



**FACULTAD DE INGENIERÍA, ARQUITECTURA Y
URBANISMO
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**

TESIS

**DESARROLLO DE MÉTODOS PARA LA
CLASIFICACIÓN POR MADUREZ DE LA FRESA
UTILIZANDO PROCESAMIENTO DE IMÁGENES
DIGITALES Y MACHINE LEARNING
PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERO(A) DE SISTEMAS**

Autor(es)

Bach. Sanchez Pardo, Samuel

<https://orcid.org/0000-0002-2822-8416>

Bach. More Villegas, Fiorella Jhajaira.

<https://orcid.org/0000-0002-5810-1846>

Asesor

Mag. Mejia Cabrera Heber Ivan

<https://orcid.org/0000-0002-0007-0928>

Línea de Investigación

**Tecnología e innovación en el desarrollo de la construcción y la
industria en un contexto de sostenibilidad.**

Sublínea de Investigación

**Gestión y sostenibilidad en las dinámicas empresariales de industrias y
organizaciones.**

Pimentel – Perú

2024

DECLARACIÓN JURADA DE ORIGINALIDAD

Quienes suscriben la **DECLARACIÓN JURADA**, somos **SANCHEZ PARDO SAMUEL, MORE VILLEGAS FIORELLA JHAJAIRA** del Programa de Estudios de **Ingeniería de sistemas** de la Universidad Señor de Sipán S.A.C, declaramos bajo juramento que somos autores del trabajo titulado:

DESARROLLO DE MÉTODO PARA LA CLASIFICACIÓN POR MADUREZ DE LA FRESA UTILIZANDO PROCESAMIENTO DE IMÁGENES DIGITALES Y MACHINE LEARNING.

El texto de mi trabajo de investigación responde y respeta lo indicado en el Código de Ética del Comité Institucional de Ética en Investigación de la Universidad Señor de Sipán, conforme a los principios y lineamientos detallados en dicho documento, en relación con las citas y referencias bibliográficas, respetando el derecho de propiedad intelectual, por lo cual informo que la investigación cumple con ser inédito, original y auténtico.

En virtud de lo antes mencionado, firman:

<p>SANCHEZ PARDO SAMUEL</p>	<p>DNI: 75448908</p>	
<p>MORE VILLEGAS FIORELLA JHAJAIRA</p>	<p>DNI: 73329403</p>	

Pimentel, 20 de junio de 2024.

NOMBRE DEL TRABAJO

DESARROLLO DE MÉTODOS PARA LA CLASIFICACIÓN POR MADUREZ DE LA FRESA UTILIZANDO PROCESAMIENTO DE I MÁG

AUTOR

Sanchez Pardo, Samuel. /

RECuento DE PALABRAS

14990 Words

RECuento DE CARACTERES

82129 Characters

RECuento DE PÁGINAS

96 Pages

TAMAÑO DEL ARCHIVO

4.5MB

FECHA DE ENTREGA

Nov 20, 2024 3:52 PM GMT-5

FECHA DEL INFORME

Nov 20, 2024 3:54 PM GMT-5

● **18% de similitud general**

El total combinado de todas las coincidencias, incluidas las fuentes superpuestas, para cada base de datos.

- 12% Base de datos de Internet
- Base de datos de Crossref
- 13% Base de datos de trabajos entregados
- 3% Base de datos de publicaciones
- Base de datos de contenido publicado de Crossref

● **Excluir del Reporte de Similitud**

- Material bibliográfico
- Coincidencia baja (menos de 8 palabras)
- Material citado

**DESARROLLO DE MÉTODOS PARA LA CLASIFICACIÓN POR MADUREZ DE
LA FRESA UTILIZANDO PROCESAMIENTO DE IMÁGENES DIGITALES Y
MACHINE LEARNING.**

Aprobación del jurado

DR. CARLOS WILLIAN ATALAYA URRUTIA

Presidente del Jurado de Tesis

MG. DAVID ENRIQUE BANCES SAAVEDRA

Secretario del Jurado de Tesis

MG. JULIANA DEL PILAR ALVA ZAPATA

Vocal del Jurado de Tesis

Dedicatoria

A Dios, por ser la fuente de sabiduría y fortaleza en cada paso de este camino, guiándome con su luz y amor incondicional. A mi madre, por su apoyo inquebrantable y por enseñarme el valor del esfuerzo y la perseverancia. Su amor y dedicación han sido mi mayor inspiración. A mi abuela, por ser el pilar de nuestra familia y por su sabiduría y cariño que me han acompañado en cada etapa de mi vida. Su ejemplo de resiliencia y amor me motiva a seguir adelante. A mi mejor amigo y compañero de tesis, Samuel, por su constante apoyo, comprensión y colaboración. Juntos hemos enfrentado retos y celebrado éxitos, y su amistad ha sido fundamental en este proceso. A mi novio, agradezco cada momento compartido y cada palabra de aliento. Gracias a cada uno de ustedes por creer en mí y en mis sueños, y por estar a mi lado en cada paso del camino.

Fiorella Jhajaira More Villegas.

A Dios, por brindarme fortaleza, sabiduría y guía a lo largo de este proceso. Su amor y su luz han sido mi fuente inagotable de esperanza y fe. A mis padres, por su amor incondicional y su apoyo constante en cada paso de mi vida. Gracias por ser mi inspiración y mi impulso para superar cada desafío. A mis hermanos, por ser mis compañeros de vida y por su compañía inigualable. A mi abuelita, por sus consejos llenos de sabiduría y por su amor sincero y protector. A mi mejor amiga y compañera de tesis, Fiorella, por su apoyo inquebrantable y su dedicación. Su amistad ha sido una fuente de motivación y alegría durante todo este proceso. A mis amigos, por su compañía, comprensión y palabras de aliento. Gracias por estar a mi lado en cada etapa de esta travesía y por llenar mi vida de momentos inolvidables.

Sanchez Pardo Samuel.

Agradecimientos

A lo largo de nuestros años universitarios, hemos contado con el apoyo y la orientación de muchas personas que, de una manera u otra, han dejado una huella profunda en el desarrollo de esta tesis. Es un honor poder expresar nuestro más sincero agradecimiento a todas ellas.

En primer lugar, agradecemos a Dios, quien nos dio la fortaleza, sabiduría y perseverancia necesarias para superar cada desafío que se presentó en este camino. Su guía ha sido fundamental para llevar a cabo este proyecto.

A nuestras familias, les debemos el más profundo agradecimiento. Su paciencia y comprensión fueron el pilar que nos sostuvo durante los momentos más difíciles. Sin su constante apoyo, este logro no habría sido posible.

También queremos reconocer a nuestros docentes y mentores, cuyo compromiso con nuestra formación académica nos ha permitido llegar hasta este punto. En particular, agradecemos al Ing. Ivan Mejia Cabrera, por su paciencia, sabiduría y por guiarnos con dedicación a lo largo de este proceso.

Finalmente, agradecemos a nuestros amigos y compañeros, quienes estuvieron a nuestro lado brindándonos apoyo y motivación. Cada una de sus palabras y gestos de aliento ha sido una fuente invaluable de inspiración.

Este trabajo es el resultado del esfuerzo y dedicación de muchas personas a lo largo de nuestra carrera. A todos ustedes, nuestro más sincero agradecimiento por ser parte de este importante logro.

Sanchez Pardo Samuel y More Villegas Fiorella Jhajaira.

Índice

Dedicatoria	5
Agradecimientos.....	6
Índice de figuras	8
Índice de tablas	10
Resumen.....	11
Abstract	12
1. INTRODUCCIÓN	13
1.1. Realidad problemática.....	13
1.2. Formulación del problema	24
1.3. Hipótesis	25
1.4. Objetivos	25
1.5. Teorías relacionadas al tema	25
2. MATERIALES Y MÉTODO.....	48
2.1. Tipo y Diseño de Investigación.....	48
2.2. Variables, Operacionalización	48
2.3. Población de estudio, muestra, muestreo y criterios de selección	50
2.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad	50
2.5. Procedimiento de análisis de datos	51
2.6. Criterios éticos	54
3. RESULTADOS Y DISCUSIÓN	55
3.1. Resultados	55
3.2. Discusión.....	67
3.3. Aporte de la investigación	69
4. CONCLUSIONES Y RECOMENDACIONES	105
4.1. Conclusiones.....	105
4.2. Recomendaciones.....	106
5. REFERENCIAS	108
6. ANEXOS.....	122

Índice de figuras

Figura 1 Proceso de maduración de la fresa.....	26
Figura 2 Representación gráfica de la arquitectura VGG16.	29
Figura 3 Representación gráfica de la arquitectura VGG19.	30
Figura 4 Representación gráfica de la arquitectura ResNet 50.	31
Figura 5 Ejemplo de redes neuronales.	34
Figura 6 Ejemplo de redes neuronales recurrentes.....	35
Figura 7 Ejemplo Multilayer perceptrons.	37
Figura 8 Ejemplo de Capa convolucional.....	38
Figura 9 Ejemplo de Capa max pooling.	39
Figura 10 Gráfica de la ecuación de la función sigmoide en un plano cartesiano.	40
Figura 11 Gráfica de la función tangente hiperbólica en un plano cartesiano.....	41
Figura 12 Gráfica de la función ReLU en un plano cartesiano.	42
Figura 13 Ejemplo de la matriz de confusión.	44
Figura 14 Gráfico de barras para el indicador de promedio de tiempo de respuesta para aprendizaje por transferencia.	56
Figura 15 Gráfico de barras para el indicador de promedio de tiempo de respuesta para aprendizaje supervisado.	57
Figura 16 Gráfico de barras para el indicador grado de consumo de CPU para aprendizaje por transferencia.	58
Figura 17 Gráfico de barras para el indicador grado de consumo de CPU para aprendizaje supervisado.	59
Figura 18 Gráfico de barras para el indicador grado de memoria RAM para aprendizaje por transferencia.	60
Figura 19 Gráfico de barras para el indicador grado de memoria RAM para aprendizaje supervisado.	61
Figura 20 Gráfico de barras para el indicador de precisión para aprendizaje por transferencia.	62
Figura 21 Gráfico de barras para el indicador de precisión para aprendizaje supervisado.	63
Figura 22 Gráfico de barras para el indicador de exactitud para aprendizaje por transferencia.....	64
Figura 23 Gráfico de barras para el indicador de exactitud para aprendizaje supervisado.	65
Figura 24 Gráfico de barras para el indicador recall para aprendizaje por transferencia.....	66
Figura 25 Gráfico de barras para el indicador recall para aprendizaje supervisado.	67
Figura 26 Esquema del 1° método.....	69
Figura 27 Esquema del 2° método.....	70
Figura 28 Estructura para tomar fotografías a las fresas.....	71
Figura 29 Binarización de la imagen con el método Otsu.	72
Figura 30 Extracción de la fresa del fondo de la imagen.....	73

Figura 31 Rotación en 90, 180 y 270 grados cada imagen.	73
Figura 32 Arquitectura del modelo VGG19.	83
Figura 33 Arquitectura del modelo ResNet 50.	84
Figura 34 Arquitectura del modelo Mobile Net V2.	85
Figura 35 Arquitectura del modelo AlexNet.	87
Figura 36 Arquitectura del modelo VGG16.	88
Figura 37 Matriz de confusión del modelo VGG19.	91
Figura 38 Histograma de exactitud del modelo VGG19.	91
Figura 39 Histograma de pérdidas del modelo VGG19.	92
Figura 40 Matriz de confusión del modelo ResNet 50.	93
Figura 41 Histograma de exactitud del modelo ResNet 50.	93
Figura 42 Histograma de pérdida del modelo ResNet 50.	94
Figura 43 Matriz de confusión del modelo MobileNet.	95
Figura 44 Histograma de exactitud del modelo MobileNet.	95
Figura 45 Histograma de pérdida del modelo MobileNet.	96
Figura 46 Matriz de confusión del modelo AlexNet.	97
Figura 47 Histograma de exactitud del modelo AlexNet.	97
Figura 48 Histograma de pérdida del modelo AlexNet.	97
Figura 49 Matriz de confusión del modelo VGG16.	98
Figura 50 Histograma de exactitud del modelo VGG16.	99
Figura 51 Histograma de pérdida del modelo VGG16.	99
Figura 52 Matriz de confusión del modelo de regresión logística.	100
Figura 53 Histograma de exactitud del modelo de regresión logística.	100
Figura 54 Matriz de confusión del modelo de Random Forest.	101
Figura 55 Histograma de exactitud del modelo de Random Forest.	102
Figura 56 Matriz de confusión del modelo SVM.	103
Figura 57 Histograma de exactitud del modelo SVM.	103
Figura 58 Matriz de confusión del modelo Naive Bayes.	104
Figura 59 Histograma de exactitud del modelo Naive Bayes.	105

Índice de tablas

Tabla 1 Operacionalización de variables.....	49
Tabla 2 Resultados del indicador promedio de tiempo de respuesta para aprendizaje por transferencia.....	56
Tabla 3 Resultados del indicador promedio de tiempo de respuesta para aprendizaje supervisado.	56
Tabla 4 Resultados del indicador grado de consumo de CPU para aprendizaje por transferencia.....	57
Tabla 5 Resultados del indicador grado de consumo de CPU para aprendizaje supervisado.	58
Tabla 6 Resultados del indicador grado de memoria RAM para aprendizaje por transferencia.	59
Tabla 7 Resultados del indicador grado de memoria RAM para aprendizaje supervisado.	60
Tabla 8 Resultados del indicador de precisión para aprendizaje por transferencia.	61
Tabla 9 Resultados del indicador de precisión para aprendizaje supervisado.	62
Tabla 10 Resultados del indicador de exactitud para aprendizaje por transferencia.	63
Tabla 11 Resultados del indicador de exactitud para aprendizaje supervisado.	64
Tabla 12 Resultados del indicador recall para aprendizaje por transferencia.	65
Tabla 13 Resultados del indicador recall para aprendizaje supervisado.	66
Tabla 14 Implementación del modelo de color HSV.....	74
Tabla 15 Implementación del código para el valor de brillo.	74
Tabla 16 Implementación del código para obtener la textura.	75
Tabla 17 Descripción de las características extraídas de las imágenes	75
Tabla 18 Función para modificar las últimas 5 capas.....	82
Tabla 19 Implementación del modelo VGG19.....	83
Tabla 20 Implementación del modelo ResNet 50.	83
Tabla 21 Implementación del modelo Mobile Net V2.	84
Tabla 22 Implementación del modelo AlexNet.	85
Tabla 23 Implementación del modelo VGG16.....	87
Tabla 24 Implementación del modelo de Regresión logística.....	88
Tabla 25 Implementación del modelo Random Forest	88
Tabla 26 Implementación del modelo SVM	89
Tabla 27 Implementación del modelo Naive bayes	89
Tabla 28 Comparación de resultados del indicador de precisión.....	90
Tabla 29 Modelos de machine learning usados en otras investigaciones. .	124

Resumen

Actualmente, la clasificación de fresas se realiza principalmente de forma manual, lo que puede provocar errores, mayor tiempo de procesamiento y pérdida de calidad. Esto representa un desafío para las industrias que venden fresas, ya que deben tomar decisiones basadas en la madurez de las frutas para optimizar la gestión de cultivos y la comercialización. Los modelos tradicionales de machine learning son complejos y requieren pasos como la extracción de características y grandes cantidades de datos para entrenarse de manera efectiva.

Para mejorar este proceso, se propuso el uso de modelos de clasificación basados en transfer learning y aprendizaje supervisado, que pueden clasificar las fresas según su nivel de madurez con menos datos de entrenamiento. En esta investigación, se utilizaron modelos de transfer learning para el desarrollo de un primer método y aprendizaje supervisado para el desarrollo de un segundo método para clasificar las fresas en tres niveles de madurez: verde, madura y muy madura. Los modelos evaluados fueron VGG19, VGG16, ResNet50, MobileNetV2, AlexNet, regresión logística, random forest, SVM y Naive bayes.

Los modelos de regresión logística y Random Forest usados en el segundo método mostraron un excelente desempeño, alcanzando una precisión del 100% en el test y del 100% en la validación, otro lado, entre en el método con transfer learning, el modelo ResNet-50 logró el mejor rendimiento con 97.66%. La implementación de estos modelos de clasificación puede mejorar la eficiencia y calidad en el proceso de selección de las fresas, beneficiando a agricultores y pequeñas industrias en su comercialización.

Palabras Clave: procesamiento de imágenes, transfer learning, clasificación de fresas, red neuronal, imágenes digitales.

Abstract

Currently, strawberry grading is mainly done manually, which can lead to errors, longer processing time and loss of quality. This presents a challenge for industries that sell strawberries, as they must make decisions based on fruit ripeness to optimize crop management and marketing. Traditional machine learning models are complex and require steps such as feature extraction and large amounts of data to train effectively.

To improve this process, the use of classification models based on transfer learning and supervised learning is proposed, which can classify strawberries according to their maturity level with less training data. In this research, transfer learning and supervised learning models were used to classify strawberries into three maturity levels: green, ripe, and very ripe. The models evaluated were VGG19, VGG16, ResNet50, MobileNetV2, AlexNet, logistic regression, random forest, SVM and Naive bayes.

The logistic regression and Random Forest models showed excellent performance, reaching 100% accuracy in the test and 100% accuracy in the validation. The implementation of these classification models can improve efficiency and quality in the strawberry selection process, benefiting farmers and small industries in their commercialization.

Keywords: image processing, transfer learning, strawberry classification, neural network, digital images, supervised learning.

1. INTRODUCCIÓN

1.1. Realidad problemática.

La automatización en los procesos industriales es fundamental para incrementar la producción, acelerar la productividad y reducir la tasa de errores en la clasificación de calidad de los productos así como los costos de producción. En el sector agrícola, la clasificación de fresas representa un desafío significativo debido a la necesidad de tomar decisiones precisas basadas en la madurez de las frutas, lo que impacta directamente en la gestión de cultivos y la comercialización [1].

En el 2023 Perú exportó 33,830 toneladas de fresa, alcanzando un valor de US \$60 millones. Esto representó un aumento del 22% en términos de volumen y un 17% en términos de valor en comparación con el año 2022 [2]. A nivel mundial, Europa es el mayor productor de fresas, con España ocupando el tercer lugar global y actuando como principal importador de fresas peruanas, que posteriormente se redistribuyen a otros países de la Unión Europea. En Perú, el cultivo de fresas en Huaral puede costar hasta S/. 25,000 soles por hectárea, con una producción estimada de 40 toneladas por hectárea en condiciones óptimas [3].

Las fresas no sólo son económicamente rentables, sino también nutricionalmente beneficiosas. Tienen un bajo contenido calórico, siendo ricas en agua y carbohidratos, y contienen vitaminas del grupo B, vitamina C, magnesio, potasio y vitamina K, lo que las convierte en antioxidantes y antiinflamatorios naturales [4].

A pesar de estos beneficios, la clasificación manual de fresas es propensa a errores y puede dañar los frutos, además de ser un proceso lento y costoso. Por ello, es crucial automatizar este proceso para mejorar la eficiencia y calidad del producto final [5].

En la literatura científica se han propuesto diferentes métodos para clasificar la madurez de las fresas de manera automática o semiautomática. Entre las cuales

encontramos, los métodos basados en el procesamiento de imágenes en escala de color HSI, que se muestra prometedor para estimar la madurez de fresas en tiempo real bajo condiciones de campo y laboratorio [6]. Otras investigaciones han implementado métodos basados en aprendizaje automático, tal es el caso de [7] que implementó un método para categorizar la madurez en papayas usando el modelo VGG-19 con base en un enfoque de transferencia, logrando una alta exactitud en la clasificación. [8] en su evaluación de diferentes modelos de aprendizaje de transferencia VGG, Inception y ResNet, observaron en sus resultados que VGG-19 fue el que obtuvo el mejor rendimiento. En [9] se propuso utilizar la arquitectura YOLO en sus versiones 3 y 4 con sus variantes derivadas de estas para estimar el nivel de madurez en bayas silvestres, logrando una precisión con YOLO v4 de 79,79% y 88,12%.

Similar es el caso del trabajo propuesto por [7], donde se propuso un método basado en el aprendizaje de transferencia, que compara 6 modelos, con el fin de clasificar el nivel de madurez en papayas, encontrando que el modelo VGG-19 logra una alta precisión.

Por otro lado también existen investigaciones donde implementaron métodos enfocados en aprendizaje supervisado, como lo hace [18] donde examinan el empleo de diversas características visuales diseñadas manualmente para clasificar frutas mediante métodos tradicionales de aprendizaje supervisado. Se evalúan diferentes características relacionadas con el color, la forma y la textura, y se comparan los resultados obtenidos al aplicar seis técnicas de aprendizaje supervisado, entre ellas: K-vecinos más cercanos, máquinas de soporte vectorial, Naïve Bayes, análisis discriminante lineal, Random Forest y redes neuronales de propagación hacia adelante y hacia atrás. Así como [002] donde proponen un método fundamentado en el análisis de componentes principales, que permite identificar los parámetros más relevantes de forma y color. La comparación de los algoritmos de clasificación utilizados ha revelado que los mayores niveles de precisión en la clasificación de almendras se obtienen al emplear máquinas de soporte vectorial en lugar de redes

neuronales artificiales. Además, los resultados experimentales mostraron que el método propuesto es notablemente robusto y eficiente en términos computacionales, lo cual facilita su aplicación en tiempo real.

Este estudio propone dos métodos para la clasificación automática e inmediata del nivel de madurez de las fresas en un ambiente controlado, utilizando modelos de transferencia de aprendizaje y modelos de aprendizaje supervisados. Se ha construido un dataset de 444 imágenes en espacio de color RGB, preprocesadas y aumentadas para incrementar el tamaño del conjunto de datos. Derivado del mismo dataset de imágenes se generó otro extrayendo las características de color HSV (Hue, Saturation, Value – Matiz, Saturación, Valor), brillo y características de textura de Haralick. Entre los modelos de aprendizaje por transferencia se evaluaron VGG-16, VGG-19, ResNet-50, MobileNet V2, AlexNet y entre los modelos de aprendizaje supervisado Regresión logística, Random Forest, Support vector machine (SVM) y Naive Bayes. Los resultados se analizarán utilizando métricas como precisión, error cuadrático medio, recall, tiempo de respuesta, y consumo de recursos (CPU y RAM), además de evaluar la matriz de confusión.

La implementación de estos modelos de clasificación promete mejorar la eficiencia y calidad en la selección de fresas, beneficiando a agricultores y pequeñas industrias en su comercialización.

A partir de los estudios de [6] se encontraron que debido a la evaluación de la calidad de las fresas es llevada por expertos de forma manual que buscan criterios de color, textura y componentes químico. Este método de control de calidad suele ser en algunas ocasiones destructivo para la condición de la fresa, por lo cual se busca métodos menos destructivos para evaluar las fresas según su color que es indicativo de su madurez, por esto se propuso un método, para el cual obtuvieron imágenes de fresas en su etapa temprana de madurez, de las cuales se obtuvo sus características utilizando el sistema de imágenes

hiperespectrales (HSI), posteriormente se aplicó el algoritmo de selección de características secuenciales (SFS) y que fueron validados con el clasificador SVM para finalmente usarse las características en el modelo de red neural AlexNet pre entrenada para clasificar la madurez de las fresas. Los resultados derivados de la extracción de características y selección con los que se entrenó el modelo AlexNet dieron una precisión de 98.6% con el grupo de datos de prueba. Se puede concluir que el uso del sistema HSI en tiempo real es prometedor para la estimación de madurez en las fresas ya sea en condiciones de campo o de laboratorio, lo cual mejoraría la gestión del tiempo de cosecha de los agricultores y productores.

En el estudio de [7] vieron que para el envasado de la fruta de la papaya para ser comercializado es necesario verificar anteriormente el nivel de madurez del fruto, el cual es subjetivo a la percepción humana, lo cual conlleva mucho tiempo y es destructivo para el fruto. Para lo cual se sugiere dos enfoques, uno de aprendizaje automático y otro de aprendizaje por transferencia, para lo cual se utilizó 300 imágenes, siendo cada 100 de un nivel de madurez, se utilizaron distintos enfoques para entrenar los modelos de clasificación, entre ellos se encuentran el patrón binario local (LBP), el histograma de gradientes orientados (HOG), la matriz de ocurrencia de coincidencias de nivel gris (GLCM), el vecino más cercano (KNN), la máquina de vectores de soporte (SVM) y Naive Bayes. Para el enfoque de transferencia los modelos VGG 16, ResNet 101, ResNet 18, ResNet 50, VGG 19, GoogleNet y AlexNet. Dando como resultado que la función KNN ponderada con HOG supera a otros modelos de clasificación basados en aprendizaje automático con un 100% de precisión y un tiempo de entrenamiento de 0,0995 s. y el modelo de clasificación basado en el enfoque de transferencia de aprendizaje, VGG 19 funciona mejor con una precisión del 100% y un tiempo de entrenamiento de 1 min 52 s. Finalmente concluyeron que aunque la función KNN ponderada con HOG y el VGG-19 logran una precisión de 100% y el menor tiempo de entrenamiento lo logra el primero. Se considera que el mejor es el VGG 19 debido a que no

es necesario el tiempo de formación de características y grandes conjuntos de datos para obtener el mismo resultado.

Para [9] lograron identificar el problema que a diferencia de otras frutas, las bayas silvestres una vez cosechada no sigue madurando, por lo cual una mala decisión en el proceso de recolección hará que algunas bayas no puedan ser comercializadas, lo que disminuye las ganancias, por lo cual se requiere un método para prevenir este inconveniente. Por esto en la investigación se propuso analizar 6 modelos de redes neuronales artificiales, los cuales fueron YOLO v3, YOLO v3-Tiny, YOLO v3-SPP, YOLO v4, YOLO v4-Small y YOLO v4-Tiny. Para clasificar entre los 3 tipos de bayas silvestres verdes, rojas y azules y para cada una 2 clases de madurez. Utilizando imágenes de 1280 x 736 para el entrenamiento de los modelos. Usando como guía a YOLO v4 el que tuvo los mejores resultados respecto a la precisión promedio con 79,79% y 88,12% y en F1 con 0.82%. El modelo YOLO v4-Tiny logró el menor tiempo con 7.8ms y uso de RAM de 1.63GB. Y el modelo YOLO v4-Small logró el menor desempeño en el error absoluto medio de 24.1%. En base a los resultados se espera que este novedoso método ayude a los productores a tomar mejores decisiones de gestión, localizar las bayas, mejorar el rendimiento y por consiguiente las ganancias de producción.

Según [10] se observaron que los estimadores actuales de madurez basados en NIR que utilizan regresores de aprendizaje automático imponen umbrales altos y que en algunas veces es destructivo con el fruto. Por lo cual se propusieron utilizar un clasificador entrenado con etiquetas asignadas a través de un umbral estándar de MS, con lo cual se entrenó un modelo KNN. Donde se evaluó la estimación indirecta de madurez (humana) con la indirecta (propuesta), dando la primera una precisión de 55,9% y la propuesta 88.2% con el modelo de clasificación KNN en el estado de la madurez con los datos de prueba. Concluyendo que los resultados logran mostrar que la técnica propuesta para estimar la madurez tiene un potencial considerable para clasificar la madurez de los mangos en situaciones prácticas.

Sobre patrones de calidad para el mercadeo de los espinos en base al color, tamaño y nivel de madurez, siendo la clasificación del nivel de madurez de los espinos muy importante para su comercialización. En el estudio [11] propusieron utilizar técnicas de inteligencia artificial y procesamiento de imágenes para clasificar los espinos según tres categorías de madures (inmaduro, maduro y demasiado maduro) con un total de 600 imágenes de las cuales se extrajeron propiedades geométricas, de color y características de textura con lo que se entrenaron dos algoritmos clásicos que fueron red neural artificial (ANN) y máquina de vectores de soporte (SVM). Dando como los resultados del entrenamiento el modelo ANN logró ser el más eficiente logrando una precisión de 99.57%, 99.16% y 98,16% para cada etiqueta y con un error cuadrático medio (MSE) de 1×10^{-3} , 8×10^{-3} y 3×10^{-3} para las fases de entrenamiento, validación y prueba, respectivamente. Concluyendo que el uso combinado de métodos de visión artificial con algoritmos de aprendizaje automático logra clasificar de forma exitosa el nivel de madurez en los espinos.

Por medio de [12] se menciona que el procedimiento clásico para juzgar los estados de madurez de las fresas se basa en decidir manualmente la apariencia, la textura, el color, el sabor y la firmeza de las fresas, lo cual necesita un largo tiempo. Por consiguiente, es fundamental plantear una regla para la categorización de los estados de madurez de las fresas y desarrollar un procedimiento económico y eficiente para detectar fácilmente y con exactitud cada estado de madurez de las fresas. Por lo cual se propuso el desarrollo de un sistema que logre ejecutar de manera automática la clasificación de la madurez de fresas para una categorización inmediata y estricta de diferentes estados de madurez de fresas, comparando 2 procedimientos de adquisición de imágenes de fresas, imágenes aéreas e imágenes cercanas al suelo, usando el mismo procedimiento de procesamiento de imágenes de aprendizaje profundo. Como consecuencia, se ordenaron 3 y 7 fases de madurez de fresa para imágenes de vehículos aéreos no tripulados (UAV) e imágenes de cámaras digitales cercanas al suelo, respectivamente. Dando como resultado que las imágenes de UAV, la más

grande exactitud promedio (mAP) de la categorización de la madurez de la fresa ha sido de 0,88 para un grupo de datos de prueba a 2 m, y la más grande exactitud promedio de categorización (AP) ha sido de 0,93 para la fruta enteramente madura. Para las imágenes de cámaras digitales cercanas al suelo, el mAP de la categorización de madurez de la fresa ha sido de 0,89, y el AP de categorización mayor ha sido de 0,94 además para la fruta enteramente madura. Concluyendo que las imágenes de cámaras digitales cercanas al suelo, el mAP de la categorización de madurez de la fresa ha sido de 0,89, y el AP de categorización mayor ha sido de 0,94.

Con base en [13] se estudió el problema de la lucha del ámbito agrícola enormemente variable y en constante cambio, lo que dificulta una óptima clasificación de pimientos maduros. Por lo cual propusieron el uso de algoritmos para categorizar de madurez de pimientos dulces rojos y amarillos, con datos adquiridos de 2 conjuntos de datos diferentes, incluidas 296 imágenes. Empezando con la adquisición de imágenes de dos cultivos distintos, pimientos dulces amarillos y rojos, a ambos cultivos se le realizó una segmentación de imágenes colocándolos en un cubículo de iluminación fija y usando la herramienta MathLab, dicha segmentación se dio en primer lugar separando los pimientos del fondo basándose en la segmentación mediante manchas. Luego se logró extraer características basadas en el color y características morfológicas, para lograr aplicar los dos algoritmos de clasificación, el primero está basado en la regresión logística y el segundo es un algoritmo de bosque aleatorio basado en Random Forest. Dando como resultado que la categorización de madurez consiguió una exactitud de 98,2% y 97,3% para clasificar en 2 clases, entre clases maduras e inmaduras de pimiento rojo y amarillo respectivamente, y una exactitud de 89,5% y 97,3% para clasificar en 4 clases de madurez. Dando como resultado que la clasificación mediante el algoritmo de bosque aleatorio fue mejor, y alcanzó un 98,2% y un 97,2% de precisión para la clasificación de madurez e inmadurez de los pimientos dulces rojos y amarillos respectivamente, y un 89,5% y un 97,3% de precisión en los cuatro niveles de clasificación

de madurez. Concluyendo que ambos procedimientos propuestos en esta indagación para la categorización de nuevos conjuntos de datos necesitan pequeñas porciones de datos, por lo cual son aplicables a los datos agrícolas.

A través de [14] vieron que la madurez es el primordial indicador de la calidad del tomate a partir de la perspectiva de los consumidores, la decisión de los estados de madurez del tomate es una inquietud industrial elemental relacionadas con la producción de tomate para obtener un producto de alta calidad. Por lo cual se propuso un enfoque de clasificación multiclase automatizado para lograr la medida y evaluación de la madurez del tomate por medio de la averiguación y categorización de los diferentes estados de madurez, que usa el estudio de Elementos Primordiales (PCA), además de los algoritmos de aparatos de Vectores de Apoyo (SVM) y Estudio Discriminante Lineal (LDA) para la sustracción de propiedades y la categorización, respectivamente. Los resultados obtenidos en el experimento demostraron que el enfoque de categorización propuesto ha logrado una exactitud de categorización de la madurez del 90,8%, usando un algoritmo SVM multiclase (OAO) con funcionalidad de núcleo lineal, una exactitud de categorización de la madurez del 84,8% usando un algoritmo SVM multiclase (OAA) con funcionalidad de núcleo lineal, y una exactitud de categorización de la madurez del 84% usando el algoritmo LDA. Concluyendo que la exactitud de la categorización de la madurez obtenida por el enfoque SVM multiclase de OAO, SVM multiclase es mejor que la exactitud de categorización de la madurez obtenida por los enfoques SVM multiclase OAA y LDA.

Mediante [15] se menciona que el cultivo de tomates es la principal fuente de recursos de los agricultores en la India, pero sin embargo presentan deficiencia en el almacenamiento inadecuado, por el cual padecen gigantes pérdidas. Para reducir estas pérdidas, un parámetro determinante que se debe mantener el control es la madurez. Por lo cual proponen implementar tres enfoques de aprendizaje de transferencia, VGG, Inception y ResNet. Para la realización del estudio se usó un grupo de datos de tomates que integraba las 3 clases de

tomates. Los modelos se evaluaron entrenando de manera iterativa con un tiempo y un tamaño de lote cambiantes. Los métodos analizados fueron el VGG 16, VGG 19, Inception V3, ResNet 101 y ResNet 152. Cada uno de estos modelos trabaja con 300 imágenes. En los resultados el modelo VGG 16 alcanzó una precisión del 94,74% y del 95,24% en las épocas 25 y 50 con un tamaño de lote de 32. Con un tamaño de lote de 64, se consigue la mayor precisión de clasificación, un 93,42%, que se mantiene para las épocas 25 y 50. El VGG 19 ofrece la mayor precisión de clasificación del 96,05% y 97,37% en las épocas 25 y 50 con un tamaño de lote de 32. En segundo lugar, ofrece una precisión del 92,11% y del 96,05% en las épocas 25 y 50 con un tamaño de lote de 64. Por consiguiente, se ha comprobado que el aprendizaje por transferencia es una solución posible para la categorización de imágenes. Y puede adoptarse en la ingesta de alimentos y la agricultura para solucionar inconvenientes involucrados con la categorización y el reconocimiento de manera económica y estricta.

A partir de [16] se estudió el problema de los procedimientos clásicos de procesamiento de imágenes no logran identificar correctamente las 4 fases de maduración entre los cultivos de olivo iraníes, o sea, Zard y Roghani, cuando éstas aún se encuentran en las ramas. Por esto se propone un modelo el cual se fundamenta en imágenes RGB y en redes neuronales convolucionales profundas más conocidas como CNN. El modelo planteado tiene diversas capas de convolución, Max-Pooling y normalización por lotes, seguidas de una capa de Universal Average Pooling (GAP). Para esta red, se evaluaron 6 optimizadores diferentes, a saber, Adagrad, SGD, SGDM, RMSProp, Adam y Nadam. Según los resultados, la exactitud universal de la técnica postulada ha sido del 91,91 %, e hizo el procesamiento de una sola trama en solamente 12,64 ms en la CPU. Esto refleja el potencial en tiempo real y la solidez del modelo para clasificar las aceitunas en rama en función de su estado de maduración. Concluyendo que el modelo propuesto puede integrarse eficazmente en un sistema de tratamiento de olivos basado en las condiciones de una rama concreta en la horticultura de precisión. Teniendo como base esta investigación este modelo logró reconocer

las 8 clases de aceitunas trabajadas.

Para [17] el uso de visión artificial para la clasificación automática de frutas enfrenta desafíos debido a la diversidad en las características de distintos tipos de frutas. Este estudio presenta un enfoque novedoso que emplea una máquina de vectores de soporte multiclase con núcleo (kSVM) para lograr una clasificación precisa y eficiente. Inicialmente, se tomaron imágenes de frutas con una cámara digital y se eliminó el fondo de cada imagen mediante un algoritmo de partición y fusión. Luego, se extrajeron características como el color, la textura y la forma de cada fruta para formar un espacio de características. A continuación, se aplicó el análisis de componentes principales (PCA) para reducir la dimensionalidad de este espacio. Basado en esto, se construyeron tres tipos de SVM multiclase: SVM Winner-Takes-All, SVM Max-Wins-Voting y SVM de grafo acíclico dirigido, utilizando tres tipos de núcleos: lineal, polinomial homogéneo y de base radial gaussiana. Los SVM fueron entrenados mediante validación cruzada estratificada de 5 pliegues con los vectores de características reducidas. Los resultados experimentales indicaron que el SVM Max-Wins-Voting con núcleo de base radial gaussiana alcanzó la mayor precisión de clasificación, con un 88.2 %. En términos de velocidad, el SVM de grafo acíclico dirigido fue el más rápido en procesamiento.

Mediante [18] el análisis de señales visuales para la clasificación y selección de frutas permite automatizar la inspección visual y el empaquetado en el sector agrícola, actividades que tradicionalmente dependen de la intervención humana. Los sistemas automáticos de clasificación multiclase enfrentan desafíos debido a la similitud de color y forma entre diferentes variedades de frutas y la variabilidad dentro de una misma categoría. Una limitación relevante de las redes neuronales profundas en este contexto es la necesidad de grandes conjuntos de datos de entrenamiento para lograr una alta precisión, los cuales no siempre están disponibles en la agricultura, especialmente para frutas y verduras diversas. Por ello, es necesario un análisis exhaustivo para identificar combinaciones de características específicas que puedan mejorar la precisión en conjuntos de datos pequeños. Este estudio

investiga el uso de características visuales seleccionadas manualmente para clasificar frutas utilizando técnicas convencionales de aprendizaje automático. Se analizan distintas características de color, forma y textura, y se comparan los resultados de seis algoritmos supervisados: K vecinos más cercanos, máquinas de vectores de soporte, Naïve Bayes, análisis discriminante lineal, Random Forest y redes neuronales de retropropagación. Se propone una combinación innovadora de características —como matiz, Color-SIFT, Transformada Wavelet Discreta y características de Haralick— para resolver el problema de clasificación de frutas, superando a otras características artesanales. Esta combinación es resistente a cambios de rotación e iluminación y maneja bien la variabilidad dentro de cada clase, logrando una identificación precisa de subcategorías y alta precisión en categorías visualmente similares. Además, se observó que ColorSIFT es especialmente efectivo en la clasificación de frutas, superando a otras características individuales. El enfoque fue entrenado y probado en el conjunto de datos público Fruits 360, y los mejores resultados se obtuvieron con redes neuronales de retropropagación, SVM y KNN, alcanzando precisiones entre el 99% y el 100%.

En el estudio de [19], el café fue identificado como una de las bebidas preferidas por los habitantes de Indonesia, donde se producen principalmente dos tipos: Arábica y Robusta. Tradicionalmente, la clasificación de los granos de café se realizaba de forma manual y dependía de los sentidos humanos. Sin embargo, se observó que este método era inconsistente, ya que la percepción humana podía variar según el estado mental o físico del evaluador, ofreciendo solo medidas cualitativas. En este estudio, la clasificación de los granos de café se llevó a cabo mediante procesamiento digital de imágenes. Los parámetros analizados incluyeron la textura, usando la Matriz de Coocurrencia de Niveles de Grises (GLCM) y sus cuatro características: Energía, Correlación, Homogeneidad y Contraste. Para extraer las características, se aplicaron diversos algoritmos de clasificación, como Bayes ingenuo, Random Forest, Máquina de Vectores de Soporte (SVM) y Regresión logística. La

evaluación del modelo se basó en parámetros como AUC, F1, CA, Precisión y Recuperación, usando un conjunto de datos de 29 imágenes de granos de café Arábica y 29 de Robusta. La validación cruzada sirvió para probar la precisión del modelo, y los resultados fueron evaluados con una Matriz de Confusión. Finalmente, se determinó que el método SVM fue el más efectivo, con valores de AUC = 1, CA = 0.983, F1 = 0.983, Precisión = 0.983 y Recall = 0.983.

En el estudio [20], se aplicaron técnicas de clasificación de aprendizaje automático H2O para categorizar manzanas de acuerdo con el color de su piel. Para cada variedad de manzana, se evaluaron 60 muestras, y los valores de color se obtuvieron en el espacio de color L^* , a^* y b^* , utilizando un espectrofotómetro portátil. Las variedades de manzana Red Delicious, Golden Delicious y Granny Smith fueron seleccionadas aleatoriamente para conformar la base de datos. En el análisis de los datos, se emplearon los algoritmos H2O Gradient Boosting Machine, H2O Random Forest y H2O Naive Bayes. El conjunto de datos se dividió en un 70% para el entrenamiento y un 30% para las pruebas. En la sección de conclusiones se detallaron los resultados del rendimiento de los clasificadores, incluyendo precisión (%), porcentaje de error (%), F-Measure, coeficiente de Kappa de Cohen, recall, precisión, verdadero positivo (TP), falso positivo (FP), verdadero negativo (TN) y falso negativo (FN). Los resultados mostraron una precisión del 100,0% para H2O Gradient Boosting Machine, 98,4% para H2O Random Forest y 100,0% para H2O Naive Bayes.

1.2. Formulación del problema

¿Qué modelos de procesamiento de imágenes digitales y algoritmos de machine learning son más efectivos para clasificar la madurez de las fresas?.

1.3. Hipótesis

Las redes convolucionales logran obtener una mayor precisión para clasificar la madurez de las fresas usando imágenes digitales.

1.4. Objetivos

1.4.1. Objetivo general

Desarrollar dos métodos para lograr la clasificación por madurez de la fresa utilizando procesamiento de imágenes digitales y machine learning.

1.4.2. Objetivos específicos

1. Construir dataset de imágenes digitales de fresas.
2. Realizar el procesamiento de las imágenes digitales de las fresas.
3. Crear un dataset con características extraídas de las imágenes procesadas.
4. Seleccionar los modelos de clasificación de machine learning a implementar.
5. Realizar la implementación de los modelos seleccionados.
6. Ejecutar pruebas y generar resultados.

1.5. Teorías relacionadas al tema

1.5.1. Madurez.

La madurez en la cosecha involucra que la fruta ha completado todo el proceso de maduración fisiológica. Por consiguiente, muestra propiedades como pigmentación, pérdida de color verde, crecimiento de la firmeza disuelta y reducción de la firmeza [21].

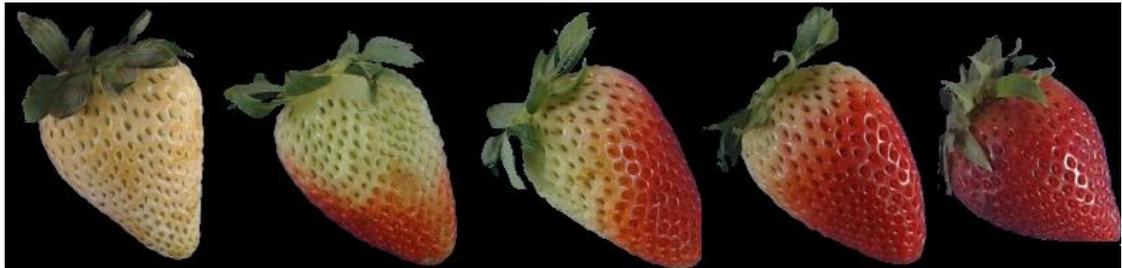


Figura 1 Proceso de maduración de la fresa.

1.5.2. Componentes de la imagen.

1.5.2.1. HSV.

El modelo de color HSV, ampliamente utilizado en computación e imágenes digitales, requiere una transformación no lineal desde el modelo RGB para obtener sus valores en coordenadas cilíndricas. Cuando se busca diferenciar objetos según su color, incluso una leve variación en el tono puede ayudar en esta distinción, por lo que se recomienda emplear una técnica para obtener el histograma de color 3D RGB. Esta técnica representa las distintas intensidades de color en los píxeles y puede aplicarse también en imágenes en escala de grises [22].

1.5.2.2. Brillo.

El brillo se refiere a la cantidad de luz en una imagen, este es calculado en función de la intensidad de los píxeles, donde valores de intensidad más altos indican áreas más brillantes y los valores bajos representan zonas más oscuras [23].

1.5.2.3. Textura.

La textura se refiere a los patrones y variaciones en la intensidad de los píxeles que reflejan la apariencia superficial de los objetos, como su rugosidad, suavidad, o estructura. La textura describe la disposición y repetición de tonalidades o colores en una región de la imagen, que pueden

incluir patrones regulares o complejos [24].

1.5.2.4. Características de matiz según Haralick.

Son un conjunto de medidas derivadas de la matriz de co-ocurrencia de niveles de gris, las cuales representan la forma en cómo varían las tonalidades de los píxeles en una imagen, a la vez describen texturas analizando patrones de repetición y relación entre tonos, como contraste, homogeneidad y correlación. Estas características pueden ser obtenidas usando la librería mahotas y la función haralick que genera un vector promedio de características de textura para la imagen img, lo cual permite representar la textura de la imagen en un único vector. Esto resulta útil para tareas de análisis y clasificación de imágenes [25].

1.5.3. Visión por computador

La visión por computador es la extracción de información de imágenes de manera automatizada, esta trata de imitar la función que realiza la vista humana para diseñar soluciones inteligentes, versátiles y dinámicas [26].

1.5.4. Inteligencia artificial

La inteligencia artificial es un programa de computadora diseñado para imitar las habilidades que solo la inteligencia humana posee, esta ayuda a que se puedan crear y aplicar algoritmos en entornos dinámicos de computación [27].

1.5.5. Teoría de clasificación

Es la clasificación de un objeto a la cual se le asigna una de varias categorías establecidas ya sean exhaustivas y exclusivas, donde no se le puede dar más de una o tampoco ninguna. [28].

1.5.6. Python

Python es un lenguaje de programación con una aplicación en diversas áreas como desarrollo de software, aplicaciones web, ciencia de datos y aprendizaje automático como transfer learning o machine learning [29].

1.5.7. Machine learning

Es el método científico que a partir del uso de un dispositivo con inteligencia computacional logra extraer relaciones entre datos y patrones. Luego estos patrones se pueden usar para la toma de decisiones y predecir comportamientos [30].

1.5.8. Deep learning

Se trata de una de las tecnologías más reconocidas del aprendizaje automático, esta posee algoritmos que son capaces de repetir o imitar las actividades que se producen dentro del cerebro humano, mediante redes neuronales artificiales, el deep learning entrena una computadora de tal forma que este logra reconocer patrones [31].

1.5.9. Aprendizaje de transferencia

Es una técnica que más sobresale dentro del Deep learning en el IA, esta técnica consiste en sacarle el mejor provecho posible a una gran cantidad de información la cual está relacionada a la resolución de un problema y darle uso en solucionar otro problema [32].

1.5.9.1. VGG-16

Propuesto por Karen Simonyan y Andrew Zisserman en 2013 pero presentado en el ILSVRC ImageNet Challenge en 2014. VGG16 es un modelo de red neural convolucional (CNN) usado comúnmente para la detección de objetos y clasificación de imágenes a gran escala [33].

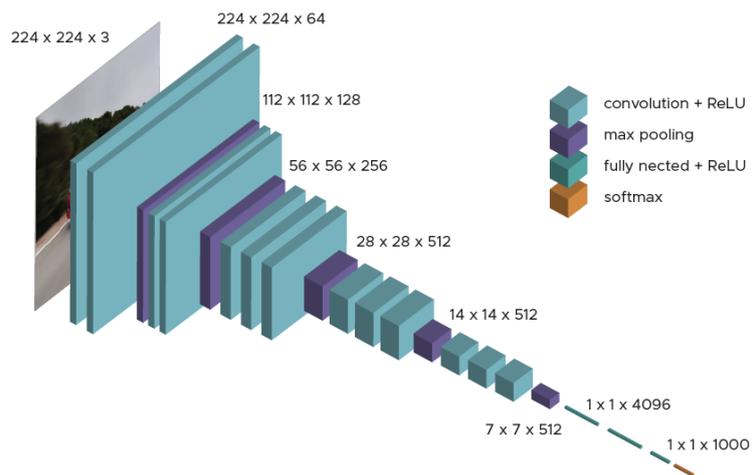


Figura 2 Representación gráfica de la arquitectura VGG16.

[33].

1.5.9.2. VGG-19

Es un modelo de red neuronal convolucional (CNN), entrenada previamente con más de un millón de imágenes que se encuentran almacenadas en el dataset de ImageNet con 19 capas de profundidad. En su estado pre entrenado es capaz de categorizar hasta 1000 categorías de objetos [34].

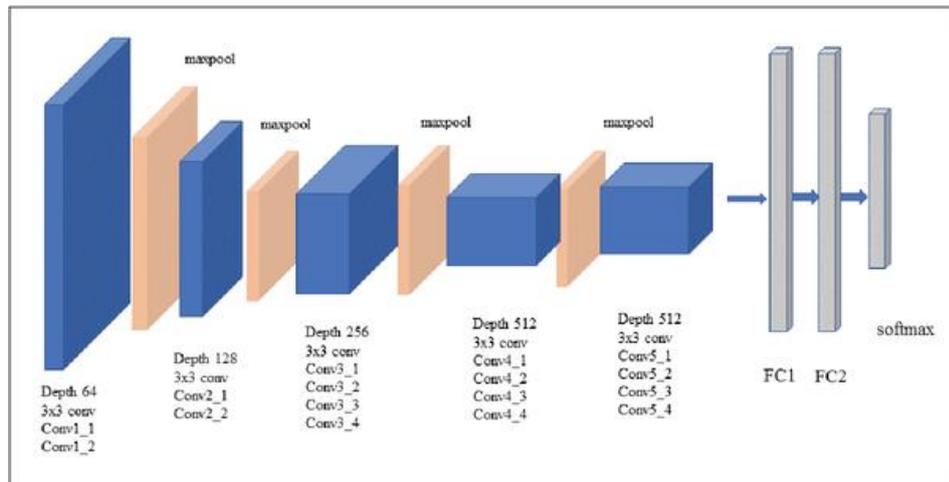


Figura 3 Representación gráfica de la arquitectura VGG19.

[35].

1.5.9.3. Resnet-50

Es un modelo de red neuronal convolucional (CNN), pre entrenada con más de un millón de imágenes que se encuentran almacenadas en la base de datos de ImageNet con 50 capas de profundidad. En su estado pre entrenado es capaz de categorizar hasta 1000 categorías de objetos [36].

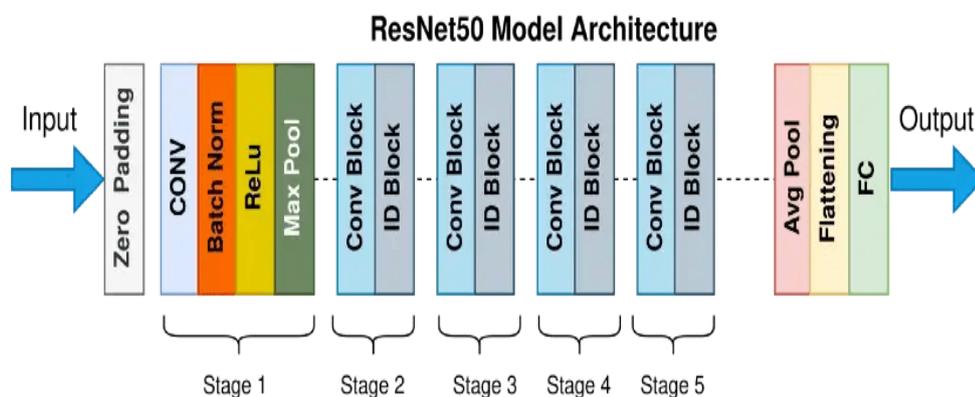


Figura 4 Representación gráfica de la arquitectura ResNet 50.

[37].

1.5.9.4. MobileNet V2

MobileNet V2 es una red neuronal convencional diseñada específicamente para la inferencia en dispositivos móviles. La estructura de MobileNet V2 consta de bloques de construcción llamados Módulos Inversos Residuales (Inverted Residual Modules), que se componen de capas convolucionales separables en profundidad, seguidas de una capa de convolución lineal. Estos módulos permiten una reducción significativa de los parámetros y del número de operaciones, sin comprometer el rendimiento [38].

1.5.9.5. AlexNet

AlexNet es una red neuronal convolucional profunda que consta de cinco capas convolucionales y tres capas completamente conectadas. La estructura de AlexNet fue diseñada para procesar imágenes de alta resolución, utilizando filtros convolucionales de tamaño grande para capturar características de nivel inferior y filtros convolucionales de tamaño pequeño para capturar características de nivel

superior [39].

1.5.10. Aprendizaje supervisado

El aprendizaje supervisado es un tipo de método en inteligencia artificial en el que un modelo aprende a hacer predicciones o clasificaciones basadas en datos etiquetados. Esto significa que, durante el entrenamiento, se le proporcionan ejemplos de entrada junto con sus respuestas correctas, lo que permite al modelo ajustar sus parámetros para predecir correctamente resultados futuros en datos similares. Es ampliamente utilizado en tareas como reconocimiento de imágenes, análisis de texto y predicción de tendencias [40].

1.5.10.1. SVM

Es un método de aprendizaje supervisado que, en el contexto de regresión (SVR), busca encontrar una función que se ajuste a los datos con un margen de error permitido. Su objetivo es minimizar el error mientras mantiene la complejidad del modelo baja, lo que ayuda a evitar el sobreajuste y mejora la generalización a nuevos datos [41].

1.5.10.2. Naive Bayes

Es un enfoque que utiliza principios Naive Bayes para modelar la relación entre variables. A diferencia de la regresión logística clásica, que proporciona estimaciones puntuales de los parámetros, el enfoque Naive Bayes trata los parámetros como variables aleatorias y permite incorporar información previa (prior) sobre ellos. Esto se traduce en distribuciones posteriores que reflejan tanto la información previa como la evidencia observada en los datos. Este método es útil para manejar la incertidumbre y proporciona intervalos de credibilidad para las predicciones, ofreciendo una visión más completa del modelo [42].

1.5.10.3. Regresión logística.

La regresión logística es un modelo estadístico usado para predecir la probabilidad de un resultado binario (cómo "sí/no") basado en una o más variables independientes. Convierte las variables en probabilidades entre 0 y 1, ideal para clasificar en dos categorías. Es ampliamente utilizada en áreas como medicina y marketing para identificar patrones y hacer predicciones categóricas [43].

1.5.10.4. Random Forest

Es un modelo que clasifica datos dividiéndolos en grupos basados en preguntas sobre sus características. Cada nodo representa una pregunta, y las ramas indican posibles respuestas. Esto permite identificar patrones complejos y no lineales en los datos, facilitando la toma de decisiones y predicciones [44].

1.5.11. Redes neuronales

Método que permite enseñar a las computadoras a procesar datos de la misma manera a como lo hace el cerebro humano. Este pertenece al proceso de aprendizaje profundo, el cual usa las neuronas interconectadas para transmitir información [45].

Las redes neuronales convolucionales (CNN) se destacan como un enfoque avanzado en el ámbito del aprendizaje profundo, permitiendo la inspección automatizada de daños en ingeniería mediante arquitecturas que procesan grandes volúmenes de datos de imágenes y ajustan parámetros específicos según las características de los daños [46].

Ecuación de una red neuronal:

$$a = \sum_{i=1}^D w_i x_i + w_0 y = g(a) = g \left(\sum_{i=1}^D (w_i x_i + w_0) \right)$$

Donde:

a = la suma ponderada de las entradas, también llamada activación de la neurona.

y = salida de la neurona.

g = función de activación o transferencia.

w_i = peso

x_i = entrada

w_0 = umbral o sesgo.

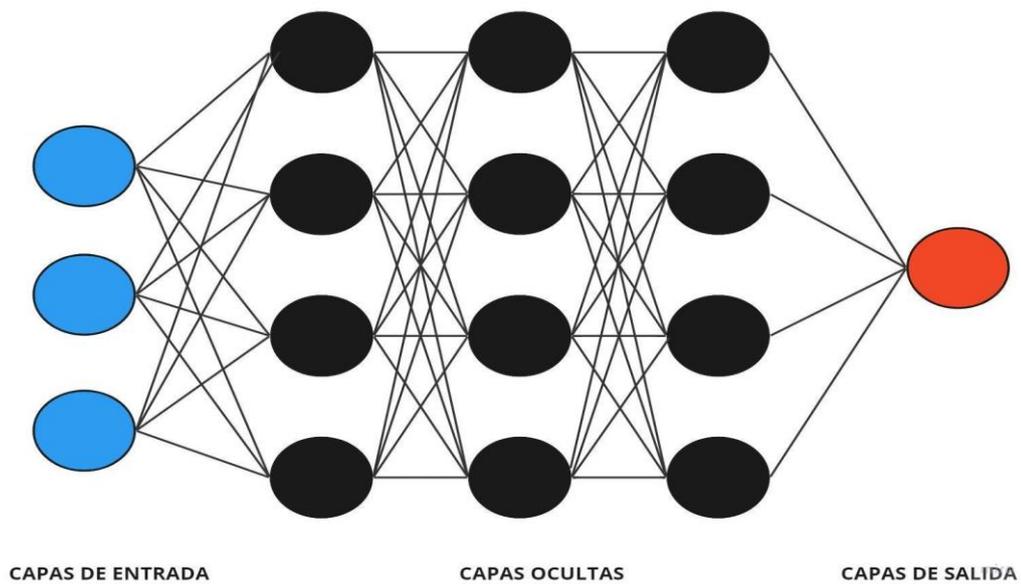


Figura 5 Ejemplo de redes neuronales.

1.5.12. Redes neuronales recurrentes

Las redes neuronales recurrentes (RNN) son un tipo de modelo que tienen conexiones hacia atrás, permitiendo que las neuronas recuerden los valores previos.

Debido a su alta carga de computación, entrenar estas redes ha sido un desafío, pero gracias a los avances tecnológicos recientes, su uso se ha popularizado y vuelto más accesible para la industria. La RNN más simple consiste en que una sola neurona recibe una entrada, produce una salida y se envía a sí misma en cada instante de tiempo. Por otro lado, una capa de neuronas recurrentes puede implementarse para que cada una reciba dos entradas: la entrada de la capa anterior y la salida anterior de la misma capa [47].

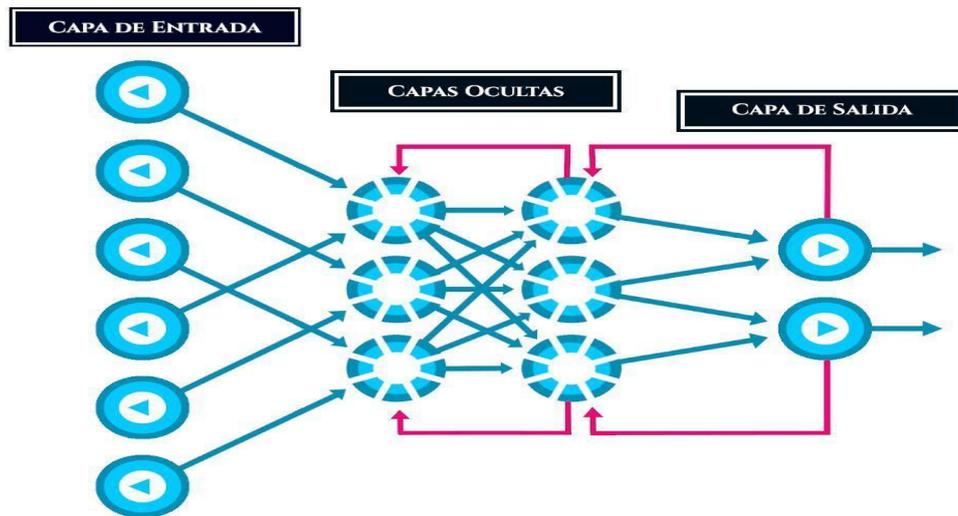


Figura 6 Ejemplo de redes neuronales recurrentes.

[48]

Cada neurona recurrente en una red neuronal tiene dos conjuntos de parámetros. Uno de ellos se utiliza para procesar la entrada de datos proveniente de la capa anterior, y el otro se aplica a la entrada de datos correspondiente al vector de salida del instante anterior. Durante la etapa de entrenamiento, los pesos de la matriz U , que afectan el estado de la red en el instante de tiempo anterior, también se actualizan utilizando el algoritmo de retropropagación (Backpropagation).[048].

$$Y_t = f(Wx_t + W_{y_{t-1}} + b)$$

Donde:

W = pesos de la matriz

f = función de activación

x_t = entrada actual

y_t = actuación actual

y_{t-1} = activación anterior

b = umbral o sesgo

1.5.13. Backpropagation

Backpropagation (o propagación hacia atrás de los errores) es un "método de cálculo que se utiliza en algoritmos diseñados para entrenar el funcionamiento de redes neuronales artificiales", y su objetivo es "minimizar los errores en el proceso de aprendizaje y mejorar la precisión de las predicciones ajustando los pesos y sesgos de cada nodo de la red". Además, el algoritmo de backpropagation se desarrolla en varias fases, que incluyen la elección del punto de entrada y de salida, la asignación de valores secundarios para modificar parámetros, la determinación del error total, la minimización del efecto del error en toda la red y el ajuste de los parámetros hasta que el error sea mínimo [49].

1.5.14. Multilayer perceptrons

Una red neuronal está conformada por diversas capas, lo cual da facilidad de solucionar inconvenientes que no son linealmente separables. Este puede estar o no conectado [50].

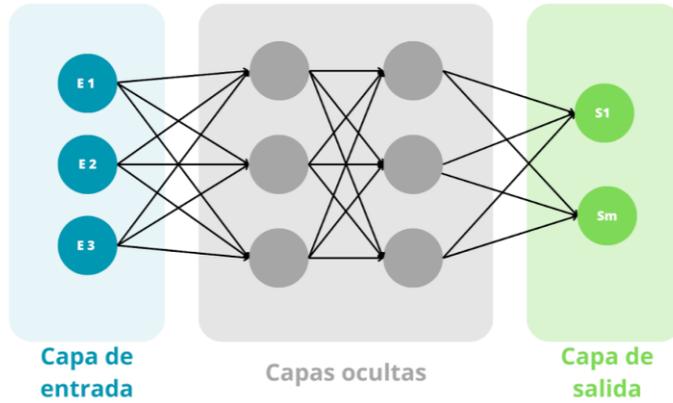


Figura 7 Ejemplo Multilayer perceptrons.

[50]

1.5.15. Capa convolucional

La capa convolucional tiene como función procesar la salida de neuronas que están conectadas a regiones locales. Además, estas son las encargadas de aplicar la convolución a las imágenes de ingreso para descubrir los patrones con las que se podrán clasificar [51].

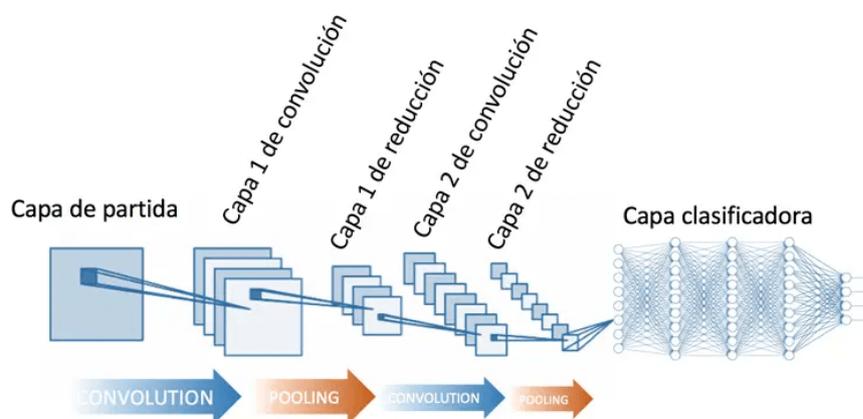


Figura 8 Ejemplo de Capa convolucional.

[52].

1.5.16. Capa Max Pooling

La capa Max pooling es de las más usadas debido que logra evitar el sobreajuste, al lograr una versión menos completa de la representación inicial [53], además van reduciendo el tamaño de los mapas de activación para facilitar la ejecución de los modelos en GPUs [51].

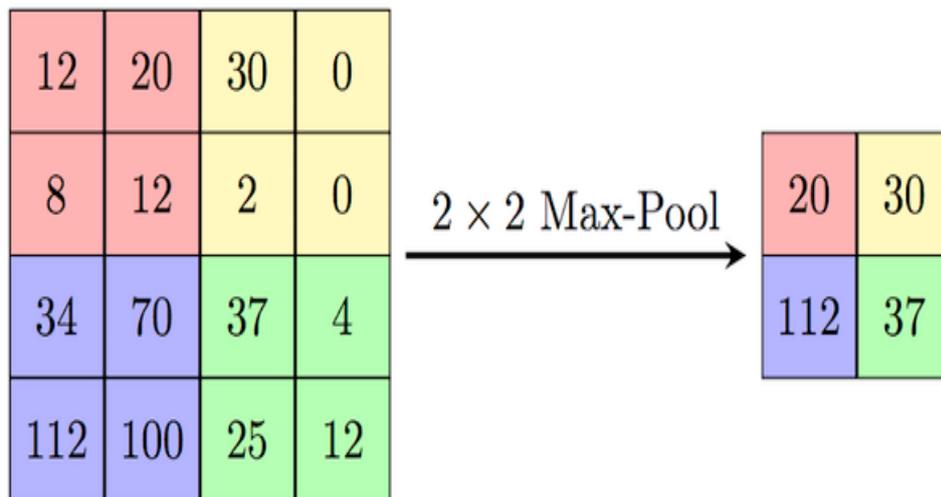


Figura 9 Ejemplo de Capa max pooling.

[54].

1.5.17. Capa densa

La capa densa o Fully-connected, es un conjunto de funciones no lineales dependientes. Donde una función individual está compuesta por una neurona o perceptrón, esta capa realiza una transformación lineal del vector de entrada por medio de una matriz de pesos para posteriormente realizar una transformación no lineal al producto por medio de una función de activación [55].

1.5.18. Capa de normalización

La capa de normalización son las que llevan a cabo operaciones sobre los

mapas de activación [56], entre las capas de normalización, la más usada es Batch Normalization [51], que asigna una transformación para mantener la salida media cerca de 0 y cercana a 1 la desviación estándar.

1.5.19. Función de activación

“La función de activación se ocupa de regresar una salida desde un costo de ingreso, comúnmente el grupo de valores de salida en un rango definido como (0,1) o (-1,1).” [57, párr.1].

1.5.19.1. Función Sigmoide

La función sigmoide es comúnmente utilizada en modelos de predicción de probabilidades debido a que su rango de salida está limitado a valores entre 0 y 1. Esto se debe a que la probabilidad de cualquier evento también se encuentra en este rango. Por lo tanto, la función sigmoide se utiliza en casos donde se necesite predecir la probabilidad como resultado [57].

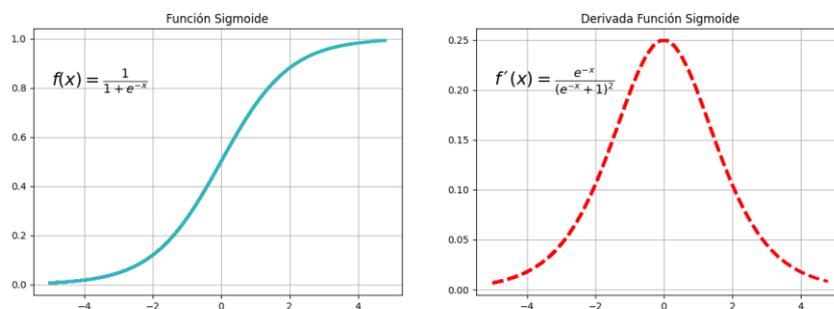


Figura 10 Gráfica de la ecuación de la función sigmoide en un plano cartesiano.

[58].

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

- Donde:
- σ =Sigma.
- e = Denota la constante exponencial.
- x = Suma ponderada de entrada.

1.5.19.2. Función Tangente hiperbólica o Gaussiana

La función que se describe es similar a la Sigmoide pero con una diferencia clave: en lugar de producir salidas en la escala de 0 a 1, produce salidas en la escala de -1 a +1. También es importante destacar que esta función es continua, lo que significa que produce resultados para todos los valores de x , no solo para valores discretos [57].

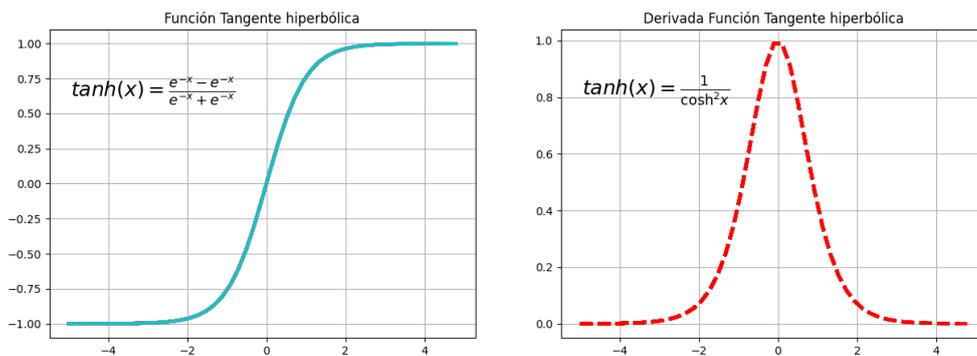


Figura 11 Gráfica de la función tangente hiperbólica en un plano cartesiano.

[58].

$$\tanh(x) = \frac{e^{-x} - e^x}{e^{-x} + e^x}$$

Donde:

\tanh = Tangente hiperbólica.

e = Denota la constante exponencial.

x = Valor de entrada.

1.5.19.3. Función RELU (Rectified Linear Unit)

La función de activación ReLU es muy común en el aprendizaje profundo, especialmente en las capas ocultas de redes neuronales convolucionales. Esta función es semirectificada, lo que implica que su salida es cero si la entrada es negativa y es igual a la entrada, si es mayor o igual a cero. Cabe destacar que no se aplica esta función en las capas de salida de la red neuronal. [58].

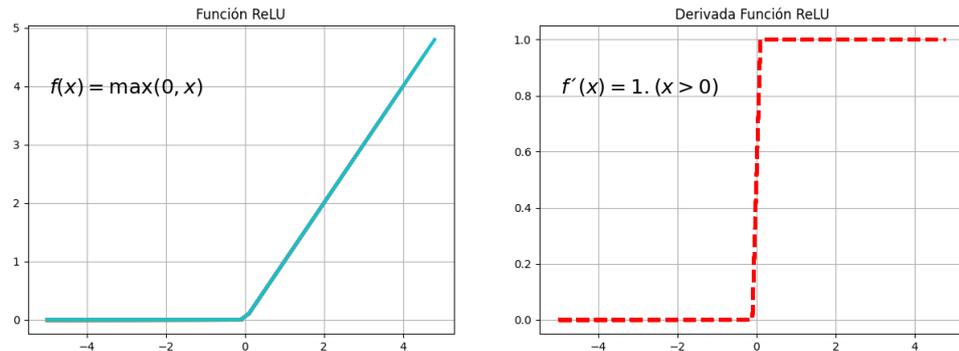


Figura 12 Gráfica de la función ReLU en un plano cartesiano.

[58].

$$LRelu(x) = \max(0.01 x, x)$$

Donde:

$$LRelu = Relu.$$

x = Valor de entrada.

1.5.19.4. Función Softmax

“Una función que convierte un vector de K valores reales en un vector de K valores reales que suman 1. Softmax transforma cualquier número en valores entre 0 y 1, para que puedan interpretarse como probabilidades.” [59, párr. 1]

1.5.20. Función de pérdida

“La función de pérdida mide la desviación entre el resultado observado y la predicción del algoritmo de aprendizaje automático.” [60, párr. 6]

Existen diferentes formas de calcular la función de pérdida, pero una de las más usadas es el error cuadrático medio (ECM)

$$ECM = \frac{1}{N} \sum_{i=1}^N (y_{pred_i} - y_{true_i})^2$$

Donde n es el número que se tiene de ejemplos para el entrenamiento, y_i es el valor verdadero de la salida y \hat{y}_i es el valor que la red neuronal predijo.

1.5.21. Algoritmos de optimización

Un algoritmo de optimización es una técnica de búsqueda cuyo objetivo es encontrar una solución a un problema de optimización tal que un conjunto en particular esté optimizado y pueda estar sujeto a muchas restricciones [61].

1.5.21.1. Adam

El método de Adam es un algoritmo poderoso y simple que puede buscar en la menor cantidad de tiempo, número de pasos o el método más conocido actualmente. Estas capacidades pertenecen al ámbito de la optimización y son un camino de conexión natural con las aplicaciones de IA cuyo objetivo es tomar la decisión correcta, lógica o sensata en el menor tiempo posible [62].

1.5.21.2. Adadelta

Adadelta es una variación del AdaGrad, en el que en lugar de escalar el factor de entrenamiento de cada dimensión, se restringe a una ventana fija del último gradiente de N . A la vez se enfoca en la disminución continua de la tasa de aprendizaje durante el entrenamiento y a la necesidad de una tasa de aprendizaje global seleccionada manualmente [63].

1.5.22. Métricas de rendimiento

1.5.22.1. Matriz de confusión

Es un medio para comprobar y visualizar el desempeño de un modelo de aprendizaje supervisado. En la matriz, las columnas muestran la predicción de las clases y las filas las instancias reales, en otras palabras los aciertos y errores del modelo luego de su entrenamiento [64].

Valores predichos	Verdadero positivo	Falso positivo
	Falso negativo	Verdadero negativo
	Valores reales	

Figura 13 Ejemplo de la matriz de confusión.

Verdadero positivo: Muestra la relación entre el valor real y la predicha cuando ambas son positivas. Representa una predicción correcta.

Verdadero negativo: Muestra la relación entre el valor real y la predicha cuando ambas son negativas. Representa una predicción correcta.

Falso negativo: Se da cuando el predicho es negativo y el valor real es positivo. Representa una predicción incorrecta.

Falso positivo: Se da cuando el predicho es positivo y el valor real es negativo. Representa una predicción incorrecta.

1.5.22.2. Precisión

“Representa por la proporción de verdaderos positivos dividido entre todos los resultados positivos (tanto verdaderos positivos, como falsos positivos)” [65, párr. 5]

$$VP/(VP+FP)$$

1.5.22.3. Exactitud

“Este método mide el porcentaje de veces que el modelo planteado ha acertado correctamente” [65, párr. 15].

$$(VP+VN)/(VP+FP+FN+VN)$$

1.5.22.4. Sensibilidad o Recall o Exhaustividad

Representa el porcentaje de casos positivos que el algoritmo de clasificación logró identificar correctamente [65].

$$VP/(VP+FN)$$

1.5.22.5. F1 score

Se utiliza para combinar medidas de precisión y recuperación únicamente costo. Esto significa que es más fácil equiparlo con un rendimiento que combina precisión e integridad en diferentes resoluciones, por lo que está optimizado [65].

$$F1 = \frac{2 * Precisión * Sensibilidad}{Precisión + Sensibilidad}$$

1.5.23. Procesamiento digital de las imágenes

Se trata de una alteración visual de una imagen con el fin de aislar algunas características particulares de la imagen. También se puede lograr el reconocimiento de objetos, todo esto con ayuda de una computadora digital [66].

1.5.23.1. Formación de la imagen

Una imagen se crea cuando se estimula el sensor de una cámara fotográfica por donde los rayos del sol interactúan con un objeto, creando una imagen bidimensional estática, dando así una imagen analógica o continua donde se puede visualizar un paisaje u objeto [67].

1.5.23.2. Canales de color

Las imágenes están compuestas de píxeles, los que a su vez están compuestos por combinaciones de colores de diferentes tonos, estos pueden ser a escalas grises que tiene un solo canal pero de diferentes niveles de gris, o entre los diferentes tipos en color el RGB que tiene canales de color azul, verde y rojo. Además

los canales de las imágenes pueden ir más allá del espectro visual en imágenes multiespectrales e hiperespectrales, donde sus canales corresponden a longitudes de onda y que tienen información espectroscópica [68].

1.5.23.3. Extracción de fondo

Al procesar una imagen se realiza un recorrido por cada píxel que la componen, para lograr distinguir los píxeles que pertenecen al fondo con los píxeles que pertenecen al objeto se puede realizar un Thresholding, en otras palabras, resta de fondo. En este procedimiento, binarizar la imagen e intentamos obtener un costo umbral que pueda separar correctamente el objeto que se dividirá del fondo [69].

1.5.23.4. Método Otsu

Es un método que realizará un análisis discriminante, es usado para realizar la umbralización de imágenes de forma automática; por otro lado este método ayudará a seleccionar los umbrales maximizados de la manera correcta [70]

1.5.23.5. Interpolación

Se trata de una de las operaciones más importantes al momento de realizar procesamiento de imágenes digitales, debido a que este proceso mejora la resolución temporal o espacial de los datos que posee una imagen, durante la modificación del tamaño de una imagen [71].

1.5.23.6. Detección de bordes

Es la comunicación entre dos regiones de diferentes niveles de grises, a su vez en la detección de bordes se puede detectar por los cambios de intensidad de luz que posee la imagen, esto contribuirá a determinar el tamaño, orientación, profundidad y propiedades del objeto a analizar [10].

1.5.24. Paradigmas

El concepto de paradigma generalmente se usa como sinónimo de "ejemplo" o cuando se considera "modelo". Pero a la vez este también se enfoca en ser una estilo o forma de pensar, lo que se compila no solo gracias a la acción interna de una comunidad especializada en particular, sino también en las competiciones en la opinión pública que se llama, es decir, grupos, profesionales o no, que están fuera de dicha comunidad [72].

1.5.24.1. Paradigma conexionista.

El uso de este paradigma se centra en la red de la neurona artificial, donde los problemas de expresión de conocimiento tienen lugar utilizando la entrada de red y el número de filas numéricas etiquetadas e inferir. Los valores de estos parámetros se ajustan mediante monitoreo o algoritmos de aprendizaje no supervisados [73].

2. MATERIALES Y MÉTODO

2.1. Tipo y Diseño de Investigación

2.1.1. Tipo de Investigación.

Esta investigación es de tipo cuantitativa dado que fueron utilizados datos numéricos, los cuales se procesaron para realizar un análisis estadístico el cual ayudó a definir un valor a las variables propuestas en la operacionalización.

2.1.2. Diseño de Investigación.

Diseño cuasi experimental, debido a que se buscó identificar el tipo de relación que tiene una variable independiente sobre una variable dependiente y qué resultado

se origina. Debido a lo expuesto previamente, esta investigación se considera cuasi experimental, esto debido a que los modelos de aprendizaje automático influyen directamente en la precisión de la clasificación del nivel de madurez de las fresas.

2.2. Variables, Operacionalización

En la operacionalización de las variables, se presentó el proceso de transformación de la variable (conceptual) a una Operativa, a través de indicadores que permitirán cuantificar la variable:

Tabla 1 Operacionalización de variables

Variable de estudio	Definición conceptual	Definición operacional	Dimensiones	Indicadores	Ítems	Instrumento	Valores finales	Tipo de variable	Escala de medición
Modelos de machine learning	Son salidas de la información que se genera en el entrenamiento del algoritmo de aprendizaje automático con datos.	Desempeño	Consumo de recursos	Promedio de tiempo de respuesta	$Tr = \sum_j^n \frac{tfj - tfi}{n}$	Registro electrónico	Tiempo de respuesta de cada modelo.	Cuantitativa discreta	Bajo: > 60% Medio: > 80% Alto: > 90%
				Grado de consumo de CPU	$Uc = \sum_j^n uc_j / n$		El total de consumo de CPU.		Bajo: > 0.6 Medio: > 0.8 Alto: > 0.9
				Grado de consumo de memoria RAM	$Cm = \sum_j^n cm_j / n$		El total de consumo de memoria RAM.		Bajo: > 60% Medio: > 80% Alto: > 90%
Clasificar el nivel de madurez en las fresas	Trata de identificar todas las fresas según el índice de madurez: madurez fisiológica, madurez comercial, madurez de consumo y sobre maduración.	Eficacia	Exactitud	Precisión	$R = \frac{TP}{TP + FN}$	Ficha de observación	La cantidad de elementos identificados correctamente como positivos.	Cualitativa categórica	Bajo: > 0.6 Medio: > 0.8 Alto: > 0.9
				Exactitud	$P = \frac{TP}{TP + FP}$		Un porcentaje total de elementos clasificados correctamente.		Bajo: > 0.6 Medio: > 0.8 Alto: > 0.9
				Recall	$recall = \frac{VP}{VP + FN}$		La cantidad que el modelo de machine learning es capaz de identificar correctamente.		Bajo: > 0.6 Medio: > 0.8 Alto: > 0.9

2.3. Población de estudio, muestra, muestreo y criterios de selección

2.3.1. Población

La población está determinada por 11 modelos de aprendizaje de transferencia y 5 modelos de aprendizaje supervisado que se revisaron en los trabajos previos. Como se muestra en la tabla del anexo 01.

2.3.2. Muestra

La muestra fue determinada de manera no probabilística por conveniencia, está conformada por 5 modelos de aprendizaje por transferencia que son VGG 16, VGG 19, ResNet 50, MobileNet V2 y AlexNet, y por 4 modelos de aprendizaje supervisado los cuales son Regresión logística, Random Forest, Support vector machine (SVM) y Naive Bayes.

El criterio que fue utilizado para la selección de estos modelos fue su desempeño en la clasificación de madurez en frutas y verduras, como se muestra en la tabla del anexo 01.

2.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad

2.4.1. Abordaje metodológico

El abordaje metodológico para esta tarea de investigación fue cuasi experimental debido a que se buscó ver qué variaciones de la variable independiente (modelos de aprendizaje por transferencia y supervisado) producen variaciones en la variable dependiente (precisión para la clasificación de madurez en las fresas).

2.4.2. Técnicas de recolección de datos

2.4.2.1. Registro electrónico

Este instrumento nos permitió registrar los resultados que se obtuvieron con la precisión, rendimiento, recuperación y exactitud. Con la finalidad de que al momento que comparamos los resultados de los modelos, este se realice en el menor tiempo posible.

2.4.2.2. Bitácora de resultados

Este de un archivo paginado de uso obligatorio para el curso por medio del cual se prueba los requerimientos necesarios para el desarrollo de la indagación, conteniendo las visualizaciones, las ideas, los datos, los convenios, los adelantos y los obstáculos provocados en cada junta a lo largo de la ejecución del trabajo de indagación.

2.5. Procedimiento de análisis de datos

2.5.1. Promedio de tiempo de respuesta

Es el valor del tiempo que demora un modelo entrenado para detectar y clasificar el nivel de madurez en las fresas en una imagen digital.

$$Tr = \sum_j^n \frac{tfj - tfi}{n}$$

Donde:

Tr: Es el promedio de tiempo de clasificación de un modelo previamente entrenamos.

Tf_j : Es el tiempo final de clasificación de un modelo previamente entrenamos.

Tf_i : Es el tiempo inicial de clasificación de un modelo previamente entrenamos.

n : Es el total de pruebas

2.5.2. Grado de consumo de CPU

Representa el consumo promedio de recursos CPU durante la detección y clasificación del modelo de machine learning.

$$Uc = \sum_j^n uc_j / n$$

Donde:

Uc: uso del CPU durante la ejecución del modelo de machine learning.

n: número de ejecuciones ocurridas durante la detección y clasificación con un modelo de machine learning.

2.5.3. Grado de consumo de memoria RAM

Representa el consumo de la memoria RAM durante el uso del modelo de machine learning para detectar y clasificar el nivel de madurez en las fresas.

$$Cm = \sum_j^n cm_j / n$$

Donde:

Cm **Grado** de consumo de memoria RAM durante la ejecución del modelo de machine learning.

Cm_j **Grado** de consumo de memoria en la prueba j para detectar y clasificar el nivel de madurez en las fresas.

n : cantidad total de pruebas.

2.5.4. Matriz de confusión

Es una herramienta utilizada en el aprendizaje supervisado para ver el desempeño de un algoritmo, donde las columnas de la matriz representan el número de precisiones de una clase y las filas las instancias de la clase real, en otras palabras ver cuáles fueron los aciertos y errores que tuvo el modelo durante el periodo de aprendizaje. (Barrios Arce, 2019)

2.5.5. Exactitud

Mide la cantidad de predicciones positivas que fueron correctas.

$$P = \frac{TP}{TP + FP}$$

Donde:

P : Es la exactitud.

TP: Es verdadero positivo

FP: Es falso positivo

2.5.6. Precisión

Mide la calidad del modelo en la clasificación del nivel de madurez en las fresas.

$$R = \frac{TP}{TP + FN}$$

Donde:

R : Es la Precisión

TP: Es verdadero positivo

FN: Es falso negativo

2.5.7. Recall

Calcula la exhaustividad, la cual muestra la cantidad que el modelo es capaz de identificar.

$$recall = \frac{VP}{VP + FN}$$

Donde:

recall : Es el recall

VP: Es verdadero positivo

FN: Es falso negativo

2.6. Criterios éticos

Se tomó el criterio ético del principio ético de la propiedad intelectual, dado que se realizó las debidas citas y referencias a los papers, libros y trabajos de investigación. Con los cuales se logró obtener un conocimiento base para poder desarrollar el presente proyecto de investigación.

3. RESULTADOS Y DISCUSIÓN

3.1. Resultados

En esta sección se presentan los resultados obtenidos de la evaluación de los métodos comparados para la clasificación de madurez en fresas. Su implementación y ejecución se realizó en la plataforma colaborativa Google Collaboratory, que contiene un entorno adecuado para implementar modelos de machine learning usando el lenguaje de programación Python.

Los conjuntos de tipos de datos fueron, fresas verdes, maduras y muy maduras. Divididos de forma aleatoria en entrenamiento (70%), prueba o test (20%) y validación (10%). Se compararon los modelos de aprendizaje por transferencia como VGG 16, VGG 19, Resnet-50, MobileNet v2, AlexNet y los modelos de aprendizaje supervisado como Regresión logística, Random Forest, Support vector machine (SVM) y Naive Bayes.. En diferentes métricas que evalúan su rendimiento y consumo de recursos.

3.1.1. Promedio de tiempo de respuesta.

3.1.1.1. Aprendizaje por transferencia.

En un periodo de entrenamiento de 150 épocas con cada modelo de clasificación, repetido 5 veces para obtener un valor promedio más fiable del tiempo de respuesta del entrenamiento. Obteniendo como resultado que el modelo que tomó más tiempo en su entrenamiento fue el modelo VGG 19 con 264.76 segundos y el que logró el menor tiempo fue mobileNet v2 con 101.2 segundos.

Tabla 2 Resultados del indicador promedio de tiempo de respuesta para aprendizaje por transferencia.

Modelo	Tiempo (segundos)
VGG 16	116.53
VGG 19	264.76
ResNet 50	166.55
MobileNet v2	101.20
AlexNet	150.42



Figura 14 Gráfico de barras para el indicador de promedio de tiempo de respuesta para aprendizaje por transferencia.

3.1.1.2. Aprendizaje supervisado.

En la evaluación del tiempo de respuesta de los modelos de aprendizaje supervisado, el modelo que tuvo el mayor tiempo fue SVM con 7.7546s y el menor tiempo lo tuvo el modelo random forest con un tiempo de 0.21s.

Tabla 3 Resultados del indicador promedio de tiempo de respuesta para aprendizaje supervisado.

<i>Modelo</i>	<i>Tiempo (segundos)</i>
---------------	--------------------------

<i>Regresión Logística</i>	<i>0,0748</i>
<i>Random Forest</i>	<i>0,21</i>
<i>SVM</i>	<i>7,7546</i>
<i>Naive Bayes</i>	<i>0,0054</i>

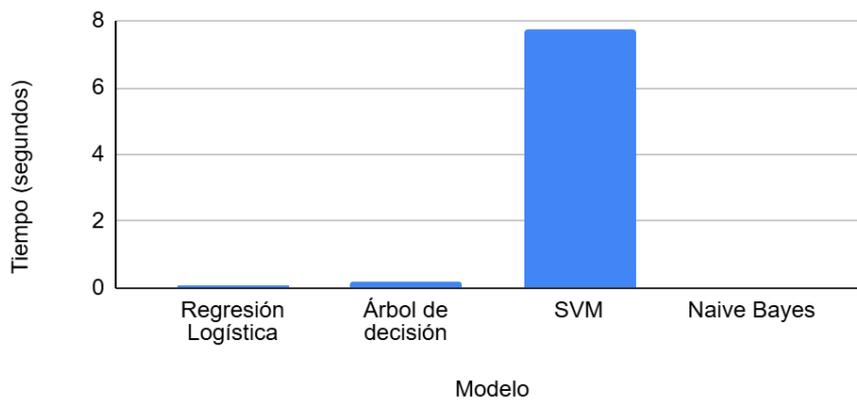


Figura 15 Gráfico de barras para el indicador de promedio de tiempo de respuesta para aprendizaje supervisado.

3.1.2. Grado de consumo de CPU.

3.1.2.1. Aprendizaje por transferencia.

El consumo de CPU al final del proceso de entrenamiento de los modelos se muestra en la **Tabla 3**. Esta tabla muestra que el modelo ResNet-50 presenta el mayor consumo de CPU, con un valor de 50.2, mientras que el modelo VGG-16 registra el menor consumo, con 33.8. Este resultado sugiere que ResNet-50 requiere más recursos computacionales en comparación con VGG-16, lo que podría estar relacionado con la complejidad y el tamaño de la red en el modelo ResNet-50.

Tabla 4 Resultados del indicador grado de consumo de CPU para aprendizaje por transferencia.

Modelo	CPU
VGG 16	33.8

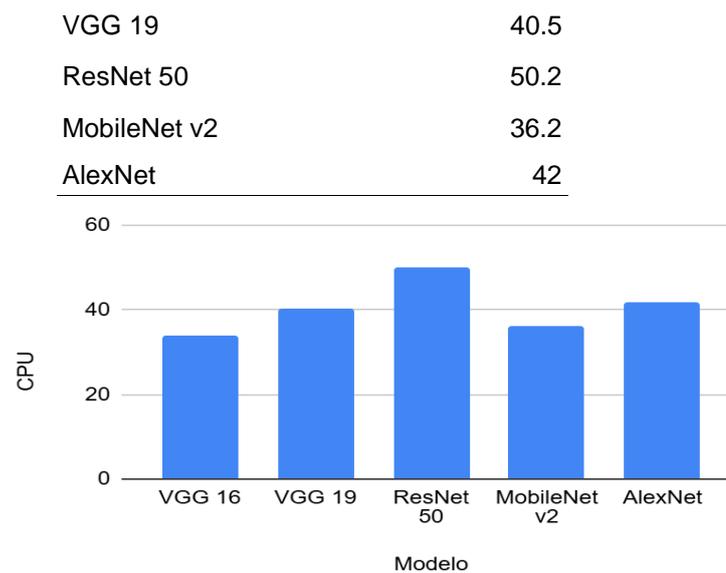


Figura 16 Gráfico de barras para el indicador grado de consumo de CPU para aprendizaje por transferencia.

3.1.2.2. Aprendizaje supervisado.

El consumo de CPU entre los modelos de aprendizaje supervisado como se muestra en la tabla 5, el mayor lo obtuvo Naive Bayes con un consumo de 95.1 y el menor consumo lo tuvo SVM con 69.7.

Tabla 5 Resultados del indicador grado de consumo de CPU para aprendizaje supervisado.

Modelo	CPU
Regresión Logística	94,2
Random Forest	71,5
SVM	69,7
Naive Bayes	95,1

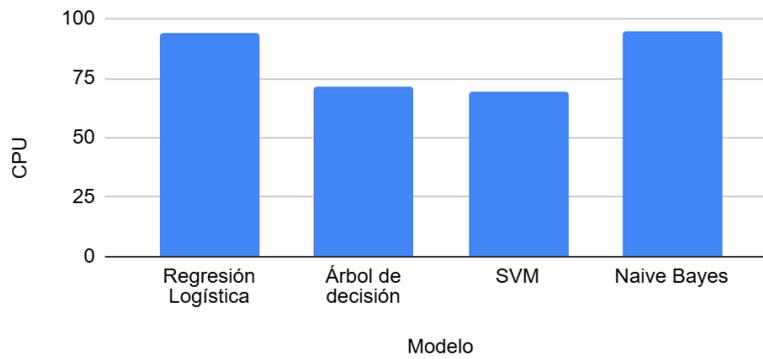


Figura 17 Gráfico de barras para el indicador grado de consumo de CPU para aprendizaje supervisado.

3.1.3. Grado de consumo de memoria RAM.

3.1.3.1. Aprendizaje por transferencia.

El consumo de RAM al término del entrenamiento de los modelos se observa en la tabla 3, donde el modelo VGG 16 tiene el mayor valor con 5.1 y el modelo VGG 19 tiene el menor con 4.0.

Tabla 6 Resultados del indicador grado de memoria RAM para aprendizaje por transferencia.

Modelo	RAM
VGG 16	5.10282135
VGG 19	4.087127686
ResNet 50	4.253105164
MobileNet v2	4.55096817
AlexNet	5.052055359

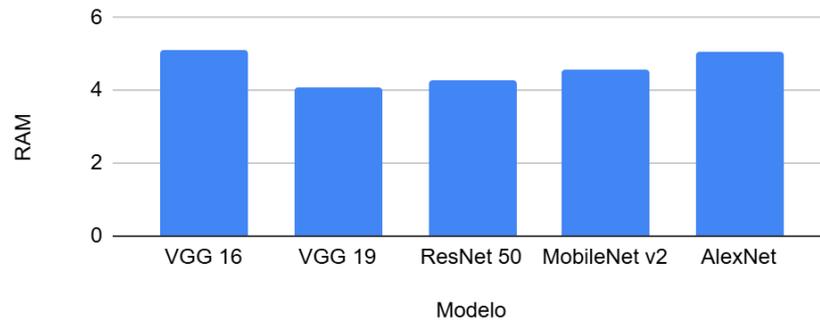


Figura 18 Gráfico de barras para el indicador grado de memoria RAM para aprendizaje por transferencia.

3.1.3.2. Aprendizaje supervisado.

En la evaluación del consumo de RAM que se muestra en la tabla 7, los modelos tuvieron un consumo muy similar, teniendo unas bajas diferencias en la medición. Siendo el modelo Naive Bayes con 1.0792 y el menos SVM con 1,0574.

Tabla 7 Resultados del indicador grado de memoria RAM para aprendizaje supervisado.

Modelo	RAM
Regresión Logística	1,0397
Random Forest	1,074
SVM	1,0574
Naive Bayes	1,0792

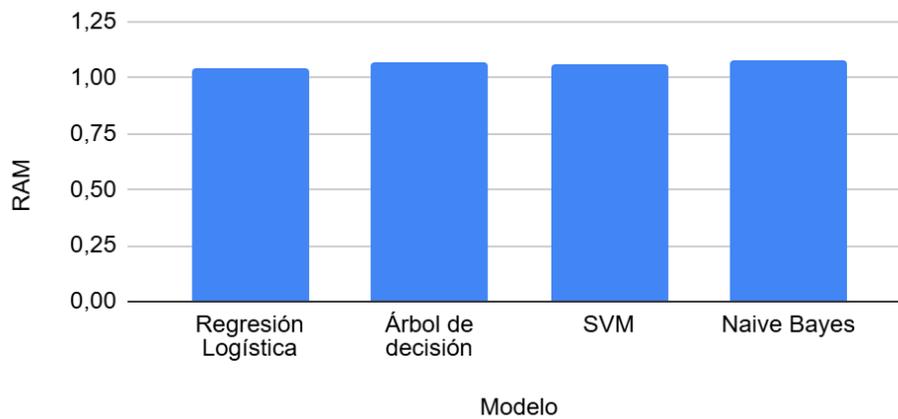


Figura 19 Gráfico de barras para el indicador grado de memoria RAM para aprendizaje supervisado.

3.1.4. Precisión.

3.1.4.1. Aprendizaje por transferencia.

Como se observa en la tabla 5 el modelo que dio excelentes resultados en el indicador precisión fue el modelo ResNet 50 obteniendo para las fresas verdes y maduras una precisión del 100% y un 93% para las muy maduras con la data de validación, seguido de los modelos VGG 19, MobileNet V2 y VGG 16, sin embargo el modelo AlexNet muestra un bajo nivel de precisión para lograr clasificar las fresas según las etiquetas establecidas.

Tabla 8 Resultados del indicador de precisión para aprendizaje por transferencia.

Modelo	Precisión		
	Maduro	Verde	Dañada
VGG 16	0.88	0.94	1
VGG 19	0.82	1	1
ResNet 50	1	1	0.93

MobileNet v2	0.79	0.75	1
AlexNet	0.53	0.45	0.46

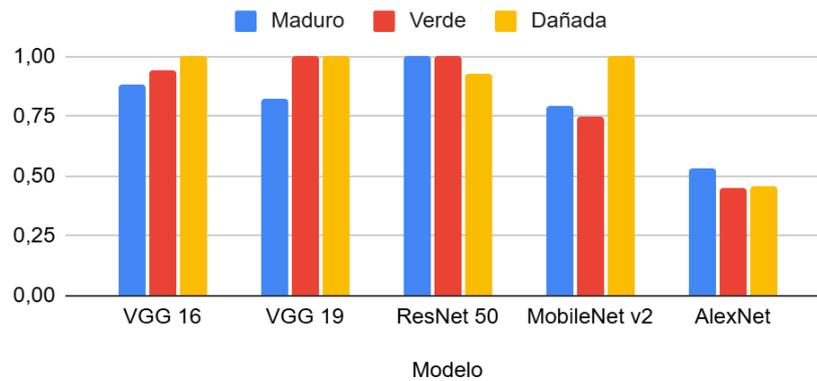


Figura 20 Gráfico de barras para el indicador de precisión para aprendizaje por transferencia.

3.1.4.2. Aprendizaje supervisado.

Como se observa en la tabla 9, los modelos de aprendizaje supervisado tuvieron en general una precisión elevada, siendo los modelos de regresión logística y Random Forest quienes lograron una precisión del 100%, por otro lado Naive bayes presentó problemas en clasificar las fresas verdes, pero obtuvo buenos resultados para las etiquetas maduro y dañadas.

Tabla 9 Resultados del indicador de precisión para aprendizaje supervisado.

Modelo	Precisión		
	Maduro	Verde	Dañada
Regresión Logística	1	1	1
Random Forest	1	1	1
SVM	1	0,9	0,93
Naive Bayes	0,95	0,71	1

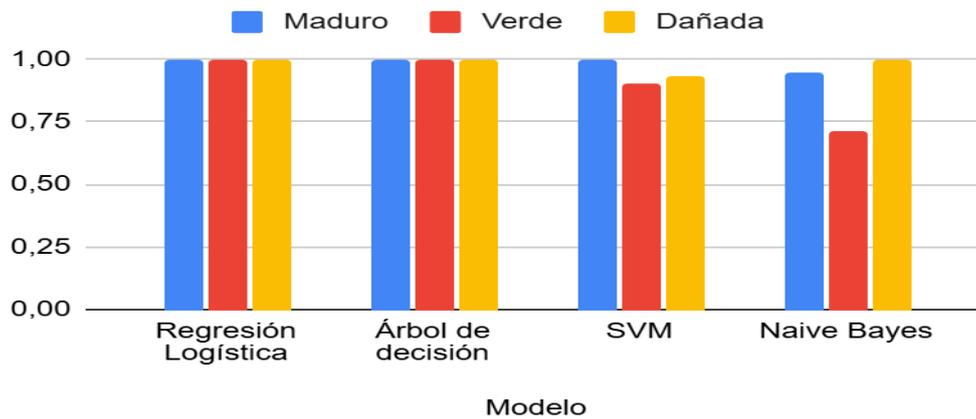


Figura 21 Gráfico de barras para el indicador de precisión para aprendizaje supervisado.

3.1.5. Exactitud.

3.1.5.1. Aprendizaje por transferencia.

Como se puede observar en la tabla 6 el modelo ResNet 50 logra el mejor resultado, obteniendo 1 y 0.9531 en la exactitud validada, seguido de los modelos VGG|6, VGG19 y Mobile Net V2, por otro lado el modelo AlexNet muestra un sobreajuste muy elevado en su exactitud de entrenamiento con respecto a la exactitud validada.

Tabla 10 Resultados del indicador de exactitud para aprendizaje por transferencia.

Modelo	accuracy	val_accuracy
VGG 16	1	0.7857
VGG 19	0.9495	0.9302
ResNet 50	1	0.9531
MobileNet v2	0.9495	0.8594

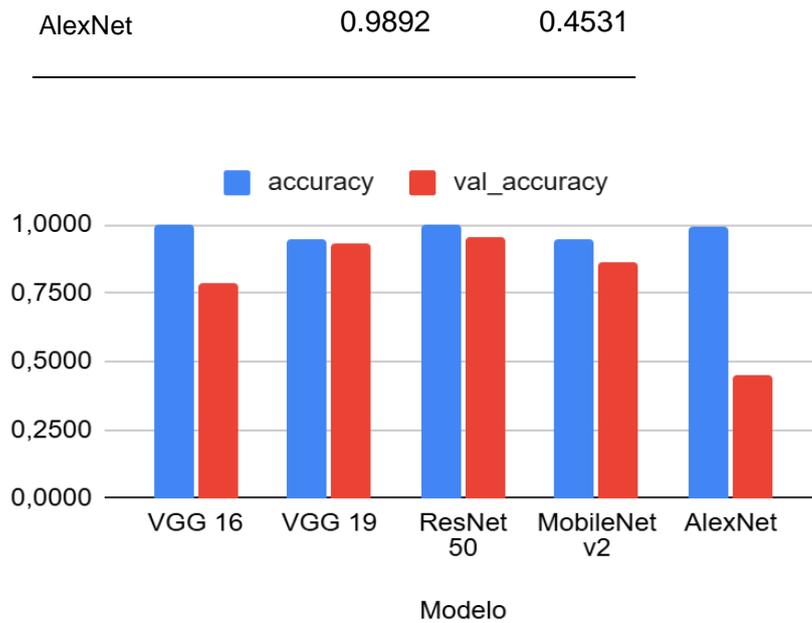


Figura 22 Gráfico de barras para el indicador de exactitud para aprendizaje por transferencia.

3.1.5.2. Aprendizaje supervisado.

En la evaluación de exactitud mostrada en la tabla 11, se obtuvo que el modelo Random Forest logró el mejor resultado con 0.9663, por otro lado el modelo que obtuvo el menor resultado fue Naive Bayes con 0.7528.

Tabla 11 Resultados del indicador de exactitud para aprendizaje supervisado.

Modelo	accuracy	val_accuracy
Regresión Logística	0,9742	0,9551
Random Forest	0,9968	0,9663
SVM	0,9645	0,9213
Naive Bayes	0,7516	0,7528

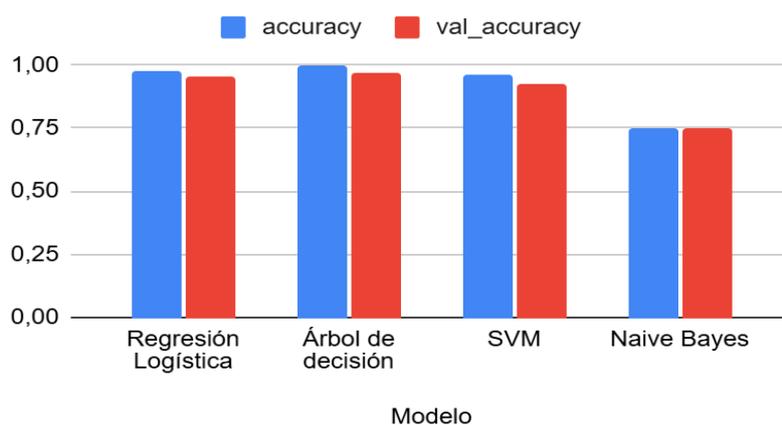


Figura 23 Gráfico de barras para el indicador de exactitud para aprendizaje supervisado.

3.1.6. Recall.

3.1.6.1. Aprendizaje por transferencia.

Como se puede observar en la tabla 7 los modelos que dieron excelentes resultados en el indicador Recall fueron los modelos VGG 16 y ResNet 50 con un valor de 1 en dos etiquetas, seguido de los modelos MobileNet V2 y VGG 19, a la vez se puede observar que el modelo AlexNet no logró un buen resultado en recall para las etiquetas establecidas.

Tabla 12 Resultados del indicador recall para aprendizaje por transferencia.

Modelo	Recall		
	Madura	Verde	Dañada
VGG16	1	1	0.79
VGG19	1	0.87	0.93
Resnet 50	1	0.93	1
Mobile Net v2	0.79	0.75	1
AlexNet	0.53	0.45	0.46

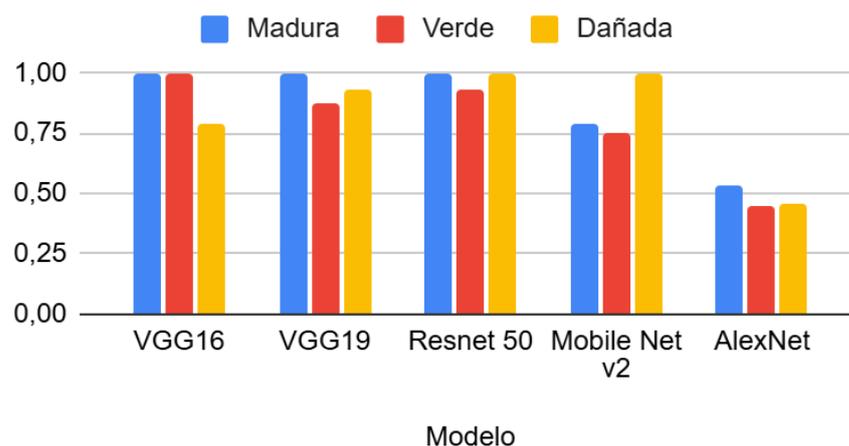


Figura 24 Gráfico de barras para el indicador recall para aprendizaje por transferencia.

3.1.6.2. Aprendizaje supervisado.

Los resultados obtenidos en el recall que se muestran en la tabla 13, muestran que los modelos de regresión logística y random forest logran excelentes resultados para clasificar las fresas según las etiquetas de madurez, siendo Naive bayes quien logra el menor resultado, debido a su bajo porcentaje en clasificar las fresas dañadas.

Tabla 13 Resultados del indicador recall para aprendizaje supervisado.

Modelo	Recall		
	Madura	Verde	Dañada
Regresión Logística	1	1	1
Random Forest	1	1	1
SVM	1	0,9	0,93
Naive Bayes	1	1	0,64

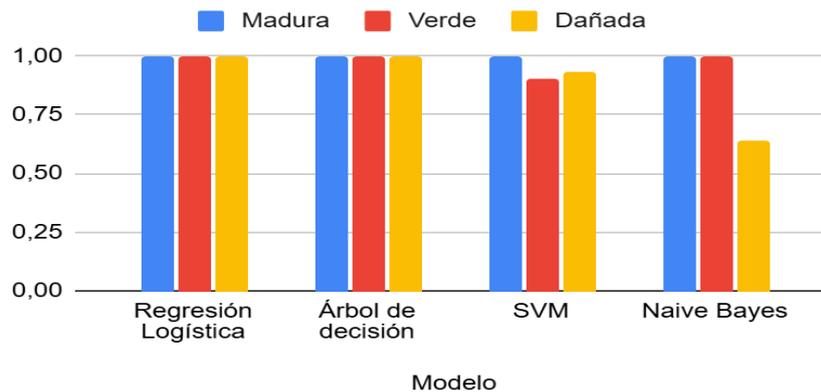


Figura 25 Gráfico de barras para el indicador recall para aprendizaje supervisado.

3.2. Discusión

Luego de la implementación de los modelos de aprendizaje por transferencia y supervisado. La revisión de los resultados, se pudo observar que de los 5 modelos elegidos ResNet 50 logra excelentes resultados para clasificar fresas en las etiquetas maduro, verde y muy maduras. De los otros modelos VGG 16 y VGG 19 también logran resultados positivos pero tienden a disminuir la exactitud con los datos de validación. Por parte, entre los 4 los modelos de aprendizaje supervisado, los modelos Random Forest y regresión logística logran excelentes resultados para clasificar las fresas por sus etiquetas de madurez.

Entre los modelos de aprendizaje por transferencia, el modelo ResNet 50 logró los mejores resultados en exactitud con 95.31% y recall de 100% para cada una de las etiquetas maduro, verde y dañadas, obtenido los mejores resultados de precisión al ser probado con 43 imágenes de validación con valores de 100%, 100% y 97% respectivamente en las categorías. Siendo

ResNet 50 quien demostró un alto rendimiento con estas imágenes, Los modelos VGG 16 y 19 logran obtener resultados cercanos obteniendo 100%, 87%, 93% y 100% 100% y 79% respectivamente en las categorías de madurez de la fresa. Por otro lado, entre los modelos de aprendizaje supervisado, los modelos de regresión logística y random forest logran un 100% de precisión para clasificar las fresas según las etiquetas de madurez.

Aunque uno de los modelos de aprendizaje por transferencia implementados logra bajos resultados, esto puede deberse a que el modelo Alexnet no viene pre entrenado como los otros modelos. Además para lograr una comparación similar entre los modelos se repitieron las mismas capas y parámetros en el entrenamiento, por lo cual sería más indicado buscar la combinación óptima de estos según cada modelo para lograr un mejor rendimiento y logren clasificar las fresas según las etiquetas establecidas con una mayor precisión.

Aunque los modelos de aprendizaje por transferencia se han hecho populares por estar pre entrenados y no necesitar grandes cantidades de datos, ni una extracción de características previa, debido que las redes convolucionales se encargan de realizar esto. Un dataset con características que influyan en la clasificación deseada puede ayudar a mejorar significativamente el nivel de precisión de los modelos. Por otra parte, los modelos de aprendizaje supervisado tienen un mejor tiempo de entrenamiento, lo que es otro punto a favor.

3.3. Aporte de la investigación

Se desarrollaron 2 métodos para la clasificación por madurez de la fresa utilizando machine learning, el primer método basado en aprendizaje por transferencia y el segundo el aprendizaje supervisado.

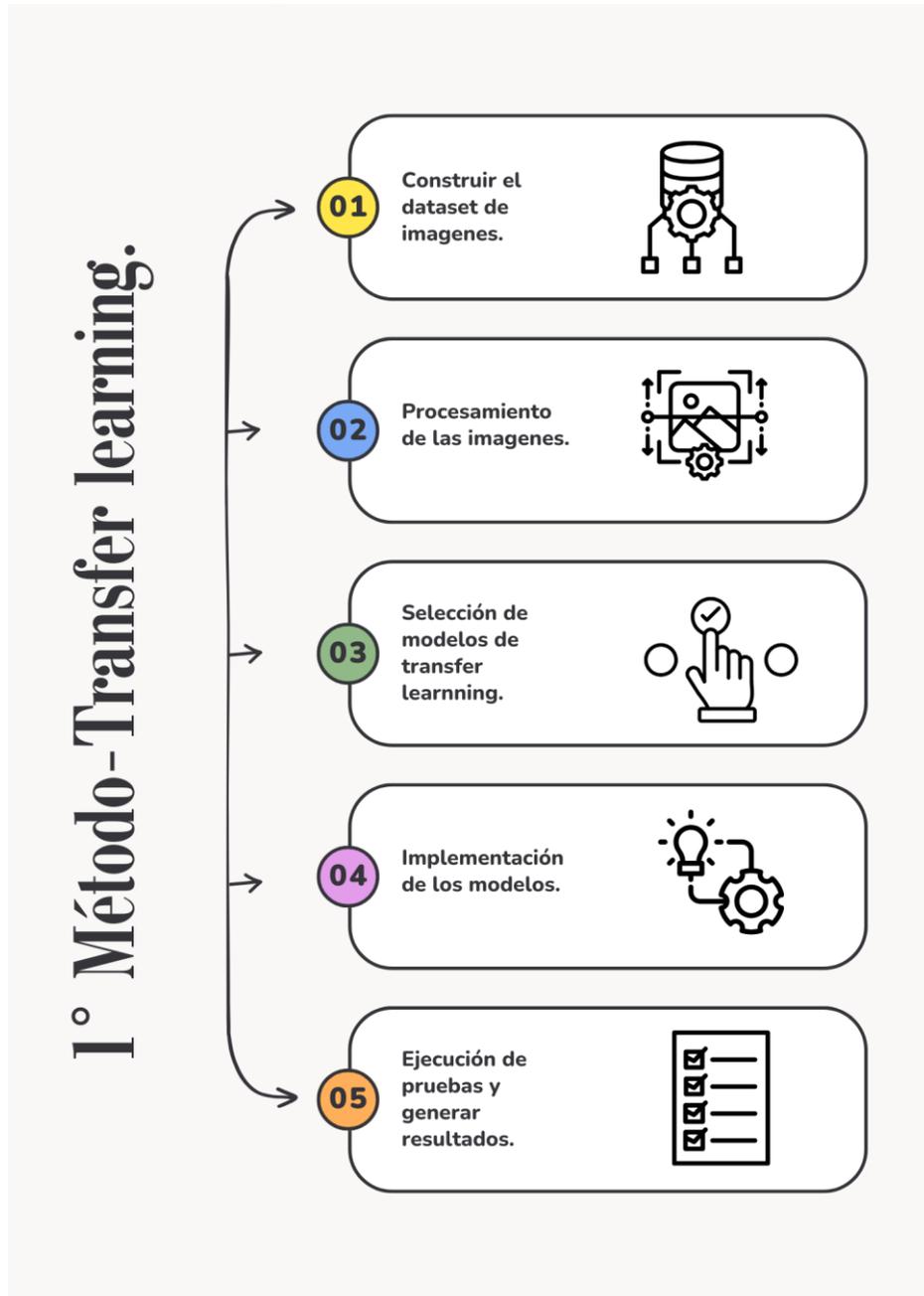


Figura 26 Esquema del 1º método.

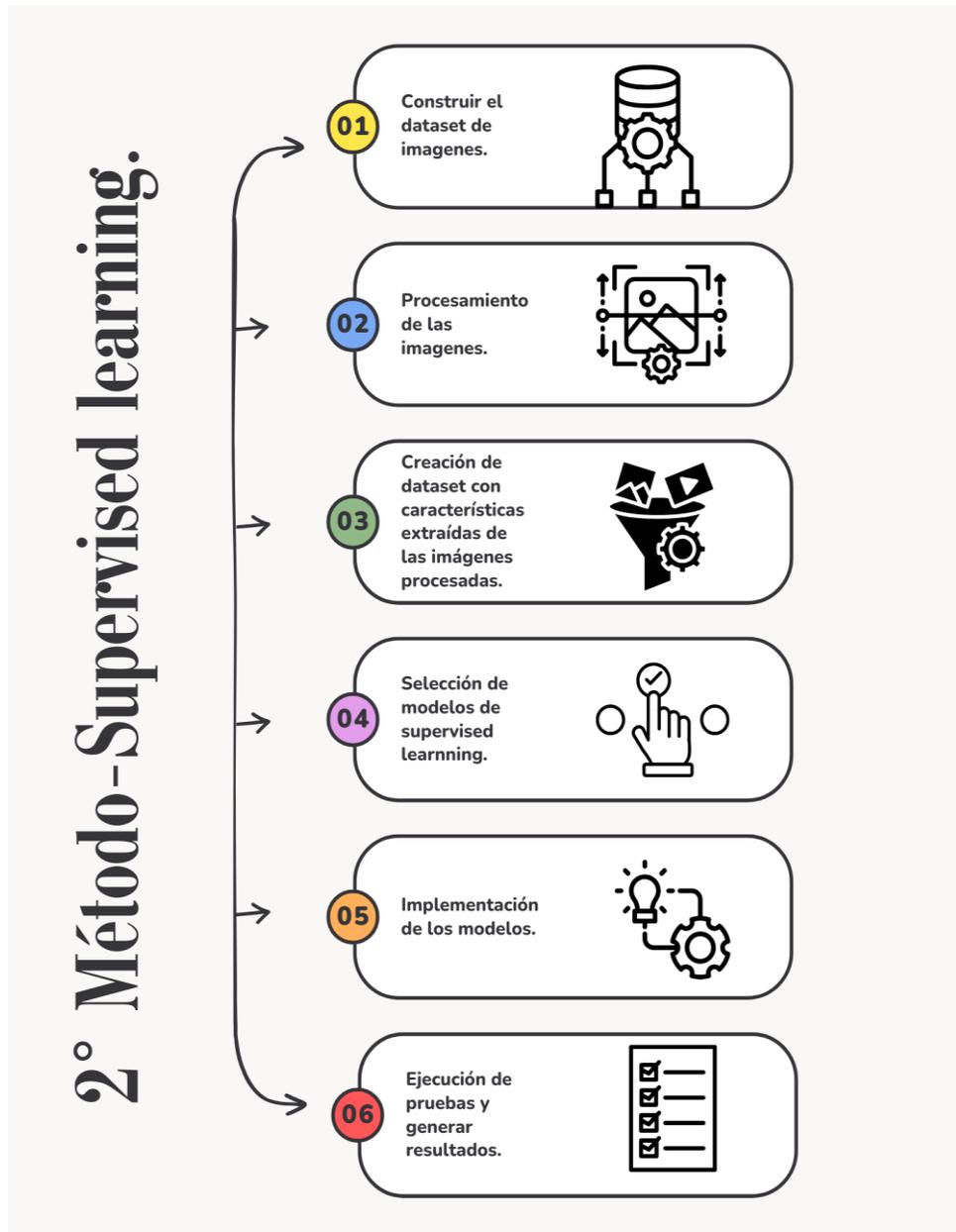


Figura 27 Esquema del 2° método.

3.3.1. Construir dataset de imágenes digitales de fresas.

Para la construcción del dataset se siguieron diferentes pasos, el primer paso consistió en implementar un protocolo a seguir para tomar las imágenes de las fresas que fueron la base para las imágenes finales del dataset.

Las imágenes fueron capturadas en un ambiente controlado usando una

cámara de 64 MP con apertura de f/1.89.

Considerando diferentes protocolos vistos en investigaciones revisadas en la revisión bibliográfica de los trabajos previos, se construyó una estructura física en un ambiente controlado para la captura de imágenes (Figura 20). Para controlar la intensidad de la luz, se utilizó una luz flash de un led. La captura de imágenes se realizó a una distancia de 21 cm de altura y 16.7 cm de diámetro de superficie, lo que permitió capturar las fresas completas de diferentes tamaños. Se mejoró la luminosidad sin rebotes para que se distribuya uniformemente la luz artificial sin provocar sombras alrededor de la fresa. Las imágenes adquiridas con dimensiones de 3456×4608 píxeles con un total de 111 imágenes, no fueron preprocesadas en este paso, para mantener toda la información original posible para aplicar las técnicas de preprocesamiento en un punto posterior de acuerdo con sus objetivos de investigación.

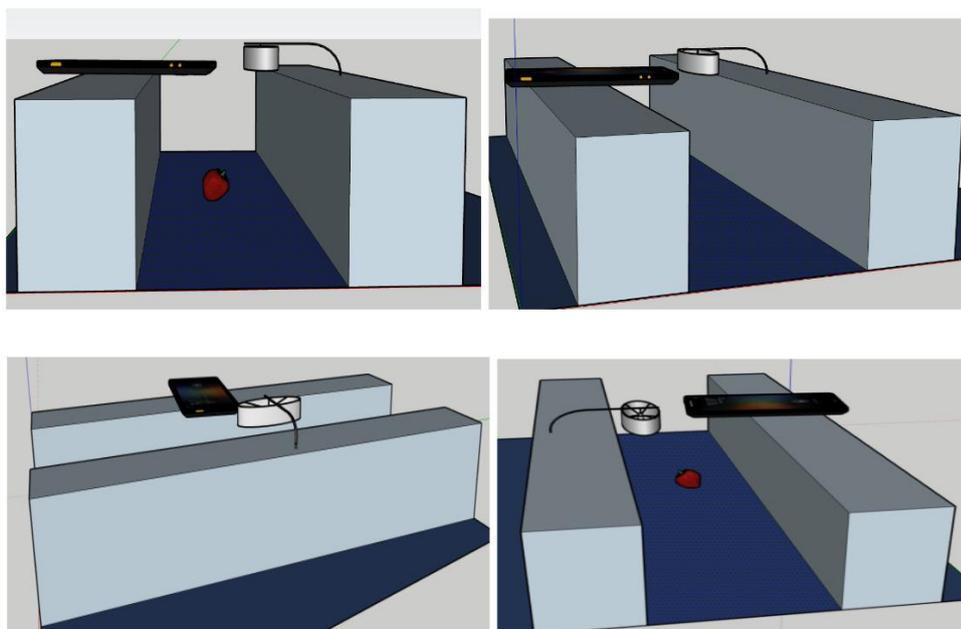


Figura 28 Estructura para tomar fotografías a las fresas.

3.3.2. Realizar el procesamiento de las imágenes digitales de las

fresas.

El segundo paso fue la eliminación del fondo, el cual fue sencillo gracias al fondo azul de la hoja usada en el protocolo de adquisición de las imágenes. En este proceso se realizó en el lenguaje de programación Python en conjunto con la librería OpenCV, lo primero fue quitar el fondo de la imagen, para lo cual se separó la imagen en sus canales de color rojo, verde y azul, adicionalmente una en escala de grises, luego fue binarizada usando el método otsu para segmentar el área de interés usando una imagen producto de la resta de la imagen del canal azul y la escala de grises, para luego buscar los contornos del objeto segmentado y hacer un recorte de la imagen que se ajustó al área del objeto de interés que también se redimensionó a un tamaño de 224 píxeles por 224 píxeles. Por último se giró la imagen 90, 180 y 270 grados, obteniendo de este modo tres imágenes adicionales por cada imagen dando como resultado 444 imágenes que conforman el conjunto de imágenes del dataset que será utilizado en el entrenamiento de los modelos de transfer learning y de donde se extraerán características para generar un segundo dataset para entrenar los modelos de aprendizaje supervisado.

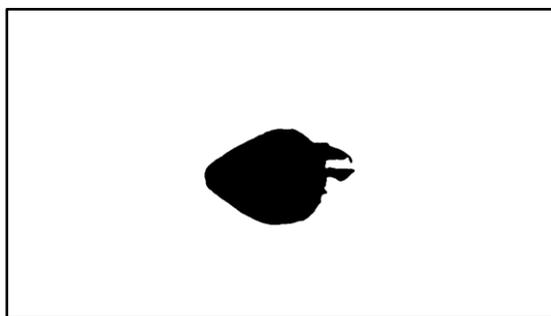


Figura 29 Binarización de la imagen con el método Otsu.



Figura 30 Extracción de la fresa del fondo de la imagen.



Figura 31 Rotación en 90, 180 y 270 grados cada imagen.

3.3.3. Crear un dataset con características extraídas de las imágenes procesadas.

Para la creación del dataset con el que serían entrenados los modelos de aprendizaje supervisado se necesito extraer diferentes características de las imágenes de las fresas obtenidos en el punto anterior. Para esto se decidió obtener los valores derivados del modelo de color HSV (Matiz, saturacion y valor), brillo y las diferentes características de textura de Haralick, estas características son descritas en la tabla 17.

Para obtener los valores del modelo de color HSV se creo una función que recibe la imagen y hace uso de la libreria OpenCV como se observa en la tabla 14, que realiza una conversion del modelo de color RGB a HSV y devuelve los valores de

cada canal de color del modelo HSV.

Tabla 14 Implementación del modelo de color HSV

```
def valoresHSV(img):  
  
    hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)  
  
    avg_color_per_row = np.average(hsv, axis=0)  
  
    avg_color = np.average(avg_color_per_row, axis=0)  
  
    return avg_color[0], avg_color[1], avg_color[2]
```

El valor de brillo se obtuvo al convertir la imagen en escala de grises, del cual luego se calcula promedio, desviación estándar y raíz cuadrada media por medio de la función `ImageStat` de la librería `PIL`, por último devolviendo el valor del brillo.

Tabla 15 Implementación del código para el valor de brillo.

```
def brillo(img):  
  
    im = Image.fromarray(img)  
  
    im.convert('L')  
  
    stat = ImageStat.Stat(im)  
  
    return stat.rms[0]
```

Para obtener las características de textura se utilizó la librería `Mahotas` que permite obtener un vector de 13 características de textura de Haralick. La función que logró esto se puede visualizar en la tabla 16.

Tabla 16 Implementación del código para obtener la textura.

```
def textures(img):  
  
return mahotas.features.haralick(img).mean(axis=0)
```

Tabla 17 Descripción de las características extraídas de las imágenes

Nombre del atributo	Descripción del atributo	Posibles valores
Matiz	Representa la posición del tono en el círculo cromático.	0-180
Saturación	Representa la distancia que hay del eje brillo blanco-negro	0-255
Valor	Representa la altura en el eje blanco-negro.	0-255
Brillo	Representa la intensidad lumínica emitida por la imagen en escala de grises	0-255
SRE	Representa las ejecuciones homogéneas cortas o largas en una imagen.	[0.00521157-0.05322444]
LRE	representa una medida de la distribución de tiradas largas en la imagen	[487.8108465-2813.837395]
GNL	mide la similitud entre los valores de la intensidad del nivel en gris de la imagen	[0.809856674-0.968791805]
RLNU	mide cuál es la similitud entre las longitudes de toda la imagen que están en ejecución.	[5202.520881-10363.55967]
RP	mide la aspereza de la textura, tomando la relación entre el número de corridas y el número de vóxeles en el ROI.	[0.180063966-0.416380195]
LGRE	Énfasis de ejecución de bajo nivel de gris, que mide la distribución de baja valores de nivel de gris	[184.402218-333.993211]
HGRE	Alto énfasis en la ejecución de nivel de gris, que mide la distribución de los valores de nivel de gris más altos	[19205.6675-39201.1437]
SRLGE	Énfasis en el nivel de grises bajo de corto plazo, permite medir la distribución conjunta de tiradas más cortas con valores de nivel gris más bajos.	[6.1042011 - 8.37071601]
SRHGE	Énfasis de alto nivel de grises	[8.56354007 -

	de corto plazo mide la distribución conjunta de tiradas más cortas con valores de nivel de gris más altos.	12.1322021]
LRLGE	énfasis de bajo nivel de gris a largo plazo	[0.0001501 - 0.0005146.]
LRHGE	énfasis de alto nivel de gris a largo plazo	[4.21295115 - 5.83102788.]
GLV	La varianza del nivel de gris mide la variación en la intensidad del nivel de gris para las carreras.	[-0.44125026 - - 0.26302175]
RLV	Es una medida de la variación en las ejecuciones para las longitudes de la ejecución	[0,971997976- 0,998437681]

3.3.4. Seleccionar los modelos de clasificación de machine learning a implementar.

Para el desarrollo de esta investigación se seleccionaron los siguientes modelos de clasificación.

3.3.4.1. AlexNet.

Se utilizó este modelo debido a su capacidad para aprovechar capas convolucionales, las cuales permiten extraer características específicas de las imágenes de entrada. Estas capas utilizan filtros convolucionales que recorren diferentes secciones de la imagen, identificando patrones visuales relevantes como contornos, texturas y formas. Esta técnica resulta clave para detectar las diferencias visuales que permiten clasificar las fresas según su nivel de madurez, mejorando la precisión y eficiencia del proceso de clasificación.

3.3.4.2. VGG 16.

Se implementó este modelo teniendo en cuenta su notable capacidad para identificar detalles minuciosos en las imágenes. Esto se logra gracias a los filtros que posee, los cuales permiten detectar características visuales finas y esenciales. Además, VGG 16 ofrece una transferencia de aprendizaje altamente efectiva, lo que facilita su uso en tareas donde los datos de entrenamiento son limitados. Otra ventaja importante es su rendimiento probado en diversas aplicaciones, así como su disponibilidad y facilidad de implementación. Asimismo, su adaptabilidad a requerimientos específicos lo convierte en una opción ideal para la clasificación de fresas por su nivel de madurez.

3.3.4.3. VGG 19.

Este modelo fue elegido debido a su arquitectura profunda, lo que le permite aprender representaciones más complejas y detalladas de las imágenes. Además, ofrece una excelente capacidad de generalización, lo que lo hace robusto frente a la variabilidad presente en las imágenes, como cambios en la iluminación, ángulos o texturas. VGG19 también destaca por su disponibilidad, facilidad de implementación y amplio soporte, lo que asegura una integración fluida en diferentes entornos. Estas características hacen de VGG 19 una opción confiable para la clasificación de fresas, ya que permite capturar detalles finos y alcanzar una alta precisión en la tarea.

3.3.4.4. MobileNet V2.

Se utilizó este modelo por su eficiencia computacional y tamaño compacto, lo que lo convierte en una opción ideal para aplicaciones donde los recursos de hardware son limitados. A pesar de su diseño

ligero, ofrece un rendimiento sólido en tareas de clasificación, manteniendo un equilibrio adecuado entre precisión y velocidad. Además, MobileNet V2 es altamente flexible, lo que le permite adaptarse a diferentes tipos de imágenes y escenarios. Su capacidad para implementar transfer learning también es una ventaja clave, ya que permite aprovechar modelos pre entrenados para mejorar la clasificación de fresas sin necesidad de grandes conjuntos de datos. Estas características lo hacen especialmente atractivo para situaciones en las que se necesita un modelo eficiente, pero potente, que pueda funcionar en dispositivos con recursos computacionales reducidos.

3.3.4.5. ResNet-50.

Se implementó este modelo debido a su capacidad avanzada para el aprendizaje profundo, lo que le permite construir representaciones complejas de las imágenes. Un aspecto clave de su arquitectura son las conexiones residuales, que facilitan el entrenamiento de redes muy profundas al mitigar el problema de la degradación del rendimiento. Además, ResNet-50 ha demostrado un rendimiento sobresaliente en numerosas aplicaciones de visión por computadora, lo que lo convierte en una opción confiable. Su capacidad para realizar transfer learning permite adaptar fácilmente el modelo a nuevas tareas con menos datos de entrenamiento. Sumado a su amplia disponibilidad de implementación y soporte, ResNet-50 se presenta como una opción sólida para lograr una alta precisión en la clasificación de fresas según su nivel de madurez.

3.3.4.6. Regresión Logística

Se implementó este modelo debido a su capacidad para manejar problemas de clasificación binaria y multinomial, lo que se adapta bien a este tipo de análisis donde existen distintos niveles de madurez. Este modelo es efectivo porque estima las probabilidades de pertenencia a una clase específica en función de características observables, como color, textura, o parámetros químicos de las fresas, lo cual facilita la toma de decisiones en escenarios de clasificación. Además la regresión logística no demanda una gran cantidad de datos para lograr buenos resultados, siendo eficiente en términos computacionales.

3.3.4.7. Random Forest

Se implementó este modelo debido a su capacidad para manejar problemas de clasificación complejos y no lineales. Este método combina múltiples árboles de decisión, generando un conjunto robusto que aumenta la precisión del modelo y reduce el riesgo de sobreajuste (overfitting). En el contexto de clasificación de fresas, donde se consideran múltiples características (como color, textura y reflectancia de luz), Random Forest puede evaluar cada una de ellas de manera efectiva y mejorar la exactitud del modelo al integrar los resultados de varios árboles de decisión. Random Forest maneja bien las variaciones en los datos y es menos sensible al ruido, lo que es útil al clasificar frutas, donde las características visuales pueden variar incluso en un mismo nivel de madurez. Además, Random Forest ofrece interpretabilidad: al analizar la importancia de cada variable, es posible identificar las características más relevantes que influyen en la madurez de las fresas, permitiendo ajustes o mejoras en el proceso de clasificación.

3.3.4.8. SVM

Se utilizó Support Vector Machine (SVM) teniendo en cuenta su eficiencia de encontrar fronteras de decisión óptimas entre clases en problemas complejos. La SVM utiliza hiperplanos en un espacio de características para separar los datos en distintas clases, lo que resulta útil en la clasificación de niveles de madurez donde puede haber solapamiento en las características visuales o químicas de las fresas. Además porque maneja datos de alta dimensionalidad sin incurrir en sobreajuste (overfitting), especialmente útil cuando se trabajan con múltiples características o texturas complejas de las fresas. Además, SVM es eficaz con conjuntos de datos pequeños o medianos, lo cual es beneficioso si el número de imágenes o muestras de fresas es limitado.

3.3.4.9. Naive Bayes

Se eligió trabajar esta investigación con Naive Bayes debido a su simplicidad, rapidez y eficacia en problemas de clasificación con varias clases. Este modelo se basa en el Teorema de Bayes, asumiendo que las características son independientes entre sí, lo cual lo hace ideal para sistemas donde el procesamiento rápido es esencial, como en aplicaciones de clasificación en tiempo real de frutas. Al clasificar fresas en diferentes niveles de madurez, Naive Bayes evalúa probabilidades basadas en características como color, textura y reflectancia de luz, facilitando una clasificación eficiente y precisa. La ventaja de Naive Bayes es que requiere una menor cantidad de datos para entrenarse y sigue siendo efectivo en escenarios con datos limitados o incompletos, lo cual es común en la clasificación agrícola donde la recolección de datos puede ser costosa. Además, el modelo es fácil de interpretar, lo que permite entender la contribución de cada característica en el proceso de clasificación.

3.3.5. Realizar la implementación de los modelos seleccionados.

3.3.5.1. Aprendizaje por transferencia

Para la implementación de los modelos se modificó las últimas 5 capas y la capa de salida según las etiquetas a clasificar, para esto se creó una función que devuelve el conjunto de estas.

Tabla 18 Función para modificar las últimas 5 capas.

```
def capas_models():
```

```
    x= Flatten(name='flatten')(last_layer)
```

```
    x = Dense(128, activation='relu', name='fc1')(x)
```

```
    x=Dropout(0.3)(x)
```

```
    x = Dense(128, activation='relu', name='fc2')(x)
```

```
    x=Dropout(0.3)(x)
```

```
    out = Dense(num_classes, activation='softmax', name='output')(x)
```

```
return out
```

Dado que se trabajó con imagen se decidió utilizar una capa Flatten para acoplar los datos de la imagen de entrada en una matriz unidimensional, capas dense con la función ReLU para anular los valores negativos proveniente de las capas anteriores y Dropout para regularizar las capas, por último la capa de salida (out) contiene el número de clases y la función “Softmax” para regularizar los valores de salida del modelo.

Luego de probar con las diferentes funciones de pérdida y de

optimización se eligió utilizar como función de pérdida “categorical_crossentropy” y el optimizador “adam”, debido que se adaptan mejor con los datos con los que se están entrenando los modelos.

3.3.5.1.1. Implementación y arquitectura del modelo VGG19

Tabla 19 Implementación del modelo VGG19.

```
# capa de entrada
image_input = Input(shape=(width_shape, height_shape, 3))

# modelo
m_VGG19 = VGG19(input_tensor=image_input,
include_top=False, weights='imagenet')
last_layer = m_VGG19.layers[-1].output
custom_model_vgg19 = Model(image_input, capas_models())

# congelar todas las capas excepto las capas densas
for layer in custom_model_vgg19.layers[:-6]:
    layer.trainable = False
```

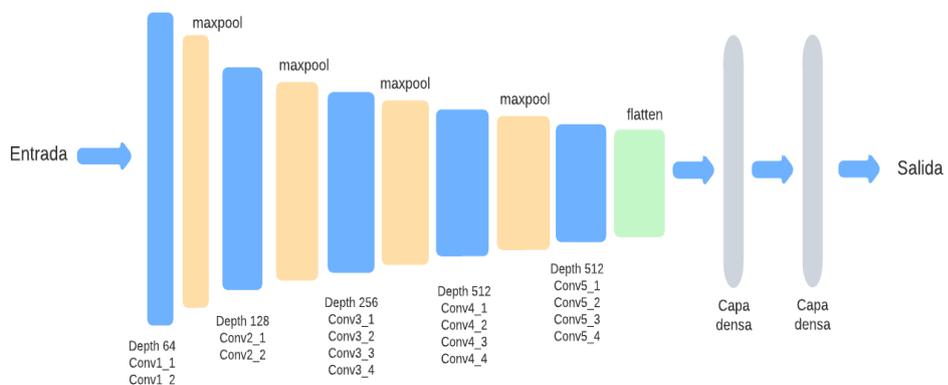


Figura 32 Arquitectura del modelo VGG19.

3.3.5.1.2. Implementación y arquitectura del modelo ResNet-50.

Tabla 20 Implementación del modelo ResNet 50.

```
# capa de entrada
image_input = Input(shape=(width_shape, height_shape, 3))

# modelo
m_Resnet50 = ResNet50(input_tensor=image_input,
include_top=False, weights='imagenet')
last_layer = m_Resnet50.layers[-1].output
custom_model_r50 = Model(image_input, capas_models())

# congelar todas las capas excepto las capas densas
for layer in custom_model_r50.layers[:-6]:
    layer.trainable = False
```

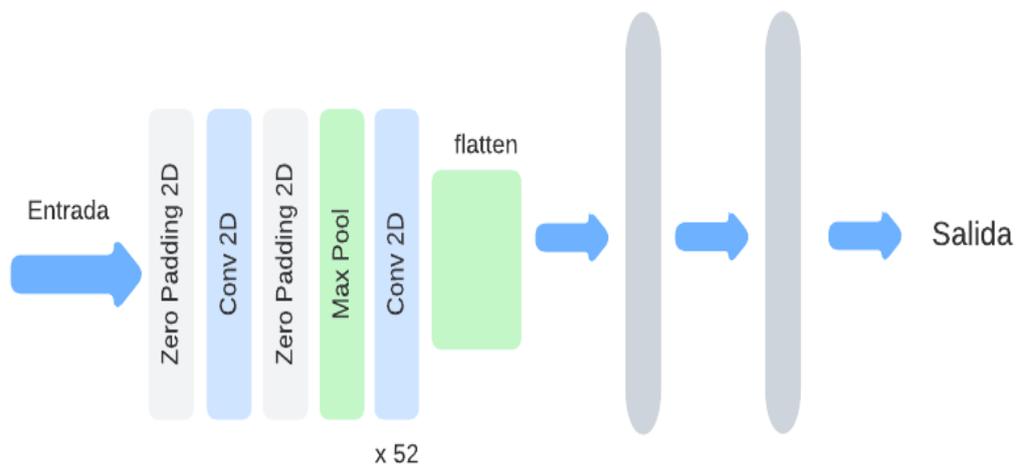


Figura 33 Arquitectura del modelo ResNet 50.

3.3.5.1.3. Implementación y arquitectura del modelo Mobile

Net V2.

Tabla 21 Implementación del modelo Mobile Net V2.

```
# capa de entrada
image_input = Input(shape=(width_shape, height_shape, 3))

# modelo
m_Mobilev2 = MobileNetV2(input_tensor=image_input,
include_top=False, weights='imagenet')

last_layer = m_Mobilev2.layers[-1].output

custom_model_mv2 = Model(image_input, capas_models())

# congelar todas las capas excepto las capas densas
for layer in custom_model_mv2.layers[:-6]:
    layer.trainable = False
```

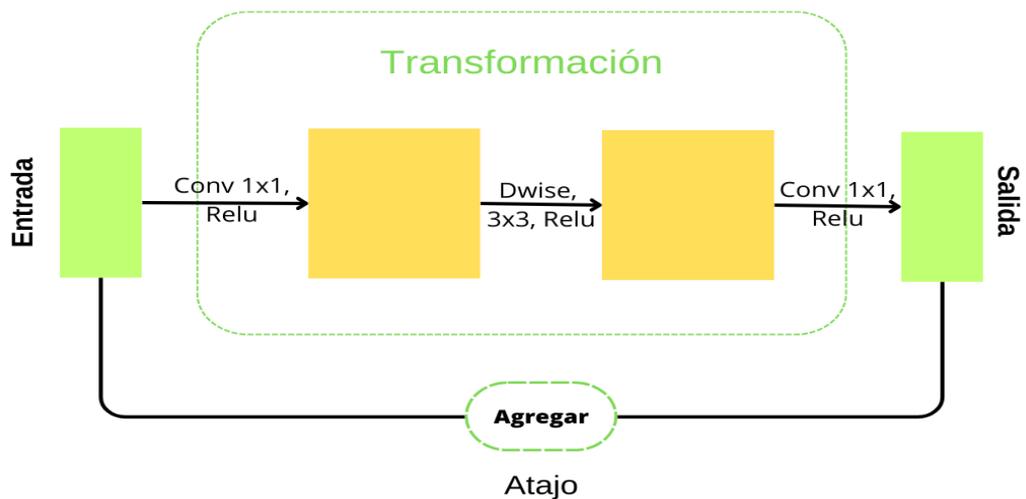


Figura 34 Arquitectura del modelo Mobile Net V2.

3.3.5.1.4. Implementación y arquitectura del modelo AlexNet.

Tabla 22 Implementación del modelo AlexNet.

```
model_alexNet = Sequential()

# 1st capa de entrada

model_alexNet.add(image_input)

# Max Pooling

model_alexNet.add(MaxPooling2D(pool_size=(2,2), strides=(2,2),
padding='valid'))

# 2nd Convolutional Layer

model_alexNet.add(Conv2D(filters=256, kernel_size=(11,11),
strides=(1,1), padding='valid', activation='relu'))

# Max Pooling

model_alexNet.add(MaxPooling2D(pool_size=(2,2), strides=(2,2),
padding='valid'))

# 3rd Convolutional Layer

model_alexNet.add(Conv2D(filters=384, kernel_size=(3,3), strides=(1,1),
padding='valid', activation='relu'))

# 4th Convolutional Layer

model_alexNet.add(Conv2D(filters=384, kernel_size=(3,3), strides=(1,1),
padding='valid', activation='relu'))# 5th Convolutional Layer

model_alexNet.add(Conv2D(filters=256, kernel_size=(3,3), strides=(1,1),
padding='valid', activation='relu'))

# Max Pooling

model_alexNet.add(MaxPooling2D(pool_size=(2,2), strides=(2,2),
```

```
padding='valid'))
# Capas modificadas
model_alexNet.add(Flatten(name='flatten'))
# 1st Fully Connected Layer y Dropout to prevent overfitting
model_alexNet.add(Dense(128, activation='relu', name='fc1'))
model_alexNet.add(Dropout(0.3))
# 2st Fully Connected Layer y Dropout to prevent overfitting
model_alexNet.add(Dense(128, activation='relu', name='fc2'))
model_alexNet.add(Dropout(0.3))
#Output Layer
model_alexNet.add(Dense(num_classes, activation='softmax',
name='output'))
```

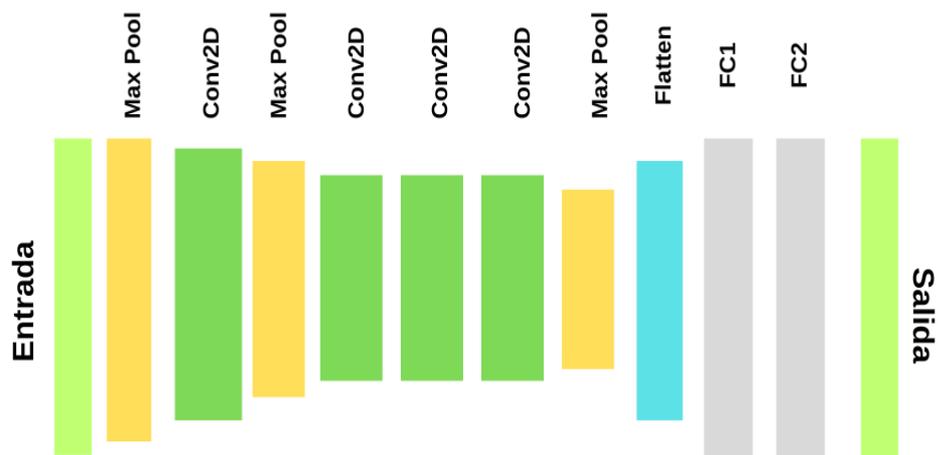


Figura 35 Arquitectura del modelo AlexNet.

3.3.5.1.5. Implementación y arquitectura del modelo VGG16.

Tabla 23 Implementación del modelo VGG16.

```
# capa de entrada  
image_input = Input(shape=(width_shape, height_shape, 3))  
  
# modelo  
model_vgg16 = VGG16(input_tensor=image_input,  
include_top=False, weights='imagenet')  
last_layer = model_vgg16.layers[-1].output  
custom_model_vgg16 = Model(image_input, capas_models())  
  
# congelar todas las capas excepto las capas densas  
for layer in custom_model_vgg16.layers[:-6]:  
    layer.trainable = False
```

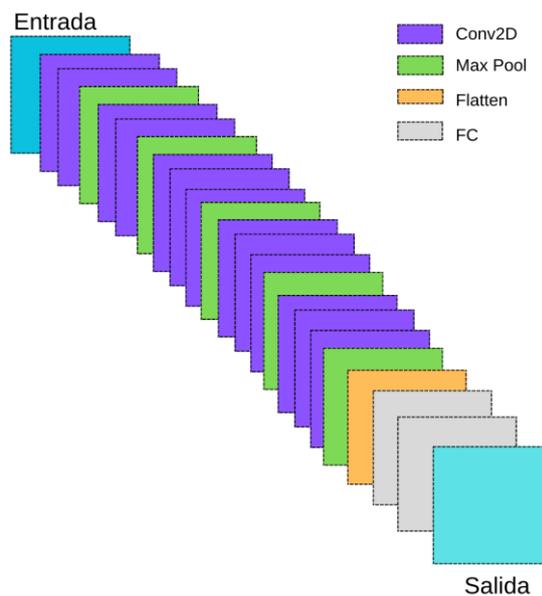


Figura 36 Arquitectura del modelo VGG16.

3.3.5.2. Aprendizaje Supervisado

3.3.5.2.1. Implementación del modelo de Regresión logística

Tabla 24 Implementación del modelo de Regresión logística

Crear el modelo

```
modelLR = LogisticRegression(solver="liblinear", max_iter=1000,  
penalty="l2")
```

Entrenar el modelo

```
modelLR.fit(X_train, y_train)
```

3.3.5.2.2. Implementación del modelo Random Forest

Tabla 25 Implementación del modelo Random Forest

Crear el modelo

```
modelRFC = RandomForestClassifier( n_estimators=100,  
random_state=2016, min_samples_leaf=8 )
```

Entrenar el modelo con el conjunto de entrenamiento

```
modelRFC.fit(X_train, y_train)
```

3.3.5.2.3. Implementación del modelo SVM

Tabla 26 Implementación del modelo SVM

Crear el modelo SVM con kernel 'linear'

```
svm_model = SVC(kernel="linear", probability=True)
```

Entrenar el modelo en el conjunto de entrenamiento

```
svm_model.fit(X_train, y_train)
```

3.3.5.2.4. Implementación del modelo de Naive bayes

Tabla 27 Implementación del modelo Naive bayes

Crear el modelo Naive Bayes

```
naive_bayes_model = GaussianNB()
```

Entrenar el modelo en el conjunto de entrenamiento

```
naive_bayes_model.fit(X_train, y_train)
```

3.3.6. Ejecutar pruebas y generar resultados.

Tabla 28 Comparación de resultados del indicador de precisión.

Precisión	Entrenamiento	Test	Validación
VGG-19	0.9495	0.9302	0.94
VGG-16	0.9639	0.9844	0.94
Resnet 50	1	0.9531	0.9766
MovilNetV2	0.9495	0.8594	0.8466
AlexNet	0.9892	0.4531	0.48
Regresión			
Logística	0.9742	1	1
Random Forest	0.9968	1	1

SVM	0.9645	0.9556	0.9433
Naive Bayes	0.7516	0.8889	0.8866

3.3.6.1. Aprendizaje por transferencia.

3.3.6.1.1. Resultados con el modelo VGG 19.

Como se observa en la tabla 28, el modelo vgg 19 tuvo excelentes resultados en su entrenamiento logrando una precisión del 94.95% en entrenamiento y 93.02% en test. En la matriz de confusión que es resultados de las imágenes de validación, se logró obtener un 100% de verdaderos positivos para las fresas maduras, un 87% de verdaderos positivos para las fresas verdes y un 13% de falsos negativos que fueron detectados como fresas maduras, por último logró obtener 93% de verdaderos positivos para las fresas dañadas y un 7% de falsos negativos que fueron detectadas como fresas maduras.

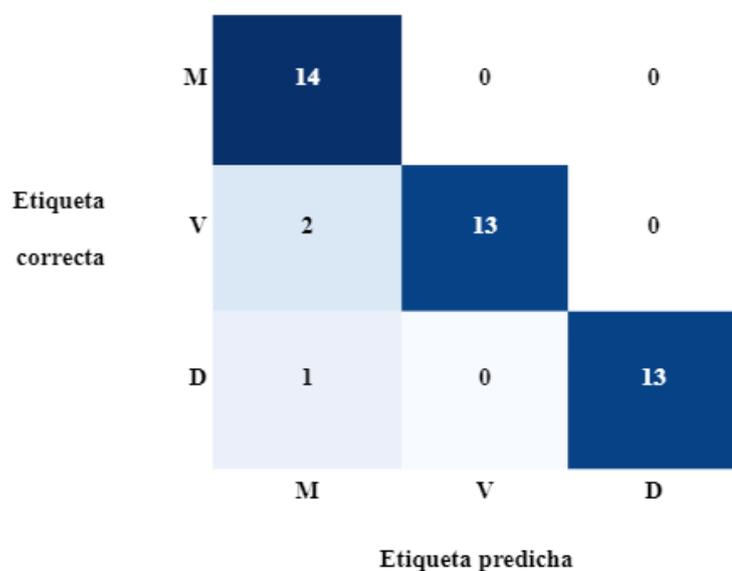


Figura 37 Matriz de confusión del modelo VGG19.

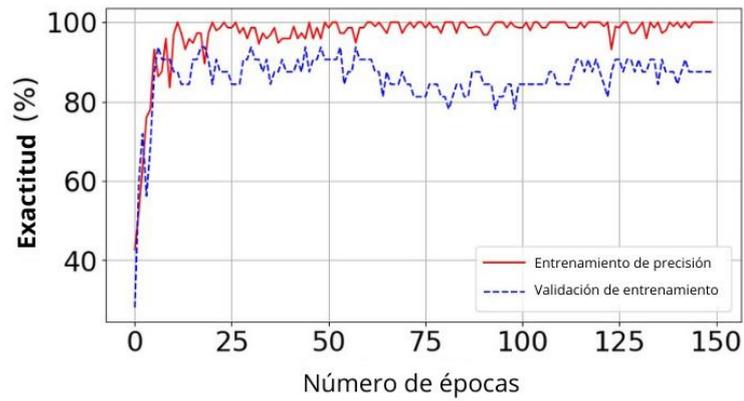


Figura 38 Histograma de exactitud del modelo VGG19.

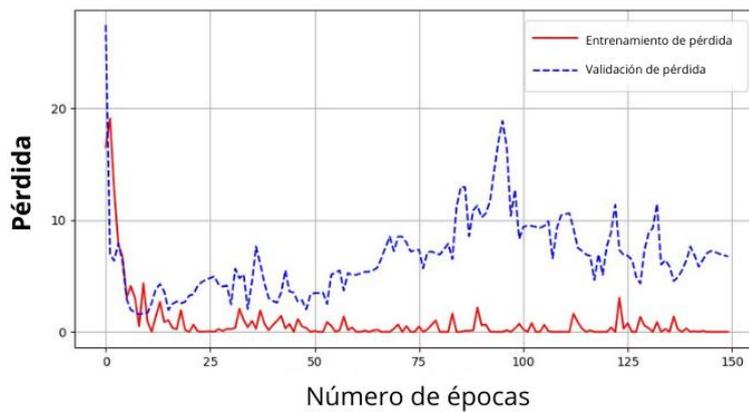


Figura 39 Histograma de pérdidas del modelo VGG19.

3.3.6.1.2. Resultados con el modelo ResNet 50.

Como se observa en la tabla 28, el modelo ResNet 50 tuvo excelentes resultados en su entrenamiento logrando una precisión del 100% en entrenamiento, 95.31% en test y 97.66% en validación. En la matriz de confusión que es resultados de las imágenes de validación, se logró obtener un 100% de verdaderos positivos para las fresas maduras, un 93% de verdaderos positivos para las fresas verdes y un

7% de falsos negativos que fueron detectados como fresas dañadas, por último logró obtener 100% de verdaderos positivos para las fresas dañadas.

Etiqueta correcta	Etiqueta predicha		
	M	V	D
M	14	0	0
V	0	14	1
D	0	0	14

Figura 40 Matriz de confusión del modelo ResNet 50.

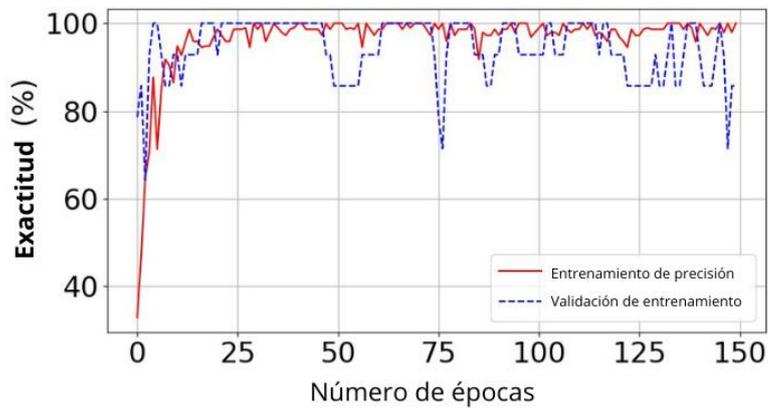


Figura 41 Histograma de exactitud del modelo ResNet 50.



Figura 42 Histograma de pérdida del modelo ResNet 50.

3.3.6.1.3. Resultados con el modelo MobileNet v2.

Como se observa en la tabla 28, aunque el modelo MobileNet v2 no tuvo resultados tan elevados como los modelos vgg 19 y resnet 50, logró obtener resultados favorables, logrando en su entrenamiento una precisión del 94.95% en entrenamiento y 85.94 en test. En la matriz de confusión que es resultados de las imágenes de validación, se logró obtener un 79% de verdaderos positivos para las fresas maduras y un

21% de falsos negativos que fueron detectados como fresas verdes, un 80% de verdaderos positivos para las fresas verdes y un 20% de falsos negativos que fueron detectados como fresas maduras, por último logró obtener 93% de verdaderos positivos para las fresas dañadas y un 7% de falsos negativos que fueron detectados como fresas verdes.

Etiqueta correcta	M	11	3	0
	V	3	12	0
	D	0	1	13
		M	V	D
		Etiqueta predicha		

Figura 43 Matriz de confusión del modelo MobileNet.

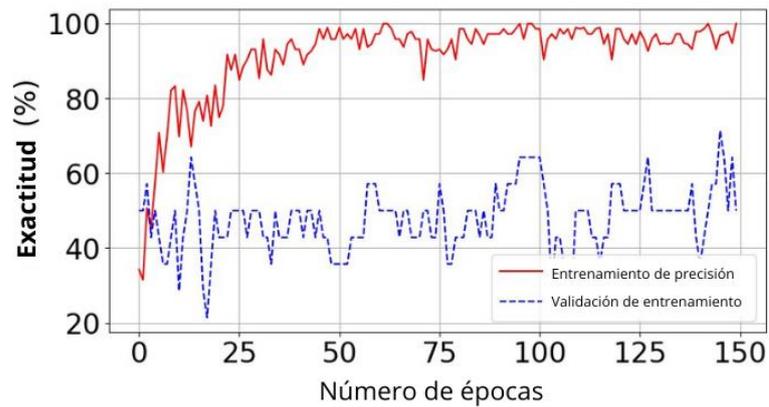


Figura 44 Histograma de exactitud del modelo MobileNet.

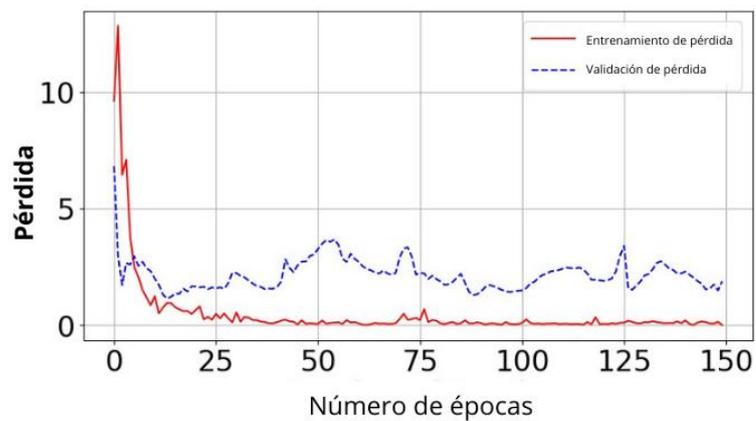


Figura 45 Histograma de pérdida del modelo MobileNet.

3.3.6.1.4. Resultados con el modelo AlexNet.

AlexNet es el modelo que tuvo los peores resultados en las pruebas como se observa en la tabla 28, teniendo un sobreajuste en su entrenamiento con 98.92% y una precisión baja en test de 45.31%. En la matriz de confusión logra un 71% de verdaderos positivos y un 29% de falsos negativos entre verdes y dañadas. Para las fresas verdes es donde obtiene los peores resultados clasificando con un 33%

de verdaderos positivos en verdes y 33% de falsos negativos en maduras y dañadas. Por último un 43% de verdaderos positivos en dañadas y 57% de falsos negativos entre maduras y verdes.

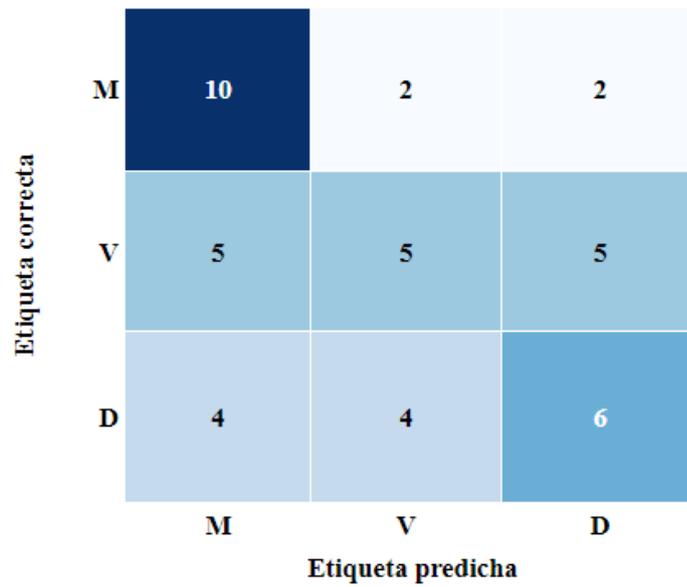


Figura 46 Matriz de confusión del modelo AlexNet.

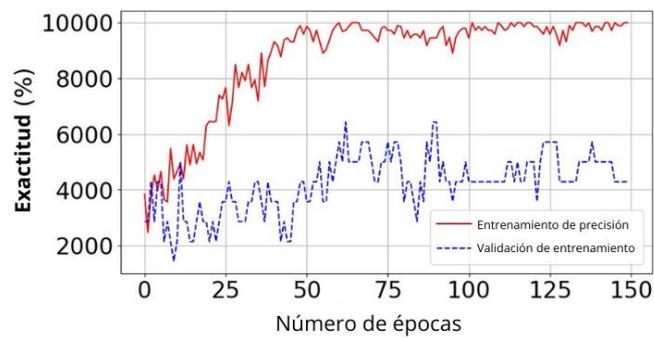


Figura 47 Histograma de exactitud del modelo AlexNet.

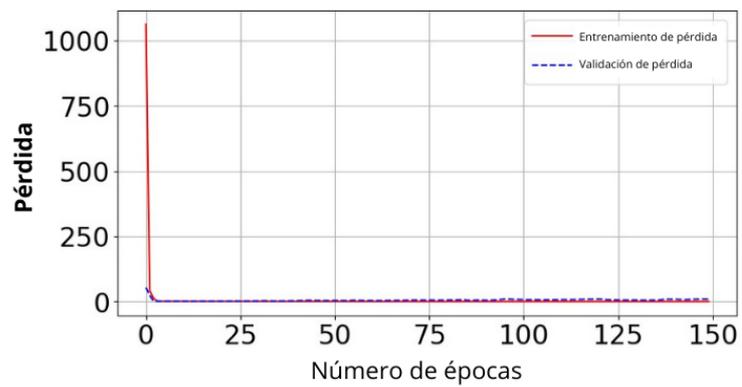


Figura 48 Histograma de pérdida del modelo AlexNet.

3.3.6.1.5. Resultados con el modelo VGG16.

Como se observa en la tabla 28, el modelo VGG 16 logra resultados elevados con una precisión de 96.39% en entrenamiento y 98.44% en test. En la matriz de confusión que es resultados de las imágenes de validación, se logró obtener un 100% de verdaderos positivos para las fresas maduras, un 100% de verdaderos positivos para las fresas verdes, por último logró obtener 79% de verdaderos positivos para las fresas dañadas y un 21% de falsos negativos que fueron detectados entre maduras y verdes.

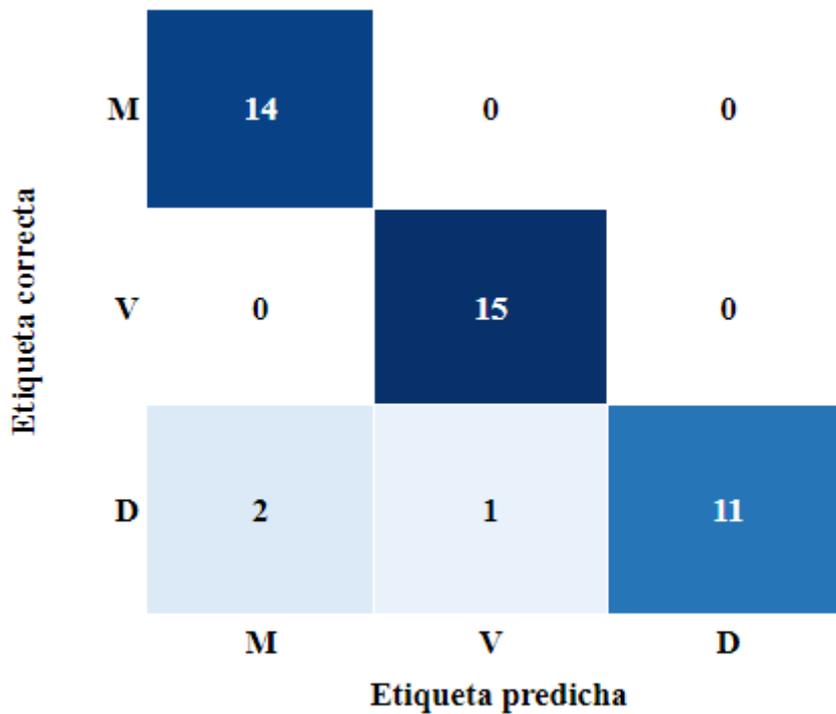


Figura 49 Matriz de confusión del modelo VGG16.

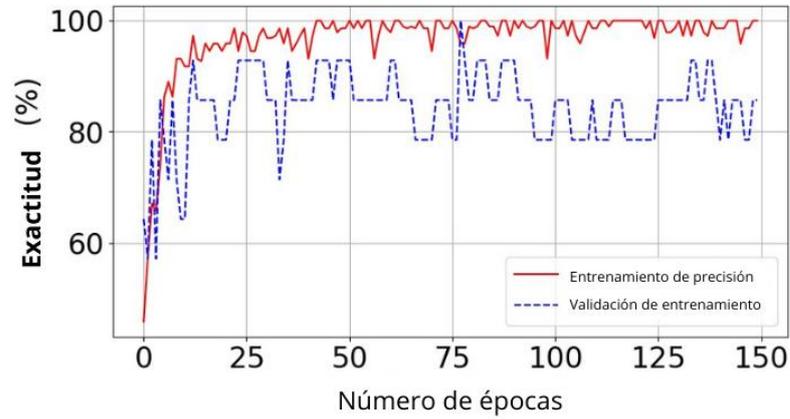


Figura 50 Histograma de exactitud del modelo VGG16.

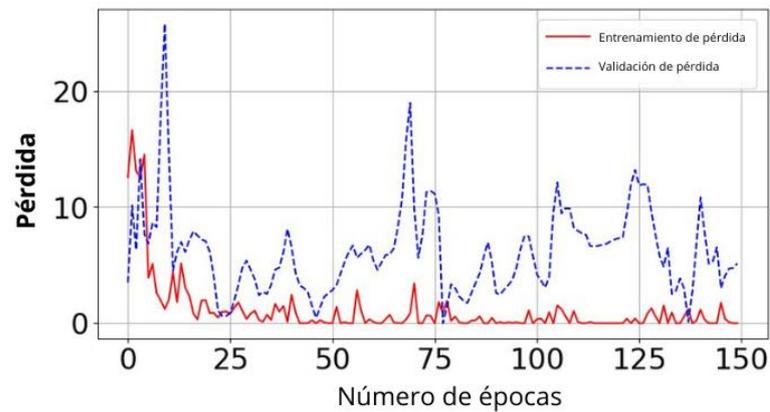


Figura 51 Histograma de pérdida del modelo VGG16.

3.3.6.2. Aprendizaje supervisado.

3.3.6.2.1. Resultados con el modelo de regresión logística.

Como se observa en la tabla 28, el modelo de regresión logística logra un excelente resultado con un 97.42% y 100% en entrenamiento y test. Y en la evaluación realizada con la data de validación mostrada en la matriz de confusión, el modelo logra una

precisión del 100% de verdaderos positivos en todas las etiquetas de madurez de las fresas.

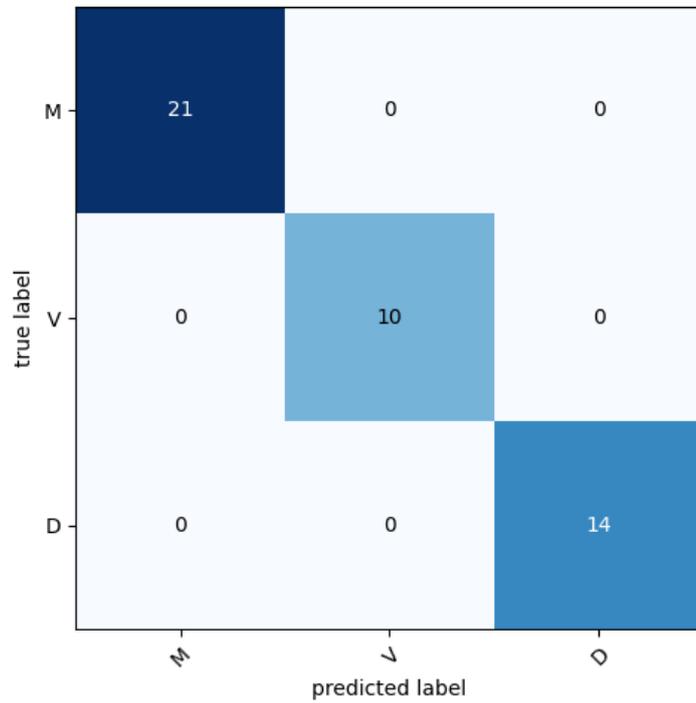


Figura 52 Matriz de confusión del modelo de regresión logística.

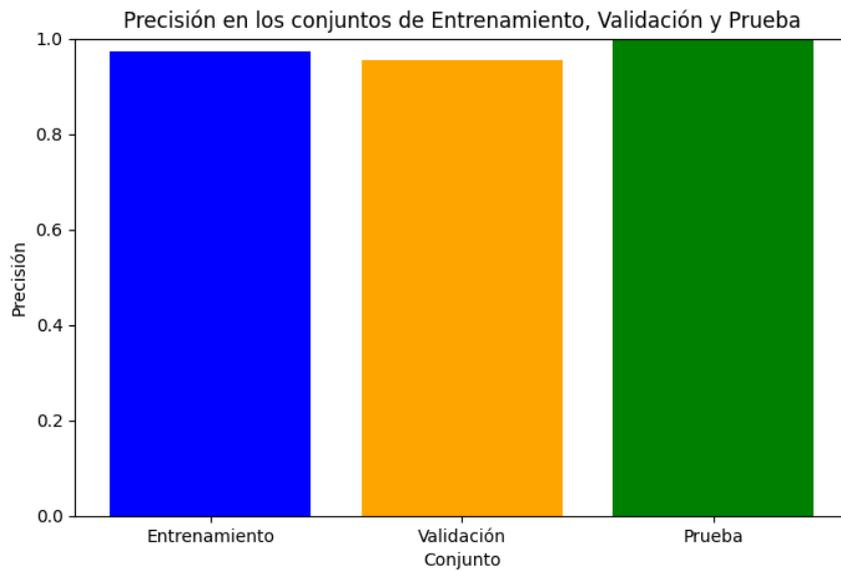


Figura 53 Histograma de exactitud del modelo de regresión logística.

3.3.6.2.2. Resultados con el modelo de Random Forest.

El modelo Random Forest logra resultados muy similares a la regresión logística pero superando en su precisión en entrenamiento con 99.68% e igualmente 100% con data de test. Durante su evaluación con la data de validación logra clasificar correctamente las etiquetas de madurez con una precisión del 100% como se muestra en la matriz de confusión.

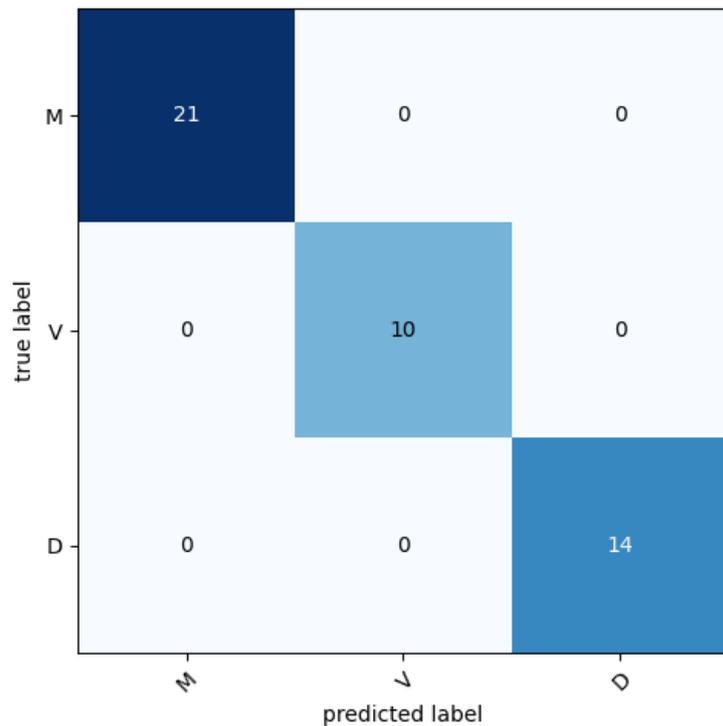


Figura 54 Matriz de confusión del modelo de Random Forest.



Figura 55 Histograma de exactitud del modelo de Random Forest.

3.3.6.2.3. Resultados con el modelo SVM.

Aunque el modelo SVM no logra el mismo resultado que los modelos de Regresión logística y Random Forest, si obtiene resultados muy favorables con un 96.45% y 95.56% en entrenamiento y test. En la evaluación con data de validación mostrada en la matriz de confusión, logró predecir correctamente en un 100% las fresas maduras, un 90% para las fresas verdes y 92.86% para las fresas dañadas.

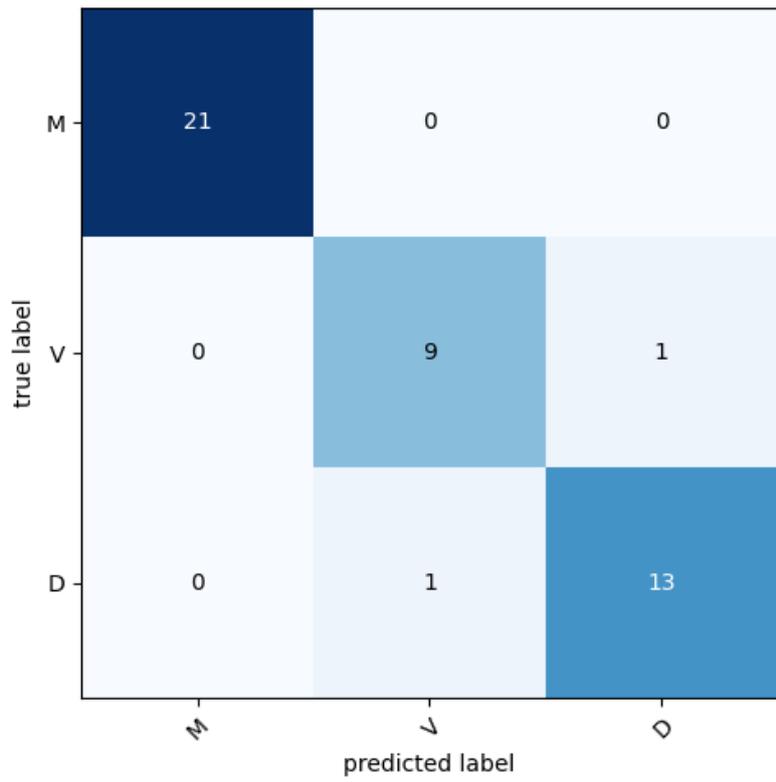


Figura 56 Matriz de confusión del modelo SVM.

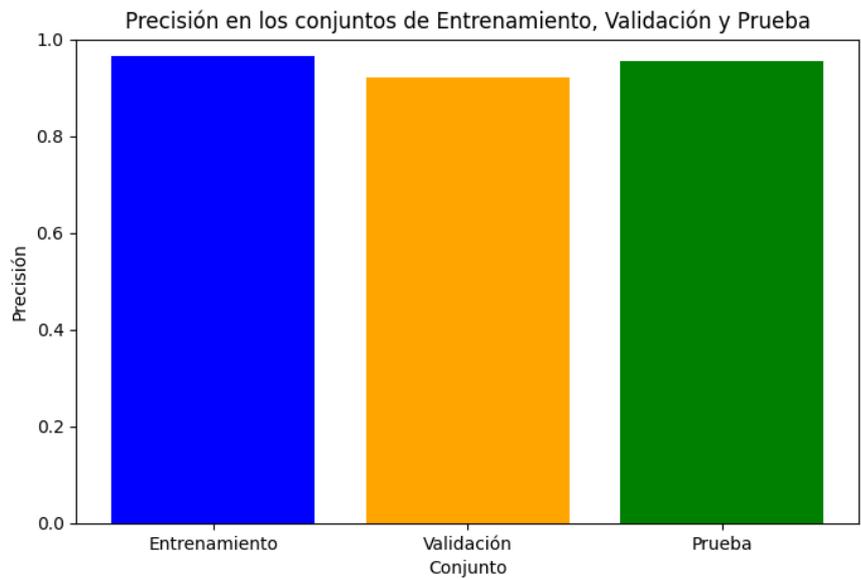


Figura 57 Histograma de exactitud del modelo SVM.

3.3.6.2.4. Resultados con el modelo Naive Bayes.

Como se puede visualizar en la tabla 28, el modelo Naive bayes es el modelo de aprendizaje supervisado que obtiene la precisión más baja pero no lo suficiente para tener un bajo rendimiento, logrando un 75.16% y 88.89% en entrenamiento y test. Logrando con data de validación como se muestra en la matriz de confusión un 100% para clasificar las fresas verdes y maduras, siendo en las fresas dañadas donde tiene un bajo desempeño donde de 14 ejemplos, solo logra predecir correctamente 9 como frutas dañadas y errando en 4 como verdes y 1 como maduro.

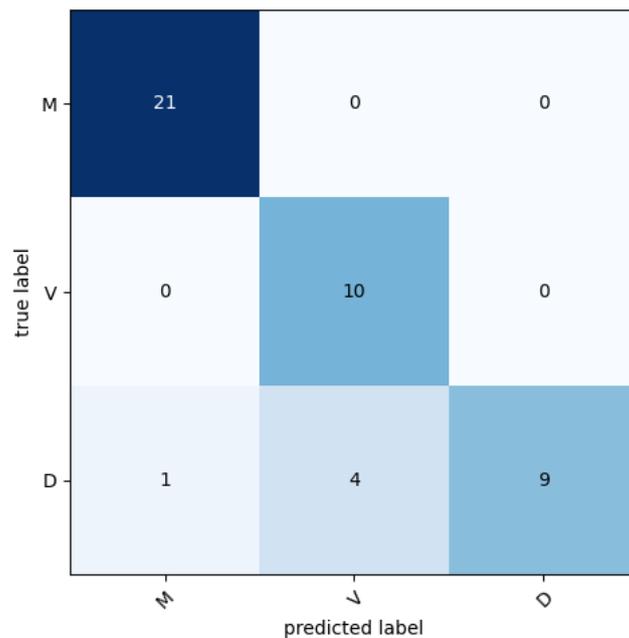


Figura 58 Matriz de confusión del modelo Naive Bayes.

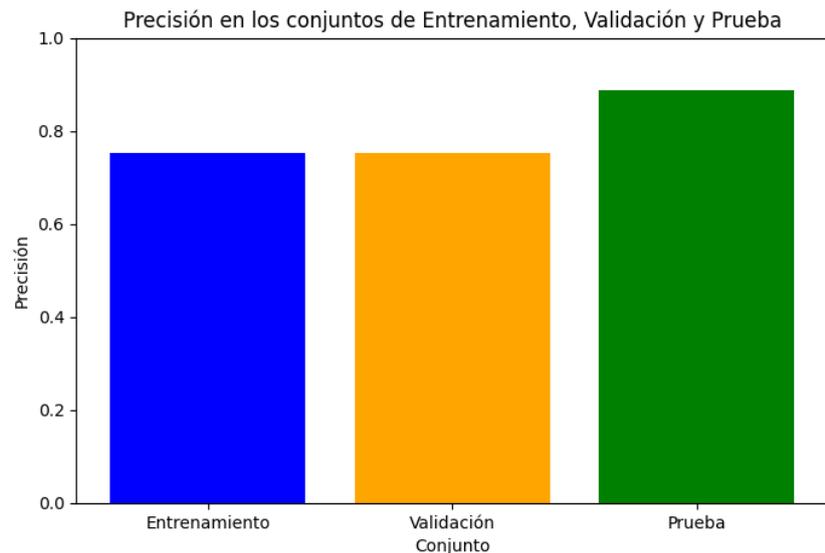


Figura 59 Histograma de exactitud del modelo Naive Bayes.

4. CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

Se llevó a cabo la construcción de una base de datos compuesta por 444 imágenes de fresas en diversos estados de madurez. Todas las imágenes fueron capturadas bajo condiciones uniformes de iluminación, contraste y fondo, utilizando la misma configuración de la cámara en todos los casos, con el fin de obtener una base de datos integral y homogénea.

A través de la implementación del método Otsu en el conjunto de datos, se logró segmentar el área de interés en las imágenes, lo cual se logró mediante la sustracción del canal azul y la escala de grises.

Un segundo dataset con características extraídas de las imágenes obtenidas en el primer dataset fue necesario para el entrenamiento de los modelos de aprendizaje supervisado usados en el segundo metodo desarrollado, entre las

características se encuentran valores del modelo de color HSV, brillo y 13 características Haralick para la textura.

Se llevó a cabo una exhaustiva revisión bibliográfica acerca de los modelos de transfer learning y aprendizaje supervisado utilizados para clasificar frutas. Este proceso permitió identificar los modelos más utilizados y reconocidos en este ámbito, entre los cuales se destacaron entre los modelos de transfer learning: VGG-19, VGG-16, ResNet-50, MobileNetV2 y AlexNet. Por parte de los modelos de aprendizaje supervisado: regresión logística, random forest, SVM y Naive bayes.

Para los modelos de transfer learning, se realizaron modificaciones en las últimas cinco capas de cada uno de los modelos mencionados, con el objetivo de adaptarlos al problema específico de detección de la madurez de la fresa.

Posteriormente, se procedió a implementar y ejecutar todos los modelos, llevando a cabo pruebas exhaustivas y obteniendo resultados significativos. Dichos resultados indicaron que los modelos de aprendizaje supervisado del segundo método logran mejores resultados, siendo la regresión logística y random forest quienes logran una precisión del 100% con datos de validación, en un corto periodo de entrenamiento. Por otra parte, aunque entre los modelos de transferencia de aprendizaje del primer método, el modelo ResNet-50 fue el mejor con 97.66%. Esto puede deberse a la elección de características extraídas con las que fueron entrenados los modelos de aprendizaje supervisado, quienes tienen una correlación alta con las etiquetas de madurez.

4.2. Recomendaciones

Considerar la utilización de una base de datos amplia y diversa que incluya imágenes de fresas en distintos niveles de madurez, adquiridas bajo condiciones

controladas. Esto garantizará la obtención de resultados más sólidos y representativos.

Explorar las técnicas de transferencia de aprendizaje, es una propuesta interesante al ser modelos pre entrando, necesitando pequeños conjuntos de datos para entrenarse. Por otro lado, una acertada elección de características puede mejorar enormemente la precisión en la clasificación objetivo. Ambos enfoques pueden lograr un alto nivel de precisión si se logra un entrenamiento adecuado de los modelos utilizados

Evaluar el impacto del tamaño del conjunto de entrenamiento en el rendimiento de los modelos. Se recomienda emplear técnicas de aumento de datos para incrementar la variabilidad de las imágenes y prevenir el sobreajuste.

Identificar posibles áreas de mejora y líneas de investigación futuras en el campo de la clasificación automática de fresas, como la incorporación de métodos de aprendizaje activo o el uso de técnicas de fusión de datos multimodales.

Proporcionar recomendaciones prácticas para la implementación de los modelos seleccionados en entornos reales de producción de fresas, teniendo en cuenta factores como la infraestructura requerida y el entrenamiento del personal.

5. REFERENCIAS

- [1] “Las 5 ventajas para automatizar los procesos industriales - Inser Robótica S.A.” <https://www.inser-robotica.com/las-5-ventajas-para-automatizar-los-procesos-industriales/> (consultado el 22 de abril de 2023).
- [2] “Fresa peruana cierra positivamente el 2023, pero evidencia una gran necesidad de cambio varietal,” Fresh Fruit - Expertos en inteligencia comercial, 2023. <https://freshfruit.pe/2024/01/14/fresa-peruana-cierra-positivamente-el-2023-pero-evidencia-una-gran-necesidad-de-cambio-varietal/> (accessed Jul. 12, 2024).
- [3] S. Maza, “Estudio de la Fresa en el Perú y el Mundo”, *Ministerio de Agricultura y Riego*, 2008, Consultado: el 22 de abril de 2023. [Online].
Disponible en:
<http://repositorio.midagri.gob.pe:80/jspui/handle/20.500.13036/451>
- [4] A. Quintas, “Propiedades y beneficios de las fresas.”, el 23 de abril de 2016.
https://cadenaser.com/programa/2016/04/15/beok/1460756690_097177.html
(consultado el 22 de abril de 2023).
- [5] INSTITUTO TECNOLÓGICO SUPERIOR DE COALCOMÁN,
“MANUAL DE PRODUCCIÓN DE FRESA EN COALCOMÁN MICHOACÁN”,
2018.
<https://www.itscoalcoman.edu.mx/content/descargas/vinculacion/MANUAL%20PARA%20CULTIVO%20DE%20FRESA%20EN%20COALCOMAN.pdf>

(consultado el 22 de abril de 2023).

- [6] Z. Gao, Y. Shao, G. Xuan, Y. Wang, Y. Liu, y X. Han, “Real-time hyperspectral imaging for the in-field estimation of strawberry ripeness with deep learning”, *Artificial Intelligence in Agriculture*, vol. 4, pp. 31–38, ene. 2020, doi: <https://doi.org/10.1016/j.aiia.2020.04.003>
- [7] S. K. Behera, A. K. Rath, y P. K. Sethy, “Maturity status classification of papaya fruits based on machine learning and transfer learning approach”, *Information Processing in Agriculture*, vol. 8, núm. 2, pp. 244–250, jun. 2021, doi: <https://doi.org/10.1016/j.inpa.2020.05.003>
- [8] N. Begum y M. K. Hazarika, “Maturity detection of tomatoes using transfer learning”, *Measurement: Food*, vol. 7, p. 100038, sep. 2022, doi: <https://doi.org/10.1016/j.meafao.2022.100038>
- [9] C. B. MacEachern, T. J. Esau, A. W. Schumann, P. J. Hennessy, y Q. U. Zaman, “Detection of fruit maturity stage and yield estimation in wild blueberry using deep learning convolutional neural networks”, *Smart Agricultural Technology*, vol. 3, p. 100099, feb. 2023, doi: <https://doi.org/10.1016/j.atech.2022.100099>
- [10] S. Sohaib *et al.*, “Mango maturity classification instead of maturity index estimation: A new approach towards handheld NIR spectroscopy”, *Infrared Phys Technol*, vol. 115, p. 103639, jun. 2021, doi: <https://doi.org/10.1016/j.infrared.2021.103639>
- [11] R. Azadnia y K. Kheiralipour, “Evaluation of hawthorns maturity level by

developing an automated machine learning-based algorithm”, *Ecol Inform*, vol. 71, p. 101804, nov. 2022, doi: <https://doi.org/10.1016/j.ecoinf.2022.101804>

[12] X. Zhou, W. S. Lee, Y. Ampatzidis, Y. Chen, N. Peres, y C. Fraise, “Strawberry Maturity Classification from UAV and Near-Ground Imaging Using Deep Learning”, *Smart Agricultural Technology*, vol. 1, p. 100001, dic. 2021, doi: <https://doi.org/10.1016/j.atech.2021.100001>

[13] B. Harel, Y. Parmet, y Y. Edan, “Maturity classification of sweet peppers using image datasets acquired in different times”, *Comput Ind*, vol. 121, p. 103274, oct. 2020, doi: <https://doi.org/10.1016/j.compind.2020.103274>

[14] N. El-Bendary, E. El Hariri, A. E. Hassanien, y A. Badr, “Using machine learning techniques for evaluating tomato ripeness”, *Expert Syst Appl*, vol. 42, núm. 4, pp. 1892–1905, mar. 2015, doi: <https://doi.org/10.1016/j.eswa.2014.09.057>

[15] N. Begum y M. K. Hazarika, “Maturity detection of tomatoes using transfer learning”, *Measurement: Food*, vol. 7, p. 100038, sep. 2022, doi: <https://doi.org/10.1016/j.meaf.2022.100038>

[16] H. Khosravi, S. I. Saedi, y M. Rezaei, “Real-time recognition of on-branch olive ripening stages by a deep convolutional neural network”, *Sci Hortic*, vol. 287, p. 110252, sep. 2021, doi: <https://doi.org/10.1016/j.scienta.2021.110252>

[17] Y. Zhang y L. Wu, “Classification of Fruits Using Computer Vision and a Multiclass Support Vector Machine,” *Sensors*, vol. 12, no. 9, pp. 12489–12505, Sep. 2012, doi: <https://doi.org/10.3390/s120912489>.

- [18] S. Ghazal, W. S. Qureshi, U. S. Khan, J. Iqbal, N. Rashid, and M. I. Tiwana, "Analysis of visual features and classifiers for Fruit classification problem," *Computers and Electronics in Agriculture*, vol. 187, pp. 106267–106267, Jun. 2021, doi: <https://doi.org/10.1016/j.compag.2021.106267>.
- [19] R Robi, and U. Hafidz, "Comparison of Tree Method, Support Vector Machine, Naïve Bayes, and Logistic Regression on Coffee Bean Image," *EMITTER International Journal of Engineering Technology*, vol. 9, no. 1, pp. 126–136, Jun. 2021, doi: <https://doi.org/10.24003/emitter.v9i1.536>.
- [20] D. Gerdan, A. Beyaz, y M. Vatandas., "Elma Çeşitlerinin Sınıflandırılması: H2O Tabanlı Kolektif Öğrenme ve Naive Baye," *Journal of Agricultural Faculty of Gaziosmanpaşa University*, vol. 37, no. 2020–1, pp. 9–16, Jan. 2020, doi: <https://doi.org/10.13002/jafag4646>.
- [21] "Penetrómetros, para determinar la dureza de la fruta" Poscosecha.com, 2024. <https://www.poscosecha.com/decco-iberica/penetrometros-para-determinar-la-dureza-de-la-fruta> (accessed Nov. 04, 2024).
- [22] L. Castelões, T. De Oliveira, y Y. Franco, "Conversores de parámetros del color a parámetros sonoros cuantificables usando los sistemas RGB, HSV y CMYK Converters from Color Parameters to Quantifiable Sound Parameters Using RGB, HSV, and CMYK Color Systems." Accessed: Nov. 03, 2024. [Online]. Available: https://d1wqtxts1xzle7.cloudfront.net/37682392/Conversores_de_parametros_da_cor_para_parametros_sonoros_-_versao_com_titulo_mais_curto_idem_8-libre.pdf?1432068793=&response-content-

[disposition=inline%3B+filename%3DConversores de parametros del color a p.pdf&Expires=1730680722&Signature=JiGkibqoa5pZyh7QYKRii8MQueDca~ekHXk-zFrRGPejhbIIDjVrJwQc~SftvsFXXpo9~rl71sRg9Pia~LOAtB1lgVLRclghYYW7lB-mHhTCOtJMqZ9Mq7-CjnPGomkH4Whp1RYAdDUt4Q1ST~QGoSe8bcDyGzALx6zQGrX7wN9XFtKe1SrNdXiCyt7T2ZHN1CllwnbBKuq3QWFiLYTNz2sHqZ-8LycKpVd5vsqgENZbUvMImT7OU2Cxz7fNt5WcEaW~6RDT3VDLfQDftla0YwO2D1GWLvhS9T6F~imRmoh1GnK7C95RxpIRKdkXedYbWaCevaNJUe9uDN95ggKbnQ_ &Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA](https://disposition=inline%3B+filename%3DConversores+de+parametros+del+color+a+p.pdf&Expires=1730680722&Signature=JiGkibqoa5pZyh7QYKRii8MQueDca~ekHXk-zFrRGPejhbIIDjVrJwQc~SftvsFXXpo9~rl71sRg9Pia~LOAtB1lgVLRclghYYW7lB-mHhTCOtJMqZ9Mq7-CjnPGomkH4Whp1RYAdDUt4Q1ST~QGoSe8bcDyGzALx6zQGrX7wN9XFtKe1SrNdXiCyt7T2ZHN1CllwnbBKuq3QWFiLYTNz2sHqZ-8LycKpVd5vsqgENZbUvMImT7OU2Cxz7fNt5WcEaW~6RDT3VDLfQDftla0YwO2D1GWLvhS9T6F~imRmoh1GnK7C95RxpIRKdkXedYbWaCevaNJUe9uDN95ggKbnQ_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA)

[23] A. Sharda, "Algoritmos de procesamiento de imágenes: ajuste del contraste y el brillo de la imagen," *Hackernoon.com*, Jan. 24, 2021.

<https://hackernoon.com/lang/es/algoritmos-de-procesamiento-de-imagen-ajuste-de-contraste-y-brillo-de-imagen-0y4y318a>

[24] A. Humeau-Heurtier, "Texture Feature Extraction Methods: A Survey," *IEEE Access*, vol. 7, pp. 8975–9000, Jan. 2019, doi:

<https://doi.org/10.1109/access.2018.2890743>

[25] L. Coelho, "mahotas Documentation Release 1.0," 2013. Accessed: Nov. 03, 2024. [Online]. Available:

https://mahotas.readthedocs.io/_downloads/en/release-1.0/pdf/

[26] R. López, "Visión por computadora - Libro online de IAAR", 2023.

<https://iaarbook.github.io/vision-por-computadora/> (consultado el 24 de abril de 2023).

[27] J. Ponce *et al.*, *Inteligencia Artificial*, 1a ed. 2014. Consultado: el 24 de abril de 2023. [Online]. Disponible en:

https://www.researchgate.net/publication/269466259_Inteligencia_Artificial

[28] M. Bramer, "Principles of Data Mining," *SpringerLink*, 2020, doi:

<https://doi.org/10.1007-978-1-4471-7493-6>

[29] "Python for Data Analytics, Scientific and Technical Applications," *IEEE Conference Publication*, 2019. [Online]. Available:

<https://ieeexplore.ieee.org/document/8701341>

[30] A. Valdez, "INTRODUCCIÓN AL MACHINE LEARNING", may 2018,

Consultado: el 24 de abril de 2023. [Online]. Disponible en:

https://www.researchgate.net/publication/338518560_INTRODUCCION_AL_MACHINE_LEARNING

[31] Team redac, "Deep Learning o Aprendizaje profundo : ¿qué es?", 2023.

<https://datascientest.com/es/deep-learning-definicion> (consultado el 24 de abril de 2023).

[32] IBM, "Aprendizaje por transferencia," *Ibm.com*, Jan. 26, 2024.

<https://www.ibm.com/es-es/topics/transfer-learning> (accessed Nov. 04, 2024).

[33] Daniel, "VGG: ¿Qué es este modelo? ¡Daniel te lo cuenta todo!,"

Formación en ciencia de datos | DataScientest.com, Apr. 06, 2022.

<https://datascientest.com/es/vgg-que-es-este-modelo-daniel-te-lo-cuenta-todo>

(accessed Nov. 04, 2024).

- [34] K. Team, “Keras documentation: VGG16 and VGG19,” *Keras.io*, 2015.
<https://keras.io/api/applications/vgg/> (accessed Nov. 04, 2024).
- [35] I. Constantin, J. Constantin, y A. Bigand, “On the Use of Deep Active Semi-Supervised Learning for Fast Rendering in Global Illumination”, *Journal of Imaging 2020, Vol. 6, Page 91*, vol. 6, núm. 9, p. 91, sep. 2020, doi:
<https://doi.org/10.3390/jimaging6090091>
- [36] The MathWorks, “ResNet-50 convolutional neural network - MATLAB resnet50”. <https://www.mathworks.com/help/deeplearning/ref/resnet50.html>
(consultado el 25 de abril de 2023).
- [37] S. Mukherjee, “The Annotated ResNet-50. ”, el 18 de agosto de 2022.
<https://towardsdatascience.com/the-annotated-resnet-50-a6c536034758>
(consultado el 25 de abril de 2023).
- [38] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, y L. C. Chen,
“MobileNetV2: Inverted Residuals and Linear Bottlenecks”, Proceedings of the
IEEE Computer Society Conference on Computer Vision and Pattern
Recognition, pp. 4510–4520, ene. 2018, doi: [10.1109/CVPR.2018.00474](https://doi.org/10.1109/CVPR.2018.00474)
- [39] R. R. Abril, “AlexNet y clasificación de imágenes • Un artículo de La
Máquina Oráculo,” *La Máquina Oráculo*, Jul. 10, 2021.
<https://lamaquinaoraculo.com/deep-learning/alexnet/> (accessed Nov. 04, 2024).
- [40] IBM, “supervised learning,” *Ibm.com*, May 10, 2024.
<https://www.ibm.com/es-es/topics/supervised-learning> (accessed Nov. 04,
2024).

[41] C. C. Rodriguez, "Maquina de Soporte Vectorial (SVM) - César Chique Rodriguez - Medium," *Medium*, Sep. 02, 2020.

<https://medium.com/@csarchiquerodriguez/maquina-de-soporte-vectorial-svm-92e9f1b1b1ac> (accessed Nov. 04, 2024).

[42] IBM, "Naive Bayes," *Ibm.com*, May 10, 2024. [https://www.ibm.com/es-](https://www.ibm.com/es-es/topics/naive-bayes#:~:text=El%20clasificador%20Naive%20Bayes%20es,para%20realizar%20tareass%20de%20clasificaci%C3%B3n)

[es/topics/naive-](https://www.ibm.com/es-es/topics/naive-bayes#:~:text=El%20clasificador%20Naive%20Bayes%20es,para%20realizar%20tareass%20de%20clasificaci%C3%B3n)

[bayes#:~:text=El%20clasificador%20Naive%20Bayes%20es,para%20realizar](https://www.ibm.com/es-es/topics/naive-bayes#:~:text=El%20clasificador%20Naive%20Bayes%20es,para%20realizar%20tareass%20de%20clasificaci%C3%B3n)

[%20tareass%20de%20clasificaci%C3%B3n](https://www.ibm.com/es-es/topics/naive-bayes#:~:text=El%20clasificador%20Naive%20Bayes%20es,para%20realizar%20tareass%20de%20clasificaci%C3%B3n) (accessed Nov. 04, 2024).

[43] IBM, "¿Qué es la regresión logística? | IBM," *Ibm.com*, Jul. 13, 2022.

<https://www.ibm.com/es-es/topics/logistic-regression> (accessed Nov. 04, 2024).

[44] IBM, "Árboles de decisión," *Ibm.com*, Oct. 20, 2022.

[https://www.ibm.com/es-es/topics/decision-](https://www.ibm.com/es-es/topics/decision-trees#:~:text=Un%20%C3%A1rbol%20de%20decisi%C3%B3n%20es,nodos%20internos%20y%20nodos%20hoja)

[trees#:~:text=Un%20%C3%A1rbol%20de%20decisi%C3%B3n%20es,nodos%](https://www.ibm.com/es-es/topics/decision-trees#:~:text=Un%20%C3%A1rbol%20de%20decisi%C3%B3n%20es,nodos%20internos%20y%20nodos%20hoja)

[20internos%20y%20nodos%20hoja](https://www.ibm.com/es-es/topics/decision-trees#:~:text=Un%20%C3%A1rbol%20de%20decisi%C3%B3n%20es,nodos%20internos%20y%20nodos%20hoja) (accessed Nov. 04, 2024).

[45] AWS, "¿Qué es una red neuronal? - Explicación de las redes neuronales artificiales - AWS", 2023. <https://aws.amazon.com/es/what-is/neural-network/>

(consultado el 24 de abril de 2023).

[46] J. Chen, M. Zhou, D. Zhang, H. Huang, y F. Zhang, "Quantification of water inflow in rock tunnel faces via convolutional neural network approach", *Autom Constr*, vol. 123, mar. 2021, doi:

<https://doi.org/10.1016/j.autcon.2020.103526>

<https://doi.org/10.1016/j.autcon.2020.103526>

[47] J. Torres Viñals, *DEEP LEARNING INTRODUCCIÓN PRÁCTICAS CON*

KERAS, 1a ed. Barcelona: Independently published, 2018. Consultado: el 1 de mayo de 2023. [Online]. Disponible en: <https://torres.ai/redes-neuronales-recurrentes/>

[48] Rubén. Rodríguez, “Redes Neuronales Recurrentes.”, el 7 de mayo de 2021. <https://lamaquinaoraculo.com/computacion/redes-neuronales-recurrentes/> (consultado el 2 de mayo de 2023).

[49] UNIR, “Algoritmo backpropagation: funciones y aplicaciones.”, el 17 de febrero de 2023. <https://www.unir.net/ingenieria/revista/backpropagation/> (consultado el 1 de mayo de 2023).

[50] D. Calvo, “Multilayer Perceptron - Neural Network”, el 8 de diciembre de 2018. <https://www.diegocalvo.es/perceptron-multicapa/> (consultado el 24 de abril de 2023).

[51] KeepCoding, “Tipos de capas de red neuronal convolucional”, el 13 de enero de 2023. https://keepcoding.io/blog/tipos-capas-red-neuronal-convolucional/#Capas_de_normalizacion (consultado el 24 de abril de 2023).

[53] D. Calvo, “Red Neuronal Convolucional CNN”, el 20 de julio de 2017. <https://www.diegocalvo.es/red-neuronal-convolucional/> (consultado el 24 de abril de 2023).

[53] Y. Noema, “All About Pooling Layers for Convolutional Neural Networks (CNN)”, el 26 de noviembre de 2021. <https://medium.com/imagescv/all-about-pooling-layers-for-convolutional-neural-networks-cnn-c4bca1c35e31>

(consultado el 24 de abril de 2023).

[54] L. Guarda, “DESARROLLO DE MODELO PARA LA IDENTIFICACIÓN DE SOMNOLENCIA BASADO EN REDES NEURONALES CONVOLUCIONALES Y PROCESAMIENTO DE SEÑALES DE ELECTROENCEFALOGRAFÍA”, Universidad de Chile, Santiago de Chile, 2018. Consultado: el 24 de abril de 2023. [Online]. Disponible en:

<https://repositorio.uchile.cl/bitstream/handle/2250/167873/Desarrollo-de-modelo-para-la-identificaci%C3%B3n-de-somnolencia-basado-en-redes-neuronales-convolucionales.pdf?sequence=1>

[55] D. Unzueta, “Convolutional Layers vs Fully Connected Layers ”, el 13 de noviembre de 2021. <https://towardsdatascience.com/convolutional-layers-vs-fully-connected-layers-364f05ab460b> (consultado el 24 de abril de 2023).

[56] Keras, “keras/batch_normalization.py at v2.10.0”, el 22 de junio de 2022. https://github.com/keras-team/keras/blob/v2.10.0/keras/layers/normalization/batch_normalization.py#L1265 (consultado el 25 de abril de 2023).

[57] D. Calvo, “Función de activación - Redes neuronales ”, el 7 de diciembre de 2018. <https://www.diegocalvo.es/funcion-de-activacion-redes-neuronales/> (consultado el 25 de abril de 2023).

[58] J. Calvo Martin, “La importancia de las funciones de activación en una red neuronal”, el 18 de agosto de 2022. <https://www.linkedin.com/pulse/la-importancia-de-las-funciones-activaci%C3%B3n-en-una-red-calvo-martin/?originalSubdomain=es> (consultado el 1 de mayo de 2023).

- [59] T. De Madera, “Función Softmax”, el 17 de mayo de 2019.
<https://deepai.org/machine-learning-glossary-and-terms/softmax-layer>
(consultado el 6 de mayo de 2023).
- [60] J. Roth, R. West, y G. Ericson, “Función de funciones de pérdida (MicrosoftML)”, el 3 de marzo de 2023. <https://learn.microsoft.com/es-es/sql/machine-learning/r/reference/microsoftml/loss?view=sql-server-ver16>
(consultado el 25 de abril de 2023).
- [61] A. P. Engelbrecht, “Appendix A: Optimization Theory”, *Comput Intell*, pp. 551–579, oct. 2007, doi: [10.1002/9780470512517.app1](https://doi.org/10.1002/9780470512517.app1)
- [62] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” *arXiv.org*, 2014. <https://arxiv.org/abs/1412.6980> (accessed Nov. 04, 2024).
- [63] L. Velasco, “Optimizadores en redes neuronales profundas: un enfoque práctico.”, el 26 de abril de 2020.
<https://velascoluis.medium.com/optimizadores-en-redes-neuronales-profundas-un-enfoque-pr%C3%A1ctico-819b39a3eb5> (consultado el 5 de mayo de 2023).
- [64] J. Barrios, “La matriz de confusión y sus métricas ”, el 26 de julio de 2019.
<https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/>
(consultado el 25 de abril de 2023).
- [65] J. A. Martínez-Heras, “IArtificial.net,” *ResearchGate*, Sep. 2023, doi: <https://doi.org/10.13140/RG.2.2.16587.77609>
- [66] J. Mejía, “Apuntes de Procesamiento Digital de Imágenes.”, San Luis

Potosí, ene. 2005. Consultado: el 25 de abril de 2023. [Online]. Disponible en:
<http://laurence.com.ar/artes/comun/Apuntes%20procesamiento%20digital%20de%20imagenes.pdf>

[67] M. Forero, “Procesamiento Digital de Imágenes ”.

[68] M. Alarcón, “LOS CANALES DE COLOR ”, el 22 de enero de 2022.
<https://ve.bebee.com/producer/los-canales-de-color> (consultado el 25 de abril de 2023).

[69] C. Cattaneo, L. Larcher, A. Ruggeri, A. C. Herrera, y E. M. Biasoni, “Métodos de Umbralización de Imágenes Digitales Basados en Entropía de Shannon y Otros,” 2015.
<https://www.semanticscholar.org/paper/M%C3%A9todos-de-Umbralizaci%C3%B3n-de-Im%C3%A1genes-Digitales-en-y-Cattaneo-Larcher/b7ac1f2dc47a54bc885002a1ba67a0346267d992> (accessed Nov. 04, 2024).

[70] A. García, y F. Leal, “Tratamiento de imágenes utilizando el algoritmo OTSU sobre FPGA | Archivo Digital UPM,” Oa.upm.es, Jul. 2019, doi:
<https://oa.upm.es/56189/>

[71] J. Vidal, “Interpolación de formas en imágenes usando morfología matemática | Archivo Digital UPM,” Oa.upm.es, Sep. 2008, doi:
<https://oa.upm.es/1282/>

[72] F. González, “¿Qué Es Un Paradigma? Análisis Teórico, Conceptual Y Psicolingüístico Del Término”, *Investigación y Postgrado*, vol. 20, núm. 1, pp.

13–54, 2005, Consultado: el 1 de mayo de 2023. [Online]. Disponible en:

http://ve.scielo.org/scielo.php?script=sci_arttext&pid=S1316-00872005000100002&lng=es&nrm=iso&tlng=es

[73] J. E. Pérez, y F. Restrepo, “Teorías cognitivas dinámicas o de tercera generación: análisis documental de artículos originales de investigación cualitativa,” *Diversitas*, vol. 18, no. 1, Jan. 2022, doi:

<https://doi.org/10.15332/22563067.7875>

[74] M. Baig, L. N. Thalluri, R. Devireddy, C. Priyanka, y R. Sulthana, “An Image Processing Approach for Grading of Mangoes based on Maturity”, *2020 International Conference on Computational Performance Evaluation, ComPE 2020*, pp. 512–516, jul. 2020, doi: [10.1109/ComPE49325.2020.9200114](https://doi.org/10.1109/ComPE49325.2020.9200114)

[75] R. Saragih y A. Emanuel, “Banana Ripeness Classification Based on Deep Learning using Convolutional Neural Network”, *3rd 2021 East Indonesia Conference on Computer and Information Technology, EIConCIT 2021*, pp. 85–89, abr. 2021, doi: [10.1109/EIConCIT50028.2021.9431928](https://doi.org/10.1109/EIConCIT50028.2021.9431928)

[76] Z. Su *et al.*, “Application of Hyperspectral Imaging for Maturity and Soluble Solids Content Determination of Strawberry With Deep Learning Approaches”, *Front Plant Sci*, vol. 12, p. 1897, sep. 2021, doi:

<https://www.frontiersin.org/journals/plant-science/articles/10.3389/fpls.2021.736334/full>

[77] D. Worasawate, P. Sakunasinha, y S. Chiangga, “Automatic Classification of the Ripeness Stage of Mango Fruit Using a Machine Learning Approach”,

- AgriEngineering* 2022, Vol. 4, Pages 32-47, vol. 4, núm. 1, pp. 32–47, ene. 2022, doi: <https://doi.org/10.3390/agriengineering4010003>
- [78] Z. Song, L. Fu, J. Wu, Z. Liu, R. Li, y Y. Cui, “Kiwifruit detection in field images using Faster R-CNN with VGG16”, *IFAC-PapersOnLine*, vol. 52, núm. 30, pp. 76–81, ene. 2019, doi:<https://doi.org/10.1016/j.ifacol.2019.12.500>
- [79] G. Moreira, S. A. Magalhães, T. Pinho, F. Neves, y M. Cunha, “Benchmark of Deep Learning and a Proposed HSV Colour Space Models for the Detection and Classification of Greenhouse Tomato”, *Agronomy* 2022, Vol. 12, Page 356, vol. 12, núm. 2, p. 356, ene. 2022, doi: <https://doi.org/10.3390/agronomy12020356>
- [80] Y. Tian, G. Yang, Z. Wang, H. Wang, E. Li, y Z. Liang, “Apple detection during different growth stages in orchards using the improved YOLO-V3 model”, *Comput Electron Agric*, vol. 157, pp. 417–426, feb. 2019, doi: <https://doi.org/10.1016/j.compag.2019.01.012>
- [81] G. Kumar, “Logistic Regression to classify Fruits Image Dataset”, in *Machine Learning Engineer Intern. AI Tech Systems*.Chennai, India, 2019. pp.3. doi: [10.13140/RG.2.2.35036.74882](https://doi.org/10.13140/RG.2.2.35036.74882)
- [82] J. De Jesus., “A method with neural networks for the classification of fruits and vegetables,” *Soft Computing*, vol. 21, no. 23, pp. 7207–7220, Jul. 2016, doi: <https://doi.org/10.1007/s00500-016-2263-2>

ANEXOS



ANEXO 01: ACTA DE REVISIÓN DE SIMILITUD DE LA INVESTIGACIÓN

Yo **Mejia Cabrera Heber Ivan** docente del curso de **Investigación II** del Programa de Estudios de **Ingeniería de Sistemas** y revisor de la investigación del (los) estudiante(s), **SANCHEZ PARDO SAMUEL, MORE VILLEGAS FIORELLA JHAJIRA**, titulada:

DESARROLLO DE MÉTODOS PARA LA CLASIFICACIÓN POR MADUREZ DE LA FRESA UTILIZANDO PROCESAMIENTO DE IMÁGENES DIGITALES Y MACHINE LEARNING.

Se deja constancia que la investigación antes indicada tiene un índice de similitud del **18%**, verificable en el reporte final del análisis de originalidad mediante el software de similitud TURNITIN. Por lo que se concluye que cada una de las coincidencias detectadas no constituyen plagio y cumple con lo establecido en la Directiva sobre índice de similitud de los productos académicos y de investigación en la Universidad Señor de Sipán S.A.C., aprobada mediante Resolución de Directorio N° 145-2022/PD-USS.

En virtud de lo antes mencionado, firma:

MEJIA CABRERA HEBER IVAN	DNI: 41639565	
---------------------------------	---------------	---

Pimentel, 19 de Julio de 2023.

ANEXO 02: ACTA DE APROBACIÓN DEL ASESOR

Yo **MEJIA CABRERA HEBER IVAN**, quien suscribe como asesor designado mediante Resolución de Facultad N° N° **760-2022/FIAU-USS**, del proyecto de investigación titulado **DESARROLLO DE MÉTODOS PARA LA CLASIFICACIÓN POR MADUREZ DE LA FRESA UTILIZANDO PROCESAMIENTO DE IMÁGENES DIGITALES Y MACHINE LEARNING.**, desarrollado por el(los) estudiante(s): **SANCHEZ PARDO SAMUEL, MORE VILLEGAS FIORELLA JHAJAIRA.**, del programa de estudios de **Ingeniería de Sistemas**, acreditó haber revisado, realizado observaciones y recomendaciones pertinentes, encontrándose expedito para su revisión por parte del docente del curso.

En virtud de lo antes mencionado, firman:

MEJIA CABRERA HEBER IVAN	DNI: 41639565	
SANCHEZ PARDO SAMUEL	DNI: 75448908	
MORE VILLEGAS FIORELLA JHAJAIRA	DNI: 73329403	

Pimentel, 13 de Julio de 2023.

Anexo 03

Tabla 29 Modelos de machine learning usados en otras investigaciones.

Año	Autores	Modelo	Objeto de estudio	Precisión
2022	X. Zhou et al.[12]	ANN	fresas	99,57%
2020	Z. Gao et al. [6]	AlexNet	fresas	98,6%
2020	M. Baig et al.[74]	MSVM	Mango	98,0%
2022	N. Begum & M. Hazarik [15]	VGG 19	tomate	97,37%
2021	R. Saragih y A. Emanuel. [75]	MobileNet V2	plátano	96,18%
2022	Z. Su et al.[76]	ResNet-50	fresas	96,14%
2022	D. Worasawate et al. [77]	FANN	mango	89,6%
2021	S. Sohaib et al [10]	KNN	mango	88,2%
2019	Z. Song et al [78]	VGG 16	kiwi	87,61 %
2022	G. Moreira et al. [79]	YOLO V4	tomate	85,81%
2019	Y. Tian et al. [80]	YOLOV3-dense	manzana	81,7%
2021	S. Ghazal et al. [18]	SVM	Varias frutas	99%
2020	D. Gerdan et al. [20]	Random Forest	Manzana	98.4%
2020	D. Gerdan et al. [20]	Naive Bayes.	Manzana	100%
2019	G. Kumar. [81]	Regresión logística.	Varias frutas	100%
2016	J. De Jesus. [82]	Red Neuronal	Varias frutas	100%