



# UNIVERSIDAD DE PIURA

**Facultad de Ingeniería y Arquitectura**

**Asignatura:**

**Control basado en datos y modelos (COD)**

**Ingenieros a cargo de la asignatura:**

**Dr. William Ipanaque Alama**

**Tema:**

**Análisis en fallas mecánicas internas en Motores AC utilizando  
imágenes termográficas**

**Integrantes (Grupo 10):**

- **Banda Moreno Carlos (Coordinador)**
- **Chunga Shimokawa Diego**
- **Serrepe Castillo Joan**
- **Imán Miranda Anthony**
- **Farias Guerrero Marcelo**

**Piura-Perú**

**2024**

## Resumen

En este trabajo se explora el uso de técnicas avanzadas para la identificación de fallas mecánicas en motores de corriente alterna mediante imágenes termográficas. La metodología se enfoca en el análisis térmico de componentes para identificar patrones de temperatura anormales que sugieren fallas. Para ello, se obtiene un conjunto de datos de diversas fuentes, tanto públicas como propia, y se aplica un proceso de preprocesamiento que incluye segmentación, extracción de características, aumentadores de contraste y filtrado para mejorar la calidad como dataset de las imágenes. Posteriormente, los datos se dividen en conjuntos de entrenamiento y validación, y se entrenan modelos de aprendizaje profundo para clasificar imágenes. La validación se realiza mediante el análisis de precisión y otros criterios de rendimiento.

Este enfoque no invasivo se apoya principalmente en redes neuronales convolucionales (CNN), como VGG16, ResNet50 y ViT, y otros algoritmos avanzados de aprendizaje automático como los modelos de Ensemble (ERT), máquinas de soporte vectorial (SVM), K-NN y ViT (Visual Transformers). Se revisa la literatura actual, abordando técnicas de procesamiento de imágenes como la ecualización de histograma (CLAHE), transformadas de wavelet (DWT) y otras técnicas de mejora de contraste. Además, se exploran métodos de clasificación avanzados como CNN, SVM, ERT, KNN, y ViT.

Los estudios previos destacan que el procesamiento termográfico, cuando se combina con modelos de inteligencia artificial, mejora significativamente la detección y diagnóstico de fallas en motores de corriente alterna, logrando una clasificación precisa incluso en entornos industriales y condiciones reales. En este contexto, el modelo que es una combinación de HSI, Wavelet y SVM ha demostrado ser particularmente efectivo, logrando resultados sobresalientes en términos de precisión y rapidez de clasificación.

## Lista de tablas

TABLA 1: PRESUPUESTO MÁXIMO	14
TABLA 2: MÉTODOS DE PREPROCESAMIENTO Y CLASIFICACIÓN POR DESARROLLAR	35
TABLA 3: CRITERIOS DE VALIDACIÓN PARA LOS MÉTODOS DE CLASIFICACIÓN	36
TABLA 4: DATASETS DISPONIBLES DE IMÁGENES TERMOGRÁFICAS	40
TABLA 5: ESTRUCTURA DE LA DATA FILTRADA	41
TABLA 6: TIPOS DE PREPROCESAMIENTO EVALUADOS	43
TABLA 7: MÉTODOS DE CLASIFICACIÓN Y SUS OBSERVACIONES	49
TABLA 8: CONJUNTO DE DATASETS DE ENTRENAMIENTO	62
TABLA 9: CONJUNTO DE DATASETS DE TESTEO	63
TABLA 10: MODELOS EVALUADOS	64
TABLA 11: RESULTADOS DE LOS MODELOS ENSAYADOS	65
TABLA 12: TIPOS DE PREPROCESAMIENTO PARA EXTRACCIÓN DE CARACTERÍSTICAS	70
TABLA 13: CONFIGURACIÓN DEL MOTOR AC EN ANÁLISIS	76
TABLA 14: ESTRUCTURA DATA4	82
TABLA 15: TIPOS DE ACM EVALUADOS DE LA DATA4	83



## Lista de imágenes

IMAGEN 1: ESTRUCTURA DE TRANSFORM (FUENTE: ALL YOU NEED ITS ATTENTION)	23
IMAGEN 2: FUNCIONAMIENTO INTERNO ATTENTION (IZQUIERDA) - ENTRADAS Y SALIDAS DE ATTENTION (DERECHO)(FUENTE: ALL YOU NEED ITS ATTENTION)	23
IMAGEN 3: ESTRUCTURA BÁSICA DE UNA ViT (FUENTE: GOOGLE RESEARCH TEAM)	25
IMAGEN 4: ESTRUCTURA BÁSICA DE UN ENCODER DE UNA ViT (FUENTE: GOOGLE RESEARCH, BRAIN TEAM)	27
IMAGEN 5: ESTRUCTURA DE METODOLOGÍA	34
IMAGEN 6: MÓDULO DE PRUEBA DEL LABORATORIO DE AUTOMATIZACIÓN (1)	37
IMAGEN 7: ESTACIÓN DE BOMBEO DE LA FACULTAD DE COMUNICACIÓN (2)	37
IMAGEN 8: ESTACIÓN DE BOMBEO DEL EDIFICIO E (3)	38
IMAGEN 9: ESTACIÓN DE BOMBEO DE AGUAS RESIDUALES DEL EDIFICIO 80 (4)	38
IMAGEN 10: MODULO DE MOTOR DEL LABORATORIO DE ELECTROTECNIA (5)	38
IMAGEN 11: FLUKE TiS65 (FUENTE: FLUKE CORPORATION)	39
IMAGEN 12: MAPA DE LAS INSTALACIONES EVALUADAS}	39
IMAGEN 13: IMAGEN OBTENIDA DE LA FLUKE TiS65-MOTOR LABSAC	39
IMAGEN 14: DISTRIBUCIÓN DE TEMPERATURA PARA LA DATA1 (FUENTE: EKSPLOATACJA I NIEZAWODNOŚĆ JOURNAL)	41
IMAGEN 15: MÉTODO DE SEGMENTACIÓN POR YOLOV5	44
IMAGEN 16: MÉTODO DE SEGMENTACIÓN OTSU	44
IMAGEN 17: MÉTODO DE SEGMENTACIÓN OTSU	45
IMAGEN 18: MÉTODO DE DIFERENCIACIÓN FFT	45
IMAGEN 19: MÉTODO DE DIFERENCIACIÓN CLAHE	46
IMAGEN 20: MÉTODO DE EXTRACCIÓN DE CARACTERÍSTICAS DWT	46
IMAGEN 21: MÉTODO DE EXTRACCIÓN DE CARACTERÍSTICAS DWT	46
IMAGEN 22: MÉTODO DE EXTRACCIÓN DE CARACTERÍSTICAS FOS	47
IMAGEN 23: MÉTODO DE EXTRACCIÓN DE CARACTERÍSTICAS GLCM	47
IMAGEN 24: GENERACIÓN DE DATOS POR DATA AUGMENTATION	48
IMAGEN 25: ESTRUCTURA DEL MÉTODO DE CLASIFICACIÓN VGG16-1	50
IMAGEN 26: ESTRUCTURA DEL MÉTODO DE CLASIFICACIÓN VGG16-2	50
IMAGEN 27: ESTRUCTURA DEL MÉTODO DE CLASIFICACIÓN RESNET50	51
IMAGEN 28: ESTRUCTURA DEL MÉTODO DE CLASIFICACIÓN INCEPTIONV3-1	51
IMAGEN 29: ESTRUCTURA DEL MÉTODO DE CLASIFICACIÓN INCEPTIONV3-2	51
IMAGEN 30: ESTRUCTURA DEL MÉTODO DE CLASIFICACIÓN EFFICIENTNETB2,B5,B7	52
IMAGEN 31: ESTRUCTURA DEL MÉTODO DE CLASIFICACIÓN ERT	52
IMAGEN 32: ESTRUCTURA DEL MÉTODO DE CLASIFICACIÓN SVM-1	53
IMAGEN 33: ESTRUCTURA DEL MÉTODO DE CLASIFICACIÓN SVM-2	53
IMAGEN 34: ESTRUCTURA DEL MÉTODO DE CLASIFICACIÓN KNN	54
IMAGEN 35: ESTRUCTURA DEL SPT (INHA UNIVERSITY)	54
IMAGEN 36: ESTRUCTURA DEL LSA (INHA UNIVERSITY)	55
IMAGEN 37: ESTRUCTURA DE LA CCT (UNIVERSITY OF OREGON)	55
IMAGEN 38: ESTRUCTURA DE UNA CROSSViT (MIT-IBM WATSON AI LAB)	56
IMAGEN 39: REPRESENTACIÓN DE LOS PARCHES EN 2 IMÁGENES	57
IMAGEN 40: ESTRUCTURA DE LA ViT APLICADA	57
IMAGEN 41: ESTRUCTURA DE LA ViT PARA DATASETS PEQUEÑOS	58
IMAGEN 42: ESTRUCTURA DE LA CCT APLICADA	59
IMAGEN 43: ESTRUCTURA DE LA CROSS ViT APLICADA	60
IMAGEN 44: MATRIZ DE CONFUSIÓN DEL MODELO 12	66
IMAGEN 45: DETECCIÓN FALLIDA YOLOV5s	67
IMAGEN 46: DETECCIÓN CORRECTA CON YOLOV5M	68
IMAGEN 47: CURVAS DE APRENDIZAJE Y MATRIZ DE CONFUSIÓN SIMPLEViT	74
IMAGEN 48: : CURVAS DE APRENDIZAJE Y MATRIZ DE CONFUSIÓN ViTFSD	74
IMAGEN 49: CURVAS DE APRENDIZAJE Y MATRIZ DE CONFUSIÓN CCT	75

IMAGEN 50: CURVAS DE APRENDIZAJE Y MATRIZ DE CONFUSIÓN CrossViT	75
IMAGEN 51: IMAGEN COMPARATIVA DE ACM PARA DISTINTAS FALLAS	77
IMAGEN 52: MOTOR ACM-LAB SAC	78
IMAGEN 53: COMPARATIVA USO Y NO USO DEL FONDO NEGRO, VISTA LATERAL DERECHA DEL MOTOR	79
IMAGEN 54: VISTA LATERAL IZQUIERDA DEL MOTOR	79
IMAGEN 55: VISTA OBLICUA DEL MOTOR	79
IMAGEN 56: IMÁGENES UBICACIÓN 2	80
IMAGEN 57: IMÁGENES UBICACIÓN 3	80
IMAGEN 58: IMÁGENES UBICACIÓN 4	81
IMAGEN 59: IMÁGENES MOTOR UBICACIÓN 5	81
IMAGEN 60: IMÁGENES UBICACIÓN 5-BOMBA ELÉCTRICA	82
IMAGEN 61: SEGMENTACIÓN OTSU-MODELO FINAL	84
IMAGEN 62: NORMALIZACIÓN Y REDIMENSIONAMIENTO-MODELO FINAL	84
IMAGEN 63: ECUALIZACIÓN-MODELO FINAL	85
IMAGEN 64: DWT- MODELO FINAL	87
IMAGEN 65: GUARDADO DE IMÁGENES Y METADATA	87
IMAGEN 66: CONFIGURACIÓN <i>Xtrain</i> y <i>Xtest</i>	88
IMAGEN 67: RESULTADOS TRAINING- MODELO 18	89
IMAGEN 68: RESULTADOS TEST-MODELO 18	89
IMAGEN 69: MATRIZ DE CONFUSIÓN MODELO 18-ETAPA 1	89
IMAGEN 70: MATRIZ DE CONFUSIÓN MODELO 18-ETAPA 2-MSL	90
IMAGEN 71: MATRIZ DE CONFUSIÓN MODELO 18-ETAPA 2-BRB	91
IMAGEN 72: PRUEBA DE VALIDACIÓN CON DATA4 AL MODELO 18	91
IMAGEN 73: DISEÑO DE LA INTERFAZ DENTRO DE QTDESIGNER	92
IMAGEN 74: BOTÓN PARA SELECCIONAR LA IMAGEN	92
IMAGEN 75: INTERFAZ UNA VEZ SELECCIONADA LA IMAGEN	93
IMAGEN 76: BOTÓN PARA ELEGIR LA IMAGEN DE MANERA MANUAL	93
IMAGEN 77: MOTOR IDENTIFICADO DE MANERA CORRECTA	93
IMAGEN 78: RESULTADOS OBTENIDOS	94
IMAGEN 79: BOTÓN CONFIGURACIONES PARA ESTABLECER LOS LÍMITES INFERIOR Y SUPERIOR DE TEMPERATURA	94
IMAGEN 80: REPORTE FINAL DE LA INTERFAZ	95



## Tabla de contenido

1. Introducción.....	8
2. Problema y justificación.....	9
3. Objetivos y alcance .....	11
3.1. Pregunta de investigación .....	11
3.1.1. Hipótesis:.....	11
3.1.2. Preguntas adicionales: .....	11
3.2. Objetivos .....	12
3.2.1. Objetivo Principal .....	12
3.2.2. Objetivo específicos .....	12
3.3. Restricciones principales del proyecto .....	13
3.3.1. Alcance.....	13
3.3.2. Costo .....	14
3.3.3. Tiempo.....	15
3.4. Otras restricciones .....	15
3.4.1. Disponibilidad de datos en el contexto real .....	15
3.4.2. Recursos computacionales .....	15
3.4.3. Modelo empírico vs real.....	15
4. Marco teórico .....	16
4.1. Redes neuronales (NN) .....	17
4.2. Motores de corriente alterna (AC) .....	17
4.3. Fallas.....	17
4.3.1. Fallas mecánicas (MF).....	17
4.3.2. Fallas Eléctricas (EF).....	17
4.4. Radiación.....	18
4.5. Termografía por infrarrojos (TI) .....	18
4.6. Preprocesamiento de imágenes (IPP).....	18
4.6.1. Filtros .....	18
4.6.2. Contrast Limited Adaptive Histogram Equalization (CLAHE) .....	19
4.6.3. Transformada de Wavelet Discreta (DWT).....	19
4.6.4. Análisis de Componentes Independientes (ICA) .....	19
4.6.5. Segmentación de imágenes (IS) .....	19
4.6.6. Bag of Visual Words (BoVW) .....	20
4.6.7. Speeded-UP- Robust Features (SURF).....	20
4.6.8. Hue Saturation Intensity (HSI) .....	20

4.6.9.	<i>Linear Discriminant Analysis (LDA)</i> .....	20
4.6.10.	<i>Principal Component Análisis (PCA)</i> .....	20
4.7.	<b>Clasificadores en base a características (IFE)</b> .....	20
4.7.1.	<i>Redes neuronales convolucionales (CNN)</i> .....	21
4.7.2.	<i>K-Nearest Neighbors (K-NN)</i> .....	21
4.7.3.	<i>Support Vector Machine (SVM)</i> .....	21
4.7.4.	<i>Random Tree (RT)</i> .....	21
4.7.5.	<i>Visual Transform (ViT)</i> .....	22
5.	<b>Estado del arte</b> .....	28
6.	<b>Metodología, materiales y procedimiento</b> .....	34
7.	<b>Desarrollo, experimentación y resultados</b> .....	40
7.1.	<b>Datasets revisados</b> .....	40
7.2.	<b>Preprocesamientos evaluados</b> .....	42
	YOLOV5 .....	43
	Otsu.....	44
	K-Means .....	44
	FFT .....	45
	CLAHE .....	45
	DWT .....	46
	SIFT-BoVW .....	46
	FOS.....	47
	GLCM .....	47
	LDA .....	47
	PCA .....	47
	Data Augmentation.....	47
7.3.	<b>Métodos de clasificación</b> .....	48
	VGG16 .....	49
	ResNet50 .....	50
	InceptionV3.....	51
	EfficientNetB2,B5 y B7 .....	52
	ERT .....	52
	SVM.....	53
	KNN.....	53
	ViT .....	54
	ViTFSD o ViT For Smaller Data .....	58
	Compact Convolutional Transformers (CCT).....	59

CrossViT.....	60
7.4. Datasets de entrenamiento y testeo .....	61
7.5. Modelos ensayados .....	64
7.6. Resultados y criterios de evaluación .....	65
8. Análisis de resultados.....	67
8.1. Análisis preprocesamientos.....	67
8.2. Análisis clasificadores.....	72
8.3. Análisis datasets.....	76
8.3.1. Laboratorio de Sistemas Automáticos de control (Lab SAC)- Ubicación 1 .....	78
8.3.2. Estación de bombeo de agua de la Facultad de Comunicación-Ubicación 2 .....	80
8.3.3. Estación de bombeo de agua para el Aulario UDEP/ Edificio E-Ubicación 3 ...	80
8.3.4. Estación de bombeo de aguas residuales del edificio 80-Ubicación 4 .....	81
8.3.5. Laboratorio de Electrotecnia-Ubicación 5 .....	81
8.4. Mejor modelo (Modelo 18).....	83
8.4.1. Resultados entrenamiento.....	89
8.4.2. Resultados prueba con data de testeo .....	89
8.4.3. Prueba de validación .....	91
8.5. Interfaz.....	92
9. Conclusiones .....	96
10. Bibliografía.....	98
11. Cronograma .....	100



## 1. Introducción

Los motores de corriente alterna (ACM) son elementos esenciales dentro de una amplia variedad de aplicaciones y cadenas de producción industriales. Sin embargo, los ACM no están exentos de sufrir fallas o problemas de funcionamiento que puedan afectar su rendimiento y en casos más graves, inclusive generar interrupciones en las operaciones en las que esté involucrado, lo cual evidentemente no es conveniente, pues en el primer caso supone en muchos casos pérdidas económicas de mediano/gran impacto de manera continua hasta que se realice el mantenimiento correctivo correspondiente, mientras que el segundo caso, ocasionaría una parada imprevista en la cadena de producción, lo cual podría desencadenar un mayor daño aparte del económico, como accidentes de mayor envergadura que comprometan la salud de una o más personas.

Las fallas mecánicas internas en los ACM pueden manifestarse de diversas maneras, como: sobrecalentamiento, vibraciones anormales, ruidos anormales y/o pérdidas de potencia. Cada una de estas fallas, si no se detectan y corrigen a tiempo, pueden derivar en daños más graves y costosos, por lo que la capacidad de identificar en etapas tempranas estos problemas es crucial para no solo asegurar la continuidad de operación de la maquina en cuestión, sino también garantizar la máxima eficiencia operativa del proceso que se esté llevando a cabo.

Una de las técnicas emergentes para la detección de fallas mecánicas en ACM es el uso de imágenes termográficas. Estas imágenes muestran la distribución de temperaturas en la superficie de los componentes del motor. Al analizar estas imágenes, es posible identificar patrones de calor anormales que pueden indicar la presencia de fallas internas. Existen muchos tipos de fallas internas en un ACM, una de las clasificaciones más usadas son las que las diferencian por el origen de la falla, tales como: Fallas térmicas, electromagnéticas, mecánicas, etc. Siendo la falla más común, las del tipo mecánico, ejemplos de este tipo de fallas tenemos: desequilibrios entre piezas, desalineaciones de los ejes, fallas en los rodamientos, en las barras del rotor o estator, etc.

El presente proyecto, titulado *"Análisis en fallas mecánicas internas en Motores AC utilizando imágenes termográficas"*, tiene como objetivo principal explorar la aplicación de la termografía infrarroja para la detección de fallas en motores de corriente alterna. Dentro del trabajo, se buscará identificar las fallas más comunes, analizar las imágenes termográficas obtenidas y evaluar la eficacia de esta técnica como herramienta de diagnóstico en el entorno industrial.

## 2. Problema y justificación

En la época contemporánea, los ACM representan un rol identitario en un abanico amplio de equipos y sistemas de diversas índole. Si bien este tipo de máquinas son conocidos por su robustez y eficiencia, presentan en algunas ocasiones un conjunto de fallas, siendo dentro de las mismas, las fallas mecánicas las que más destacan por su frecuencia de aparición. Dichas fallas pueden comprometer su rendimiento o su funcionamiento. Las fallas mecánicas ya sean internas, tales como el desgaste de los rodamientos, fallas en las barras del rotor o estator; y/o externas como la desalineación del eje con la carga, pueden provocar una disminución significativa en la eficiencia del motor, así como un aumento en el consumo de energía y el riesgo de daños si no se detectan a tiempo.

Una de las dificultades en el mantenimiento de los ACM es la detección temprana de estos problemas/defectos mecánicos, ya que muchas veces las señales no son tan evidentes hasta que el fallo es cercano al nivel crítico y/o su impacto es evidente de una u otra forma, y dado que los métodos tradicionales de diagnóstico, como el monitoreo de vibraciones o análisis de corriente, pueden no ser lo suficientemente sensibles o precisos para identificar algunos tipos de falla en una o todas sus etapas. Entonces surge la necesidad de desarrollar técnicas de diagnóstico que sean más prácticas, económicas y principalmente que puedan detectar estos problemas antes de que se conviertan en fallas graves.

Es en este contexto que el uso de un estudio termográfico por imágenes se presenta como una herramienta interesante por desarrollar. Al analizar las imágenes térmicas, es posible identificar patrones anormales de temperatura que pueden ser indicativos de varios tipos de falla en los ACM, por lo que no solo es un método más práctico, sino que también es escalable a todos los tipos de falla conocidos.

Sin embargo, a pesar de su potencial, el uso de imágenes termográficas para la detección de fallas en ACM aún no ha sido adoptado en la industria. Esto se debe en parte a la falta de estudios exhaustivos que demuestren su eficacia en comparación con otras técnicas de diagnóstico, y debido a que la precisión de dicho método está íntimamente vinculado a la existencia de un *Dataset* de entrenamiento ya existente, lo cual requiere a su vez que la/las fallas, lo cual representa a su vez otro problema y es que el número de fallas posibles son muy diversas, y para cada una de las mismas deberá existir evidencia fotográfica.

Tal y como se señaló antes, el uso de las imágenes termográficas permite tener una visión detallada de las temperaturas en el interior del motor, aunque es importante matizar que, dependiendo del material de la carcasa del motor y las temperaturas de operación, y ofrece una manera más o menos eficiente de monitorear el estado de los ACM en operación, esto es debido a que los gradientes de temperaturas y la ubicación de los mismos, son el factor principal de distinción entre fallas, es por ello que dependiendo de la captura de estas características, la eficacia del presente método aumenta o disminuye; Sin embargo, es un hecho categórico que a pesar de las muchas y diversas dificultades señaladas previamente, este enfoque permite identificar y abordar los problemas correspondientes a una falla de un ACM mejor que otros métodos.

A su vez es importante también mencionar que en un mundo donde la sostenibilidad es cada vez más relevante, la capacidad de mantener los ACM en condiciones óptimas tiene un impacto positivo en el consumo de energía, ya que los motores que operan con fallas son menos eficientes y consumen más energía que uno en condiciones óptimas, lo que se traduce en un mayor impacto ambiental y en costos operativos elevados para las empresas, por lo tanto, la presente investigación sobre la aplicación de la termografía infrarroja en la detección de fallas mecánicas en ACM no solo es relevante desde un punto de vista técnico, sino también desde una perspectiva económica y ambiental. De esta forma, es válido afirmar que este proyecto busca de una manera u otra contribuir a complementar las investigaciones previas y proporcionar una base sólida para la adopción de esta tecnología en la industria.

Asimismo, el presente trabajo tiene como objetivo. En primer lugar, al demostrar la efectividad de la termografía infrarroja como herramienta de diagnóstico, promover su adopción de manera generalizada en la industria, mejorando las prácticas de mantenimiento predictivo, que, a su vez, ayudará a reducir los tiempos de inactividad no planificados y los costos de mantenimiento, lo que beneficiaría no solamente a las empresas, reduciendo gastos innecesarios o no previstos, sino también, a los consumidores.

En segundo lugar, este proyecto trata de contribuir al desarrollo de metodologías más precisas y eficientes para la detección de fallas mecánicas en ACM. Al revisar datos e investigaciones, y presentar análisis detallados de metodologías previas y nuevas, decantando todo ello en un modelo sugerido y las observaciones precisas y necesarias respecto al mismo.

### 3. Objetivos y alcance

#### 3.1. Pregunta de investigación

¿ El uso de imágenes termográficas, también conocido como análisis termográfico combinada con métodos de clasificación por medio de redes neuronales convolucionales (CNN) puede ser considerada como una alternativa en el campo de la detección de fallas mecánicas internas y/o externas en motores AC?

##### 3.1.1. Hipótesis:

- Hipótesis Nula(H0): El uso combinado del análisis termográfico en conjunto con un método clasificador, en este caso redes neuronales convolucionales (CNN) no sirve para elaborar un modelo clasificador completo, cuyo rendimiento es lo suficientemente elevado con respecto a otros métodos convencionales, como para ser considerada su implementación en la industria.
- Hipótesis Alternativa (H1): El uso combinado del análisis termográfico en conjunto con un método clasificador, en este caso redes neuronales convolucionales (CNN) sirve para elaborar un modelo clasificador completo, cuyo rendimiento es lo suficientemente elevado con respecto a otros métodos convencionales, como para ser considerada su implementación en la industria.

##### 3.1.2. Preguntas adicionales:

- ¿ Cuales de las fallas mecánicas en general en el objeto de estudio en cuestion (ACM) pueden ser detectadas de manera efectiva utilizando el metodo propuesto?
- ¿Cuál es la precisión de cada uno de los diferentes metodos clasificadores probados para el paso de la clasificacion o identificacion de fallas mecánicas propuestas en el objeto de estudio (ACM)?
- ¿ Qué técnicas de preprocesamiento han mostrado tener un impacto positivo en el funcionamiento general del modelo, es decir: mejorar la calidad de los datasets revisados y/o acrecentar la diferenciacion entre clases, por ende, la precisión general del modelo?
- ¿ Afecta el tamaño del conjunto de datos, y su respectiva clasificacion por tipos de clase, en este caso, fallas a la capacidad de generalización del modelo clasificador, es decir a su escalabilidad frente a otro tipo de fallas?
- ¿ Es factible la mejora de la detección de fallas combinando el metodo propuesto junto con otras técnicas de diagnóstico, como el análisis de vibraciones o no se ha dado una mejora sustancial?
- ¿ Qué impacto tiene la calidad de las imágenes termográficas, es decir la resolucion por pixeles, en la efectividad del modelo clasificador?
- ¿ Cómo se comparan los resultados del modelo propuesto obtenidas en ambientes y/o condiciones controladas versus condiciones reales, es decir, un equipo y/o contexto del cual no se cuenta con un dataset?
- ¿ Cuáles son los beneficios económicos y operativos de implementar un sistema como el propuesto en una planta industrial?



## 3.2. Objetivos

### 3.2.1. Objetivo Principal

El objetivo principal de este proyecto es explorar y evaluar la efectividad de usar el metodo obejto de estudio (Analisis termografico + CNN) como técnica para la detección de fallas mecánicas en motores AC (ACM).

### 3.2.2. Objetivo específicos

- **Identificar los tipos de fallas mecánicas en los ACM que serán objeto de estudio:** Se realizará una búsqueda y revisión sistemática de estudios previos como papers, tesis, etc. relacionados o lo suficientemente conectado al tema de estudio para identificar las fallas más comunes en motores de corriente alterna, pero sobre todo tipos de fallas que ya hayan sido evaluadas parcialmente o por completo por alguna parte del método propuesto.
- **Recopilación de data:** Posterior a la identificación de las fallas que se evaluarán, se debe reunir un conjunto de datos de imágenes termográficas relevantes, que incluyan diferentes la mayor de fallas mecánicas en motores AC seleccionadas antes. Y en caso necesario, la creación de nuevas imágenes a través de experimentos controlados con ayuda de la Universidad de Piura.
- **Preprocesamiento de datos:** Aplicar técnicas de preprocesamiento a las imágenes recopiladas para mejorar la calidad de los datos, eliminar el ruido y/o acrecentar diferencias entere clases. Esto incluirá la normalización de las imágenes, la aplicación de filtros y la segmentación de regiones de interés. Todo aquello con el mero objetivo de garantizar que los datos utilizados para entrenar la red neuronal sean precisos, distintivos entre sí y representativos de cada falla.
- **Clasificación de fallas:** Probar diversos métodos de clasificación, para ello primero se probarán los métodos de la bibliografía consultada y posteriormente y en caso se vea necesario se crearán modelos propios. Para este paso también será necesario dividir la data por clases en caso no lo este, y más importante haberle aplicado los preprocesamientos detallado en el objetivo previo.
- **Entrenamiento del método clasificador:** Desarrollar y entrenar los métodos que hayan dado un desempeño adecuado. Por supuesto, se probarán diferentes arquitecturas de redes neuronales y se ajustarán los parámetros que se consideren necesarios para maximizar el rendimiento, la precisión en la detección de fallas y su robustez frente a varios tipos de *Datasets* y entornos.



- **Evaluación del método clasificador escogido:** Se llevarán a cabo pruebas para corroborar el desempeño del método clasificador de fallas mecánicas. Por suerte, los softwares, entornos de desarrollo y ejecución, así como las librerías a usar ya brindan métricas como *Accuracy*, *F1-score* y *Recall*, y un símil simplificado de una matriz de confusión para medir la efectividad del método. Aparte, también se llevará a cabo una comparativa con más de un dataset, esto para corroborar que el modelo en cuestión no se ha sobre ajustado a únicamente un tipo de entorno.
- **Escalabilidad del modelo propuesto:** También se explorarán técnicas avanzadas, como el ajuste fino de hiperparámetros y el uso de modelos ensamblados, los cuales combinan más de una sola arquitectura, para mejorar la capacidad del modelo de detectar fallas en otros entornos, pues en un modelo ensamblado pueden haber más de 2 inputs, por ej: Imágenes y señales de vibraciones y/o mediciones de frecuencias.
- **Contribución a los ODS:** Este proyecto también busca contribuir a los Objetivos de Desarrollo Sostenible (ODS), específicamente al **ODS 9: Industria, Innovación e Infraestructura** y al **ODS 12: Producción y Consumo Responsables**. A través del desarrollo de tecnologías avanzadas de diagnóstico predictivo, de esta forma no solo se promueve la innovación en la industria, sino que también se contribuye a un uso más responsable de los recursos.

### 3.3. Restricciones principales del proyecto

#### 3.3.1. Alcance.

Este proyecto abarca todo el ciclo de creación de un sistema de detección de fallas basado en redes neuronales y análisis termográfico, desde la recopilación de datos hasta la implementación y validación del modelo. A continuación, se detallan los aspectos clave del alcance:

1. **Tipo de motor y tipo de falla:** Dado que el proyecto se centra en el uso de imágenes termográficas de diversas fuentes, incluyendo bases de datos públicas tanto como privadas. Dicho trabajo se centrará en motores de corriente alterna (ACM), pero solo del tipo jaula de ardilla, utilizados en aplicaciones industriales y con respecto a las fallas, únicamente evaluará las fallas del tipo mecánico.
2. **Tipos de preprocesamiento de imágenes:** Se implementarán técnicas de procesamiento específicamente diseñados para imágenes termográficas, además es importante señalar que dicho preprocesamiento no debe incurrir con dificultades con el paso de extracción de características, es decir no debe comprender la estructura original de la imagen.

3. **Métodos de clasificación únicamente con redes neuronales:** Para el propósito de este trabajo se diseñará al método de clasificación tomando en cuenta únicamente los modelos de identificación que usen redes neuronales. Aun así, dentro de los modelos explorados, se probarán diferentes arquitecturas, desde redes convolucionales hasta modelos más simples, según los resultados que ambos tipos de método nos proporcionen al momento de evaluarlos.
4. **Rendimiento y eficiencia mínima por modelo:** El entrenamiento del método propuesto se llevará a cabo según las métricas antes expuestas, y se tomara en cuenta únicamente modelos cuyo *accuracy* sea mayor al 85%, esto se hará para reducir el número de modelos a probar y de esta forma optimizar los modelos que mejores resultados hayan dado.
5. **Conjuntos de datos:** Solo se puede trabajar modelos de redes neuronales en conjunto con la existencia de un dataset de calidad decente, para ello tal y como se ha explicado antes, se trabajará únicamente con los Datasets que cumplan con las siguientes características: Múltiples tipos de falla, Cantidad de imágenes superior a 100 por clase, Resolución mínima de imagen: 640x512.

### 3.3.2. Costo

Al tratarse de un trabajo de investigación semestral de carácter meramente académico, se buscará que el costo del trabajo sea el menor posible, aunque sin escatimar en las herramientas que sean vitales para el desarrollo del mismo, como presupuesto máximo se ha considerado un aproximado de S/.600, a continuación, se presenta una tabla con los gastos pensados:

Ítem	Costo	Unidad	Total
Licencia ColabPro	S/. 40	3	S/. 120
Norma ASTM E1934-99a	S/. 250	1	S/. 250
Viáticos	S/. 50	5	S/. 250

*Tabla 1: Presupuesto máximo*

Es importante sin embargo resaltar que la compra de la Norma ASTM es un gasto que se prevé dispensar, pues actualmente estamos tratando de obtener una copia pasada de dicha norma por medio del Laboratorio de Electrotecnia de la Universidad de Piura.

Con respecto a los gastos por viáticos estos se refieren a un fondo mutuo, donde en caso de que se presente algún tipo de problema de leve o mediana gravedad involucrada con alguno de los integrantes del grupo, dicho fondo sirva para mantener la actividad de dicho miembro y que de esta manera no se vea afectado su desempeño en el trabajo.

### **3.3.3. Tiempo**

Tomando en cuenta como fecha de inicio del presente proyecto el día 09 de agosto del 2024, siendo pues, este día en el que se nos asignó el tema del presente trabajo, se considera que este proyecto cuenta con 13 semanas y 4 días para ser culminado, por lo que el grado de temas abarcado se ceñirá únicamente a los objetivos antes relatados.

## **3.4. Otras restricciones**

### **3.4.1. Disponibilidad de datos en el contexto real**

Para lograr un buen desempeño de esta herramientas, se requiere que sea entrenada con un conjunto de información, la cual a su vez debe sacarse del contexto en el cual se aplicará dicha herramienta, o en caso esto último no se pueda, con al menos un contexto lo más cercano al real, esto significa que el modelo propuesto en este trabajo conseguirá mejor resultados para un contexto específico si se entrena con su propio *dataset*.

### **3.4.2. Recursos computacionales**

El entrenamiento de redes neuronales requiere una gran capacidad computacional, y en muchos casos, inclusive hardware especializado (como GPUs) para obtener resultados en un tiempo razonable. Dicha limitación nos restringe la capacidad de explorar modelos más complejos a los que se verán.

### **3.4.3. Modelo empírico vs real**

Es correcto afirmar que, dado que el método propuesto consiste en un análisis empírico del proceso, información se perderá durante el proceso, puesto que el fenómeno de estudio es físico, por lo que no se puede entender al presente modelo propuesto como una representación total del proceso a evaluar.

#### 4. Marco teórico

Los motores de inducción del tipo jaula de ardilla son componentes eléctricos ampliamente conocidos por su uso en la industria debido a un conjunto variado de posibilidades. Sin embargo, muchas veces son sometidos a condiciones de funcionamiento no favorables, desencadenando la aparición de fallas que, si no son detectadas y solucionadas a tiempo, pueden causar paradas no planificadas, daños a equipos adyacentes y pérdidas económicas considerables.

Para evitar todos estos problemas, en toda empresa que emplea dichos equipos se hace uso de programas de mantenimiento, dentro de estos programas, la detección pronta de fallas en los motores eléctricos de corriente alterna (ACM) representa una gran ventaja. En este contexto, los métodos empleados más comunes han sido el análisis de vibraciones, la firma de corriente del motor y sobre todo la inspección visual y/o auditiva.

Aunque los métodos antes mencionado han presentado una gran efectividad, también tienen algunas limitaciones, como que pueden resultar invasivos (Inspección visual interna), y que además pueden ser susceptibles a interferencias externas (Análisis de frecuencia y vibraciones). En dicho contexto, como una alternativa no invasiva ni destructiva, surge desde hace unas décadas el uso del análisis termográfico, el cual es capaz de brindarnos información valiosa del componente en cuestión, a través del análisis de la distribución de temperatura superficial. Que además combinado con tecnologías pujantes como las redes neuronales, puede facilitar el factor humano que componía dicho método, mejorando aún más la eficiencia de dicha técnica.

Este proceso conjunto (análisis termográfico + clasificador NN) nos permitirá la detección de fallas basadas en patrones térmicos anómalos desarrollados en presencia de un problema mecánico y eléctrico, e inclusive distinguir entre las mismas para poder elucubrar respecto a la correcta medida correctiva; aunque obviamente para la obtención y buena ejecución de esta proceso serán necesarias técnicas de procesamiento de imágenes antes y después de ingresarlas al modelo clasificador RN, que serán explicadas en los siguientes puntos del presente apartado.

Por ello, para el adecuado entendimiento de este documento, es necesaria la previa definición de ciertos términos y palabras consideradas fundamentales en cuanto al tema de motores y todo lo relacionado:



#### 4.1. Redes neuronales (NN)

Una red neuronal es una técnica de inteligencia artificial, que tiene como objetivo enseñar a las computadoras a procesar datos de una manera similar a como lo haría el cerebro humano, utiliza nodos o neuronas interconectadas en una estructura organizada en capas (Amazon Web, s.f.).

#### 4.2. Motores de corriente alterna (AC)

Dispositivos electromecánicos que convierten energía eléctrica en mecánica, son ampliamente usados en industrias y en variedad de aplicaciones comerciales como bombas, ventiladores, compresores y sistemas de transporte. Dentro de esta clasificación se encuentran los motores de inducción de jaula de ardilla y de rotor bobinado, así como los motores síncronos.

#### 4.3. Fallas

Defectos, averías o condiciones anómalas que afectan el funcionamiento óptimo de un activo, en el contexto de los motores AC, estas pueden ser de naturaleza mecánica o eléctrica.

##### 4.3.1. Fallas mecánicas (MF)

Son causadas por problemas físicos en los componentes mecánicos del motor, entre estas se incluyen:

- **Desgaste en los rodamientos (BD):** Cuya principal causa es la fricción y la fatiga del material, provoca vibraciones excesivas que en cierto punto afectan notablemente la eficiencia del motor.
- **Desalineamiento (MSL):** Ocurre si las partes móviles del motor no están alineadas adecuadamente, también pueden causar vibraciones excesivas, aceleramiento en el desgaste de los componentes, etc.

##### 4.3.2. Fallas Eléctricas (EF)

Ocurren por anomalías en los circuitos eléctricos o componentes electromagnéticos del motor, entre estas se incluyen:

- **Cortocircuito en las bobinas del estator (ECF):** Causado por el desgaste excesivo del aislamiento entre bobinas, permitiendo el contacto entre dos conductores, puede provocar un flujo de corriente no deseado, capaz de dañar el motor.



- **Rotura de barras del rotor (BRB):** Este tipo de fallas es común en motores de jaula de ardilla, se refiere a la ruptura de una o más barras conductoras del rotor, es generalmente causada por fatiga o vibraciones anómalas, generan una pérdida de torque que puede llegar a ser significativa.

En cuanto a los fundamentos de la termografía, disciplina en la cual se basará el estudio del presente trabajo, serán necesarios también ciertos conceptos:

#### **4.4. Radiación**

La radiación es la energía emitida por la materia en forma de ondas electromagnéticas (o fotones) como resultado de los cambios en las configuraciones electrónicas de los átomos o moléculas, no requiere la presencia de un medio intervector (Yunus A. Cengel, 2011).

#### **4.5. Termografía por infrarrojos (TI)**

Es la ciencia que estudia el uso de dispositivos optoelectrónicos para detectar y medir la radiación, para posteriormente obtener la temperatura de la superficie bajo análisis. Los dispositivos que nos ayudan a la obtención de estas imágenes son las cámaras termográficas (Fluke Corporation, 2009).

Una de las etapas cruciales en el desarrollo del proyecto será el preprocesamiento de imágenes, para esto se presentan los conceptos clave estudiados:

#### **4.6. Preprocesamiento de imágenes (IPP)**

Conjunto de técnicas que se aplican a las imágenes antes de ejecutar cualquier análisis o procesamiento complejo. Su objetivo es mejorar la calidad, eliminación de ruido y/o resaltar características.

##### **4.6.1. Filtros**

La aplicación de filtros es una de las principales técnicas en el procesamiento de imágenes digitales, se pueden aplicar en el dominio del espacio o en el dominio de la frecuencia, tienen como objetivo suavizar la imagen, eliminar ruido, detectar bordes, etc. (Universidad de Sevilla, s.f.)

#### **4.6.2. *Contrast Limited Adaptive Histogram Equalization (CLAHE)***

Método que construye un histograma de intensidad por pixel, para luego aumentar la longitud de cada valor, de esta manera logra aumentar el contraste en toda la imagen.

#### **4.6.3. *Transformada de Wavelet Discreta (DWT)***

Es una herramienta matemática utilizada para la compresión tanto de imágenes como de audio, es una descomposición de señales en frecuencias. Permite analizar imágenes en diferentes escalas y frecuencias (Natalia Nieto, 2008)

#### **4.6.4. *Análisis de Componentes Independientes (ICA)***

Es un método computacional utilizado para separar una señal multivariante en subcomponentes independientes constituyentes, se utiliza en una amplia gama de aplicaciones, como el procesamiento de señales, reconocimiento de patrones, procesamiento de imágenes, etc. (Aalto university, 2008)

#### **4.6.5. *Segmentación de imágenes (IS)***

Es una técnica de visión por computadora que separa una imagen digital en distintos grupos de píxeles (segmentos) para facilitar la detección de objetos, entre otras tareas (IBM, s.f.). Existen numerosas técnicas de segmentación:

- *Umbralización*: Clasifican píxeles en función a si su intensidad está por encima o por debajo de un determinado valor umbral. Uno de los métodos más usados para determinar el valor umbral es el método de Otsu.
- *Histogramas*: Representan la frecuencia de los valores de píxeles en la imagen, se usan para determinar umbrales.
- *Detección de bordes*: Identifican límites de objetos o clases a través de la existencia de discontinuidades o contrastes.
- *Algoritmos watersheds*: Usan las imágenes convertidas a escala de grises y generan un mapa topográfico en base a la luminosidad de los píxeles.
- *Segmentación basada en clústeres*: Se dividen los datos visuales en grupos de píxeles con valores similares, una de las variantes más comunes es la técnica K-means.

#### **4.6.6. *Bag of Visual Words (BoVW)***

Es una técnica usada en trabajos de visión por computadora y aprendizaje automático, representa una imagen como un conjunto de palabras visuales que extraen características notables en la imagen (bordes, esquinas, texturas, etc.) (Karen Simoyan, 2015).

#### **4.6.7. *Speeded-UP- Robust Features (SURF)***

Es un algoritmo de detección de características en imágenes, tiene como objetivo identificar características significativas que puedan ser usadas en tareas de comparación, coincidencia, etc. Algunas de sus principales ventajas son su rapidez y robustez.

#### **4.6.8. *Hue Saturation Intensity (HSI)***

Modelo descriptivo de color similar al modelo RGB, donde se describen valores de los mismos, pero esta vez en base a las características de Intensidad, saturación y tono.

#### **4.6.9. *Linear Discriminant Analysis (LDA)***

Técnica para filtrar características de forma que las variables de ingreso sean reducidas en una dimensión que representa mucho mejor la dinámica de un proceso o imagen.

#### **4.6.10. *Principal Component Analysis (PCA)***

Es un método que mediante el análisis de una matriz de tamaño definido que almacena varios valores correspondientes a las características extraídas de una o más imágenes, calcula la matriz de covarianza, de la cual, mediante el cálculo de autovalores y vectores propios, tras lo cual puede determinar los grupos de características más representativas de la matriz original.

### **4.7. Clasificadores en base a características (IFE)**

Es la aplicación de técnicas con el objetivo de reducción de la complejidad y dimensionalidad de una imagen gracias a la extracción de características relevantes (Gonzales Pajares, 2016).

En cuanto a los algoritmos de detección y clasificación de fallas en imágenes, se presentan los modelos estudiados:

#### 4.7.1. *Redes neuronales convolucionales (CNN)*

Son arquitecturas de red para Deep Learning (DL) eficaces para la detección de patrones en imágenes, con el fin de reconocer objetos, clases y categorías, poseen capas de convolución en su estructura, estas capas pueden aplicar filtros convolucionales a las imágenes, para activar diferentes características de estas (MathWorks, s.f.), entre estas arquitecturas encontramos:

- *VGG16*: Desarrollada por Visual Geometry Group (VGG), presentada en 2014, se basa en 13 capas convolucionales y 3 capas completamente conectadas.
- *ResNet (Residual Networks)*: Desarrollada por ImageNet en 2015, su principal característica es el uso de bloques residuales, que permitan el armado de redes con muchas capas, existiendo ResNet-50, ResNet-101 o ResNet-152, con 50, 101 y 152 capas respectivamente.
- *Inception (Google Net)*: Desarrollada por Google, introduce el uso de bloques inception, permitiendo usar filtros de múltiples tamaños al mismo tiempo.
- *MobileNet*: Arquitectura ligera, pensada para equipo con recursos limitados o móviles, utiliza convoluciones separables, para reducir el número de parámetros.

#### 4.7.2. *K-Nearest Neighbors (K-NN)*

Es un clasificador de aprendizaje supervisado no paramétrico, que realiza clasificaciones o predicciones sobre la agrupación de un conjunto de datos, en base a la proximidad.

#### 4.7.3. *Support Vector Machine (SVM)*

Es un algoritmo de aprendizaje supervisado que se usa en problemas de clasificación y regresión, su objetivo es la obtención de un hiperplano que separe dos clases de puntos de datos.

#### 4.7.4. *Random Tree (RT)*

Es un modelo de aprendizaje automático utilizado para predicción y clasificación, que se caracteriza por emplear un enfoque aleatorio en su construcción, seleccionando de manera aleatoria un subconjunto de datos y características en cada fase del proceso. Su principal ventaja es que ayuda a



prevenir el sobreajuste del modelo a datos específicos, mejorando así su capacidad para hacer predicciones más generales y precisas.

#### 4.7.5. *Visual Transform (ViT)*

Modelo que adapta la arquitectura de *Transforms* para poder recepcionar imágenes, mediante el agregado de un paso previo donde la imagen es segmentada en un grupo finito de partes a los cuales se les conocer como *batches* y se le adapta de tal forma que pueda ser ingresada al *Transformer Encoder*. Para este método en particular y su correcta comprensión es importante tener en cuenta los siguientes conceptos claros:

- Multilayer Perceptron (MLP): Tecnología basada en el funcionamiento de una neurona, conforma la base de todos los modelos basados en Deep Learning.
- Feed Foward: Metodología de aprendizaje que permite que los MLP aprendan por cada *epoch* entrenada.
- Patch Division: División de una imagen en *patches* o imágenes más pequeñas, los cuales contienen un número menor de información.
- Positional Encoding ( $E_{pos}$ ): Matriz que guarda el posicionamiento de cada *Patch* en una sola matriz.
- Encoder Stack: Herramienta encargada de aplanar un input cualquiera y convertirlo en un vector representativo.
- Decodar Stack: Herramienta que convierte un vector representativo en un output o vector probabilístico que sirva como predicción de alguna clase.
- Auto Attention: Función encargada de relacionar N inputs con M outputs de tal forma que para una predicción M+1 discierne que N input es más importante para realizar la predicción.
- Multi Heading Self Attention: Múltiples funciones Auto Attention conectadas en paralelo.

Para un mejor entendimiento de este último modelo se presentará la siguiente explicación:



Tomando el modelo básico de una transformer normal tenemos:

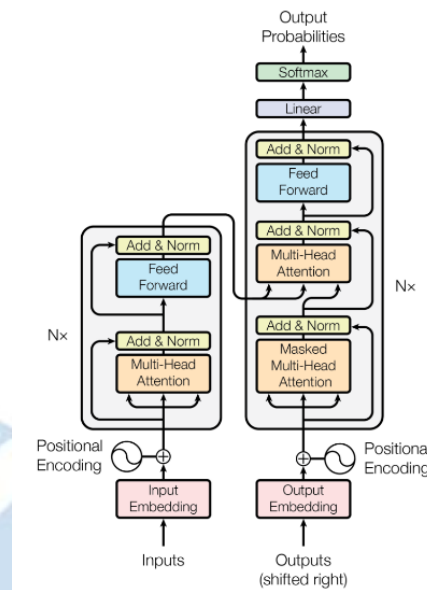


Imagen 1: Estructura de Transform (Fuente: All You Need its Attention)

Como se puede ver existen un proceso embedding donde los datos de entrada son computados de tal forma que se vuelven vectores representativos, a estos se les agrega el dato de la posición y finalmente se le ingresa a una capa Multi-Head Attention, esta primera capa se puede resumir su funcionamiento a las siguientes imágenes:

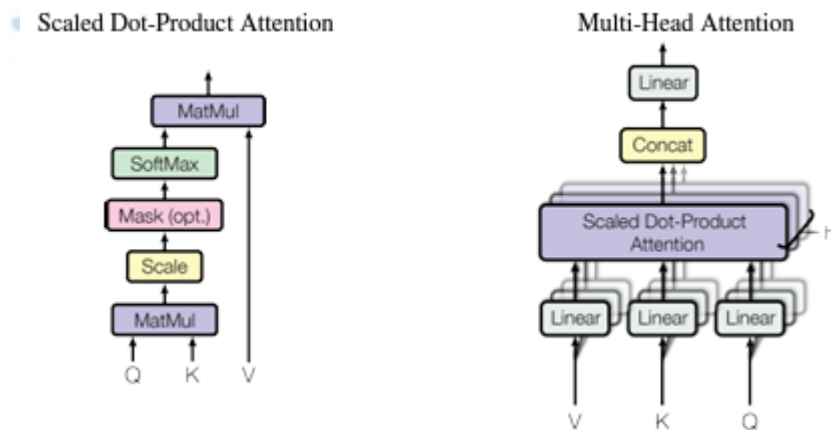


Imagen 2: Funcionamiento interno Attention (izquierda) - Entradas y salidas de Attention (derecho)(Fuente: All You Need its Attention)

Donde se puede observar cómo funciona una función Multi-Head Attention, las entradas que tiene este modelo son Values (V), Keys (K) y Queries (Q), las cuales funcionan mediante las siguientes ecuaciones:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Esta ecuación representa únicamente la multiplicación de los vectores  $K_i$  y  $Q_i$  para cada valor posible de  $i$ , esto con el objetivo de obtener la relación de correlación entre  $K$  y  $Q$ , pues los vectores que más se acerquen su multiplicación vectorial será mayor, mientras que, si se alejan entre sí, su multiplicación se acercara a 0, a esta tupla de productos se le multiplica la matriz  $V$  que representa el Positional Encoding, de esta forma del bloque o función Attention salen los vectores representativos de cada input así como la interrelación entre los mismos.

Esto puede resumir el comportamiento más básico de una función Attention, sin embargo, en una transformer estas capas Multi-Head son usadas 2 de maneras: almacenar las interrelaciones de los inputs con respecto a si mismo, este tipo de uso se puede observar los Transformers lo usan a la entrada de los inputs y outputs, además se usan también para relacionar los inputs con los outputs de forma que el decoder (Capa lineal + Función Softmax) obtenga un vector donde existan 2 auto relaciones y una relación input-output, un ejemplo práctico de esto último sería el siguiente:

*Mi – gato – tiene – hambre → My – cat – is – ...*

$(i_1) - (i_2) - (i_3) - (i_4) \rightarrow (o_1) - (o_2) - (o_3) - (o_4)$

Tengo una 4 palabras en español como input y 3 outputs en inglés, asumiendo que quiero determinar cuál es la cuarta palabra del proceso de traducción, entonces los Transformers permitirían no solo conocer un 4to valor en base a la iteración y aprendizaje de los 4 inputs ( $i_N$ ) y los 4 outputs ( $o_N$ ) disponibles, sino que también permite conocer la auto relación entre los 4 inputs, entre los 3 outputs previos y entre input y output, lo cual es extremadamente similar al proceso que sigue nuestro cerebro al realizar un análisis.

Finalmente, con respecto al uso de Transforms para imágenes tenemos la siguiente imagen:

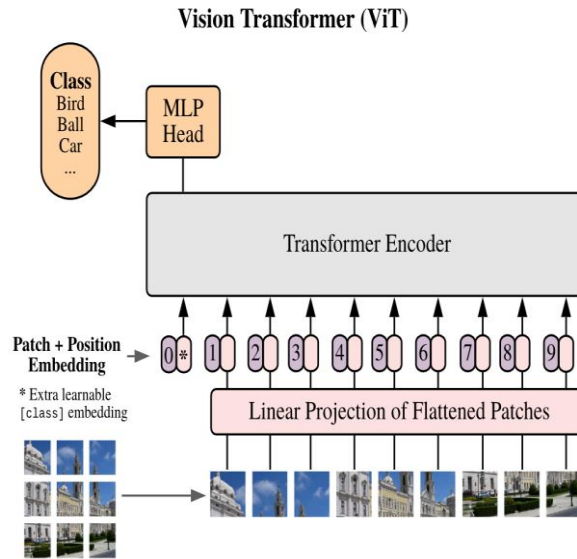


Imagen 3: Estructura básica de una ViT (Fuente: Google Research Team)

Acá se plantea un método de aplicación de los Transformers a imágenes mediante la descomposición de las mismas en  $N$  “patches” de tamaño  $P \times P$  píxeles, cada uno de estos patches es aplanado y proyectado a un espacio de dimensión fija, se refieren a esta proyección como los embeddings de los patches, esto nos permite tratar a una imagen 2D como una secuencia de datos, similar a como trataríamos a un texto, sin embargo, el transformer no es capaz de identificar la posición correspondiente a cada patch, es por eso que un embedding de posición es agregado a cada embedding de patch, para mantener la información de la posición. Los embeddings usados en el paper citado fueron de una dimensión (1,2,3,4...). Estos vectores resultantes formados por ambos embeddings ingresan al encoder.

El encoder del transformer presentado por (Vaswani, y otros, 2017) se compone de capas de MSA (multihead self-attention) alternadas con bloques MLP (multilayer perceptron), antes de cada bloque es aplicada una normalización y después de los mismos, se efectúan conexiones residuales.

Las conexiones residuales, introducidas por He et al. (2015) aplicadas a las redes ResNet, son una técnica que cambia la forma en como aprende cada capa, en lugar de usar una transformación directa de su entrada, se añade la entrada de la capa directamente a la salida.

$$Output_{capa} = F(x) + x$$

Donde:

$F(x)$ : son las transformaciones aplicadas en la capa

$x$ :: es la entrada a la capa

Esto ayuda solucionar problemas de degradación de gradiente, si solo se aplicaran las transformaciones  $F(x)$  en todas las capas, donde generalmente  $x$  corresponde a la salida de una capa anterior, el gradiente se propaga hacia atrás y puede disminuir (desaparición) o aumentar (explosión) exponencialmente, al realizar esta conexión residual después de las transformaciones, el gradiente se mantiene mucho mas estable a lo largo de las capas.

Acotado esto, (Dosovitskiy, y otros, 2021) nos plantean las ecuaciones que rigen su modelo:

$$z_0 = [x_{class}; x_p^1 E; x_p^2 E; \dots; x_p^N E] + E_{pos}$$

Donde:

$x_{class}$  es un token especial agregado para tener una representación de la imagen completa

$x_p^N$  son los patches en los que está dividida la imagen, siendo  $N$  el número de estos

$E$  es la matriz encargada de la linealización y proyección de los patches

$E_{pos}$  es la matriz que contiene los embeddings de posición de cada patch.

$z_0$  es la entrada al encoder

El encoder se define:

$$\begin{aligned} z'_l &= MSA(LN(z_{l-1})) + z_{l-1} & l = 1 \dots L \\ z_l &= MLP(LN(z'_l)) + z'_l & l = 1 \dots L \end{aligned}$$

Donde:

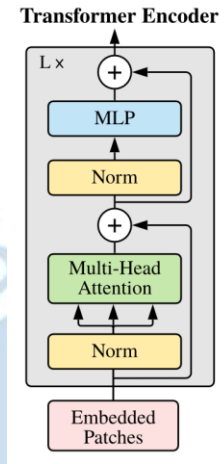
$LN()$  es la normalización

$z_{l-1}$  es la entrada a la capa MSA

$z'_l$  es la salida de la capa MSA, después de la operación de atención y la conexión residual, al mismo tiempo es la entrada a la capa MLP

$z_l$  es la salida del bloque MLP, después de la normalización, transformación y conexión residual

El bloque MLP contiene dos capas con una función de activación GELU no lineal.



*Imagen 4: Estructura básica de un encoder de una ViT (Fuente: Google Research, Brain Team)*



## 5. Estado del arte

La detección de fallas mecánicas en motores eléctricos es una labor fundamental en la industria debido a la diversa gama de aplicaciones de estos equipos. Los métodos convencionales de diagnóstico como el análisis de corriente suelen ser invasivos dado a que requiere contacto directo e interrupción del funcionamiento en planta. En consecuencia, el análisis termográfico surge como técnica no invasiva ni destructiva para la detección de fallas de diversa índole en motores eléctricos. Bajo esta metodología se emplea el uso de cámaras térmicas para capturar imágenes infrarrojas que revelan los gradientes de temperatura en los componentes del motor e identificar puntos calientes anómalos que indican fallas en la máquina. En los últimos años se ha profundizado en la detección de problemas en motores mediante el uso de las imágenes termográficas acompañado de técnicas avanzadas de procesamiento de imágenes y algoritmos de aprendizaje automático para automatizar y perfeccionar esta tarea.

Entonces para continuar con el presente trabajo, se presentarán los enfoques históricos más relevantes y recientes usados en la detección de fallas en los ACM mediante el uso de imágenes infrarrojas analizando la metodología propuesta por los mismos, resaltando que técnicas de procesamiento, métodos de eliminación de ruido y algoritmos de clasificación, etc. Que se hayan usado antes de la presente investigación.

Tenemos en primer lugar a (Singh Kalyan & Syal, 2023), que describen como la inteligencia artificial y el aprendizaje profundo se están empleando para mejorar las técnicas convencionales de diagnóstico de fallas en motores de inducción a medida que estos se vuelven más comunes. Resalta el gran potencial para la exploración en estudios futuros y la dificultad de detectar fallas en tiempo real actualmente por los métodos laboriosos de descomposición de señales.

En cuanto respecta al procesamiento de imágenes térmicas, el modelo de color HSV considera la región de tono (HUE) del modelo HSV debido a la utilidad para evaluación y medición del color de un objeto como la rojez o amarilleo predominantes en la imagen térmica. Además de técnicas para detección de bordes como Roberts, Prewitt y Otsu con el fin de segmentar las imágenes térmicas de equipos eléctricos defectuosos. La comparativa de rendimiento de umbralización de estos métodos se realizó calculando el error cuadrático medio y la relación señal ruido de pico siendo el método de Otsu el más efectivo para determinar las regiones más calientes en este tipo de imágenes en los trabajos de (Haider, Doegar, & Verma, 2018).

Si bien es cierto, trabajar con el color de las imágenes térmicas resaltando las zonas de calor es una buena técnica de preprocesamiento, no se toma en cuenta el ruido visual de la imagen. Entonces, la transformada discreta de Wavelet en dos dimensiones (2D-DWT) mejorando la relación señal-ruido para luego reducir la dimensionalidad de características a extraer identificando las más relevantes. Acto seguido calculan la MD (Distancia de Mahalanobis) para clasificar las características según su relevancia descartándose las menos relevantes. Todos estos datos son usados para entrenar un CDT (Complex Decission Tree) para predecir el estado de los rodamientos basándose en un esquema de validación cruzada, obteniéndose una precisión de clasificación de 99.85% (Padma Kala, Palanivel, Prasad Singh, Arulkumaran, & Hameed Shnain, 2021).

En contraste, (Xu, Teoh, & Ibrahim, 2024) diseñan un método híbrido para la detección de fallas en los rodamientos de motores de inducción cuyo preprocesamiento se basa en el uso de CLAHE, menos complejo que los anteriores mencionados. Sin embargo, se logra una mejor visualización de los detalles en las imágenes térmicas para resaltar los gradientes de calor y anomalías de temperatura para luego ser usadas en Inception V3 pero con un módulo SE integrado en su arquitectura para potenciar la extracción de características y en la última capa de clasificación de Inception V3 se reemplaza por un algoritmo (SVM) mejorando la capacidad de representación y generalización del modelo siendo este capaz de clasificar hasta 11 tipos diferentes de fallos de motor con una gran precisión acelerando el proceso de aprendizaje y estabilizando la pérdida durante el entrenamiento.

(Mahami, Rahmoune, Bettahar, & Benazzouz, 2021) presenta una nueva metodología para la clasificación de fallas en motores de inducción trifásicos. Como técnica de preprocesamiento de imágenes proponen el uso de BoVW (Bag of Visual Words) y SURF (Speeded-UP- Robust Features) para la extracción de características visuales. Siendo un modelo empleando ERT (Extremely Randomized Tree) con gran precisión y estabilidad en la detección y clasificación de fallas superando a métodos tradicionales como KNN, SVM y DT.

Después del preprocesamiento de las imágenes, se debe considerar un clasificador entre las fallas de los motores y el estado óptimo de operación de la máquina, entre ellos, los métodos basados en aprendizaje profundo utilizando imágenes térmicas pueden clasificar de manera efectiva las condiciones de los motores de inducción con una alta precisión y baja tasa de error. VGG-16 y ResNet-16, dos arquitecturas de redes neuronales

convolucionales surgen como las alternativas más fiables en este campo siendo la primera superior a la otra debido a su gran capacidad de aprendizaje profundo y manejo efectivo del problema de la desaparición del gradiente.

Del mismo modo, el uso de una red neuronal convolucional puede demandar un alto costo computacional, en este contexto (Karim Ayon, Siraj, Samad, Uddin, & Choi, 2024) introducen un enfoque de aprendizaje utilizando SqueezeNet modificada, un modelo de CNN de solo 7 capas con 16 veces menos parámetros que la original reduciendo significativamente el tiempo de entrenamiento y uso de recursos computacionales haciéndolo ideal para dispositivos móviles o sistemas embebidos. Del mismo modo, se diseñó una red prototípica aplicando Few-Shot Learning para que el modelo aprenda a clasificar fallos eficazmente con pocos ejemplos. En este estudio se llegó a la conclusión que la implementación de modelos más ligeros no necesariamente implica pérdidas significativas en la eficiencia en la detección de fallas en motores eléctricos usando imágenes térmicas.

Estos resultados han sugerido que los métodos basados en DL pueden predecir eficientemente las condiciones del motor incluso cuando se entrenan con datos pequeños y desbalanceados (Parveen, Hussain, Kumar, & Chowdhry, 2021).

Pese a ello, algunas arquitecturas con pesos pre-entrenados pueden presentar problemas. Entonces, se puede preparar el Dataset con el fin de forzar la falla de manera extrema y que esta se observe de manera evidente en la cámara infrarroja logrando una precisión de 95% empleando técnicas de aumento de datos para simular escenarios reales, la presencia de ruido y la precisión del modelo (Trejo-Chavez, y otros, 2023).

Aparte de la metodología mencionada anteriormente, existen algoritmos de clasificación que usan las características recopiladas en la etapa de preprocesamiento. Por ejemplo, la extracción de FOS (First Order Statistics) como datos extraídos de las imágenes. En (Sakalli & Koyuncu, 2022), realizan una comparativa entre clasificadores como redes neuronales, KNN (K-Nearest Neighbors), LR (Logistics Regressions) y SVM (Support Vector Machine). Llegando a la conclusión de que para el uso de FOS, se destaca RF (Random Forest) como mejor clasificador debido a su equilibrio entre precisión y estabilidad.

Mientras que (Navaid Sabah & Abdin Jaffery, 2022) basan su modelo en tres niveles de procesamiento incluyendo la mayoría de los puntos resaltados por los

anteriores autores. En el nivel bajo normalizan la data recortando, rotando y ampliando las imágenes y utilizando ecualización de histograma para mejorar el contraste de las imágenes. En el nivel medio se emplea una red de segmentación semántica con SegNet pre-entrenada con pesos de VGG-16. Para esto, se etiquetan y clasifican las imágenes en 3 clases: fondo, bobinado y falla. Posterior a ello, se entrena y valida la red hasta alcanzar una precisión del 99.29% destacando regiones de interés en las áreas calientes de los devanados. Finalmente, en el procesamiento de alto nivel se establecen tres variables de temperatura como entradas a un sistema de interferencia difusa clasificando el motor en tres niveles: regular, alerta y alarma.

Los estudios revisados muestran que la termografía infrarroja, complementada con técnicas avanzadas de procesamiento de imágenes y algoritmos de aprendizaje automático, ha demostrado ser altamente eficaz para la identificación y clasificación de fallas en motores eléctricos. Las actuales metodologías lograron mejoras en precisión y estabilidad de la detección de fallas mediante la combinación de diferentes técnicas, como el uso de modelos de color HSV, técnicas de eliminación de ruido (2D-DWT, CLAHE) y métodos de segmentación y extracción de características (BoVW, SURF, FOS). Complementados con algoritmos de clasificación como CNN, ERT y SVM permitiendo un análisis más complejo en la predicción de fallas.

Enfoques basados en aprendizaje profundo en arquitecturas como VGG-16, ResNet-50 e Inception V3 con módulos SE muestran resultados particulares por su capacidad de manejar Datasets desbalanceados y la posibilidad de implementarse en sistemas con recursos computacionales limitados como en SqueezeNet para su uso en sistemas embebidos en entornos industriales.

También existen soluciones novedosas, en (Vaswani, y otros, 2017) en la búsqueda de solucionar el problema de la pérdida de información en las RNN durante el entrenamiento, en ese contexto surgió la función *Attention* como una forma de relacionar los inputs y outputs, de esta forma la pérdida de información es mínima y la eficiencia del modelo aumenta. Este nuevo método llamado Transformer permite a los elementos analizados, llamados tokens, “elegir” a que otros tokens prestar atención, logrando de esta forma acelerar el proceso y no perder relación entre *patches* y aun mejor combinado con el uso de las funciones *Attention* en paralelo (*Multi-Head Attention*).

Es por ello que en (Dosovitskiy, y otros, 2021) se explora la posibilidad de usar la arquitectura propia de una *Transformer* en el ámbito de *computer vision* para tareas de



clasificación mediante el uso de una división en *patches* de la imagen original con el objetivo de reducir el peso computacional del modelo y agilizar las operaciones; asimismo usa un *Position Embedding* que sirva de índice para cada *patch*, con estos 2 valores y tras ser sometidos a un aplanamiento, se pueden aplicar un *Multi-Head Self Attention* (MSA) que permite predecir con criterios de relación, es decir aprende a saber que valores previos son los que deben influir más en la predicción futura específica. Este método es ampliamente usado para Datasets muy grandes dada su gran eficiencia además de robustez, y es este paper en conjunto con (Devlin, Chang, Lee, & Toutanova, 2019) que demostró que su empleo es extrapolable a usos que se pensaban imposibles por su costo computacional.

Sin embargo, los Transformers destacan de gran manera cuando se les entrena con Datasets grandes, esto es con respecto a otros modelos como por ejemplo las CNN, mientras que en Datasets pequeños no se nota una gran diferencia de rendimiento, es por ello que (Lee, Lee, & Song, 2021) usa 2 cosas: un tipo de partición de data en *Patches* especial denominado *Shifted Patch Tokenization* (SPT) donde mediante un conjunto de pasos establecidos amplía el número de *patches*, un ajuste de la función Self-Attention para hacer que se enfoque solo en los tokens relevantes (no nulls), eliminando la posibilidad de un token de construir relaciones con dichos tokens nulls.

A pesar de que las ViT según la bibliografía consultada se desempeñan mejor que las CNN, por ejemplo, en el caso de contar con datasets grandes, no dándose el mismo caso para datasets pequeños, por lo cual (Hassani, y otros, 2021) presento el concepto de Compact Convolutional Transformer (CCP), el cual se basa en no usar una simple división de la imagen para la separación en *patches*, sino en aplicar sobre estos *patches* una convolución que ayude a captar características locales similares a como lo hace una CNN, además de conservar la información espacial de la imagen.

Otros enfoques válidos para afrontar esta situación es la que plantea (Chen, Fan, & Panda, 2021) el cual se centra en una doble tokenization a escalas diferentes, es decir una imagen, la tokeniza o la divide en *patches* grandes y además hace el mismo proceso pero para *patches* más pequeños, es decir obtiene 2 tokens por imagen, de esta forma captura no solo la información global de la imagen, sino también la información local de los *patches* pequeños, ambos tokens son ingresados a un encoder propio el cual es conocido como *cross-attention*, y según los resultados observados, se notó un aumento en la eficiencia del modelo con este nuevo enfoque.

Finalmente, se lleva a la conclusión de que la integración de técnicas de procesamiento de imágenes avanzadas con algoritmos de aprendizaje automático ha transformado la detección de fallas en un proceso más eficiente y preciso. Sin embargo, aún se presentan desafíos como la detección en tiempo real y la mejora en clasificación con Datasets pequeños, aspectos que en el futuro serán claves para investigaciones con el fin de optimizar y mejorar el diagnóstico de fallas en motores eléctricos.



## 6. Metodología, materiales y procedimiento

En este capítulo exploraremos el método de investigación que se llevara a cabo con respecto al tema propuesto, así mismo de los materiales tanto de hardware como de software que se usaran para llevar a cabo la experimentación ideada, finalmente se explicara a detalle el procedimiento a seguir.

En primer lugar, es importante señalar que el presente trabajo es de carácter investigativo-experimental, el cual tras identificar una problemática (Fallas internas de un ACM) y una propuesta novedosa de solución (Imágenes termográficas), se propone a: revisar las herramientas actuales (métodos de segmentación, arquitecturas de CNN, estructuras de datos, etc.), experimentar con los mismos, con el fin de determinar la combinación adecuada de herramientas y modelos para obtener el mejor resultado posible.

Una vez logrado esto, se procederán a realizar las pruebas correspondientes al modelo ideado, para ello se usarán métricas de diversa índole que se han mencionado previamente en los apartados del capítulo 3 y 4 del presente informe y que además se detallaran su uso durante todo el capítulo de desarrollo.

Es por ello que la metodología para el presente trabajo se ha dividido en 3 etapas: Investigación, experimentación, validación. De las cuales cada una posee un conjunto de características propias que se pueden apreciar a continuación:



*Imagen 5: Estructura de metodología*

En la etapa de investigación, se realizará una búsqueda de en primer lugar papers o trabajos de investigación (trabajos de pregrado o postgrado) respecto al tema objeto de estudio, esto con el fin de determinar cuáles serán los objetivos, alcance y posibles técnicas a usar para lograr lo antes planteado, para ello se hará uso de un conjunto de portales web de talla nacional e internacional relacionados con la investigación científica, tales como: IEEEExplore, Scopus, ResearchGate, SciELO, etc.

De igual forma, durante esta búsqueda de información y tras tener en claro el objeto de estudio principal del presente trabajo, se procederá a revisar en primer lugar, los repositorios de las investigaciones consultadas, así como buscar en repositorios libres donde existen siempre un conjunto de información libre y gratuita respecto a diversos temas relacionados con las ciencias de computación, tales como: Kaggle, Github, Google Dataset Search, etc. De esta etapa se concluye con un conjunto exacto de técnicas y modelos, así como de un conjunto de datos (Datasets) que nos permitan alcanzar los objetivos planteados.

Una vez se haya concluido con la etapa previa, se procederá con la parte de experimentación donde se evaluarán los resultados obtenidos con las técnicas de preprocesamiento recopiladas, el objetivo principal de esta subetapa es conseguir la máxima diferenciación entre clases, de igual manera, dentro de esta subetapa, específicamente técnicas de extracción de características, se aplicará una evaluación para corroborar que las técnicas antes empleadas no alteraron las características innatas de la imagen original, propiedades como: desviación estándar, valor máximo, valor mínimo, correlación, etc.

Con respecto a las técnicas de preprocesamiento podemos listarlas por características específicas y por el procedimiento a seguir, para ello se puede consultar la Tabla 2 a continuación:

Método de preprocesamiento				Método de clasificación
Segmentación	Extracción de características	Diferenciador	Filtros	
Yolov5	FOS	CLAHE	FFT	CNN
Otsu Threshold	GLCM	Data Augmentation	LDA	SVM
K-means clustering	DWT	HSI	PCA	ERT
	CNN <sup>1</sup>			KNN
				ViT

*Tabla 2: Métodos de preprocesamiento y clasificación por desarrollar*

<sup>1</sup> Parte de una CNN puede ser usada como preprocesamiento (Consultar imagen 30)

Tras realizar lo antes mencionado, se diseñará un modelo clasificador que permita diferenciar primero 2 imágenes por tipo de falla o en caso así lo sea, sano. Para ello probaremos 3 métodos: el uso de una CNN, SVM, ERT y finalmente ViT. Después de conseguir satisfactoriamente resultados con ambos métodos podremos proceder con la última parte.

En la última parte, se procederá a realizar un análisis de validación, para ello se evaluará las estadísticas que representan el buen o mal desempeño de la CNN o ERT, dentro de las cuales se seleccionaron las siguientes:

Método de clasificación	CNN	SVM	KNN	ERT	ViT
Criterio por evaluar	Accuracy				
	Loss	-	Loss	-	Loss
	-	Matriz de confusión simplificada			
	-	-	-	OOB	-
	-	-	-	Numero de arboles	-
	-	-	-	Profundidad promedio	-
	Peso computacional del modelo				
	Rapidez de entrenamiento modelo				

*Tabla 3: Criterios de validación para los métodos de clasificación*

Con respecto a la última subetapa detallada en la imagen 1, es menester, comprobar de manera experimental empírica el funcionamiento adecuado de los modelos planteados y desarrollados en las etapas previas, es por ello que se probara los modelos con un pequeño Dataset fabricado a partir de motores pertenecientes a varias estaciones de bombeo y módulos de motor pertenecientes a varios laboratorios, las cuales incluyen: El módulo del motor de laboratorio de automatización (1), una estación de bombeo de aguas del edificio de la Facultad de Comunicación (2), una estación de bombeo de agua potable del edificio E (3), una estación de bombeo de aguas residuales del edificio 80 (4), un módulo de motor del laboratorio de electrotecnia (5) . Todo ello es gracias a la ayuda brindada por el equipo del laboratorio de mecánica, los cuales han accedido a prestarnos una cámara Termográfica Fluke Tis65; al igual que el equipo de electrotecnia y otros departamentos, que gracias a ellos hemos accedido a todas las instalaciones mencionadas



Esto nos servirá no solo para comprobar el buen desempeño de los modelos con máquinas reales, sino también para corroborar y medir la robustez de los mismos, a fin de determinar qué tan grande es la reducción de precisión de un entorno virtual con respecto a uno real.



*Imagen 6: Módulo de prueba del laboratorio de automatización (1)*



*Imagen 7: Estación de bombeo de la facultad de comunicación (2)*



*Imagen 8: Estación de bombeo del Edificio E (3)*



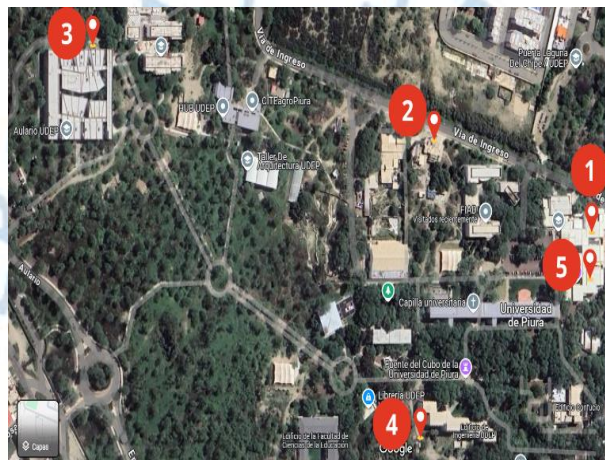
*Imagen 9: Estación de bombeo de aguas residuales del Edificio 80 (4)*



*Imagen 10: Modulo de motor del laboratorio de electrotecnia (5)*



*Imagen 11: Fluke TiS65 (Fuente: Fluke Corporation)*



*Imagen 12: Mapa de las instalaciones evaluadas}*



*Imagen 13: Imagen obtenida de la Fluke TiS65-Motor labSAC*

## 7. Desarrollo, experimentación y resultados

Una vez que ya se ha explicado de manera general cuáles serán los pasos y criterios por seguir, podemos pasar a resumir de manera específica que Datasets, métodos de preprocesamiento, clasificadores, criterios de evaluación se han usado y que tal ha sido su desempeño en el transcurso del trabajo.

### 7.1. Datasets revisados

Con respecto a los Datasets consultados y obtenidos podemos enumerarlos mediante la siguiente tabla:

Dataset	Fuente	N° Imágenes	N° Clases	Observación
Data1	Eksploracja i Niezawodność Journal	10007	105	Las fallas se han evaluado para distintos rangos de corriente de funcionamiento y se duplican, pues se posee 2 fuentes fotográficas para una misma falla.
Data2	Babol Noshirvani University of Technology	369	11	Las imágenes son obtenidas de un motor en funcionamiento libre, es decir sin carga alguna.
Data3	Universidad Autónoma de Querétaro	80	4	Las fallas inducidas son muy poco realistas
Data4	Universidad de Piura	479	1	Únicamente imágenes de motores sanos

*Tabla 4: Datasets disponibles de imágenes termográficas*

Con respecto a los Datasets es necesario aclarar que de todos los antes mencionados, solo se ha usado Data1 para el entrenamiento de los modelos clasificadores, puesto que es el que se ha considerado cumple todos los requerimientos antes planteados, a su vez también se debe esclarecer que para solucionar el problema del exceso de clases, se ha considera plausible la unificación de varias clases y el descarte de otras muchas siguiendo los 2 siguientes criterios: Unir todas las imágenes sin distintivo de corriente de funcionamiento para el caso de una misma falla en el



primer caso (unificación) y descarte de las imágenes de 1 de las cámaras, pues la otra presenta mejor resolución (descarte) en el segundo caso.

Habiendo entendido este contexto se presenta a continuación la estructura de la data objeto de estudio a continuación:

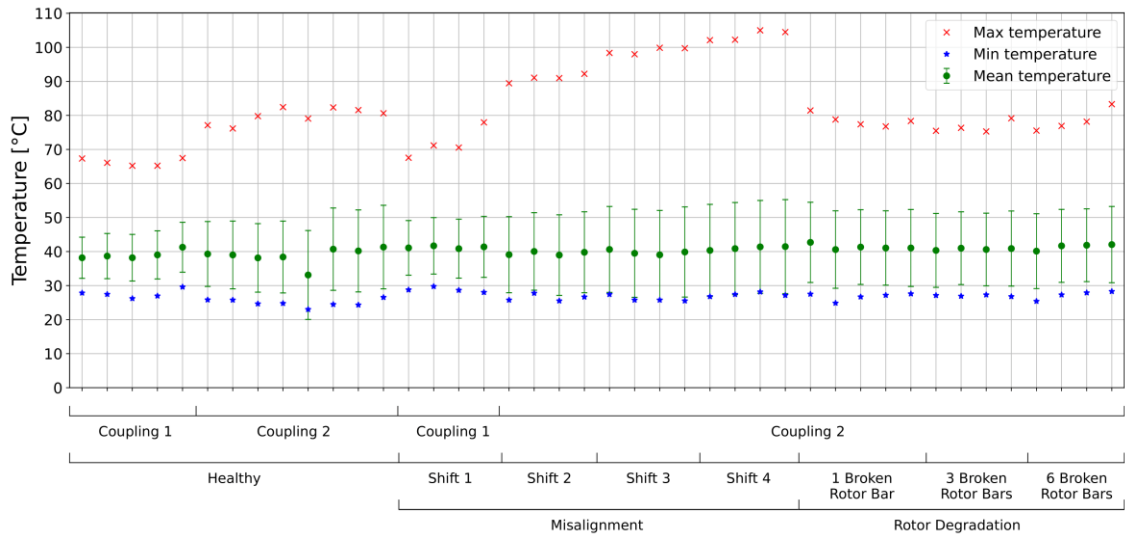


Imagen 14: Distribución de temperatura para la Data1 (Fuente: *Eksploracja i Niezawodność journal*)

La cual posee la siguiente cantidad de datos y clases:

Data set	Acoplamiento	Clase	Nº Datos
Data1	Coupling1	Healthy	394
		Missalingment1	1235
		Start up	2569
	Coupling2	Healthy	1495
		Missalingment2	900
		Missalingment3	1100
		Missalingment4	1178
		Rotor Failure1	185
		Rotor Failure3	454
		Rotor Failure6	487

Tabla 5: Estructura de la data filtrada

Llegados a este punto, es importante resaltar que para las pruebas de algunos de los modelos que se han llevado a cabo se consideró que el acoplamiento Coupling1 no tenía el suficiente número de clases por lo que se trabajó únicamente con las imágenes del Coupling2, mientras que en otros modelos si se usaron, pues nos puede



servir como segundo valor de validación, ya que a pesar de pertenecer al mismo dataset, presentara una dinámica diferente de interacción que las imágenes del “Coupling2”, posee una clase de “Missalingment”, por lo que la correcta o no identificación de esta clase con el modelo creado a partir de las otras imágenes puede ayudar a su mejora continua.

## 7.2. Preprocesamientos evaluados

De la misma forma tenemos una tabla que describe todos los métodos de preprocesamiento revisados y testados, así como cuál fue la apreciación que se concluyó al usarlos en la siguiente tabla:

Preprocesamiento	Tipo	Observación
Otsu	Segmentación	Eficiente pero solo puede “diferenciar entre 2 clases”
FFT	Diferenciación/Filtro	Permite acrecentar contrastes y de esta forma acrecentar las diferencias por clase.
YOLOV5	Segmentación	Muy útil, preciso y además logra identificar ACM en imágenes termográficas con una muy buena precisión.
CLAHE	Diferenciación/Filtro	Logro acrecentar contrastes en la imagen, sin embargo, depende de un filtro pasa alto previo para mejorar dicho proceso.
DWT	Extracción de características	Mejora el rendimiento del clasificador SVM, puesto que le proporciona un data input mucho más adecuado.
SIFT-BoVW	Extracción de características	Permite realizar una segmentación, pues identifica muy bien limites, sin embargo, es en conjunto con su método clasificador el que peor precisión ha obtenido.
K-means	Segmentación	Menor rendimiento que Otsu, pero permite “Diferenciar entre más de 2 clases”

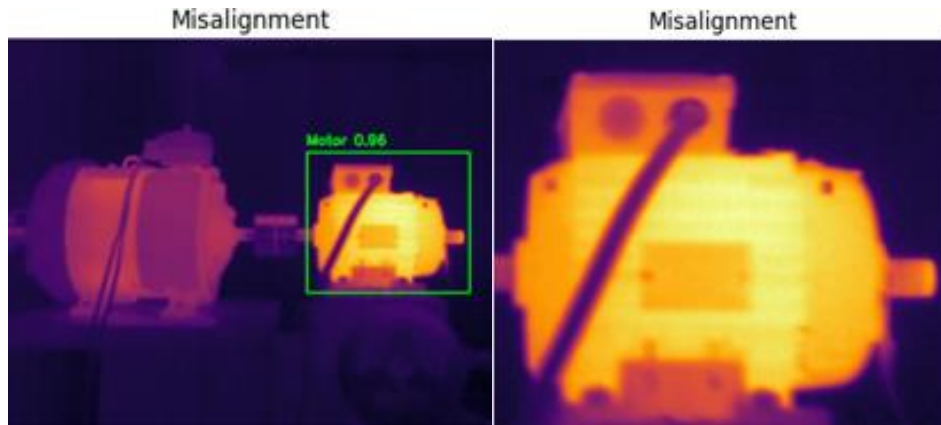
FOS	Extracción de características	Sirve para identificar patrones de intensidad por clase, también sirve para un posible análisis termográfico simple, pues permite convertir los valores de intensidad de pixeles en temperatura.
GLCM	Extracción de características	Indica un conjunto de información mucho más completa y de interés para la tarea de clasificación.
Data Augmentation	Aumento de data	Técnica ampliamente conocida en el ámbito, ha dado buenos resultados evitando el overfitting.
LDA	Diferenciación/Filtro	No significo una mejora sustancial en la precisión del modelo clasificador
PCA	Diferenciación/Filtro	No probó una mejora identificable o de gran impacto en los modelos probados.

*Tabla 6: Tipos de preprocesamiento evaluados*

Para que pueda ser mucho más entendible y visible todo lo señalado en la tabla 6, se presentara también mediante una breve descripción del proceso acompañado de imágenes, que es lo que hace cada preprocesamiento y porque es que se les han hecho las observaciones correspondientes:

### YOLOV5

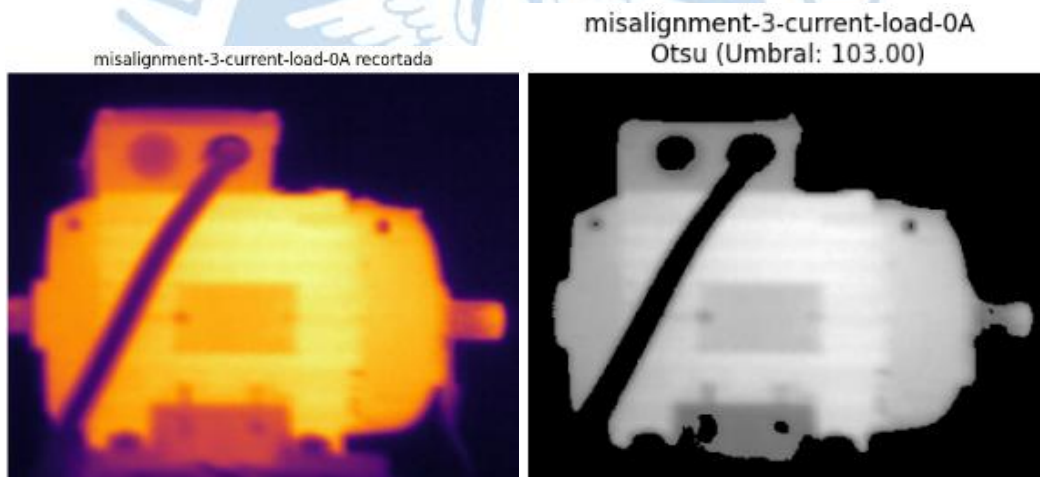
Herramienta ampliamente conocida, sirve para poder identificar clases para las cuales se le ha entrenado con un conjunto de imágenes etiquetadas, en nuestro caso se usaron 300 imágenes tanto de Data1 (180 imágenes ) como de Data2 (120 imágenes), puesto que se tratan de distintos tipos de imágenes de ACM, de esta forma nos aseguramos de que identifica correctamente a dicho objeto en cualquier imagen de carácter termográfico. Una vez entrenada la red, se usó a la misma para segmentar al motor, puesto no se quiere que la carga se vea involucrada en el análisis, puesto que dado tal o cual contexto, esta puede variar, y se pretende que el modelo propuesto sea escalable.



*Imagen 15: Método de segmentación por YOLOV5*

### Otsu

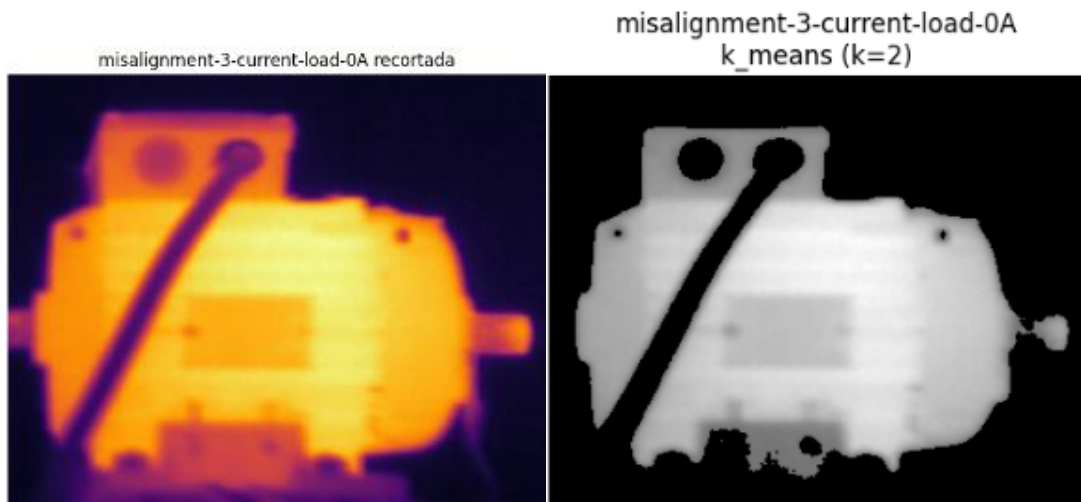
A partir de cálculos matemáticos halla un umbral donde la covarianza entre la intensidad de píxeles de la imagen por debajo y por encima del umbral es el mínimo posible, sirve para identificar regiones de interés.



*Imagen 16: Método de segmentación Otsu*

### K-Means

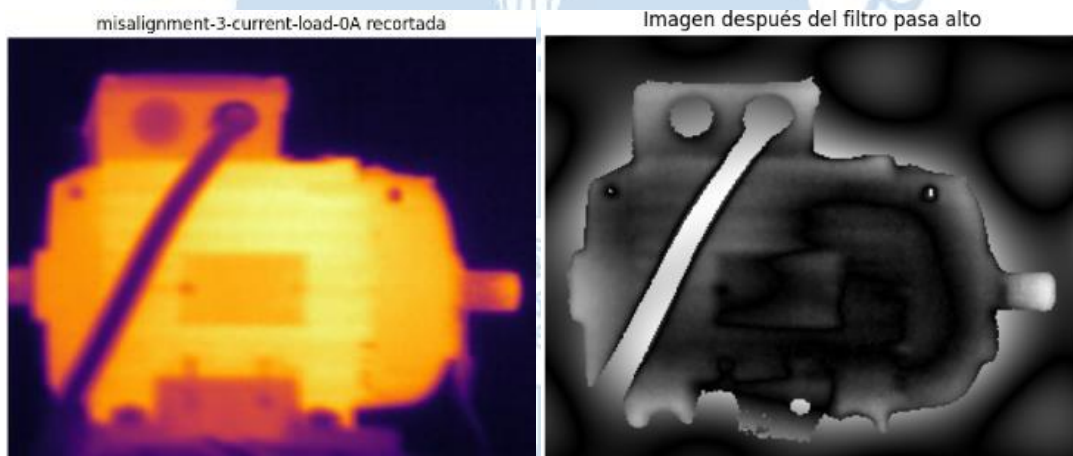
Similar método al de Otsu con la variedad que en este caso no se calcula un umbral, sino un valor máximo de límite y que además permite segmentar a la imagen en más de 2 grupos, cosa que no puede hacer el método de Otsu.



*Imagen 17: Método de segmentación Otsu*

### FFT

Mediante la transformada rápida de Fourier (FFT) se puede identificar las frecuencias de intensidad de la imagen y se le puede aplicar un filtro, esto sirve para acrecentar los contrastes.

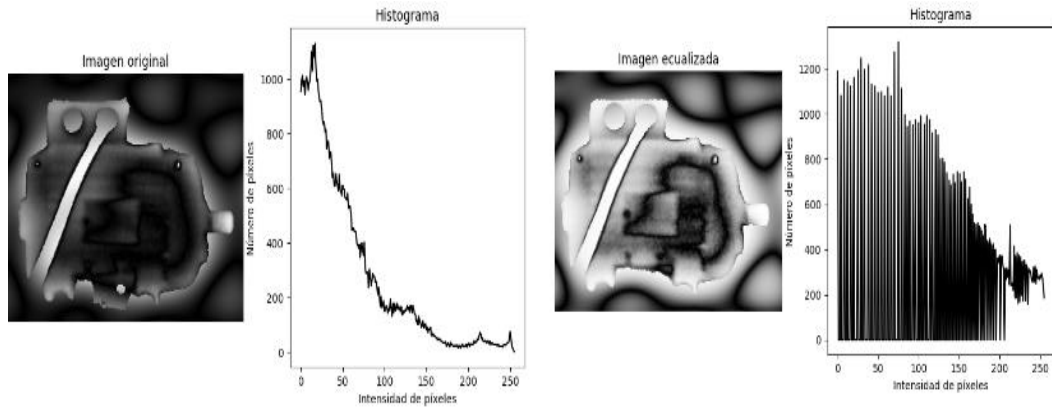


*Imagen 18: Método de diferenciación FFT*

### CLAHE

Este método consiste en hacer evidentes los pequeños detalles a través del aumento del contraste de forma adaptativa, lo que evitará contrastes excesivos que podría amplificar ruidos no deseados, en el contexto de imágenes termográficas, puede ser de utilidad para resaltar zonas con cambios de temperatura no tan evidentes.

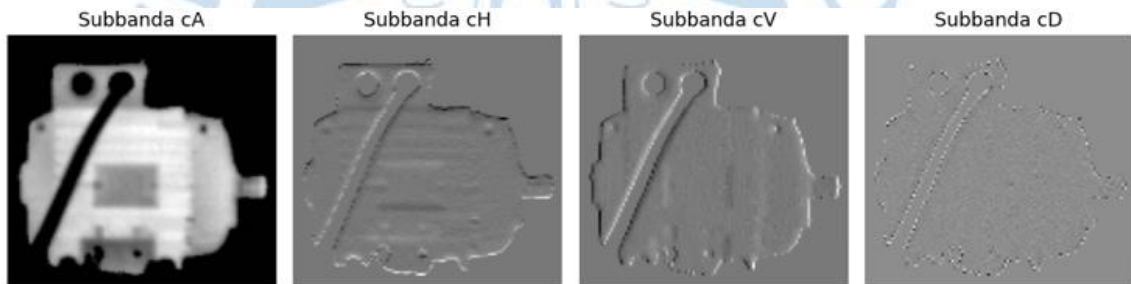




*Imagen 19: Método de diferenciación CLAHE*

### DWT

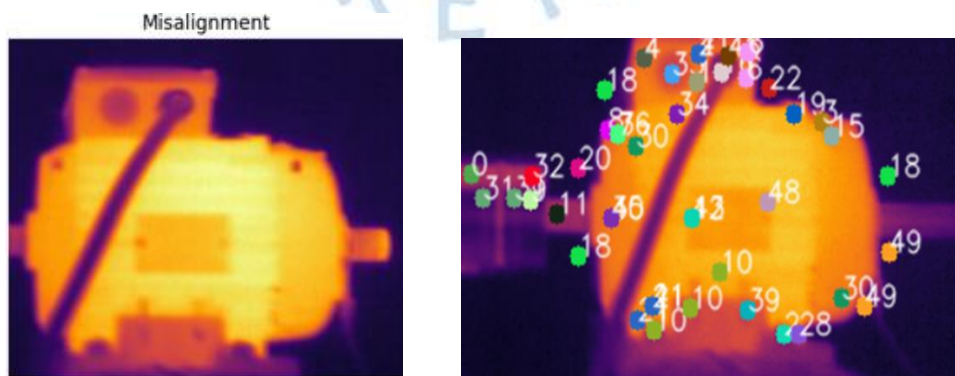
Método que genera a partir de una imagen de entrada, 4 imágenes de salida, las cuales corresponden a diferentes rangos de frecuencia de intensidad de pixel, en nuestro caso pudiendo obtener los siguientes resultados:



*Imagen 20: Método de extracción de características DWT*

### SIFT-BoVW

Mediante el uso de un algoritmo, identifica puntos donde la variación de intensidad entre pixeles es bajo y asigna un radio y una dirección por cada punto crítico, con dicha información luego genera un histograma.



*Imagen 21: Método de extracción de características DWT*



## FOS

Herramienta que calcula las características principales de una imagen, intensidad máxima de pixel, intensidad mínima, intensidad media, covarianza, etc.

Imagen	Media	Varianza	Desviación Estándar	Asimetría	Curtosis
imagen_0.png	61.71213	2130.697847	46.15948274	1.832901	2.601704
imagen_1.png	62.6044	2331.376332	48.28432802	1.776234	2.250029
imagen_2.png	72.37719	2165.510021	46.53504079	2.246141	4.133076
imagen_3.png	57.51032	2032.67504	45.08519757	1.966105	3.283663
imagen_4.png	50.40992	1924.043893	43.86392473	2.124524	4.272267
imagen_5.png	49.24174	1993.750524	44.65143361	2.118781	4.235394
imagen_6.png	52.07803	1972.632727	44.41433019	2.14005	4.294202
imagen_7.png	50.90589	2022.760983	44.97511516	2.116758	4.230763
imagen_8.png	46.02393	1986.415254	44.56921868	2.165518	4.61898
imagen_9.png	47.95286	1952.831646	44.19085478	2.20475	4.741205
imagen_10.png	45.11593	1965.698086	44.33619386	2.173449	4.610371
imagen_11.png	48.1983	1911.702196	43.72301678	2.242676	4.968415
imagen_12.png	45.3232	2026.255582	45.01394875	2.18446	4.697631
imagen_13.png	45.76936	2025.194198	45.0021577	2.182252	4.687402
imagen_14.png	56.9588	1677.337209	40.95530746	2.74602	7.514977
imagen_15.png	47.51338	1882.308659	43.38558123	2.281946	5.247238

Imagen 22: Método de extracción de características FOS

## GLCM

Herramienta muy similar al FOS, extrae datos estadísticos de la imagen.

GLCM Properties current-load-4A				
Contrast: [[	309.87	550.58	275.69	340.75]
[	699.56	550.58	639.08	340.75]
[	1071.7	1190.6	1045.4	765.06]]
Correlation: [[	0.97174	0.94967	0.97477	0.96885]
[	0.93627	0.94967	0.94138	0.96885]
[	0.90249	0.89105	0.90388	0.92999]]
Energy: [[	0.034106	0.027298	0.030457	0.027841]
[	0.025862	0.027298	0.022853	0.027841]
[	0.022017	0.020457	0.019395	0.021023]]
Homogeneity: [[	0.27917	0.17759	0.21708	0.18844]
[	0.16852	0.17759	0.12844	0.18844]
[	0.12646	0.1016	0.093506	0.11422]]

Imagen 23: Método de extracción de características GLCM

## LDA

Es un método que sirve para discernir que tipo de características varían más en un conjunto de datos, de esta forma filtra los valores más importantes, los cuales posteriormente serán ingresados al método de clasificación.

## PCA

Al igual que el LDA es un método que sirve para discernir que tipo de características son más importantes de acuerdo con su valor de covarianza entre sí.

## Data Augmentation

Herramienta que se usa para aumentar el número de imágenes a partir de una sola, mediante la duplicación de la imagen original, pero con algún cambio general (Orientación, giros, etc.), en nuestro caso se usaron los siguientes parámetros:

**Rotation\_range=10:** Las imágenes pueden ser rotadas aleatoriamente hasta 10 grados en cualquier dirección.

**Width\_shift\_range=0.1:** La imagen se podrá desplazar hasta un 10% de su ancho a la izquierda o derecha.

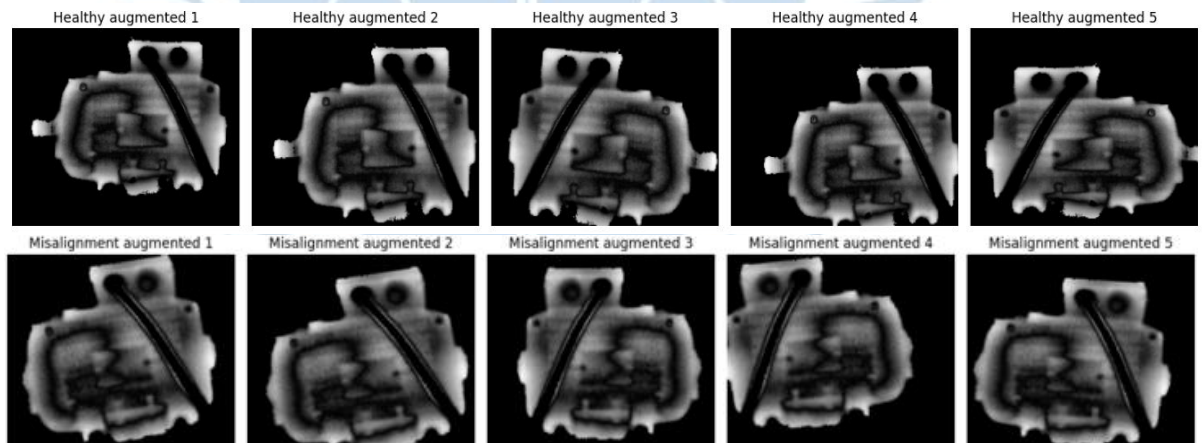
**Height\_shift\_range=0.1:** La imagen se podrá desplazar hasta un 10% de su altura hacia arriba o abajo.

**Zoom\_range=0.1:** La imagen puede ser ampliada o reducida hasta un 10% del tamaño original

**Horizontal\_flip=True:** Algunas imágenes generadas pueden ser volteadas horizontalmente.

**Fill\_mode='nearest':** Los nuevos pixeles serán rellenados copiando al pixel más cercano, en este caso, serán pixeles negros.

He aquí unos ejemplos de cómo funciona esta herramienta:



*Imagen 24: Generación de datos por Data Augmentation*

### 7.3. Métodos de clasificación

Continuando con el desarrollo del modelo, tras realizar la evaluación de los métodos de preprocesamiento vistos en la tabla 6, debemos mencionar todos los métodos de clasificación que se han considerado, así como sus características principales:

Método de clasificación	Estructura	Observaciones
VGG16	Convolutional neural network	Es de estructura simple y fácil de comprender.
ResNet50		Ampliamente usada para labores de clasificación.
InceptionV3		Permite la reducción de convoluciones mediante factorización.
EfficientNetB2		Modelo mucho más pesado que las otras CNN pero que no muestra una mejora comparativa con respecto a otras CNN.
EfficientNetB5		
EfficientNetB7		
ERT	Ensemble supervised machine learning method	Posee un método interesante de clasificación basado en la prueba y error.
SVM	Supervised learning algorithm	Es un método que solo admite entradas 1D
KNN	Non-parametric, supervised learning classifier	Dio peores resultados que cuando se usó SVM como clasificador
ViT	Transformer	Excelentes resultados, pero gran peso computacional en comparación con SVM
ViTFSD	Transformer	Mejoro los resultados obtenidos en comparación con las CNN y ViT, pero demoro más en entrenar
CCT	Transformer	Buenos resultados, pero hubo una época en la que el resultado fue pésimo, pero es mucho más liviana que la ViTFSD
CrossViT	Transformer	El modelo más pesado de todos y no represento una mejora sustancial

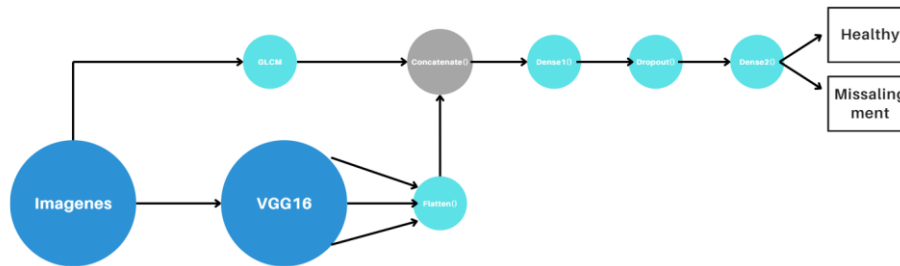
*Tabla 7: Métodos de clasificación y sus observaciones*

A continuación, y al igual que los preprocesamientos, se detallará la estructura que se le dará a cada método de clasificación a continuación:

### VGG16

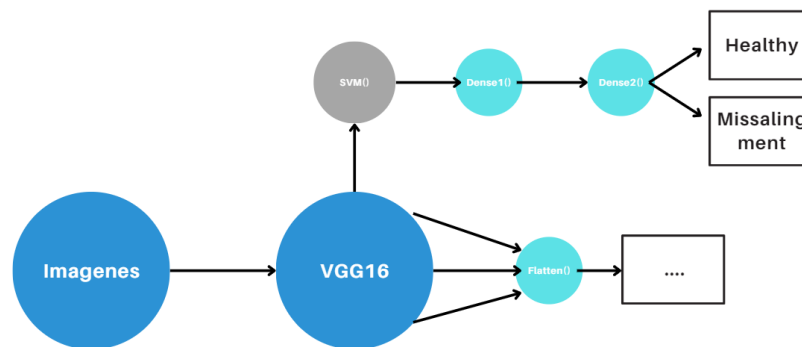
En el caso de VGG16 se dieron 2 casos:

- VGG16-1: Se mantuvo íntegra la estructura original de la CNN, pero a la salida de esta se le concateno la metadata obtenida de la imagen mediante GLCM, para lograr eso, la salida de VGG16 se le aplico una función `flatten()`, esta salida se le conecto a una capa Dense y luego se le aplico una función `dropout()` para tratar de minimizar el overfitting pues apagaba la mitad de las neuronas aleatoriamente, finalmente se le conecta la salida de esto a una capa Dense de 2 salidas.



*Imagen 25: Estructura del método de clasificación VGG16-1*

- VGG16-2: Se entrena a la arquitectura normal de VGG16, pero se extraen, o en términos correctos, no llegan a usarse nunca las capas Dense de la CNN encargadas de la clasificación, puesto que, se usará un SVM para clasificar, por lo que en este caso solo se extraerán los pesos asignados a los características calculadas por el entrenamiento de la VGG16.



*Imagen 26: Estructura del método de clasificación VGG16-2*

### ResNet50

En el caso de la CNN ResNet50 se probó un entrenamiento doble usando al igual que en el caso de la VGG16 metadata obtenida mediante GLCM, la estructura del modelo se detalla a continuación:



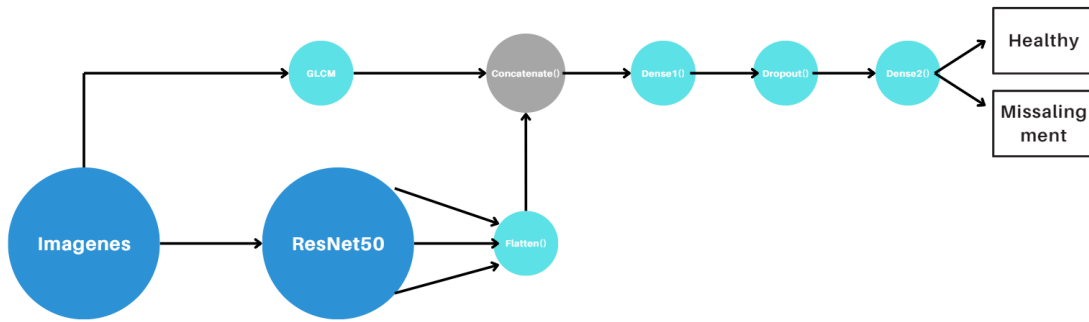


Imagen 27: Estructura del método de clasificación ResNet50

### InceptionV3

En el caso de la arquitectura que tienen involucrados a la CNN inceptionV3 se tienen 2 casos, cada uno de los cuales presenta la similitud que únicamente se usan las capas *Convolutional* y *Pooling* de la arquitectura original, pues las demás capas *Dense* no son usadas para la tarea de clasificación, pues para ello se usan; en el primer caso una SVM (InceptionV3-1) y en el segundo una LDA y posteriormente una KNN (InceptionV3-2).

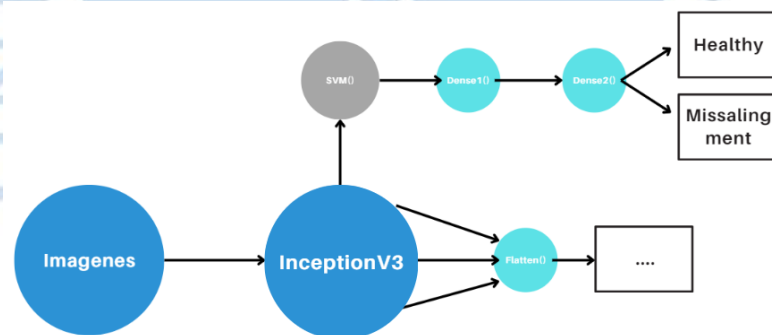


Imagen 28: Estructura del método de clasificación InceptionV3-1

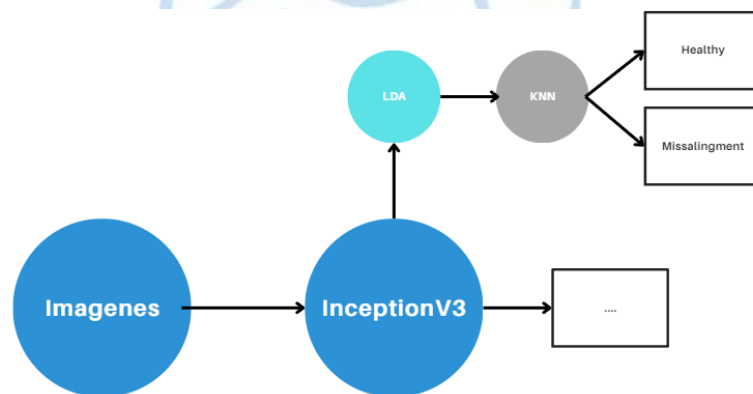
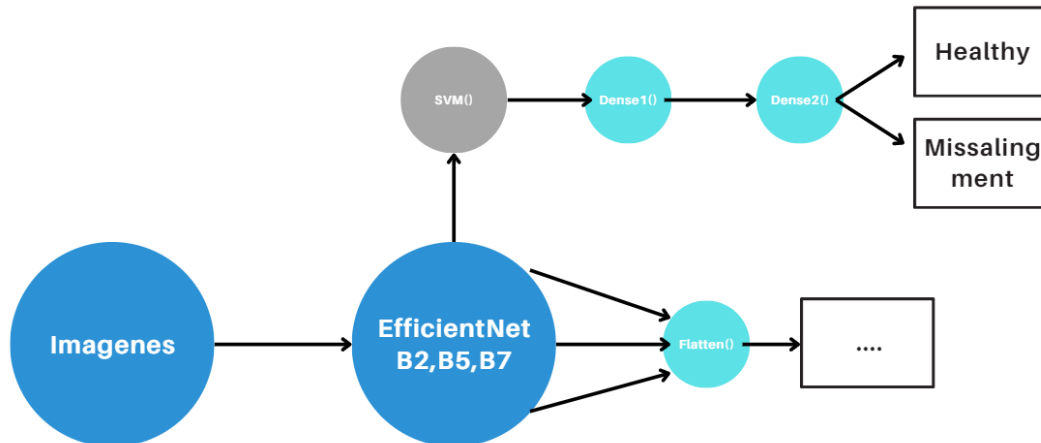


Imagen 29: Estructura del método de clasificación InceptionV3-2

### EfficientNetB2,B5 y B7

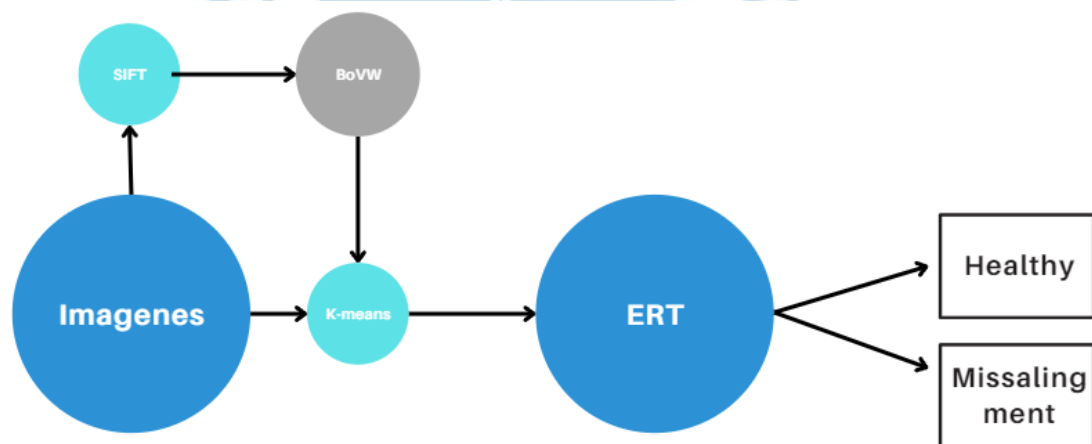
De igual forma que en el caso de la estructura de la VGG16-2 y que el primer caso de la InceptionV3-1, en esta estructura solo se usan los pesos entrenados de la red con respecto a las características de las imágenes *Input*.



*Imagen 30: Estructura del método de clasificación EfficientNetB2,B5,B7*

### ERT

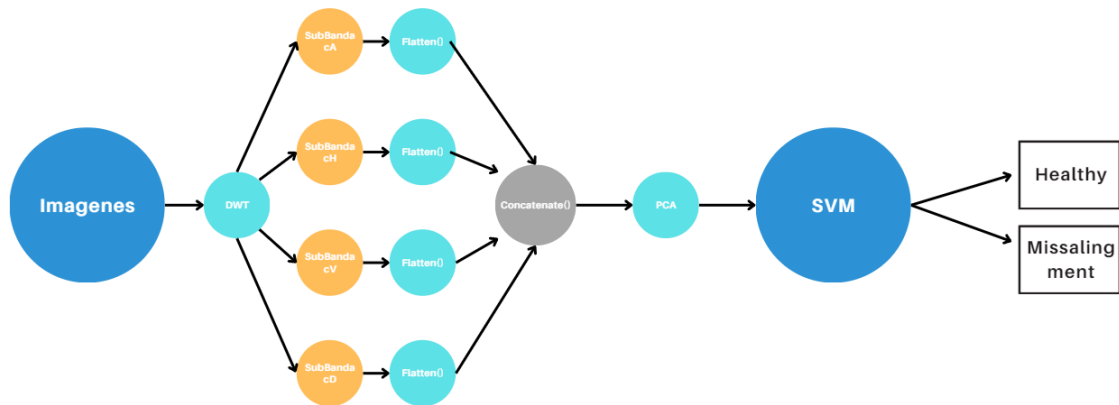
Este caso es particular pues la arquitectura de un ERT no le permite el ingreso de imágenes del tipo que usamos, al contrario que las CNN, por lo que debe haber un tratamiento previo, es por ello y dado que se encontró un método interesante, donde mediante el uso de SIFT+K-means que se pueden obtener las características de las imágenes en forma de matrices que representan histogramas correspondientes a lo detallado en el apartado donde se hablaba del método de preprocesamiento BoVW.



*Imagen 31: Estructura del método de clasificación ERT*

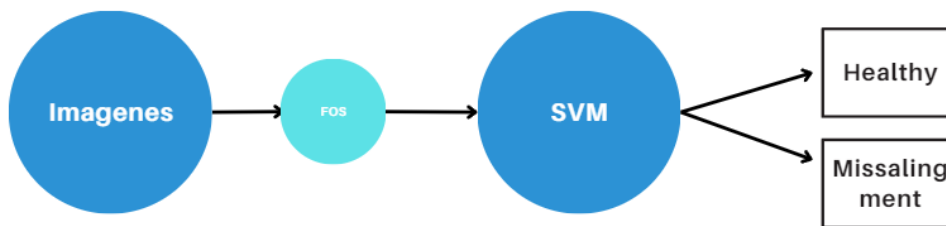
## SVM

En el caso que veremos a continuación es un método diferente a los previamente vistos, pues en este caso las características de interés son calculadas no mediante una CNN, sino mediante una PCA, a esta estructura se le conocerá como SVM-1



*Imagen 32: Estructura del método de clasificación SVM-1*

Sin embargo, dado que la herramienta DWT no es el único método por el cual se pueden extraer características de una imagen, también se ideó un método con el cual se combinada el clasificador SVM con el extractor de parámetros FOS, a dicho método se le conocerá como SVM-2.



*Imagen 33: Estructura del método de clasificación SVM-2*

## KNN

De forma muy similar a como lo fue la clasificación mediante el uso de una CNN como extractor de características, en este caso las características que se extrajeron, fueron mediante un método mucho menos sofisticado, simplemente se calcularon las FOS de cada imagen mediante la biblioteca correspondiente, posterior a ello se le aplicó un LDA y dichos datos o parámetros fueron ingresados a una KNN que funciona como clasificador.

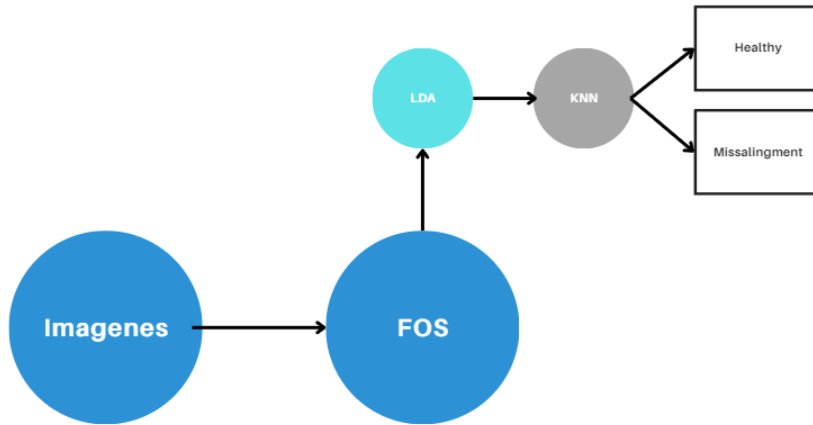


Imagen 34: Estructura del método de clasificación KNN

## ViT

Tal y como se explicó en el capítulo 4.7.5 esta arquitectura destaca por su robustez y eficiencia, sin embargo, no resaltan dichas características en el caso de un entrenamiento con pobre data, es por ello que se usaron modelos específicos de ViT enfocados a resolver esta problemática, dichos modelos y su funcionamiento se detallan a continuación:

- Shifted Patch Tokenization (SPT): permite a los parches capturar más información mediante un campo de recepción ampliado y se logra de la siguiente manera:
  - i. Desplazamiento de la imagen de entrada un total de píxeles correspondiente a la mitad del tamaño del parche, en las 4 direcciones diagonales.
  - ii. Las 4 imágenes desplazadas son concatenadas con la original.
  - iii. Se descompone la concatenación resultante en parches como se haría normalmente, para luego aplanarlos y proyectarlos, obteniendo 5 veces más información que en una ViT tradicional.

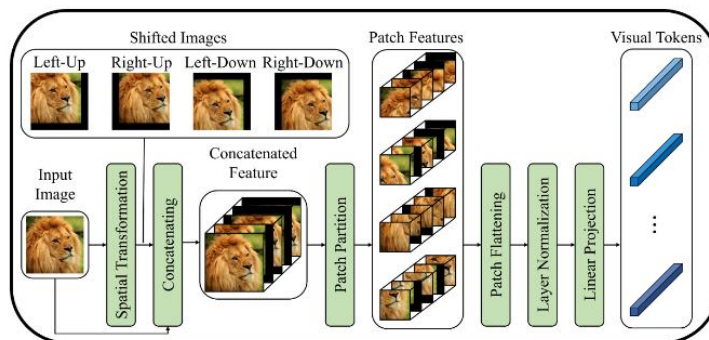


Imagen 35: Estructura del SPT (Inha University)

- Locality Self-Attention (LSA): Es un ajuste al mecanismo de atención tradicional para hacer que se enfoque solo en tokens relevantes, eliminando la posibilidad de un token de construir relaciones con él mismo.

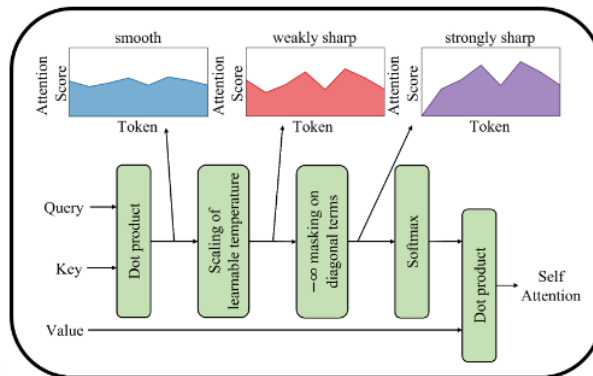


Imagen 36: Estructura del LSA (Inha University)

- Compact Convolutional Transformer (CCT): el cual se basa ya no usar una simple división de la imagen para la separación en parches, sino en aplicar sobre estos parches una convolución que ayude a captar características locales similares a como lo hace una CNN, además de conservar la información espacial de la imagen.

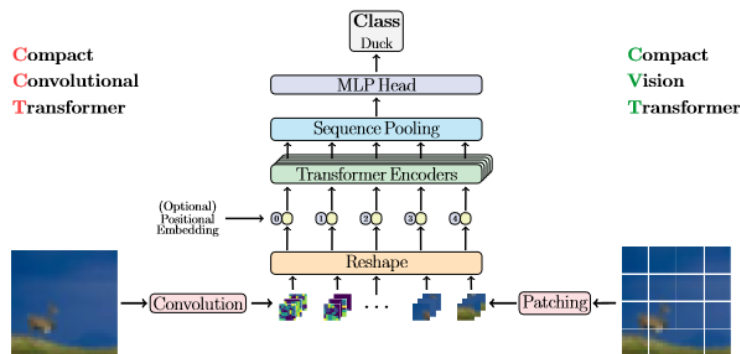


Imagen 37: Estructura de la CCT (University of Oregon)

- CrossViT: mecanismo que permite a ambas ramas generar relaciones entre ellas, maximizando la eficacia, mediante una doble tokenización a dos escalas diferentes, la escala en pequeños parches ayuda a capturar información local de pequeñas áreas, mientras que los parches grandes permiten capturar información más global de la imagen, estas dos ramas de tokens son ingresadas a su propio encoder y analizadas juntas.



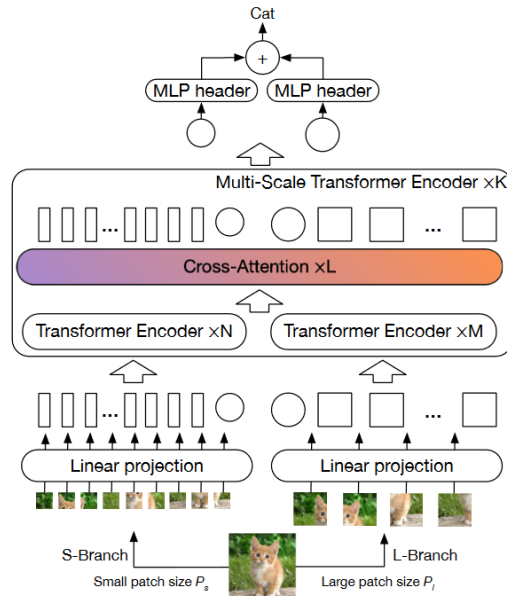


Imagen 38: Estructura de una CrossViT (MIT-IBM Watson AI Lab)

En cuanto a sus parámetros de desarrollo y resultados de entrenamiento tenemos

#### ViT o Simple ViT

- Tamaño de las imágenes:  $256 \times 256$  píxeles
- Tamaño de los patches:  $32 \times 32$  píxeles, cada imagen se dividirá en parches de  $32 \times 32$  píxeles, dando un total de 64 patches.
- Número de clases: 3 (Healthy, Misalignment y broken rotor bars)
- Dimensión del embedding: 512, cada parche de proyectará en un vector de 512 de dimensión.
- Profundidad del transformer: 8, define cuantos bloque de atención y Feed forward tiene el modelo.
- Número de cabezas de atención: 8, define el número de cabezas de atención presente en cada bloque.
- Dimensión del MLP en el Feed-forward: 1024, es la dimensión de la capa oculta del Feed forward
- Tasa de dropout: 0.2, es una técnica de regularización que ayuda a evitar el sobreajuste, significa que el 20% de las neuronas en los bloques de atención y Feed forward se apagan aleatoriamente durante el entrenamiento.
- Tasa de dropout en el embedding: 0.2, ayuda a regularizar la proyección de los parches.

A continuación, se presenta un ejemplo de la división en parches de algunas imágenes del dataset.

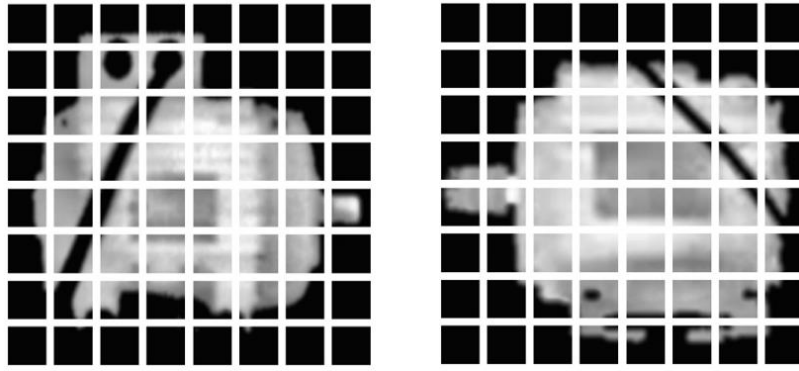


Imagen 39: Representación de los parches en 2 imágenes

Se presenta un esquema de la estructura de la ViT:

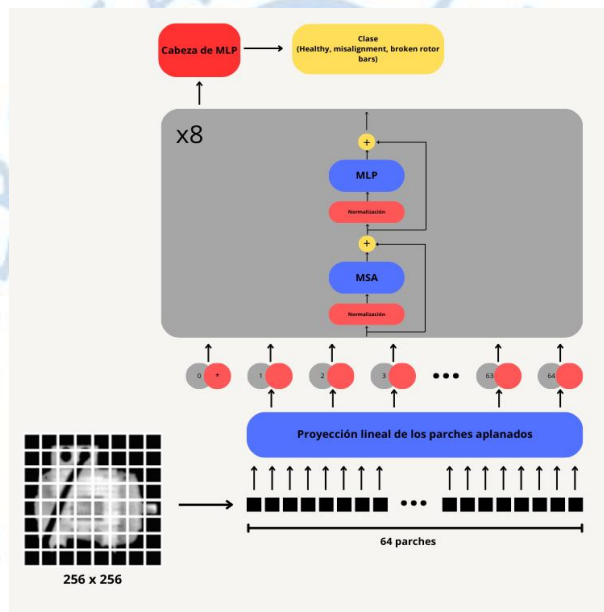


Imagen 40: estructura de la ViT aplicada

Una vez establecidos los parámetros del modelo, se pueden delimitar los parámetros de entrenamiento del mismo.

Al analizar los datos disponibles, se observa una clara descompensación entre las cantidades de imágenes en las 3 clases, es por eso que se optó por usar pesos de clase durante el entrenamiento, para compensar este desequilibrio:

$$class\ weight_{healthy} = \frac{9280}{3136 * 3} = 0.986$$

$$class\ weight_{misalignment} = \frac{9280}{4745 * 3} = 0.6519$$

$$class\ weight_{broken\ rotor\ bars} = \frac{9280}{1399 * 3} = 2.211$$

Otros parámetros utilizados en el entrenamiento fueron:

Optimizador: se utilizó el Adam (Adaptive Moment Estimation)

Taza de aprendizaje:  $1 \times 10^{-4}$

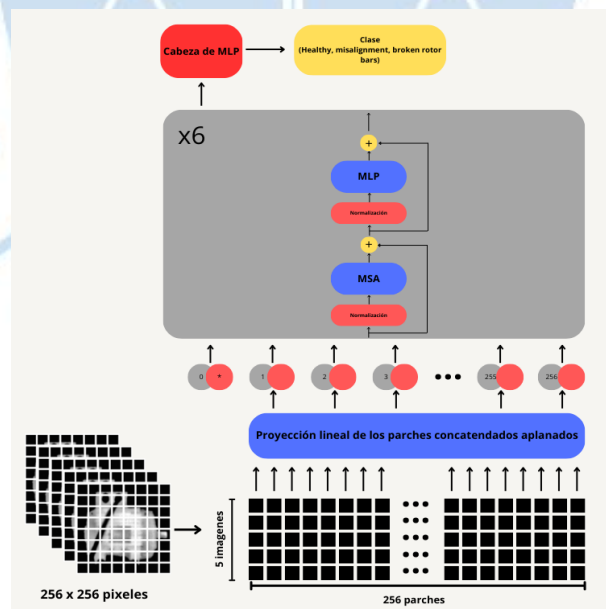
Decaimiento de pesos:  $1 \times 10^{-5}$

Número de épocas: 20

### ViTFSD o ViT For Smaller Data

- Tamaño de la imagen: 256 x 256 píxeles
- Tamaño de los parches: 16 x 16 píxeles
- Numero de clases: 3
- Dimensión del embedding de cada parche: 1024
- Número de capas del transformer: 6
- Número de cabezas en la capa de atención multi-cabeza: 16,
- Dimensión oculta de la red Feed-forward (MLP): 1024
- Tasa de dropout aplicada en las capas de atención: 0.12,
- Tasa de dropout aplicada a los embeddings de entrada: 0.12

A continuación, se presenta un esquema de la estructura del modelo:



*Imagen 41: Estructura de la ViT para datasets pequeños*

Los pesos de clase fueron los mismos calculados para el modelo de ViT tradicional.

Optimizador: se utilizó el Adam (Adaptive Moment Estimation)

Taza de aprendizaje:  $1 \times 10^{-4}$

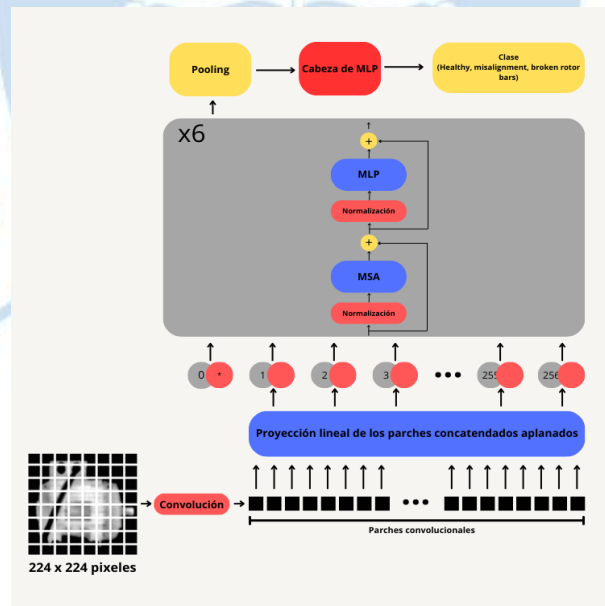
Decaimiento de pesos:  $1 \times 10^{-5}$

Número de épocas: 20

### Compact Convolutional Transformers (CCT)

- Tamaño de entrada de la imagen: 224 x 224 píxeles
- Dimensión de los embeddings para cada token: 384,
- Número de capas convolucionales: 4,
- Tamaño del filtro de convolución: 7,
- Stride de la convolución: 2, define como se reducen las dimensiones de la imagen.
- Número de capas del transformer: 12
- Número de cabezas de atención en cada capa: 6
- Tasa de dropout: 0.3,
- Número de clases: 3

Se presenta un esquema de la estructura del modelo:



*Imagen 42: Estructura de la CCT aplicada*

Los pesos de clase fueron los mismos calculados para el modelo de ViT tradicional.

Optimizador: se utilizó el Adam (Adaptive Moment Estimation)

Taza de aprendizaje:  $1 \times 10^{-4}$

Decaimiento de pesos:  $1 \times 10^{-5}$

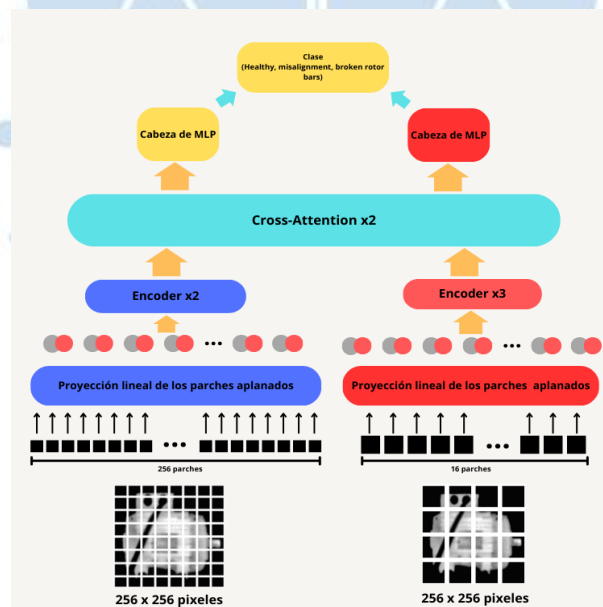


Número de épocas: 15

### CrossViT

- Tamaño de la imagen: 256 x 256
- Número de clases: 3
- Número de bloques de codificación multi-escala: 4.
- Dimensión de los parches de alta resolución: 16 x 16
- Numero de capas en el encoder de alta resolución: 2
- Cabezas de atención en la rama de alta resolución: 8
- Dimensión de los parches de baja resolución: 64 x 64
- Número de capas en el encoder de baja resolución: 3
- Número de cabezas de atención en la rama de baja resolución: 8
- Número de rondas en el bloque de atención cruzada: 2
- Número de cabezas de atención en el bloque de atención cruzada: 8
- Taza de dropout: 10%
- Taza de dropout en los embeddings: 10%

Se presenta un esquema de la estructura del modelo:



*Imagen 43: Estructura de la Cross ViT aplicada*

Los pesos de clase fueron los mismos calculados para el modelo de ViT tradicional.

Optimizador: se utilizó el Adam (Adaptive Moment Estimation)

Taza de aprendizaje:  $1 \times 10^{-5}$

Decaimiento de pesos:  $1 \times 10^{-5}$

Número de épocas: 15

#### 7.4. Datasets de entrenamiento y testeo

Culminada la etapa donde se ha detallado los tipos de métodos de clasificación probables a usar, debemos establecer los protocolos de testeo y entrenamiento, pues este es un paso de igual de importante que los antes repasados, sobre todo para corroborar errores de *overfitting*, de esta forma tendremos los siguientes Datasets de *training* y *testing*:

Dataset Train	Fuente	N° Datos	Clases
Train1	Data1	939 img (80%) + metadata (4GLCM x 939 img)	Healthy
	Coupling2	1480 img (80%) + metadata (4GLCM x 1480 img)	Missalingment1
Train2	Data1	939 img (80%) + 1878 img (Data augmentation) + metadata (4GLCM x 2817 img)	Healthy
	Coupling2	1480 img (80%) + 1480 img (Data augmentation) + metadata (4GLCM x 2960 img)	Missalingment1
Train3	Data1	1009 img (80%) + 4 sub-bandas por imagen (DWT)	Healthy
	Coupling2	1439 img (80%) + 4 sub-bandas por imagen (DWT)	Missalingment1
Train4	Data1	1511 img (80%)	Healthy
	Coupling1 y 2	3538 img (80%)	Missalingment1
Train5	Data1	1196 img (80%)	Healthy
	Coupling2 gris	720 img (80%)	Missalingment2

		888 img (80%)	Missalingment3
		942 img (80%)	Missalingment4
		148 img (80%)	RotorBar1
		363 img (80%)	RotorBar3
		390 img (80%)	RotorBar6
Train6	Data1+Data4	1782 img (80%) + 4 sub-bandas por imagen (DWT) + metadata (2GLCM x 1782 img)	Healthy
		3530 img (80%) + 4 sub-bandas por imagen (DWT) + metadata (2GLCM x 3530 img)	Missalingment
		901 img (80%) + 4 sub-bandas por imagen (DWT) + metadata (2GLCM x 901 img)	Rotor Bar

Tabla 8: Conjunto de Datasets de entrenamiento

Dataset test	Fuente	Nº Datos	Clases
Test1	Data1 Coupling2	322 img (20%) + metadata (4GLCM x 322 img)	Healthy
		319 img (20%) + metadata (4GLCM x 319 img)	Missalingment1
Test2	Data1 Coupling2	322 img (20%) + 644 (Data augmentation) + metadata (4GLCM x 966 img)	Healthy
		319 img (20%) + 638 (Data augmentation) + metadata (4GLCM x 957 img)	Missalingment1
Test3	Data1 Coupling2	252 img (20%) + 4 sub-bandas por imagen (DWT)	Healthy

		360 img (20%) + 4 sub-bandas por imagen (DWT)	Missalingment1
Test4	Data1 Coupling1 y 2	378 img (20%)	Healthy
		885 img (20%)	Missalingment1
Test5	Data1 Coupling2 gris	299 img (20%)	Healthy
		180 img (20%)	Missalingment2
		222 img (20%)	Missalingment3
		236 img (20%)	Missalingment4
		37 img (20%)	RotorBar1
		91 img (20%)	RotorBar3
		98 img (20%)	RotorBar6
Test6	Data1+Data4	445 img (20%) + 4 sub-bandas por imagen (DWT) + metadata (2GLCM x 445 img)	Healthy
		883 img (20%) + 4 sub-bandas por imagen (DWT) + metadata (2GLCM x 883 img)	Missalingment
		225 img (20%) + 4 sub-bandas por imagen (DWT) + metadata (2GLCM x 225 img)	Rotor Bar

Tabla 9: Conjunto de Datasets de testeo

Como se puede observar en la tabla 8 y 9, se ha respetado un criterio de separación, pues todos los *Datasets* de *train* y *test* siempre representan el 80 y 20% de la totalidad de datos por ingresar al modelo, sería interesante probar en algún futuro proyecto alguna otra distribución como 70-30% para observar el comportamiento de nuevos modelos, aunque tal y como se verá más adelante en el capítulo de discusión de resultado, no se espera una mejora significativa.



Habiendo repasado tanto las etapas de preprocesamiento, clasificación y entrenamiento podemos pasar entonces a detallar los modelos que se han ensayado y modelado , tendríamos entonces la siguiente tabla:

### 7.5. Modelos ensayados

Para el ensamblado de todos los modelos se ha usado un conjunto de técnicas de preprocesamiento, así como un método de clasificación, es importante resaltar que inclusive dentro de un mismo tipo de clasificador existen variaciones, es por eso que existe un índice que lo denote.

Modelo	Preprocesamiento	Clasificador	Datasets
Modelo 1	YOLOV5s+Otsu+FFT+CLAHE+GLCM	VGG16-1	Train-test1
Modelo 2	YOLOV5s+Otsu+FFT+CLAHE+GLCM+Data Augmentation	ResNet50	Train-test2
Modelo 3	YOLOV5s+Otsu+FFT+CLAHE+DWT	SVM-1	Train-test3
Modelo 4	YOLOV5s+Otsu+FFT+CLAHE	EfficientNetB2	Train-test4
Modelo 5		EfficientNetB5	
Modelo 6		EfficientNetB7	
Modelo 7		VGG16-2	
Modelo 8		InceptionV3-1	
Modelo 9		SVM-2	
Modelo 10		KNN	
Modelo 11		InceptionV3-2	
Modelo 12	YOLOV5+SIFT-BoVW	ERT	Train-test5
Modelo 13	YOLOV5m+Otsu+HSI+CLAHE+GLCM+DWT+PCA	ViT	Train-test6
Modelo 14		ViTFSD	Train-test6
Modelo 15		CCT	Train-test6
Modelo 16		CrossViT	Train-test6
Modelo 17	YOLOV5m	ViT	Train-test6

Tabla 10: Modelos evaluados

## 7.6. Resultados y criterios de evaluación

Tal y como se señaló en el apartado 3.2.2 respecto a objetivos específicos, para la labor de evaluación se hará uso de 3 valores referenciales, así como del uso de una matriz de confusión. Nos vemos en la necesidad entonces de explicar antes a que se vienen refiriendo dichos valores, pues cada uno de ellos brinda información valiosa respecto al desempeño del modelo.

Model	Precisión		Recall		F1-score	
	Healthy	Failed	Healthy	Failed	Healthy	Failed
Modelo 1	1	0.78	0.72	1	0.84	0.87
Modelo 2	0.98	0.79	0.74	0.99	0.85	0.88
Modelo 3	0.99	0.992	1	0.97	0.993	0.995
Modelo 4	1	0.78	0.84	1	0.91	0.88
Modelo 5	0.99	0.8	0.85	0.99	0.92	0.89
Modelo 6	1	0.9	0.89	0.89	0.91	0.9
Modelo 7	1	0.8	0.82	0.84	0.9	0.92
Modelo 8	0.98	1	1	0.99	0.99	0.99
Modelo 9	0.84	0.82	0.68	0.92	0.75	0.87
Modelo 10	0.88	0.87	0.78	0.93	0.83	0.9
Modelo 11	0.83	0.9	0.84	0.89	0.84	0.9
Modelo 12	1	0.998	0.95	0.996	0.97	0.997
Modelo 13	0.99	0.96	0.943	0.999	0.97	0.979
Modelo 14	0.995	0.997	0.995	1	0.995	0.998
Modelo 15	1	0.999	0.998	1	0.999	0.999
Modelo 16	0.997	0.998	0.997	1	0.997	0.999
Modelo 17	0.998	0.955	0.936	0.999	0.966	0.976

*Tabla 11: Resultados de los modelos ensayados*

**Accuracy o precisión:** Valor que mide cuántos de los ejemplos clasificados como una clase (por ejemplo, "Healthy" o "Misalignment") son realmente de esa clase.

$$\text{Precisión} = \frac{\text{Verdaderos positivos}}{\text{Verdaderos positivos} + \text{Falsos positivos}}$$

Por ejemplo, para el caso del modelo1 en la tabla 10 o 11, se puede entender que el 100% de imágenes clasificadas como “*Healthy*” realmente pertenecen a dicha clase y que por otro lado el 78% de imágenes etiquetadas como “*Missalingment*” pertenecen a la otra.

**Recall:** Parámetro que mide la capacidad del modelo para encontrar todos los ejemplos verdaderos de una clase.

$$Recall = \frac{Verdaderos\ positivos}{Verdaderos\ positivos + Falsos\ negativos}$$

Por ejemplo, para el caso del modelo1 en la tabla 11, se puede entender que solo se identificaron correctamente el 72% de imágenes “*Healthy*” dentro de toda la clase y que por otro lado el 100% de imágenes de la clase “*Missalingment*” fueron identificadas.

**F1-Score:** es la media armónica entre la precisión y el *recall*.

$$Recall = \frac{Precisión \times Recall}{Precisión + Recall}$$

**Matrices de confusión:** Medida grafica que representa los valores FP(falso positivo), FN(falso negativo), TP(Positivo verdadero) y TN(Verdadero negativo) por cada clase, dado que la gran mayoría de los modelos solo con 2 clases, el hecho de realizar una para cada uno de ellos no tiene sentido, en ese tipo de casos suministra más información respecto al desempeño del modelo los valores antes mencionados; pero tal y como se verá para el caso del modelo 12, puede existir una matriz de mayor dimensión que 2x2, a continuación, se presentan dicha matriz de confusión:

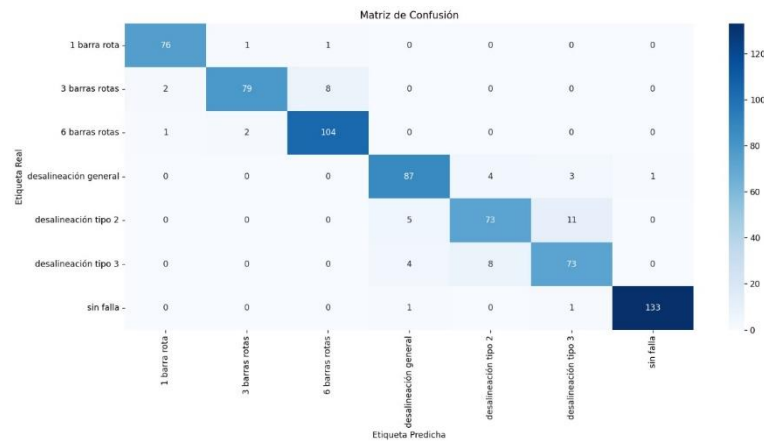


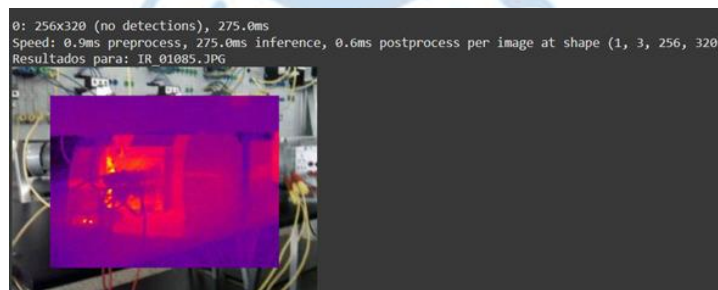
Imagen 44: Matriz de confusión del modelo 12

## 8. Análisis de resultados

En este capítulo repasaremos los resultados obtenidos, se comentará lo aprendido del uso de cada herramienta, cuales se han determinado como útiles y cuales no, asimismo se analizarán los resultados de los métodos clasificadores y su rendimiento en base a los criterios del capítulo previo, para finalmente establecer en base a lo antes comentado un modelo clasificador (Preprocesamiento + clasificador). Finalmente se comentará las pruebas de validación a la que se le someterá a este último modelo y cuáles son sus limitaciones.

### 8.1. Análisis preprocesamientos

Con respecto a los preprocesamientos de segmentación es interesante comentar que a pesar de usar YOLOV5 para la segmentación principal o identificación del ACM y Otsu o K-means para la segmentación secundaria o identificación de la región de interés (ROI) es posible usar YOLOV5 para ambas tareas o inclusive usar Otsu Thresholding como único método de segmentación para partir identificando un ROI directamente, sin embargo en las pruebas realizadas con estos métodos destaco en primer lugar el uso de un método de identificación primario, pues de esta forma se evitaba caer en el error de calcular un ROI sin haber confirmado antes que el ACM se encontraba en la imagen alcanzada, y de la misma forma se seleccionó Otsu puesto que es un modelo de acceso libre y de gran robustez, esto no quiere decir que no hubieron problemas con el uso de esta herramienta, pues particularmente con data4, es decir el grupo de imágenes obtenidas de múltiples instalaciones de la Universidad de Piura hubieron problemas con la identificación de los ACM, tal y como lo indica la imagen 41 que se muestra a continuación:

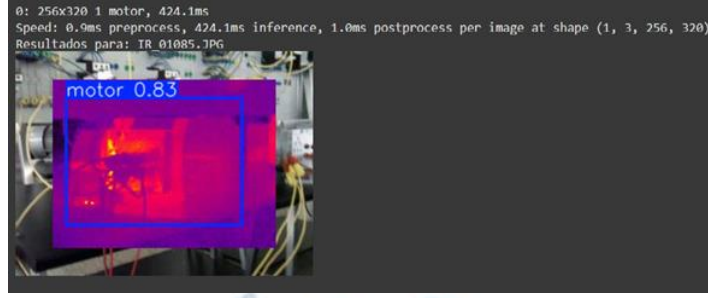


*Imagen 45: Detección fallida YOLOV5s*

Es por ello que se decidió migrar del modelo específico usado hasta ese entonces YOLOV5s, que se trata de la versión más pequeña de la 5ta generación de



este algoritmo de identificación, hacia la versión YOLOV5m, obteniendo muchos mejores resultados:



*Imagen 46: Detección correcta con YOLOV5m*

Prosiguiendo con los preprocesamientos de segmentación, evaluaremos el rol ahora de los métodos de segmentación secundaria, en el presente trabajo se han usado únicamente 2 métodos (Otsu y K-means), dándonos ambos muy buenos resultados para la tarea encomendada, sin embargo, terminamos por decidimos por el uso de Otsu en detrimento de K-means debido a su simplicidad operacional, que se detalla brevemente en las siguientes ecuaciones:

$$w_{1(T)} = \sum_{i=0}^T P(i), \quad w_{2(T)} = \sum_{i=T+1}^{L-1} P(i)$$

$$\mu_{1(T)} = \frac{\sum_{i=0}^T i \cdot P(i)}{w_{1(T)}}, \quad \mu_{2(T)} = \frac{\sum_{i=T+1}^{L-1} i \cdot P(i)}{w_{2(T)}}$$

Donde  $w_1$  representa la probabilidad de que un pixel pertenezca a la clase 1  $w_2$  de que lo haga a la clase 2, mientras que  $\mu_1$  es la media de la clase 1 habiendo establecido un umbral (T) de tal forma que los pixeles de intensidad inferior a dicho umbral sean clase 1 y los superiores clase 2, tras ello se calcula la varianza intra ( $\sigma_w^2$ ) e inter-clase ( $\sigma_b^2$ ):

$$\sigma_w^2 = w_1 \sigma_1^2 + w_2 \sigma_2^2, \quad \sigma_b^2 = w_1 w_2 \cdot (\mu_1 - \mu_2)^2$$

Para finalmente optimizarse de tal forma que se establezca un nuevo umbral T que maximice la varianza inter-clase y minimice la varianza intra-clase.

Por otro lado, el método K-means usa una función de optimización o también conocida como función de costo (J), la cual debe ser minimizada y se representa en la siguiente ecuación:

$$J = \sum_{j=1}^k \sum_{x_i \in C_j} \|x_i - \mu_j\|^2$$

Siendo  $k$  el número de clusters o agrupamientos,  $x_i$  el vector que indica la intensidad de cada pixel y  $\mu_j$  el centroide del  $j$ -ésimo cluster, que se calcula de la siguiente forma:

$$\mu_j = \frac{1}{|C_j|} \sum_{x_i \in C_j} x_i$$

Donde  $C_j$  es una agrupación de píxeles, en caso de que sean 2 clusters como en nuestro caso, solo existirán  $C_1$  y  $C_2$

Tras armar 2 clusters iniciales en base a puntos aleatorios, de tal forma que los demás puntos o píxeles de la imagen más cercanos a un cluster u otro le pertenezcan a dicho cluster. Posterior a ello, se calcula y recalcula la función de costo hasta alcanzar su valor mínimo de tal forma que, una vez encontrada la función mínima, se posee 2 clusters nuevos de tal forma que ahora los píxeles en efecto se guardan en el cluster cuya “distancia de color” es la mínima, lo cual quiere decir que se agrupa en 2 clases a las que cada grupo de pixel posee una relación, en este caso por intensidad y contraste.

Sin embargo, la herramienta K-means aparte de ser un proceso iterativo cuyo funcionamiento es mucho más complejo de entender que el de Otsu, constituye una herramienta cuyo objetivo no es reducido a la separación de 1 clase como si lo es Otsu, pues al contrario que este último, K-means permite la separación de una imagen en más de 2 clases, característica que en este caso no se requiere, pues solo se desea separar el ROI (ACM) del fondo, por lo que por todo lo antes repasado se considera que el uso de Otsu por encima de K-means es adecuado.

Prosiguiendo con el análisis de las herramientas de preprocesamiento, tenemos a los preprocesamientos para extracción de características, de los cuales destacan 2 grupos principales los cuales son extracción de características simples y extracción de características complejas, esto es apreciable en la siguiente tabla:

Métodos de preprocesamiento-Extracción de características	
Características simples	Características complejas
FOS	DWT
GLCM	CNN

*Tabla 12: Tipos de preprocesamiento para extracción de características*

Esta distinción viene debido a que tanto FOS como GLCM extraen características simples de la imagen, datos como: Media, varianza, asimetría (FOS) y Contraste, Energía, Homogeneidad (GLCM), mientras que, por otro lado, las herramientas DWT y CNN analizan y extraen características de capas más profundas, en el caso de las DWT analiza la imagen a diversos estados de frecuencia, haciendo una separación de patrones en base a esto, mientras que las CNN calculan y asignan pesos a características inobservables de las imágenes que han pasado por un proceso de convolución a fin de reducir la dimensionalidad de sus componentes.

Es por todo ello que existe una diferencia sustancial en ambos tipos de preprocesamiento, a pesar de pertenecer al mismo tipo de preprocesamiento, en base a esto y tras tomar en cuenta los resultados obtenidos de la tabla 11 es que se decidió que tan solo uno de cada tipo sea el que deba usarse para la elaboración de un modelo completo.

En cuanto al debate si FOS o GLCM, tras evaluar el funcionamiento de ambas técnicas y dado que las GLCM analizan parámetros similares a las FOS, pero tomando en cuenta la influencia de pixeles vecinos, se optó por esta última como método de preprocesamiento para extracción de características simples.

Mas complicado es discernir entre la DWT y la CNN, puesto que la dinámica de ambos procesos no son similares en absoluto, sin embargo observando el rendimiento de modelos como el 4-11 y comparando su rendimiento con respecto al modelo 3 que uso DWT se observó que la precisión no disminuyo, pero el factor determinante en este caso fue el tiempo de entrenamiento de cada modelos, pues los modelos que usaron CNN demoraron un promedio de 1-6 hrs en entrenar, mientras que el modelo 3 demoro tan solo 1hr, tomando en cuenta además que este último modelo uso una mayor cantidad de datos para entrenar, podemos establecer que el uso de DWT para extraer características es similar en eficiencia al uso de una CNN, pero le supera en términos de rendimiento, por lo cual para el modelo final, se decidió a DWT como método de preprocesamiento de extracción de características complejas.

Con respecto a los métodos para diferenciar aún más las imágenes entre clase se debe comentar que, CLAHE ha destacado como una gran herramienta para aumentar el contraste en las imágenes tal y como se observa en la imagen 19 del capítulo previo. Lo mismo no se puede comentar sin embargo del método de Data Augmentation, el cual fue pensado como un método que permita aumentar las diferencias observables por parte del modelo, esto es, mediante el aumento sintético de data, ya que si el modelo recibía una mayor cantidad de datos de una misma clase con ligeros cambios, podría aprender a discernir mucho mejor las características propias de esta clase, es justamente por el particular uso de esta herramienta que este tipo de herramientas/métodos se les conoce como diferenciadores/aumentadores de contraste en el presente documento.

A pesar de ello, la idea que se tenía del uso del Data Augmentation para mejorar el entrenamiento cayo en desuso pues no se considera una herramienta valiosa, ello es deducible al observar los resultados del modelo 2 con respecto al modelo 1, pues este último a pesar de no usar DA, consiguió mejores resultados (Tabla 11), además de representar un costo muy importante tanto en el tiempo de entrenamiento como en el peso del modelo.

Un método que cobro fuerza al momento de volver a revisar la bibliografía es el uso de HSI como método directo para acrecentar el contraste, ello es de manera previa a CLAHE, para que de esta manera no se pierda la dinámica original de la imagen.

Finalmente, con respecto a los métodos de preprocesamiento de filtrado es menester comentar que, el filtro FFT a pesar de que lleva a cabo una labor de filtrado, lo hace con el objetivo de acrecentar contraste en la imagen y por lo explicado antes, CLAHE ya realiza esta acción de una manera mucho más efectiva, por lo que sería redundante mantener estos 2 preprocesamientos y, por ende, solo se mantuvo CLAHE para futuros modelos.

Por otro lado, con respecto a PCA y LDA, en este caso ambos métodos filtran una cantidad finita de parámetros de forma que solo los parámetros más representativos de cada imagen pasan hacia la siguiente capa, y en esta labor ha destacado la PCA, la cual tiene un mismo rendimiento que LDA, pero es su simpleza de funcionamiento en conjunto con su compatibilidad con SVM la que la hizo determinante para ser usada como filtro final de características.



## 8.2. Análisis clasificadores

En cuanto a los métodos de clasificación evaluados se pueden agrupar en 3 principales grupos CNN, Algoritmos de aprendizaje automatizado y ViT.

Con respecto al primer grupo se evaluaron 6 arquitecturas diferentes y con un enfoque distinto, pues en algunos casos como el modelo 1 o el modelo 2 se usaron para entrenar y predecir, mientras que en los modelos 4,6,11, así como muchos otros modelos más se usó únicamente su capacidad de entrenar los pesos de las características, los cuales serán luego ingresados a un clasificador de otra categoría tales como un SVM y KNN.

Específicamente en el uso de las CNN como clasificadores se ha concluido que la eficiencia de estas herramientas demuestra que es una opción válida a usar para solucionar el problema planteado, sin embargo, se ha detectado que su eficiencia es posible mejorar al menos un poco si es que se reemplaza la última capa decoder por un SVM, que es una herramienta cuya superioridad en dicha tarea ha sido demostrada en (Agarap, 2017) donde comparo ambas metodologías, particularmente en este trabajo no se ha detectado una diferencia tan apreciable entre ambos modelos, sin embargo dado que según la bibliografía revisada, si existió dicha mejora se decidió continuar con un modelo cuyo clasificador final sea un bloque SVM.

Con respecto al uso de los modelos basados en algoritmos de aprendizaje automático tenemos a: SVM, KNN y ERT, dado que ya comentamos los resultados de SVM, nos centraremos en los otros 2 modelos, no sin antes resaltar que SVM es un algoritmo independiente de las CNN, es decir puede ser usado como un método de clasificación por sí mismo, como lo demuestran los modelos del 3 al 9, de los cuales es importante resaltar el particular buen resultado obtenido del modelo 3, que destaca por ser a la vez que liviano, de rápido entrenamiento y de buena precisión.

Mientras que, por otro lado, el modelo 10 que usa KNN y FOS muestra un declive apreciable en la precisión con respecto a sus coetáneos, esto sin embargo es en gran parte debido a la pérdida de información resultado de aplicar un FOS, puesto que a diferencia de los otros modelos que usaban miles de características por imagen, el modelo 10 tan solo uso 6 por imagen, es entendible entonces el descenso negativo percibido, sin embargo a pesar de ser un resultado comparativamente negativo, destaca por ser un modelo super liviano y sencillo que posee un porcentaje de



precisión bastante aceptable, por lo que su uso en algún aspecto de la industria no debería ser descartado.

De forma similar se estructuró el modelo 11, que combina en esta ocasión el método de extracción de características de una CNN en este caso del tipo Inception V3 para luego someterla a un filtrado mediante LDA e ingresarlo al método clasificador KNN, dándonos con la sorpresa que también tuvo un resultado negativo en comparación con los otros modelos ensayados, optando por ello descartar el uso de KNN como método clasificador.

Además, continuando con el método de BoVW se exploró el uso de ERT para predecir tomando como inputs los histogramas generados en el método antes especificado, el resultado obtenido es sin lugar a duda prometedor, aunque el peso del modelo es un aspecto que detrimenta un poco al modelo, pues a pesar de no ser de la escala de las CNN, aun es un modelo mucho más pesado que otros modelos de mejor o similar precisión.

Finalmente, durante una segunda revisión del estado del arte surgió una herramienta novedosa que no se había tomado en cuenta dado su prácticamente nulo uso en el presente campo, los modelos Transformers o Attention Models son RNN que poseen la capacidad de aprender inter relaciones entre datos ingresados y son la base de los modelos de generación de lenguaje más avanzados, pero que sorprendentemente pueden ser aplicados a un campo como el de computing vision.

En cuanto a este método y su uso es relevante mencionar que se implementaron 4 modelos (13,14,15 y 16) que se centran en solucionar el principal problema entre modelos de este tipo de arquitectura y datasets pequeños, tomando en cuenta que normalmente los Transformers se entrenan con millones de datos y en nuestro caso no se alcanzan ni las 2 decenas de miles.

Los resultados de dichas pruebas fueron que todos los modelos incluyendo el modelo básico ViT dieron resultados de predicción casi perfecto, y más importante aún, demoraron en entrenar menos que las CNN, por lo que en definitiva los ViT son una mejor alternativa que una CNN, aunque asimismo es importante destacar que a medida que el número de datos disminuyen, el rendimiento entre ambas opciones será muy parecido.

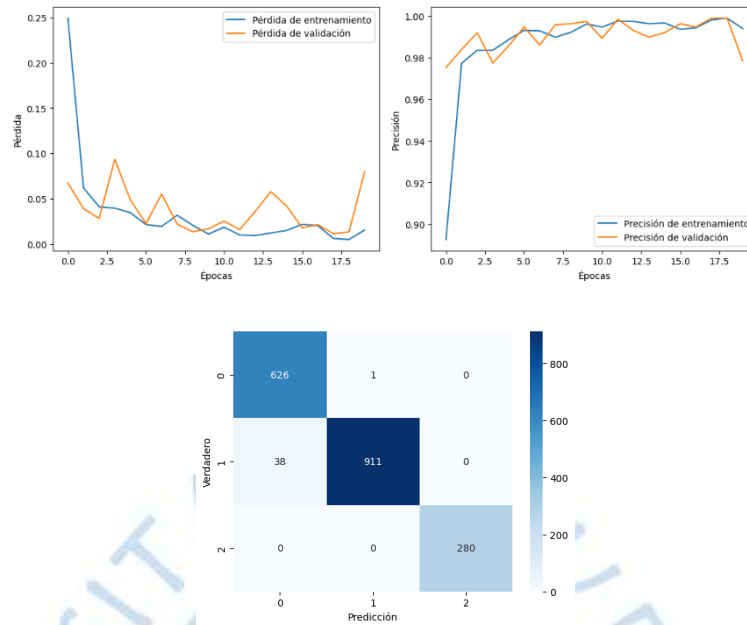


Imagen 47: Curvas de aprendizaje y matriz de confusión simpleViT

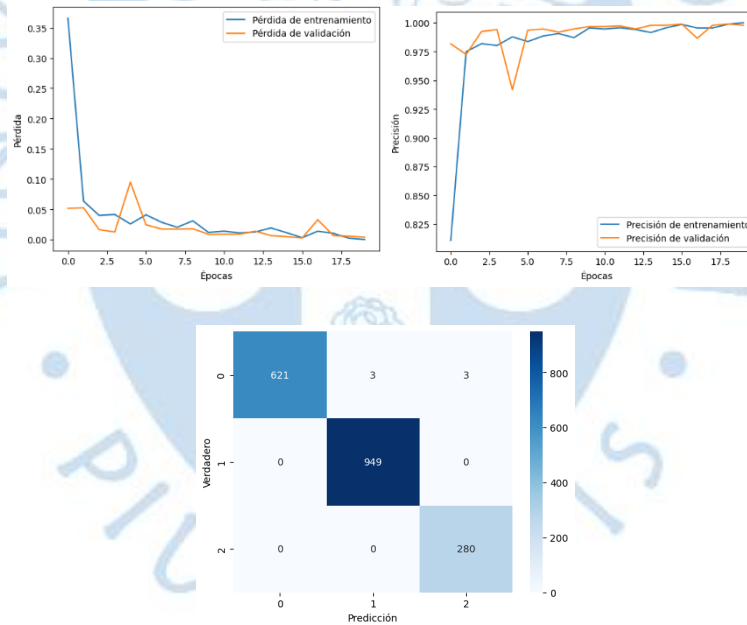


Imagen 48: : Curvas de aprendizaje y matriz de confusión ViTFSD

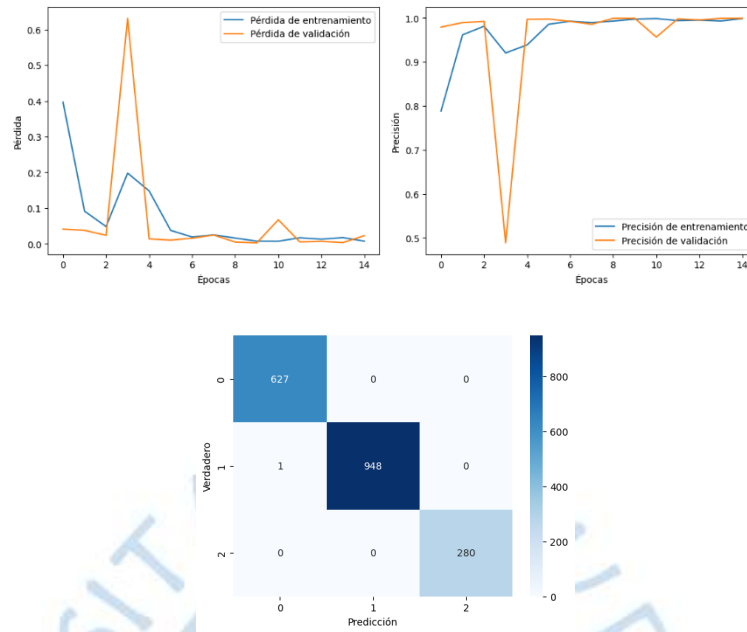


Imagen 49: Curvas de aprendizaje y matriz de confusión CCT

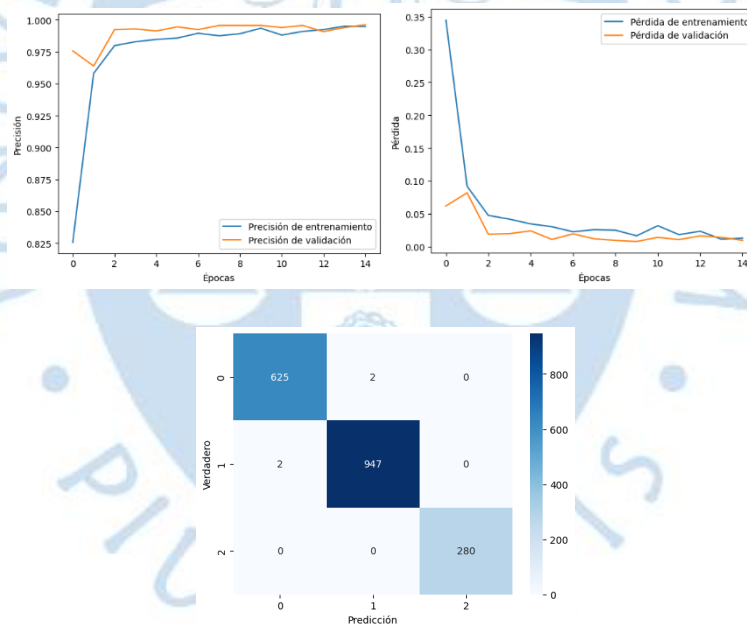


Imagen 50: Curvas de aprendizaje y matriz de confusión CrossViT

Algo mucho más importante a resaltar fue el modelo 17, cuyos datos de entrada ni siquiera fueron tratados y aun así el resultado de predicción fue elevado (Tabla 11), lo que nos demuestra que no solo las CNN son capaces de hallar relaciones entre imágenes a un nivel que a la vista humana sería imposible, sino que también las ViT lo hacen y de manera más rápida y eficiente.

Aunque los modelos por ViT pueden parecer que son los mejores, son modelos que, si bien es cierto, entrenaron más rápido que los modelos basados en CNN, no son de rápido entrenamiento como si lo son los modelos 3, 9 y 10.

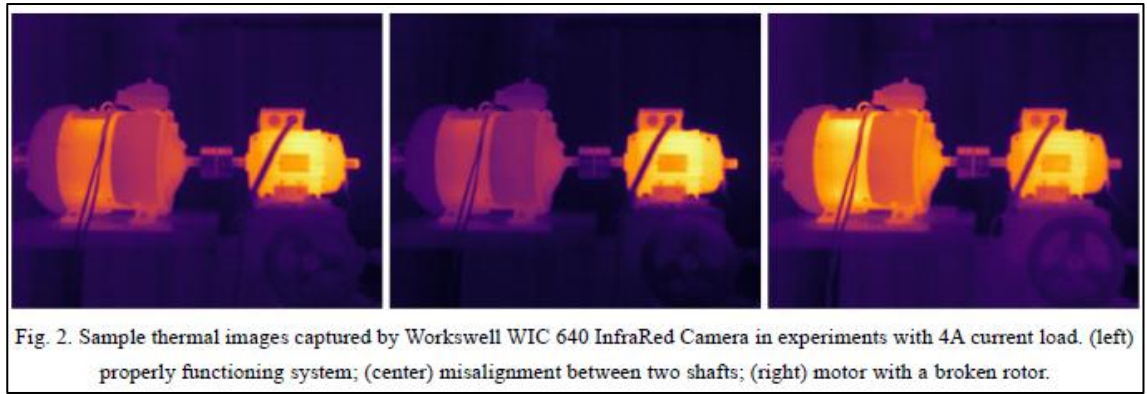
Por lo que tras revisar todos los resultados se determinó que los únicos modelos interesantes por mejorar son el modelo 3 (Basado en SVM) y los ViT, pero dadas las restricciones existentes especificadas en el capítulo 3.3.3 se optara por desarrollar el modelo 3 con las nuevas tecnologías descubiertas durante el desarrollo de este trabajo, a este modelo se le conocer como modelo 18 o mejor modelo.

### 8.3. Análisis datasets

En cuanto a los datasets es relevante analizar las 2 principales fuente de datos de las cuales se nutre este trabajo, en primer lugar, tenemos a la data obtenida de (Piechocki, Pajchrowski, Kraft, Wolkiewicz, & Pawel, 2023), donde a pesar de contar con una gran cantidad de imágenes y un número considerable de clases, se observa que el tipo de maquina objeto de estudio es la misma.

Parameter	Value
Producer	Celma Indukta
Type	Sh 90L-4
Phase	3-phase
Rated power	1.5 kW
Input current	6.1 / 3.5 A
Rated voltage	230 / 400 V
Rated frequency	50 Hz
Efficiency	79%
Speed	1410 RPM
Shaft diameter	24 mm

*Tabla 13: Configuración del motor AC en análisis  
(Fuente: (Piechocki, Pajchrowski, Kraft, Wolkiewicz, & Pawel, 2023))*



*Imagen 51: Imagen comparativa de ACM para distintas fallas  
(Fuente: (Piechocki, Pajchrowski, Kraft, Wolkiewicz, & Pawel, 2023) )*

Esto explicaría el motivo de porque la mayoría de los modelos entrenados, incluyendo los detallados en el presente trabajo dan tan buenos resultados, pues los patrones de falla al pertenecer a una misma máquina son ampliamente perceptibles para un modelo clasificador como los evaluados, por ejemplo de la imagen 51 es perceptible una clara diferenciación entre grados de temperatura, operativa tanto en la carga como en el ACM, y si bien es cierto, dicha dinámica no necesariamente se repite para cada clase e imagen, el modelo 9 ya ha demostrado que un modelo simple y liviano puede detectar patrones en esta data de manera sencilla.

Esto sin embargo no es extrapolable a industria, pues en dicho contexto, no existe un solo tipo de motor por analizar y menos aún en una misma sala o con un mismo fondo y/o carga de funcionamiento, por lo cual los resultados observados si bien son altamente prometedores, deberían ser tomados como una oportunidad de extrapolar este tipo de investigaciones y promover la obtención de data independiente.

Concatenando lo anteriormente mencionado respecto a la problemática de la data, es que surgió la idea de obtener un nuevo dataset en base a motores de diversa índole, uso, carga, contexto de funcionamiento, etc. Es por ello que, en coordinación con múltiples entes institucionales de la Universidad de Piura, se coordinó la visita a múltiples lugares y áreas de funcionamiento donde se podrían encontrar este tipo de máquinas.

Para configurar la cámara se deben hacer un par de ajustes, como mover el enfoque del lente para una imagen más nítida, quitar los recuadros que aparecen con bordes o indicando los valores de temperatura máxima, colocar la emisividad



correctamente, etc. Todas estas modificaciones sirven para usar la cámara de manera correcta, en este caso se hizo uso del manual donde se explica cómo usar la cámara.

Tal y como se detalló en el capítulo 6, se tomaron fotos de múltiples ambientes, cada uno con sus propias particularidades y problemas por resolver, es por ello que se debe hacer los siguientes comentarios respecto a dichos procesos de obtención de data:

### **8.3.1. Laboratorio de Sistemas Automáticos de control (Lab SAC)- Ubicación 1**

El motor era el adecuado para colocar las imágenes tomadas al modelo ya entrenado y verificar si se trata de un motor sano, para ser más precisos, categoría “healthy”, pues es un motor que no presenta problemas, podría considerarse que tiene un pequeño desalineamiento, pero esto no genera cambios de temperatura tan altos como para que se detecte como una falla, tal vez a largo plazo y si se usa el motor con mayor frecuencia podría ser que ocurra una falla, pero este no es el caso por ahora.

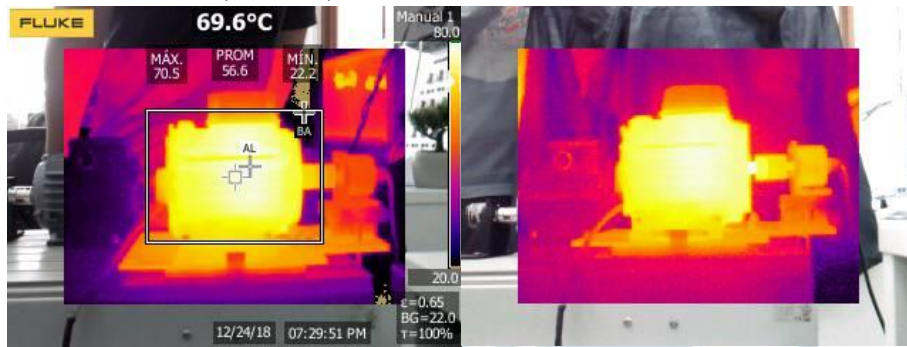


*Imagen 52: Motor ACM-LabSAC*

Como se observa en la imagen 52, se usó una fondo color oscuro, favoreciendo a las imágenes, en realidad la cámara tiene una configuración de enfoque que puede reducir la influencia de un fondo no uniforme, sin embargo y dado que se contaba con un fondo disponible, se decidió usar este último para lograr captar mejor la imagen.

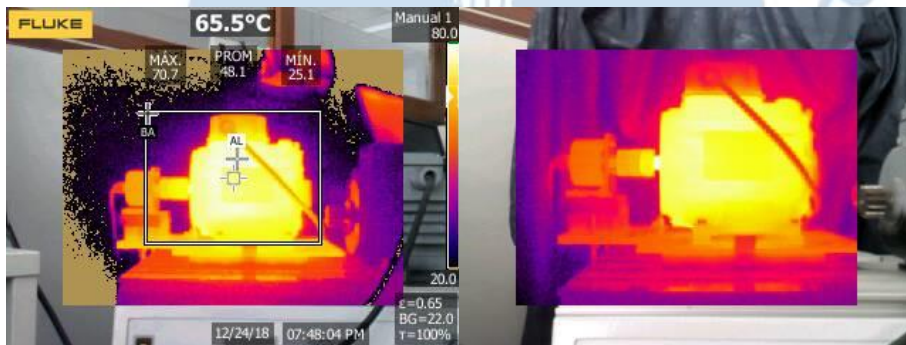
En la imagen 53 se observa claramente cómo es que el uso de este fondo dio mejores resultados. Al momento de querer usar las imágenes registradas y usar el modelo, se percibió que las imágenes aun conservaban la interfaz, por lo que se tuvo que modificar la configuración de la cámara y volver a registrar fotos, obteniéndose así 144 imágenes de este motor, en distintos ángulos.

- Vista lateral (derecha) del motor:



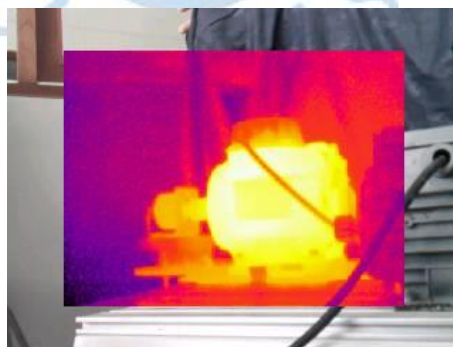
*Imagen 53: Comparativa uso y no uso del fondo negro, vista lateral derecha del motor*

- Vista lateral (izquierda) del motor:



*Imagen 54: Vista lateral izquierda del motor*

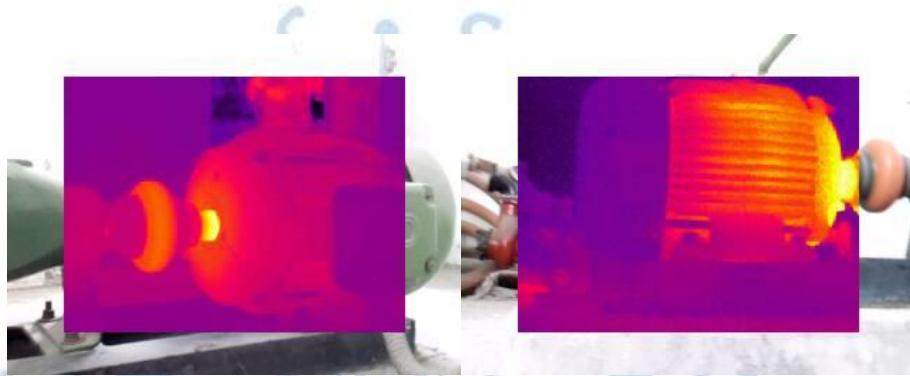
- Vista oblicua del motor:



*Imagen 55: Vista oblicua del motor*

### 8.3.2. Estación de bombeo de agua de la Facultad de Comunicación- Ubicación 2

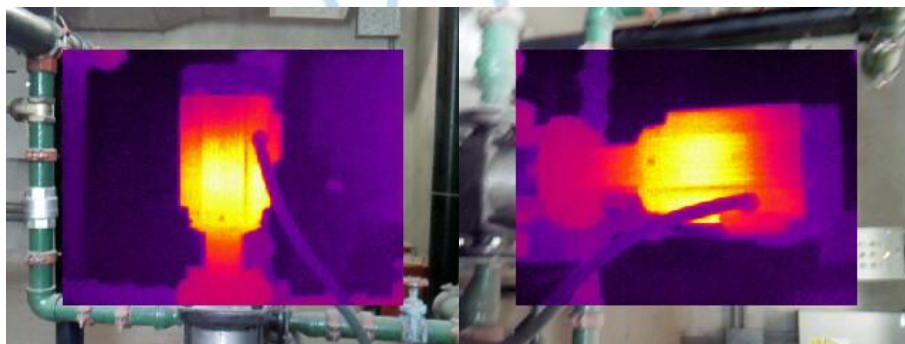
Se tomaron 25 imágenes en la estación de bombeo. Como se aprecia en las imágenes se optó por variar la posición de la cámara para poner a prueba el modelo de identificación del modelo y que el modelo de detección de fallas esté entrenado bajo múltiples condiciones. Además, las zonas de calor se concentran en la zona del eje con el acople del impulsor de la bomba lo cual puede relacionar fallas con el rotor desalineado y generar confusión en el modelo a la hora de clasificación.



*Imagen 56: Imágenes Ubicación 2*

### 8.3.3. Estación de bombeo de agua para el Aulario UDEP/ Edificio E- Ubicación 3

Estos motores debido al tipo de instalación están ubicados de manera vertical a diferencia del resto de ubicaciones visitadas tomándose alrededor de 63 imágenes. Se observa una mayor distribución de la intensidad calorífica a diferencia de la anterior estación siendo similar al resto de imágenes del motor operando a condiciones nominales. Nuevamente la detección y clasificación de los motores resulta ser un nuevo reto para los modelos creados debido a la ubicación de estos y el hecho de que en una sola imagen pueden aparecer más motores funcionando.

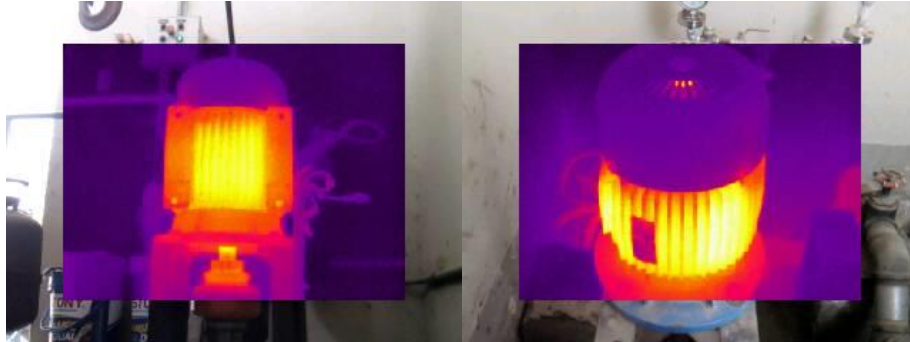


*Imagen 57: Imágenes Ubicación 3*



#### 8.3.4. Estación de bombeo de aguas residuales del edificio 80-Ubicación 4

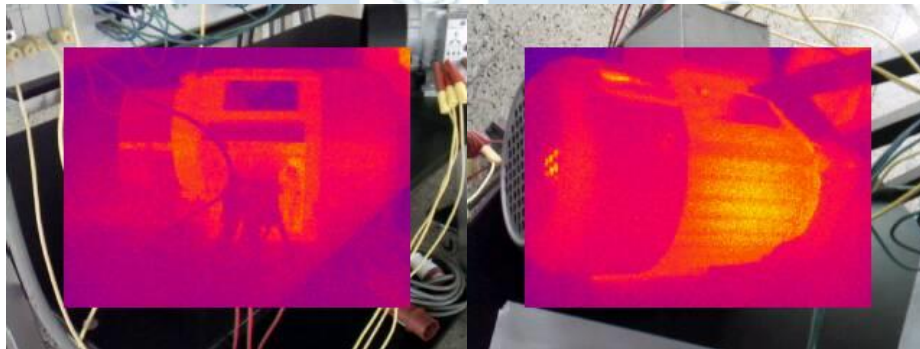
Se tomaron 80 imágenes en esta ubicación, el banco de bombas está distribuido de igual manera que la estación de bombeo anterior con la particularidad de que se aprecian zonas de calor mucho más intensas en el eje del motor debido a la carga dado a la demanda de potencia que tienen las aguas residuales a comparación del agua potable por la diferencia de densidad de los fluidos.



*Imagen 58: Imágenes Ubicación 4*

#### 8.3.5. Laboratorio de Electrotecnia-Ubicación 5

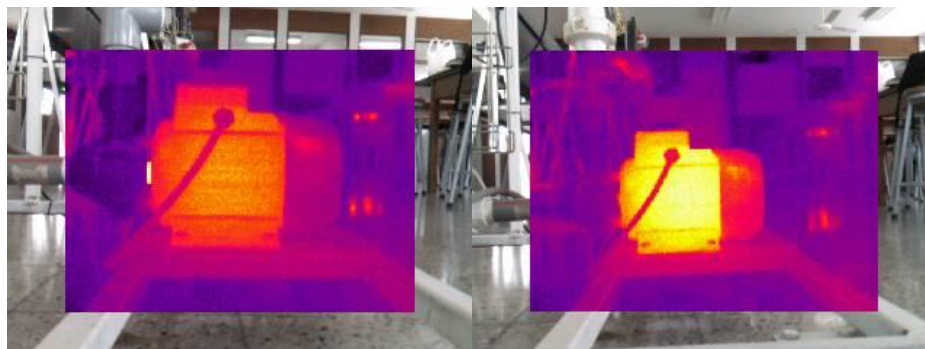
En este ambiente se encuentran 8 motores trifásicos en vacío con fines didácticos para los cursos de ingeniería y las bombas eléctricas pertenecientes al área de energía de los cuales se tomaron 167 imágenes: 77 para un motor trifásico y 90 para la bomba perteneciente a un módulo didáctico.



*Imagen 59: Imágenes Motor Ubicación 5*

Las imágenes capturadas para el motor trifásico fueron complicadas de observar un incremento en la temperatura de estos debido a la gran disipación de calor provenientes del ventilador integrado y las aberturas del motor que regulan adecuadamente la temperatura del motor. Al no haber un contraste en la intensidad de píxeles del fondo y la máquina, resultará más compleja la identificación del

motor mediante el Yolov5 que había sido entrenado en pruebas anteriores lo que puede resultar en un reentrenamiento o inclusive en la selección de un modelo más sofisticado.



*Imagen 60: Imágenes Ubicación 5-bomba eléctrica*

El módulo de control de nivel de tanques del área de energía contiene una bomba eléctrica de corriente alterna cuya potencia es controlada por el variador de frecuencia en el tablero eléctrico del sistema. En la ilustración “” se observa el funcionamiento a un valor de frecuencia baja y en la “” en un valor alto mostrando un gran contraste en la corriente de operación de este entrando en las clases de funcionamiento del motor del Dataset inicial.

Finalmente se puede resumir la estructura de esta data recopilada mediante las siguientes tablas:

N° de sub Dataset	Ubicación	N° de Imágenes	Observación
1	Laboratorio de Sistemas Automáticos de Control	144	El motor llego a una elevada temperatura de manera rápida
2	Estación de bombeo de agua para la Facultad de Comunicación	25	Se observa una concentración de temperatura en el eje del motor
3	Estación de bombeo de agua para el Aulario UDEP	63	Motores colocados de manera vertical
4	Estación de bombeo de residuos del Edificio 80	80	Mayor calentamiento debido a la mayor demanda de potencia de la carga
5	Laboratorio de Electrotecnia	167	El motor en vacío presentaba fugas de ventilación y el motor del módulo de tanques fue variando su potencia

*Tabla 14: Estructura Data4*



Ubicación del motor	Marca	Potencia [Hp]	Voltaje [V]	RPM	Frecuencia [Hz]
1	SIEBER	0.5	400	2780	50
2	DELCROSA	5.96	220/380	3600	60
3	-	3.8	220/380	3511	60
4	DELCROSA	4.8	220/380	1740	60
5	MASUKI	2	220/380	1650	60
	EBBARA	1	220/380	3350	60

Tabla 15: Tipos de ACM evaluados de la Data4

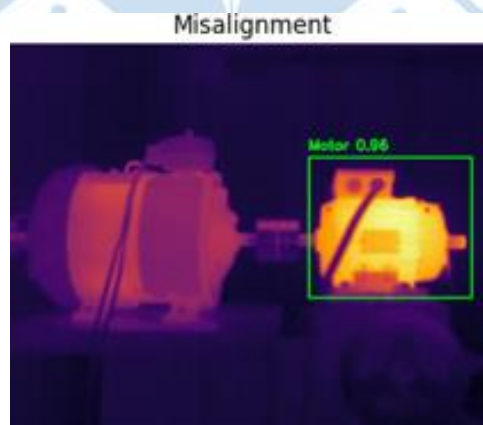
#### 8.4. Mejor modelo (Modelo 18)

Como se ha venido detallando en capítulos previos, el propósito de este trabajo no es proponer un modelo en específico como tal, sino más bien realizar una revisión sistemática de las herramientas que actualmente existe en este ámbito, sin embargo, las comparaciones son inevitables y en este contexto, es necesario quedarnos con un modelo para el establecimiento final de una propuesta para la solución de nuestra problemática.

Habiendo explicado esto, se procederá a detallar la configuración específica de dicho modelo, así como el funcionamiento de las herramientas que lo componen:

##### YOLOV5m

Modelo de peso mediano de la 5ta versión del modelo desarrollado por Ultralytics, en esta etapa se identifica si existe o no un motor en la imagen, esto es apreciable en la imagen 15:



### Otsu Thresholding

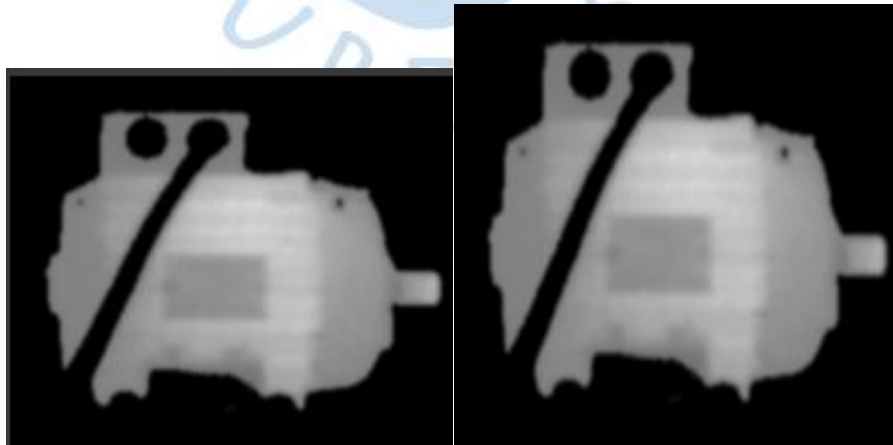
Con las imágenes del motor detectado y recortado, será segmentado mediante Otsu ya que a pesar de que la imagen ha sido reducida a un área de interés que aún contiene fondo que agrega información o características irrelevantes para la detección. Otsu halla un umbral donde la covarianza entre la intensidad de píxeles de la imagen por debajo y por encima del umbral sea el mínimo posible, lo cual sirve para identificar regiones de interés, por eso se cambió a la escala de grises para una mejor identificación.



*Imagen 61: Segmentación Otsu-modelo final*

### HSI

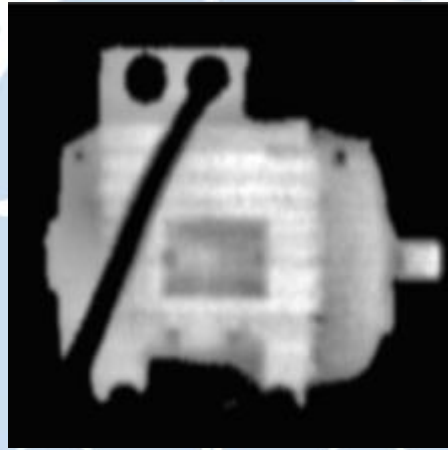
Las imágenes ya segmentadas por Otsu siguen aún en BGR el cual es el formato de la cámara, se cambia a RGB y son divididas por 255 con el fin de normalizar el valor de píxeles al rango de 0 a 1 y se calcula la intensidad de estos como la media de los 3 canales (HSI).



*Imagen 62: Normalización y redimensionamiento-modelo final*

### CLAHE

Se emplea CLAHE con el fin de mejorar el contraste de una imagen dividiéndola en pequeñas regiones de 8x8 y aplicando una ecualización local de histograma, lo cual se traduce en redistribuir las intensidades para que el histograma sea más uniforme. La división realizada es establecida por el límite de corte seleccionado, en este caso se estableció un valor de 3 controlando cuan agresivo se mejora el contraste local, ya que con 2 no se acrecentaban mucho las diferencias. Antes de CLAHE, se normaliza nuevamente la imagen con el fin de que los valores de intensidad de pixeles estén en el rango de 0 a 255 y al tipo uint8(formato) necesario para la aplicación del filtro.



*Imagen 63: Ecualización-modelo final*

### GLCM

Ahora se extraen las características del Dataset empleando la matriz de co-ocurrencia de nivel de gris (GLCM) simétrica y normalizada con una distancia de 1 píxel entre los pixeles a considerar devolviendo el contraste (variación de intensidad entre píxel y sus vecinos) y la homogeneidad (proximidad de distribución en la GLCM a la diagonal principal). El GLCM usa más características como la Dissimilarity, energía y entropía, pero estas características no serán usadas por temas de eficiencia computacional, además investigando se encontró que usar muchas características podría generar un sobreajuste, tal y como se aprecia en la imagen 23:

GLCM Properties current-load-4A				
Contrast: [[	309.87	550.58	275.69	340.75]
[	699.56	550.58	639.08	340.75]
[	1071.7	1190.6	1045.4	765.06]]
Correlation: [[	0.97174	0.94967	0.97477	0.96885]
[	0.93627	0.94967	0.94138	0.96885]
[	0.90249	0.89105	0.90388	0.92999]]
Energy: [[	0.034106	0.027298	0.030457	0.027841]
[	0.025862	0.027298	0.022853	0.027841]
[	0.022017	0.020457	0.019395	0.021023]]
Homogeneity: [[	0.27917	0.17759	0.21708	0.18844]
[	0.16852	0.17759	0.12844	0.18844]
[	0.12646	0.1016	0.093506	0.11422]]

## DWT

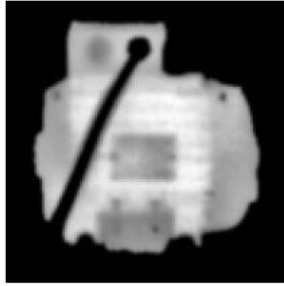
Una vez extraídas las características se emplea la transformada discreta de Wavelet (DWT) a las imágenes para trabajar con las 4 sub-bandas aproximadas. En primer lugar, cA siendo la sub-banda de baja frecuencia representando una versión simplificada de la imagen original manteniendo sus características generales y la mayor parte de la información de energía. Por otro lado, cH representa las frecuencias altas en la dirección horizontal resaltando transiciones o cambios rápidos en la intensidad de la imagen a lo largo de los bordes o contornos horizontales. El cV representa las frecuencias altas en la dirección vertical, es decir, los contrastes bruscos en los contornos verticales. cD contiene las frecuencias en ambas direcciones mostrando principalmente los cambios de intensidad en las diagonales resaltando bordes y contornos. Toda esta información de las sub-bandas será de suma importancia para realizar un mayor contraste entre las características que puede obtener cada imagen favoreciendo a un clasificador, mejorando su entrenamiento.

Las 4 sub-bandas extraídas por la DWT son aplanadas, es decir, se convierten en un vector de una dimensión y se unen en un solo vector representando de forma comprimida y capturando las diferentes frecuencias y sus detalles. Este proceso se realiza dado a que la mayoría de los algoritmos de clasificación esperan que los datos de entrada estén en forma de vectores en lugar de matrices (2D). Luego, se almacenarán las características combinadas (sub-bandas y GLCM normalizadas) en una matriz “x” y en otra llamada “y” las clases:

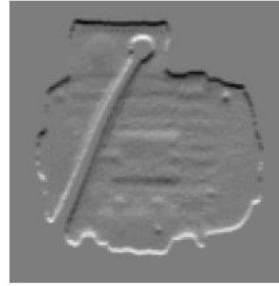
- 0 para “healthy”,
- 1 para “Misalignment”

- 2 para “broken rotor bars”

Subbanda cA:



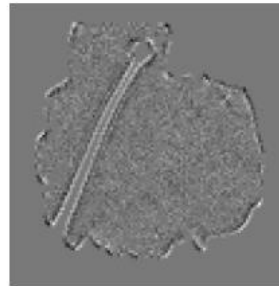
Subbanda cH:



Subbanda cV:



Subbanda cD:



*Imagen 64: DWT- modelo final*

Estas matrices son convertidas de arrays a Numpy con la finalidad de favorecer su eficiencia y facilidad en el uso en algoritmos de aprendizaje automático, ya que Numpy es una biblioteca especialmente para matrices.

Se arma la matriz X con todas las características de las 9071 imágenes y se arma la matriz Y que tiene las etiquetas, las matrices se arman con las 4 sub-bandas aplanadas obteniendo 50176 + 2GLCM 50178. Y son guardadas en la nube para evitar realizar este proceso reiteradamente.

Matrices X y y cargadas con éxito.

Forma de la matriz X: (9071, 50178)

Forma de la matriz y: (9071,)

[+ Code](#)

```
X_main = X[:, :50176]
X_last = X[:, 50176:]
print(X_main.shape)
print(X_last.shape)
```

(9071, 50176)

(9071, 2)

*Imagen 65: Guardado de imágenes y metadata*

Como ya se comentó en la matriz X, se almacenan los primeros 50176 componentes correspondientes a las sub-bandas en una matriz llamada “ $X_{main}$ ” y



otra con el contraste y la homogeneidad llamada “ $X_{last}$ ”. La primera matriz se normaliza en un rango de media 0 y desviación estándar 1 siendo necesario en la entrada del clasificador SVM para que todas las características tengan la misma escala/importancia y optimizar su rendimiento.

### PCA

Se utiliza PCA (Análisis de componentes principales) para reducir la dimensión de  $X_{main}$  a 50 componentes principales manteniendo la mayor parte de la varianza de sus datos minimizando problemas como overfitting y el tiempo de entrenamiento, recordando que las otras 2 son las del GLCM. Seguido se concatena nuevamente la matriz reducida con la información proveniente de las sub-bandas con la de GLCM en una sola con el propósito de usar la información más relevante en el entrenamiento. Finalmente se dividen los datos en conjuntos de entrenamiento y prueba.

Estos nuevos datos de “x” son divididos en 80% para entrenamiento y 20% en validación para luego balancear las clases, esto se debe por la alta cantidad de imágenes de motores sin fallas en proporción de los motores con fallas mecánicas para evitar el overfitting en el entrenamiento. Y se calculan los pesos de las 3 clases para equilibrar el conjunto de datos. Para una correcta parametrización con el objetivo de optimizar el entrenamiento y obtener los mejores resultados se realiza una búsqueda de los mejores hiperparámetros con la función “Grid Search” variando los valores de “C”, “gamma” y “Kernel”, por ejemplo, se colocan valores de C y la función “Grid Search” nos otorgó el mejor valor de C, lo mismo se hace con gamma, en el caso de Kernel, de los tres tipos: rbf, poly y linear, según la función “Grid Search” el mejor para usar es linear. Se pre-entrena el modelo con estos nuevos parámetros. Con los mejores parámetros obtenidos con “Grid Search” y el Dataset dividido, se procede a realizar el entrenamiento.

```
# Dividir los datos en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X_combined, y, test_size=0.2, random_state=42, stratify=y)

print(len(X_train))
print(len(X_test))
print(X_train.shape)
print(X_test.shape)

7256
1815
(7256, 52)
(1815, 52)
```

*Imagen 66: Configuración  $X_{train}$  y  $X_{test}$*

### 8.4.1. Resultados entrenamiento

Terminado este, se evalúa obteniéndose una precisión de 99.72%, el margen de separación que es la distancia entre los hiperplanos separadores de las clases de 5.669. Por último, se obtienen las siguientes métricas y la matriz de confusión mostrando una gran eficacia en la clasificación.

Precisión del SVM con PCA parcial y C=1: 99.72%  
Margen de separación: 5.669349086528589

*Imagen 67: Resultados training- Modelo 18*

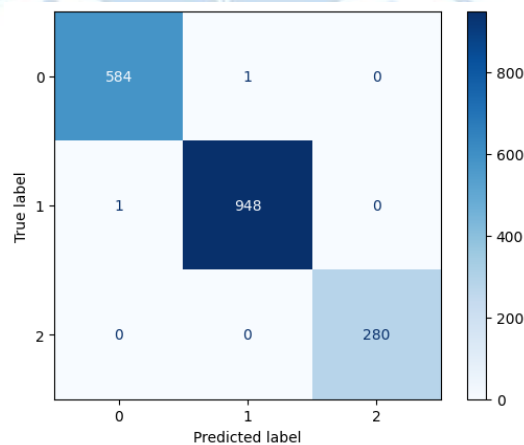
### 8.4.2. Resultados prueba con data de testeo

Para todas las clases se aprecian resultados favorables en la precisión y el recall, lo que puede deberse a similitud en las imágenes, pero con los modelos antes realizados cuando se tenía sólo una data se consiguieron valores muy similares, ahora con la data agregada se sigue obteniendo buenos resultados, lo cual corrobora la robustez del modelo.

	precision	recall	f1-score	support
Clase 0	0.99	0.99	0.99	586
Clase 1	1.00	1.00	1.00	949
Clase 2	1.00	1.00	1.00	280
accuracy			1.00	1815
macro avg	1.00	1.00	1.00	1815
weighted avg	1.00	1.00	1.00	1815

*Imagen 68: Resultados Test-Modelo 18*

- Matriz de confusión



*Imagen 69: Matriz de confusión Modelo 18-etapa 1*

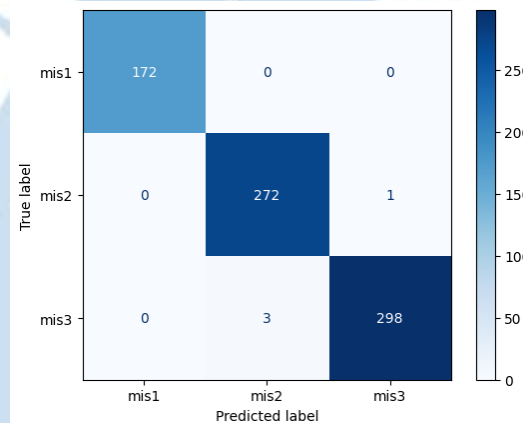
En la matriz de confusión se observa que no hay problemas con las clases para las predicciones, salvo un valor en la clase 0(healthy), y en la clase 1(Misalignment).

- Resultados con 3 clases más para Misalignment y broken rotor bars

Misalignment:

- Desalineamiento grado 1 (mis1)
- Desalineamiento grado 2 (mis2)
- Desalineamiento grado 3 (mis3)

El procedimiento para obtener estos resultados es el mismo, a continuación, se mostrará la matriz de confusión y se comentará:



*Imagen 70: Matriz de confusión Modelo 18-etapa 2-MSL*

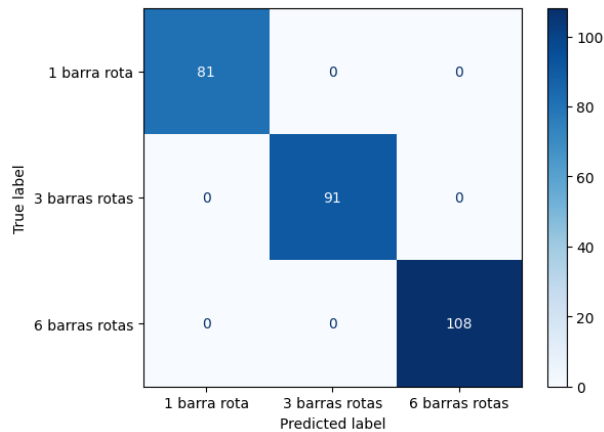
- Para la clase mis1 tiene 172 predicciones correctas con 0 errores
- Para la clase mis2 tiene 272 predicciones correctas con 0 errores
- Para la clase mis3 tiene 298 predicciones correctas con 3 errores

A pesar de tener poca data el modelo ha logrado diferenciar los tipos de desalineamiento sin errores salvo el desalineamiento de grado 3 donde hubo 3 errores.

Broken rotor bars:

- 1 barra rota del rotor
- 2 barras rotas del rotor
- 3 barras rotas del rotor

Matriz de confusión:



*Imagen 71: Matriz de confusión Modelo 18-eta 2-BRB*

- Para la clase 1 barra rota se han obtenido 81 predicciones correctas con 0 errores.
- Para la clase 2 barras rotas se han obtenido 91 predicciones correctas con 0 errores.
- Para la clase 6 barras rotas se han obtenido 108 predicciones correctas con 0 errores.

#### 8.4.3. Prueba de validación

Como se ha venido detallando en capítulos previos, se sospecha que los modelos pueden estar sufriendo de Overfitting, motivo por el cual se ha decidido realizar una prueba donde un dataset de validación compuesto únicamente por motores en un buen estado (Healthy) pero de diverso tamaño, posicionamiento, potencia, ubicación, carga, etc. Es decir, Data4, sirva para probar la robustez de este modelo, ello es antes de usarlas como data de entrenamiento, dicha prueba se llevó a cabo teniendo únicamente el YOLOV5m entrenado con dichas imágenes, a fin de que, si pueda reconocer motores y en base a los motores encontrados, y tras aplicar los métodos antes detallados en el capítulo 8.4.1 se obtuvo el siguiente resultado:

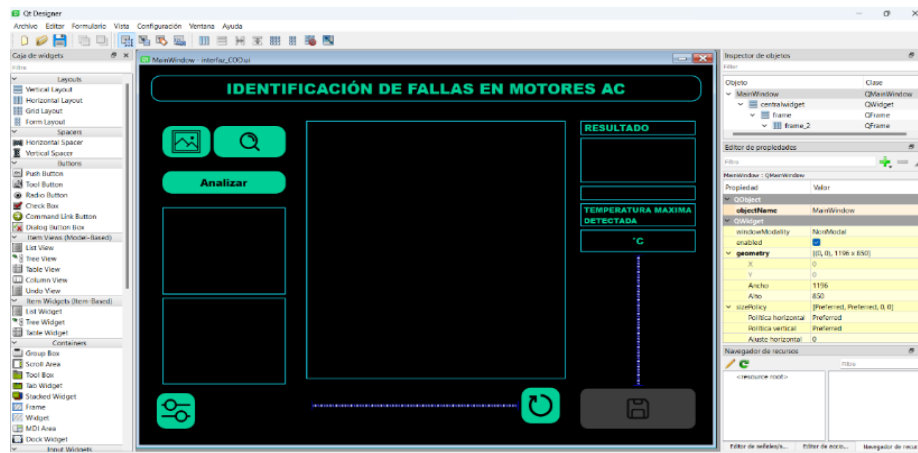
```
[32] 1 from sklearn.metrics import accuracy_score
      2 accuracy = accuracy_score(y, predicciones)
      3 print("Precisión:", accuracy*100)

⇒ Precisión: 100.0
```

*Imagen 72: Prueba de validación con Data4 al modelo 18*

## 8.5. Interfaz

Como producto final, se desarrolló una interfaz de usuario (GUI), con el objetivo de hacer más sencilla y amigable para el usuario la detección de fallas en motores en aplicaciones reales, se utilizó la librería pyQT5 para la creación de la GUI, la cual ofrece una amplia serie de widgets personalizables y una aplicación especializada para el diseño, la cual es QtDesigner.



*Imagen 73: Diseño de la interfaz dentro de QtDesigner*

Ya preparado el diseño de la interfaz en QtDesigner, es necesario añadir las funcionalidades mediante la codificación en Python. Se decidió implementar los mejores modelos de SVM disponibles (clasificación jerárquica), debido a su sencillez y baja exigencia computacional, haciendo la aplicación fácil de ejecutar localmente en computadoras con sistema operativo Windows.

A continuación se detallarán los pasos a seguir para obtener una predicción en la aplicación:

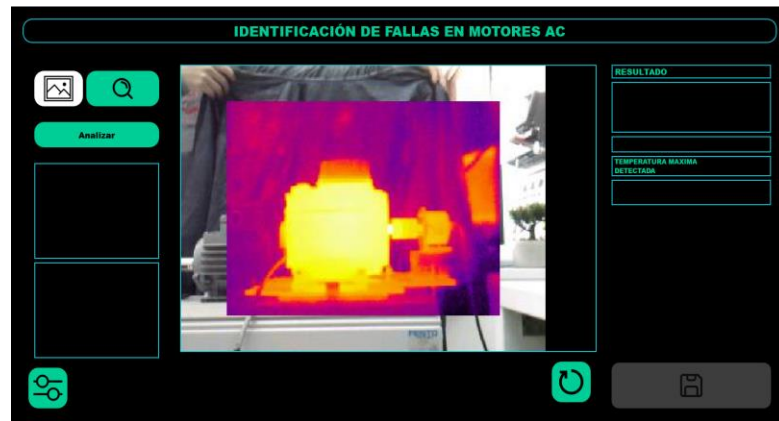
### **Selección de la imagen a analizar.**

Con el botón presente en la siguiente imagen, se abrirá un cuadro de diálogo donde se podrá elegir la imagen desde los archivos locales del dispositivo.



*Imagen 74: Botón para seleccionar la imagen*



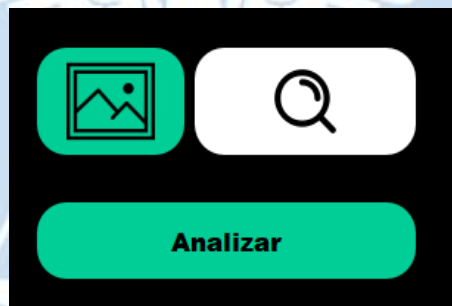


*Imagen 75: Interfaz una vez seleccionada la imagen*

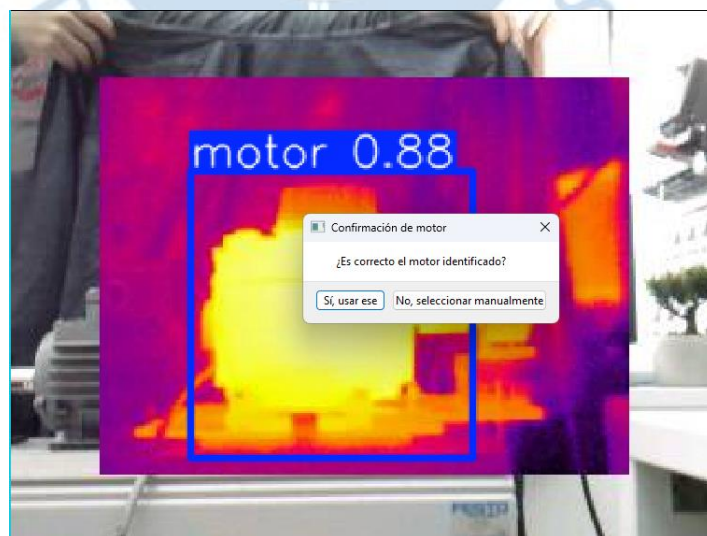
Una vez seleccionada la imagen, se mostrará la siguiente interfaz:

### **Identificación del motor a analizar en la imagen**

Con el siguiente botón, el modelo YOLO identificará el motor en la imagen ingresada, en caso no lo identifique de manera correcta, se dará la opción de seleccionarlo por cuenta propia.



*Imagen 76: Botón para elegir la imagen de manera manual*



*Imagen 77: Motor identificado de manera correcta*

Como se observa en la imagen 77 al momento de ingresar la imagen saltará una alerta para confirmar si es correcto el motor identificado, así se puede estar seguro para seguir con el análisis y reporte.

### Análisis y Reporte

Ya que se tiene identificado el motor, ya sea de manera manual o automáticamente, se procede al análisis por parte del modelo SVM, mediante el uso del botón “analizar”. Los resultados son mostrados en la parte derecha de la aplicación como se observa en la imagen 78.

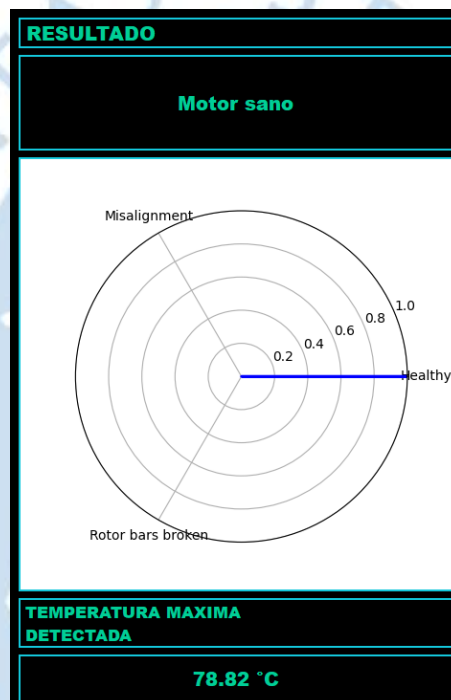


Imagen 78: Resultados obtenidos

The screenshot shows a configuration screen for temperature ranges. It starts with the instruction: 'Ingresar el rango de temperatura si es conocido, caso contrario, el rango predeterminado es 20° C - 80° C'. Below this, there are two input fields: 'Temperatura mínima:' (Minimum temperature) with a value of '20' and 'Temperatura máxima:' (Maximum temperature) with a value of '80'. Each field has '+' and '-' buttons for adjustment. At the bottom right, there is a green 'Aceptar' (Accept) button.

Imagen 79: Botón configuraciones para establecer los límites inferior y superior de temperatura

Cabe resaltar que se pueden establecer los límites inferior y superior de temperatura, a través del botón configuraciones, ubicado en la parte inferior izquierda, tal y como se demuestra en la imagen 79.

Finalmente, la aplicación nos ofrece la posibilidad de generar un reporte en pdf de la imagen analizada, el cual contiene los resultados obtenidos los cuales son: la clase predicha por el SVM, el gráfico de radar con las probabilidades de cada clase y la temperatura máxima estimada en la imagen, esto es apreciable en la imagen 80 a continuación:

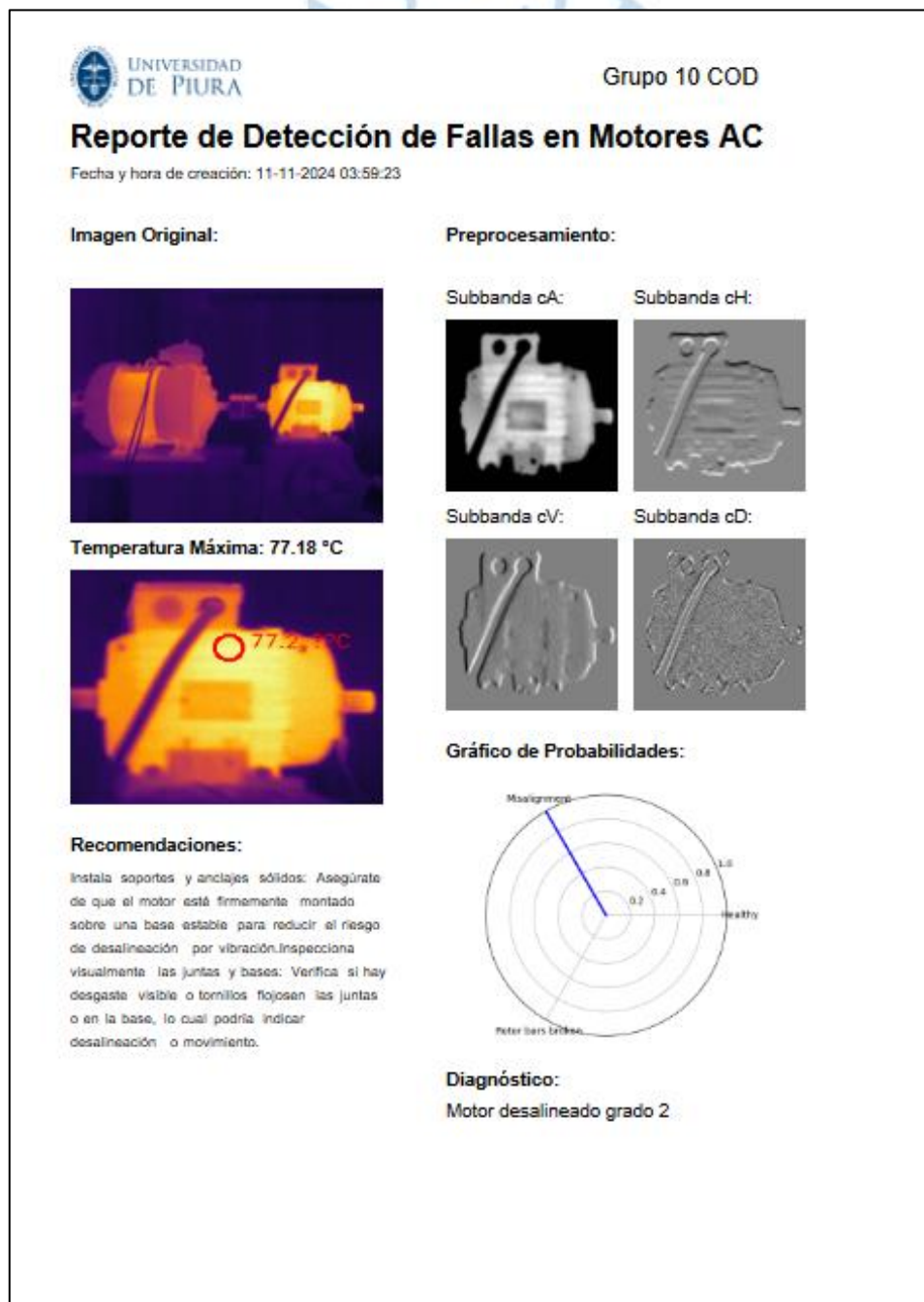


Imagen 80: Reporte final de la interfaz

## 9. Conclusiones

- I. Durante el desarrollo de las pruebas, la transformada discreta de wavelet demostró ser el preprocesamiento con resultados más precisos, llegando a alcanzar hasta un 99.18% de precisión en su evaluación con SVM, lo que nos indica una clara diferenciación de las clases, sin embargo, debido a la naturaleza del *Dataset* disponible, siempre está el riesgo de sobreentrenamiento del modelo, esto se evidencia al analizar el margen de separación del modelo, es por eso que se aplicaron técnicas como la *crossvalidation*, y se variaron parámetros como la cantidad de componentes analizadas o el parámetro de regularización. Todos estos experimentos obtuvieron porcentajes de precisión igualmente bastante altos, además de márgenes de separación mucho más altos, corroborando la validez del modelo y haciéndolo mucho más robusto.
- II. Dentro del análisis realizado con PCA (Principal Component Analysis) en el modelo 3, se detectó que dentro de las 50176 características obtenidas de las 4 sub bandas mediante DWT, el 95% de la varianza era representada por solamente los primeros 593 componentes de cada imagen, esto se obtuvo mediante la experimentación del parámetro de regularización que tiene implementado los SVM mediante el uso de Kernel RDF (radial basis function)
- III. Los modelos que han usado CNN de gran profundidad como lo han sido EfficientNetB2, B5 y B7 no han demostrado ser buenos prospectos, pues comparativamente con otros modelos, a pesar de conseguir una muy buena precisión, el tiempo de entrenamiento representa por mucho un problema mayor en cuanto a su implementación industrial.
- IV. El método de clasificación SVM-1 y SVM-2 el cual ha empleado un SVM como herramienta primordial de clasificación, ha demostrado un mejor desempeño en esta tarea que la herramienta KNN para la misma tarea.
- V. El modelo que emplea FOS como extractor de características es dada su precisión y velocidad de entrenamiento, un modelo plausible a implementar en caso lo que se busque sea un sistema de respuesta rápido, pues la carga computacional que consume es la menor de todos los modelos evaluados, obviamente tomando en cuenta también que, a su vez, han sido los modelos menos precisos entre los mismos (Modelo 11 y 12).
- VI. En caso se desee aplicar un sistema de control predictivo o simplemente un protocolo de mantenimiento predictivo usando análisis termográfico en conjunto con de redes neuronales, dada la particularidad de estas últimas, lo más adecuado sería suministrar al menos un conjunto de imágenes termográficas del motor sano en el contexto en el que trabajara, para que el método clasificador pueda al menos aprender los patrones propios de un motor sano, de esta forma nos aseguraremos siempre de que al menos la tarea de identificación de fallas en un ACM la cumplirá de manera eficiente.

- VII. El uso de las ViT es en definitiva un paso inevitable en el área de computer vision, pues tal y como ha demostrado el presente trabajo, su uso para este tipo de tareas es cercano al perfecto, inclusive frente a datasets pequeños, teniendo la ventaja que dichos porcentajes de precisión fueron obtenidos sin preprocesamiento alguno más que la detección del motor en sí (Modelo 17).
- VIII. Se ha demostrado que el uso de Kernel linear para el clasificador SVM dio mejores resultados que el Kernel Radial basis function (RDF), pues aumento la rapidez de entrenamiento, sin perder eficiencia.





## 10. Bibliografía

- Aalto university. (2008). *Independent Component Analysis*. Espoo.
- Agarap, A. F. (2017). An Architecture Combining Convolutional Neural Network (CNN) and Support Vector Machine (SVM) for Image Classification. *Cornell University*.
- Amazon Web. (s.f.). *¿Qué es una red neuronal?* Obtenido de <https://aws.amazon.com/es/what-is/neural-network/>
- Chen, C.-F., Fan, Q., & Panda, R. (2021). CrossViT: Cross-Attention Multi-Scale Vision Transformer for Image Classification. *Cornell University*.
- Chu, X., Tian, Z., Wang, Y., Zhang, B., Ren, H., Wei, X., . . . Shen, C. (2021). Twins: Revisiting the Design of Spatial Attention in Vision Transformers. *NeurIPS*.
- Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). Pre-training of deep bidirectional transformers for language. *NAACL*.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., . . . Houlsby, N. (2021). An Image is Worth 16x16 Words: Transformers For Image Recognition At Scale. *ICLR*.
- Fluke Corporation, T. S. (2009). *Introducción a los principios de la termografía*. American Technical Publishers.
- Gonzales Pajares, E. A. (2016). *Conceptos y métodos en visión por computador*. España: Grupo de Visión del Comité Español de Automática (CEA). Obtenido de <https://dialnet.unirioja.es/servlet/articulo?codigo=8620076>
- Haider, M., Doegar, A., & Verma, R. (2018). Fault Identification in Electrical Equipment using Thermal Image Processing. *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*, 853-859.
- Hassani, A., Walton, S., Shah, N., Abuduweili, A., Li, J., & Shi, H. (2021). Escaping the Big Data Paradigm with Compact Transformers. *Cornell University*.
- He, K., Zhang, X., Ren, S., & Sun, J. (2022). Deep Residual Learning for Image Recognition. *Cornell University*.
- IBM. (s.f.). *¿Qué es la segmentación de imágenes?* Recuperado el 31 de agosto de 2024, de <https://www.ibm.com/es-es/topics/image-segmentation>
- Karen Simoyan, A. Z. (2015). *VERY DEEP CONVOLUTIONAL*.
- Karim Ayon, S. T., Siraj, F., Samad, A., Uddin, J., & Choi, K. (2024). Few-Shot Lightweight SqueezeNet Architecture for Induction Motor Fault Diagnosis Using Limited Thermal Image Dataset. *IEEE Access*, 50986-50998.
- Lee, S. H., Lee, S., & Song, B. C. (2021). Vision Transformer for Small-Size Datasets. *Computer Vision and Pattern Recognition*.
- Mahami, A., Rahmoune, C., Bettahar, T., & Benazzouz, D. (2021). Induction motor condition monitoring using infrared thermography imaging and ensemble learning techniques. *Advances in Mechanical Engineering*, 1-13.

- MathWorks. (s.f.). *¿Qué son las redes neuronales convolucionales?* Recuperado el 2024 de Setiembre de 1, de <https://la.mathworks.com/discovery/convolutional-neural-network.html>
- Natalia Nieto, D. M. (2008). *EL USO DE LA TRANSFORMADA WAVELET DISCRETA EN LA RECONSTRUCCIÓN DE SEÑALES SINUSOIDALES.*
- Navaid Sabah, A., & Abdin Jaffery, Z. (2022). Fault Detection of Induction Motor Using Thermal Imaging. *Global Conference on Emerging Technologies*, 84-90.
- Padma Kala, S., Palanivel, R., Prasad Singh, M., Arulkumaran, G., & Hameed Shnain, A. (2021). Infrared Thermography-Based Induction Motor Bearing Fault Detection using Hybrid Method. *Third International Conference on Distributed Computing and Electrical Circuits and Electronics*.
- Parveen, S., Hussain, T., Kumar, D., & Chowdhry, B. (2021). INDUCTION MOTOR FAULT DETECTION AND CLASSIFICATION USING THERMAL IMAGES AND DEEP LEARNING. *Gyancity Journal of Electronics and Computer Science*, 16-32.
- Piechocki, M., Pajchrowski, T., Kraft, M., Wolkiewicz, M., & Pawel, E. (2023). Unraveling Induction Motor State through Thermal Imaging and Edge Processing: A Step towards Explainable Fault Diagnosis. *Eksploatacja i Niezawodnosc*.
- Sakalli, G., & Koyuncu, H. (2022). Discrimination of Electrical Motor Faults in Thermal Images by using First-order Statistics and Classifiers. *International Congress on Human-Computer Interaction, Optimization and Robotic Applications*.
- Singh Kalyan, G., & Syal, P. (2023). Recent Advancements of Thermal Imaging in Induction Motor: A Review. *5th International Conference on Energy, Power and Enviroment: Towards Flexible Green Energy Technologies*.
- Trejo-Chavez, O., Cruz-Albarran, I., Resendiz Ochoa, E., Salinas Aguilar, A., Morales Hernandez, L., Basurto Hurtado, J., & Perez Ramirez, C. (2023). A CNN-Based Methodology for Identifying Mechanical Faults in Induction Motors Using Thermography. *Machines*.
- Universidad de Sevilla. (s.f.). *Programa de la asignatura PID-Filtros digitales*. Obtenido de <https://grupo.us.es/gtocom/pid/programa1.html>
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A., . . . Polosukhin, I. (2017). Attention Is All You Need. *Cornell University*.
- Xu, L., Teoh, S., & Ibrahim, H. (2024). A deep learning approach for electric motor fault diagnosis based on modified InceptionV3. *scientific reports*.
- Yunus A. Cengel, A. J. (2011). *Transferencia de calor y masa*.

11. Cronograma

Cronograma de Proyecto de investigación Semestral del curso Control Basado en Datos y Modelos (COD)  
Responsables del proyecto: Carlos Banda Moreno (CB), Diego Alfonso Chunga Shimokawa (DC), Anthony Iman Miranda (AI), Joan Serrepe Castillo (JS)  
y Marcelo Farias Guerrero (MF)

