	75.0	75.3	78.7
GD	34	58.5	71.2	50.6	52.0
GDX	32	74.5	90.0	63.9	70.2
GDM	21	63.0	76.4	45.9	68.5

Tomado de “A semi-automatic approach for thermographic inspection of electrical installations within buildings” (Nazmul Huda et al., 2012).

En este mismo sentido, otro estudio señala que la termografía infrarroja (IRT) ha desempeñado un papel muy importante en el monitoreo e inspección de defectos térmicos

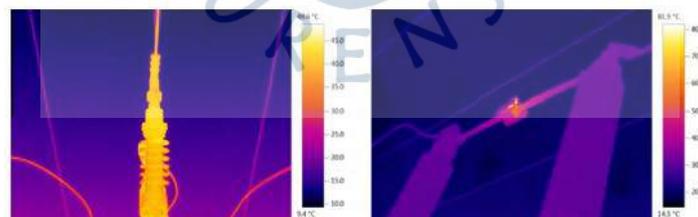
de equipos eléctricos sin la necesidad de apagarlos, lo que tiene una importancia considerable en la estabilidad de los sistemas de energía (Zou & Huang, 2015); presentando diversas ventajas, como la detección sin contacto, la ausencia de interferencias electromagnéticas, la seguridad, la fiabilidad y la gran cobertura de inspección. Indican además que el análisis manual de imágenes infrarrojas para detectar defectos y clasificar el estado del equipo puede llevar mucho tiempo y esfuerzo, y también puede dar lugar a resultados de diagnóstico incorrectos. Con la finalidad de evitar el análisis manual de imágenes infrarrojas, se proponen muchos métodos inteligentes de diagnóstico de fallas para equipos eléctricos, pero existen dos dificultades al usarlos, uno de los cuales se basa en encontrar la región de interés, y otro en extraer características que pueden representar la condición del equipo eléctrico, ya que es difícil segmentar imágenes infrarrojas debido a sus distribuciones sobre-centradas y contrastes de baja intensidad. En la **Figura 41** se muestran las imágenes originales con las que se realizaron los experimentos.

**Figura 40.** Clasificación utilizando el análisis discriminante.

Classification using discriminant analysis.				
True group				
Put into group	Critical	Warning	Normal	Overall
Critical	93	36	1	130
Warning	30	117	33	180
Normal	2	40	148	190
Total N	125	193	182	500
N correct	93	117	148	358
Accuracy (%)	74.4	60.6	81.3	71.6

Tomado de "A semi-automatic approach for thermographic inspection of electrical installations within buildings" (Nazmul Huda et al., 2012).

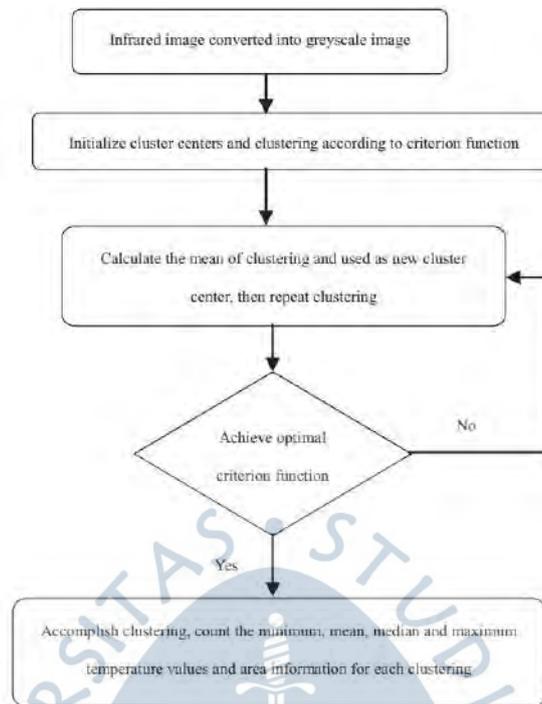
**Figura 41.** Imágenes infrarrojas originales.



Tomado de "A novel intelligent fault diagnosis method for electrical equipment using infrared thermography" (Zou & Huang, 2015).

Es por ello que proponen una metodología de diagnóstico inteligente con el objetivo de clasificar las diferentes condiciones que presentan los equipos eléctricos; en la primera etapa, se agrupa una imagen infrarroja de equipos eléctricos utilizando el algoritmo K-means, luego se extraen las características estadísticas que contienen información de temperatura y área en cada región (**Figura 42**).

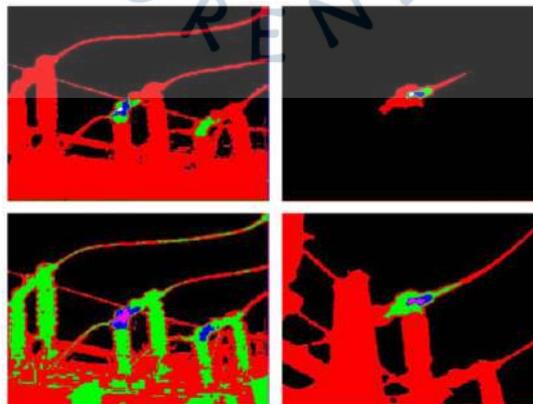
**Figura 42.** *K-means: extracción de características.*



Tomado de “*A novel intelligent fault diagnosis method for electrical equipment using infrared thermography*” (Zou & Huang, 2015).

Para seleccionar las características sobresalientes que pueden representar mejor la condición del equipo eléctrico, algunas o todas las características estadísticas de cada región, se combinan como datos de entrada para el clasificador de máquina de vectores de soporte (SVM). El rendimiento de SVM se compara con el de la red neuronal de Back Propagation (BP), y los resultados de la comparación muestran que el SVM logra un mejor rendimiento con una precisión del 97,8495%. En la **Figura 43** se aprecia los resultados del clustering.

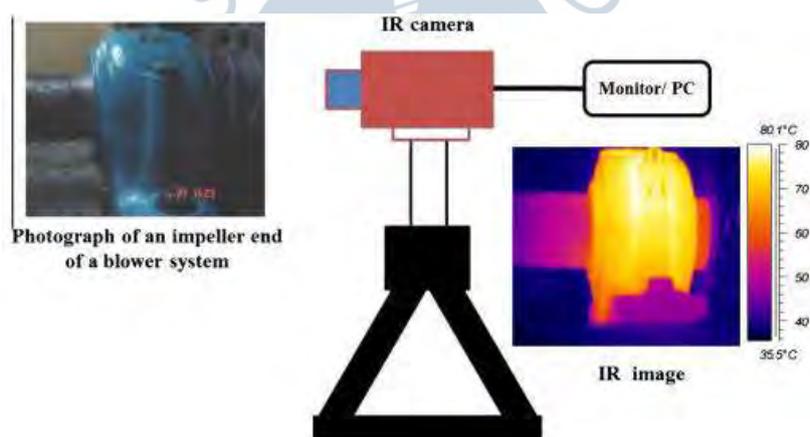
**Figura 43.** *Resultados del clustering.*



Tomado de “*A novel intelligent fault diagnosis method for electrical equipment using infrared thermography*” (Zou & Huang, 2015).

Uno de los indicadores más relevantes al momento de analizar la salud de las estructuras de componentes, equipos o materiales es la temperatura, esto debido a que las conexiones corroídas de energía, elementos dañados, entre otros, pueden generar incrementos anómalos de temperatura. La IRT o termografía infrarroja es una herramienta de monitoreo de estas condiciones, la cual es ampliamente aceptada en escenarios donde la temperatura se mide en tiempo real y sin contacto. La IRT permite la detección temprana de fallas en los equipos y procesos industriales defectuosos en condiciones de funcionamiento, lo que reduce el tiempo de inactividad del sistema, las averías catastróficas y los costos de mantenimiento. En los últimos años, se evidenció un crecimiento constante en el uso de IRT como técnica de monitoreo de condición en estructuras civiles, instalaciones eléctricas, maquinarias y equipos, alteración de materiales por diferentes condiciones de carga, problemas generados por corrosión y procesos de soldadura. La termografía también se aplica en las industrias nuclear, aeroespacial, alimentaria, papelera, maderera y plástica (Figura 44).

**Figura 44.** Configuración experimental para monitoreo de condición con IRT.

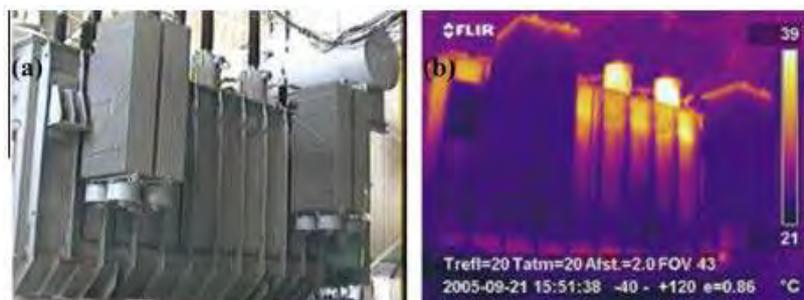


Tomado de "A novel intelligent fault diagnosis method for electrical equipment using infrared thermography" (Zou & Huang, 2015).

Por lo indicado en el párrafo anterior, los autores autores de otra investigación se centran en los avances de la IRT como una herramienta de monitoreo de condición sin contacto y no invasiva para maquinarias, equipos y procesos (Bagavathiappan et al., 2013), es por ello que se discuten en detalle varias aplicaciones de monitoreo de condiciones, junto con algunos conceptos básicos de IRT, procedimientos experimentales y técnicas de análisis de datos. Se concluye que la termografía infrarroja proporciona información pertinente sobre la salud y la eficiencia de los equipos o procesos, que son cruciales para evitar un colapso catastrófico o una parada de emergencia, además que las aplicaciones de las modernas herramientas de procesamiento de imágenes térmicas infrarrojas adquiridas junto con los enfoques basados en inteligencia artificial, pueden aumentar aún más el proceso de toma de decisiones rápidas y sin interferencia humana. A diferencia de otras metodologías de

monitoreo de condición, la termografía infrarroja proporciona una imagen del objeto codificada en pseudocolor en tiempo real y la manifestación visual de defectos (**Figura 45**).

**Figura 45.** IRT en el mantenimiento preventivo en transformadores eléctricos.



Tomado de "Infrared thermography for condition monitoring - A review," (Bagavathiappan et al., 2013).

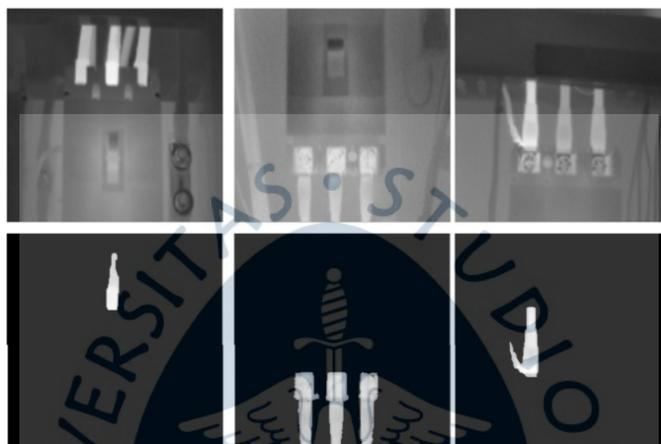
Indican además que es necesario monitorear el estado térmico de los equipos eléctricos para mantener la confiabilidad del sistema eléctrico, ya que la degradación de los equipos eléctricos puede provocar un sobrecalentamiento excesivo, lo que a su vez puede generar la eventual avería del equipo. Además, la falla del equipo requiere mucho costo de mantenimiento, mano de obra y puede causar lesiones o incluso la muerte, por lo tanto, el proceso de reconocimiento de las condiciones del equipo como normales y defectuosas es un paso esencial para mantener la confiabilidad y estabilidad del sistema.

Otro estudio (Huda & Taib, 2013) propone la monitorización del estado de los equipos eléctricos mediante termografía infrarroja, indicando que el análisis manual de la imagen térmica para detectar defectos y clasificar el estado de los equipos requiere mucho tiempo, esfuerzos y también puede conducir a resultados de diagnóstico incorrectos; un sistema inteligente que pueda separar el equipo automáticamente podría ayudar a superar estos problemas. Analiza un sistema de clasificación inteligente para las condiciones de los equipos que utilizan redes neuronales; el análisis de imagen extrae tres conjuntos de características, la estadística de primer orden basada en estadísticas, la matriz de coincidencia de nivel de grises y las características de intensidad basadas en componentes que se utilizan como datos de entrada para las redes neuronales. Las redes neuronales de perceptrón multicapa (MLP), se entrenan utilizando cuatro algoritmos de entrenamiento diferentes: Scale conjugate gradient (SCG), Bayesian Regulazation (BR), Resilient back propagation (RP) y Levenberg-Marquardt (LM). En la **Figura 46** se observan algunos ejemplos de imágenes térmicas en escala de grises antes y después de ser procesadas.

Las características de entrada para las redes MLP se evaluaron analizando el rendimiento de 3 conjuntos de características (i) características basadas en histograma de primer orden, (ii) características de matriz de coincidencia de nivel de grises (GLCM) y (iii) características de intensidad basadas en componentes. Finalmente, se seleccionaron 3 características de intensidad, por lo que la red MLP con algoritmos de entrenamiento SCG y

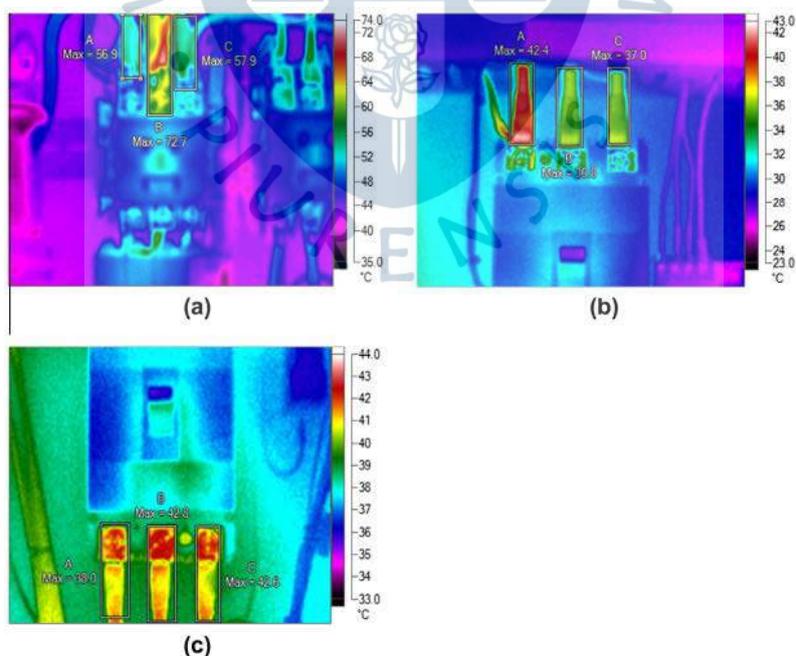
LM logró una tasa de reconocimiento máxima de 82.89% y 74.25% para la fase de entrenamiento y prueba, respectivamente. Después de seleccionar las mejores características, la red MLP con el algoritmo Levenberg-Marquardt (LM) logró los mejores resultados para clasificar las condiciones de los equipos eléctricos. Finalmente, en la etapa de análisis de rendimiento, las características de intensidad basadas en componentes produjeron los mejores resultados en comparación con los otros dos conjuntos de características (**Figura 47**).

**Figura 46.** Imágenes grises térmicas antes y después del procesamiento de imágenes.



Tomado de "Suitable features selection for monitoring thermal condition of electrical equipment using infrared thermography" (Huda & Taib, 2013).

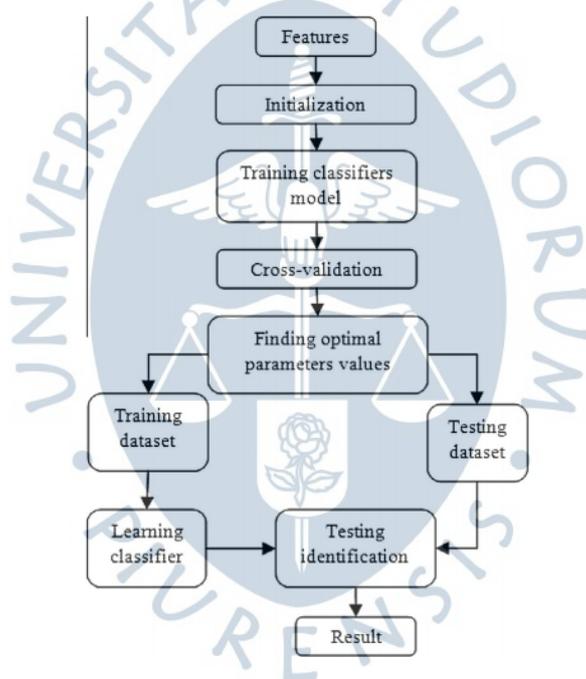
**Figura 47.** Equipo eléctrico (a) Nivel I (b) Nivel II y (c) Nivel III.



Tomado de "Suitable features selection for monitoring thermal condition of electrical equipment using infrared thermography" (Huda & Taib, 2013).

Otra propuesta investiga la idoneidad de las características de entrada y el clasificador para identificar fallas térmicas dentro de las instalaciones eléctricas (Jadin et al., 2014). Las características se extraen de las imágenes térmicas de equipos eléctricos y se clasifican usando una red neuronal artificial de perceptrón multicapa (MLP) y una máquina de vectores de soporte (SVM). En los experimentos, se evaluaron los rendimientos de clasificación de varias características de entrada. Los índices de rendimiento de clasificación comúnmente utilizados, que incluyen sensibilidad, especificidad, precisión, área bajo curva (AUC), característica de funcionamiento del receptor (ROC) y F1-score, se emplean para identificar la característica de entrada más adecuada, así como la mejor configuración de clasificadores. Los resultados experimentales demuestran que la combinación de características Tmax, T delta y DTbg producen la mejor característica de entrada para la detección de fallas térmicas. El flujo para encontrar la configuración óptima de parámetros se plasma en la **Figura 48**.

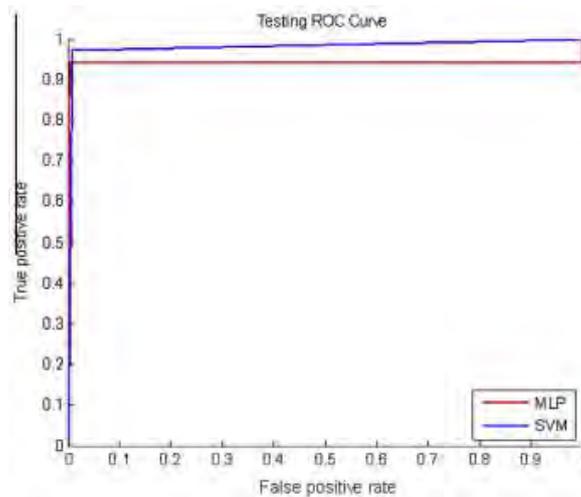
**Figura 48.** Flujo para encontrar la configuración óptima de parámetros.



Tomado de *“Feature extraction and classification for detecting the thermal faults in electrical installations”* (Jadin et al., 2014).

Este estudio presentó un método para encontrar características de entrada adecuadas para detectar la falla térmica en instalaciones eléctricas. A partir del resultado experimental, las características de entrada más adecuadas son la combinación de temperatura máxima, temperatura promedio de la región y diferencia de temperatura entre el objetivo y región de referencia; sin embargo, elegir el tipo de clasificador entre la red neuronal SVM y MLP no tiene un efecto significativo sobre la precisión de la clasificación. Finalmente, luego de evaluar el uso de los clasificadores de red neuronal RBF, kernel SVM y MLP, se indica que la implementación de SVM utilizando la función de núcleo de base radial (RBF) produce un rendimiento ligeramente mejor que la red neuronal artificial MLP (**Figura 49**).

Figura 49. Curva ROC.



Tomado de "Feature extraction and classification for detecting the thermal faults in electrical installations" (Jadin et al., 2014).

La **Tabla 1** resume las principales investigaciones encontradas en la bibliografía.

**Tabla 1.** Principales investigaciones – Clasificación de imágenes térmicas.

Paper	N° imágenes	Accuracy	Técnica
GPU Cuda JSEG Segmentation Algorithm associated with Deep Learning Classifier for Electrical Network Images Identification	No especificado	> 90% (Train)	JSEG YOLO (Alexnet - CNN)
Predictive Maintenance of Power Substation Equipment by Infrared Thermography Using a Machine-Learning Approach	150	84%	MLP MLP + corte de gráfico
A semi-automatic approach for thermographic inspection of electrical installations within buildings	500	78.5%	PCA MLP + algoritmo LM
A novel intelligent fault diagnosis method for electrical equipment using infrared	281	97.8%	SVM y BP K-means
Suitable features selection for monitoring thermal condition of electrical equipment using infrared thermography	336	74.25%	MLP + algoritmo LM

Paper	N° imágenes	Accuracy	Técnica
Feature extraction and classification for detecting the thermal faults in electrical installations	240	99%	SVM (RBF) MLP

Nota. Elaboración propia a partir de “GPU Cuda JSEG Segmentation Algorithm associated with Deep Learning Classifier for Electrical Network Images Identification” (Fambrini et al., 2018), “Predictive Maintenance of Power Substation Equipment by Infrared Thermography Using a Machine-Learning Approach” (Ullah et al., 2017), “A semi-automatic approach for thermographic inspection of electrical installations within buildings” (Nazmul Huda et al., 2012), “A novel intelligent fault diagnosis method for electrical equipment using infrared thermography” (Nazmul Huda et al., 2012), “Suitable features selection for monitoring thermal condition of electrical equipment using infrared thermography” (Zou & Huang, 2015), y “Feature extraction and classification for detecting the thermal faults in electrical installations” (Jadin et al., 2014).





## Capítulo 2 Metodología

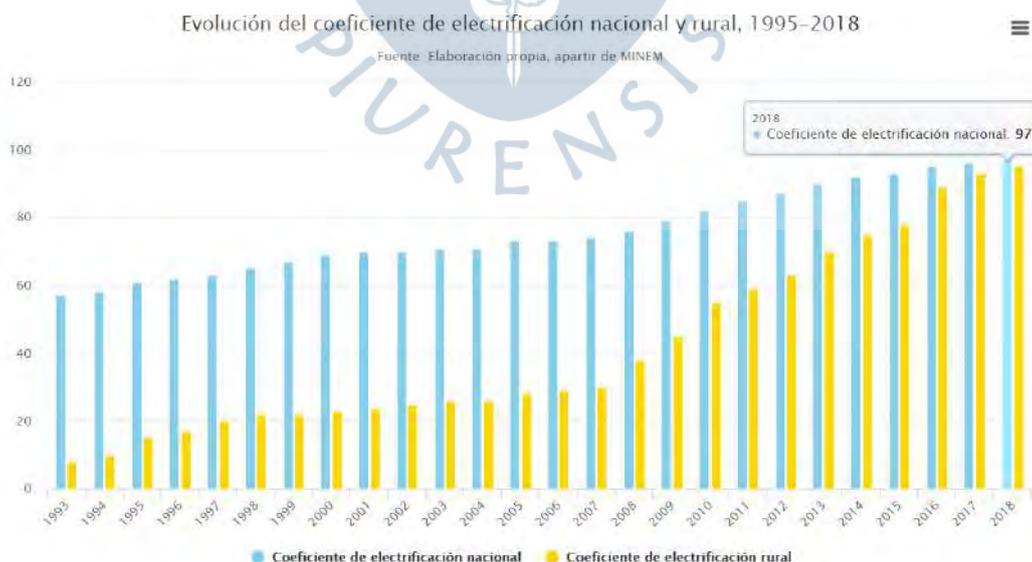
### 1.1 Área de estudio

La investigación se desarrolló con el apoyo de una empresa peruana dedicada a la distribución y comercialización de energía eléctrica en las regiones de Piura y Tumbes, cuyo personal estuvo a cargo de la captura de imágenes térmicas.

### 1.2 Justificación

Según el informe de la Agencia Supervisora de Inversiones en Energía y Minería (OSINERGMIN), con el aumento del coeficiente de electrificación nacional, Perú se ubica en una de las primeras posiciones, con un 97% (**Figura 50**). La legislación peruana ha venido promoviendo y exigiendo que las empresas del sector mejoren sus indicadores de desempeño. Siendo algunos de los más importantes: el Índice de duración de interrupción promedio del sistema (SAIDI) y el Índice de frecuencia de interrupción promedio del sistema (SAIFI) (Kornatka, 2017).

**Figura 50.** Evolución del coeficiente de electrificación nacional peruano.



(\*) Nota: 2018 incluye el efecto de la inversión privada (Paneles Solares).

Tomado de "SAIDI y SAIFI por departamentos" (Osinermin, 2019).

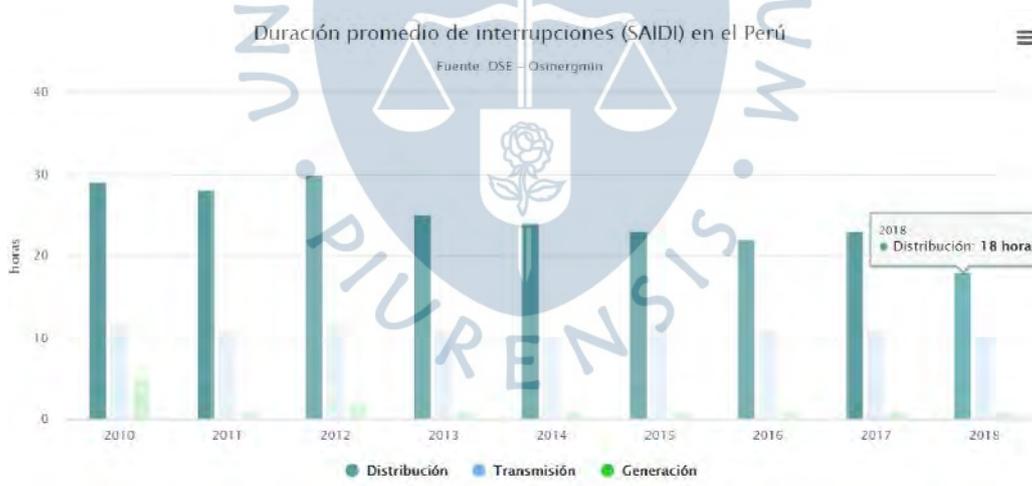
Los indicadores para 2019 en Perú, según OSINERGMIN, son 18 horas (SAIDI) y 8 veces (SAIFI) en todo el país; sin embargo, en las ciudades alejadas de la capital, estos indicadores son mucho más altos y, por lo tanto, de peor rendimiento. Piura, por ejemplo, presenta un SAIDI de 22.08 horas y un SAIFI de 12.21 veces, el cual es aproximadamente el doble del nivel promedio nacional (ver **Figura 51**, **Figura 52** y **Figura 53**).

**Figura 51.** Duración promedio de las interrupciones (SAIDI) en Perú.



Tomado de “SAIDI y SAIFI por departamentos” (Osinergmin, 2019).

**Figura 52.** Frecuencia promedio de interrupciones (SAIFI) en Perú.

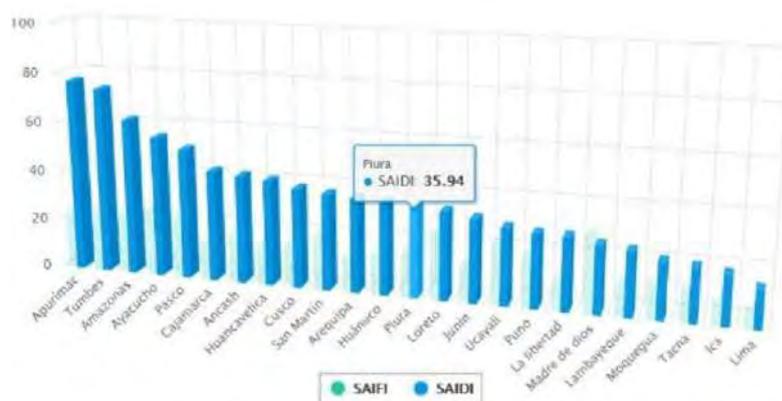


Tomado de “SAIDI y SAIFI por departamentos” (Osinergmin, 2019).

**Figura 53.** SAIDI y SAIFI por regiones

## SAIDI y SAIFI por departamentos – 2018

Fuente: DSR- Osinergmin



Tomado de “SAIDI y SAIFI por departamentos” (Osinergmin, 2019).

En este tipo de ciudades peruanas, la distribución de energía eléctrica se lleva a cabo por empresas públicas con problemas para invertir y financiar nuevas tecnologías y desarrollar investigación aplicada. Sin embargo, durante algunos años han estado aplicando análisis de imágenes térmicas como herramienta de mantenimiento predictivo, el cual les permite predecir fallas en equipos principales como subestaciones eléctricas y líneas de transmisión eléctrica en media (10kV o 22.9kV) y alta tensión (60kV) (Marinescu et al., 2017).

Algunas de las ventajas del mantenimiento predictivo son (Yan et al., 2017):

- Disminución del apagado no programado
- Aumento del uso adecuado de mano de obra
- Aumento del volumen de salida de las plantas.
- Disminuye los gastos de gestión de los instrumentos.
- Aumenta la vida útil de los dispositivos.

Además, debido a que el mantenimiento predictivo contribuye a prevenir fallas imprevistas, evita incurrir en mayores costos de reparación, tiempos más largos para su atención y, por lo tanto, mejoramiento de los indicadores SAIDI y SAIFI; otras de sus ventajas incluyen programar inspecciones en los puntos de falla para determinar los materiales, herramientas y el personal necesarios para corregir los puntos de falla, minimizar el tiempo de interrupción y evitar paradas, desconexiones o interrupciones innecesarias de la fuente de alimentación para verificar los ajustes del cableado del equipo eléctrico.

El proceso que la empresa considera, incluye cinco fases (**Figura 54**), las cuales se explican en los siguientes puntos:

1. Captura de imágenes: externalización de este proceso a una compañía especializada en captura de imágenes térmicas. Este proceso se realiza a una calendarización de visitas a campo con la finalidad de realizar la captura de imágenes térmicas.

2. Entrega de imágenes: Luego que la empresa externa captura las imágenes, entrega el material a un técnico certificado en análisis de imágenes térmicas perteneciente a la compañía de distribución.
3. Clasificación manual: El especialista en imágenes térmicas divide de manera manual todo el set de datos en dos categorías: líneas y subestaciones eléctricas, esto debido a que son equipos que requieren de análisis diferentes.
4. Análisis de fallos: Luego, el especialista analiza imagen por imagen todo el set de datos e búsqueda de fallos en los quipos, los cuales se evidencian por la aparición de puntos calientes en las imágenes térmicas.
5. Plan de acción: Por último, acorde al análisis realizado, el especialista detecta elabora el informe correspondiente indicando los puntos calientes encontrados, y finalmente se toman las medidas correspondientes, tales como: desconexión programada del equipo, ajustes y corrección.

**Figura 54.** Proceso de detección de fallos.



Nota. Elaboración propia.

### 1.3 Objetivos

#### 1.3.1 Objetivo general

Diseñar un clasificador para el mantenimiento predictivo de una red de distribución eléctrica utilizando Deep Learning.

#### 1.3.2 Objetivos específicos

- Etiquetar las imágenes térmicas en dos categorías de equipos: líneas y subestaciones eléctricas.
- Seleccionar y entrenar distintas arquitecturas de clasificación las imágenes térmicas.
- Comparar y analizar los resultados obtenidos.

### 1.4 Hipótesis

El diseño de un clasificador para el mantenimiento predictivo de una red de distribución eléctrica utilizando Deep Learning, contribuirá a que las empresas de distribución eléctrica puedan tomar medidas preventivas sobre los problemas de los equipos de líneas y subestaciones reduciendo de esta manera los inconvenientes que estos aspectos traen consigo.

### 1.5 Propuesta de solución

La presente tesis, pretende automatizar las fases 3 y 4 del proceso (**Figura 55**), mediante la aplicación de Redes Neuronales Convolucionales (CNN), con el objetivo de

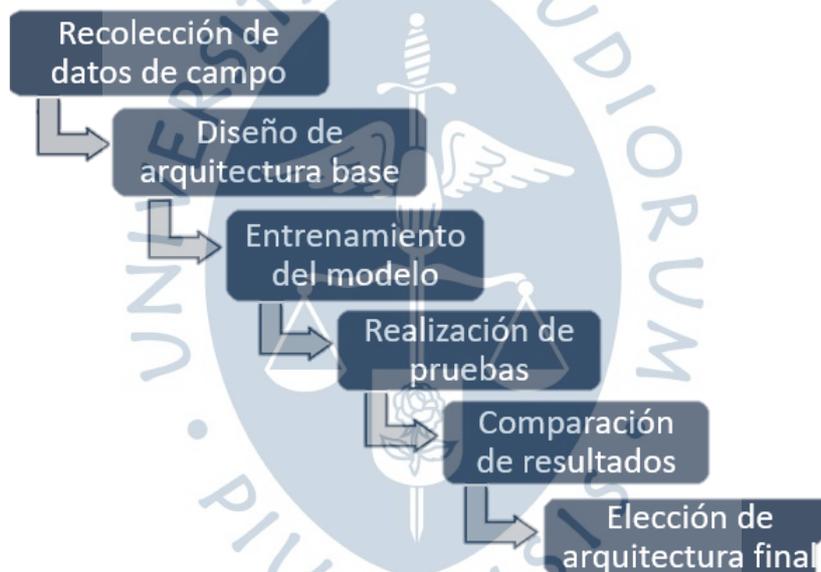
diseñar un clasificador de imágenes térmicas que sirva como mecanismo de ayuda en el mantenimiento predictivo. La **Figura 56** muestra cada uno de los pasos desarrollados en la tesis, partiendo desde la recolección de datos de campo (imágenes térmicas), hasta llegar a la elección de una arquitectura, luego de haber realizado los entrenamientos y pruebas correspondientes. Estos últimos se realizaron mediante la herramienta Google Colaboratory, en su versión gratuita (Carneiro et al., 2018).

**Figura 55.** Propuesta de solución.



Nota. Elaboración propia.

**Figura 56.** Metodología general del proyecto.



Nota. Elaboración propia.



## Capítulo 3 Modelo

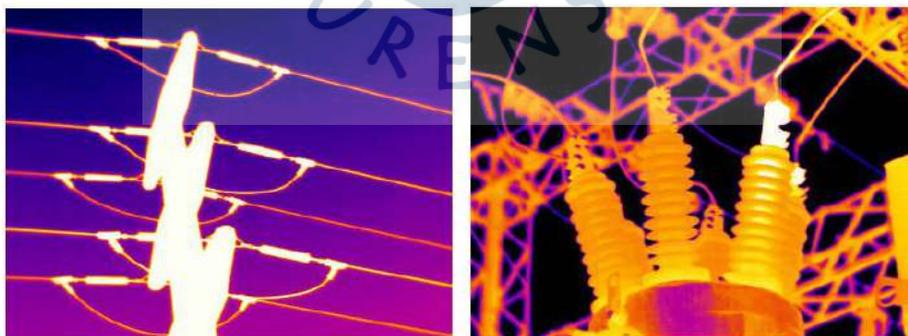
### 2.1 Set de datos

En una campaña de recolección de datos se genera gran cantidad de imágenes de equipos eléctricos, por lo tanto, reducir el tiempo de clasificación es una contribución importante al proceso; esto se debe a que normalmente el personal calificado para desarrollar este proceso, realiza muchas otras tareas, y cualquier contribución que reduzca el tiempo o elimine el trabajo repetitivo es extremadamente importante, además, que el porcentaje de error humano se reduce implícitamente.

Entonces, acorde a lo descrito en la **Figura 54**, el proceso inicia con la captura de imágenes, para ello se utiliza la cámara infrarroja TP8S, la cual incluye un rango espectral de 8-14 $\mu$ m, y densidad térmica de 0.08°C a 130 °C.

La base de datos utilizada para este trabajo contiene imágenes térmicas con una resolución de 384 x 288 píxeles. En la **Figura 57** se aprecian algunos ejemplos de imágenes térmicas de equipos de Líneas de transmisión y Subestaciones eléctricas, las cuales corresponden a las dos clases o categorías a clasificar.

**Figura 57.** Imagen típica de equipos de líneas de transmisión y subestaciones eléctricas.

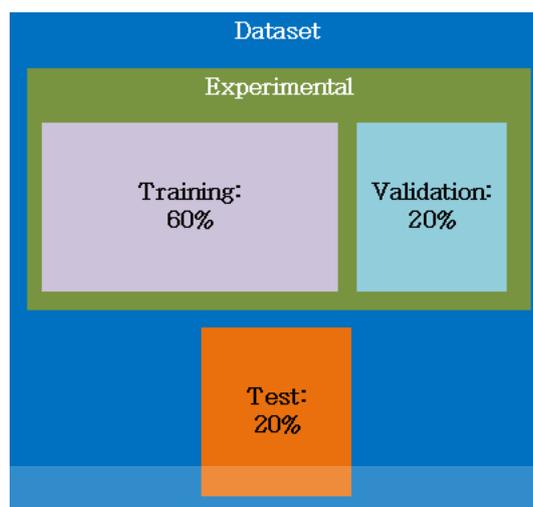


(a) Línea de transmisión

(b) Subestación eléctrica

Nota. Elaboración propia.

Se consideró un set de datos con un total de 815 imágenes térmicas, además de aplicar una división de los mismos de acuerdo a la **Figura 58** y **Tabla 2**.

**Figura 58.** Distribución del dataset.

Nota. Elaboración propia.

**Tabla 2.** Distribución del dataset.

Clase	Entrenamiento	Validación	Pruebas	TOTAL
Línea	206	68	68	342
Subestación	283	95	95	473
<b>TOTAL</b>	<b>489</b>	<b>163</b>	<b>163</b>	<b>815</b>

Nota. Elaboración propia.

## 2.2 Arquitecturas de transfer learning

La investigación partió por la realización de un estudio comparativo de once arquitecturas de transfer learning, con el objetivo de realizar una clasificación binaria de dos categorías: líneas y subestaciones eléctricas, tal como se muestra en la **Tabla 3**, la cual fue elaborada mediante consulta de información oficial de TorchVision (PyTorch, 2019). Cabe resaltar que ShuffleNet, es la única arquitectura que pesa 0 MB debido a que, en su versión actual, no permite el uso de sus pesos previamente entrenados.

**Tabla 3.** Arquitecturas de TF utilizadas – Peso y parámetros.

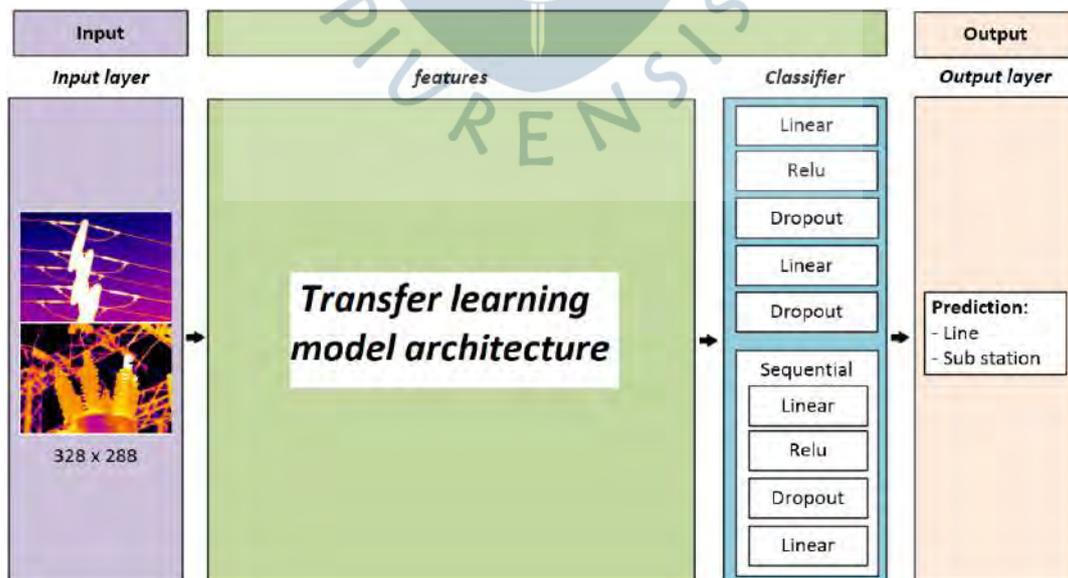
#	Arquitectura	Peso en MB	Millones de parámetros
1	AlexNet	233	61.1
2	VGG16	528	138.36
3	VGG19	548	143.67

#	Arquitectura	Peso en MB	Millones de parámetros
4	ResNet	230	60.19
5	DenseNet	77.4	20.01
6	GoogLeNet	49.7	13
7	MobileNet v2	13.6	3.5
8	ResNeXt	340	44.55
9	Wide ResNetx	243	126.89
10	MNASNet	16.9	4.38
11	ShuffleNet	0*	7.39

Nota. Elaboración propia.

Se diseñó una arquitectura base (ver **Figura 59**), en la que cada una de las 11 arquitecturas fue ingresada, realizando un cambio en la capa de clasificación, en la que se seleccionó una salida binaria. El set de datos de entrenamiento corresponde a la entrada del modelo, y la salida de este es la clasificación de las imágenes térmicas en líneas y subestaciones eléctricas. Además, se configuraron los 11 modelos con los hiperparámetros mostrados en la **Tabla 4**.

**Figura 59.** Arquitectura base.



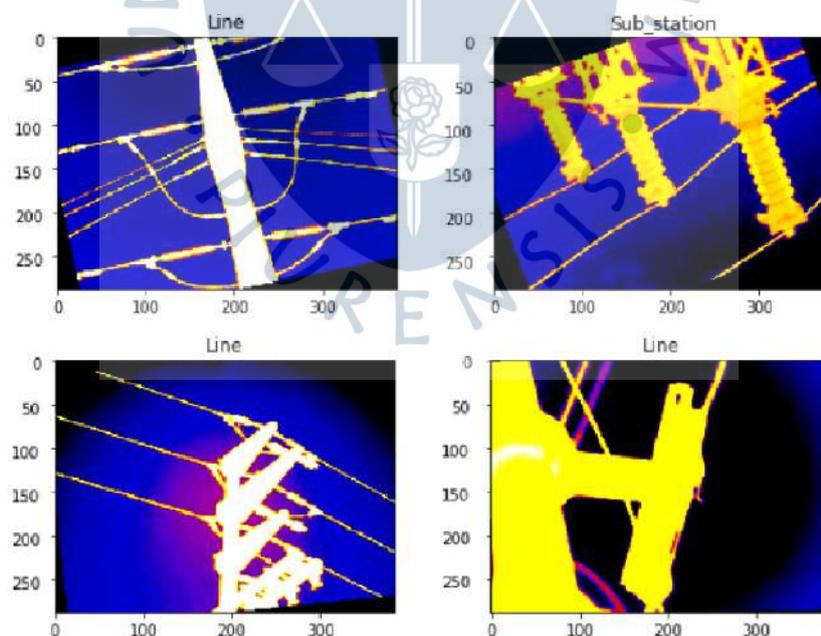
Nota. Elaboración propia.

**Tabla 4.** Configuración de los principales hiperparámetros.

Hiperparámetro	Valor
Tasa de aprendizaje	0.0001
Imágenes por Batch	32
Número de épocas	20
Función de costo	Multi Margin Loss
Optimizador	Adam

Nota. Elaboración propia.

Además de estas configuraciones, se normalizaron los datos de entrada mediante el cálculo de la media y desviación estándar, la aplicación de data augmentation mediante el uso de las transformaciones random rotation, random vertical flip y random horizontal flip; lo cual contribuyó en la realización de entrenamientos con un grado de calidad más alto. En la **Figura 60** se aprecian algunos ejemplos de las imágenes térmicas luego de haber aplicado las transformaciones en mención.

**Figura 60.** Imágenes con data augmentation.

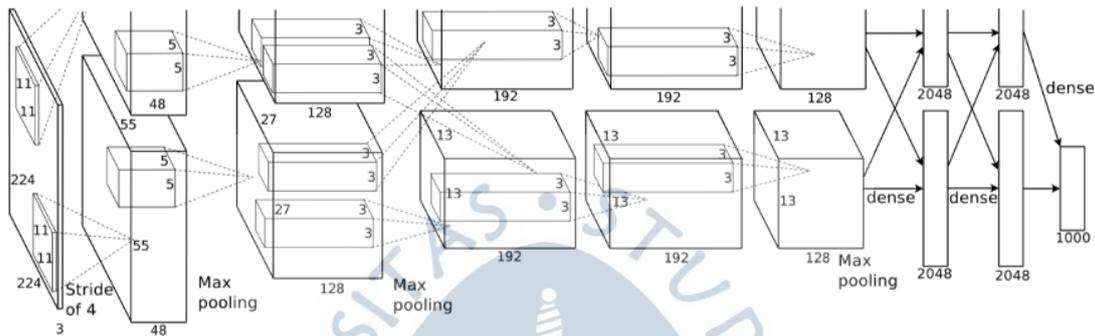
Nota. Elaboración propia.

A continuación, se presentan las descripciones y resultados de las arquitecturas de transfer learning utilizadas:

### 2.2.1 AlexNet

AlexNet es una CNN compuesta por ocho capas, 3 fully connected y 5 max-pooling. La función de activación que utiliza es Rectified Linear Units (ReLU), y fue entrenada con una de las bases de datos de imágenes más conocidas y utilizadas: ImageNet (Almisreb et al., 2018), en la que cada imagen presenta un tamaño de 224x224 píxeles, y van siendo transformadas en cada capa, hasta tener como salida la clasificación de las mismas (**Figura 61**).

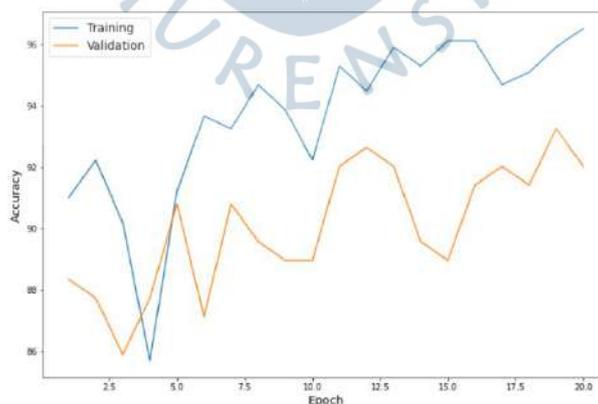
**Figura 61.** Arquitectura AlexNet.



Tomado de *“ing AlexNet Deep Transfer Learning for Ear Recognition”* (Almisreb et al., 2018).

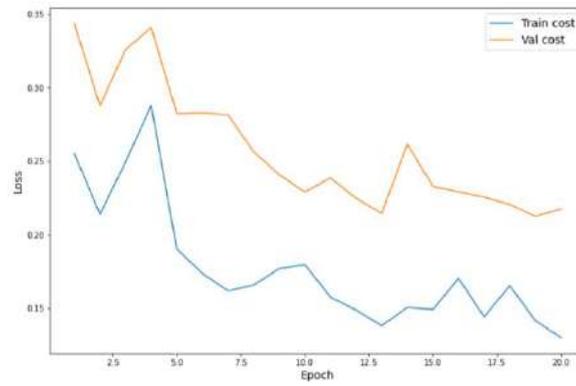
Luego de entrenar el modelo, se obtuvieron exactitudes de 95.50%, 91.41% y 93.86% en el entrenamiento, validación y pruebas respectivamente, con un F1-score de 91.44%, y un tiempo de ejecución de 6.30 minutos. La **Figura 62** muestra el resultado de la precisión en el entrenamiento y validación, la **Figura 63** la tasa de pérdida, la **Figura 64** la matriz de confusión con los porcentajes de cada una de las clases con los datos de prueba, y la **Figura 65** la curva ROC en la que se aprecia un valor AUC de 0.90.

**Figura 62.** AlexNet: Accuracy en Entrenamiento y Validación.



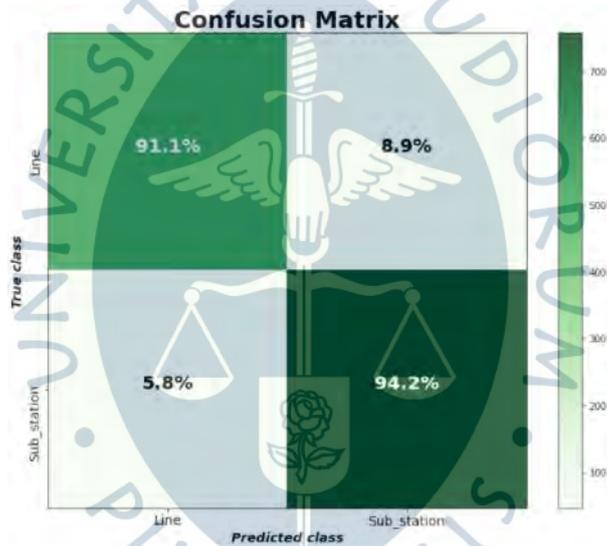
Nota. Elaboración propia.

**Figura 63.** AlexNet: Tasa de pérdida en Entrenamiento y Validación.



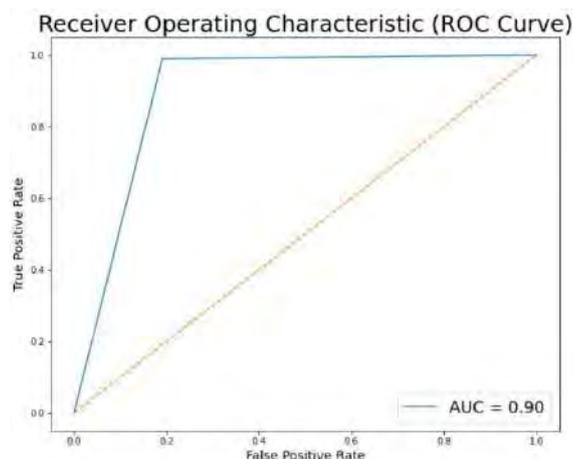
Nota. Elaboración propia.

**Figura 64.** AlexNet: Matriz de confusión.



Nota. Elaboración propia.

**Figura 65.** AlexNet: Curva ROC.

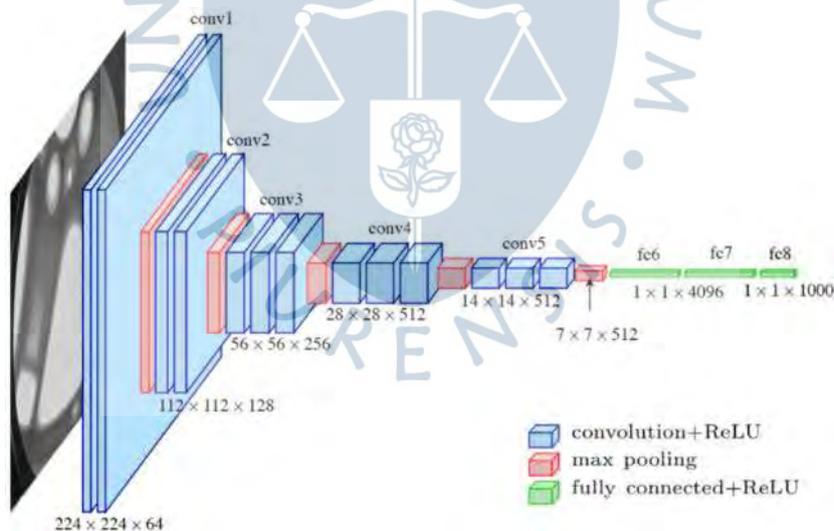


Nota. Elaboración propia.

### 2.2.2 VGG16

Es una CNN integrada por 16 capas (**Figura 66**). Al igual que AlexNet, fue entrenada con ImageNet, pero con esta arquitectura, se obtuvieron mejoras significativas debido a que se consideró un conjunto de filtros de tamaño 3x3 en lugar de los grandes filtros de cada kernel de AlexNet (S. Liu & Deng, 2016).

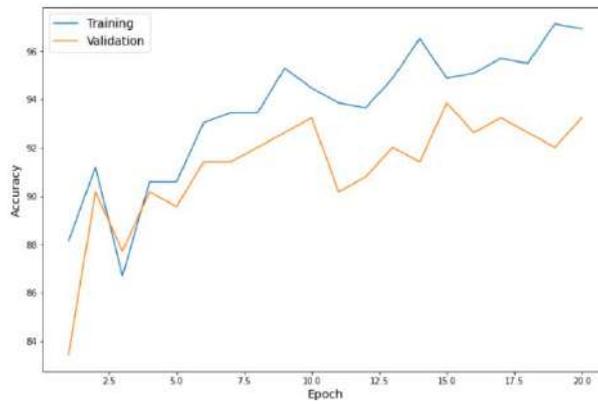
**Figura 66.** Arquitectura VGG16.



Tomado de "Very deep convolutional neural network based image classification using small training sample size" (S. Liu & Deng, 2016).

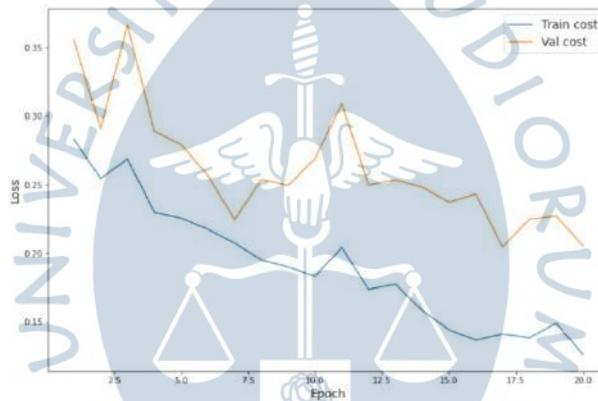
Luego de entrenar el modelo, se lograron obtener exactitudes de 95.91% en el entrenamiento, 91.41% en la validación y 95.09% en las pruebas, con un F1-score de 95.11%, y un tiempo de ejecución de 12.48 minutos. La **Figura 67** muestra el resultado de la precisión en el entrenamiento y validación, la **Figura 68** la tasa de pérdida, la **Figura 69** la matriz de confusión con los porcentajes de cada una de las clases con los datos de prueba, y la **Figura 70** la curva ROC en la que se aprecia un valor AUC de 0.94.

**Figura 67.** VGG16: Accuracy en Entrenamiento y Validación.



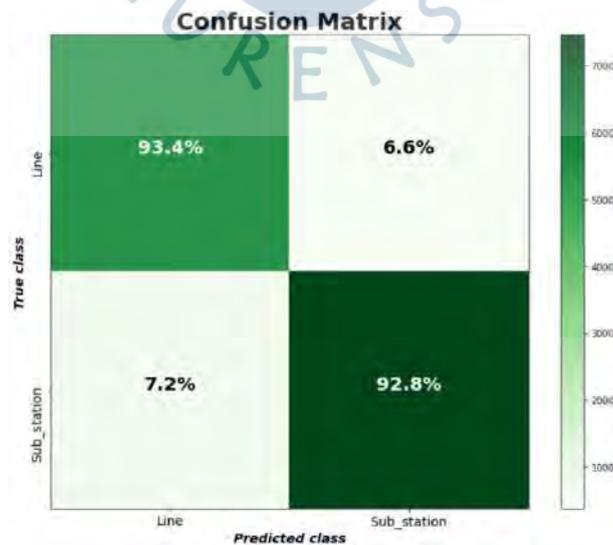
Nota. Elaboración propia.

**Figura 68.** VGG16: Tasa de pérdida en Entrenamiento y Validación.

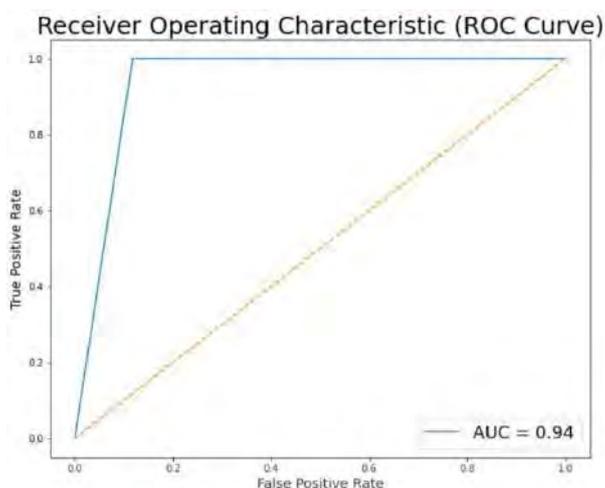


Nota. Elaboración propia.

**Figura 69.** VGG16: Matriz de confusión.



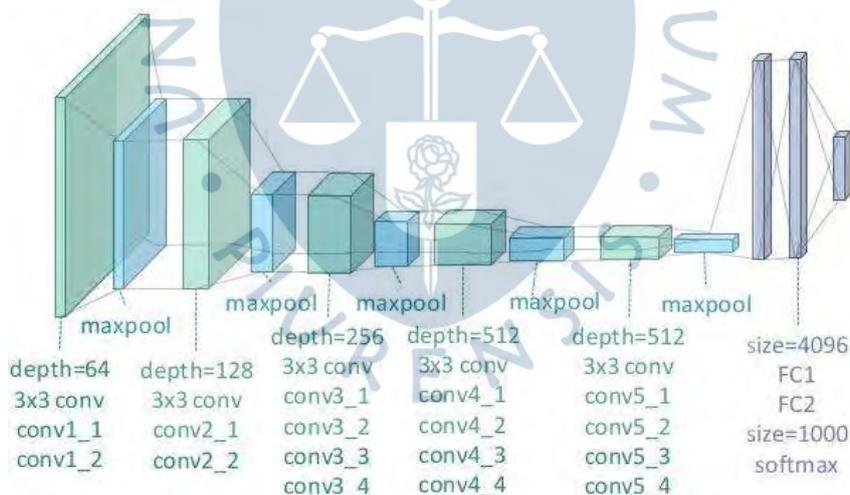
Nota. Elaboración propia.

**Figura 70.** VGG16: Curva ROC.

Nota. Elaboración propia.

### 2.2.3 VGG19

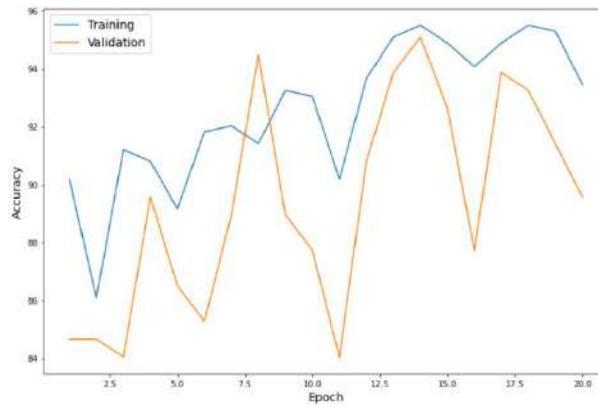
A diferencia de VGG16, VGG19 es una CNN compuesta por 19 capas convolucionales (**Figura 71**), divididas entre SoftMax, MaxPool y fully-connected, las cuales generan alrededor de 143,000,000 de parámetros (Xiao et al., 2020).

**Figura 71.** Arquitectura VGG19.

Tomado de *“Application of a Novel and Improved VGG-19 Network in the Detection of Workers Wearing Masks”* (Xiao et al., 2020).

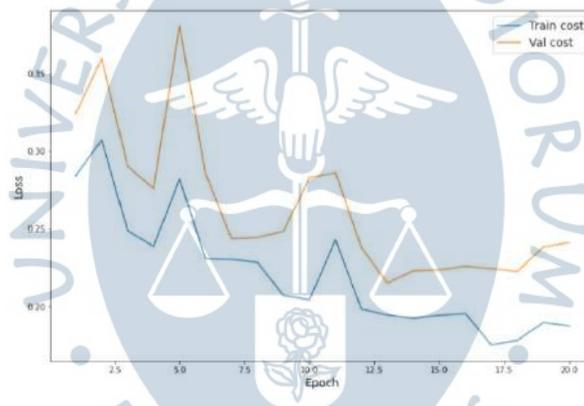
Luego de entrenar el modelo, se obtuvieron exactitudes de 93.66%, 90.18% y 96.31% en el entrenamiento, validación y pruebas respectivamente, con un F1-score de 92.55%, y un tiempo de ejecución de 14.50 minutos. La **Figura 72** muestra el resultado de la precisión en el entrenamiento y validación, la **Figura 73** la tasa de pérdida, la **Figura 74** la matriz de confusión con los porcentajes de cada una de las clases con los datos de prueba, y la **Figura 75** la curva ROC en la que se aprecia un valor AUC de 0.93.

**Figura 72.** VGG19: Accuracy en Entrenamiento y Validación.



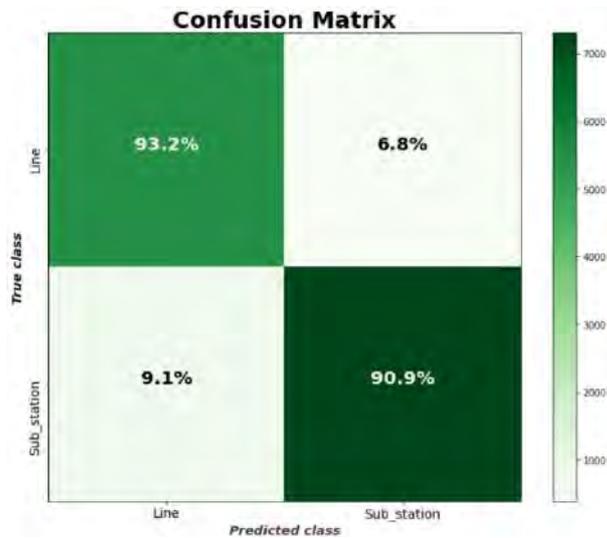
Nota. Elaboración propia.

**Figura 73.** VGG19: Tasa de pérdida en Entrenamiento y Validación.



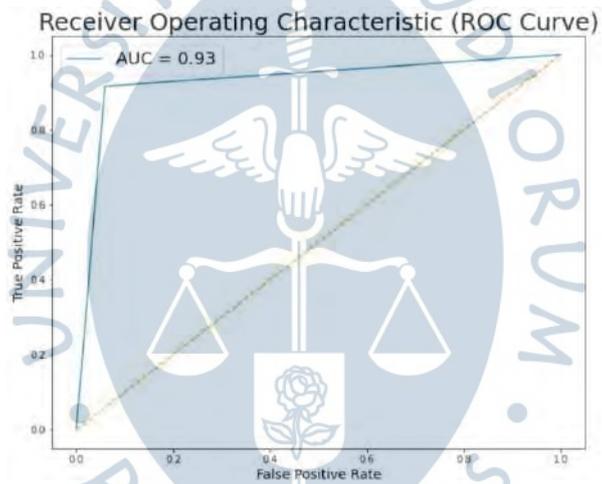
Nota. Elaboración propia.

**Figura 74.** VGG19: Matriz de confusión.



Nota. Elaboración propia.

**Figura 75.** VGG19: Curva ROC.

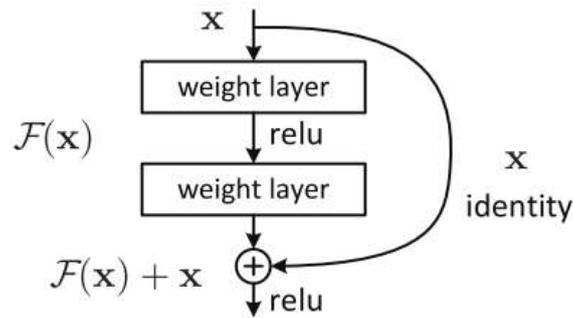


Nota. Elaboración propia.

#### 2.2.4 ResNet

ResNet (Figura 76) es una CNN que agrega una conexión residual mediante una capa de identidad, buscando que el número de capas incremente. La capa identidad pasa por todas las demás de manera directa, con lo que se logran mejores resultados en el entrenamiento (Budhiman et al., 2019).

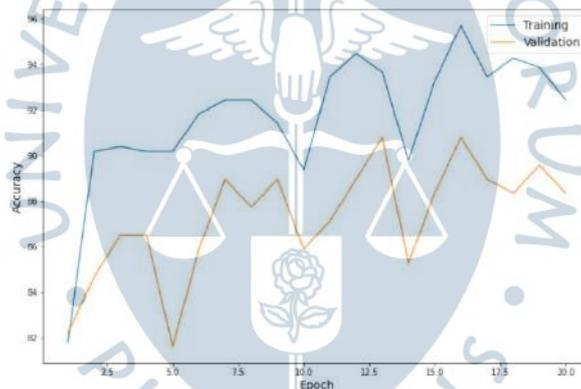
**Figura 76.** Bloque de la arquitectura ResNet.



Tomado de “Melanoma Cancer Classification Using ResNet with Data Augmentation” (Budhiman et al., 2019).

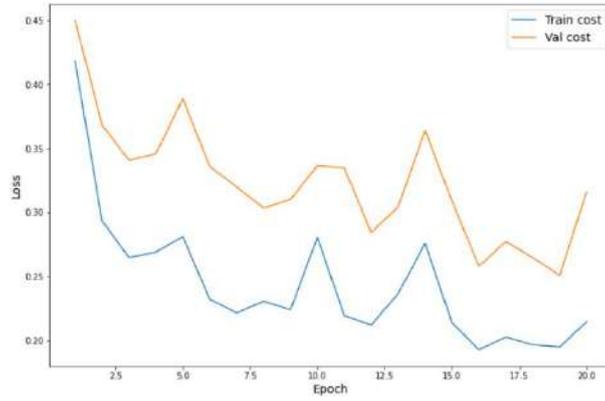
Luego de entrenar el modelo, se obtuvieron exactitudes de 93.66%, 84.66% y 94.47% en el entrenamiento, validación y pruebas respectivamente, con un F1-score de 92.92%, y un tiempo de ejecución de 13.58 minutos. La **Figura 77** muestra el resultado de la precisión en el entrenamiento y validación, la **Figura 78** la tasa de pérdida, la **Figura 79** la matriz de confusión con los porcentajes de cada una de las clases con los datos de prueba, y la **Figura 80** la curva ROC en la que se aprecia un valor AUC de 0.93.

**Figura 77.** ResNet: Accuracy en Entrenamiento y Validación.



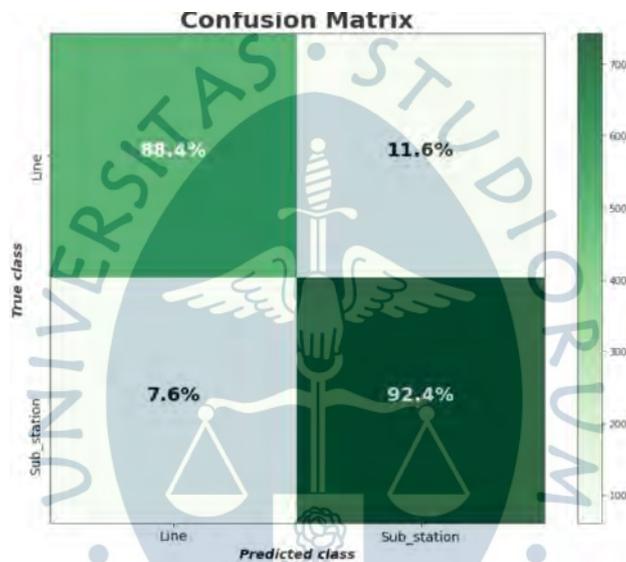
Nota. Elaboración propia.

**Figura 78.** ResNet: Tasa de pérdida en Entrenamiento y Validación.



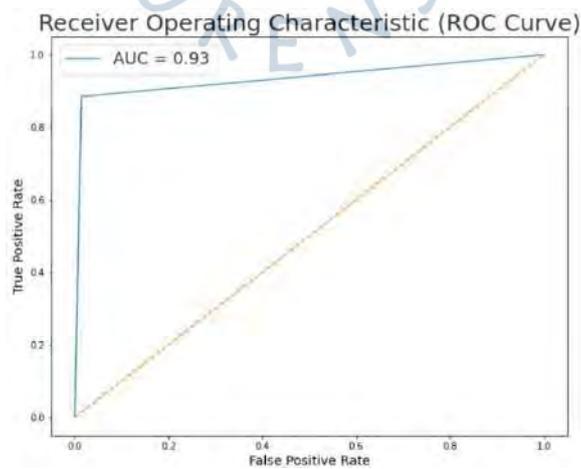
Nota. Elaboración propia.

**Figura 79.** ResNet: Matriz de confusión.



Nota. Elaboración propia.

**Figura 80.** ResNet: Curva ROC.

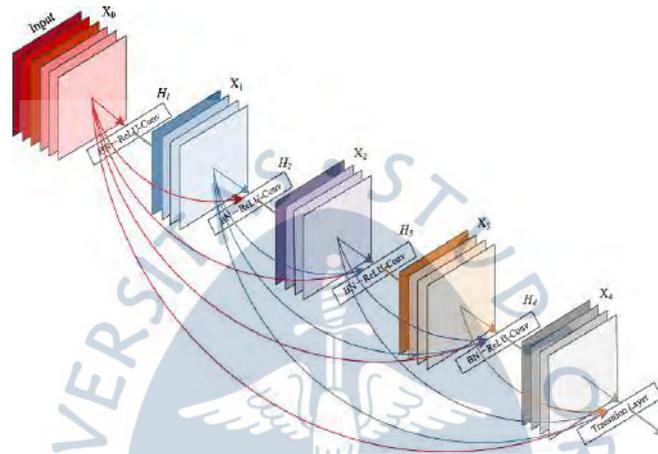


Nota. Elaboración propia.

### 2.2.5 DenseNet

DenseNet (**Figura 81**) es una red neuronal convolucional configurada para que cada capa obtenga entradas adicionales de las demás capas anteriores a esta, además de que pase o comparta sus mapas de características a las capas subsiguientes. Esto quiere decir que existe una concatenación de las capas anteriores y posteriores. DenseNet realiza esta configuración intentando obtener una alta exactitud con la menor cantidad de parámetros (Zhang et al., 2019).

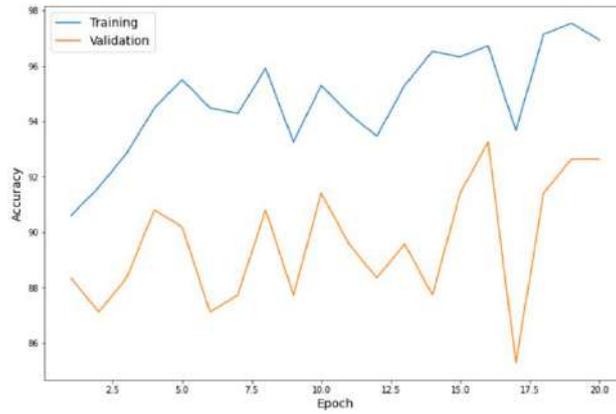
**Figura 81.** Bloque de la arquitectura DenseNet.



Tomado de “Multiple feature reweight DenseNet for image classification” (Zhang et al., 2019).

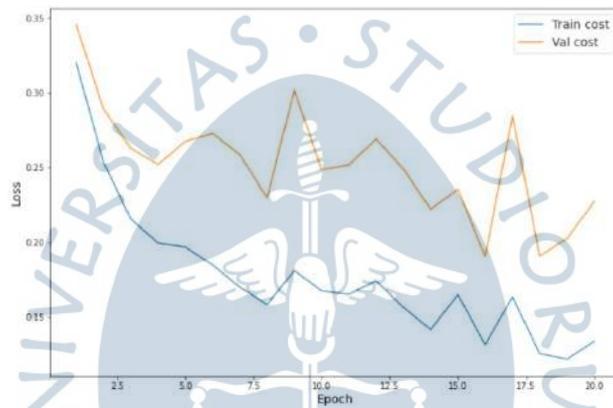
Luego de entrenar el modelo, se obtuvieron exactitudes de 96.52%, 92.02% y 98.15% en el entrenamiento, validación y pruebas respectivamente, con un F1-score de 92.53%, y un tiempo de ejecución de 6.67 minutos. La **Figura 82** muestra el resultado de la precisión en el entrenamiento y validación, la **Figura 83** la tasa de pérdida, la **Figura 84** la matriz de confusión con los porcentajes de cada una de las clases con los datos de prueba, y la **Figura 85** la curva ROC en la que se aprecia un valor AUC de 0.92.

**Figura 82.** DenseNet: Accuracy en Entrenamiento y Validación.



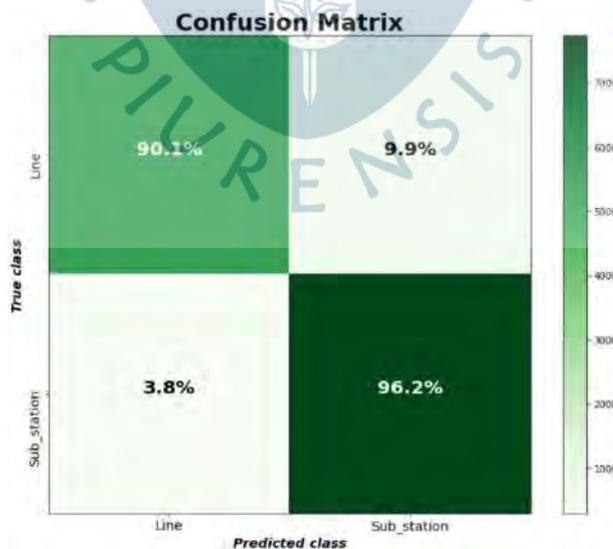
Nota. Elaboración propia.

**Figura 83.** DenseNet: Tasa de pérdida en Entrenamiento y Validación.



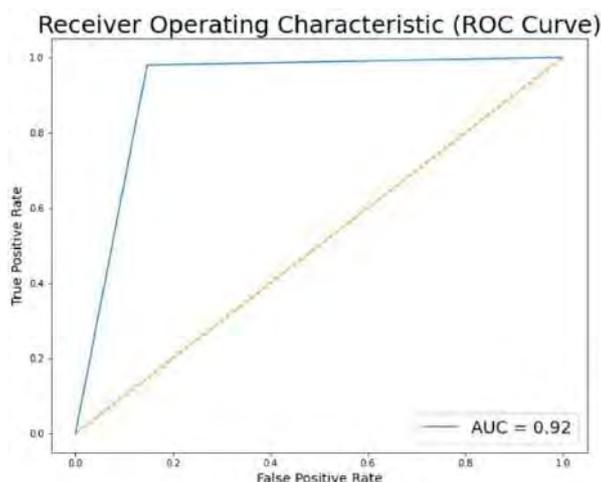
Nota. Elaboración propia.

**Figura 84.** DenseNet: Matriz de confusión.



Nota. Elaboración propia.

**Figura 85.** DenseNet: Curva ROC.

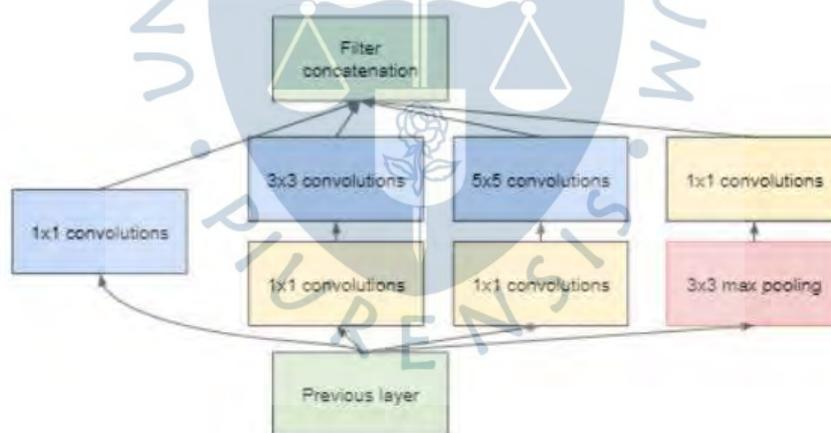


Nota. Elaboración propia.

## 2.2.6 GoogLeNet

CNN desarrollada y diseñada por Google (Fang, 2018), es objetivo principal es clasificar imágenes, es por ello que esta red neuronal convolucional se basa en la arquitectura Inception (Szegedy et al., 2015), a través del uso de módulos que posibilitan la opción de elegir filtros convolucionales de tamaños diferentes. En la **Figura 86** se muestra un módulo de esta arquitectura.

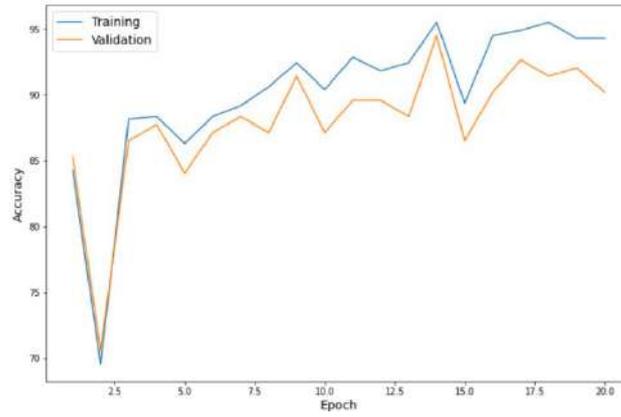
**Figura 86.** Módulo inception con reducciones de dimensión.



Tomado de “Understanding GoogLeNet Model - CNN Architecture” (GeeksforGeeks, 2020).

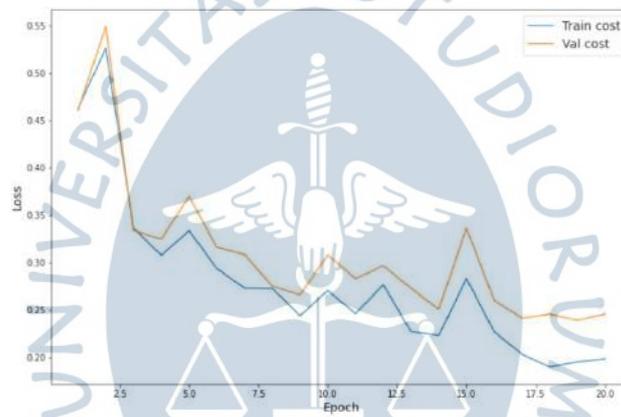
Luego de entrenar el modelo, se obtuvieron exactitudes de 95.09%, 88.95% y 93.86% en el entrenamiento, validación y pruebas respectivamente, con un F1-score de 90.73%, y un tiempo de ejecución de 6.02 minutos. La **Figura 87** muestra el resultado de la precisión en el entrenamiento y validación, la **Figura 88** la tasa de pérdida, la **Figura 89** la matriz de confusión con los porcentajes de cada una de las clases con los datos de prueba, y la **Figura 90** la curva ROC en la que se aprecia un valor AUC de 0.89.

**Figura 87.** GoogLeNet: Accuracy en Entrenamiento y Validación.



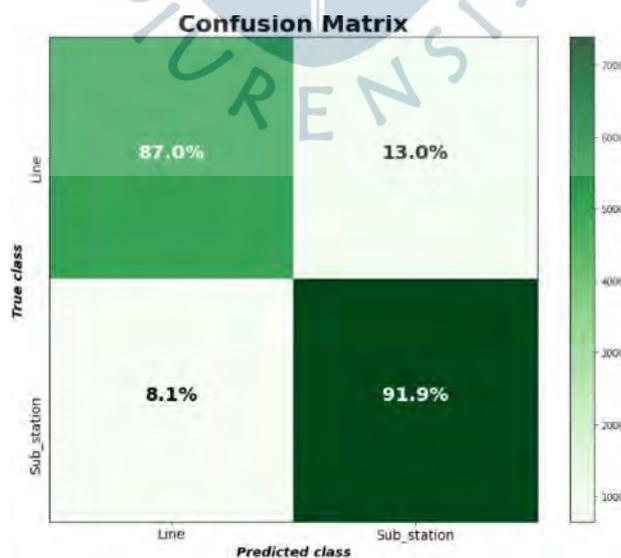
Nota. Elaboración propia.

**Figura 88.** GoogLeNet: Tasa de pérdida en Entrenamiento y Validación.



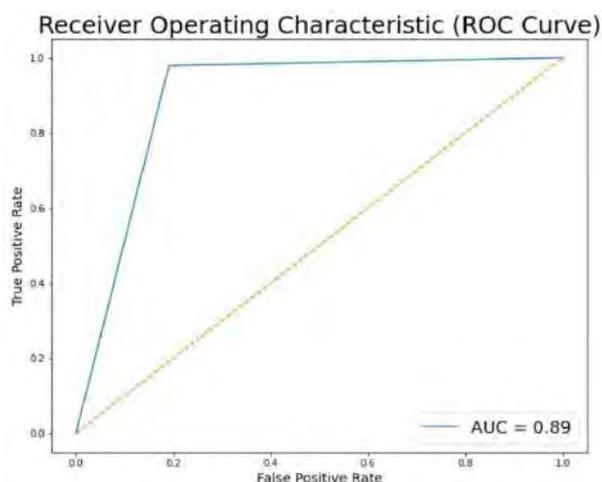
Nota. Elaboración propia.

**Figura 89.** GoogLeNet: Matriz de confusión.



Nota. Elaboración propia.

**Figura 90.** GoogLeNet: Curva ROC.

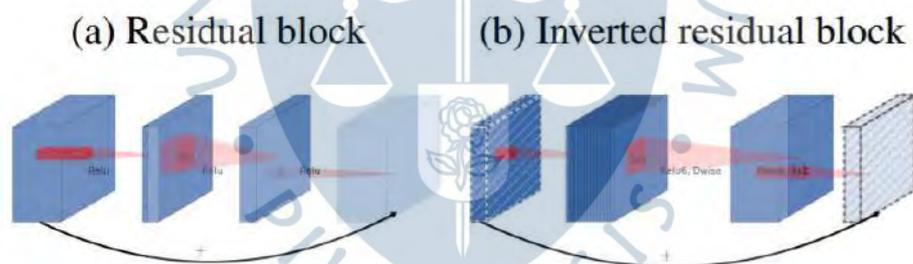


Nota. Elaboración propia.

### 2.2.7 MobileNet v2

MobileNet es una red neuronal convolucional que utiliza una estructura residual invertida, es decir que las entradas y salidas de los bloques residuales son capas cada vez más delgadas (Figura 91), denominadas como cuello de botella, las cuales son opuestas a los tradicionales modelos residuales que incluyen una representación expandida en la entrada (Fu et al., 2019).

**Figura 91.** Bloques de la arquitectura MobileNet v2.

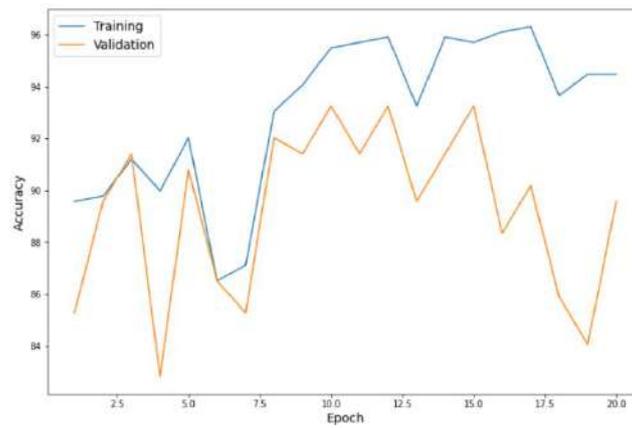


Tomado de "ext Detection for Natural Scene based on MobileNet V2 and U-Net" (Fu et al., 2019).

Luego de entrenar el modelo, se obtuvieron exactitudes de 94.68%, 88.95% y 95.70% en el entrenamiento, validación y pruebas respectivamente, con un F1-score de 93.08%, y un tiempo de ejecución de 5.65 minutos. La Figura 92 muestra el resultado de la precisión en el entrenamiento y validación, la

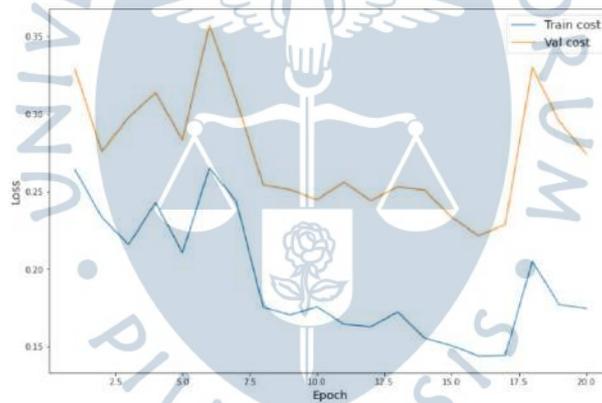
Figura 93 la tasa de pérdida, la Figura 94 la matriz de confusión con los porcentajes de cada una de las clases con los datos de prueba, y la Figura 95 la curva ROC en la que se aprecia un valor AUC de 0.93.

**Figura 92.** *MobileNet v2: Accuracy en Entrenamiento y Validación.*



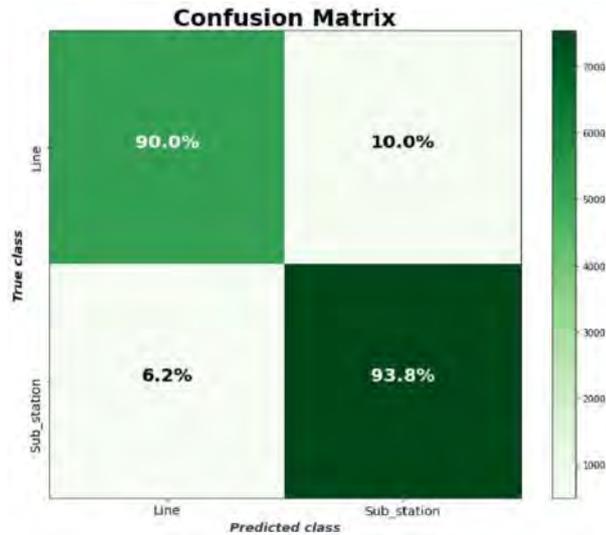
Nota. Elaboración propia.

**Figura 93.** *MobileNet v2: Tasa de pérdida en Entrenamiento y Validación.*



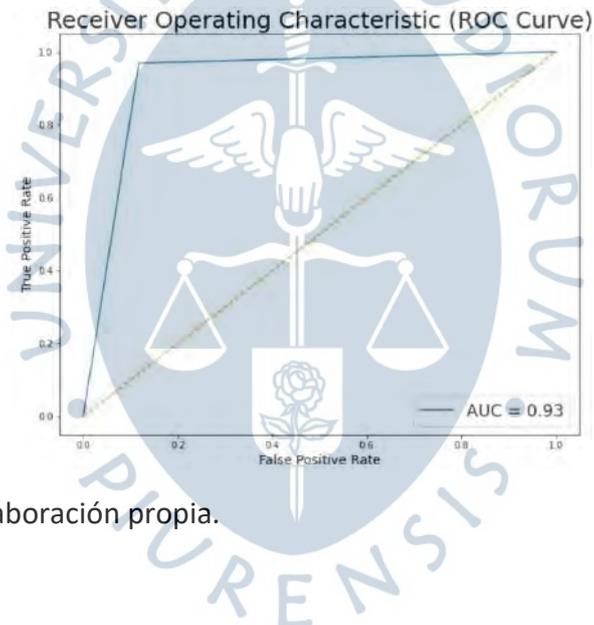
Nota. Elaboración propia.

**Figura 94.** *MobileNet v2: Matriz de confusión.*



Nota. Elaboración propia.

**Figura 95.** MobileNet v2: Curva ROC.

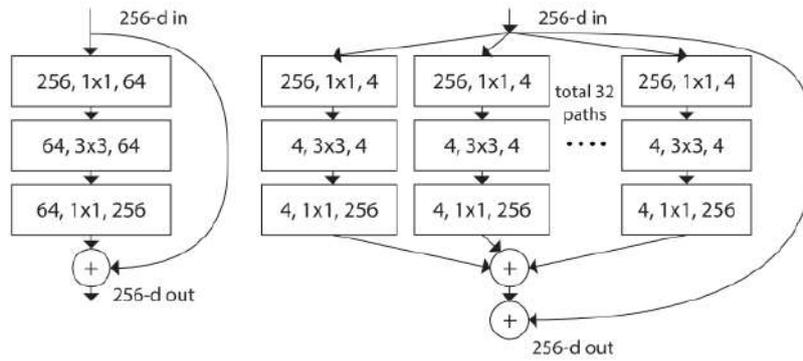


Nota. Elaboración propia.

### 2.2.8 ResNeXt

ResNext es una variación de la CNN ResNet; la diferencia más significativa es que la primera incrementa la cantidad de rutas paralelas o caminos de las conexiones residuales (**Figura 96**), es decir, con diversas ramas (Qiu et al., 2019).

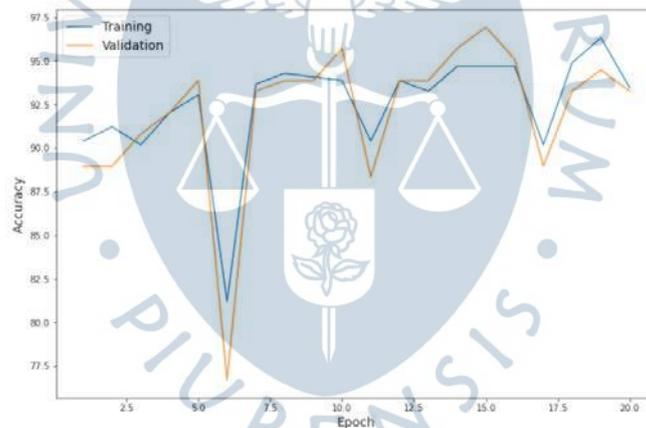
Figura 96. Bloque de la arquitectura ResNeXt.



Tomado de “Mapping human settlements with multi-seasonal sentinel-2 imagery and attention-based resnext” (Qiu et al., 2019).

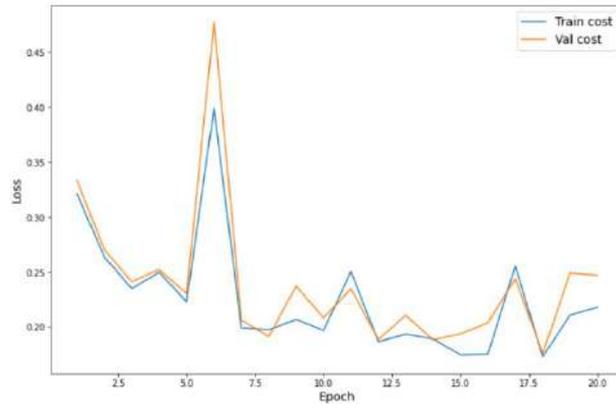
Luego de entrenar el modelo, se obtuvieron exactitudes de 94.06%, 92.02% y 95.09% en el entrenamiento, validación y pruebas respectivamente, con un F1-score de 93.32%, y un tiempo de ejecución de 20.21 minutos. La **Figura 97** muestra el resultado de la precisión en el entrenamiento y validación, la **Figura 98** la tasa de pérdida, la **Figura 99** la matriz de confusión con los porcentajes de cada una de las clases con los datos de prueba, y la **Figura 100** la curva ROC en la que se aprecia un valor AUC de 0.94.

**Figura 97.** ResNeXt: Accuracy en Entrenamiento y Validación.



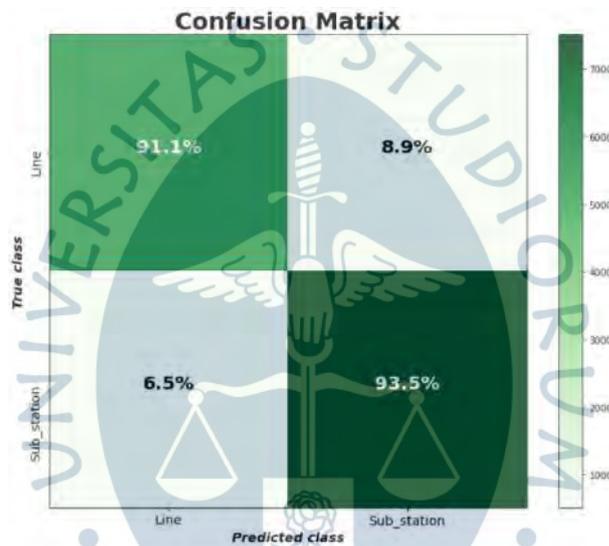
Nota. Elaboración propia.

**Figura 98.** ResNeXt: Tasa de pérdida en Entrenamiento y Validación.



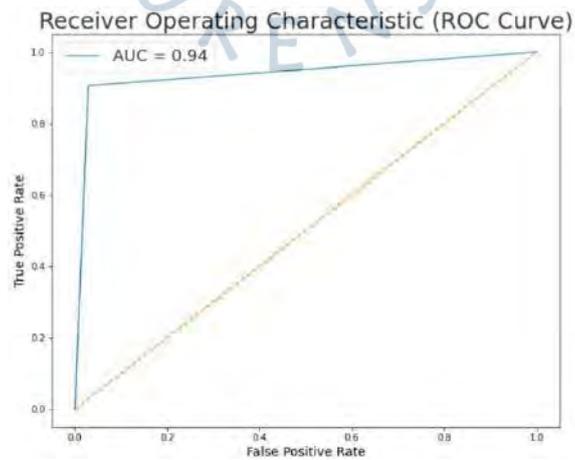
Nota. Elaboración propia.

**Figura 99.** ResNeXt: Matriz de confusión.



Nota. Elaboración propia.

**Figura 100.** ResNeXt: Curva ROC.

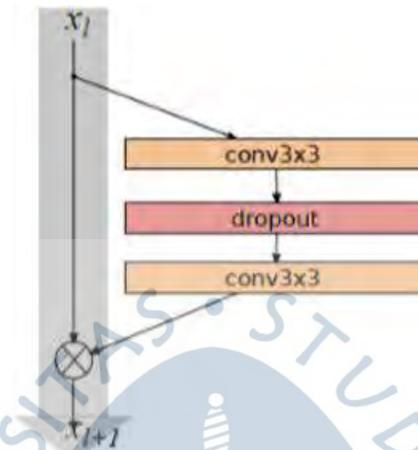


Nota. Elaboración propia.

### 2.2.9 Wide ResNet

Variación de la CNN ResNet, que aumenta el ancho de la red residual y disminuye la profundidad del modelo, siendo los bloques más utilizados: wide-dropout (**Figura 101**), basic-wide, bottleneck y basic (Zagoruyko & Komodakis, 2016).

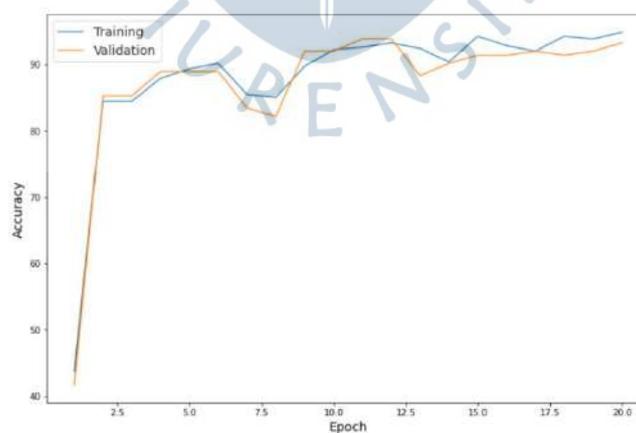
**Figura 101.** Bloque wide-dropout.



Tomado de “Wide Residual Networks” (Zagoruyko & Komodakis, 2016).

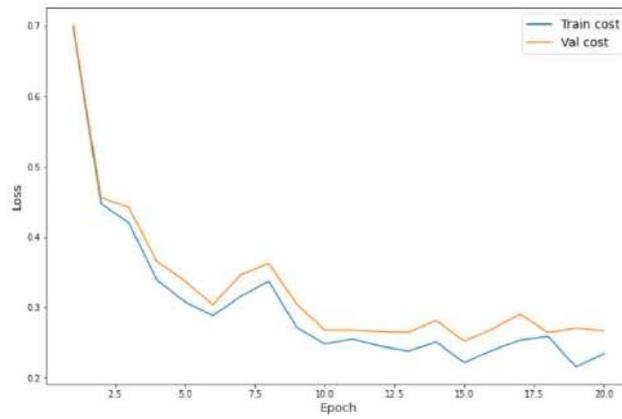
Luego de entrenar el modelo, se obtuvieron exactitudes de 94.68%, 90.18% y 96.31% en el entrenamiento, validación y pruebas respectivamente, con un F1-score de 92.46%, y un tiempo de ejecución de 20.03 minutos. La **Figura 102** muestra el resultado de la precisión en el entrenamiento y validación, la **Figura 103** la tasa de pérdida, la **Figura 104** la matriz de confusión con los porcentajes de cada una de las clases con los datos de prueba, y la **Figura 105** la curva ROC en la que se aprecia un valor AUC de 0.92.

**Figura 102.** Wide ResNet: Accuracy en Entrenamiento y Validación.



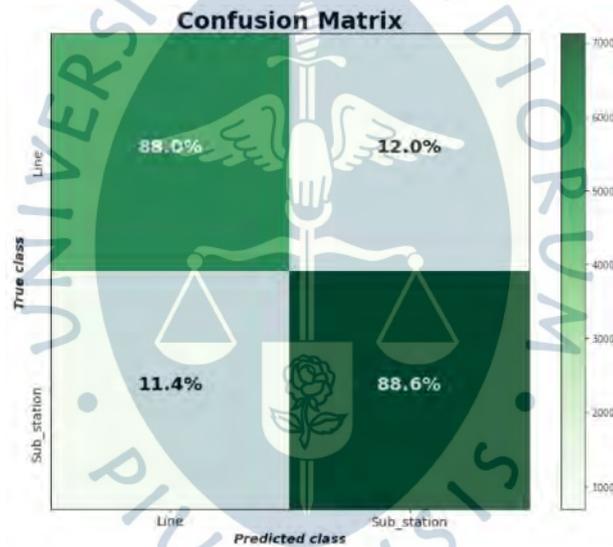
Nota. Elaboración propia.

**Figura 103.** Wide ResNet: Tasa de pérdida en Entrenamiento y Validación.



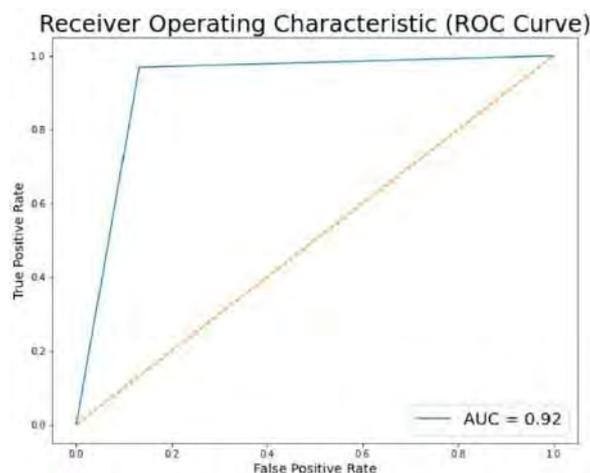
Nota. Elaboración propia.

**Figura 104.** Wide ResNet: Matriz de confusión.



Nota. Elaboración propia.

**Figura 105.** Wide ResNet: Curva ROC.

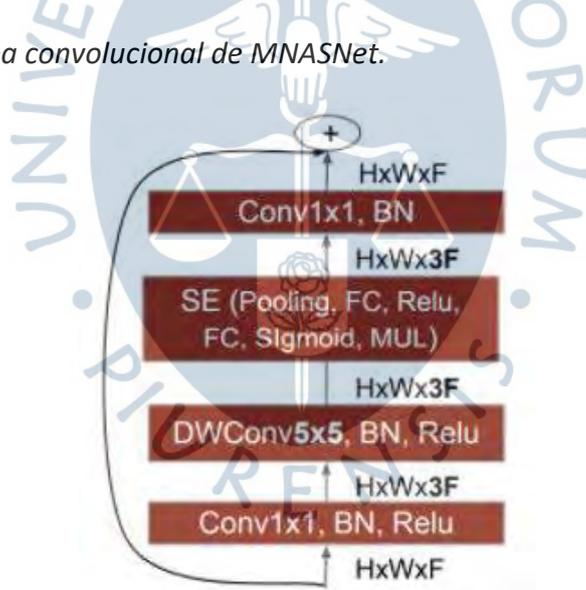


Nota. Elaboración propia.

### 2.2.10 MNASNet

CNN similar a MobileNet, ya que se diseñó y optimizó para su uso en dispositivos móviles, para ello intenta equilibrar los aspectos de precisión y latencia (Tan et al., 2019). En la **Figura 106**, se observa la configuración de una capa convolucional con un kernel de tamaño 5x5.

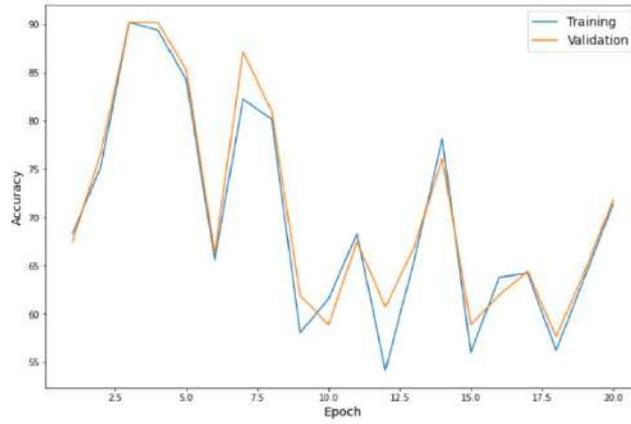
**Figura 106.** Capa convolucional de MNASNet.



Tomado de "Mnasnet: Platform-aware neural architecture search for mobile" (Tan et al., 2019).

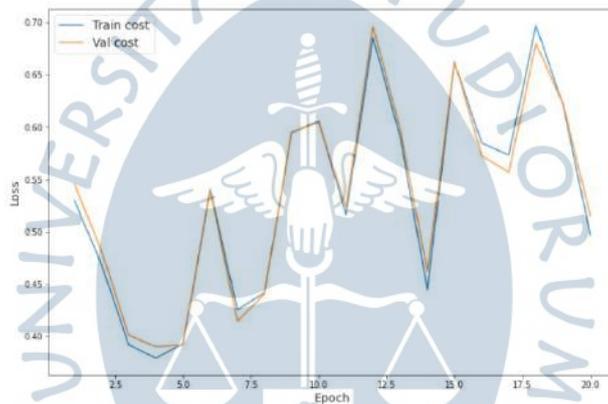
Luego de entrenar el modelo, se obtuvieron exactitudes de 71.41%, 69.93% y 79.14% en el entrenamiento, validación y pruebas respectivamente, con un F1-score de 85.91%, y un tiempo de ejecución de 6.34 minutos. La **Figura 107** muestra el resultado de la precisión en el entrenamiento y validación, la **Figura 108** la tasa de pérdida, la **Figura 109** la matriz de confusión con los porcentajes de cada una de las clases con los datos de prueba, y la **Figura 110** la curva ROC en la que se aprecia un valor AUC de 0.86.

**Figura 107.** MNASNet: Accuracy en Entrenamiento y Validación.



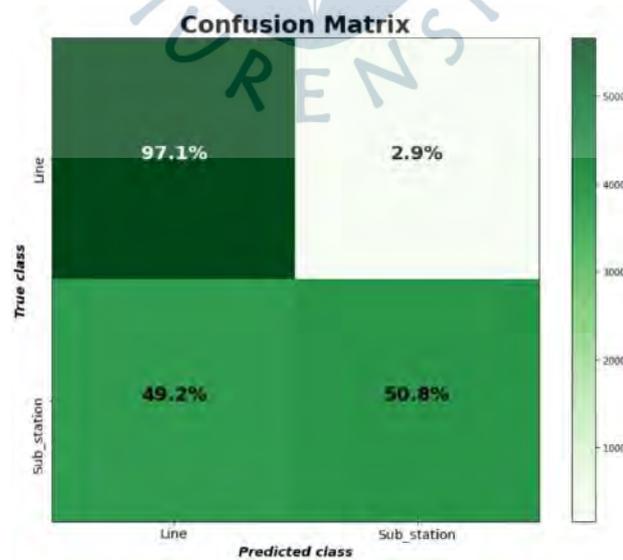
Nota. Elaboración propia.

**Figura 108.** MNASNet: Tasa de pérdida en Entrenamiento y Validación.



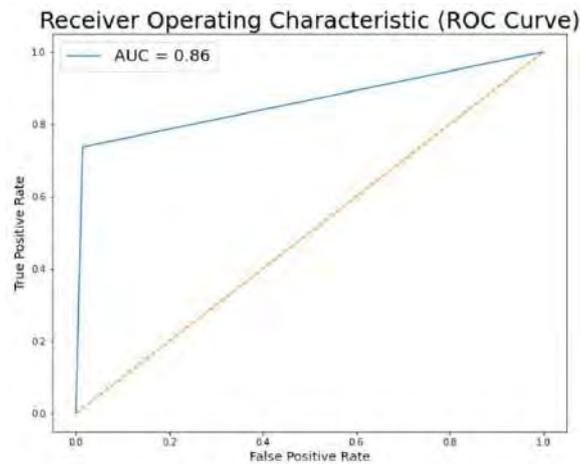
Nota. Elaboración propia.

**Figura 109.** MNASNet: Matriz de confusión.



Nota. Elaboración propia.

**Figura 110.** MNASNet: Curva ROC.

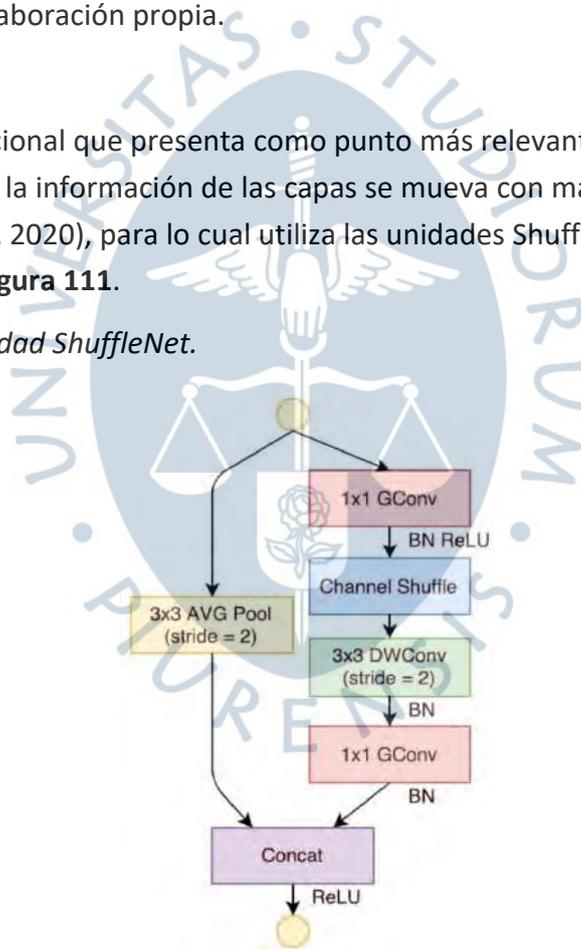


Nota. Elaboración propia.

### 2.2.11 ShuffleNet

Red neuronal convolucional que presenta como punto más relevante el reorganizar los canales, buscando que la información de las capas se mueva con mayor facilidad en cada uno de ellos (Y. Li & Lv, 2020), para lo cual utiliza las unidades ShuffleNet; un ejemplo de esta se aprecia en la **Figura 111**.

**Figura 111.** Unidad ShuffleNet.

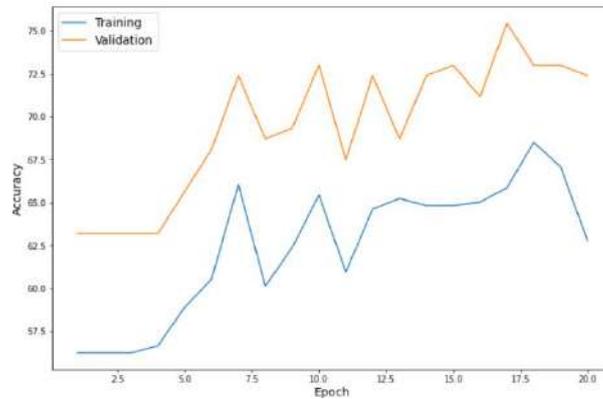


Tomado de “SS-YOLO: An Object Detection Algorithm based on YOLOv3 and ShuffleNet” (Y. Li & Lv, 2020).

Luego de entrenar el modelo, se obtuvieron exactitudes de 62.78%, 68.09% y 76.68% en el entrenamiento, validación y pruebas respectivamente, con un F1-score de 36.86%, y un tiempo de ejecución de 5.83 minutos. La **Figura 112** muestra el resultado de la precisión en el entrenamiento y validación, la **Figura 113** la tasa de pérdida, la **Figura 114** la matriz de confusión con los porcentajes de cada una de las clases con los datos de prueba, y la **Figura**

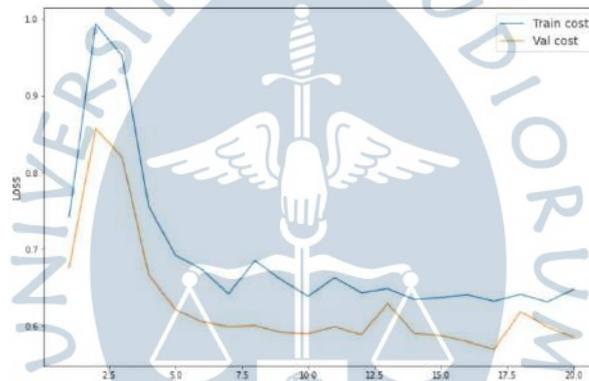
115 la curva ROC en la que se aprecia un valor AUC de 0.50.

**Figura 112.** *ShuffleNet: Accuracy en Entrenamiento y Validación.*



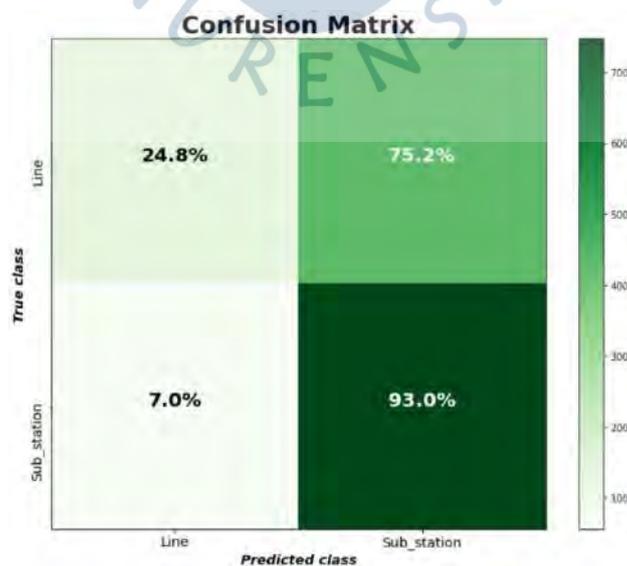
Nota. Elaboración propia.

**Figura 113.** *ShuffleNet: Tasa de pérdida en Entrenamiento y Validación.*



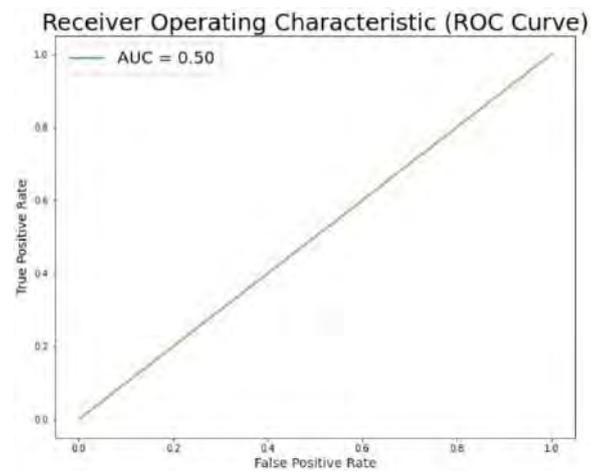
Nota. Elaboración propia.

**Figura 114.** *ShuffleNet: Matriz de confusión.*



Nota. Elaboración propia.

Figura 115. *ShuffleNet*: Curva ROC.



Nota. Elaboración propia.





## Capítulo 4 Resultados

Todas las arquitecturas descritas en el capítulo anterior fueron incluidas en la arquitectura base, utilizándose el optimizador Adam y la función de costo Multi Margin Loss. Las exactitudes de las once arquitecturas fueron plasmadas en la Tabla 5, las tasas de pérdida en la Tabla 6, mientras que en la Tabla 7 se muestra la comparativa de cada una de ellas, mediante el uso de la técnica F1-score (1), la cual se calcula en base a (2) y (3).

$$F1_{score} = 2 * \frac{precision * recall}{precision + recall} \quad (1)$$

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (2)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (3)$$

**Tabla 5.** Porcentaje de exactitud cada arquitectura.

Arquitectura	Entrenamiento	Validación	Pruebas
DenseNet	96.52	92.02	98.15
VGG19	93.66	90.18	96.31
Wide ResNetx	94.68	90.18	96.31
MobileNet v2	94.68	88.95	95.70
VGG16	95.91	91.41	95.09
ResNeXt	94.06	92.02	95.09
ResNet	93.66	84.66	94.47
AlexNet	95.50	91.41	93.86

Arquitectura	Entrenamiento	Validación	Pruebas
GoogLeNet	95.09	88.95	93.86
MNASNet	71.41	69.93	79.14
ShuffleNet	62.78	68.09	76.68

Nota. Elaboración propia.

**Tabla 6.** Tasa de pérdida cada arquitectura.

Arquitectura	Entrenamiento	Validación	Pruebas
VGG16	0.130571	0.195825	0.106733
VGG19	0.175259	0.257767	0.124051
MobileNet v2	0.179914	0.283927	0.124497
DenseNet	0.134446	0.198757	0.126488
AlexNet	0.127838	0.226890	0.134585
GoogLeNet	0.205683	0.242438	0.164706
ResNet	0.220600	0.295843	0.175979
ResNeXt	0.215153	0.241967	0.179627
Wide ResNetx	0.246464	0.271462	0.19046
MNASNet	0.507049	0.529690	0.395779
ShuffleNet	0.647315	0.579723	0.524782

Nota. Elaboración propia.

**Tabla 7.** Comparación de arquitecturas: F1-score.

Arquitectura	Precision	Recall	F1-score
VGG16	96.12	94.12	95.11
ResNeXt	92.86	93.79	93.32

Arquitectura	Precision	Recall	F1-score
MobileNet v2	93.62	92.54	93.08
ResNet	92.36	93.48	92.92
VGG19	92.25	92.85	92.55
DenseNet	93.48	91.59	92.53
Wide ResNetx	93.13	91.8	92.46
AlexNet	93.03	89.91	91.44
GoogLeNet	92.11	89.39	90.73
MNASNet	85.71	86.11	85.91
ShuffleNet	29.14	50.00	36.82

Nota. Elaboración propia.

En estas tablas resumen se aprecia que VGG16 es la arquitectura con la tasa de pérdida más baja, mientras que DenseNet es la que obtiene el porcentaje exactitud más elevado. Al analizar los resultados de F1-score se obtiene un valor de F1-score de 95.11%, el cual corresponde a la arquitectura VGG16. Es importante destacar que ShuffleNet obtuvo los peores resultados debido a que fue la única que red en la que no se usó transfer learning. Además de los resultados que evidencian que VGG16 es la arquitectura con mejores prestaciones, la **Tabla 8**, fue elaborada como complemento al análisis, con la finalidad de ver los tiempos de entrenamiento y las exactitudes obtenidas en el entrenamiento de cada categoría.

**Tabla 8.** Duración y exactitud de los entrenamientos por clase.

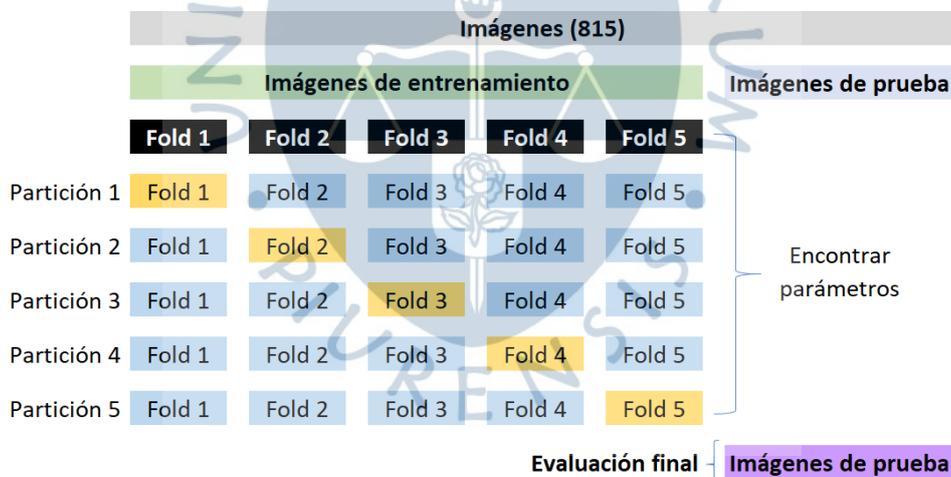
Arquitectura	Línea	Subestación	Tiempo (minutos)
MobileNet v2	89.96	93.77	5.65
ShuffleNet	24.75	93.00	5.83
GoogLeNet	86.99	91.90	6.02
AlexNet	91.06	94.16	6.30
MNASNet	97.14	50.81	6.34

Arquitectura	Línea	Subestación	Tiempo (minutos)
DenseNet	90.08	96.21	6.67
VGG16	93.43	92.81	12.48
ResNet	88.37	92.38	13.58
VGG19	93.19	90.91	14.50
Wide ResNetx	87.99	88.57	20.03
ResNeXt	91.13	93.47	20.21

Nota. Elaboración propia.

Hasta este punto, se utilizó la técnica de separación hold-out (60-20-20), sin embargo, esta suele presentar un grado de variabilidad (Amat Rodrigo, 2020), por lo que, se decidió entrenar nuevamente los modelos realizando validación cruzada, considerando 5-folds, tal como se aprecia en la **Figura 116**.

**Figura 116.** Validación cruzada con 5-folds.



Nota. Elaboración propia.

Estos resultados se plasmaron en la **Tabla 9**, y tal como se sugiere en (Pacheco et al., 2016), se debe realizar una comparación y análisis estadístico de los mismos. Es por ello que se realizó el análisis de residuos, el cual mostró que una prueba paramétrica sería adecuada ya que los residuos se distribuyen en una dispersión similar (ver **Figura 117**). A pesar de esto, al analizar la probabilidad normal con la finalidad de verificar que estos se aproximen a una distribución normal, se obtuvo que ciertos datos se ubican fuera del intervalo de confianza, para ello se consideró un p-value de cero y un coeficiente de Shapiro–Wilk de 0.932 (ver

**Figura 118**). Por lo que, se transformaron los datos mediante el uso de la raíz cuadrada,

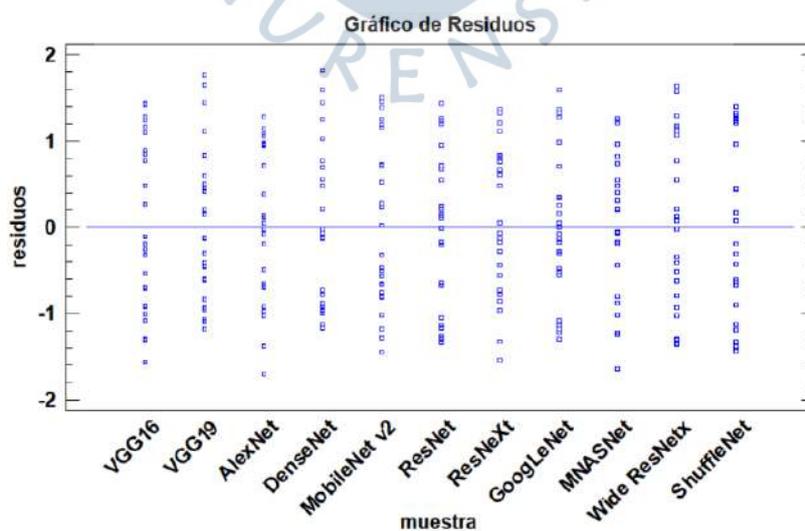
sin embargo, los datos siguieron ubicándose fuera del intervalo de confianza.

**Tabla 9.** Validación cruzada - resultados.

Arquitectura	Fold 1	...	Fold 5	Promedio
VGG16	93.87	...	98.16	96.81
VGG19	93.87	...	98.77	95.83
AlexNet	90.18	...	99.38	95.34
ResNeXt	94.48	...	98.16	94.6
DenseNet	88.34	...	98.16	94.36
MobileNet v2	90.18	...	96.32	93.99
ResNet	93.87	...	95.09	93.74
GoogLeNet	88.96	...	93.86	93.5
MNASNet	86.51	...	98.15	92.52
Wide ResNetx	84.66	...	94.48	89.82
ShuffleNet	68.09	...	72.39	70.06

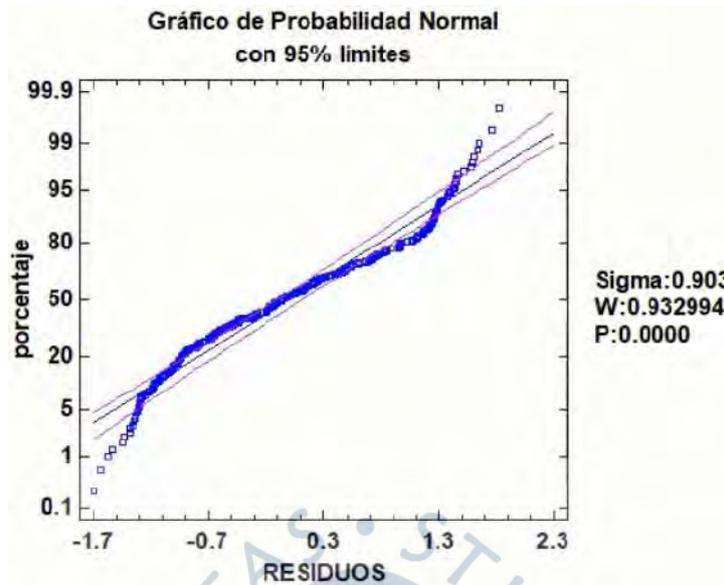
Nota. Elaboración propia.

**Figura 117.** Gráfico de residuos.



Nota. Elaboración propia.

Figura 118. Gráfico de probabilidad normal.



Nota. Elaboración propia.

Entonces, como las poblaciones analizadas no se aproximan a una distribución normal, la aplicación de una prueba paramétrica no es suficiente, por lo que se utilizó una prueba no paramétrica. En este caso, se consideró el Test de Friedman, el cual no necesita que se cumpla el principio de homocedasticidad o normalidad mediante varianzas iguales. La **Tabla 10** muestra los resultados de este test, con un p-value igual a cero, afirmándose que existen diferencias en las poblaciones analizadas, evidenciándose que VGG16 presenta un rendimiento superior a las otras 10 arquitecturas.

**Tabla 10.** Resultados estadísticos – Test de Friedman.

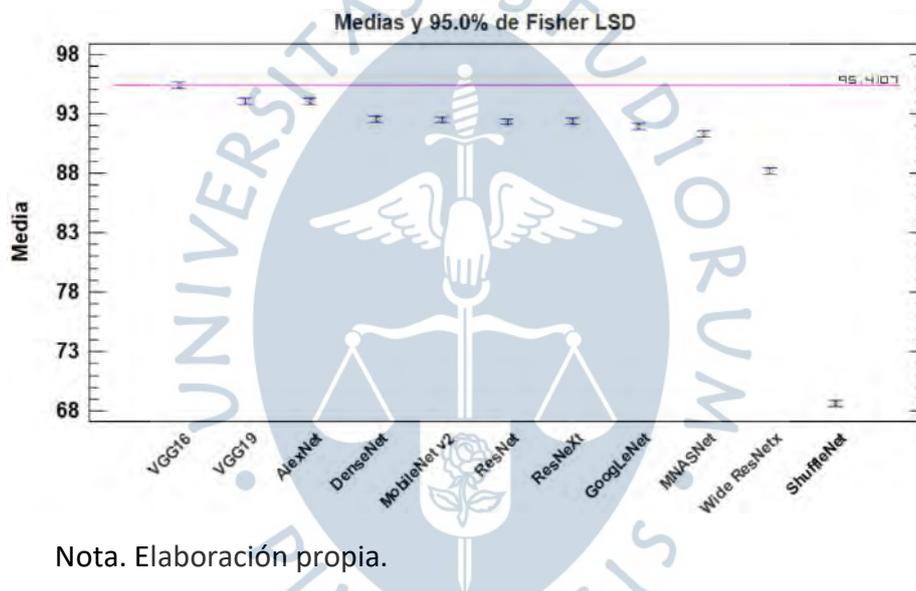
Ranking	Arquitectura	Rango promedio
1	VGG16	10.68
2	AlexNet	9.28
3	VGG19	9.28
4	DenseNet	6.48
5	MobileNet v2	6.32
6	ResNeXt	6.06
7	ResNet	5.78
8	GoogLeNet	5.16

Ranking	Arquitectura	Rango promedio
9	MNASNet	3.96
10	Wide ResNetx	2.00
11	ShuffleNet	1.00

Nota. Elaboración propia.

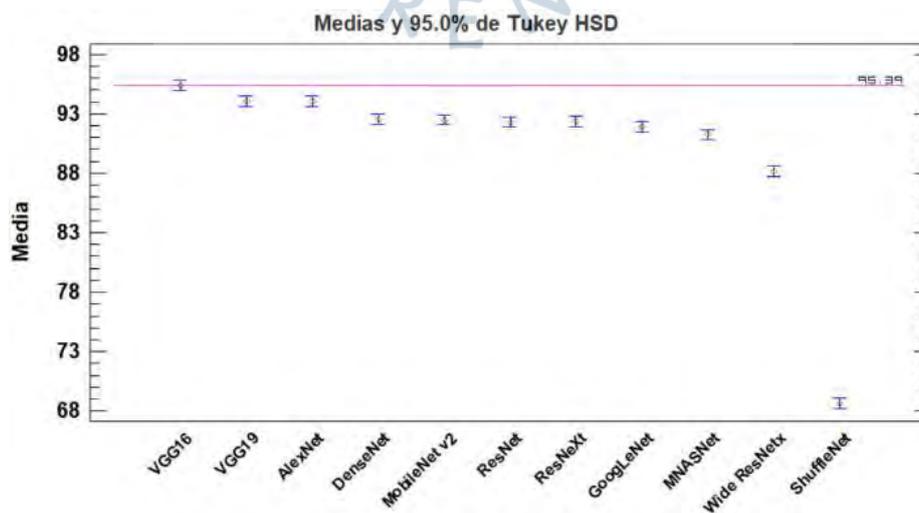
Al analizar las medias, se obtuvieron resultados similares, en la **Figura 119** se trabajó con el método Fisher LSD, mientras que en la **Figura 120** con Tukey HSD. En ambos casos VGG16 presenta un rendimiento más elevado en comparación las demás arquitecturas.

**Figura 119.** Gráfico de medias – Fisher LSD.



Nota. Elaboración propia.

**Figura 120.** Gráfico de medias – Tukey HSD.



Nota. Elaboración propia.

Resultados similares se obtienen al analizar las medianas (**Figura 121**), en la que VGG16 se encuentra por encima de las demás arquitecturas.

**Figura 121.** Gráfico de medianas.



Nota. Elaboración propia.

Finalmente, se ejecutó el análisis de comparaciones múltiples, a través de la prueba con múltiples rangos, tal como se aprecia en la **Tabla 11**. Este resultado muestra que VGG16 es diferente y superior a las otras 10 arquitecturas analizadas.

**Tabla 11.** Prueba de múltiples rangos.

Arquitectura	Media	Grupos homogéneos
ShuffleNet	68.6560	X
Wide ResNetx	88.1820	X
MNASNet	91.2540	X
GoogLeNet	91.9068	XX
ResNet	92.3080	X
ResNeXt	92.3704	X
MobileNet v2	92.4816	X
DenseNet	92.5336	X

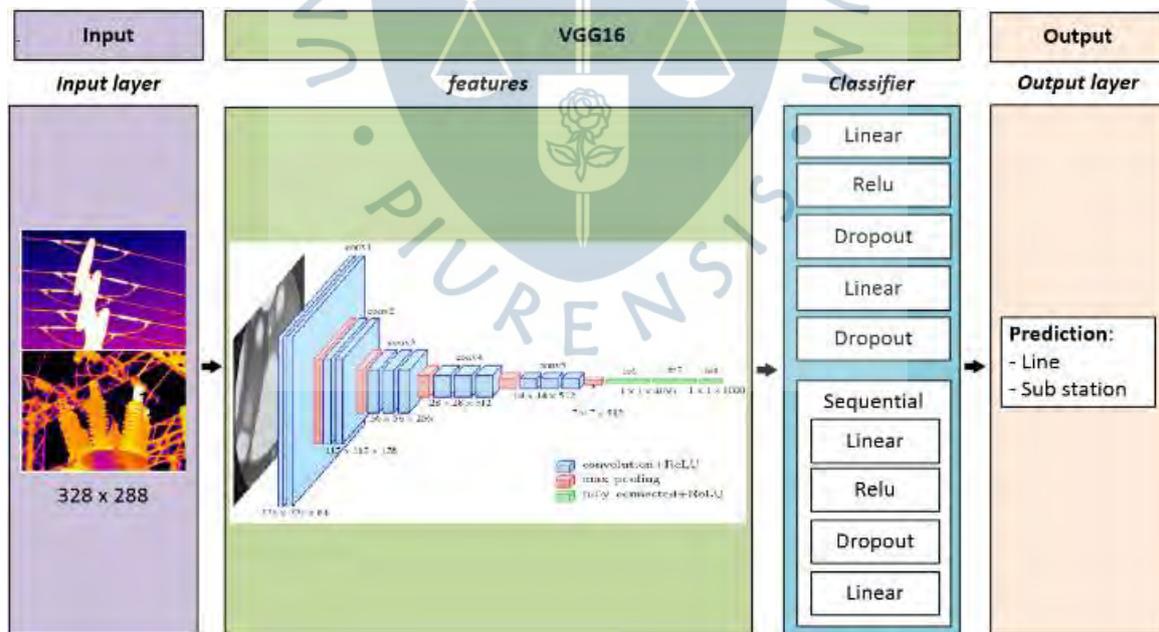
Arquitectura	Media	Grupos homogéneos
AlexNet	94.0500	X
VGG19	94.0652	X
VGG16	95.3756	X

Nota. Elaboración propia.

La arquitectura original de VGG16 se detalla en la **Figura 66**, por lo que, acorde a los resultados obtenidos, el diseño final de la arquitectura de clasificación binaria de imágenes térmicas se define tal como se aprecia en la **Figura 122**, teniendo como entrada al set de datos de imágenes, la cuales pasan por la CNN previamente entrenada VGG16, y finalmente, la salida corresponde a la clasificación de las clases o categorías analizadas (líneas y subestaciones eléctricas).

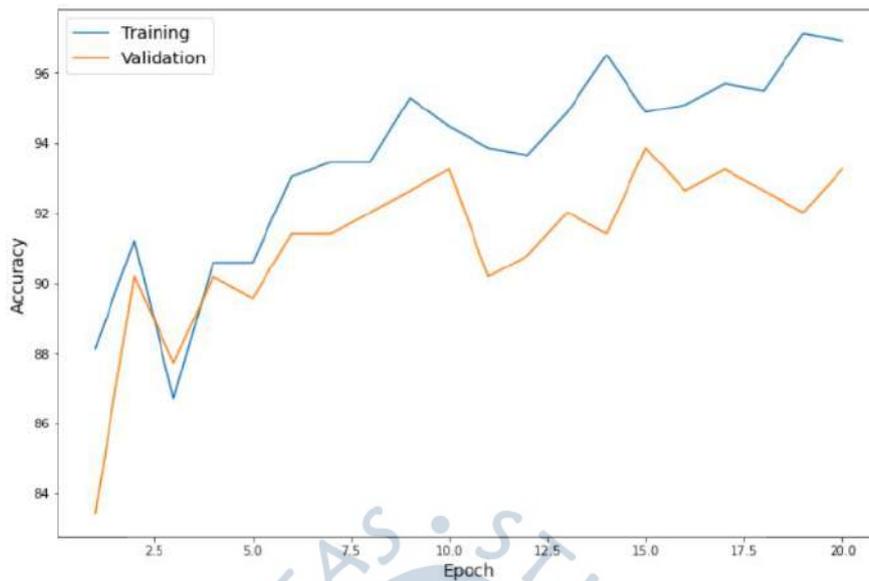
Este modelo obtuvo exactitudes de 95.91% en el entrenamiento y 91.41% en la validación (**Figura 123**) posteriormente se realizaron pruebas con un set de datos que el modelo nunca analizó, obteniendo una exactitud del 94.43% para Líneas y 92.81% para subestaciones eléctricas (ver Matriz de confusión de la **Figura 124**).

**Figura 122.** Propuesta de arquitectura de transfer learning.



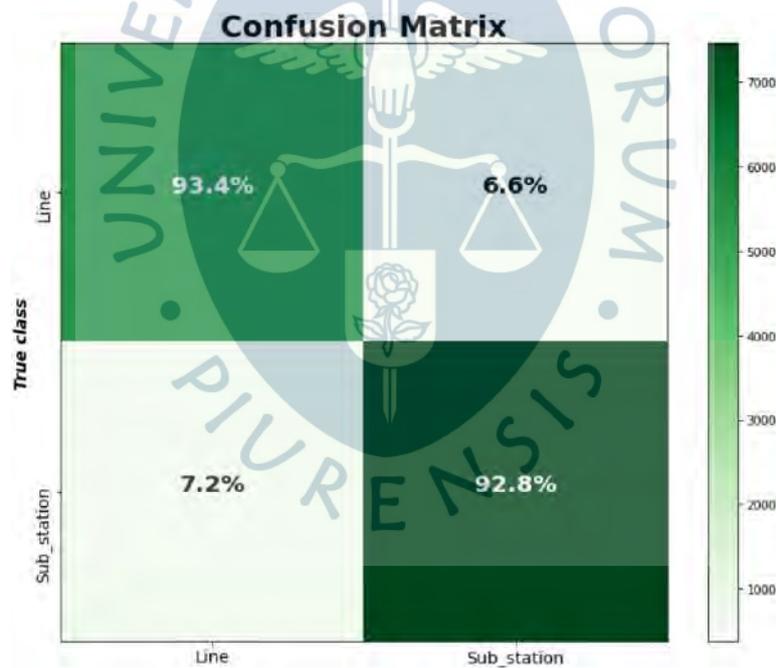
Nota. Elaboración propia.

**Figura 123.** Accuracy: entrenamiento y validación del modelo.



Nota. Elaboración propia.

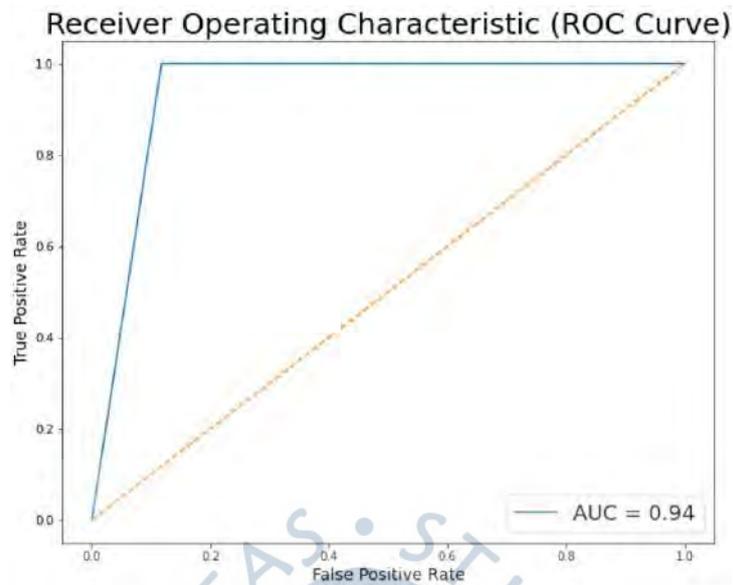
**Figura 124.** Matriz de confusión.



Nota. Elaboración propia.

Adicionalmente se graficó la exactitud del modelo con VGG16 mediante la curva ROC, en la que se aprecia la AUC o área bajo la curva corresponde a un valor de 94%, evidenciando un alto rendimiento del modelo propuesto (ver **Figura 125**)

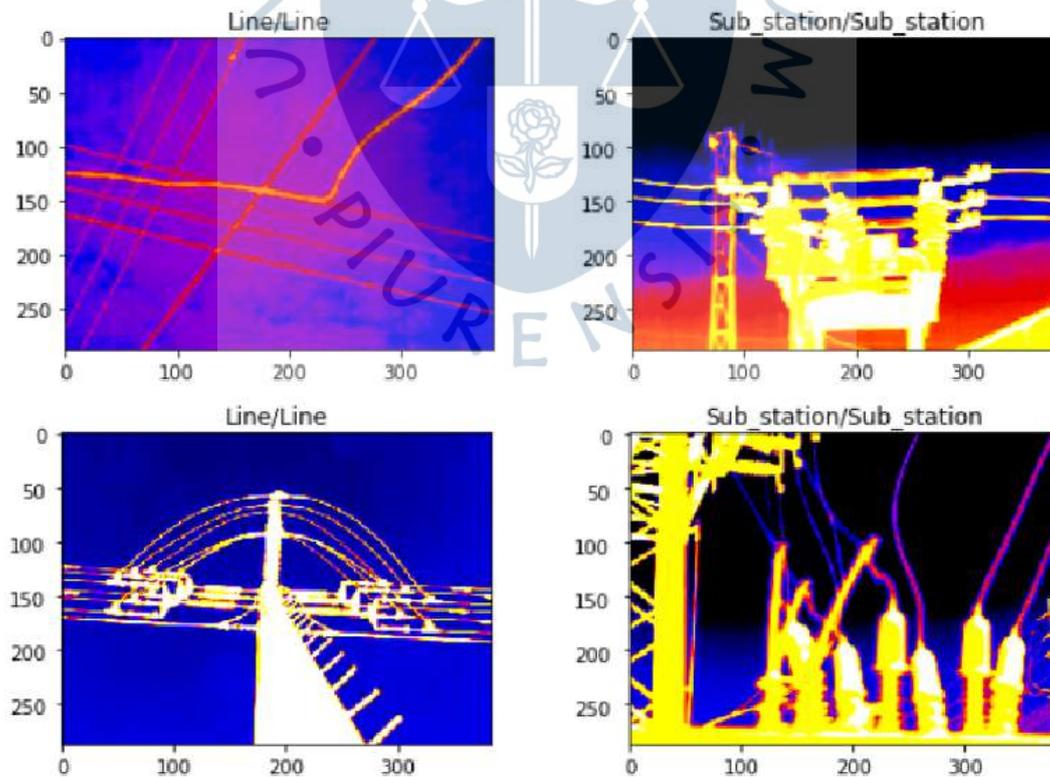
**Figura 125.** Curva ROC.



Nota. Elaboración propia.

Fruto del entrenamiento, validación y pruebas del modelo, en la Figura 126 se plasman algunas predicciones reales del mismo. Cada imagen tiene un título, la parte de la izquierda corresponde al valor real, y el de la derecha a la predicción del modelo.

**Figura 126.** Predicciones del modelo con VGG16.



Nota. Elaboración propia.



## Conclusiones

### a. Conclusiones

El objetivo principal de la investigación fue diseñar un clasificador para el mantenimiento predictivo de una red de distribución eléctrica utilizando Deep Learning, es por ello que se entrenaron, validaron y probaron 11 arquitecturas de transfer learning, con la finalidad de realizar una clasificación binaria de imágenes térmicas de líneas y subestaciones eléctricas. VGG16 presentó resultados superiores frente a las 10 arquitecturas restantes, mostrando una exactitud del 95.09% y un valor de F1-score igual a 95.11%.

La comparación de las 11 arquitecturas se realizó dividiendo el set de datos mediante la técnica tradicional hold-out (60-20-20), sin embargo, con la finalidad de eliminar posibles fuentes de variabilidad, los modelos fueron entrenados nuevamente y analizados mediante la aplicación y análisis del Test de Friedman, el cual confirmó que VGG16 fue superior a sus contrincantes, logrando aprender adecuadamente de los datos de entrada y generalizar los comportamientos frente a la clasificación de nuevas imágenes térmicas.

Con la solución planteada en la presente tesis, se diseñó un clasificador de imágenes automatizado mediante el uso de técnicas de visión por computador que facilita el procesamiento de los set de datos a analizar, y contribuye en la reducción de tiempo al encontrar posibles fallos en los equipos, además del consecuente mejoramiento de los indicadores de calidad del suministro eléctrico, debido a que se pueden realizar programaciones e inspecciones de los puntos de falla con la finalidad de determinar los materiales, herramientas y personal necesarios para corregirlos, minimizar el tiempo de interrupción y evitar paradas, desconexiones o interrupciones innecesarias de la fuente de alimentación.

### b. Conclusiones: aporte adicional (ver Apéndice 01)

En relación a la detección de puntos calientes, las técnicas tradicionales más utilizadas son K-means clustering, Support vector machine (SVM), Multilayer Perceptron (MLP) y Fuzzy system; sin embargo, arquitecturas como YOLO o R-CNN no han sido aplicadas con el rigor necesario en la detección de puntos calientes en equipos de subestaciones eléctricas, más sí en estaciones fotovoltaicas.

El etiquetado manual de puntos calientes fue realizado por un experto en el área mediante la herramienta Labellmg, sin embargo, se pasó de tener 815 imágenes a 138 (116 subestaciones y 22 líneas), ya que solo estas presentaban puntos calientes, por lo que se procedió a limitar el estudio a subestaciones eléctricas.

Los resultados de la aplicación de YOLO en su versión 4 muestran un valor de F1-score superior al 68% en la detección de puntos calientes en las subestaciones eléctricas con un set de datos de imágenes térmicas personalizado, sin embargo, el entrenamiento es sumamente lento.

YOLOv5 small arroja resultados similares, pero con una considerable reducción de la cantidad de tiempo de ejecución y número de épocas, obteniendo valores de F1-score de 68.23% y un mAP de 69.42% en las primeras 16 pruebas con 200 épocas y un tamaño de batch de 16, sin data augmentation, y con la aplicación de transfer learning.

Se realizaron 16 nuevos experimentos con la versión medium de YOLOv5, obteniendo un mAP de 81.99% y un valor de F1-score de 78.57% (200 épocas, tamaño de batch de 8, sin data augmentation, y con la aplicación de transfer learning), los cuales son claramente superiores a los resultados de la versión small. En relación a los 32 experimentos realizados con YOLOv5l y YOLOv5x, se obtuvieron resultados similares a los obtenidos con YOLOv5m.

Los 64 experimentos realizados con YOLOv5 muestran una tendencia a incrementar su precisión si se integran versiones más robustas y complejas del algoritmo, tales como YOLOv5n6, YOLOv5s6, YOLOv5m6, YOLOv5l6, YOLOv5x6, etc.

**Figura 127.** Resumen de resultados.



Nota. Elaboración propia.

c. Conclusión final

Ambas conclusiones son un claro indicio de que la clasificación de imágenes y detección de puntos calientes en equipos eléctricos mediante el uso imágenes térmicas y técnicas de visión por computador, contribuyen a descubrir fallos de manera temprana; con lo cual, las empresas de distribución de energía pueden analizar grandes sets de datos sin la necesidad de tener contacto directo con los equipos, evitar paradas de planta innecesarias, y reducir la tasa de error humano que este tipo de observaciones trate consigo. El resumen de la investigación se aprecia en la Figura 128.

d. Trabajos futuros

Trabajos futuros de esta investigación deben contemplar los siguientes aspectos:

- Contar con una base de datos de imágenes térmicas más grande.
- Variar hiperparámetros distintos a los considerados en esta investigación con la finalidad de obtener precisiones más altas.
- Realizar nuevos entrenamientos con otras arquitecturas de clasificación de imágenes.
- Realizar nuevos entrenamientos con otros algoritmos de detección de objetos.
- Integrar los modelos dentro de una cámara infrarroja con la finalidad de tener resultados en tiempo real. Esta alternativa puede ser considerada como proyecto patentable.





## Glosario

- Mantenimiento predictivo: conjunto de tareas o técnicas ejecutadas con la finalidad de detectar posibles fallos o defectos en la maquinaria, principalmente en temas industriales; permite conocer de antemano y en etapas tempranas, problemas que a futuro pueden provocar grandes pérdidas de tiempo y dinero debido a los tiempos muertos o paradas de emergencia (Kanawaday & Sane, 2018).
- Líneas eléctricas: medios físicos a través de los cuales se efectúa tanto la transmisión, así como la distribución de la energía eléctrica (Ebrahimi et al., 2018).
- Subestaciones eléctricas: equipos que tienen por finalidad la realización de transformaciones de frecuencia, tensión, número de fases, o conexión de circuitos, por lo que, generalmente se ubican cerca de las centrales generadores de energía o en zonas no urbanas (Iglesias-Urkiá et al., 2019).
- Termografía: Técnica utilizada para la determinación de temperatura de distintos objetos sin la necesidad de tener contacto físico con los mismos; para ello capta la radiación infrarroja del espectro electromagnético mediante el uso de cámaras térmicas (Wang et al., 2018).
- Inteligencia Artificial (IA): combinación de algoritmos utilizados para la creación de máquinas que simulen las capacidades del cerebro humano (Flasiński & Flasiński, 2016).
- Aprendizaje automático (ML): está incluido dentro de la Inteligencia artificial, buscado una generalización del comportamiento de los conjuntos de datos de entrada, o visto de otra manera, busca obtener predicciones de comportamientos, basados en el análisis de patrones en grandes conjuntos de datos (Carvalho et al., 2019).
- Aprendizaje profundo (DL): forma parte del aprendizaje automático, siendo su objetivo principal que los sistemas informáticos intenten imitar automáticamente los comportamientos y razonamientos de los seres humanos (Wan et al., 2020).
- Redes neuronales artificiales (ANN): elementos que buscan simular el proceso sináptico de las neuronas del cerebro humano (Golodov et al., 2017).
- Redes Neuronales Convolucionales (CNN): ANN mayormente utilizadas en problemas de clasificación de imágenes debido a su complejidad (Brunetti et al., 2018).

- Transfer Learning (TF): está incluido como parte del Deep Learning. Su funcionamiento principal radica en no entrenar una red neuronal desde cero, sino utilizar una red pre entrenada o entrenada previamente con otros set de datos; en otras palabras, busca reusar los pesos, y en algunos casos la arquitectura, de un modelo entrenado con set de datos de entrada de tamaño considerable n(Phan et al., 2020).
- VGG16: CNN integrada por 16 capas, entrenada con la base de datos ImageNet, considera un conjunto de filtros de tamaño 3x3 en lugar de los grandes filtros de cada kernel de AlexNet (S. Liu & Deng, 2016).
- YOLO: abreviatura de “Solo miras una vez”, del inglés “You Only Look Once”, es un algoritmo de detección de objetos que utiliza redes neuronales con la finalidad de realizar detección de objetos en tiempo real; es popular gracias a los niveles de precisión y velocidad que obtiene luego de entrenarse (Du, 2018).



### Referencias bibliográficas

- Agrahari, A. (2020, February 9). *Neural Networks From Scratch*. <https://www.dataninja.in/series/neural-networks-from-scratch/>
- Ahmed, M. M., Huda, A. S. N., & Mat Isa, N. A. (2015). Recursive construction of output-context fuzzy systems for the condition monitoring of electrical hotspots based on infrared thermography. *Engineering Applications of Artificial Intelligence*, 39, 120–131. <https://doi.org/10.1016/J.ENGAPPAI.2014.11.010>
- Almisreb, A. A., Jamil, N., & Din, N. M. (2018). Utilizing AlexNet Deep Transfer Learning for Ear Recognition. *Proceedings - 2018 4th International Conference on Information Retrieval and Knowledge Management: Diving into Data Sciences, CAMP 2018*, 8–12. <https://doi.org/10.1109/INFRKM.2018.8464769>
- Alshalali, T., & Josyula, D. (2018). Fine-tuning of pre-trained deep learning models with extreme learning machine. *Proceedings - 2018 International Conference on Computational Science and Computational Intelligence, CSCI 2018*, 469–473. <https://doi.org/10.1109/CSCI46756.2018.00096>
- Amat Rodrigo, J. (2020, November). *Validación de modelos predictivos (machine learning): Cross-validation, OneLeaveOut, Bootstrapping*. *Cienciadedatos*. [https://www.cienciadedatos.net/documentos/30\\_cross-validation\\_oneleaveout\\_bootstrap](https://www.cienciadedatos.net/documentos/30_cross-validation_oneleaveout_bootstrap)
- ATRIA Innovation. (2019, October 22). *Qué son las redes neuronales y sus funciones*. <https://www.atriainnovation.com/que-son-las-redes-neuronales-y-sus-funciones/>
- Aung, Z., Mikhaylov, I. S., & Aung, Y. T. (2020). Artificial Intelligence Methods Application in Oil Industry. *Proceedings of the 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering, EIConRus 2020*, 563–567. <https://doi.org/10.1109/EIConRus49466.2020.9039330>
- Bagavathiappan, S., Lahiri, B. B., Saravanan, T., Philip, J., & Jayakumar, T. (2013). Infrared thermography for condition monitoring - A review. In *Infrared Physics and Technology* (Vol. 60, pp. 35–55). Pergamon. <https://doi.org/10.1016/j.infrared.2013.03.006>

- Bezares, J. (2019, September 25). *Soluciones punteras en termografía, al servicio de la Industria Automotriz*. <https://bcbingenieria.com/soluciones-punteras-en-termografia-al-servicio-de-la-industria-automotriz/>
- Briese, C., Schlüter, M., Lehr, J., Maurer, K., & Krüger, J. (2020). Towards Deep Learning in Industrial Applications Taking Advantage of Service-Oriented Architectures. *Procedia Manufacturing*, 43, 503–510. <https://doi.org/10.1016/j.promfg.2020.02.182>
- Brunetti, A., Buongiorno, D., Trotta, G. F., & Bevilacqua, V. (2018). Computer vision and deep learning techniques for pedestrian detection and tracking: A survey. *Neurocomputing*, 300, 17–33. <https://doi.org/10.1016/j.neucom.2018.01.092>
- Budhiman, A., Suyanto, S., & Arifianto, A. (2019). Melanoma Cancer Classification Using ResNet with Data Augmentation. *2019 2nd International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2019*, 17–20. <https://doi.org/10.1109/ISRITI48646.2019.9034624>
- Cachada, A., Barbosa, J., Leitão, P., Geraldes, C. A. S., Deusdado, L., Costa, J., Teixeira, C., Teixeira, J., Moreira, A. H. J., Moreira, P. M., & Romero, L. (2018). Maintenance 4.0: Intelligent and Predictive Maintenance System Architecture. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, 2018-September*, 139–146. <https://doi.org/10.1109/ETFA.2018.8502489>
- Carneiro, T., Da Nobrega, R. V. M., Nepomuceno, T., Bian, G. Bin, De Albuquerque, V. H. C., & Filho, P. P. R. (2018). Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications. *IEEE Access*, 6, 61677–61685. <https://doi.org/10.1109/ACCESS.2018.2874767>
- Carvalho, T. P., Soares, F. A. A. M. N., Vita, R., Francisco, R. da P., Basto, J. P., & Alcalá, S. G. S. (2019). A systematic literature review of machine learning methods applied to predictive maintenance. *Computers and Industrial Engineering*, 137, 106024. <https://doi.org/10.1016/j.cie.2019.106024>
- Casaverde, L. A., Risco, R. H., & Perez, D. A. (2021). A simulation study of the performance of GPC and PID applied to a binary distillation column. *2021 IEEE International Conference on Automation/24th Congress of the Chilean Association of Automatic Control, ICA-ACCA 2021*. <https://doi.org/10.1109/ICAACCA51523.2021.9465198>
- Cengil, E., & Cinar, A. (2019, September 1). Multiple classification of flower images using transfer learning. *2019 International Conference on Artificial Intelligence and Data Processing Symposium, IDAP 2019*. <https://doi.org/10.1109/IDAP.2019.8875953>
- Ciampa, F., Mahmoodi, P., Pinto, F., & Meo, M. (2018). Recent Advances in Active Infrared Thermography for Non-Destructive Testing of Aerospace Components. *Sensors*, 18(2), 609. <https://doi.org/10.3390/s18020609>

- Cipriani, G., D'Amico, A., Guarino, S., Manno, D., Traverso, M., & Dio, V. Di. (2020). Convolutional Neural Network for Dust and Hotspot Classification in PV Modules. *Energies 2020, Vol. 13, Page 6357, 13(23)*, 6357. <https://doi.org/10.3390/EN13236357>
- Copeland, M. (2021, March 10). *Diferencia entre AI, Machine Learning y Deep Learning*. <https://la.blogs.nvidia.com/2021/03/10/cual-es-la-diferencia-entre-ai-machine-learning-y-deep-learning/>
- d'Acremont, A., Fablet, R., Baussard, A., & Quin, G. (2019). CNN-Based Target Recognition and Identification for Infrared Imaging in Defense Systems. *Sensors 2019, Vol. 19, Page 2040, 19(9)*, 2040. <https://doi.org/10.3390/S19092040>
- De Benedetti, M., Leonardi, F., Messina, F., Santoro, C., & Vasilakos, A. (2018). Anomaly detection and predictive maintenance for photovoltaic systems. *Neurocomputing, 310*, 59–68. <https://doi.org/10.1016/j.neucom.2018.05.017>
- De máquinas y herramientas. (2015, October 11). *Introducción a las cámaras termográficas*. <https://www.demaquinasyherramientas.com/herramientas-de-medicion/introduccion-a-las-camaras-termograficas>
- Dragomir, A., Adam, M., Andrușcă, M., Munteanu, A., & Boghiu, E. (2017). Considerations regarding infrared thermal stresses monitoring of electrical equipment. *2017 11th International Conference on Electromechanical and Power Systems, SIELMEN 2017 - Proceedings, 2017-January*, 100–103. <https://doi.org/10.1109/SIELMEN.2017.8123307>
- Du, J. (2018). Understanding of Object Detection Based on CNN Family and YOLO. *Journal of Physics: Conference Series, 1004(1)*, 012029. <https://doi.org/10.1088/1742-6596/1004/1/012029>
- Ebrahimi, A., Scott, J., & Ghorbani, K. (2018). Transmission lines terminated with LC resonators for differential permittivity sensing. *IEEE Microwave and Wireless Components Letters, 28(12)*, 1149–1151. <https://doi.org/10.1109/LMWC.2018.2875996>
- Ebrahim, M., Al-Ayyoub, M., & Alsmirat, M. A. (2019). Will Transfer Learning Enhance ImageNet Classification Accuracy Using ImageNet-Pretrained Models? *2019 10th International Conference on Information and Communication Systems, ICICS 2019*, 211–216. <https://doi.org/10.1109/IACS.2019.8809114>
- Fambrini, F., Iano, Y., Caetano, D. G., Rodriguez, A. A. D., Moya, C., Carrara, E., Arthur, R., Cabello, F. C., Zubem, J. Von, Del Val Cura, L. M., Destro Filho, J. B., Campos, J. R., & Saito, J. H. (2018). GPU Cuda JsEG segmentation Algorithm associated with Deep Learning Classifier for Electrical Network Images Identification. *Procedia Computer Science, 126*, 557–565. <https://doi.org/10.1016/j.procs.2018.07.290>
- Fang, T. (2018). A Novel Computer-Aided Lung Cancer Detection Method Based on Transfer Learning from GoogLeNet and Median Intensity Projections. *2018 IEEE International*

*Conference on Computer and Communication Engineering Technology, CCET 2018*, 286–290.  
<https://doi.org/10.1109/CCET.2018.8542189>

Fatima, M., & Pasha, M. (2017). Survey of Machine Learning Algorithms for Disease Diagnostic. *Journal of Intelligent Learning Systems and Applications*, 09(01), 1–16.  
<https://doi.org/10.4236/jilsa.2017.91001>

Flasiński, M., & Flasiński, M. (2016). History of Artificial Intelligence. In *Introduction to Artificial Intelligence* (pp. 3–13). Springer International Publishing. [https://doi.org/10.1007/978-3-319-40022-8\\_1](https://doi.org/10.1007/978-3-319-40022-8_1)

Fu, K., Sun, L., Kang, X., & Ren, F. (2019). Text Detection for Natural Scene based on MobileNet V2 and U-Net. *Proceedings of 2019 IEEE International Conference on Mechatronics and Automation, ICMA 2019*, 1560–1564. <https://doi.org/10.1109/ICMA.2019.8816384>

Galindo, E. A., Perdomo, J. A., & Figueroa-García, J. C. (2020). Comparative study between multiclass support vector machines, neural networks and self-organized neuro-fuzzy inference system for classification problems. *Informacion Tecnologica*, 31(1), 273–286.  
<https://doi.org/10.4067/S0718-07642020000100273>

Galvez, R. L., Dadios, E. P., Bandala, A. A., & Vicerra, R. R. P. (2019, March 12). Threat object classification in x-ray images using transfer learning. *2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management, HNICEM 2018*.  
<https://doi.org/10.1109/HNICEM.2018.8666344>

Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., Martinez-Gonzalez, P., & Garcia-Rodriguez, J. (2018). A survey on deep learning techniques for image and video semantic segmentation. In *Applied Soft Computing Journal* (Vol. 70, pp. 41–65). Elsevier Ltd.  
<https://doi.org/10.1016/j.asoc.2018.05.018>

GeeksforGeeks. (2020, May 3). *Understanding GoogLeNet Model - CNN Architecture*.  
<https://www.geeksforgeeks.org/understanding-googlenet-model-cnn-architecture/>

Ghahramani, A., Castro, G., Karvigh, S. A., & Becerik-Gerber, B. (2018). Towards unsupervised learning of thermal comfort using infrared thermography. *Applied Energy*, 211, 41–49.  
<https://doi.org/10.1016/j.apenergy.2017.11.021>

Golodov, V., Zavei-Boroda, A., Ivanov, S., & Nikolskaya, K. (2017). Development of a deep learning neural network for human movements analysis. *RPC 2017 - Proceedings of the 2nd Russian-Pacific Conference on Computer Technology and Applications, 2017-December*, 72–74.  
<https://doi.org/10.1109/RPC.2017.8168071>

Gong, X., Yao, Q., Wang, M., & Lin, Y. (2018). A Deep Learning Approach for Oriented Electrical Equipment Detection in Thermal Images. *IEEE Access*, 6, 41590–41597.  
<https://doi.org/10.1109/ACCESS.2018.2859048>

- Greco, A., Pironti, C., Saggese, A., Vento, M., & Vigilante, V. (2020). A deep learning based approach for detecting panels in photovoltaic plants. *ACM International Conference Proceeding Series*. <https://doi.org/10.1145/3378184.3378185>
- Guo, X., Liu, X., Zhu, E., & Yin, J. (2017). Deep Clustering with Convolutional Autoencoders. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 10635 LNCS, 373–382. [https://doi.org/10.1007/978-3-319-70096-0\\_39](https://doi.org/10.1007/978-3-319-70096-0_39)
- Haenlein, M., & Kaplan, A. (2019). A Brief History of Artificial Intelligence: On the Past, Present, and Future of Artificial Intelligence. *California Management Review*, 61(4), 5–14. <https://doi.org/10.1177/0008125619864925>
- Han, S., Yang, F., Yang, G., Gao, B., Zhang, N., & Wang, D. (2020). Electrical equipment identification in infrared images based on ROI-selected CNN method. *Electric Power Systems Research*, 188, 106534. <https://doi.org/10.1016/J.EPSR.2020.106534>
- Hitawala, S., & Cheriton, D. R. (2018). *Comparative Study on Generative Adversarial Networks*. <https://doi.org/10.48550/arxiv.1801.04271>
- Huang, M.-H., & Rust, R. T. (2018). Artificial Intelligence in Service. *Journal of Service Research*, 21(2), 155–172. <https://doi.org/10.1177/1094670517752459>
- Huda, A. S. N., & Taib, S. (2013). Suitable features selection for monitoring thermal condition of electrical equipment using infrared thermography. *Infrared Physics and Technology*, 61, 184–191. <https://doi.org/10.1016/j.infrared.2013.04.012>
- Huerta Herraiz, Á., Pliego Marugán, A., & García Márquez, F. P. (2020). Photovoltaic plant condition monitoring using thermal images analysis by convolutional neural network-based structure. *Renewable Energy*, 153, 334–348. <https://doi.org/10.1016/J.RENENE.2020.01.148>
- Iglesias-Urkiá, M., Casado-Mansilla, D., Mayer, S., Bilbao, J., & Urbieta, A. (2019). Integrating Electrical Substations Within the IoT Using IEC 61850, CoAP, and CBOR. *IEEE Internet of Things Journal*, 6(5), 7437–7449. <https://doi.org/10.1109/JIOT.2019.2903344>
- Jadin, M. S., Taib, S., & Ghazali, K. H. (2014). Feature extraction and classification for detecting the thermal faults in electrical installations. *Measurement: Journal of the International Measurement Confederation*, 57, 15–24. <https://doi.org/10.1016/j.measurement.2014.07.010>
- Janssens, O., Van De Walle, R., Locufier, M., & Van Hoecke, S. (2018). Deep Learning for Infrared Thermal Image Based Machine Health Monitoring. *IEEE/ASME Transactions on Mechatronics*, 23(1), 151–159. <https://doi.org/10.1109/TMECH.2017.2722479>
- Jocher Glenn. (2021, October). *YOLOv5 in PyTorch*. <https://github.com/ultralytics/yolov5>

- Johnson, D. (2022, March 8). *Back Propagation Neural Network: What is Backpropagation Algorithm in Machine Learning?* <https://www.guru99.com/backpropogation-neural-network.html>
- Jung, D., Zhang, Z., & Winslett, M. (2017). Vibration analysis for iot enabled predictive maintenance. *Proceedings - International Conference on Data Engineering*, 1271–1282. <https://doi.org/10.1109/ICDE.2017.170>
- Jung, Y. (2018). Multiple predicting K-fold cross-validation for model selection. *Journal of Nonparametric Statistics*, 30(1), 197–215. <https://doi.org/10.1080/10485252.2017.1404598>
- Kadechkar, A., Moreno-Eguilaz, M., Riba, J. R., & Capelli, F. (2019). Low-Cost Online Contact Resistance Measurement of Power Connectors to Ease Predictive Maintenance. *IEEE Transactions on Instrumentation and Measurement*, 68(12), 4825–4833. <https://doi.org/10.1109/TIM.2019.2899481>
- Kanawaday, A., & Sane, A. (2018). Machine learning for predictive maintenance of industrial machines using IoT sensor data. *Proceedings of the IEEE International Conference on Software Engineering and Service Sciences, ICSESS, 2017-November*, 87–90. <https://doi.org/10.1109/ICSESS.2017.8342870>
- Karimi, G. (2021, April 15). *Introduction to YOLO Algorithm for Object Detection*. Section. <https://www.section.io/engineering-education/introduction-to-yolo-algorithm-for-object-detection/>
- Karimi, S., Musilek, P., & Knight, A. M. (2018). Dynamic thermal rating of transmission lines: A review. *Renewable and Sustainable Energy Reviews*, 91, 600–612. <https://doi.org/10.1016/j.rser.2018.04.001>
- Kaur, T., & Gandhi, T. K. (2019). Automated brain image classification based on VGG-16 and transfer learning. *Proceedings - 2019 International Conference on Information Technology, ICIT 2019*, 94–98. <https://doi.org/10.1109/ICIT48102.2019.00023>
- Kim, J., & Cho, J. (2020). Exploring a Multimodal Mixture-Of-YOLOs Framework for Advanced Real-Time Object Detection. *Applied Sciences 2020, Vol. 10, Page 612, 10(2)*, 612. <https://doi.org/10.3390/APP10020612>
- Kone, C. (2019, November 5). *Introducing Convolutional Neural Networks in Deep Learning*. <https://towardsdatascience.com/introducing-convolutional-neural-networks-in-deep-learning-400f9c3ad5e9>
- Kornatka, M. (2017, June 30). Distribution of SAIDI and SAIFI indices and the saturation of the MV network with remotely controlled switches. *Proceedings of the 2017 18th International Scientific Conference on Electric Power Engineering, EPE 2017*. <https://doi.org/10.1109/EPE.2017.7967243>

- LastNameSeguí, P. (2021, November). *Uso y aplicaciones de la termografía*. <https://ovacen.com/uso-y-aplicaciones-de-la-termografia/>
- Lee, S. Y., & Teoh, S. S. (2019). A Survey on Infrared Thermography Based Automatic Electrical Fault Diagnosis Techniques. *Lecture Notes in Electrical Engineering*, 547, 537–542. [https://doi.org/10.1007/978-981-13-6447-1\\_68](https://doi.org/10.1007/978-981-13-6447-1_68)
- Lile, C., & Yiqun, L. (2018). Anomaly detection in thermal images using deep neural networks. *Proceedings - International Conference on Image Processing, ICIP, 2017-September*, 2299–2303. <https://doi.org/10.1109/ICIP.2017.8296692>
- Lima, G. R., Member, T. T. B., Baptista, J. M., Monteiro, P. R. D., & Lima, L. (2018). Virtual reality as a tool for electrical substations teaching. *SBSE 2018 - 7th Brazilian Electrical Systems Symposium*, 1–6. <https://doi.org/10.1109/SBSE.2018.8395776>
- Liu, Q., Dong, M., Chen, F. F., Lv, W., & Ye, C. (2019). Single-machine-based joint optimization of predictive maintenance planning and production scheduling. *Robotics and Computer-Integrated Manufacturing*, 55, 173–182. <https://doi.org/10.1016/j.rcim.2018.09.007>
- Liu, S., & Deng, W. (2016). Very deep convolutional neural network based image classification using small training sample size. *Proceedings - 3rd IAPR Asian Conference on Pattern Recognition, ACPR 2015*, 730–734. <https://doi.org/10.1109/ACPR.2015.7486599>
- Liu, Y., & Wang, S. (2021). A quantitative detection algorithm based on improved faster R-CNN for marine benthos. *Ecological Informatics*, 61, 101228. <https://doi.org/10.1016/J.ECOINF.2021.101228>
- Li, X. (2019). Design of Infrared Anomaly Detection for Power Equipment Based on YOLOv3. *2019 3rd IEEE Conference on Energy Internet and Energy System Integration: Ubiquitous Energy Network Connecting Everything, EI2 2019*, 2291–2294. <https://doi.org/10.1109/EI247390.2019.9061852>
- Li, Y., & Lv, C. (2020). SS-YOLO: An Object Detection Algorithm based on YOLOv3 and ShuffleNet. *Proceedings of 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference, ITNEC 2020*, 769–772. <https://doi.org/10.1109/ITNEC48623.2020.9085091>
- Lucchi, E. (2018). Applications of the infrared thermography in the energy audit of buildings: A review. In *Renewable and Sustainable Energy Reviews* (Vol. 82, pp. 3077–3090). Elsevier Ltd. <https://doi.org/10.1016/j.rser.2017.10.031>
- Marinescu, A. D., Cristescu, C., Popescu, T. C., & Safta, C. A. (2017). Assessing the Opportunity to Use the Infrared Thermography Method for Predictive Maintenance of Hydrostatic Pumps. *Proceedings of 8th International Conference on Energy and Environment: Energy Saved Today Is Asset for Future, CIEM 2017*, 270–274. <https://doi.org/10.1109/CIEM.2017.8120790>

- Nazmul Huda, A. S., Taib, S., Jadin, M. S., & Ishak, D. (2012). A semi-automatic approach for thermographic inspection of electrical installations within buildings. *Energy and Buildings*, *55*, 585–591. <https://doi.org/10.1016/j.enbuild.2012.09.014>
- Netzelmann, U., Walle, G., Lugin, S., Ehlen, A., Bessert, S., & Valeske, B. (2016). Induction thermography: principle, applications and first steps towards standardisation. *Quantitative InfraRed Thermography Journal*, *13*(2), 170–181. <https://doi.org/10.1080/17686733.2016.1145842>
- Ngo, G. C., & Macabebe, E. Q. B. (2017). Image segmentation using K-means color quantization and density-based spatial clustering of applications with noise (DBSCAN) for hotspot detection in photovoltaic modules. *IEEE Region 10 Annual International Conference, Proceedings/TENCON*, 1614–1618. <https://doi.org/10.1109/TENCON.2016.7848290>
- Nguyen, D. T., Nguyen, T. N., Kim, H., & Lee, H. J. (2019). A high-throughput and power-efficient fpga implementation of yolo cnn for object detection. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *27*(8), 1861–1873. <https://doi.org/10.1109/TVLSI.2019.2905242>
- Niazi, K. A. K., Akhtar, W., Khan, H. A., Yang, Y., & Athar, S. (2019). Hotspot diagnosis for solar photovoltaic modules using a Naive Bayes classifier. *Solar Energy*, *190*, 34–43. <https://doi.org/10.1016/j.solener.2019.07.063>
- Oppermann, A. (2018, April 27). *Deep Learning meets Physics: Restricted Boltzmann Machines*. <https://towardsdatascience.com/deep-learning-meets-physics-restricted-boltzmann-machines-part-i-6df5c4918c15>
- Osinermin. (2019). SAIDI y SAIFI por departamentos. <https://observatorio.osinermin.gob.pe/saidi-saifi-departamentos>
- Pacheco, F., Valente de Oliveira, J., Sánchez, R. V., Cerrada, M., Cabrera, D., Li, C., Zurita, G., & Artés, M. (2016). A statistical comparison of neuroclassifiers and feature selection methods for gearbox fault diagnosis under realistic conditions. *Neurocomputing*, *194*, 192–206. <https://doi.org/10.1016/j.neucom.2016.02.028>
- Pérez-Aguilar, D. A., Risco-Ramos, R. H., & Casaverde-Pacherrez, L. (2021). Transfer learning for binary classification of thermal images. *Ingenius*, *26*, 71–86. <https://doi.org/10.17163/INGS.N26.2021.07>
- Perez, D., Risco, R., & Casaverde, L. (2020). Analysis of the implementation of Blockchain as a mechanism for digital and transparent food traceability in Peruvian social programs. *Proceedings of the 2020 IEEE 27th International Conference on Electronics, Electrical Engineering and Computing, INTERCON 2020*. <https://doi.org/10.1109/INTERCON50315.2020.9220244>
- Phan, T. V., Sultana, S., Nguyen, T. G., & Bauschert, T. (2020). Q - TRANSFER: A Novel Framework for Efficient Deep Transfer Learning in Networking. *2020 International Conference on Artificial*

*Intelligence in Information and Communication, ICAIIC 2020*, 146–151.  
<https://doi.org/10.1109/ICAIIIC48513.2020.9065240>

Pierdicca, R., Paolanti, M., Felicetti, A., Piccinini, F., & Zingaretti, P. (2020). Automatic Faults Detection of Photovoltaic Farms: solAIr, a Deep Learning-Based System for Thermal Images. *Energies 2020, Vol. 13, Page 6496, 13(24)*, 6496. <https://doi.org/10.3390/EN13246496>

Procházka, A., Charvátová, H., Vyšata, O., Kopal, J., & Chambers, J. (2017). Breathing Analysis Using Thermal and Depth Imaging Camera Video Records. *Sensors, 17(6)*, 1408. <https://doi.org/10.3390/s17061408>

PyTorch. (2019). *torchvision.models*. <https://pytorch.org/docs/stable/torchvision/models.html>

Qiu, C., Schmitt, M., Taubenbock, H., & Zhu, X. X. (2019, May 1). Mapping human settlements with multi-seasonal sentinel-2 imagery and attention-based resnext. *2019 Joint Urban Remote Sensing Event, JURSE 2019*. <https://doi.org/10.1109/JURSE.2019.8809009>

Rajayogi, J. R., Manjunath, G., & Shobha, G. (2019, December 1). Indian Food Image Classification with Transfer Learning. *CSITSS 2019 - 2019 4th International Conference on Computational Systems and Information Technology for Sustainable Solution, Proceedings*. <https://doi.org/10.1109/CSITSS47250.2019.9031051>

Ramchoun, H., Amine, M., Idrissi, J., Ghanou, Y., & Ettaouil, M. (2016). Multilayer Perceptron: Architecture Optimization and Training. *International Journal of Interactive Multimedia and Artificial Intelligence, 4(1)*, 26. <https://doi.org/10.9781/ijimai.2016.415>

Redacción EC. (2018, February 28). *Se licitarán líneas de transmisión por US\$/100 millones*. *Dirario El Comercio*. <https://elcomercio.pe/economia/licitaran-lineas-transmision-us-100-millones-noticia-501006-noticia/>

Redondo, J. (2019, July 9). *Subestaciones Eléctricas ¿Qué son y para qué sirven? - Erenovable.com*. <https://erenovable.com/subestaciones-electricas/>

Ren, Y., Yu, Y., Li, J., & Zhang, W. (2020). Design of photovoltaic hot spot detection system based on deep learning. *Journal of Physics: Conference Series, 1693(1)*, 012075. <https://doi.org/10.1088/1742-6596/1693/1/012075>

Risco-Ramos, R. H., Pérez-Aguilar, D. A., Casaverde-Pacherrez, L. A., & Vásquez-Díaz, E. H. (2022). Use of a business intelligence framework in the management of the quality of the electricity supply in small and medium-sized companies. *DYNA, 89(221)*, 31–40. <https://doi.org/10.15446/dyna.v89n221.99085>

Risco, R. H., Pérez, D. A., Casaverde, L. A., & La-Rosa, G. (2022). Application of data mining techniques to define an operating model for a hydraulic turbogenerator. *Información Tecnológica, 33(6)*, 31–44. <https://doi.org/10.4067/S0718-07642022000600031>

- Risco, R., Perez, D., & Casaverde, L. (2020). System identification models' fit using error histogram analysis and the Hampel filter as computational tools. *Proceedings of the 2020 IEEE 27th International Conference on Electronics, Electrical Engineering and Computing, INTERCON 2020*. <https://doi.org/10.1109/INTERCON50315.2020.9220230>
- Roell, J. (2017, June 26). *Understanding Recurrent Neural Networks: The Preferred Neural Network for Time-Series Data*. <https://towardsdatascience.com/understanding-recurrent-neural-networks-the-preferred-neural-network-for-time-series-data-7d856c21b759>
- Sakib, N., & Wuest, T. (2018). Challenges and Opportunities of Condition-based Predictive Maintenance: A Review. *Procedia CIRP*, 78, 267–272. <https://doi.org/10.1016/j.procir.2018.08.318>
- Salazar, A. M., & Macabebe, E. Q. B. (2016). Hotspots Detection in Photovoltaic Modules Using Infrared Thermography. *MATEC Web of Conferences*, 70, 10015. <https://doi.org/10.1051/MATECCONF/20167010015>
- Sancho Caparrini, F. (2021, January 13). *Self Organizing Maps (SOM) in NetLogo*. <http://www.cs.us.es/~fsancho/?e=136>
- Selcuk, S. (2017). Predictive maintenance, its implementation and latest trends. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 231(9), 1670–1679. <https://doi.org/10.1177/0954405415601640>
- shao, haidong, Xia, M., Han, G., Zhang, Y., & Wan, J. (2020). Intelligent fault diagnosis of rotor-bearing system under varying working conditions with modified transfer CNN and thermal images. *IEEE Transactions on Industrial Informatics*, 1–1. <https://doi.org/10.1109/tii.2020.3005965>
- Suryansh, S. (2018, March 12). *Gradient Descent: All You Need to Know*. <https://medium.com/hackernoon/gradient-descent-aynk-7cbe95a778da>
- Szajewska, A. (2017). Development of the Thermal Imaging Camera (TIC) Technology. *Procedia Engineering*, 172, 1067–1072. <https://doi.org/10.1016/j.proeng.2017.02.164>
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 07-12-June-2015*, 1–9. <https://doi.org/10.1109/CVPR.2015.7298594>
- Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., & Le, Q. V. (2019). Mnasnet: Platform-aware neural architecture search for mobile. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2019-June*, 2815–2823. <https://doi.org/10.1109/CVPR.2019.00293>
- Tsanakas, J. A., & Botsaris, P. N. (2012). An infrared thermographic approach as a hot-spot detection tool for photovoltaic modules using image histogram and line profile analysis.

*International Journal of Condition Monitoring*, 2(1), 22–30.  
<https://doi.org/10.1784/204764212800028842>

Ullah, I., Yang, F., Khan, R., Liu, L., Yang, H., Gao, B., & Sun, K. (2017). Predictive Maintenance of Power Substation Equipment by Infrared Thermography Using a Machine-Learning Approach. *Energies*, 10(12), 1987. <https://doi.org/10.3390/en10121987>

Vijayan, V. (2020, August 3). *Deep Reinforcement Learning: Value Functions, DQN, Actor-Critic method, Back-propagation through stochastic functions*. <https://medium.com/@vishnuvijayanpv/deep-reinforcement-learning-value-functions-dqn-actor-critic-method-backpropagation-through-83a277d8c38d>

Vrbancic, G., & Podgorelec, V. (2020). Transfer Learning With Adaptive Fine-Tuning. *IEEE Access*, 8, 196197–196211. <https://doi.org/10.1109/access.2020.3034343>

Wai, Y. J., Yussof, Z. bin M., bin Salim, S. I., & Chuan, L. K. (2018). Fixed point implementation of Tiny-Yolo-v2 using OpenCL on FPGA. *International Journal of Advanced Computer Science and Applications*, 9(10), 506–512. <https://doi.org/10.14569/IJACSA.2018.091062>

Wang, Z., Tian, G. Y., Meo, M., & Ciampa, F. (2018). Image processing based quantitative damage evaluation in composites with long pulse thermography. *NDT and E International*, 99, 93–104. <https://doi.org/10.1016/j.ndteint.2018.07.004>

Wan, S., Qi, L., Xu, X., Tong, C., & Gu, Z. (2020). Deep Learning Models for Real-time Human Activity Recognition with Smartphones. *Mobile Networks and Applications*, 25(2), 743–755. <https://doi.org/10.1007/s11036-019-01445-x>

Wei, S., Li, X., Ding, S., Yang, Q., & Yan, W. (2019). Hotspots Infrared detection of photovoltaic modules based on Hough line transformation and Faster-RCNN approach. *2019 6th International Conference on Control, Decision and Information Technologies, CoDIT 2019*, 1209–1214. <https://doi.org/10.1109/CODIT.2019.8820333>

Woschank, M., Rauch, E., & Zsifkovits, H. (2020). A Review of Further Directions for Artificial Intelligence, Machine Learning, and Deep Learning in Smart Logistics. *Sustainability*, 12(9), 3760. <https://doi.org/10.3390/su12093760>

Xiao, J., Wang, J., Cao, S., & Li, B. (2020). Application of a Novel and Improved VGG-19 Network in the Detection of Workers Wearing Masks. *Journal of Physics: Conference Series*, 1518(1), 43. <https://doi.org/10.1088/1742-6596/1518/1/012041>

Xue, D., Zhou, X., Li, C., Yao, Y., Rahaman, M. M., Zhang, J., Chen, H., Zhang, J., Qi, S., & Sun, H. (2020). An Application of Transfer Learning and Ensemble Learning Techniques for Cervical Histopathology Image Classification. *IEEE Access*, 8, 104603–104618. <https://doi.org/10.1109/ACCESS.2020.2999816>

Yakunin, S., Benin, B. M., Shynkarenko, Y., Nazarenko, O., Bodnarchuk, M. I., Dirin, D. N., Hofer, C., Cattaneo, S., & Kovalenko, M. V. (2019). High-resolution remote thermometry and

thermography using luminescent low-dimensional tin-halide perovskites. *Nature Materials*, 18(8), 846–852. <https://doi.org/10.1038/s41563-019-0416-2>

Yang, R., & He, Y. (2016). Optically and non-optically excited thermography for composites: A review. *Infrared Physics and Technology*, 75, 26–50. <https://doi.org/10.1016/j.infrared.2015.12.026>

Yang, W., Luo, J., Wu, X., Li, X., Jiang, Z., & Pan, Z. (2018). Image tactile perception with an improved JSEG algorithm. *International Journal of Performability Engineering*, 14(1), 77–88. <https://doi.org/10.23940/ijpe.18.01.p9.7788>

Yan, J., Meng, Y., Lu, L., & Li, L. (2017). Industrial Big Data in an Industry 4.0 Environment: Challenges, Schemes, and Applications for Predictive Maintenance. *IEEE Access*, 5, 23484–23491. <https://doi.org/10.1109/ACCESS.2017.2765544>

Yıldız, İ., Tian, P., Dy, J., Erdoğmuş, D., Brown, J., Kalpathy-Cramer, J., Ostmo, S., Peter Campbell, J., Chiang, M. F., & Ioannidis, S. (2019). Classification and comparison via neural networks. *Neural Networks*, 118, 65–80. <https://doi.org/10.1016/j.neunet.2019.06.004>

Yuan, C., Sun, X., & Lv, R. (2016). Fingerprint liveness detection based on multi-scale LPQ and PCA. *China Communications*, 13(7), 60–65. <https://doi.org/10.1109/CC.2016.7559076>

Yu, S. S., Chu, S. W., Wang, C. M., Chan, Y. K., & Chang, T. C. (2018). Two improved k-means algorithms. *Applied Soft Computing Journal*, 68, 747–755. <https://doi.org/10.1016/j.asoc.2017.08.032>

Zagoruyko, S., & Komodakis, N. (2016). Wide Residual Networks. *Proceedings of the British Machine Vision Conference 2016, 2016-September*, 87.1-87.12. <https://doi.org/10.5244/C.30.87>

Zenisek, J., Holzinger, F., & Affenzeller, M. (2019). Machine learning based concept drift detection for predictive maintenance. *Computers and Industrial Engineering*, 137, 106031. <https://doi.org/10.1016/j.cie.2019.106031>

Zhang, K., Guo, Y., Wang, X., Yuan, J., & Ding, Q. (2019). Multiple feature reweight DenseNet for image classification. *IEEE Access*, 7, 9872–9880. <https://doi.org/10.1109/ACCESS.2018.2890127>

Zhao, Z., Zhen, Z., Zhang, L., Qi, Y., Kong, Y., & Zhang, K. (2019). Insulator Detection Method in Inspection Image Based on Improved Faster R-CNN. *Energies 2019, Vol. 12, Page 1204*, 12(7), 1204. <https://doi.org/10.3390/EN12071204>

Zimmerman, D. W., & Zumbo, B. D. (2010). Relative power of the wilcoxon test, the friedman test, and repeated-measures ANOVA on ranks. *Journal of Experimental Education*, 62(1), 75–86. <https://doi.org/10.1080/00220973.1993.9943832>

Zou, H., & Huang, F. (2015). A novel intelligent fault diagnosis method for electrical equipment using infrared thermography. *Infrared Physics and Technology*, 73, 29–35. <https://doi.org/10.1016/j.infrared.2015.08.019>



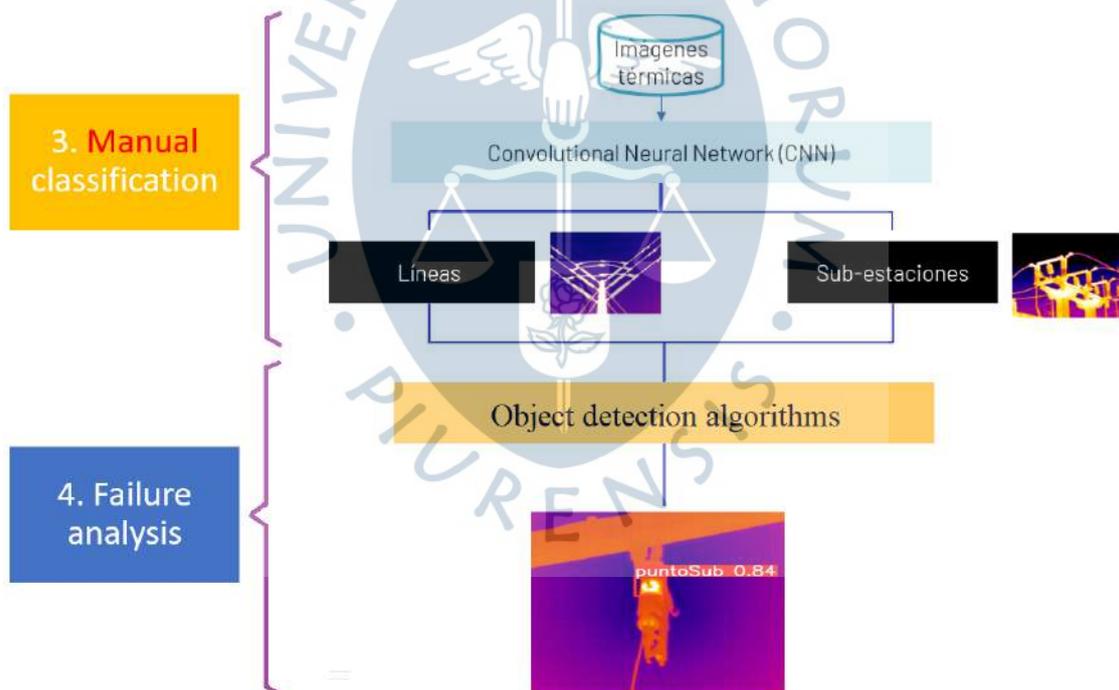


## Apéndices

### Apéndice 01: Aporte adicional - Detección de puntos calientes

Además de la clasificación de imágenes térmicas, y como aporte adicional a la investigación, se realizaron 65 experimentos entrenando modelos de detección de objetos, con la finalidad de encontrar zonas con presencia de altas temperaturas, siendo esto es el principal síntoma de problemas en las líneas o subestaciones eléctricas (Fig. 1).

Fig. 1. Metodología técnica.



Nota. Elaboración propia.

#### Objetivo

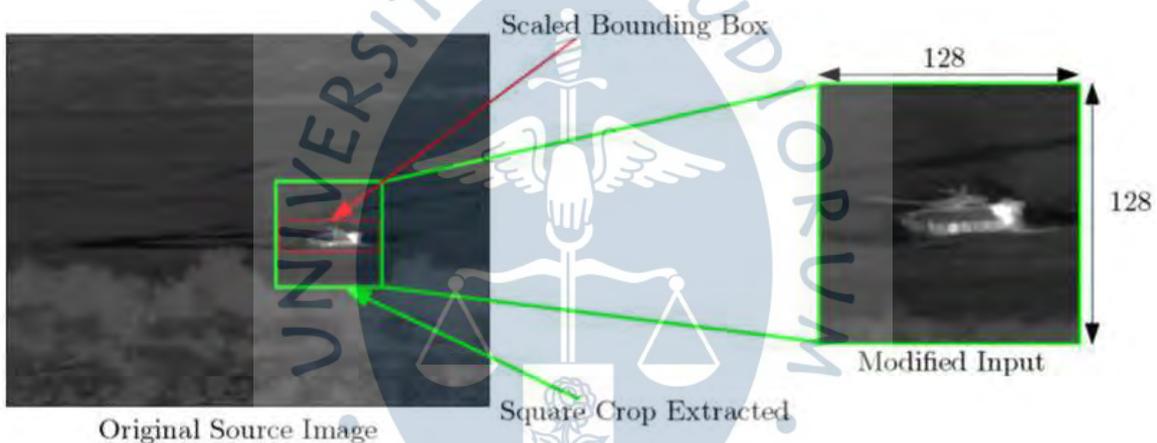
- Utilizar y comparar distintos algoritmos de segmentación para detectar puntos calientes en las imágenes térmicas.

## Investigaciones previas

En la encuesta sobre técnicas de diagnóstico automático de fallas eléctricas basadas en termografía infrarroja se indica que algunas de las técnicas más utilizadas son K-means clustering, support vector machine (SVM), Multilayer Perceptron (MLP), sistemas difusos (fuzzy system), e inclusive la combinación de los mismos (Lee & Teoh, 2019).

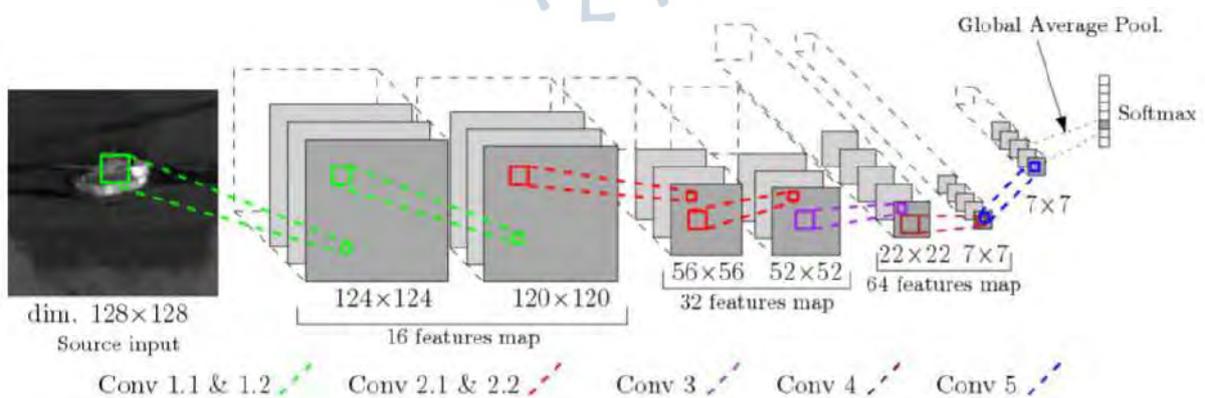
Una investigación estudia la identificación y el reconocimiento de objetos mediante la utilización de imágenes infrarrojas aplicadas a un entorno militar (d'Acremont et al., 2019), donde no se dispone de un conjunto de datos a gran escala, para ello presentan una arquitectura CNN con la incorporación de data augmentation y fine-tuning, demostrando que la arquitectura propuesta es superior frente al uso de máquinas de vectores de soporte (SVM). En la Fig. 2 se aprecia la extracción de características y en la Fig. 3 la arquitectura propuesta.

**Fig. 2.** Extractor de características.



Tomado de "NN-Based Target Recognition and Identification for Infrared Imaging in Defense Systems" (d'Acremont et al., 2019).

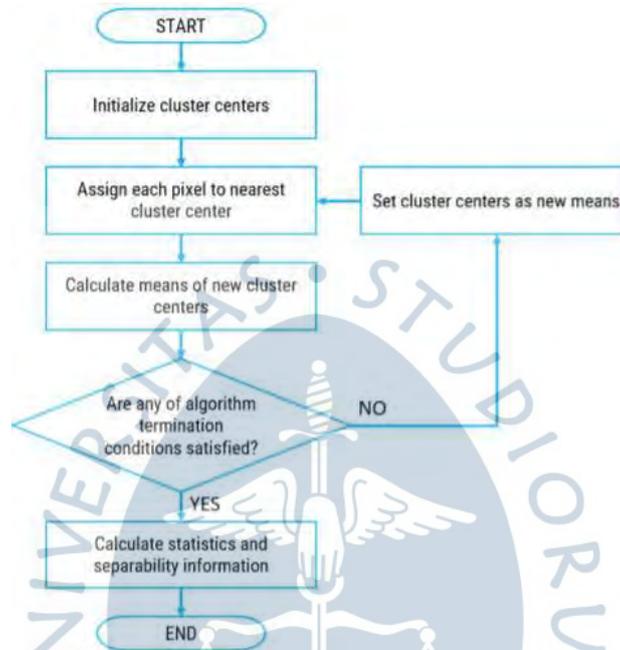
**Fig. 3.** Arquitectura propuesta.



Tomado de "NN-Based Target Recognition and Identification for Infrared Imaging in Defense Systems" (d'Acremont et al., 2019).

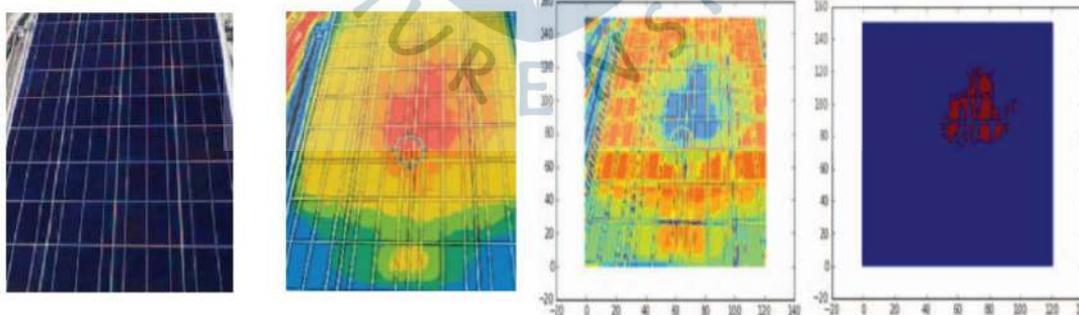
Otro estudio realiza la segmentación de imágenes con la finalidad de detectar puntos calientes en módulos fotovoltaicos (Ngo & Macabebe, 2017), para lograr este objetivo utilizan el algoritmo de agrupación K-means (ver Fig. 4). El modelo presenta resultados favorables al segmentar las zonas más calientes de los módulos fotovoltaicos, tal como se aprecia en la Fig. 5.

**Fig. 4.** Flujo del algoritmo K-means.



Tomado de "Image segmentation using K-means color quantization and density-based spatial clustering of applications with noise (DBSCAN) for hotspot detection in photovoltaic modules" (Ngo & Macabebe, 2017).

**Fig. 5.** Detección de puntos calientes (Ngo & Macabebe, 2017).

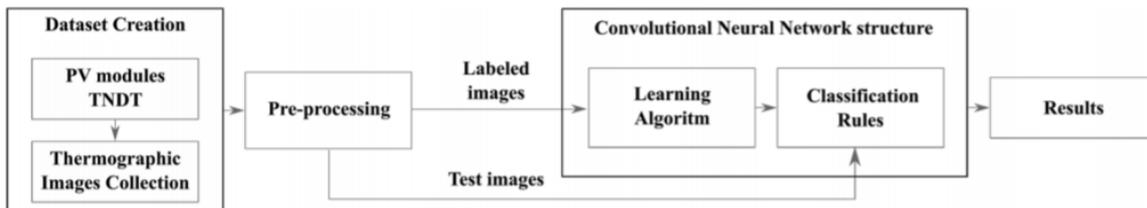


Tomado de "Image segmentation using K-means color quantization and density-based spatial clustering of applications with noise (DBSCAN) for hotspot detection in photovoltaic modules" (Ngo & Macabebe, 2017).

Otro estudio plantea el uso de técnicas de inteligencia artificial con la finalidad de clasificar pérdidas de energía en sistemas fotovoltaico, esto se realiza mediando el uso de ensayos termográficos. Se plantea la hipótesis de que el bajo rendimiento de los módulos fotovoltaicos se debe a condiciones de funcionamiento anormales, como fallas o mal

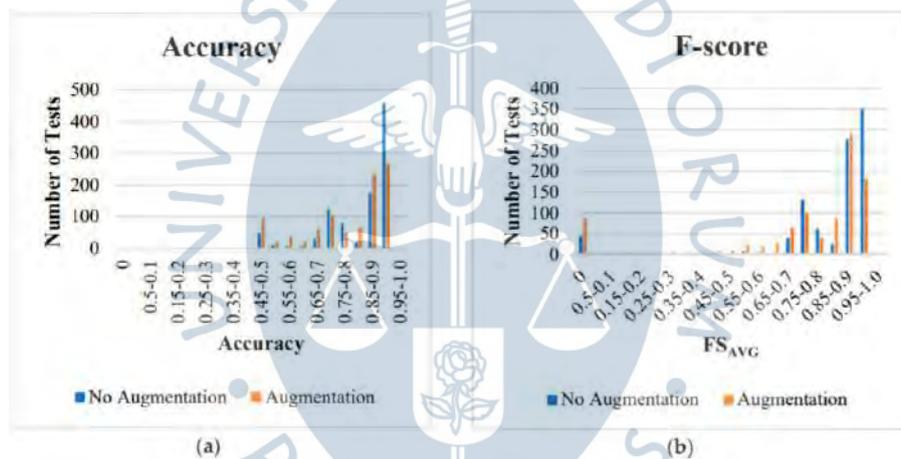
funcionamiento. El enfoque propuesto permite clasificar automáticamente las imágenes termográficas de la red neuronal convolucional (CNN) del sistema (ver Fig. 6), logrando una precisión del 98% (ver Fig. 7). En comparación con otros enfoques en la literatura, ofrece numerosas ventajas, que incluyen velocidad de ejecución, rapidez de diagnóstico, reducción de costos y pérdidas en la producción de electricidad (Cipriani et al., 2020).

**Fig. 6.** Construcción de la CNN



Tomado de “Convolutional Neural Network for Dust and Hotspot Classification in PV Modules” (Cipriani et al., 2020).

**Fig. 7.** Resultados del modelo (a) Accuracy (b) F1-score

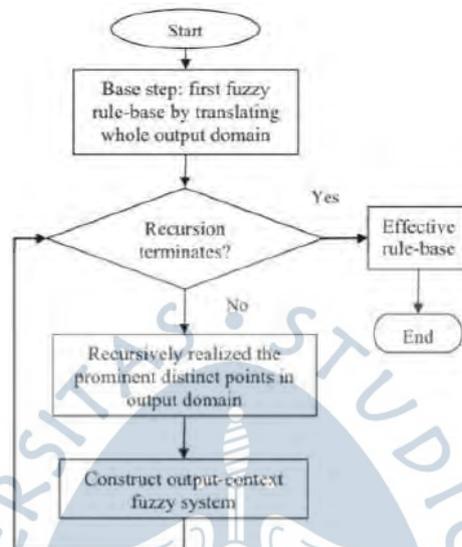


Tomado de “Convolutional Neural Network for Dust and Hotspot Classification in PV Modules” (Cipriani et al., 2020).

La tecnología de termografía infrarroja se utiliza actualmente en diversas aplicaciones, incluido el diagnóstico de fallas en equipos eléctricos; las anomalías térmicas se diagnostican identificando y clasificando las condiciones de los puntos calientes en componentes eléctricos. Es por ello que en la investigación se propuso un nuevo sistema difuso de contexto de salida construido recursivamente para caracterizar la condición de los puntos calientes eléctricos (Ahmed et al., 2015), para ello se capturan las imágenes térmicas con una cámara infrarroja, y las características de intensidad se extraen de cada punto caliente. Luego se aplicó el sistema difuso construido recursivamente (RCFS) para analizar automáticamente las condiciones de las anomalías térmicas (ver Fig. 8). La clasificación de los puntos calientes se basa en tres categorías (i) normales (ii) en advertencia y (iii) críticas. El RCFS propuesto considera las distinciones importantes en el dominio de salida mediante el uso de un método auto organizado. Una encuesta de 253 regiones detectadas muestra que el RCFS produce una

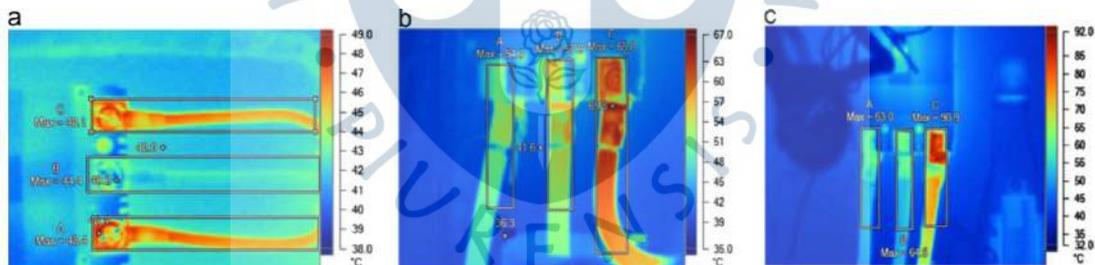
precisión de prueba del 92,3% y el 80% para clasificar las condiciones en dos y tres clases, respectivamente (ver Fig. 9). La evaluación de diagnóstico termográfico muestra que el sistema inteligente identifica automáticamente los límites aceptables de las condiciones de puntos calientes.

**Fig. 8.** Flujo del algoritmo RCFS.



Tomado de "Recursive construction of output- context fuzzy systems for the condition monitoring of electrical hotspots based on infrared thermography" (Ahmed et al., 2015).

**Fig. 9.** Condición (a) Normal (b) Advertencia y (c) Crítica.

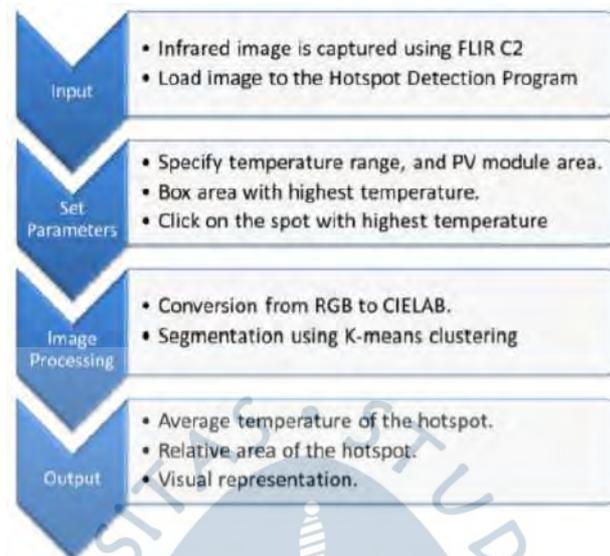


Tomado de "Recursive construction of output- context fuzzy systems for the condition monitoring of electrical hotspots based on infrared thermography" (Ahmed et al., 2015).

Un trabajo investigativo desarrolló una técnica no invasiva que puede detectar el calentamiento localizado y cuantificar el área de los puntos calientes (Fig. 10), una posible causa de degradación en los sistemas fotovoltaicos (Salazar & Macabebe, 2016). Esto se hace mediante el uso de termografía infrarroja, una técnica de evaluación no destructiva que permite la inspección sin contacto en tiempo real. En este enfoque, se tomaron imágenes térmicas de un módulo fotovoltaico en funcionamiento utilizando una cámara infrarroja, las cuales se analizaron mediante un algoritmo de detección de puntos calientes implementado en MATLAB (Fig. 11). Antes del procesamiento de imágenes, estas se convirtieron al espacio

de color CIE L \* a \* b, lo que hizo que la implementación de la agrupación en clústeres de k-means sea computacionalmente eficiente en el aislamiento de puntos calientes.

**Fig. 10.** Flujo de la detección de puntos calientes.



Tomado de “Hotspots Detection in Photovoltaic Modules Using Infrared Thermography” (Salazar & Macabebe, 2016).

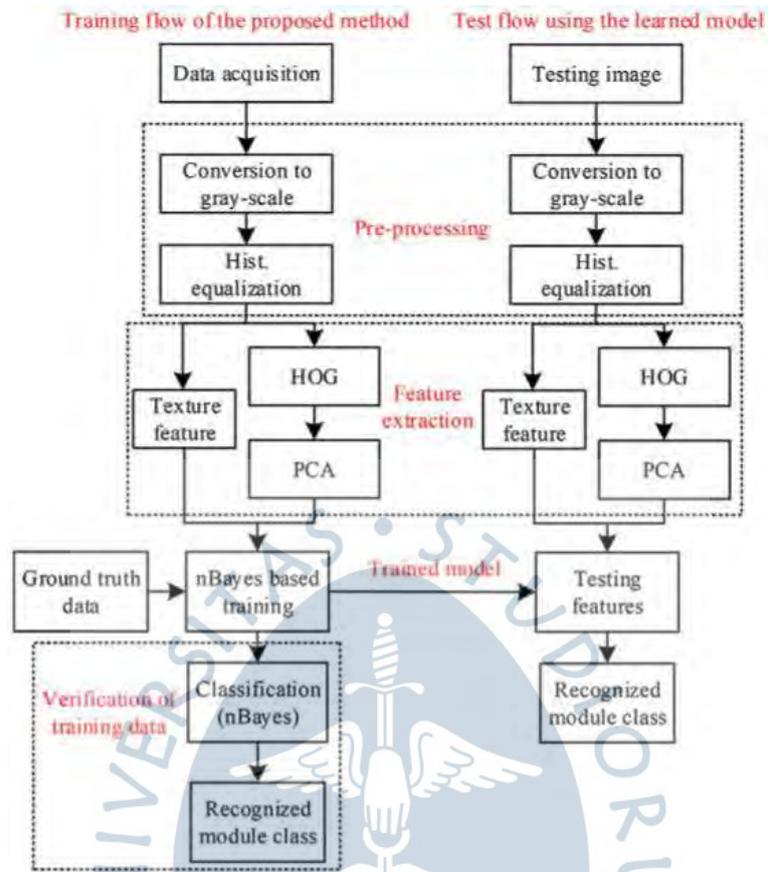
**Fig. 11.** (a) Imagen termográfica (b) Punto caliente y (c) Área y temperatura



Tomado de “Hotspots Detection in Photovoltaic Modules Using Infrared Thermography” (Salazar & Macabebe, 2016).

Los puntos calientes en los módulos fotovoltaicos debido a defectos y condiciones operativas pueden ocasionar una operación poco confiable y eficiente, además de un posible fallo en todo el sistema fotovoltaico. Desde el punto de vista de supervisión, los puntos críticos deben detectarse y categorizarse para su posterior mantenimiento. Los puntos calientes se detectan, evalúan y categorizan de forma única mediante el uso de una técnica de aprendizaje automático en imágenes térmicas de módulos fotovoltaicos (Fig. 12). Para la clasificación se utilizaron la textura y el histograma de características de gradiente (HOG) de las imágenes térmicas de los módulos fotovoltaicos (Fig. 13). Los puntos categorizados se detectaron entrenando el algoritmo de aprendizaje automático Naive Bayes. Los resultados experimentales se realizaron en un sistema fotovoltaico de 42,24 kWp, alcanzando una tasa de reconocimiento media de alrededor del 94,1% para el conjunto de 375 muestras (Niazi et al., 2019).

**Fig. 12.** Método nBayes y modelo de aprendizaje.



Tomado de "Hotspot diagnosis for solar photovoltaic modules using a Naive Bayes classifier" (Niazi et al., 2019).

**Fig. 13.** Extracción de características del HOG.

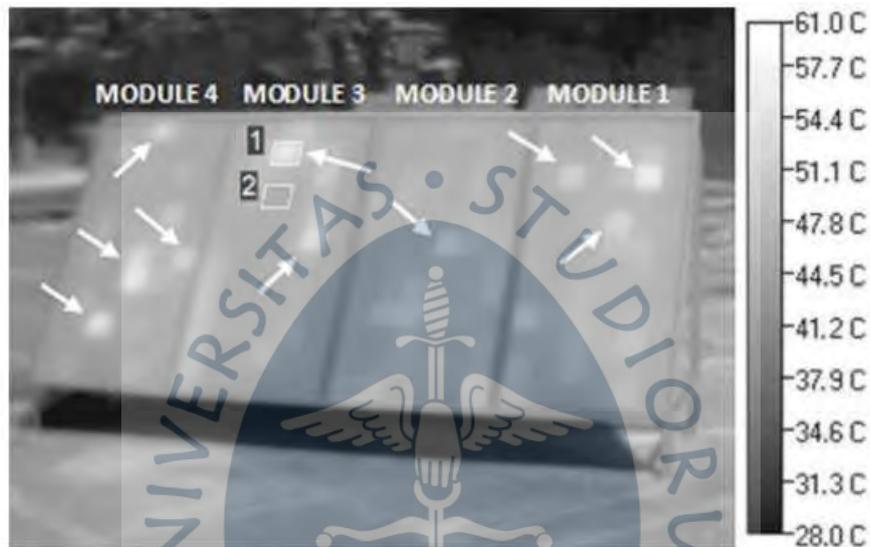


Tomado de "Hotspot diagnosis for solar photovoltaic modules using a Naive Bayes classifier" (Niazi et al., 2019).

Otro estudio analiza el uso de una herramienta de detección de puntos calientes para módulos fotovoltaicos, basada en la monitorización del estado mediante termografía infrarroja (Fig. 14). La experimentación del enfoque previsto incluye varios conjuntos diarios de mediciones termográficas de matrices fotovoltaicas. El procesamiento de las imágenes

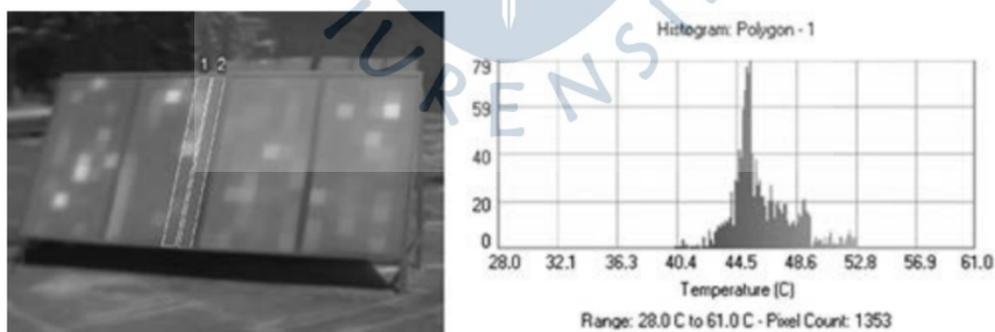
térmicas mediante un histograma de imagen básico y análisis de perfil de línea, proporciona datos útiles para la detección y cuantificación de puntos calientes dentro de cada módulo fotovoltaico (Fig. 15). Las variaciones específicas en las características morfológicas de los histogramas y los perfiles lineales de las imágenes térmicas prueban la presencia de defectos, su origen dentro del módulo y su impacto en la degradación del rendimiento del campo fotovoltaico debido al efecto de calentamiento de los puntos calientes (Tsanakas & Botsaris, 2012).

**Fig. 14.** Imagen térmica con puntos calientes.



Tomado de “An infrared thermographic approach as a hot-spot detection tool for photovoltaic modules using image histogram and line profile analysis” (Tsanakas & Botsaris, 2012).

**Fig. 15.** Puntos calientes e histograma.

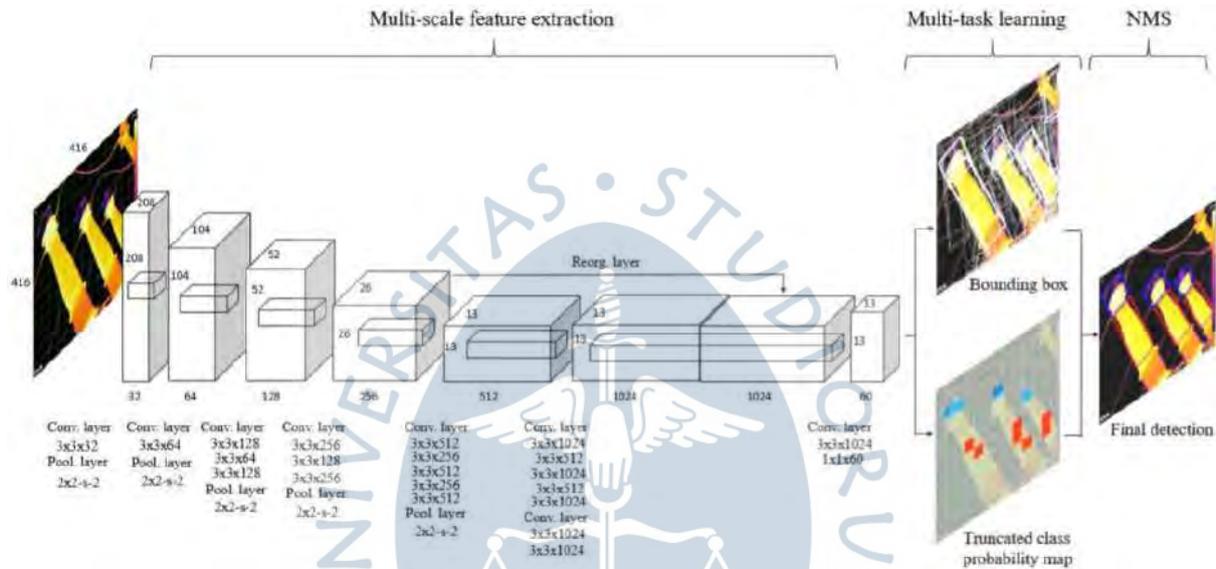


Tomado de “An infrared thermographic approach as a hot-spot detection tool for photovoltaic modules using image histogram and line profile analysis” (Tsanakas & Botsaris, 2012).

Otro paper presenta un enfoque de aprendizaje profundo para detectar piezas de equipos en tiempo real (Gong et al., 2018). Se propone una red neuronal convolucional profunda basado en el algoritmo de detección de objetos YOLO (You Only Look Once), tal

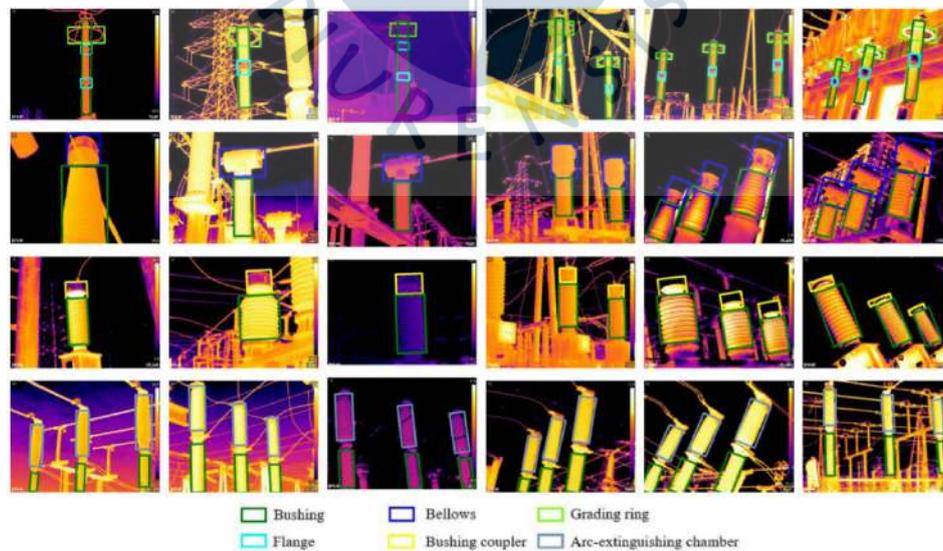
como se aprecia en la Fig. 16; el objetivo fue predecir las coordenadas, el ángulo de orientación y el tipo de clase de cada parte del equipo. También se incorpora en el modelo una consistencia previa relativa a la orientación entre las partes para mejorar los resultados de la predicción. Para la evaluación, se construyó un gran conjunto de imágenes que contiene varios tipos de escenarios. Los experimentos con el conjunto de datos mostraron que el método es robusto al ruido, alcanzando 93.7% de precisión cuando la intersección sobre el umbral de unión es 0,5 y se ejecuta a 20 fps en la GPU (ver Fig. 17).

**Fig. 16.** Framework utilizado.



Tomado de "A Deep Learning Approach for Oriented Electrical Equipment Detection in Thermal Images" (Gong et al., 2018).

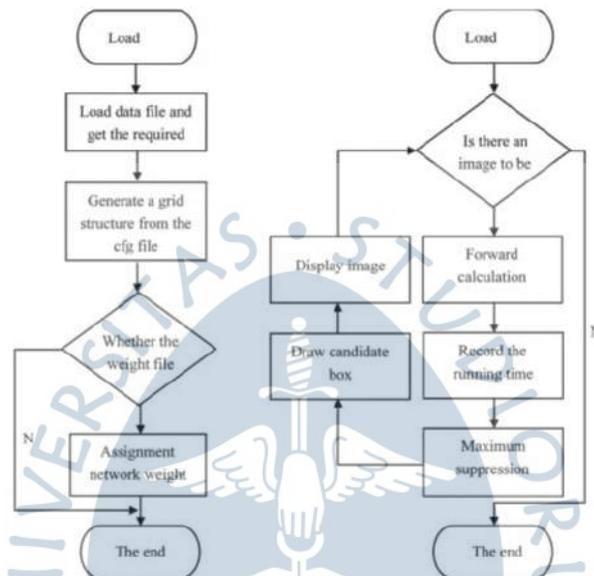
**Fig. 17.** Predicciones del modelo



Tomado de "A Deep Learning Approach for Oriented Electrical Equipment Detection in Thermal Images" (Gong et al., 2018).

Se realizó el entrenamiento de la red YOLOv3 (ver Fig. 18) mediante imágenes infrarrojas recopiladas en campo, logrando la detección de equipos de energía y puntos de falla en tiempo real, y determina qué áreas del equipo de energía son anormales (X. Li, 2019). Se probó el modelo entrenado YOLOv3, el valor mAP obtenido es corresponde a 34,63% y la tasa de recuperación, 21%. El tiempo de funcionamiento del Jetson Nano fue de 0.7 a 0.9 s, lo que satisface los requisitos para las pruebas de equipos de potencia (Fig. 19).

**Fig. 18.** Flujo del proceso.



Tomado de “*Design of Infrared Anomaly Detection for Power Equipment Based on YOLOv3*” (X. Li, 2019).

**Fig. 19.** Interfaz de detección de objetos.

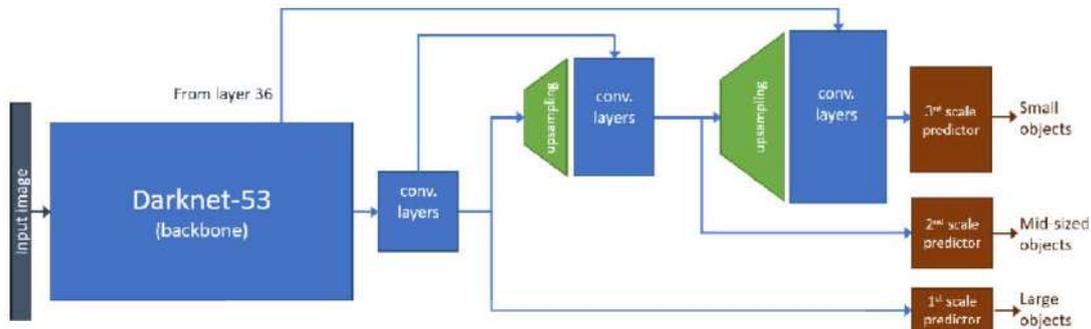


Tomado de “*Design of Infrared Anomaly Detection for Power Equipment Based on YOLOv3*” (X. Li, 2019).

Se aborda el problema de la detección de paneles fotovoltaicos utilizando un marco de red neuronal convolucional llamado YOLO (Fig. 20). Se demuestra que es capaz de segmentar paneles de una imagen de forma eficaz y eficiente (Greco et al., 2020). El método

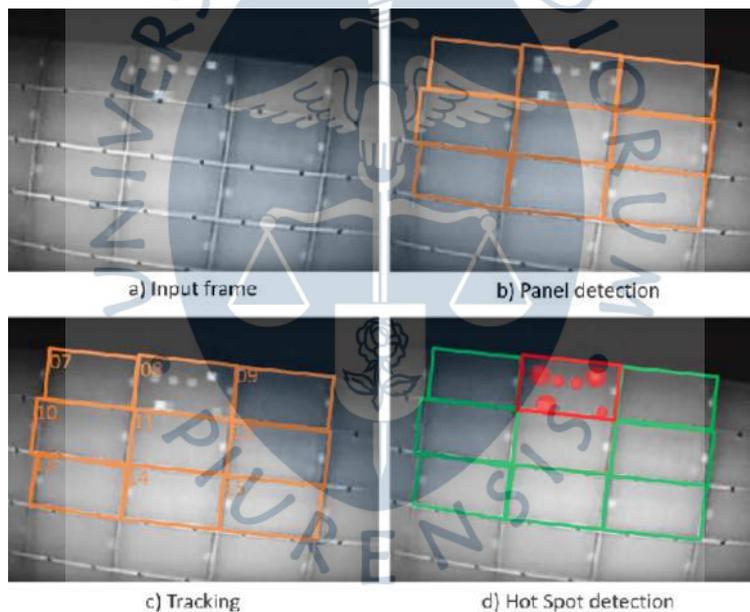
se evalúa cuantitativamente y se compara con los enfoques de detección de paneles fotovoltaicos existentes en el mayor conjunto de datos de referencia disponible públicamente; los resultados experimentales confirman su robustez (Fig. 21).

**Fig. 20.** Flujo del proceso – Dark-net.



Tomado de “A deep learning based approach for detecting panels in photovoltaic plants” (Greco et al., 2020).

**Fig. 21.** a) Imagen de entrada, b) Detección, c) Rastreo, d) Puntos calientes.

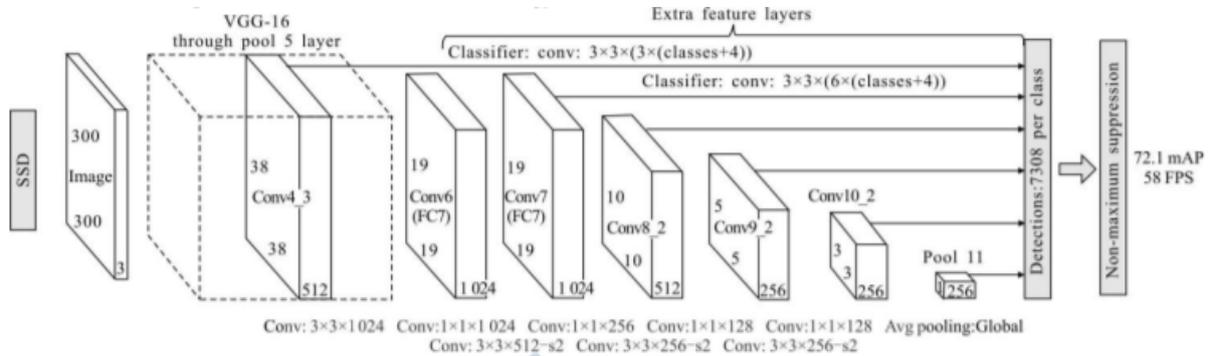


Tomado de “A deep learning based approach for detecting panels in photovoltaic plants” (Greco et al., 2020).

Se diseñó un algoritmo mejorado de Detector de caja múltiple de disparo único (SSD) para la detección de puntos calientes fotovoltaicos (Ren et al., 2020). El algoritmo utilizó la red MobileNet para reemplazar la estructura de red neuronal convolucional VGG16 en el SSD original (Fig. 22.). Esta red es una estructura de convolución separable en profundidad, por lo que su uso en la extracción de características ayudó a reducir la cantidad de parámetros en la estructura y lograr el propósito de acelerar la red. Los resultados experimentales muestran que el algoritmo mejorado puede detectar los puntos calientes de la matriz fotovoltaica con

buena confianza, baja tasa de detección y buena robustez (Fig. 24). En comparación con el algoritmo You Only Look Once (YOLO) y el algoritmo SSD original, la velocidad de detección mejoró significativamente.

**Fig. 23.** Arquitectura del modelo.



Tomado de "Design of photovoltaic hot spot detection system based on deep learning" (Ren et al., 2020).

**Fig. 24.** Resultados de la detección de puntos calientes fotovoltaicos.



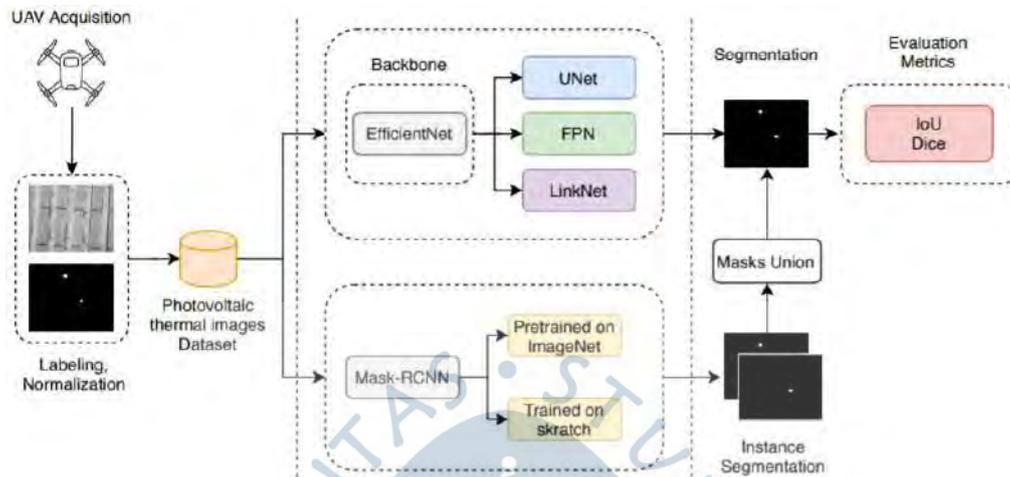
Tomado de "Design of photovoltaic hot spot detection system based on deep learning" (Ren et al., 2020).

Se propone un sistema de inteligencia artificial basado en el aprendizaje profundo para la detección de células anómalas en imágenes fotovoltaicas (Fig. 25) obtenidas de vehículos aéreos no tripulados equipados con un sensor infrarrojo (Pierdicca et al., 2020). El sistema se basa en la arquitectura de red neuronal convolucional basada en la región de máscara (Mask R-CNN), adoptada porque realiza simultáneamente la detección de objetos y la segmentación de instancias, lo que lo hace útil para la tarea de inspección automatizada. Se compararon y evaluaron los rendimientos de tres redes neuronales profundas (UNet, FPN et LinkNet). Los resultados muestran la efectividad de la propuesta en términos de intersección sobre unión (IoU) y el coeficiente de Dice (Fig. 26).

Otro estudio presenta un enfoque para identificar paneles y detectar puntos calientes, además de establecer sus ubicaciones (Huerta Herraiz et al., 2020). Se consideraron 2 redes neuronales convolucionales basadas en regiones para generar una estructura de detección robusta. Los datos se adquirieron y luego se procesaron automáticamente, lo que permitió la detección de fallas durante la inspección. El proceso de entrenamiento, la adquisición y

procesamiento de datos y la generación de resultados se muestra en la Fig. 27. La precisión, eficiencia y el desempeño del enfoque bajo diferentes escenarios reales se evalúan estadísticamente obteniendo resultados satisfactorios (Fig. 28).

**Fig. 25.** Flujo del modelo solAir.



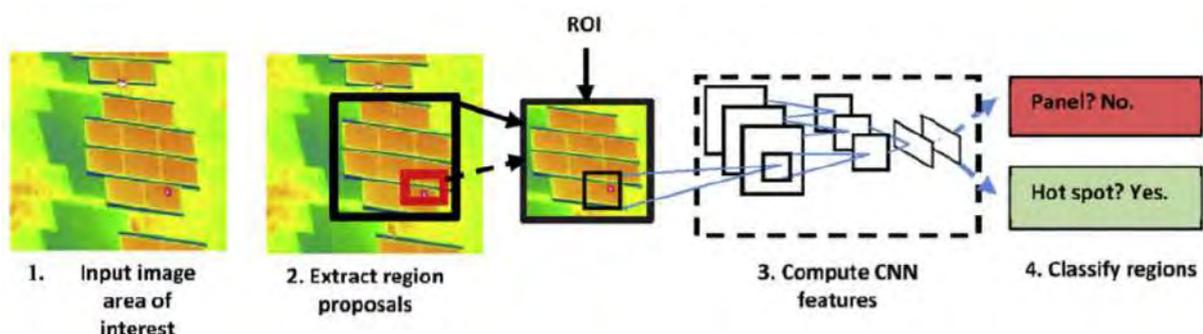
Tomado de “Automatic Faults Detection of Photovoltaic Farms: solAir, a Deep Learning-Based System for Thermal Images” (Pierdicca et al., 2020).

**Fig. 26.** Detección de puntos calientes.



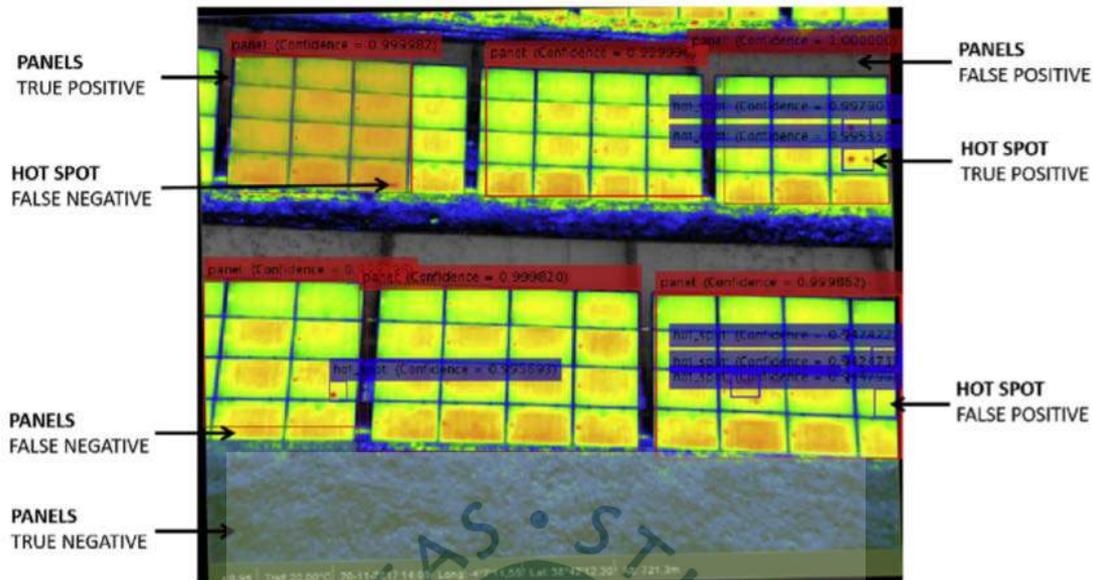
Tomado de “Automatic Faults Detection of Photovoltaic Farms: solAir, a Deep Learning-Based System for Thermal Images” (Pierdicca et al., 2020).

**Fig. 27.** Sistema propuesto.



Tomado de “Photovoltaic plant condition monitoring using thermal images analysis by convolutional neural network-based structure” (Huerta Herraiz et al., 2020).

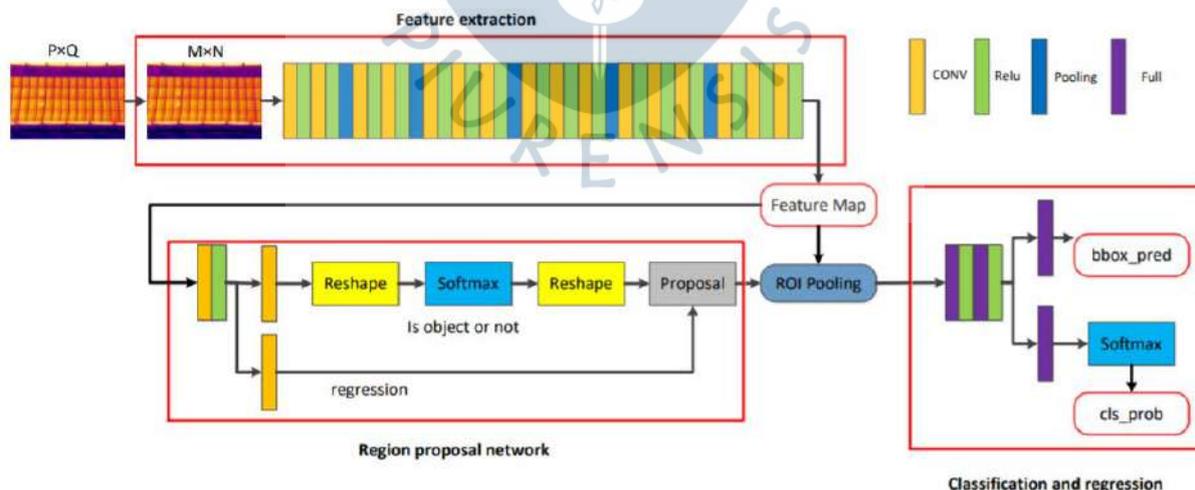
Fig. 28. Detección de puntos calientes.



Tomado de "Photovoltaic plant condition monitoring using thermal images analysis by convolutional neural network-based structure" (Huerta Herraiz et al., 2020).

Una investigación propone dos enfoques para detectar puntos calientes en imágenes infrarrojas de módulos fotovoltaicos (Wei et al., 2019). La tecnología clásica de procesamiento de imágenes digitales utiliza principalmente la transformación de línea Hough y un operador para detectar puntos de acceso. El modelo de aprendizaje profundo se basa en Faster-RCNN (Fig. 29) y el aprendizaje por transferencia, los cuales funcionan mejor con mayores recursos computacionales, arrojando un valor de F1-score de 95.15% (Fig. 30).

Fig. 29. Estructura Faster R-CNN.

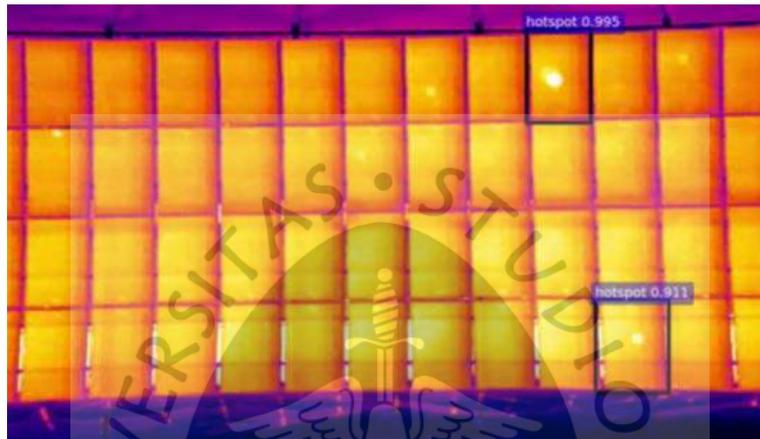


Tomado de "Hotspots Infrared detection of photovoltaic modules based on Hough line transformation and Faster-RCNN approach" (Wei et al., 2019).

Otro trabajo propone un método basado en la red neuronal convolucional profunda (CNN) para el reconocimiento de imágenes infrarrojas de equipos eléctricos (Han et al., 2020).

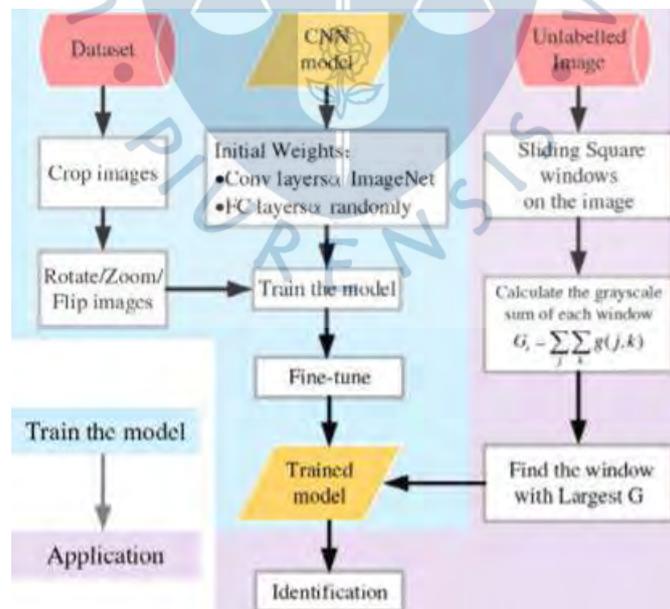
En primer lugar, se crea un modelo de identificación de CNN profundo basado en MobileNet con pesos iniciales en ImageNet, además del uso de data augmentation, considerándose 3547 imágenes de equipos de subestación (Fig. 31). Los resultados demuestran que la precisión de predicción del método propuesto alcanza el 97,72% en la validación, y el método de selección de ROI mejora la confianza en un 8% en la prueba. Como resultado, este método puede promover la eficiencia del cálculo y tiene buenas perspectivas de aplicación en dispositivos integrados como cámaras y robots de subestaciones (Fig. 32).

**Fig. 30.** *Detección de puntos calientes.*



Tomado de “Hotspots Infrared detection of photovoltaic modules based on Hough line transformation and Faster-RCNN approach” (Wei et al., 2019).

**Fig. 31.** *Flujo de trabajo.*

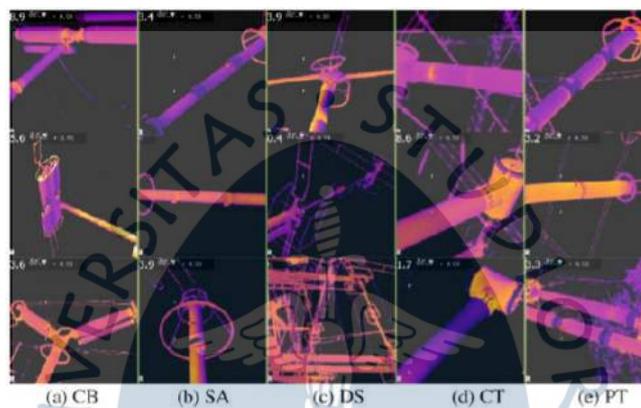


Tomado de “Electrical equipment identification in infrared images based on ROI-selected CNN method” (Han et al., 2020).

Se propuso un algoritmo de detección cuantitativa para bentos marinos basado en Faster R-CNN (Fig. 32). Una unidad de selección adaptativa del núcleo de convolución está

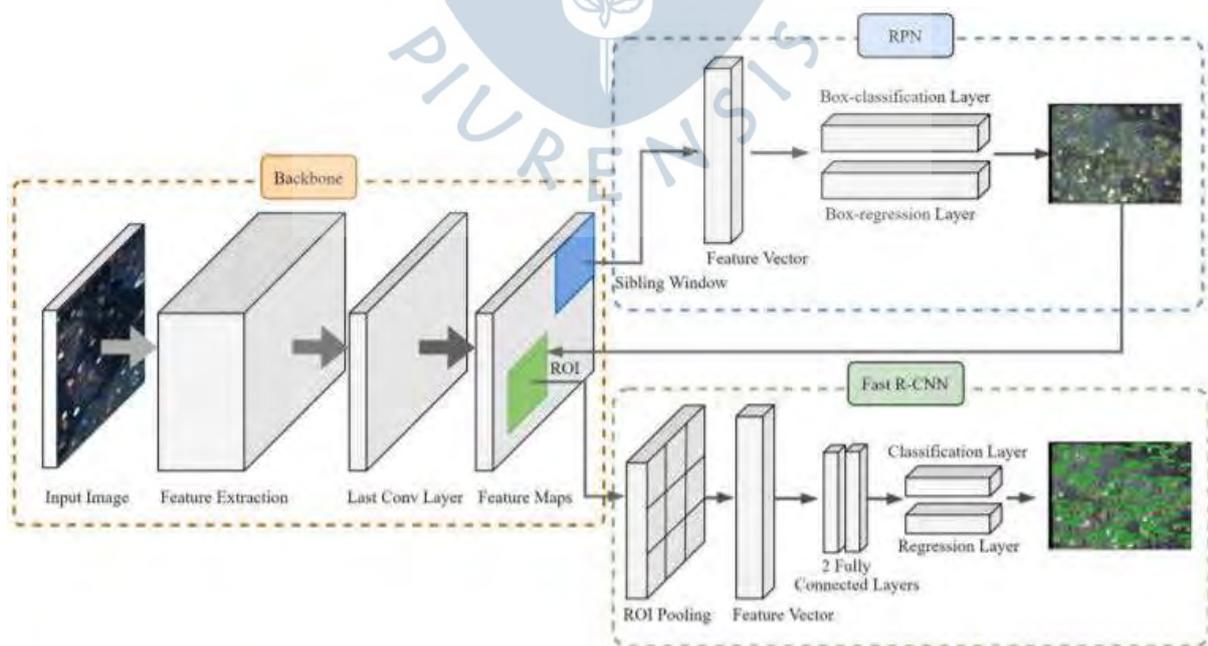
incrustada en la red para mejorar la capacidad de extracción de características. En base a esto, se introdujo la fusión de características de múltiples resoluciones para diseñar una estructura de características de deconvolución para la detección de objetos pequeños (Y. Liu & Wang, 2021). El uso de transfer learning también se empleó para entrenar el modelo propuesto y reducir la limitación del pequeño conjunto de datos. Los resultados muestran que, en comparación con el Faster R-CNN original, el algoritmo propuesto mejora la precisión de reconocimiento de bentos marinos del 93,25% al 96,32% y reduce el error medio de 16,53 a 7,38. Esta mejora refleja que el algoritmo propuesto es más adecuado para la detección cuantitativa de objetos pequeños y densos en el fondo marino (Fig. 33).

**Fig. 32.** Predicciones del modelo.



Tomado de “*Electrical equipment identification in infrared images based on ROI-selected CNN method*” (Han et al., 2020).

**Fig. 33.** Arquitectura de Faster R- CNN.



Tomado de “A quantitative detection algorithm based on improved faster R-CNN for marine benthos” (Y. Liu & Wang, 2021).

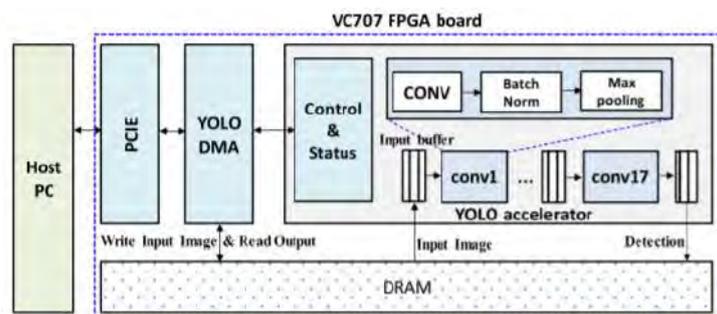
**Fig. 34.** Resultados del modelo



Tomado de “A quantitative detection algorithm based on improved faster R-CNN for marine benthos” (Y. Liu & Wang, 2021).

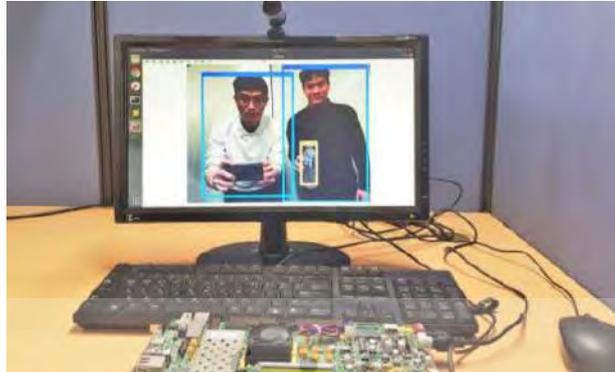
Un grupo de investigadores (Nguyen et al., 2019) presentó un acelerador de hardware que implementa una CNN basada en YOLO (Fig. 35). Los parámetros de YOLO CNN se vuelven a entrenar y cuantificar con el conjunto de datos PASCAL VOC utilizando peso binario y activación flexible de bits bajos. El peso binario permite almacenar todo el modelo de red en bloques RAM de una matriz de puertas programables en campo (FPGA). En el diseño propuesto, todas las capas convolucionales están completamente canalizadas para una mejor utilización del hardware. Esta CNN fue implementada usando VC707 FPGA, y logra un rendimiento de 1.877 operaciones por segundo (TOPS) a 200 MHz con procesamiento por lotes mientras consume 18.29 W de potencia, lo que muestra una mejora considerable en la eficiencia energética. En cuanto a la precisión de detección de objetos, alcanza una precisión media (mAP) del 64,16% para el conjunto de datos PASCAL VOC 2007 que es sólo un 2,63% menor que el mAP de la red YOLO (Fig. 36).

**Fig. 35.** Arquitectura propuesta.



Tomado de “A high-throughput and power-efficient fpga implementation of yolo cnn for object detection” (Nguyen et al., 2019).

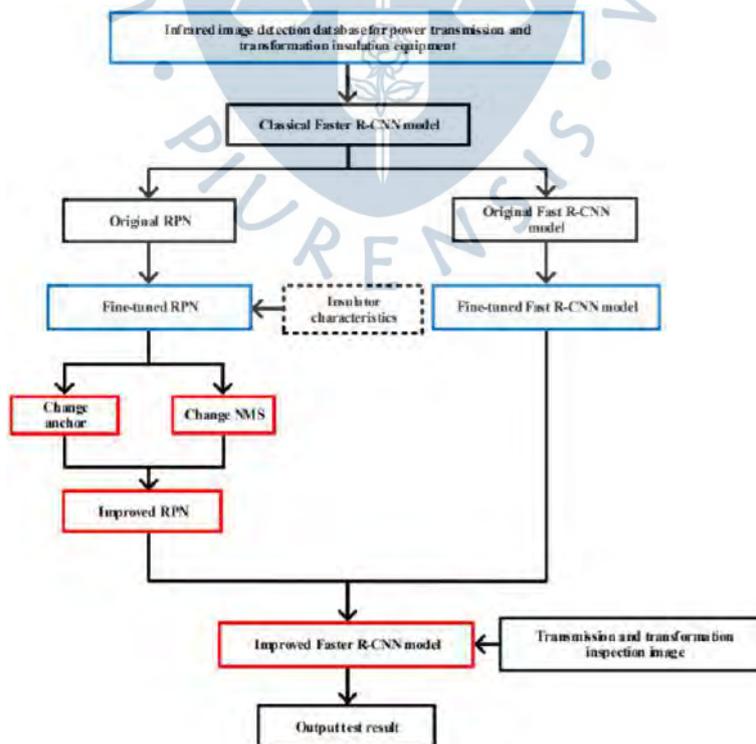
**Fig. 36.** Resultados del modelo con la placa FPGA VC707.



Tomado de “A high-throughput and power-efficient fpga implementation of yolo cnn for object detection” (Nguyen et al., 2019).

Finalmente, se presentó un método de inspección de imágenes de aisladores basado en la red neuronal convolucional R-CNN (Zhao et al., 2019), mediante la construcción de un conjunto de datos de detección de equipos y el uso de fine-tuning del modelo (Fig. 37). Los resultados muestran que el valor de precisión promedio (AP) del modelo se incrementó a 0.818 con el uso de VGG-16.

**Fig. 37.** Framework utilizado.



Tomado de *“Insulator Detection Method in Inspection Image Based on Improved Faster R-CNN”* (Zhao et al., 2019).



## Arquitecturas de detección de objetos

Object detection es una parte de la denominada computer visión o visión por computador, en donde se busca la detección de uno o varios objetos, ya sea en imágenes o videos. Son ejemplos de objetos: personas, animales, edificios, sillas, rocas, etc. La finalidad es responder a las dos preguntas que se plasman en la Fig. 38 (Du, 2018).

**Fig. 38.** *Object detection - preguntas.*

¿Cuál es el objeto?

Identificación del objeto en una imagen.

¿Dónde está el objeto?

Ubicación exacta del objeto en la imagen.

Tomado de “*Understanding of Object Detection Based on CNN Family and YOLO*” (Du, 2018).

El aporte adicional de la tesis se centró en analizar y probar el algoritmo de detección de objetos más utilizado en la revisión literaria del tema: YOLO, además de algunas de sus variaciones (v3, v4, v5, entre otros).

**Fig. 39.** *Ventajas de YOLO.*

Velocidad

Alta, debido a la predicción en tiempo real.

Presición

Alta, debido a su baja tasa de errores y gran nivel de precisión

Aprendizaje

Alta capacidad de aprendizaje en la detección de objetos

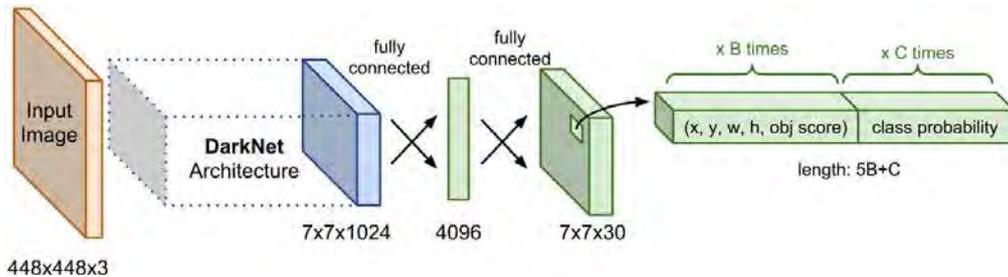
Tomado de “*Understanding of Object Detection Based on CNN Family and YOLO*” (Du, 2018).

YOLO (abreviatura de “Solo miras una vez”, del inglés “You Only Look Once”) utiliza redes neuronales con la finalidad de realizar detección de objetos en tiempo real; es popular gracias a los niveles de precisión y velocidad que obtiene luego de entrenarse. Algunas de sus aplicaciones son la detección de personas, animales, parquímetros o señales de tráfico. En comparación a otros, detecta y reconoce varios objetos en tiempo real en una imagen, para ello realiza la detección como si fuera un problema de regresión, proporcionando probabilidades de cada una de las clases que se detectan en las imágenes, lo cual es realizado en una única ejecución del algoritmo. Mediante la red neuronal convolucional (CNN) se van

prediciendo muchas probabilidades de las clases, además de los cuadros que delimitan a los objetos. Algunas de las ventajas de YOLO se aprecian en la Fig. 39 (Du, 2018).

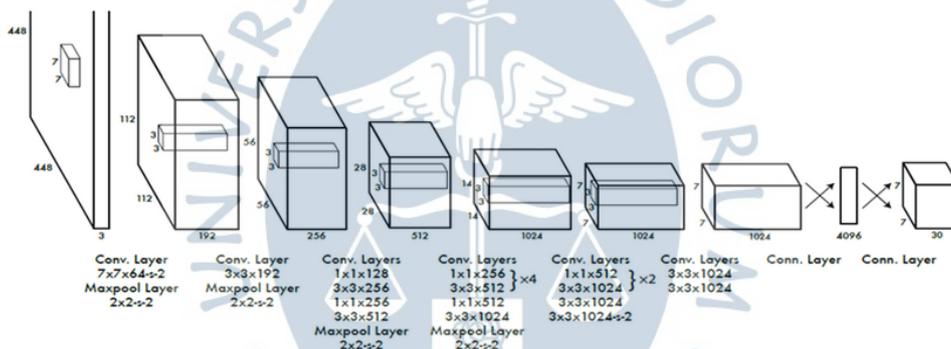
YOLO está definida bajo la arquitectura general que se muestra en la Fig. 40, mientras que en la Fig. 41 se puede apreciar un mayor detalle de la misma.

**Fig. 40.** YOLO – Arquitectura general.



Tomado de “Exploring a Multimodal Mixture-Of-YOLOs Framework for Advanced Real-Time Object Detection” (Kim & Cho, 2020).

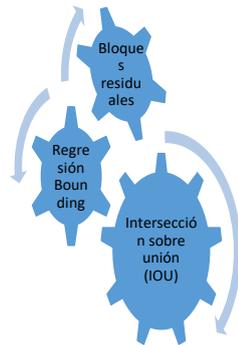
**Fig. 41.** YOLO – Arquitectura detallada.



Tomado de “Exploring a Multimodal Mixture-Of-YOLOs Framework for Advanced Real-Time Object Detection” (Kim & Cho, 2020).

Para lograr su cometido, YOLO utiliza 3 técnicas fundamentalmente (ver Fig. 42): (i) Bloques residuales, mediante la división de la imagen en un conjunto de celdas de una cuadrícula de dimensión de  $S \times S$ , en la que cada celda detectará los objetos que se encuentren en su interior, tal como se aprecia en la Fig. 43 (ii) Regresión de Bounding box, el cual es el contorno que resalta un determinado objeto dentro de una imagen, y presenta atributos como ancho (bw), altura (bh), clase (c) y centro del cuadro delimitador (bx, by), la regresión de un predice la altura, el ancho, el centro y la clase de los objetos (ver Fig. 44), y finalmente (iii) la Intersección sobre unión (IOU), mediante el cual, cada una de las cajas o bounding boxes se superponen, obteniendo una única caja de salida que encierra al objeto en estudio (ver Fig. 45), además de un porcentaje de confianza (G. Karimi, 2021).

**Fig. 42.** YOLO – Técnicas.



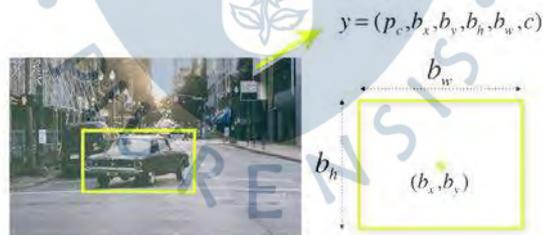
Nota. Elaboración propia a partir de “*Dynamic thermal rating of transmission lines: A review*” (G. Karimi, 2021).

**Fig. 43.** YOLO – Bloques residuales.



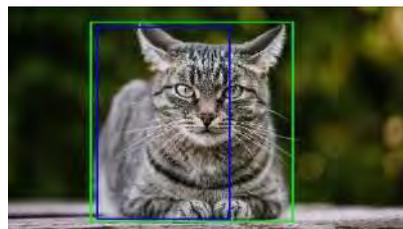
Tomado de “*Dynamic thermal rating of transmission lines: A review*” (G. Karimi, 2021).

**Fig. 44.** YOLO – Regresión Bounding box.



Tomado de “*Dynamic thermal rating of transmission lines: A review*” (G. Karimi, 2021).

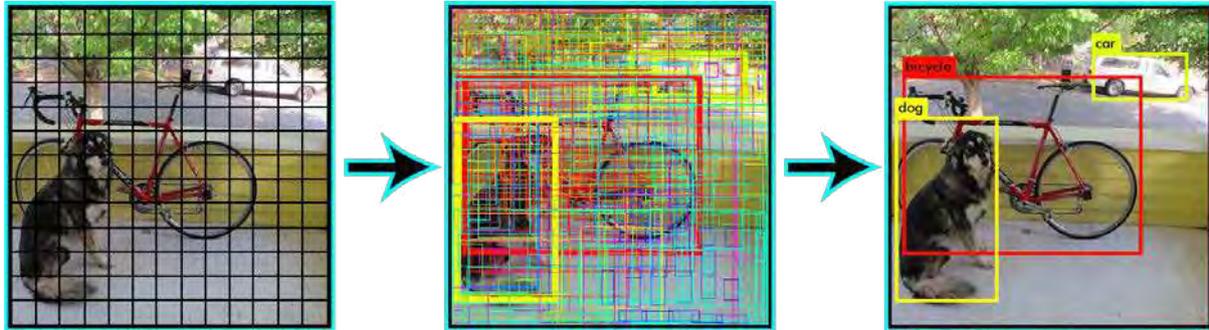
**Fig. 45.** YOLO – Intersección sobre unión (IOU).



Tomado de “*Dynamic thermal rating of transmission lines: A review*” (G. Karimi, 2021).

Al combinar estas 3 técnicas, se obtienen resultados como el que se aprecia en la Fig. 46.

**Fig. 46.** YOLO – Combinación de técnicas.

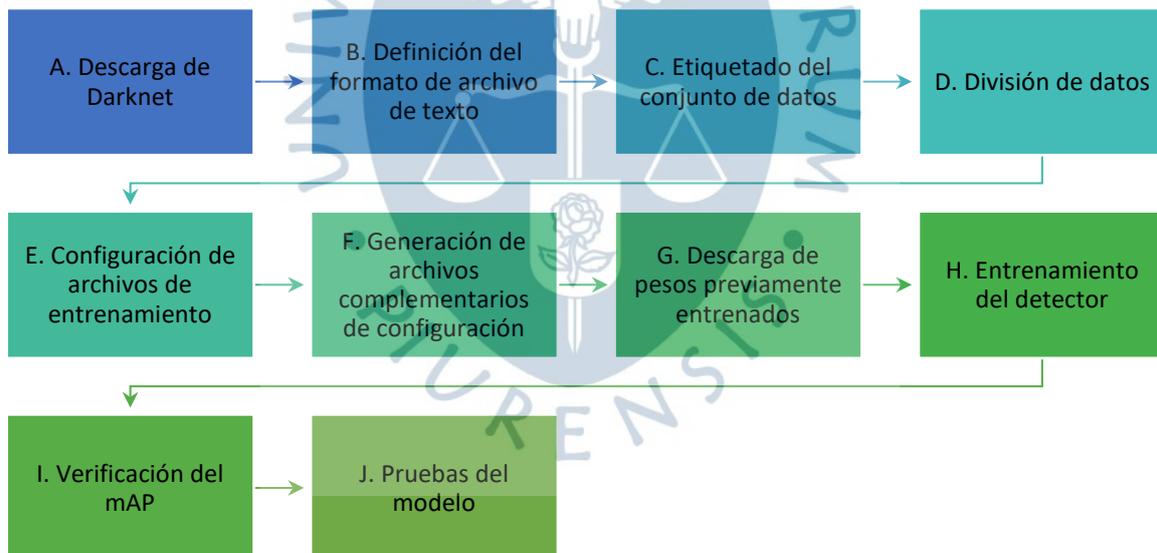


Tomado de “Dynamic thermal rating of transmission lines: A review” (G. Karimi, 2021).

## 1. YOLO v4

La versión 4 del algoritmo YOLO se implementó siguiendo los pasos mostrados en la Fig. 47.

**Fig. 47.** Pasos para la implementación de YOLO v4.



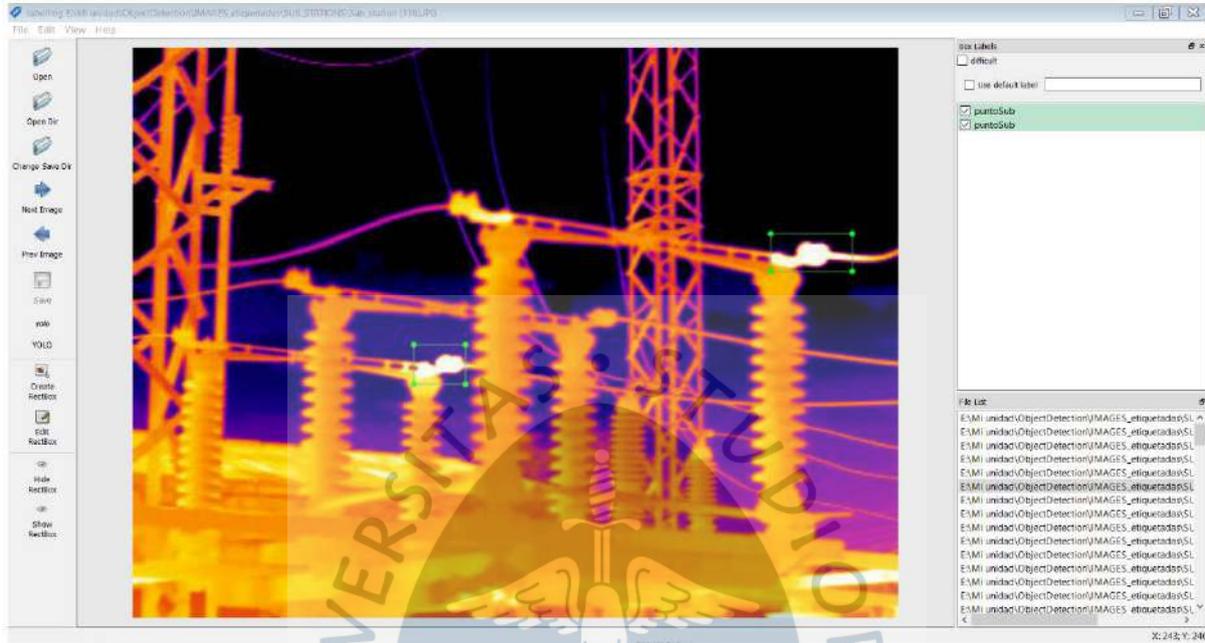
Nota. Elaboración propia.

- A. **Descarga de Darknet.** Darknet es la implementación oficial de YOLO, debido a que se está trabajando con PyTorch, esta es la opción más recomendada, en caso de requerir trabajar con TensorFlow, es necesario la descarga de DarkFlow.
- B. **Definición del formato de archivo de texto.** Se realizó de acuerdo con el siguiente formato:

[número de categoría] [centro del objeto en X] [centro del objeto en Y] [ancho del objeto en X] [ancho del objeto en Y]

- C. **Etiquetado del conjunto de datos.** El set de datos fue etiquetado por un experto en detección de puntos calientes en imágenes térmicas mediante la herramienta Labellmg con el formato YOLO (ver Fig. 48).

**Fig. 48.** *Etiquetado con Labellmg.*

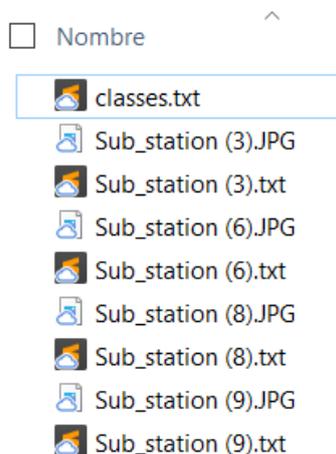


Nota. Elaboración propia.

La definición del formato del paso B, se ve plasmado en esta etapa, ya que cada punto caliente etiquetado genera un archivo de texto asociado a la imagen en estudio en la que se ubica la clase y las coordenadas del objeto etiquetado (ver Fig. 49). En la

Fig. 50 se aprecia el formato relacionado a un punto caliente, mientras que en la Fig. 51, a dos puntos calientes. Este formato se repite en cada fila según el número de puntos calientes etiquetados.

**Fig. 49.** *Archivo de texto – coordenada de los puntos calientes.*



Nota. Elaboración propia.

**Fig. 50.** Archivo de texto – 1 punto caliente.

```
Sub_station (9).txt x
1 0.247396 0.432292 0.161458 0.281250
```

Nota. Elaboración propia.

**Fig. 51.** Archivo de texto – 2 puntos calientes.

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

Nota. Elaboración propia.

- D. **División de datos.** Inicialmente se contaba con 815 imágenes, 342 líneas y 473 subestaciones eléctricas (ver **Tabla 2**), pero luego de la selección de imágenes con puntos calientes, esta cantidad se redujo significativamente a 138, de las cuales 22 corresponden a líneas, y 116 a subestaciones. De las 138 imágenes restantes, se decidió sub dividir el data el set en conjuntos de entrenamiento y validación, obteniendo las cantidades que se detallan en la Tabla I.

**Tabla I.** Object detection - Distribución del dataset.

Clases	Training	Validation	TOTAL
Line	17	5	22
Sub station	93	23	116

Nota. Elaboración propia.

El conjunto de datos de entrenamiento es obligatorio para el modelo, y el de validación fue creado con la finalidad de probar las predicciones luego del entrenamiento del mismo. Debido a que el número de imágenes de líneas es sumamente bajo, se optó por entrenar el modelo y continuar con la investigación únicamente con la base de imágenes de subestaciones eléctricas.

- E. **Configuración de archivos de entrenamiento.** Los archivos configurados fueron:

a. *Custom.cfg*

- i. Batch de = 64 y subdivisiones = 16.
- ii. Las demás configuraciones se realizan en base a la cantidad de clases a entrenar en el detector, en este caso 1 (subestaciones).
  1. max\_batches = 6000

- a. `max_batches = (# de clases) * 2000`. Teniendo como número mínimo a 6000.
- b. `max_batches = 10000`)
2. `steps = 4800, 5400`
  - a. `stpes = (80% de max_batches), (90% de max_batches)` (si tenemos `max_batches = 10000`, entonces `steps = 8000, 9000`)
3. `classes = 1` (subestaciones).
4. Filtros en las 3 capas convolucionales de YOLO.
  - a. `filtros = (# de clases + 5) * 3`. Entonces `filtros = 6*3=18`
- b. *obj.names* Este archivo contiene una línea por cada clase del modelo, en este caso, una, correspondiente a las subestaciones eléctricas (ver Fig. 52).

**Fig. 52.** Archivo de configuración: *obj.names*.

```
obj.names
puntoSub
```

Nota. Elaboración propia.

- c. *obj.data*. Contiene el número de clases y los archivos de texto que se detallan en la etapa F, además de la ruta de ubicación del backup que se generará cada ejecución de 1000 épocas, y en donde se almacenará el mejor modelo obtenido (ver Fig. 53).

**Fig. 53.** Archivo de configuración: *obj.data*.

```
obj.data
classes = 1
train = data/train.txt
valid = data/test.txt
names = data/obj.names
backup = /mydrive/ObjectDetection/YOLO/yolov4_sub_stations/backup
```

Nota. Elaboración propia.

- F. **Generación de archivos complementarios de configuración.** Estos archivos contienen las rutas de todas las imágenes de entrenamiento y validación (ver ejemplo en la Fig. 54) subidos de manera temporal en la máquina virtual de Google Colab.

**Fig. 54.** Archivo de configuración: *train.txt*.

```
train.txt X
data/obj/46bffa67f6381212.jpg
data/obj/76b61bd5960a322f.jpg
data/obj/7bdee48d3c08ecb1.jpg
data/obj/bd68c499ba0b5f85.jpg
data/obj/fcfe48a6bca3e713.jpg
```

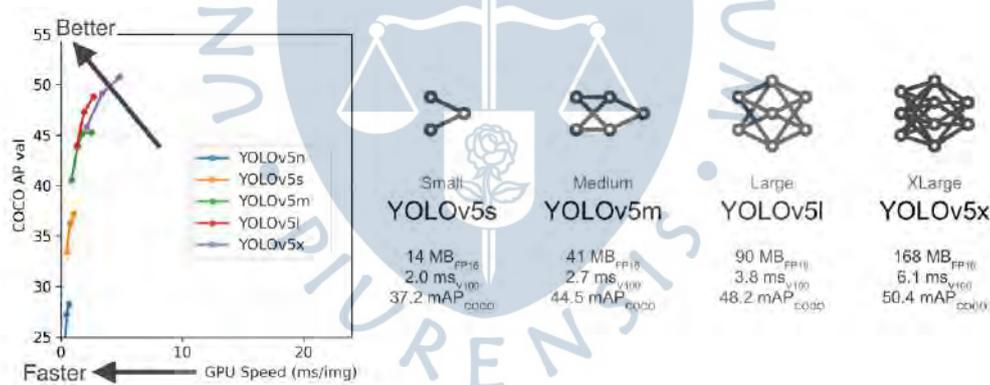
Nota. Elaboración propia.

- G. **Descarga de pesos previamente entrenados.** Con la finalidad de obtener mejores resultados, se recomienda descargar los pesos de las capas convolucionales de YOLO v4, esto ayuda a que el detector sea mucho más preciso y se logre un ahorro de tiempo en el entrenamiento. El tamaño aproximado es de 162.16 MB.
- H. **Entrenamiento del detector.** Con las configuraciones de las etapas anteriores se procedió a entrenar y monitorear la ejecución del modelo.
- I. **Verificación del mAP.** Luego del entrenamiento del modelo se analizan métricas y gráficos como el mAP (Mean Average Precision).
- J. **Pruebas del modelo.** La configuración del paso E permite elegir la ruta donde se almacenan las copias de seguridad cada 1000 épocas, es por ello que en la etapa de pruebas podemos elegir alguno de ellos o el mejor modelo almacenado.

## 2. YOLOv5

El algoritmo YOLOv5 dispone de distintas variaciones (Jocher Glenn, 2021), entre las que se encuentra a YOLOv5 small (YOLOv5s), YOLOv5 medium (YOLOv5m), YOLOv5 large (YOLOv5l), YOLOv5 extra large (YOLOv5x), entre otras, tal como se aprecia en la Fig. 55.

**Fig. 55.** Variaciones del algoritmo YOLOv5.



Adaptado de “YOLOv5 in PyTorch” (Jocher Glenn, 2021).

La implementación de YOLOv5 se realizó siguiendo los pasos detallados anteriormente, sin embargo, la aplicación de data augmentation (Fig. 56) y la configuración del formato de texto de las coordenadas (Fig. 57) de cada punto caliente se realizó mediante la plataforma Roboflow,

**Fig. 56.** Data augmentation YOLOv5 - Roboflow.

AUGMENTATIONS

Outputs per training example: 3

Flip: Horizontal, Vertical

90° Rotate: Clockwise, Counter-Clockwise, Upside Down

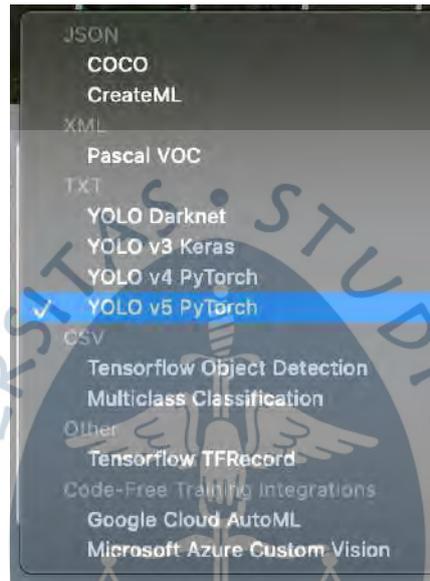
Crop: 0% Minimum Zoom, 20% Maximum Zoom

Rotation: Between -15° and +15°

Shear: ±15° Horizontal, ±15° Vertical

Nota. Elaboración propia.

**Fig. 57.** Formato para la implementación de YOLOv5 - Roboflow.



Nota. Elaboración propia.

Luego de haber aplicado las transformaciones mostradas anteriormente, se obtuvo un set de datos con 278 imágenes (ver Fig. 58).

**Fig. 58.** Set de datos YOLOv5 - Roboflow.



Nota. Elaboración propia.

## Resultados

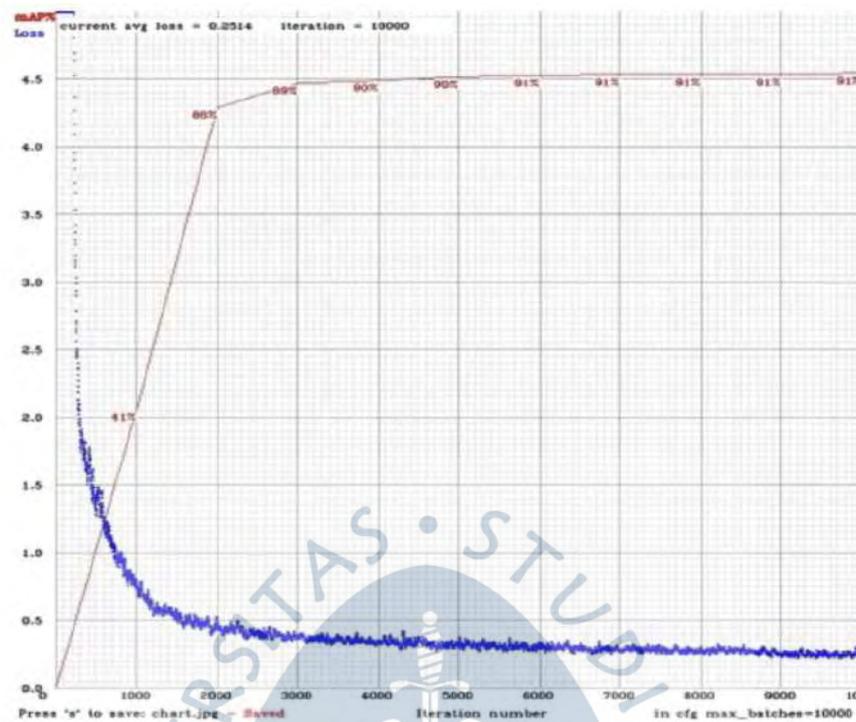
### 1. YOLOv4 – Resultados

Luego de la ejecución del modelo YOLO v4 configurado mediante los pasos mostrados

previamente, se debió obtener un resultado gráfico similar al mostrado en la Fig. 59, sin embargo, debido a las limitaciones del uso gratuito de Google Colab, este deja de ejecutarse cada cierto tiempo y no permite graficar los resultados de manera completa.



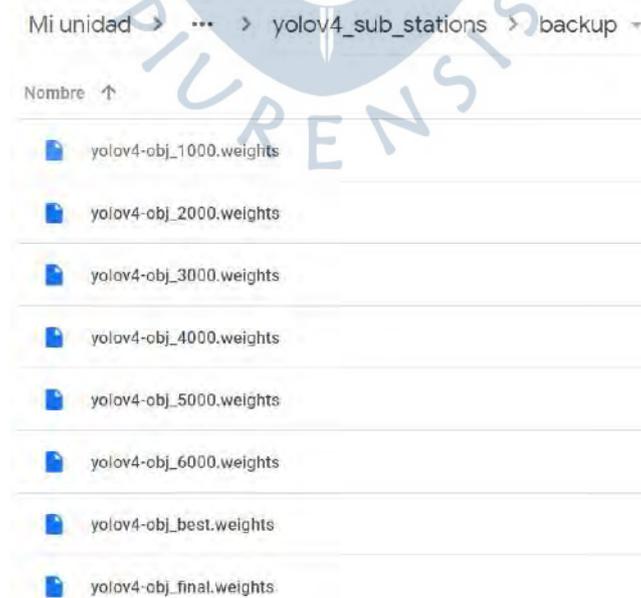
Fig. 59. YOLO v4 - Resultados esperados.



Nota. Elaboración propia.

Es por este motivo que se almacenaron los modelos y sus pesos cada 1000 épocas, además de guardar el último en ejecución y el que obtuvo mejores resultados, con la finalidad de continuar la ejecución y no perder resultados debido a las limitaciones de Google Colab (ver Fig. 60).

Fig. 60. YOLO v4 – Backup de modelos.



Nota. Elaboración propia.

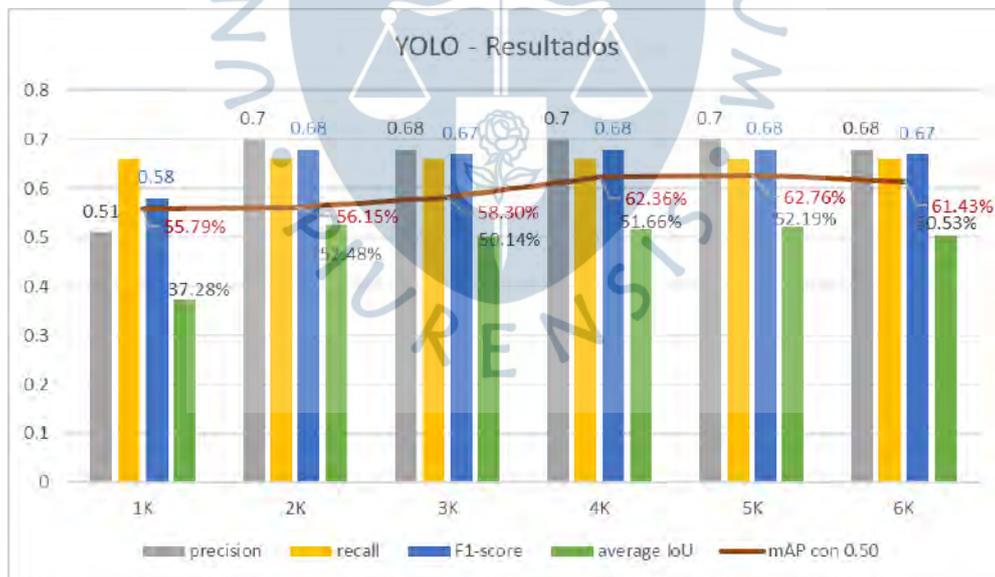
En base a estos modelos se procedió a obtener métricas como Precision, Recall, F1-score, average IoU y mAP, los cuales se plasman en la Tabla II y Fig. 61. Finalmente, en Fig. 62 y Fig. 63 se evidencia la detección de puntos calientes en diversas imágenes térmicas de subestaciones eléctricas.

**Tabla II. YOLO v4 – Resultados por época.**

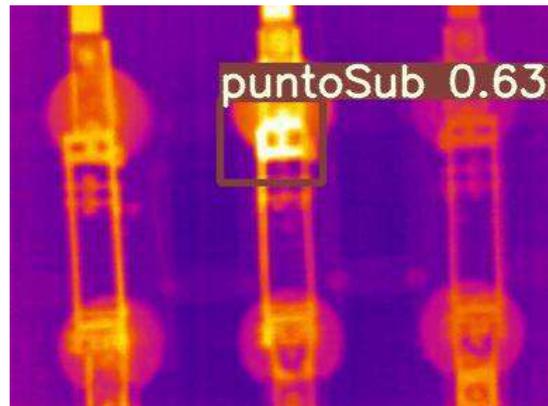
Arquitectura	1K	2K	3K	4K	5K	6K
precision	0.51	0.7	0.68	0.7	0.7	0.68
recall	0.66	0.66	0.66	0.66	0.66	0.66
F1-score	0.58	0.68	0.67	0.68	0.68	0.67
average IoU	37.28%	52.48%	50.14%	51.66%	52.19%	50.53%
mAP con 0.50	55.79%	56.15%	58.30%	62.36%	62.76%	61.43%

Nota. Elaboración propia.

**Fig. 61. YOLO v4 – Resultados.**



Nota. Elaboración propia.

**Fig. 62.** YOLO v4 – Detección de puntos calientes (1).

Nota. Elaboración propia.

**Fig. 63.** YOLO v4 – Detección de puntos calientes (2).

Nota. Elaboración propia.

## 2. YOLOv5 - Resultados

Se realizaron experimentos con cada variación del algoritmo YOLOv5 haciendo un total de 64 experimentos (16 con YOLOv5 small, 16 con YOLOv5 medium, 16 con YOLOv5 large, y 16 con YOLOv5 extra-large). Se consideraron 200 épocas, distintos tamaños de batch, además del uso de data augmentation y transfer learning mediante los pesos previamente entrenados de YOLOv5, tal como se aprecia en la Tabla III.

**Tabla III.** YOLOv5 - Experimentos realizados.

Exp.	Algoritmo	Batch size	Data augmentation	Transfer learning
E1	YOLOv5s (small)	4	No	No
E2	YOLOv5s (small)	8	No	No

Exp.	Algoritmo	Batch size	Data augmentation	Transfer learning
E3	YOLOv5s (small)	16	No	No
E4	YOLOv5s (small)	32	No	No
E5	YOLOv5s (small)	4	No	Yes
E6	YOLOv5s (small)	8	No	Yes
E7	YOLOv5s (small)	16	No	Yes
E8	YOLOv5s (small)	32	No	Yes
E9	YOLOv5s (small)	4	Yes	No
E10	YOLOv5s (small)	8	Yes	No
E11	YOLOv5s (small)	16	Yes	No
E12	YOLOv5s (small)	32	Yes	No
E13	YOLOv5s (small)	4	Yes	Yes
E14	YOLOv5s (small)	8	Yes	Yes
E15	YOLOv5s (small)	16	Yes	Yes
E16	YOLOv5s (small)	32	Yes	Yes
E17	YOLOv5m (medium)	4	No	No
E18	YOLOv5m (medium)	8	No	No
E19	YOLOv5m (medium)	16	No	No
E20	YOLOv5m (medium)	32	No	No
E21	YOLOv5m (medium)	4	No	Yes
E22	YOLOv5m (medium)	8	No	Yes
E23	YOLOv5m (medium)	16	No	Yes
E24	YOLOv5m (medium)	32	No	Yes

Exp.	Algoritmo	Batch size	Data augmentation	Transfer learning
E25	YOLOv5m (medium)	4	Yes	No
E26	YOLOv5m (medium)	8	Yes	No
E27	YOLOv5m (medium)	16	Yes	No
E28	YOLOv5m (medium)	32	Yes	No
E29	YOLOv5m (medium)	4	Yes	Yes
E30	YOLOv5m (medium)	8	Yes	Yes
E31	YOLOv5m (medium)	16	Yes	Yes
E32	YOLOv5m (medium)	32	Yes	Yes
E33	YOLOv5l (large)	4	No	No
E34	YOLOv5l (large)	8	No	No
E35	YOLOv5l (large)	16	No	No
E36	YOLOv5l (large)	32	No	No
E37	YOLOv5l (large)	4	No	Yes
E38	YOLOv5l (large)	8	No	Yes
E39	YOLOv5l (large)	16	No	Yes
E40	YOLOv5l (large)	32	No	Yes
E41	YOLOv5l (large)	4	Yes	No
E42	YOLOv5l (large)	8	Yes	No
E43	YOLOv5l (large)	16	Yes	No
E44	YOLOv5l (large)	32	Yes	No
E45	YOLOv5l (large)	4	Yes	Yes
E46	YOLOv5l (large)	8	Yes	Yes

Exp.	Algoritmo	Batch size	Data augmentation	Transfer learning
E47	YOLOv5l (large)	16	Yes	Yes
E48	YOLOv5l (large)	32	Yes	Yes
E49	YOLOv5xl (extra large)	4	No	No
E50	YOLOv5xl (extra large)	8	No	No
E51	YOLOv5xl (extra large)	16	No	No
E52	YOLOv5xl (extra large)	32	No	No
E53	YOLOv5xl (extra large)	4	No	Yes
E54	YOLOv5xl (extra large)	8	No	Yes
E55	YOLOv5xl (extra large)	16	No	Yes
E56	YOLOv5xl (extra large)	32	No	Yes
E57	YOLOv5xl (extra large)	4	Yes	No
E58	YOLOv5xl (extra large)	8	Yes	No
E59	YOLOv5xl (extra large)	16	Yes	No
E60	YOLOv5xl (extra large)	32	Yes	No
E61	YOLOv5xl (extra large)	4	Yes	Yes
E62	YOLOv5xl (extra large)	8	Yes	Yes
E63	YOLOv5xl (extra large)	16	Yes	Yes
E64	YOLOv5xl (extra large)	32	Yes	Yes

Nota. Elaboración propia.

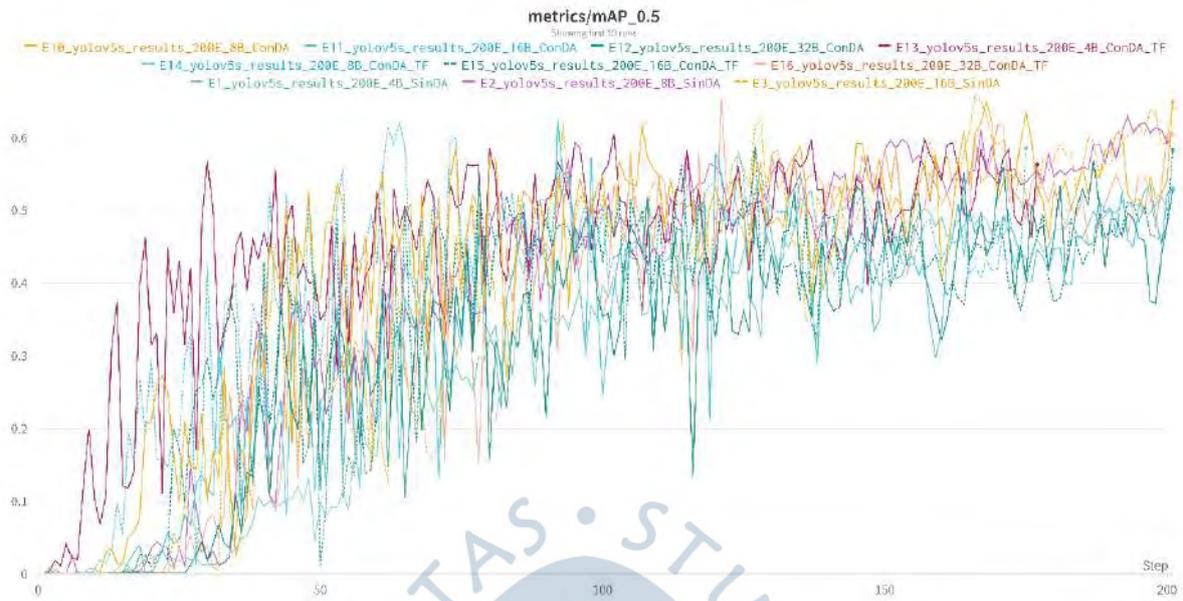
La Tabla IV muestra el detalle de la tasa de pérdida, precisión, recall, F1-score, mAP y la época en la que obtuvo el mejor mAP con el algoritmo YOLOv5s. El detalle de cada resultado se observa en la Fig. 64, Fig. 65, Fig. 65, Fig. 66 y Fig. 67.

**Tabla IV.** YOLOv5 small – Resultados.

Exp.	Loss	Precision	Recall	F1-score	mAP	Mejor época
E1	0.05484	47.16%	67.57%	55.55%	50.26%	157
E2	0.05137	75.97%	51.35%	61.28%	59.77%	153
E3	0.04212	59.97%	64.86%	62.32%	65.08%	167
E4	0.04340	55.95%	75.68%	64.34%	64.70%	143
E5	0.04060	60.40%	78.38%	68.23%	69.42%	171
E6	0.03988	59.17%	78.33%	67.42%	65.44%	190
E7	0.04106	79.98%	53.99%	64.47%	68.99%	146
E8	0.04953	57.57%	70.27%	63.29%	57.24%	193
E9	0.04657	80.00%	54.05%	64.51%	64.65%	115
E10	0.03710	57.43%	72.97%	64.27%	64.96%	168
E11	0.03740	56.39%	59.46%	57.89%	53.07%	134
E12	0.03750	55.26%	56.76%	56.00%	52.85%	185
E13	0.03718	67.72%	56.70%	61.72%	56.30%	76
E14	0.03360	57.12%	64.80%	60.72%	59.94%	74
E15	0.02952	58.31%	75.68%	65.87%	58.82%	127
E16	0.03107	58.47%	64.86%	61.50%	60.52%	200

Nota. Elaboración propia.

**Fig. 64.** YOLOv5 small – mAP.



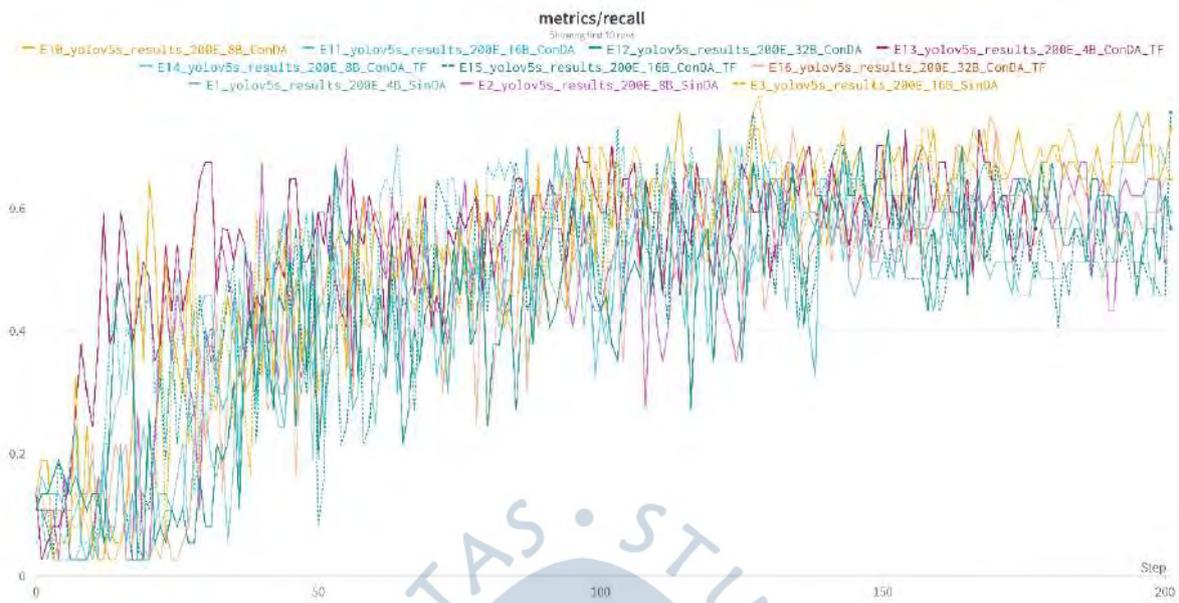
Nota. Elaboración propia.

**Fig. 65.** YOLOv5 small – Precision.



Nota. Elaboración propia.

**Fig. 66.** YOLOv5 small – Recall.



Nota. Elaboración propia.

**Fig. 67.** YOLOv5 small – Loss.



Nota. Elaboración propia.

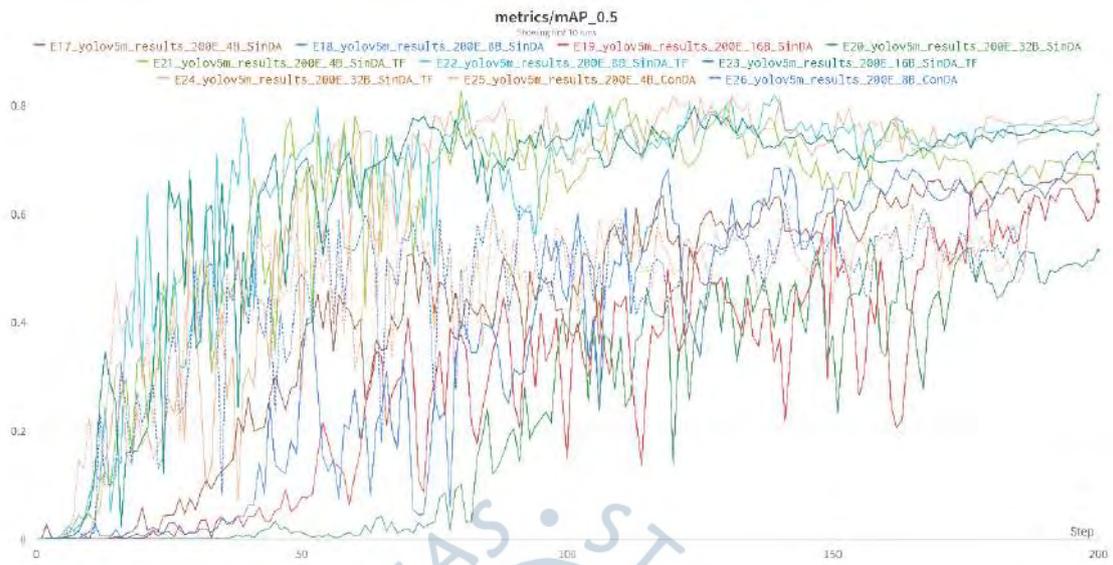
La Tabla V muestra el detalle de la tasa de pérdida, precisión, recall, F1-score, mAP y la época en la que obtuvo el mejor mAP con el algoritmo YOLOv5m. El detalle de cada resultado se observa en la Fig. 68, Fig. 69, Fig. 70 y Fig. 71.

**Tabla V.** YOLOv5 medium – Resultados.

Exp.	Loss	Precision	Recall	F1-score	mAP	Mejor época
E17	0.05050	59.01%	70.27%	64.15%	62.57%	160
E18	0.05133	86.13%	51.35%	64.34%	68.49%	139
E19	0.04590	70.27%	70.26%	70.27%	64.73%	194
E20	0.05628	65.71%	62.16%	63.89%	53.40%	175
E21	0.04042	70.72%	78.38%	74.35%	72.87%	107
E22	0.04541	70.21%	89.18%	78.57%	81.99%	139
E23	0.04426	74.34%	78.31%	76.28%	75.69%	182
E24	0.04362	74.28%	70.24%	72.20%	75.97%	154
E25	0.05280	84.05%	45.95%	59.41%	61.87%	86
E26	0.04952	88.62%	43.24%	58.12%	61.65%	86
E27	0.05051	58.95%	62.16%	60.51%	62.80%	88
E28	0.04908	49.97%	75.68%	60.19%	63.09%	63
E29	0.05071	91.60%	59.46%	72.11%	73.20%	108
E30	0.04914	64.82%	64.86%	64.84%	66.17%	69
E31	0.04743	69.42%	67.50%	68.45%	68.35%	37
E32	0.04665	81.42%	59.46%	68.73%	70.96%	63

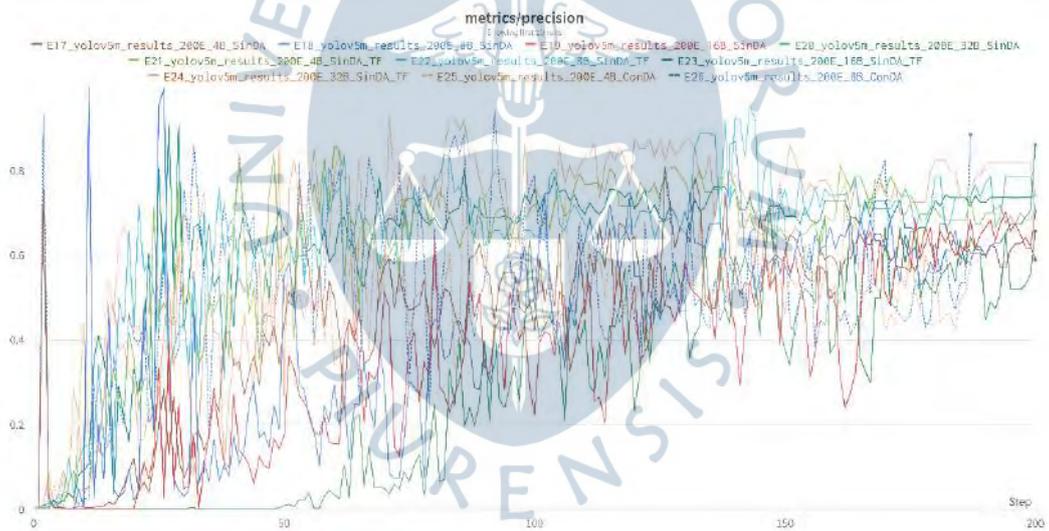
Nota. Elaboración propia.

**Fig. 68.** YOLOv5 medium – mAP.



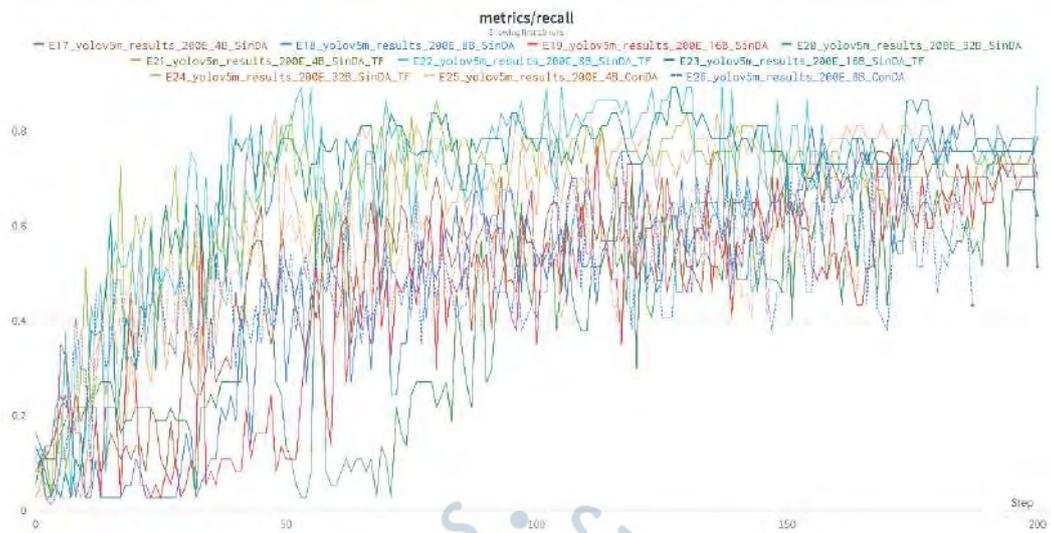
Nota. Elaboración propia.

**Fig. 69.** YOLOv5 medium – Precision.



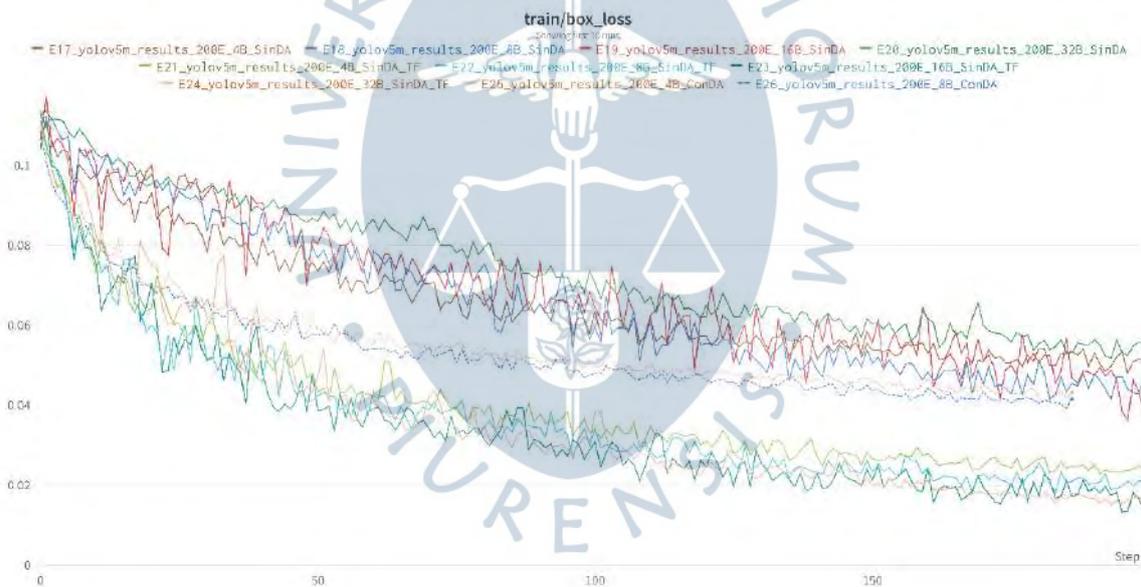
Nota. Elaboración propia.

**Fig. 70.** YOLOv5 medium – Recall.



Nota. Elaboración propia.

**Fig. 71.** YOLOv5 medium – Loss.



Nota. Elaboración propia.

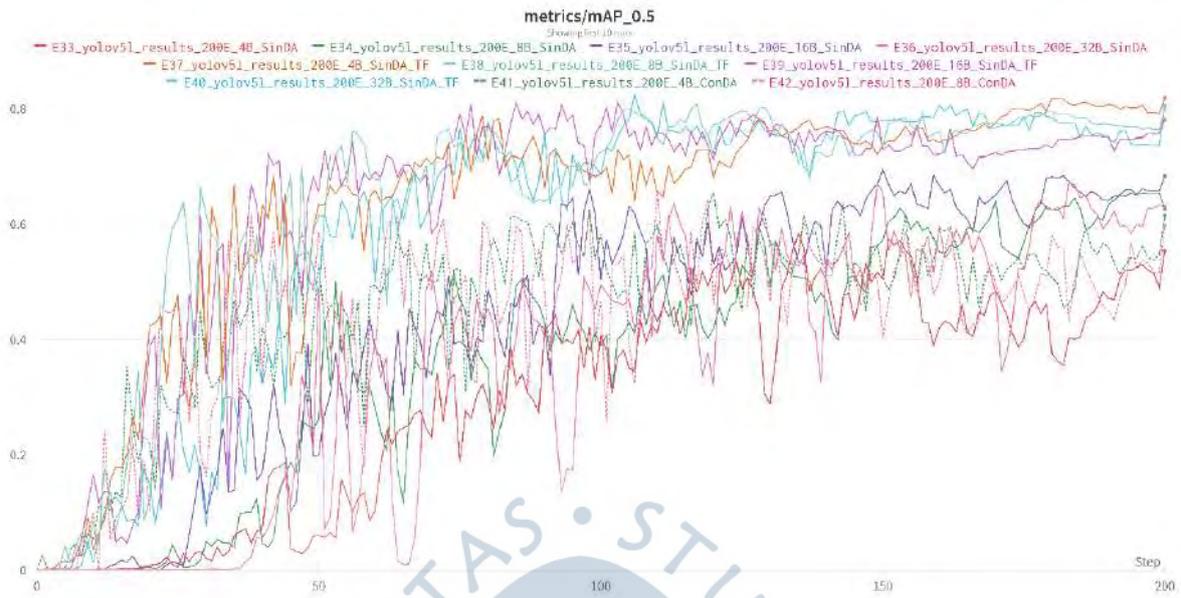
La Tabla VI muestra el detalle de la tasa de pérdida, precisión, recall, F1-score, mAP y la época en la que obtuvo el mejor mAP con el algoritmo YOLOv5I. El detalle de cada resultado se observa en la Fig. 72, Fig. 73, Fig. 74 y Fig. 75.

**Tabla VI.** YOLOv5 large – Resultados.

Exp.	Loss	Precision	Recall	F1-score	Map	Mejor época
E33	0.05487	80.70%	45.95%	58.55%	55.38%	154
E34	0.04841	60.97%	67.57%	64.10%	62.77%	153
E35	0.04331	55.97%	75.68%	64.35%	68.42%	182
E36	0.03910	61.86%	70.13%	65.74%	63.07%	123
E37	0.01974	89.79%	72.97%	80.51%	81.88%	180
E38	0.01602	73.80%	83.78%	78.48%	79.98%	128
E39	0.01615	68.11%	81.08%	74.03%	78.10%	125
E40	0.01396	89.18%	67.57%	76.88%	80.56%	160
E41	0.04325	57.14%	64.86%	60.76%	61.48%	136
E42	0.03757	73.04%	51.27%	60.25%	59.81%	162
E43	0.03571	52.99%	70.27%	60.42%	63.03%	60
E44	0.03288	49.12%	75.65%	59.56%	65.17%	126
E45	0.01628	73.53%	67.56%	70.42%	71.16%	182
E46	0.01431	80.47%	78.38%	79.41%	77.68%	124
E47	0.01702	82.20%	64.86%	72.51%	71.97%	43
E48	0.01196	69.22%	72.95%	71.04%	70.16%	165

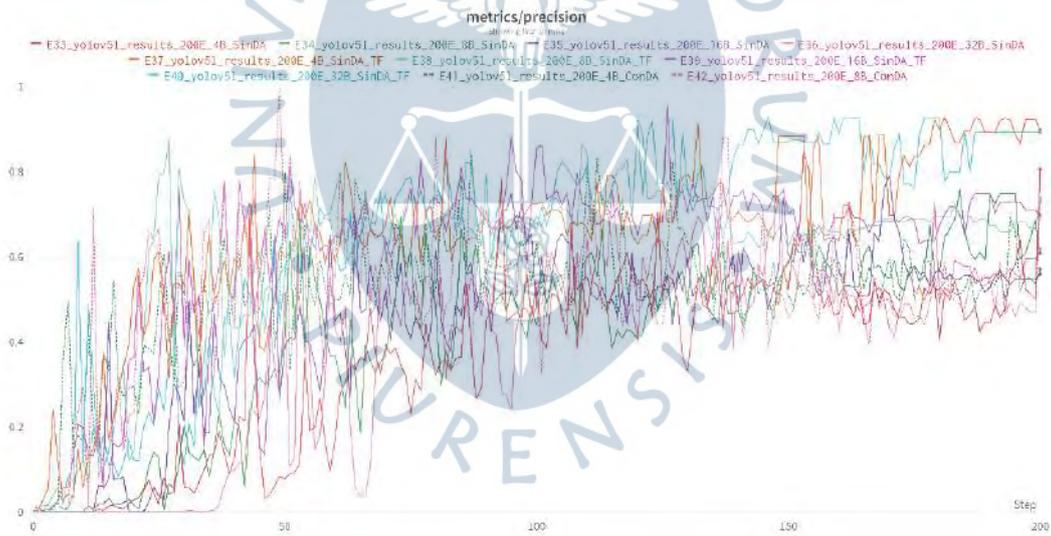
Nota. Elaboración propia.

**Fig. 72.** YOLOv5 large – mAP.



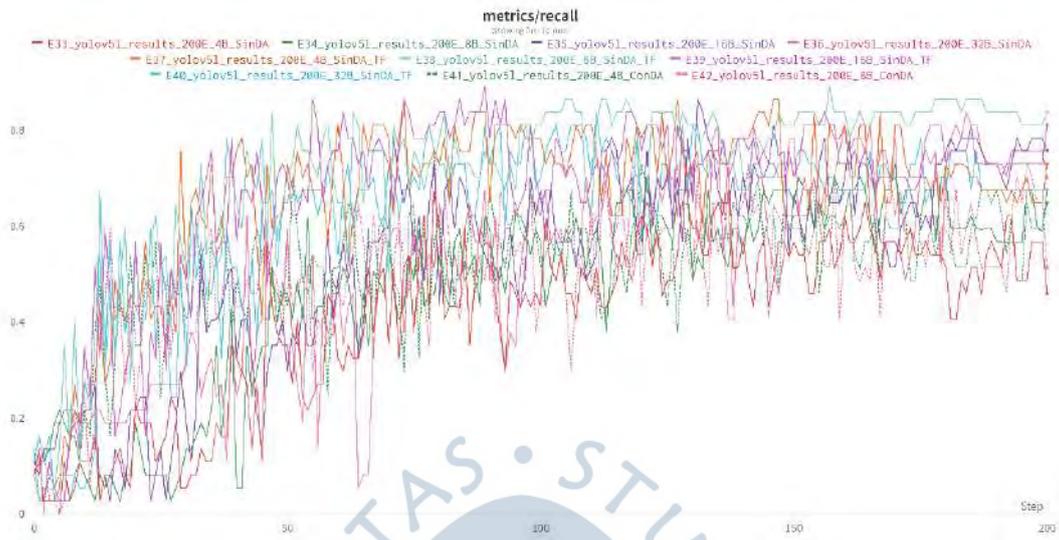
Elaboración propia.

**Fig. 73.** YOLOv5 large – Precision.



Nota. Elaboración propia.

**Fig. 74.** YOLOv5 large – Recall.



Nota. Elaboración propia.

**Fig. 75.** YOLOv5 large – Loss.



Nota. Elaboración propia.

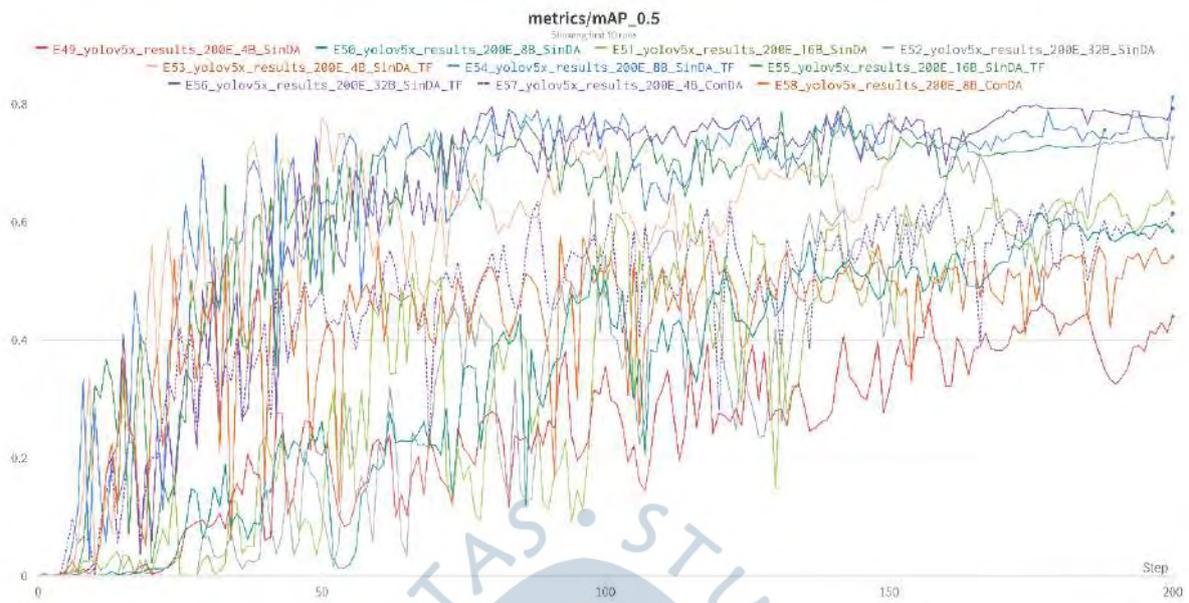
La Tabla VII muestra el detalle de la tasa de pérdida, precisión, recall, F1-score, mAP y la época en la que obtuvo el mejor mAP con el algoritmo YOLOv5xl. El detalle de cada resultado se observa en la Fig. 76, Fig. 77, Fig. 78 y Fig. 79.

**Tabla VII.** YOLOv5 extra-large – Resultados.

Exp.	Loss	Precision	Recall	F1-score	mAP	Mejor época
E49	0.06242	51.32%	51.28%	51.30%	44.01%	184
E50	0.04919	53.29%	64.86%	58.51%	58.92%	198
E51	0.04517	62.42%	67.57%	64.89%	63.48%	198
E52	0.04052	67.51%	67.57%	67.54%	74.29%	196
E53	0.02470	77.04%	72.56%	74.73%	77.62%	50
E54	0.01478	72.09%	83.70%	77.46%	78.78%	178
E55	0.01468	69.76%	81.06%	74.99%	75.57%	87
E56	0.01247	73.17%	81.08%	76.92%	79.25%	178
E57	0.04357	57.12%	64.86%	60.74%	61.42%	101
E58	0.04474	49.99%	64.86%	56.47%	54.13%	181
E59	0.03493	56.51%	70.27%	62.64%	64.20%	156
E60	0.03677	77.26%	45.91%	57.59%	58.66%	98
E61	0.01463	70.96%	59.45%	64.70%	63.36%	165
E62	0.01390	76.66%	62.16%	68.65%	68.92%	89
E63	0.01015	71.42%	81.03%	75.92%	73.42%	109
E64	0.01134	73.78%	83.78%	78.46%	77.01%	92

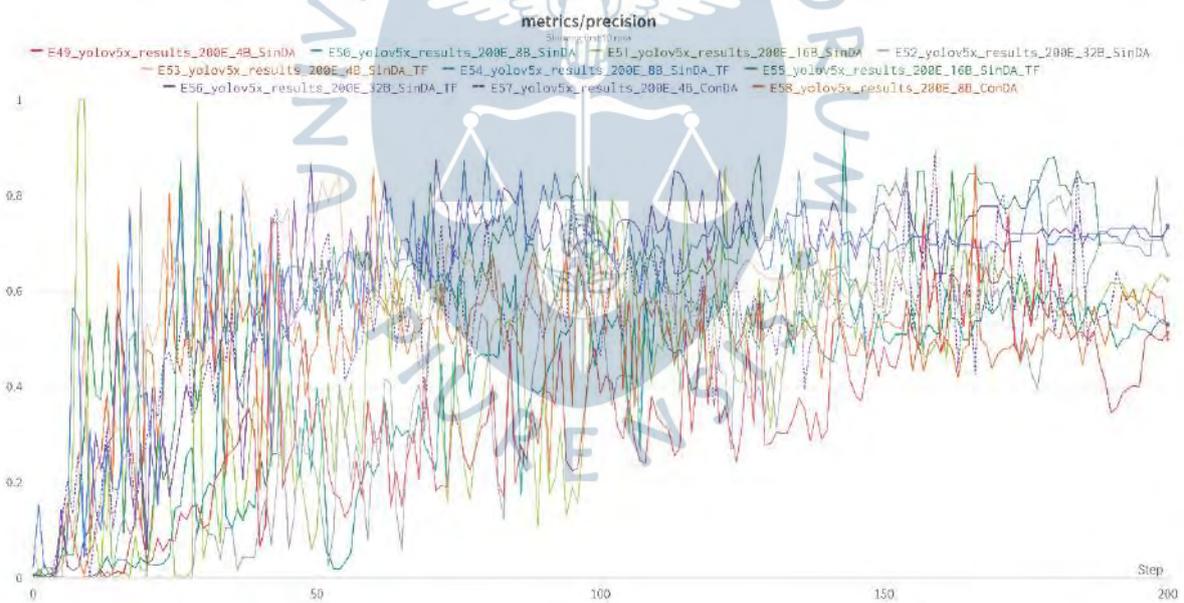
Nota. Elaboración propia.

**Fig. 76.** YOLOv5 extra-large – mAP.



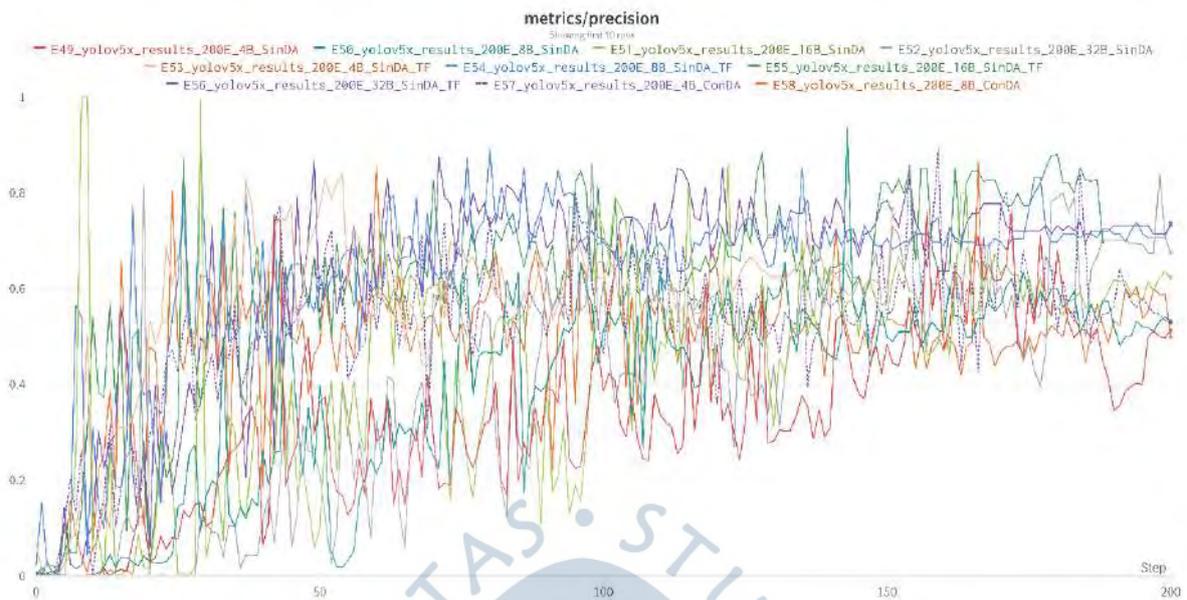
Nota. Elaboración propia.

**Fig. 77.** YOLOv5 extra-large – Precision.



Nota. Elaboración propia.

**Fig. 78.** YOLOv5 extra-large – Recall.



Nota. Elaboración propia.

**Fig. 79.** YOLOv5 extra-large – Loss.



Nota. Elaboración propia.

En base a estos resultados, se aprecia que en la mayoría de los experimentos el uso de data augmentation afecta el entrenamiento, y la técnica de transfer learning contribuye positivamente a que el modelo se entrene de manera más favorable, obteniendo un valor de mAP de 81.99%. La Tabla VIII muestra un ranking de los resultados obtenidos en los 64 experimentos y la Tabla IX, el mejor resultado de cada variación del algoritmo.

**Tabla VIII.** YOLOv5 – Cuadro resumen de resultados.

Exp.	Ranking	Loss	Precision	Recall	F1-score	mAP	Mejor época
E1	63	0.05484	47.16%	67.57%	55.55%	50.26%	157
E2	52	0.05137	75.97%	51.35%	61.28%	59.77%	153
E3	31	0.04212	59.97%	64.86%	62.32%	65.08%	167
E4	34	0.04340	55.95%	75.68%	64.34%	64.70%	143
E5	22	0.04060	60.40%	78.38%	68.23%	69.42%	171
E6	29	0.03988	59.17%	78.33%	67.42%	65.44%	190
E7	23	0.04106	79.98%	53.99%	64.47%	68.99%	146
E8	56	0.04953	57.57%	70.27%	63.29%	57.24%	193
E9	35	0.04657	80.00%	54.05%	64.51%	64.65%	115
E10	32	0.03710	57.43%	72.97%	64.27%	64.96%	168
E11	61	0.03740	56.39%	59.46%	57.89%	53.07%	134
E12	62	0.03750	55.26%	56.76%	56.00%	52.85%	185
E13	57	0.03718	67.72%	56.70%	61.72%	56.30%	76
E14	50	0.03360	57.12%	64.80%	60.72%	59.94%	74
E15	54	0.02952	58.31%	75.68%	65.87%	58.82%	127
E16	49	0.03107	58.47%	64.86%	61.50%	60.52%	200
E17	44	0.05050	59.01%	70.27%	64.15%	62.57%	160
E18	25	0.05133	86.13%	51.35%	64.34%	68.49%	139
E19	33	0.04590	70.27%	70.26%	70.27%	64.73%	194
E20	60	0.05628	65.71%	62.16%	63.89%	53.40%	175
E21	17	0.04042	70.72%	78.38%	74.35%	72.87%	107

Exp.	Ranking	Loss	Precision	Recall	F1-score	mAP	Mejor época
E22	1	0.04541	70.21%	89.18%	78.57%	81.99%	139
E23	12	0.04426	74.34%	78.31%	76.28%	75.69%	182
E24	11	0.04362	74.28%	70.24%	72.20%	75.97%	154
E25	45	0.05280	84.05%	45.95%	59.41%	61.87%	86
E26	46	0.04952	88.62%	43.24%	58.12%	61.65%	86
E27	42	0.05051	58.95%	62.16%	60.51%	62.80%	88
E28	39	0.04908	49.97%	75.68%	60.19%	63.09%	63
E29	16	0.05071	91.60%	59.46%	72.11%	73.20%	108
E30	28	0.04914	64.82%	64.86%	64.84%	66.17%	69
E31	27	0.04743	69.42%	67.50%	68.45%	68.35%	37
E32	20	0.04665	81.42%	59.46%	68.73%	70.96%	63
E33	58	0.05487	80.70%	45.95%	58.55%	55.38%	154
E34	43	0.04841	60.97%	67.57%	64.10%	62.77%	153
E35	26	0.04331	55.97%	75.68%	64.35%	68.42%	182
E36	40	0.03910	61.86%	70.13%	65.74%	63.07%	123
E37	2	0.01974	89.79%	72.97%	80.51%	81.88%	180
E38	4	0.01602	73.80%	83.78%	78.48%	79.98%	128
E39	7	0.01615	68.11%	81.08%	74.03%	78.10%	125
E40	3	0.01396	89.18%	67.57%	76.88%	80.56%	160
E41	47	0.04325	57.14%	64.86%	60.76%	61.48%	136
E42	51	0.03757	73.04%	51.27%	60.25%	59.81%	162
E43	41	0.03571	52.99%	70.27%	60.42%	63.03%	60

Exp.	Ranking	Loss	Precision	Recall	F1-score	mAP	Mejor época
E44	30	0.03288	49.12%	75.65%	59.56%	65.17%	126
E45	19	0.01628	73.53%	67.56%	70.42%	71.16%	182
E46	8	0.01431	80.47%	78.38%	79.41%	77.68%	124
E47	18	0.01702	82.20%	64.86%	72.51%	71.97%	43
E48	21	0.01196	69.22%	72.95%	71.04%	70.16%	165
E49	64	0.06242	51.32%	51.28%	51.30%	44.01%	184
E50	53	0.04919	53.29%	64.86%	58.51%	58.92%	198
E51	37	0.04517	62.42%	67.57%	64.89%	63.48%	198
E52	14	0.04052	67.51%	67.57%	67.54%	74.29%	196
E53	9	0.02470	77.04%	72.56%	74.73%	77.62%	50
E54	6	0.01478	72.09%	83.70%	77.46%	78.78%	178
E55	13	0.01468	69.76%	81.06%	74.99%	75.57%	87
E56	5	0.01247	73.17%	81.08%	76.92%	79.25%	178
E57	48	0.04357	57.12%	64.86%	60.74%	61.42%	101
E58	59	0.04474	49.99%	64.86%	56.47%	54.13%	181
E59	36	0.03493	56.51%	70.27%	62.64%	64.20%	156
E60	55	0.03677	77.26%	45.91%	57.59%	58.66%	98
E61	38	0.01463	70.96%	59.45%	64.70%	63.36%	165
E62	24	0.01390	76.66%	62.16%	68.65%	68.92%	89
E63	15	0.01015	71.42%	81.03%	75.92%	73.42%	109
E64	10	0.01134	73.78%	83.78%	78.46%	77.01%	92

Nota. Elaboración propia.

**Tabla IX.** YOLOv5 – Mejor resultado de cada variación del algoritmo.

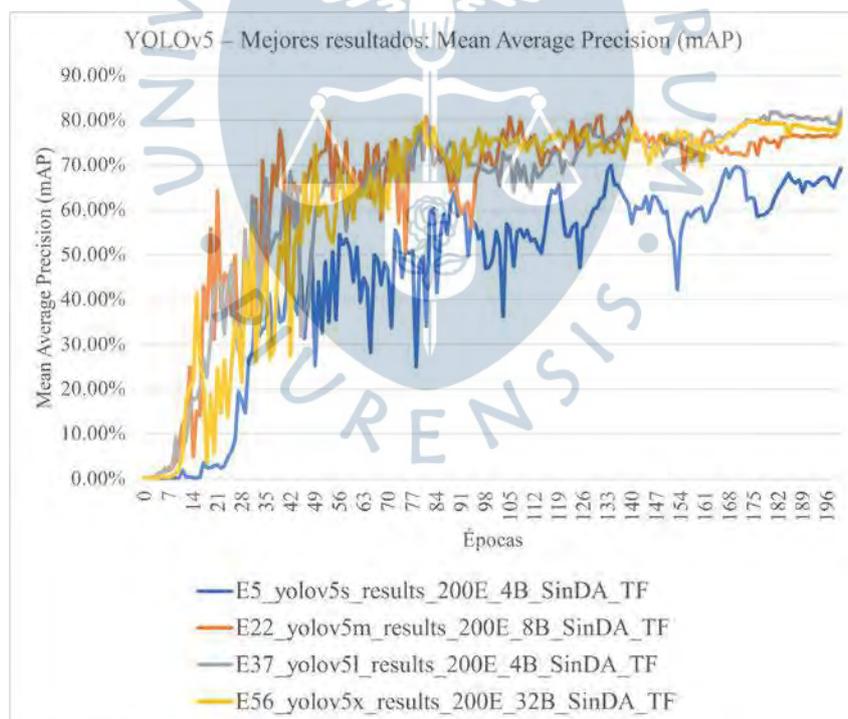
Algoritmo	Exp.	Loss	Precision	Recall	F1-score	Map	Mejor época
YOLOv5s	E5	0.04060	60.40%	78.38%	68.23%	69.42%	171
YOLOv5m	E22	0.04541	70.21%	89.18%	78.57%	81.99%	139
YOLOv5l	E37	0.01974	89.79%	72.97%	80.51%	81.88%	180
YOLOv5xl	E56	0.01247	73.17%	81.08%	76.92%	79.25%	178

Nota. Elaboración propia.

Los mejores resultados de cada algoritmo se observan de manera gráfica en la Fig. 80 (mAP), Fig. 81 (precision), Fig. 82 (recall), Fig. 83 (tasa de pérdida - entrenamiento) y Fig. 84 (tasa de pérdida - validación). En los 4 casos, las mejores prestaciones se obtuvieron con el uso de transfer learning y sin la aplicación de data augmentation.

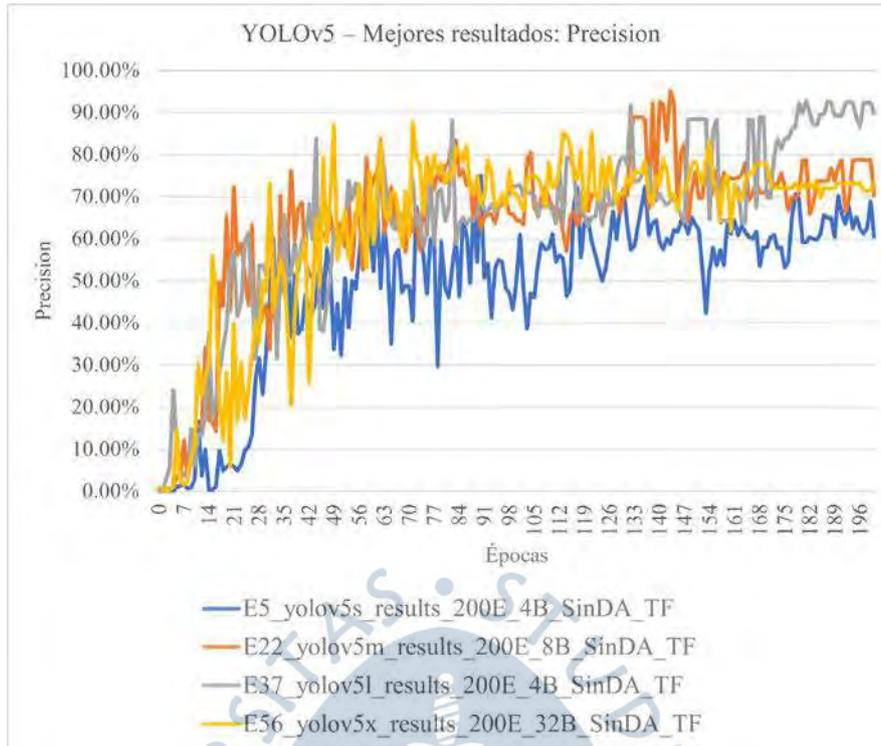
El detalle y resultados de los 64 experimentos puede consultarse en [wandb.ai/daniel-perez](https://wandb.ai/daniel-perez).

**Fig. 80.** YOLOv5 – Mejores resultados: Mean Average Precision (mAP).



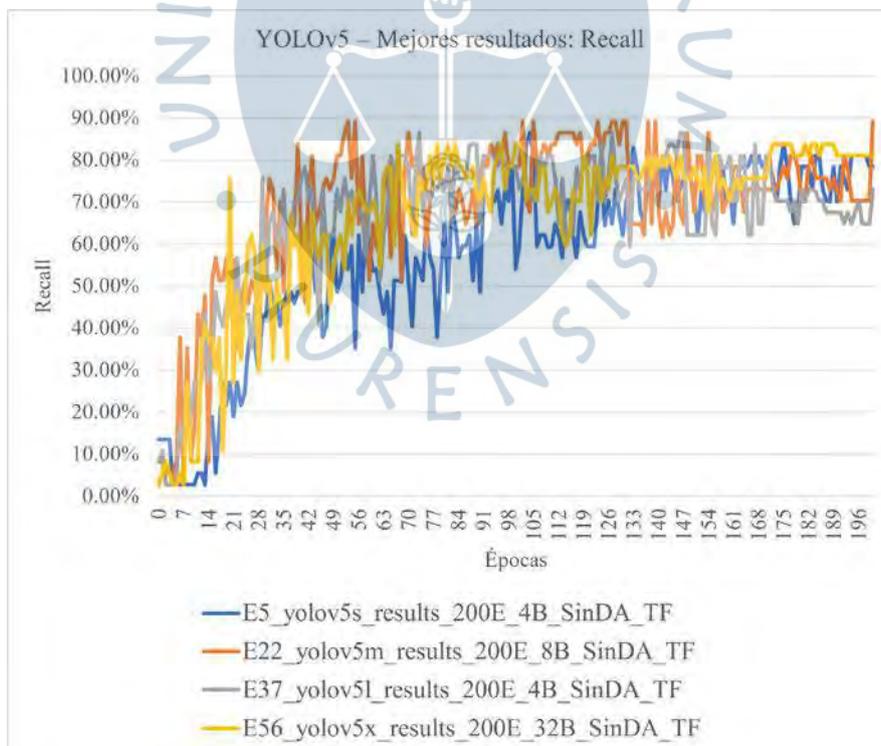
Nota. Elaboración propia.

**Fig. 81.** YOLOv5 – Mejores resultados: Precision.



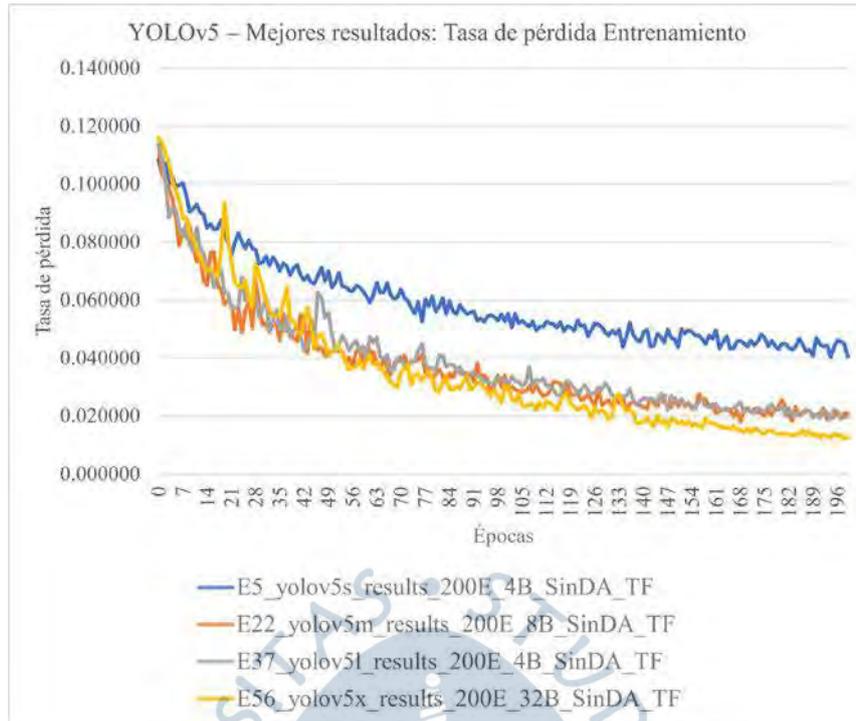
Nota. Elaboración propia.

**Fig. 82.** YOLOv5 – Mejores resultados: Recall.



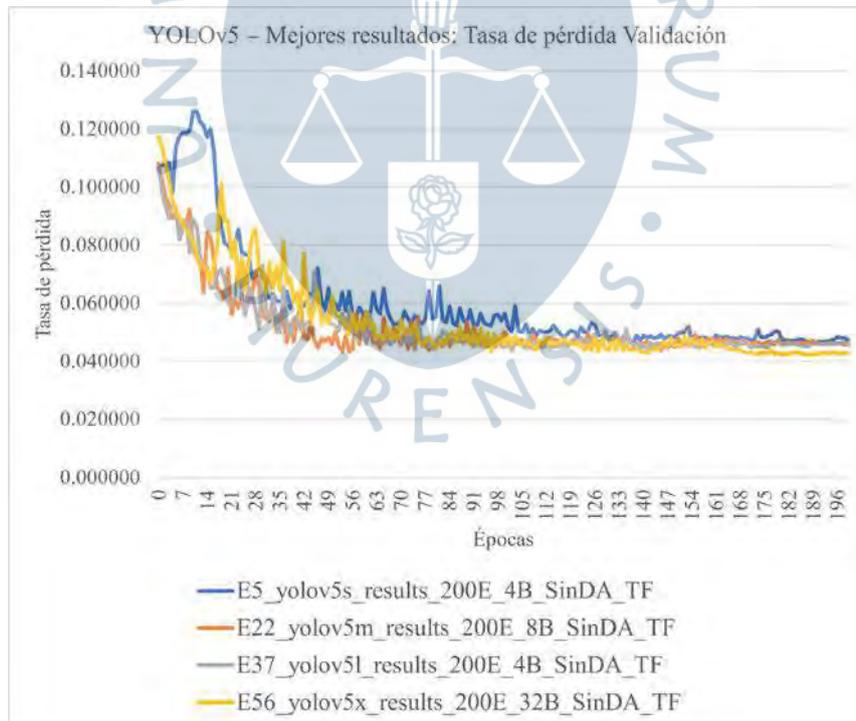
Nota. Elaboración propia.

**Fig. 83.** YOLOv5 – Mejores resultados: Tasa de pérdida Entrenamiento.



Nota. Elaboración propia.

**Fig. 84.** YOLOv5 – Mejores resultados: Tasa de pérdida Validación.

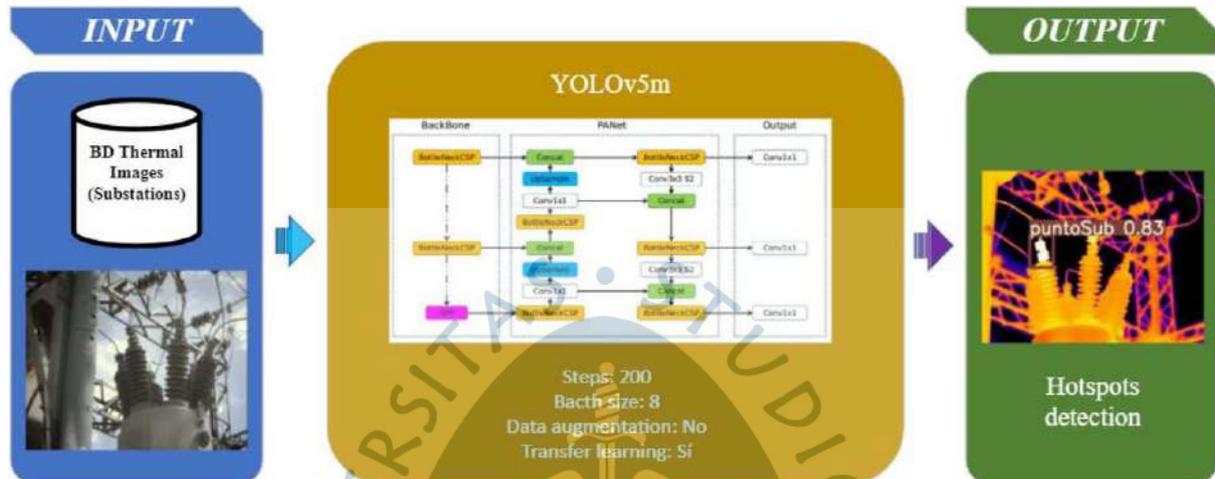


Nota. Elaboración propia.

Entonces, de los 64 experimentos realizados, el mejor modelo fue el entrenado con

200 épocas, un tamaño de batch de 8, sin data augmentation, y con el uso de transfer learning, correspondiente al experimento E22. Por lo que el modelo propuesto para esta investigación se representa en la Fig. 85, en el cual se ingresa la base de datos de imágenes térmicas, se aplica el algoritmo YOLOv5m y obtenemos como salida la detección de puntos calientes en las imágenes térmicas de subestaciones eléctricas.

**Fig. 85.** Modelo propuesto – detección de puntos calientes (YOLOv5m).



Nota. Elaboración propia.

Finalmente, algunas predicciones del modelo se muestran en la Fig. 86, Fig. 87, Fig. 88 y Fig. 89.

**Fig. 86.** YOLOv5m – Detección de puntos calientes (1).



Nota. Elaboración propia.

**Fig. 87.** YOLOv5m – Detección de puntos calientes (2).



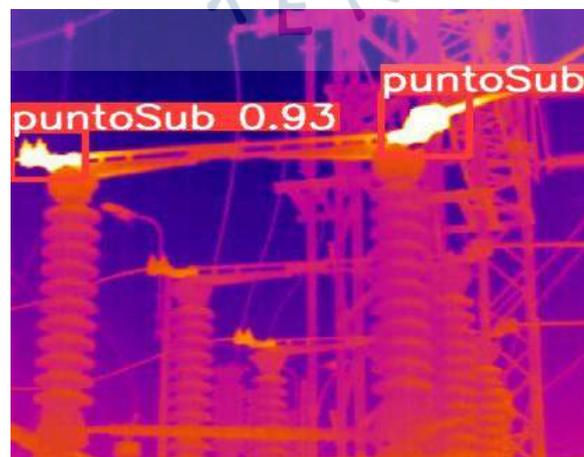
Nota. Elaboración propia.

**Fig. 88.** YOLOv5m – Detección de puntos calientes (3).



Nota. Elaboración propia.

**Fig. 89.** YOLOv5m – Detección de puntos calientes (4).



Nota. Elaboración propia.

## Apéndice 02: Artículo en revista WoS

<b>REVISTA</b>	INGENIUS Revista de Ciencia y Tecnología
<b>AÑO</b>	2021
<b>TÍTULO</b>	Transfer learning for binary classification of thermal images
<b>DOI</b>	<a href="https://doi.org/10.17163/ings.n26.2021.07">https://doi.org/10.17163/ings.n26.2021.07</a>

**Fig. 90.** Paper: *Transfer learning for binary classification of thermal images.*

### Transfer Learning for Binary Classification of Thermal Images

**Daniel Alexis Pérez-Aguilar**

Universidad de Piura

<https://orcid.org/0000-0003-4514-2873>

**Redy Henry Risco-Ramos**

Universidad de Piura

<https://orcid.org/0000-0002-3491-6020>

**Luis Casaverde-Pacherrez**

Universidad de Piura

<https://orcid.org/0000-0003-4037-1802>

### Keywords

fine-tuning, Friedman test, pre-training, thermal images, transfer learning

### Abstract

The classification of thermal images is a key aspect in the industrial sector, since it is usually the starting point for the detection of faults in electrical equipment. In some cases, this task is automated through the use of traditional artificial intelligence techniques, while in others, it is performed manually, which can lead to high rates of human error. This paper presents a comparative analysis between eleven transfer learning architectures (AlexNet, VGG16, VGG19, ResNet, DenseNet, MobileNet v2, GoogleNet, ResNeXt, Wide ResNet, MNASNet and ShuffleNet) through the use of fine-tuning, in order to perform a binary classification of thermal images in an electrical distribution network. For this, a database with 815 images is available, divided using the 60-20-20 hold-out technique and cross-validation with 5-Folds, to finally analyze their performance using Friedman test. After the experiments, satisfactory results were obtained with accuracies above 85 % in 10 of the previously trained architectures. However, the architecture that was not previously trained had low accuracy; with this, it is concluded that the application of transfer learning through the use of previously trained architectures is a proper mechanism in the classification of this type of images, and represents a reliable alternative to traditional artificial intelligence techniques.

Tomado de “*Transfer learning for binary classification of thermal images*” (Pérez-Aguilar et al., 2021).

**Apéndice 03: Artículo en revista Latindex**

<b>REVISTA</b>	CAMPUS – Facultad de ingeniería y arquitectura USMP
<b>AÑO</b>	2019
<b>TÍTULO</b>	Analysis of the implementation of LoRA and LoraWAN as a future scenario of industry 4.0 in Peruvian agro-industry sector
<b>DOI</b>	<a href="https://doi.org/10.24265/campus.2020.v25n29.10">https://doi.org/10.24265/campus.2020.v25n29.10</a>

**Fig. 91.** Paper: *Analysis of the implementation of LoRA and LoraWAN as a future scenario of industry 4.0 in Peruvian agro-industry sector.*

**Análisis de la implementación de LoRa y LoraWAN como escenario futuro de la Industria 4.0 en el sector agroindustrial peruano**

*Daniel Pérez, Redy Risco*

Resumen

El panorama actual del sector agroindustrial exige la inclusión de nuevas tecnologías que contribuyan a solucionar los constantes problemas a los que se enfrenta. Internet of Things se presenta como una alternativa para mitigarlos; sin embargo, las implementaciones de esta tecnología se realizan con el uso de redes de datos tradicionales como Wifi o redes celulares, las cuales representan un alto costo y consumo de energía. En esta investigación se realizó un análisis de la arquitectura y funcionamiento de la tecnología LoRa y el protocolo LoRaWAN, además de plantearse una serie de recomendaciones de cómo implementarla conjuntamente con otras tecnologías emergentes como es el BlockChain, todo esto, aplicado a la agroindustria peruana.

Palabras clave

Agroindustria, BlockChain, IoT, LPWAN, Perú

Tomado de *“Analysis of the implementation of LoRA and LoraWAN as a future scenario of industry 4.0 in Peruvian agro-industry sector”* (Pérez-Aguilar et al., 2019).

### Apéndice 04: Artículo en revista Scopus

<b>REVISTA</b>	DYNA – Facultad de Minas – Universidad Nacional de Colombia
<b>AÑO</b>	2022
<b>TÍTULO</b>	Use of a business intelligence framework in the management of the quality of the electricity supply in small and medium-sized companies
<b>DOI</b>	<a href="https://doi.org/10.15446/dyna.v89n221.99085">https://doi.org/10.15446/dyna.v89n221.99085</a>

**Fig. 92.** Paper: *Use of a business intelligence framework in the management of the quality of the electricity supply in small and medium-sized companies.*

Use of a business intelligence framework in the management of the quality of the electricity supply in small and medium-sized companies

Uso de un marco de inteligencia empresarial en la gestión de la calidad del suministro eléctrico en pequeñas y medianas empresas

DOI:  
<https://doi.org/10.15446/dyna.v89n221.99085>

Palabras clave:  
 analytics; business intelligence; clustering; CRISP-DM; data mining. (en)  
 analítica; inteligencia de negocios; agrupamiento; CRISP-DM; minería de datos. (es)

[PDF \(English\)](#)

Redy Risco-Ramos  
 Universidad de Piura, Piura, Perú  
<https://orcid.org/0000-0002-3491-6020>

Daniel Pérez-Aguilar  
 Universidad de Piura, Piura, Perú  
<https://orcid.org/0000-0003-4514-2873>

Luis Casaverde-Pacherrez  
 Universidad de Piura, Piura, Perú  
<https://orcid.org/0000-0003-4037-1802>

Edilberto Vásquez-Díaz  
 Universidad de Piura, Piura, Perú  
<https://orcid.org/0000-0003-0413-1625>

Tomado de “*Use of a business intelligence framework in the management of the quality of the electricity supply in small and medium-sized companies*” (Risco-Ramos et al., 2022).

### Apéndice 05: Artículo en revista SciELO

<b>REVISTA</b>	Información tecnológica
<b>AÑO</b>	2022
<b>TÍTULO</b>	Aplicación de técnicas de minería de datos para determinar un modelo de operación de un turbogenerador hidráulico
<b>DOI</b>	<a href="http://dx.doi.org/10.4067/S0718-07642022000600031">http://dx.doi.org/10.4067/S0718-07642022000600031</a>

**Fig. 93.** Paper: *Aplicación de técnicas de minería de datos para determinar un modelo de operación de un turbogenerador hidráulico.*

## Aplicación de técnicas de minería de datos para determinar un modelo de operación de un turbogenerador hidráulico

Redy H. Risco\*, Daniel A. Pérez, Luis A. Casaverde y Gerson La-Rosa

Universidad de Piura, Piura-Perú (correo-e: redy.risco.r@alum.udep.edu.pe; daniel.perez.a@alum.udep.edu.pe; luis.casaverde@alum.udep.edu.pe; gerson.larosa@udep.edu.pe)

\* Autor a quien debe ser dirigida la correspondencia

Recibido Mar. 10, 2022; Aceptado May. 17, 2022; Versión final Jul. 15, 2022, Publicado Dic. 2022

### Resumen

Este estudio aplica un modelo de referencia que implementa la minería de datos como parte de un entorno de inteligencia de negocios para analizar la operación de un turbogenerador hidráulico. Los datos analizados provienen del sistema de control y adquisición de datos de una central hidroeléctrica. Se genera un modelo utilizando análisis de componentes principales. Luego, se evalúa el modelo con nuevas observaciones utilizando los gráficos de control estadístico multivariante: cuadrado del error de predicción y  $T^2$  de Hotelling y finalmente se implementa el análisis conectando las tablas generadas con una herramienta convencional de inteligencia de negocios. El modelo se prueba en cinco muestras correspondientes a periodos previos a una parada del turbogenerador. Se concluye que el modelo de referencia utilizado puede ser aplicado al análisis de la operación de un turbogenerador hidráulico, que los gráficos de control son capaces de detectar comportamientos atípicos de operación y que es necesario el conocimiento del proceso a analizar para una adecuada interpretación de los resultados.

*Palabras clave:* minería de datos; control estadístico multivariante; análisis de componentes principales; gráficos de control estadístico

Tomado de “Aplicación de técnicas de minería de datos para determinar un modelo de operación de un turbogenerador hidráulico” (R. H. Risco et al., 2022).

### Apéndice 06: Artículo en congreso Scopus (1)

<b>CONGRESO</b>	IEEE International Conference on Automation/XXIV Congress of the Chilean Association of Automatic Control (ICA-ACCA)
<b>AÑO</b>	2021
<b>TÍTULO</b>	A simulation study of the performance of GPC and PID applied to a binary distillation column
<b>DOI</b>	<a href="https://doi.org/10.1109/ICAACCA51523.2021.9465198">https://doi.org/10.1109/ICAACCA51523.2021.9465198</a>

**Fig. 94.** Paper: A simulation study of the performance of GPC and PID applied to a binary distillation column.

**A simulation study of the performance of GPC and PID applied to a binary distillation column**

Publisher: IEEE [Cite This](#) [PDF](#)

Luis A. Casaverde; Redy H. Risco; Daniel A. Pérez [All Authors](#)

**Abstract**

**Abstract:**  
In this work, the comparison of a control system based on coupled PIDs and a GPC for MIMO systems applied to a Wood and Berry model distillation column has been carried out. The control law has been obtained based on the free and forced responses. Then, the design of independent TITO controls was carried out, using the Simulink/Matlab environment. Likewise, the evaluation of the interactions has been developed by using the relative gain matrix (RGA), to later perform the coupling of the loops with PIDs. Within the study, the behavior in the face of changes in the prediction horizon, changes in the control horizon, and the behavior in the face of changes in lambda and delta weights in the objective function have been analyzed, and additionally, the behavior in the face of disturbances was observed, this to make a comparison between the performance of a PID control (coupled system) and a GPC control. It has been found that the GPC control strategy presents a much better performance concerning the rise time, to the establishment time for the strategy with coupled PIDs. However, the PID still has the advantage of faster tuning due to the tuning tools.

**Document Sections**

- I. Introduction
- II. Methodology
- III. Comparison Between GPC and PID
- IV. Conclusions

**Authors**

Tomado de "A simulation study of the performance of GPC and PID applied to a binary distillation column" (Casaverde et al., 2021).

### Apéndice 07: Artículo en congreso Scopus (2)

<b>CONGRESO</b>	IEEE XXVII International Conference on Electronics, Electrical Engineering and Computing (INTERCON)
<b>AÑO</b>	2020
<b>TÍTULO</b>	Analysis of the implementation of Blockchain as a mechanism for digital and transparent food traceability in Peruvian social programs
<b>DOI</b>	<a href="https://doi.org/10.1109/INTERCON50315.2020.9220244">https://doi.org/10.1109/INTERCON50315.2020.9220244</a>

**Fig. 95.** Paper: *Analysis of the implementation of Blockchain as a mechanism for digital and transparent food traceability in Peruvian social programs.*

The screenshot shows the Scopus interface for the paper. At the top, the title is displayed in bold. Below the title, the publisher is listed as IEEE, with buttons for 'Cite This' and 'PDF'. The authors are listed as Daniel Pérez, Rody Risco, and Luis Casaverde, with a link to 'All Authors'. A box indicates '103 Full Text Views'. On the right, there are icons for a registered trademark, sharing, copyright, a document, and a notification bell. The abstract section is titled 'Abstract' and contains the following text: 'This paper studies the problems of the traditional food industry in Peruvian social programs, and how the Blockchain would serve as a solution that includes the immutable registration of information allowing the tracking of food by each responsible party in the process. It was observed that millions of people become ill or die from eating infected, spoiled, or intoxicated food and that these problems are directly linked to the registration and communication areas of the participants in the supply chain. It was concluded that combining technologies such as the Internet of Things (IoT) and Blockchain is an extremely attractive option for the transparent tracking of food from its production to its sale and consumption; likewise, using smart contracts ensures compliance with all the conditions necessary for its consumption. Finally, it should be mentioned that, by being able to track in seconds, the reduction of time and operating costs is imminent.' To the left of the abstract, there is a 'Document Sections' list with four items: I. Introduction, II. Analysis, III. Proposal, and IV. Conclusions.

Tomado de “*Analysis of the implementation of Blockchain as a mechanism for digital and transparent food traceability in Peruvian social programs*” (Perez et al., 2020).

### Apéndice 08: Artículo en congreso Scopus (3)

<b>CONGRESO</b>	IEEE XXVII International Conference on Electronics, Electrical Engineering and Computing (INTERCON)
<b>AÑO</b>	2020
<b>TÍTULO</b>	System identification models' fit using error histogram analysis and the Hampel filter as computational tools
<b>DOI</b>	<a href="https://doi.org/10.1109/INTERCON50315.2020.9220230">https://doi.org/10.1109/INTERCON50315.2020.9220230</a>

**Fig. 96.** Paper: *System identification models' fit using error histogram analysis and the Hampel filter as computational tools.*

**System identification models' fit using error histogram analysis and the Hampel filter as computational tools**

Publisher: IEEE [Cite This](#) [PDF](#)

Redy Risco, Daniel Pérez, Luis Casaverde [All Authors](#)

27 Full Text Views

**Abstract**

**Abstract:** In the present investigation, we use the error histogram analysis as a computational tool to define whether the model resulting from a system identification process should continue to be fitted, and the Hampel filter for the elimination of outliers as a tool that also avoids on model over-parameterization. To do this, we use three data sets from a four-cylinder BMW diesel engine, to identify a linear model, and then, with that model, analyze the error and its histogram in a data set (without noise, with noise and with outliers). The analysis of the histogram of the error was found to be a useful tool for detecting white noise and helps to avoid overfitting, in addition to the fact that the Hampel filter allowed detecting and eliminating atypical samples. The software used was MATLAB.

**Document Sections**

- I. Introduction
- II. Data set
- III. Analysis
- IV. Conclusions

Tomado de “*System identification models' fit using error histogram analysis and the Hampel filter as computational tools*” (R. Risco et al., 2020).