



UNIVERSIDAD
DE PIURA

FACULTAD DE INGENIERÍA

**Detección de nutrientes del suelo y planta, y pestes en
campos de cultivo de banano orgánico con *Machine
Learning***

Trabajo de Investigación para optar el Grado de
Bachiller en Ingeniería Mecánico - Eléctrica

**Renato Edgardo Castañeda Valdivieso
José Renato Guerrero Meza
Bruno Eduardo Renteros Parra
José Alejandro Villanueva Mejía**

Asesor:
Dr. Ing. William Ipanaqué Alama

Piura, marzo de 2021



Resumen

El presente trabajo facilita el seguimiento de la composición nutricional del suelo de cultivo mediante la identificación de deficiencias de nutrientes a través del reconocimiento de imágenes de hojas de banano utilizando una red neuronal convolucional entrenada con “transfer learning” y “fine tuning”.

En esta investigación se utilizó un conjunto de datos original de fotografías, que se compone de imágenes de hojas de banano sanas y hojas con deficiencias conocidas de nitrógeno, potasio y fósforo. Así mismo para la detección de plagas se utilizó un conjunto de fotografías público, el cual fue encontrado en un repositorio y se escogieron aquellas plagas que afectaban al banano. Posteriormente, se realizó un aumento a este conjunto de fotografías mediante transformaciones lineales y las imágenes resultantes fueron pre-procesadas en diferentes espacios de color para ser utilizadas como entradas a la red neuronal.

Se logró obtener un modelo con alta precisión que pudo ser validado a través de diferentes métricas. Finalmente, se desarrolló un prototipo de plataforma web para que los agricultores en un futuro pudieran acceder al sistema.

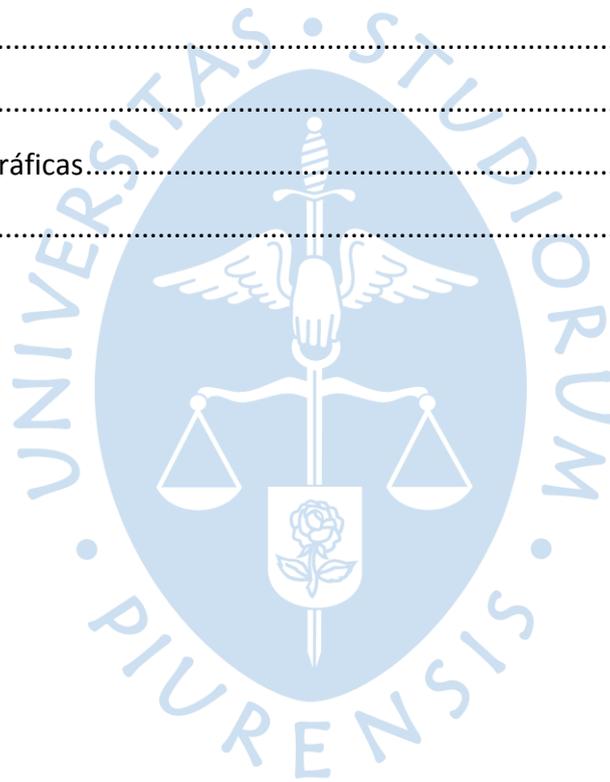


Tabla de contenido

Introducción	13
Capítulo 1.....	15
Generalidades.....	15
1.1 Planteamiento del problema.....	15
1.2 Justificación del estudio.....	15
1.3 Objetivos.....	16
1.3.1 Objetivo general.....	16
1.3.2 Objetivos específicos.....	16
Capítulo 2.....	17
Marco teórico	17
2.1 El banano orgánico.....	17
2.1.1 Fruto.....	18
2.1.2 Siembra.....	18
2.1.3 Fertilización del banano.....	19
2.1.4 Plagas.....	21
2.1.5 Cosecha.....	22
2.1.6 Economía.....	23
2.1.7 Métodos actuales de control en la cadena en el Perú.....	23
2.2 Cultivo de banano en Filipinas.....	24
Capítulo 3.....	25
Estado del arte.....	25
3.1 Investigaciones de carácter científico sobre nutrientes en plantas y plagas.....	25

3.1.1	Diagnostics of macronutrient deficiencies in three legume species	25
3.1.2	Corn Leaf Diseases Diagnostic Techniques Based on Image Recognition	26
3.1.3	Unhealthy region of citrus leaf detection using image processing techniques.	27
3.1.4	Detection and classification of diseases of banana plant using local binary pattern and support vector machine.	28
3.1.5	Multiple nutrient deficiency detection in paddy leaf using color and pattern analysis.	29
3.1.6	Machine Learning regression technique for cotton leaf disease detection and controlling using IoT	30
3.1.7	AI-powered banana diseases and pest detection	31
3.1.8	LeafCheckIT: A Banana Leaf Analyzer for Identifying Macronutrient Deficiency	32
3.1.9	CNN for leaf image-based plant disease classification	33
3.1.10	Identification of Apple Leaf Diseases Based on Deep CNN	34
3.1.11	Acquiring and preprocessing leaf images for automated plant identification: understanding the tradeoff between effort and information gain.	35
3.1.12	Identification of Plant Nutrient Deficiencies Using CNN	36
3.1.13	Crop pest recognition in natural scenes using convolutional neural networks.	37
3.1.14	SMART-Fertilizer	38
3.1.15	Tumaini App	38
Capítulo 4	39
Metodología de la investigación	39
4.1	Procedimientos a emplear para el desarrollo del proyecto.....	39
4.1.1	Design Thinking	39
4.1.2	Adquisición de data	40
4.1.3	Preprocesamiento de imágenes	40
4.1.4	Entrenamiento de CNN	43
4.1.5	Interpretación y validación de resultados	44
4.1.6	Detección de plagas utilizando TensorFlow	45
4.2	Materiales.....	52
4.2.1	Dataset	52
4.2.2	Google Colab	52

4.2.3 Microsoft Visual Studio	53
Capítulo 5.....	55
Resultados	55
5.1 CIELAB.....	56
5.2 HSV.....	59
5.3 YCbCr	62
5.4 Ecuación de histograma	65
5.5 Análisis de Resultados	70
5.6 Plataforma	72
5.7 Plagas.....	73
Conclusiones.....	75
Referencias bibliográficas.....	77
Apéndices	81





Lista de tablas

Tabla 1 Diagnostics of macronutrient deficiencies in three legume species.....	25
Tabla 2 Corn Leaf Diseases Diagnostic Techniques Based on Image Recognition	26
Tabla 3 Unhealthy region of citrus leaf detection using image processing techniques	27
Tabla 4 Detection and classification of diseases of banana plant using local binary pattern and support vector machine.....	28
Tabla 5 Multiple nutrient deficiency detection in paddy leaf using color and pattern analysis	29
Tabla 6 Machine Learning regression technique for cotton leaf disease detection and controlling using IoT.....	30
Tabla 7 AI-powered banana diseases and pest detection.....	31
Tabla 8 A Banana Leaf Analyzer for Identifying Macronutrient Deficiency	32
Tabla 9 CNN for leaf image-based plant classification.....	33
Tabla 10 Identification of Apple Leaf Diseases Based on Deep CNN	34
Tabla 11 Acquiring and preprocessing leaf images for automated plant identification: understanding the tradeoff between effort and information gain	35
Tabla 12 Identification of Plant Nutrient Deficiencies Using CNN	36
Tabla 13 Crop pest recognition in natural scenes using convolutional neural networks ...	37
Tabla 14 SMART-Fertilizer	38
Tabla 15 Tumaini App.....	38
Tabla 16 Comparación de los métodos de preprocesamiento	71



Lista de figuras

Figura 1. Planta de banano y sus partes	17
Figura 2. Banano listo para la exportación.....	18
Figura 3. Imagen original (izquierda) y conversión al espacio de color HSV (derecha)	41
Figura 4. Imagen original (izquierda) y conversión al espacio de color CIELAB (derecha) .	42
Figura 5. Imagen original (superior) y conversión al espacio de color YCbCr (inferior)	43
Figura 6. Imagen original (superior) y conversión al espacio de color (inferior)	44
Figura 7. Matriz de confusión.....	45
Figura 8. Etiquetado de objetos usando LabelImg.....	46
Figura 9. Training loss y accuracy del CIELAB durante el Transfer Learning.....	56
Figura 10. F1-Score, recall, precision, support obtenido durante Transfer Learning.	56
Figura 11. Training loss y accuracy del CIELAB durante el Fine Tuning.	57
Figura 12. F1-Score, recall, precision, support obtenido durante Fine Tuning.....	57
Figura 13. Matriz de confusión del conjunto del entrenamiento.	58
Figura 14. Matriz de confusión del conjunto del testeo.	58
Figura 15. Training loss y accuracy del HSV durante el Transfer Learning	59
Figura 16. F1-Score, recall, precision, support obtenido durante Transfer Learning.	59
Figura 17. Training loss y accuracy del HSV durante el Fine Tuning.	60
Figura 18. F1-Score, recall, precision, support obtenido durante Fine Tuning.....	60
Figura 19. Matriz de confusión del conjunto del entrenamiento.	61
Figura 20. Matriz de confusión del conjunto de validación	61
Figura 21. Matriz de confusión del conjunto del testeo.	61
Figura 22 Training loss y accuracy del YCbCr durante el Transfer Learning.	62
Figura 23 F1-Score, recall, precision, support obtenido durante Transfer Learning.	62

Figura 24 Training loss y accuracy del YCbCr durante el Fine Tuning	63
Figura 25 F1-Score, recall, precision, support obtenido durante Fine Tuning.....	63
Figura 26 Matriz de confusión del conjunto del entrenamiento.	64
Figura 27 Matriz de confusión del conjunto de validación.	64
Figura 28 Matriz de confusión del conjunto del testeo.	64
Figura 29 F1-Gráfica de Training Loss y Accuracy durante Transfer Learning	65
Figura 30 F1-Gráfica de Training Loss y Accuracy durante Fine Tuning.....	65
Figura 31 F1-Score, recall, precision, support obtenido durante Transfer Learning	66
Figura 32 F1-Score, recall, precision, support obtenido durante Fine Tuning.....	66
Figura 33 Matriz de confusión del conjunto de validación.	67
Figura 34 Matriz de confusión del conjunto de entrenamiento.	68
Figura 35 Matriz de confusión del conjunto de testeo.	69
Figura 36 Resultados durante el entrenamiento de la red, en el “Transfer Learning”	69
Figura 37 Resultados durante el entrenamiento de la red, en el “Fine Tuning”	70
Figura 38 Valor de pérdida con Transfer Learning y Fine Tuning.	71
Figura 39 Valor de precisión con Transfer Learning y Fine Tuning.	72
Figura 40 Prototipo de plataforma.....	72
Figura 41 Detección del escarabajo pulga.....	73
Figura 42 Detección de mosca blanca.....	73
Figura 43 Detección de araña roja.	74
Figura 44 Agrupaciones de mosca blanca en distintos puntos de la imagen	74

Introducción

La agronomía es una de las actividades comerciales con mayor crecimiento en el Perú. Uno de los cultivos con creciente importancia es el del banano orgánico, específicamente en el norte del país. En los últimos años los agricultores pequeños, además de empresas con grandes hectáreas, han iniciado el cambio de su mercado objetivo hacia la exportación a los vastos mercados de Asia, Europa y América. Para aumentar el margen de ganancia y poder competir con productores extranjeros, se debe iniciar la implementación de tecnología la cual permita ahorrar o abaratar costos.

La aplicación de técnicas de Machine Learning e IoT para ayudar a los agricultores a facilitar el análisis de nutrientes de las plantaciones (sin recurrir a los análisis en laboratorio), así como la detección de plagas, resultarían en ahorros de tiempo y dinero considerables. Con lo cual se aumentaría la efectividad de la cadena productiva y la competitividad de los productores en el mercado internacional.

El presente informe detalla los objetivos generales y específicos del trabajo académico; describe el estado de la agricultura de banano (incluyendo los requerimientos de la planta) y su cadena productiva; y presenta estudios realizados por investigadores extranjeros en materia de detección de carencias de nutrientes y plagas mediante procesamiento de imágenes que servirán como marco para el desarrollo de nuestro trabajo.



Capítulo 1

Generalidades

1.1 Planteamiento del problema

Para asegurar una buena producción (tanto en calidad, como en cantidad) de banano que cumpla con los estándares de exportación internacional, los agricultores deben realizar análisis de suelos para asegurar que las plantas tengan a disposición los macronutrientes suficientes. Para este análisis se obtienen muestras de varias secciones de las áreas de cultivo, posteriormente se llevan a un laboratorio y se obtiene la composición de nutrientes de las muestras. Además, se debe monitorear estrictamente la aparición y propagación de plagas y enfermedades que puedan afectar al banano orgánico. La falta de un sistema automatizado, que asegure confiabilidad y pueda ser utilizado in situ que detecte y comunique las lecturas de nutrientes del suelo a una base de datos accesible por el agricultor justifica el desarrollo del presente proyecto. Además, se busca que el sistema mencionado también sea capaz de detectar la aparición de plagas en sus etapas iniciales, de tal manera que el agricultor pueda tomar medidas antes de que se afecten los cultivos seriamente.

1.2 Justificación del estudio

Se propone un sistema capaz de detectar deficiencia de nutrientes en la planta de banano y capaz de identificar plagas. Con esto se busca facilitar y simplificar la laboriosa tarea de los agricultores al realizar análisis de suelos (Tejada & Gara, 2017), que comprende desde la recolección de muestras hasta el envío de estas a un laboratorio para recién conocer la composición de nutrientes. Asimismo, la detección e identificación de plagas se propone como una ayuda al agricultor en el reconocimiento de las mismas, y esta información podría ser almacenada en una base de datos para futuro análisis estadístico. Cabe resaltar que lo desarrollado en este informe podría ser utilizado en futuros trabajos con drones, para la detección de nutrientes y de plagas en tiempo real.

1.3 Objetivos

1.3.1 Objetivo general

Diseño una plataforma que sirva de apoyo al agricultor para poder estimar que macronutrientes tiene en deficiencia su planta de banano, con la finalidad de obtener un mejor producto en la cosecha del banano orgánico, así como una posible interfaz para la detección de plagas en el banano.

1.3.2 Objetivos específicos

El desarrollo de una plataforma haciendo uso del aprendizaje profundo y redes neuronales convolucionales para lograr una estimación y predicción de las deficiencias con un alto grado de exactitud. Así mismo hacer uso de la identificación de objetos para poder identificar las plagas que se presenten en el banano y su posible aplicación para automatizar el proceso de la toma de datos y la identificación de las plagas.



Capítulo 2

Marco teórico

2.1 El banano orgánico

El banano es una planta tropical con pseudotallos aéreos sobre los cuales crecen yemas laterales o “hijos”. Sus hojas están dispuestas en forma de espiral y las bases foliares rodean al tallo, formando el pseudotallo. Lo que diferencia al banano orgánico del banano convencional es su cultivo sin ningún tipo de pesticidas (Torres, 2012).

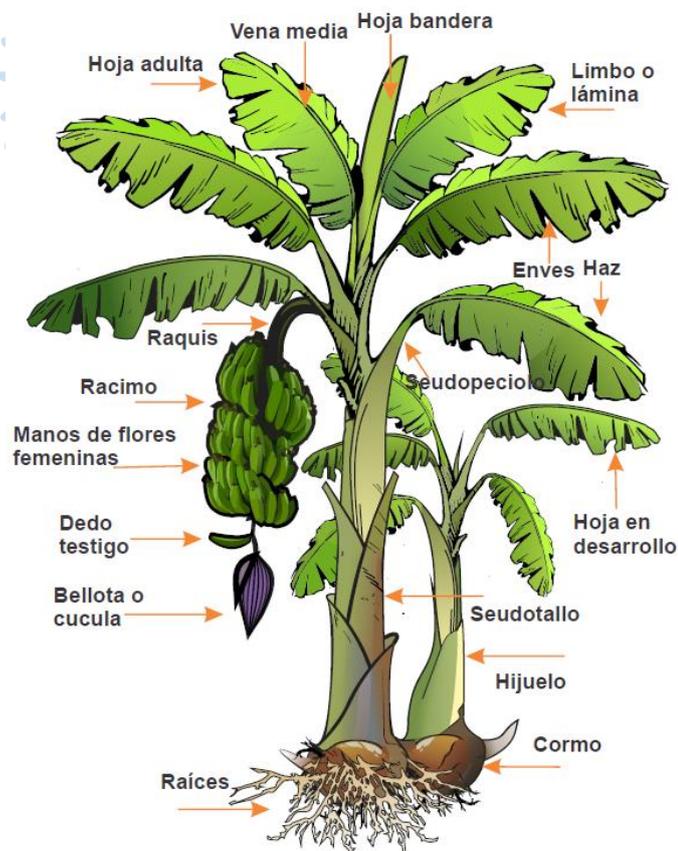


Figura 1. Planta de banano y sus partes

Fuente: Torres, 2012

2.1.1 Fruto

Tiene una longitud promedio de 7.5 pulgadas y un peso aproximado de 200g (Guzmán Velásquez, 2014). Es un fruto suave y carnoso, y está compuesto por tres carpelos que son los últimos órganos florales que aparecen, los cuales se fusionan para formar el estilo y el estigma (Torres, 2012).

Para que la fruta se pueda desarrollar por completo tienen que pasar entre 10 a 13 semanas. En lugares que tienen largos periodos de luminosidad y temperaturas que circunda los 20°C y 26°C, el desarrollo del fruto puede demorar entre 14 a 16 semanas (Torres, 2012).

El peso de la fruta incrementa de manera lineal hasta los tres primeros meses, que es el momento en el que se corta para la exportación. Después, experimenta un aceleramiento en el crecimiento, sobrepasando los límites permisibles para la exportación; a partir de ese momento inicia la maduración (Torres, 2012).



Figura 2. Banano listo para la exportación

Fuente: Torres, 2012

2.1.2 Siembra

Para la siembra se tienen que tomar en cuenta múltiples factores, como la pendiente del terreno, el sistema de explotación, las vías de acceso, etc. La siembra puede ser en cuadro, rectángulo o triángulo, en suelos de muy poca pendiente; sin embargo, si la pendiente es considerable, la siembra debe hacerse en curvas de nivel que protejan al suelo de la erosión (Vargas & Rojas, 2013).

Un sistema apto para explotaciones permanentes suele ser el sistema de doble hilera o surco, sus ventajas son las siguientes:

- Los anchos surcos permiten asociarlo con alguna leguminosa, lo cual supondría ingresos adicionales.
- Su cobertura controla la maleza

2.1.3 Fertilización del banano

La correcta aplicación de aditivos al banano orgánico es de gran importancia ya que estos nos permiten, tener a nuestro cultivo en las mejores condiciones posibles para su correcto crecimiento (Jaramillo, 2013).

Es por eso, que hay que tener en cuenta que para realizar un correcto procedimiento de fertilización es necesario el tener un área destinada para la mezcla y preparación de los fertilizantes a aplicar en los cultivos. Así mismo las dosis a aplicar de los fertilizantes en los cultivos dependerá del estado en el que se encuentre este, esto se podrá saber a partir de un análisis foliar y del suelo propiamente (Jaramillo, 2013).

Así mismo es de gran necesidad tener un plan de fertilización para así asegurar un correcto crecimiento de la planta desde la siembra hasta que se realice la cosecha, el análisis de la raíz y del agua de cultivo también son de gran importancia ya que es en el agua por donde se transportan los aditivos para los cultivos de banano durante el riego, esta es una medida tomada en los últimos años para lograr automatizar el proceso de la fertilización a través del riego.

Dentro de los aditivos usados en la fertilización por riego, tenemos unos de gran importancia que son los macroelementos, en los siguientes subcapítulos se hablará de ellos, los cuales como ya se mencionó antes, son fundamentales para el desarrollo de la planta, siendo: nitrógeno(N), fósforo(P), potasio(K), magnesio (Mg), calcio (Ca) y azufre(S).

2.1.3.1 Nitrógeno. Elemento de esencial importancia en la formación de aminoácidos, ácidos nucleicos, proteínas, etc. Ya que en el banano es de gran importancia el obtener y hacer crecer una planta vigorosa, fruta grande y bien formada. Es considerado un macronutriente que como antes se mencionó permite el crecimiento de un buen fruto lo cual permite que este logre cumplir con los estándares a los cuales se quiere llegar. Dentro de las dosis requeridas son 350-600 kg N/ha/año (Jaramillo, 2013).

Como parte de la identificación de la deficiencia del nitrógeno en la planta del banano podemos identificar que las hojas se empiezan a tornar en un verde más claro y en severos casos amarillentas. Otra característica de la deficiencia de nitrógeno en los cultivos es el tamaño de los frutos, ya que como antes se mencionó el nitrógeno contribuye a la obtención de frutos grandes que cumplan con los estándares del mercado, por lo que cuando este macronutriente se encuentra en deficiencia se puede asumir que el cultivo tiene deficiencias de este. Un dato importante para considerar es que mientras más arenoso sea el suelo se necesitará más nitrógeno en el suelo de cultivo.

2.1.3.2 Fósforo. Macronutriente que es indispensable al momento de la aplicación a la plantación, pero su importancia decrece después de la tercera cosecha. Una de las funciones del fósforo en la planta es el de conductor de energía, así como, el control de la síntesis de almidones. Otra característica del fósforo en el banano es que la pérdida de la importancia de este es que cosecha tras cosecha el fósforo es pasado de las madres a los hijos, entonces este macronutriente no se da por perdido y su aplicación solo será en pequeñas cantidades y esto pasa de generación a generación (Jaramillo, 2013).

Para identificar la deficiencia de este macronutriente es observando la hoja del banano ya que se suele presenciar clorosis en la hoja de la planta lo cual significaría que se tornaría en un color un poco amarillento que es muy notorio en los bordes de la hoja. Y en casos extremos de deficiencia de este macronutriente ocasionaría que se produzca la necrosis temprana en el tejido de la hoja de la planta, lo cual puede desembocar en la muerte temprana de la hoja de la planta. En cuanto a los requerimientos de la planta están establecidos entre 75 a 150 kg de P/ha.

2.1.3.3 Potasio. Para la nutrición del banano este mineral es de suma importancia ya que debido a su ausencia en la nutrición del banano puede causar frutos delgados, cortos, de muy bajo peso y tenderán a la maduración temprana lo cual sería algo negativo para la exportación de este fruto. (torres 2012). Otra característica importante es que ayuda a mantener a la planta correctamente hidratada y la regulación de la apertura de las estomas (Torres, 2012).

Este macroelemento, debido a su importancia debe ser siempre aplicado ya que cuando se produce la deficiencia de este se puede observar de manera muy evidente en las puntas de las hojas de las plantas y más aún cuando el proceso de cultivo del banano es muy recurrente en la utilización del suelo de cultivo. Una de estas características más importantes es la presentación de clorosis donde las hojas se tornan de un color amarillento o naranja. Sin embargo, lo que más resalta en distintos grados de necrosis ya que avanza de manera regular desde los bordes de las plantas hasta la nervadura de la planta. Otra deficiencia notable es en el fruto final el cual es uno de los daños que ha causado en la industria. Su requerimiento oscila entre 650 a 900 kg de K/ha (Jaramillo, 2013). Característica importante a tomar en cuenta es que el fertilizante debe ser aplicado en la zona que posea máxima absorción.

2.1.3.4 Calcio. Este macronutriente un componente importante de la pared celular así mismo ayuda a reducir la susceptibilidad a las enfermedades, cabe resaltar que la presencia de este en la planta aumenta la duración de la vida verde de la fruta e incrementa la dureza de la epidermis. Este macroelemento es absorbido por las raíces de la planta y es de vital importancia para el desarrollo de esta y su funcionamiento (Haifa, 2014; Tixier et al., 2010).

La deficiencia de este es muy notoria ya que ocasiona una disminución de forma significativa la calidad de la fruta. También cuando existe deficiencia de este macroelemento aparece enanismo general en los frutos de la planta, empieza a reducirse el tamaño de las

hojas, en algunos casos puede ocasionar la pudrición de la planta. Otra característica notoria es la aparición de clorosis y necrosis típica, pero esto antes mencionado solo ocurre en casos severos (Haifa, 2014).

2.1.3.5 Magnesio. Como parte de sus funciones principales al ser un componente crucial en la molécula de la clorofila. Este es requerido en una gran cantidad de enzimas involucradas en la transferencia de fosfatos (Haifa, 2014).

La deficiencia de este se hace notoria de manera visual en la hoja del banano que se manifiesta en una clorosis amarillenta de la zona central de la lámina, a la vez que los bordes del área de la nervadura central permanecen verdes. Es muy común encontrar deficiencia de magnesio en plantaciones viejas en las que se ha aplicado poco de este macronutriente. Otra característica notoria es el pobre crecimiento de la planta y su bajo rendimiento (Haifa, 2014).

2.1.3.6 Azufre. Macronutriente de gran importancia en la formación de proteínas. Así mismo este es requerido para la formación de la clorofila (Haifa, 2014).

Su deficiencia se manifiesta en las hojas reducidas de tamaño y cloróticas, las cuales suelen ser hojas jóvenes y se tornan en un color blanco-amarillento. Es raro que se presente esta deficiencia debido a que este es frecuentemente suministrado en los fertilizantes. En cuanto al rendimiento se podrá notar su disminución, así como la aparición de parches necróticos en los márgenes de las hojas (Haifa, 2014).

2.1.4 Plagas

2.1.4.1 Mosca blanca. La mosca blanca es una plaga que se alimenta de muchas especies vegetales (polífaga), como sandía, melón, pepino, tomate y lechuga, entre otras plantas de hortalizas, succionando la savia de envés de la hoja y provocando un tono amarillo en las hojas, un tamaño menor al esperado y anomalías en el desarrollo del fruto. (SEMINIS, 2016)

Las moscas blancas pican la planta para alimentarse, perforan las células de la hoja y succionan su savia. Al hacer esto, debilitan y decoloran la planta y afectan el desarrollo de los frutos. (SEMINIS, 2016)

2.1.4.2 Escarabajo negro. Picudo negro del plátano. Plaga de importancia en el cultivo del plátano, el daño lo producen las larvas al abrir galerías en el tallo o en el falso tallo, de 6 a 8 mm de diámetro. Es una plaga importante del cultivo del plátano. El picudo adulto es negro y mide 10-15 mm. Vive libremente, aunque es más común encontrarlo entre las vainas foliares, en el suelo en la base de la planta asociado con los residuos del cultivo. (Picudo Negro Del Plátano - EcuRed, n.d.)

Es una plaga muy activa de noche y muy susceptible a la desecación. Los adultos pueden permanecer en la misma mata por largos períodos de tiempo, y sólo una pequeña parte de ellos podrá moverse a una distancia mayor de 25 m durante un período de 6 meses.

Los picudos vuelan raramente. La diseminación ocurre principalmente a través del material de plantación infestado. (Picudo Negro Del Plátano - EcuRed, n.d.)

2.1.4.3 Araña roja. Es una araña roja típica de forma globosa. Se considera una importante plaga en el cultivo de plátanos y bananos en los cuales los daños se centran en las hojas por la actividad de la alimentación, las que luego se van decolorando, posteriormente pasan a color pardo o gris, en ataques severos provoca la defoliación del árbol. La hembra es más grande que el macho y mide alrededor de 0,6 mm de largo, es de color rojo intenso y de forma achatada, mientras que el macho mide 0,3 mm, es amarillo rojizo y de forma triangular. (Ácaro Rojo Del Plátano - EcuRed, n.d.)

Su ciclo de vida, como el de todos los ácaros, es muy corto. Con altas temperaturas y sequía pueden ocurrir de 15 a 17 generaciones por año, encontrándose en un momento determinado todas las fases de desarrollo. El desarrollo masivo de esta plaga se ha observado con temperaturas medias diarias de 23 a 25 °C y ausencia de lluvia o riego. (Ácaro Rojo Del Plátano - EcuRed, n.d.)

2.1.5 Cosecha

La cosecha constituye una de las etapas finales en campo en el cultivo técnico del banano, demanda criterio, labor oportuna a fin de evitar pérdidas en campo. Es un complemento a todo el proceso iniciado desde la siembra. La cosecha oportuna permitirá obtener una fruta de calidad y fresca. El criterio de cosecha depende de varios factores que se deben de tener en cuenta a la hora de proceder, como, por ejemplo: edad de la fruta, calibre y demanda de esta. (AGROBANCO, 2012)

Referente a la edad de la fruta, esto depende al comportamiento fisiológico de la variedad a cultivar y de los parámetros de cosecha que se establezcan. (AGROBANCO, 2012) Luego de haber enfundado los racimos del banano, el número de semanas que transcurre es de 12 pero puede variar dependiendo de las condiciones climáticas.

El calibre permite programar la cosecha a partir de una medida que va relacionado a la distancia del mercado, es así que, cuando más alejado este el calibre de la fruta disminuirá a fin de evitar una maduración. (AGROBANCO, 2012)

A nivel internacional se utiliza como medida un calibrador que está graduado en 32avo de pulgada, con este instrumento se puede calcular el diámetro de la fruta y dar las indicaciones sobre la cosecha. Este calibrador puede ser fijo o automático. Finalmente, el calibre de la cosecha de la fruta dependerá de la distancia del mercado y las especificaciones del comprador. (AGROBANCO, 2012)

Referente a la demanda de la fruta, este factor es importante porque va a determinar los volúmenes a cosechar y tener en cuenta los criterios a considerarse al momento de realizar la orden de corte. Cuando se realiza el enfunde y encintaje de los racimos, se puede estimar los volúmenes a cosechar durante el año. (AGROBANCO, 2012)

2.1.6 Economía

En el mercado mundial el comercio del banano convencional se orienta a cientos de países importadores del mundo; sin embargo, en el caso del comercio de banano orgánico que es el que exporta Perú en un 100%, el nivel de diversificación de sus mercados es muy limitado, de ahí que la unión europea y siete mercados son los que absorben el porcentaje completo de las importaciones del 2017. Entre los mercados de destino que resaltan por su importante participación también se encuentra Estados Unidos, junto con la unión europea representan en conjunto un promedio del 93% del total colocado por Perú a todos los países. (Ministerio de Agricultura y Riego, 2018)

Las exportaciones peruanas del banano orgánico con destino a la unión europea, en el período del 2001-2017 se ha incrementado a una tasa promedio anual récord de 33% mientras que el crecimiento de las exportaciones a todo el mundo ha sido de un 22%, desde ya este dinamismo exportador es el reflejo del comportamiento de las exportaciones en la primera década del siglo. De esta forma, se puede apreciar una relación directa entre la paulatina apertura del mercado europeo y el incremento de nuestras exportaciones, dinamismo que se acelera desde el 2007 cuando se implementa el programa de reducción arancelaria y de mayor profundidad a favor del Perú en el marco de un acuerdo de libre comercio suscrito en la unión europea, actualmente vigente y en plena aplicación. (Ministerio de Agricultura y Riego, 2018)

El volumen de exportación hacia dicho mercado ha venido creciendo de una manera sostenida en el tiempo, alcanzando un nivel histórico con 126,5 mil toneladas; lo cual es explicado por el crecimiento global de las importaciones peruanas hacia los nichos de mercado que demandan mayor banano orgánico visto como un alimento saludable, en las que destacan además de Estados Unidos, un mercado tradicional como Japón y nuevos mercados como Corea del Sur, Canadá, Emiratos Árabes, etc. Sin embargo, hay mercados aún no explorados como China, Rusia, Irán, Siria, Argentina, que tienen una importante demanda y con quienes se debe trabajar intensamente un programa de difusión y promoción del consumo del banano orgánico. (Ministerio de Agricultura y Riego, 2018)

Perú tiene 15,000 hectáreas de banano orgánico, cultivados con fines de exportación. Durante el año 2018, se exportaron unas 200,000 toneladas, las cuales representan aproximadamente 170 millones de dólares. (Rosales, 2019) Se espera un crecimiento constante en los años siguientes. Las principales regiones productoras de este producto son Piura, Tumbes y Lambayeque.

2.1.7 Métodos actuales de control en la cadena en el Perú

En el Perú, los agricultores toman diferentes medidas de control en busca de la mejora del proceso y en el aumento de la producción del banano, esto se debe principalmente por la

alta competitividad que se genera por los distintos países que tienen como bases la importación de productos agrícolas.

El uso de cintas de colores permite llevar un control y programar así el orden de cosecha o corte, generalmente se cosechan 3 colores diferentes en una semana. El número de cintas de colores va a depender de las políticas de la empresa u organización. El número de colores puede variar entre 8 y 12 colores diferentes de cintas para identificar una semana. (AGROBANCO, 2012)

Dependiendo de la variedad, la fruta puede ser cosechada en un determinado tiempo. Por ejemplo, si se cosecha la fruta que ha sido encintada con la cinta azul hace doce semanas, entonces la orden del primer corte o cosecha se dará en esa semana con un determinado calibre o diámetro de dedos, el segundo corte de esa misma cinta será en la siguiente semana con un calibre menor al primero y la semana después se realizará el corte final o barrido de todos los racimos que tengan la cinta azul. Esto se hace porque esta última fruta ya ha alcanzado su máximo desarrollo fisiológico. De la misma forma se hace con los otros colores de las cintas. (AGROBANCO, 2012)

2.2 Cultivo de banano en Filipinas

Es el principal cultivo frutícola de Filipinas en cuanto a superficie, volumen y valor de producción. Tres cuartas partes del total de productores de banano concentran casi el 90% de la superficie plantada de este cultivo. Los bananos solían cultivarse como alimento de primera necesidad o para su venta en mercados locales. No se exportaron hasta que a finales de los años sesenta llegaron empresas transnacionales en busca de bajos costos de producción y transporte cerca de Japón. (César, 2014)

La producción y las exportaciones de banano han aumentado en los últimos treinta años a una tasa de las tasas más elevadas de entre los exportadores de banano del mundo. En 2000, se exportó una cantidad estimada en 1.6 millones de toneladas de banano. Filipinas era el cuarto país de importancia entre los exportadores de banano y contribuyó con casi 13% del suministro mundial. Su proximidad a China, Taiwán, Corea y Japón ha permitido desarrollar sus exportaciones sobre la base de estos mercados. Incluso ha logrado desplazar a Ecuador del mercado chino, en el 2005 ya era el tercer abastecedor de banano más importante del mundo con un 15% de participación y actualmente es el segundo mercado más importante, apuntando al mercado ruso. Dos terceras partes de los bananos exportados se cultivan en la Isla de Mindanao. (César, 2014)

Capítulo 3

Estado del arte

3.1 Investigaciones de carácter científico sobre nutrientes en plantas y plagas

3.1.1 *Diagnostics of macronutrient deficiencies in three legume species*

Tabla 1 Diagnostics of macronutrient deficiencies in three legume species

Bibliografía	Early diagnostics of macronutrient deficiencies in three legume species by color image analysis (Wiwart et al., 2009)
Objetivo	Detección de condiciones de deficiencia de nitrógeno, fósforo, potasio y magnesio, mediante análisis de colores de imágenes digitales.
Lugar/Muestra	Polonia
Metodología	El color se describió en los sistemas H, S, I y $L^* a^* b^*$. El método aplicado en el estudio, que consistió en calcular las distancias euclidianas entre el color de las hojas en los nudos sucesivos, permite una determinación muy precisa de las respuestas de las especies analizadas a las deficiencias de cuatro macronutrientes. Adicionalmente, proporciona una base para desarrollar un modelo de sistema automatizado para el diagnóstico temprano de deficiencias de nutrientes en los cultivos.
Resultados	Se desarrolla un método de detección preciso para el dataset muestreado.
Conclusiones	El método de distancias euclidianas resulto efectivo para detectar la variación de los parámetros H, S, I y L^*, a^*, b^*

Fuente: Elaboración propia

3.1.2 Corn Leaf Diseases Diagnostic Techniques Based on Image Recognition

Tabla 2 Corn Leaf Diseases Diagnostic Techniques Based on Image Recognition

Bibliografía	Corn Leaf Diseases Diagnostic Techniques Based on Image Recognition (Zhu et al., 2012)
Objetivo	Reconocimiento automáticamente de enfermedades de las hojas de maíz por técnicas de procesamiento digital de imágenes y método de reconocimiento de patrones
Lugar/Muestra	China
Metodología	<p>El método se divide en tres pasos. Primero, las imágenes en escala de grises se obtienen a partir de imágenes en color, que se mejora mediante el método de ecualización de histograma, y se elimina el ruido no deseado de la imagen.</p> <p>En segundo lugar, las manchas de la enfermedad se segmentaron de las hojas según el método de umbral iterativo y métodos morfológicos. Finalmente, se extraen los parámetros característicos de la forma de las manchas de enfermedad, como área, perímetro, rectangularidad, circularidad y complejidad de forma, que se utilizan para identificar y diagnosticar enfermedades</p>
Resultados	Los resultados muestran que las enfermedades de la hoja de maíz de las 30 imágenes podrían diagnosticarse bien con una tasa de diagnóstico del 80%.
Conclusiones	La tecnología de reconocimiento automático de enfermedades del maíz mediante técnicas de procesamiento de imágenes digitales y métodos de reconocimiento de patrones puede identificar la mayoría de las enfermedades del maíz. El método morfológico puede mejorar la eficacia de segmentación de imágenes que ha sido segmentada por el método iterativo de umbralización de las técnicas de diagnóstico de enfermedades basadas en el reconocimiento de imágenes 341.

Fuente: Elaboración propia

3.1.3 Unhealthy region of citrus leaf detection using image processing techniques.

Tabla 3 Unhealthy region of citrus leaf detection using image processing techniques

Bibliografía	Gavhale, Kiran R. Gawande, Ujwalla. Hajari, Kamal O. 2014. "Unhealthy region of citrus leaf detection using image processing techniques." (Gavhale et al., 2014)
Objetivo	Técnicas de procesamiento de imagen con el fin de ser utilizadas para la detección temprana de enfermedades a partir de la inspección de las hojas de la planta.
Lugar/Muestra	India
Metodología	Inicia con la recopilación y subdivisión de las distintas enfermedades, las cuales terminan presentándose en las hojas de las plantas. A partir de esto, se inicia el preprocesamiento de las imágenes tomadas, haciendo un mejoramiento de color por escala (CES) y luego una conversión del color, en este caso se optó por dos que fueron el Ycbr y el Lab. Después de esto, se hace una segmentación de las imágenes usando la agrupación de k-medias, a lo cual se le aplica un método estadístico de agrupación llamado GLCM. Por último, se utiliza la máquina de vectores de soporte (SVM), en específico los modelos radiales y polinomiales.
Resultados	Se obtuvo con el modelo polinomial un porcentaje de exactitud de 95% mientras que con el radial se obtuvo 96%. Siendo resultados bastante similares.
Conclusiones	A partir del método realizado en este paper, el uso de productos químicos puede ser reducido y mejorar la salud de las plantas en el cultivo. Además, se observó que la conversión de color utilizada mejora el análisis.

Fuente: Elaboración propia

3.1.4 *Detection and classification of diseases of banana plant using local binary pattern and support vector machine.*

Tabla 4 Detection and classification of diseases of banana plant using local binary pattern and support vector machine.

Bibliografía	Aruraj, Akshaya, Alex, Ashish, Subathra, M. S.P., Sairamya, N. J., Thomas George, S., Vinodh Edwards, S. E. 2019. "Detection and classification of diseases of banana plant using local binary pattern and support vector machine." (Aruraj et al., 2019)
Objetivo	La metodología propuesta se divide en dos partes: extracción de características de textura mediante el uso de patrones binarios locales y la clasificación del banano entre las enfermedades y cuando la planta se encuentra sana utilizando support vector machine (SVM).
Lugar/Muestra	India
Metodología	El procedimiento inicia con la adquisición de imágenes del banano, y a estas imágenes se le hace un cambio de tamaño y la conversión de RGB a una escala de grises. Luego, se hace la extracción de características de textura mediante el uso de patrones binarios locales y de textura. Finalmente, en la clasificación se utiliza SVM y K-nearest neighbor (KNN).
Resultados	A partir de los modelos utilizados, se obtiene un valor máximo de exactitud del 90.9%, lo cual permite clasificar las distintas enfermedades y las plantas saludables.
Conclusiones	Con los resultados obtenidos, se observa que la metodología utilizada da una relación entre las características obtenidas de las imágenes. El clasificador SVM da un valor de exactitud bastante alto, lo cual tiene una buena confiabilidad para ser utilizado por los agricultores, de esta forma poder prevenir distinto tipo de enfermedades que pueda tener las plantas.

Fuente: Elaboración propia

3.1.5 Multiple nutrient deficiency detection in paddy leaf using color and pattern analysis.

Tabla 5 Multiple nutrient deficiency detection in paddy leaf using color and pattern analysis

Bibliografía	Latte, M. V. Shidnal, Sushila. 2016. "Multiple nutrient deficiency detection in paddy leaf using color and pattern analysis." (Latte & Shidnal, 2016)
Objetivo	Identificar la deficiencia de nutrientes a partir del color de las hojas. Este análisis se hace a partir del patrón de colores en RGB. De esta forma, se clasifican las imágenes en las saludables y a las que les falta nitrógeno, fósforo y potasio.
Lugar/Muestra	India
Metodología	Se utilizó un total de 500 imágenes para el entrenamiento del programa y luego 100 se usaron para testarlo. Las imágenes tenían un valor alto de píxeles, por lo que se les cambió el tamaño a 256 píxeles de cada lado. El ruido dado en el cambio de tamaño se redujo utilizando un filtro medio. Debido a que en las plantas más saludables predomina el color verde, se analiza la relación entre G/R y G/B para encontrar una relación y proceder a la clasificación del resto de las imágenes de las hojas.
Resultados	A partir de la relación de G/R y G/B se clasificó las imágenes en las distintas categorías, obteniéndose un valor del 85% de similitud para los valores promedios.
Conclusiones	El modelo tiene un valor de exactitud de 90% y puede ser mejorado aumentando la base de imágenes con las cuales se entrenó, además tiene la capacidad de mejorar la detección y detectar más de dos deficiencias.

Fuente: Elaboración propia

3.1.6 Machine Learning regression technique for cotton leaf disease detection and controlling using IoT

Tabla 6 Machine Learning regression technique for cotton leaf disease detection and controlling using IoT

Bibliografía	Sarangdhar, Adhao Asmita. Pawar, Vijaya. 2017. "Machine Learning regression technique for cotton leaf disease detection and controlling using IoT."
Objetivo	El uso de un sistema de detección y control de enfermedades presentes en las hojas utilizando Vector Support Machine para clasificarlas, además de la implementación de un sistema de comunicación utilizando IoT.
Lugar/Muestra	India
Metodología	<p>Para la recopilación de datos y el envío de estos se utilizó un Raspberry Pi, al cual estaba conectado sensores de humedad de suelo, humedad y temperatura del aire. La data recopilada se enviaba a una base de datos y se direccionaba a una aplicación para sistemas Android mediante la cual los agricultores pueden revisar los parámetros.</p> <p>En base al sistema de machine Learning, el proceso inicia con la adquisición de imágenes, utilizan 900 imágenes. El cambio de tamaño a la cual se redujo es en 250 pixeles cada lado. El ruido de las imágenes es eliminado en el preprocesamiento utilizando un filtro medio.</p> <p>Además, para la mejora de la imagen se realiza la conversión de los colores pasando de RGB a YCbCr, luego se cambia a blanco y negro para luego aplicarse RGB en la imagen, por ultimo se pasa a una escala de grises.</p> <p>En la clasificación se utiliza el Support Vector Machine, para el entrenamiento y el testeo con técnica de regresión, la cual permite encontrar las relaciones no lineales entre los vectores de entrada y las variables a partir de encontrar el mejor hiperplano. El mejor hiperplano es el que se encuentra en la mayor distancia entre los vectores de entrenamiento.</p>
Resultados	De las 900 imágenes que se usaron, 629 se destinaron para entrenar y 271 para testear al programa, obteniéndose un valor promedio de 83.26% de exactitud con el programa.
Conclusiones	El fácil uso de una aplicación para los agricultores les ayuda a mantener la información de los parámetros que se están midiendo. Además, el uso de SVM tiene un buen valor de exactitud el cual puede incrementarse más si se le entrena con una mayor cantidad de imágenes.

Fuente: Elaboración propia

3.1.7 AI-powered banana diseases and pest detection

Tabla 7 AI-powered banana diseases and pest detection

Bibliografía	Selvaraj, Michael Gomez, Alejandro Vergara, Henry Ruiz, Nancy Safari, Sivalingam Elayabalan, Walter Ocimati, and Guy Blomme. 2019. "AI-Powered Banana Diseases and Pest Detection." <i>Plant Methods</i> 15(1):1–11. (Selvaraj et al., 2019)
Objetivo	El objetivo es poder desarrollar un sistema que permita la detección de enfermedades y plagas. Con la finalidad de que los agricultores tengan una mejor gestión de sus cultivos.
Lugar/Muestra	África y sur de la india
Metodología	Se tomaron y recopilaron un gran conjunto de datos de imágenes de daños de las enfermedades del banano y los síntomas de las plagas para así luego mediante redes neuronales convolucionales hacer un análisis completo de las imágenes, y así mismo entrar a la neurona con las diferentes imágenes para lograr que detecte con exactitud que plaga o enfermedad tiene o puede padecer. Cabe resaltar que para lograr todo esto se obtuvo un total de 18000 imágenes de las enfermedades o plagas del banano, cada una de estas en diferentes condiciones medioambientales. Luego de tener toda una base de datos de las enfermedades que puedan aparecer en el banano se realizó la identificación de las imágenes de la base de datos. Una vez con todo esto se pasó elaborar la arquitectura de las redes neuronales convolucionales y luego entrenarlas.
Resultados	Se desarrollaron un total de 6 modelos diferentes los cuales fueron desarrollado a partir de los 18 tipos de enfermedades, utilizando imágenes recogidas de diferentes partes de la planta bananera, para así lograr su precisión. Lograron un 90% de precisión en gran parte de los modelos probados.
Conclusiones	Recalcan la importancia de considerar los factores como el clima, humedad, época del año, lluvias, variedad, nutrición. Y sumamente la localización del cultivo ya que hay enfermedades que solo se presentan en un solo país o localidad. Es sumamente importante la parte de la identificación de los tipos de imágenes así mismo como su rigurosa identificación. Se puede extender el estudio para desarrollar una aplicación completamente automatizada para ayudar a los miles de agricultores del banano.

Fuente: Elaboración propia

3.1.8 LeafCheckIT: A Banana Leaf Analyzer for Identifying Macronutrient Deficiency

Tabla 8 LeafCheckIT: A Banana Leaf Analyzer for Identifying Macronutrient Deficiency

Bibliografía	Tejada, Jonilyn A., and Glenn Paul P. Gara. 2017. "LeafCheckIT: A Banana Leaf Analyzer for Identifying Macronutrient Deficiency." ACM International Conference Proceeding Series 458–63.
Objetivo	El primer objetivo es definir las características de imagen para clasificar las imágenes de las hojas. Estas imágenes serán usadas para el entrenamiento del clasificador de hojas sanas, deficientes en macroelementos y aquellas que no sean del fruto en estudio, banana. El siguiente paso es la digitalización a través de una aplicación web y móvil para el propósito mencionado.
Conceptos	Random Forest: El algoritmo de bosques aleatorios es una técnica de aprendizaje automático que se utiliza cada vez más para la clasificación de imágenes y la creación de variables continuas. (Hording).
Muestra	Para el entrenamiento se han usado 705 imágenes de las cuales 50 era de banano sano, 155 con deficiencia de P y 255 con deficiencia de N.
Metodología	<p>Primero se llevó a cabo una revisión de literatura para conocer oportunamente los síntomas en las hojas del banano con deficiencia de macroelementos. Luego se tomaron las fotografías con la cámara del iPhone 5 en zonas de Davao del Norte y la provincia del Valle de Compostela. Para verificar los datos obtenidos por el programa se envió las muestras tomadas para que se les realizara un análisis.</p> <p>Se utilizó Python pues tenía compatibilidad con la multiplataforma, OpenCV para facilitar la clasificación de características, la técnica DenseSIFT para extraer la textura de las imágenes.</p> <p>Para seleccionar la característica de color adecuada se exploraron diferentes modelos cromáticos como RGB, HSV, YCbCr y CIE Lab.</p>
Resultados	Para la extracción de características se determinó que con la técnica DenseSIFT, los modelos cromáticos RGB, CIE Lab, YCrCb tenían mayor exactitud. En cuanto al clasificador a usar se probó con SVM, ANN, Naive Bayes y Random Forest. De estos el que tuvo mayor grado de exactitud fue el Random Forest. Su porcentaje de precisión para la deficiencia de N, P, K y hojas fue de 0.898, 0.966, 0.856 y 0.818 respectivamente.

Fuente: Elaboración propia

3.1.9 CNN for leaf image-based plant disease classification

Tabla 9 CNN for leaf image-based plant disease classification

Bibliografía	Jadhav, S. B. (2019). Convolutional Neural Networks for Leaf Image-Based Plant Disease Classification. IAES International Journal of Artificial Intelligence (IJ-AI), 8(4), 328. https://doi.org/10.11591/ijai.v8.i4.pp328-341
Objetivo	Identificación de enfermedades en las plantas de soya, mediante transfer learning, con diferentes tipos de redes convolucionales
Muestra	Department of Electronics and Telecommunication Engineering, Bharati Vidyapeeths College of Engineering Kolhapur, (Research Scholar -VTU), Belagavi, Karnataka, India Maratha Mandala Engineering College, Belagavi, Karnataka, India Shri. Chattrapati Shivajiraje College of Engineering, Bhor, Pune, India
Metodología	Clasificación de varias categorías de enfermedades de las plantas de soya. Como parte de la metodología seguida, tenemos la obtención de el data set, con estos se procedió a realizar experimentos con las diferentes arquitecturas de redes neuronales convolucionales. Así mismo se estudiaron y definieron los parámetros con los cuales se iban a utilizar las redes neuronales.
Resultados	Como resultado se obtuvieron diferentes valores de acuerdo con los diferentes modelos usados, siendo GoogleNet y VGG16 los modelos con mayor accuracy, Obteniendo un valor de 96.4%
Conclusión	Se demostró el uso de modelos pre-entrenados como extractores de características y clasificadores que llegaron a superar a los clasificadores tradicionales, así mismo como la eficacia de la modificación de los hiperparámetros, como el minbatch, máximo número de épocas, y la tasa de aprendizaje de sesgo.

Fuente: Elaboración propia

3.1.10 Identification of Apple Leaf Diseases Based on Deep CNN

Tabla 10 Identification of Apple Leaf Diseases Based on Deep CNN

Bibliografía	Liu, B., Zhang, Y., He, D., & Li, Y. (2017). Identification of Apple Leaf Diseases Based on Deep Convolutional Neural Networks. <i>Symmetry</i> , 10(1), 11. https://doi.org/10.3390/sym10010011
Objetivo	Identificar enfermedades en la hoja de la planta de manzano, para así evitar que esta prolifere en las demás plantas arruinando la cosecha garantizando el desarrollo saludable de la industria de la manzana, utilizando el aprendizaje profundo mediante las redes neuronales convolucionales.
Muestra	Qingyang county, Gansu Province, China and Baishui county, Shanxi Province, China.
Metodología	Para la metodología en primer lugar se hizo la adquisición de las imágenes, luego de estos se hizo el procesamiento de las imágenes, el cual incluía disminuir todos aquellos desperfectos causados por el ruido presente en la imagen. Posterior a esto se procedió a construir la red neuronal convolucional, en la cual se tomaron en cuenta las capas y las funciones que las envuelven a estas. Posterior a esto se procedió a la experimentación y evaluación.
Conclusión	A partir de la data generada de las distintas perturbaciones que aplicaron a las imágenes y de las modificaciones de las diferentes arquitecturas utilizadas se obtuvo como resultado final un valor de accuracy de 97.62%

Fuente: Elaboración propia

3.1.11 Acquiring and preprocessing leaf images for automated plant identification: understanding the tradeoff between effort and information gain.

Tabla 11 Acquiring and preprocessing leaf images for automated plant identification: understanding the tradeoff between effort and information gain

Bibliografía	Rzanny, M., Seeland, M., Wäldchen, J. et al. Acquiring and preprocessing leaf images for automated plant identification: understanding the tradeoff between effort and information gain. <i>Plant Methods</i> 13, 97 (2017). https://doi.org/10.1186/s13007-017-0245-8
Objetivo	El objetivo principal de este trabajo de investigación es realizar una comparación entre tres estrategias de preprocesamiento, en relación con el esfuerzo requerido y la ganancia de información.
Lugar/Muestra	Praderas repartidas por la ciudad de Jena, ubicada en el este de Alemania.
Metodología	En este artículo, se estudia sistemáticamente nueve tipos de imágenes y tres estrategias de preprocesamiento. Los tipos de imágenes varían en términos de las condiciones de grabación de imágenes in situ: perspectiva, iluminación y fondo, mientras que las estrategias de preprocesamiento comparan imágenes no preprocesadas, recortadas y segmentadas. Se extraen características de la imagen usando una red neuronal convolucional, se clasifican especies usando los vectores de características resultantes y se discute la precisión de la clasificación.
Resultados	Se obtiene la mayor precisión de clasificación usando imágenes de retroiluminación destructivas, es decir, sosteniendo la hoja arrancada contra el cielo para la adquisición de imágenes. Recortar la imagen al límite de la hoja mejora sustancialmente la precisión, mientras que la segmentación obtiene similar precisión con un esfuerzo sustancialmente mayor. El uso o desuso permanente de flash tiene efectos insignificantes. Obtener imágenes de la parte trasera de una hoja con una textura típicamente más fuerte no resulta en mayor precisión, pero incrementa notablemente el costo de adquisición.
Conclusiones	En conclusión, la forma en que se adquieren y preprocesan las imágenes de las hojas tiene un efecto sobre la precisión del clasificador entrenado en ellos. Por primera vez, este estudio proporciona una guía sistemática permitiendo a los investigadores invertir sabiamente los recursos de adquisición disponibles mientras se obtiene la precisión de clasificación óptima.

Fuente: Elaboración propia

3.1.12 Identification of Plant Nutrient Deficiencies Using CNN

Tabla 12 Identification of Plant Nutrient Deficiencies Using CNN

Bibliografía	U. Watchareeruetai, P. Noinongyao, C. Wattanapaiboonsuk, P. Khantiviriya and S. Duangsrissai, "Identification of Plant Nutrient Deficiencies Using Convolutional Neural Networks," 2018 International Electrical Engineering Congress (iEECON), Krabi, Thailand, 2018, pp. 1-4, doi: 10.1109/IEECON.2018.8712217.
Objetivo	Se propone un método novedoso de análisis de imágenes para identificar las deficiencias de nutrientes en la planta en función de su hoja.
Lugar/Muestra	King Mongkut's Institute of Technology Ladkrabang
Metodología	Primero el método propuesto divide una imagen de entrada de hoja en pequeños bloques. En segundo lugar, cada bloque de píxeles de hoja alimenta a un conjunto de redes neuronales convolucionales (CNN). Cada CNN está capacitada específicamente para deficiencia de nutrientes y se utiliza para decidir si un bloque presenta cualquier síntoma de la deficiencia de nutrientes. A continuación, las respuestas de todas las CNN se integran para producir una respuesta para el bloque utilizando una estrategia de "el ganador se lleva todo". Finalmente, las respuestas de todos los bloques se integran en uno usando un perceptrón multicapa para producir una respuesta final para la hoja entera.
Resultados	A pesar de que el método propuesto no pudo lograr una identificación muy precisa, fue notablemente más exitoso que las personas entrenadas.
Conclusiones	Los resultados experimentales presentan la viabilidad de utilizar CNN para identificar deficiencias de nutrientes en plantas basado en sus hojas, e indican su superioridad sobre los humanos en comparación a su rendimiento.

Fuente: Elaboración propia

3.1.13 Crop pest recognition in natural scenes using convolutional neural networks.

Tabla 13 Crop pest recognition in natural scenes using convolutional neural networks

Bibliografía	Li, Y., Wang, H., Dang, L. M., Sadeghi-Niaraki, A., & Moon, H. (2020). Crop pest recognition in natural scenes using convolutional neural networks. <i>Computers and Electronics in Agriculture</i> , 169(December 2019), 105174. https://doi.org/10.1016/j.compag.2019.105174
Objetivo	Este documento presenta un método de reconocimiento de plagas de cultivos que reconoce con precisión diez especies comunes al aplicar varias redes neuronales convolucionales profundas (CNN).
Lugar/Muestra	Department of Computer Science and Engineering, Sejong University, Seoul, Republic of Korea
Metodología	Después del proceso de recopilación de datos, el fondo natural fue preprocesado mediante dos diferentes métodos de eliminación de fondo. Y luego, se generaron más imágenes aplicando técnicas de aumento de datos. A continuación, las imágenes se introdujeron en cinco modelos de aprendizaje profundo y el modelo de CNN más adecuado fue seleccionado.
Resultados	En este documento, el modelo mejorado de GoogLeNet se compara con el modelo ResNet101 ajustado. El resultado experimental muestra que el modelo GoogLeNet ajustado funciona mejor que el ResNet101 ajustado tanto en precisión como en tiempo de entrenamiento.
Conclusiones	El experimento muestra que el modelo GoogLeNet superó a los otros modelos en términos de precisión, complejidad del modelo y solidez. El modelo perfeccionado logró un 5.91% adicional de precisión que el modelo original.

Fuente: Elaboración propia

3.1.14 SMART-Fertilizer

Tabla 14 SMART-Fertilizer

Link	https://www.smart-fertilizer.com/?gclid=Cj0KCOjwm9D0BRCMARIsAIfvfIZJP9jKRCYjGCgz07iHEw3QZ4lxoFWQRZ0FUd4aFG8N-lcC2lgT7-YaAjJYEALw_wcB#our_products_content
¿Qué es?	Plataforma de gestión inteligente de fertilizante para agricultores
Función	<p>Mediante la lectura del análisis de suelo, en la plataforma, diseñan un plan mensual de acuerdo con los requerimientos de la planta, así mismo ofrece una interpretación de los resultados obtenidos en el análisis de suelo.</p> <p>Otra característica importante es que permite saber la calidad del agua que le es suministrada al cultivo. Este es un gran ejemplo y paso en lo que vendría a ser la digitalización de los datos, ya que este software le permitirá al agricultor almacenar todos los datos ya sea del fertilizante suministrado como de los nutrientes de su cultivo.</p>

Fuente: Elaboración propia

3.1.15 Tumaini App

Tabla 15 Tumaini App

Link	https://play.google.com/store/apps/details?id=ciat.cgiar.org.tumaini&hl=es_PE
¿Qué es?	Aplicación móvil para la detección de plagas y enfermedades en plantas de banano.
Función	<p>Es una aplicación que mediante inteligencia artificial permite a los agricultores y trabajadores poder identificar y brindar ciertas recomendaciones para combatir estas enfermedades.</p> <p>La performance de la aplicación cuenta con 3 pasos que son el tomar la foto, subirla y el agricultor recibirá un diagnóstico de la imagen con posibles sugerencias. Cabe resaltar que todo el proceso se hace mediante el procesamiento de imágenes.</p> <p>La finalidad de la app es poder ayudar a los agricultores en la producción de alimentos.</p>

Fuente: Elaboración propia

Capítulo 4

Metodología de la investigación

4.1 Procedimientos a emplear para el desarrollo del proyecto

4.1.1 *Design Thinking*

➤ Empatiza

Comienza el proceso con una profunda comprensión de las necesidades de los agricultores quienes en este caso serán los usuarios, así como las empresas con la misma problemática y interesados en la solución que se está planteando. En este proceso debemos estar en la capacidad de poder ponernos en los zapatos del otro para poder entender la problemática y poder generar soluciones consecuentes con sus realidades (Dinngo, n.d.).

➤ Define

Con toda la información recopilada durante la fase anterior, se sintetiza y se toma en cuenta todo aquello que nos aporta valor y nos lleva al alcance de nuevas perspectivas interesantes. Durante esta fase dedicaremos tiempo a identificar problemas cuyas soluciones serán la clave para la obtención de nuestro producto final innovador (Dinngo, n.d.).

➤ Ideación

Durante esta etapa lo que se hace es obtener la mayor cantidad de ideas y diferentes opciones. Hay que dejar de lado los juicios de valor, así mismo las ideas más raras a veces son las que generarán las soluciones más innovadoras (Dinngo, n.d.).

➤ Prototipa

Durante esta etapa lo que se realiza es volver las ideas realidad, se empiezan a construir los prototipos y se empiezan a hacer las ideas palpables, esto nos ayuda a empezar a visualizar las posibles soluciones, permitiéndonos poder observar en que se debe mejorar o refinar antes de llegar al resultado final (Dinngo, n.d.).

➤ Testea

En esta etapa se pondrá a prueba los diferentes prototipos diseñados, en esta etapa se desarrolla la primera experiencia del usuario con la solución innovadora en desarrollo. Esta etapa tiene una gran importancia ya que nos ayudará a ultimar las correcciones que haya que realizar a nuestro prototipo se caracteriza por ser una etapa crucial durante todo este proceso. Como parte final de esta etapa se busca evolucionar la idea hasta llegar a la solución que se estaba buscando (Dinngo, n.d.).

4.1.2 Adquisición de data

Para la adquisición de la data se recurrió a enviar un correo a los autores de la aplicación “Leak Banana”, utilizada para la identificación de deficiencia de macronutrientes como el fósforos, potasio y nitrógeno, aplicación desarrollada por los ingenieros Jonilyn Tejada y Glen Gara de la Universidad de la Inmaculada concepción en Filipinas.

El data set que se nos brindó está conformado por 530 imágenes y divididas en 4 subcarpetas, “healthy”, “nitrogen”, “potassium” y “phosphorus”. Este conjunto de imágenes ha sido tomado con un iPhone 5, y todas se encuentran en diferentes tamaños y formatos distintos.

4.1.3 Preprocesamiento de imágenes

4.1.3.1 Espacio de color HSV. Expresa el color en tres componentes: Matiz, Saturación, Valor (Hue, Saturation, Value). Los posibles valores para el matiz varían entre 0 y 360, mientras que los valores para los otros dos atributos van desde 0 a 100.

El modelo HSV está basado en coordenadas cilíndricas y, en realidad, es una transformación no lineal del sistema RGB. Por lo tanto, es posible transformar directamente un color desde el sistema HSV al sistema RGB, y viceversa. Existen otros dos sistemas de color relacionados al HSV: el sistema HSL (Luminosity) y el sistema HSI (Intensity).

Este sistema de color es bastante interesante porque permite la separación de los tres componentes de un color específico. Es ampliamente usado en sistemas de visión artificial, ya que es una poderosa herramienta para el desarrollo de algoritmos de procesamiento de imágenes basados en el modelo de percepción del color humano. De hecho, el modelo HSV es muy adecuado para caracterizar colores en términos prácticos para la interpretación humana, diferente a los modelos RGB y CMYK.

Preprocesamiento:

El preprocesamiento aplicado a las imágenes del dataset consistió en el siguiente proceso:

1. Se comienza con una imagen en espacio-color RGB
2. Se importa la librería cv2
3. Se aplica la transformación mediante la siguiente expresión: `cv2.cvtColor(imagen, cv2.COLOR_RGB2HSV)`

4. Por el contrario, si se desea cambiar de HSV a RGB, se utiliza la siguiente transformación: `cv2.cvtColor(imagen, cv2.COLOR_HSV2RGB)`

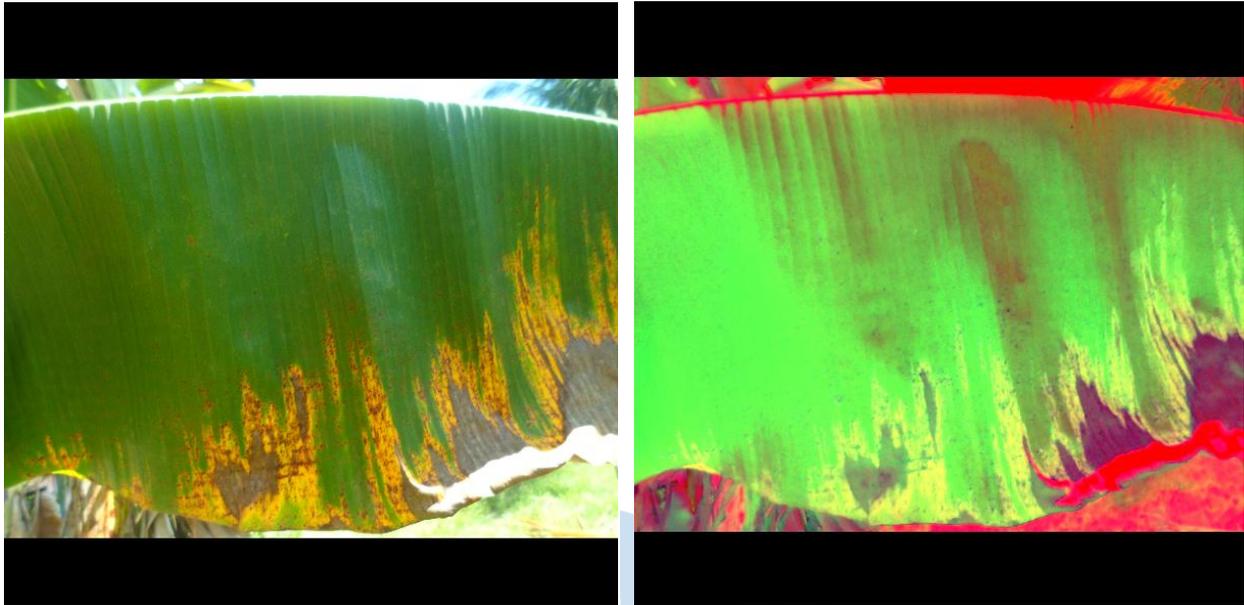


Figura 3. Imagen original (izquierda) y conversión al espacio de color HSV (derecha)

Fuente: Elaboración propia

4.1.3.2 Espacio de color CIELAB. Expresa el color en tres valores: L^* para la luminosidad del negro (0) al blanco (100), a^* del verde (-) al rojo (+) y b^* del azul (-) al amarillo (+). CIELAB fue diseñado para que la misma cantidad de cambio numérico en estos valores corresponda aproximadamente a la misma cantidad de cambio percibido visualmente.

El valor de luminosidad, L^* , representa el negro más oscuro en $L^* = 0$ y el blanco más brillante en $L^* = 100$. Los canales de color, a^* y b^* , representan verdaderos valores de gris neutro en $a^* = 0$ y $b^* = 0$. El eje a^* representa el componente verde-rojo, con verde en la dirección negativa y rojo en la dirección positiva. El eje b^* representa el componente azul-amarillo, con azul en la dirección negativa y amarillo en la dirección positiva.

Para los algoritmos de Python utilizados en el desarrollo del presente trabajo, los rangos numéricos de los canales, L , a y b son: L : 0-100, a : ± 127 , b : ± 127 .

El componente L se asemeja mucho a la percepción humana de luminosidad. Por lo tanto, se puede utilizar para realizar correcciones precisas del balance de color modificando las curvas de salida en los componentes a^* y b^* , o para ajustar el contraste de luminosidad utilizando el componente L^* . Por lo cual se perfila como un espacio de color apropiado para manipular y segmentar las tonalidades características de las hojas de banano y contribuir a la detección de parámetros para la identificación de deficiencias (en las investigaciones citadas se ha hecho uso de este espacio de color para aplicaciones afines).

Preprocesamiento:

El preprocesamiento aplicado a las imágenes del dataset consistió en el siguiente proceso:

1. Se comienza con una imagen en espacio-color RGB
2. Se convierte a espacio-color $L^*a^*b^*$ con la librería cv2 de Python
3. Se aplica una ecualización de histograma sobre el componente a^* del espacio-color $L^*a^*b^*$, para mejorar el contraste de colores en la gama verde-rojo, los cuales son las tonalidades características de la hoja de banano
4. Por último se retorna al espacio RGB para poder aprovecharse en una red neuronal.



Figura 4. Imagen original (izquierda) y conversión al espacio de color CIELAB (derecha)

Fuente: Elaboración propia

4.1.3.3 Espacio de color YCbCr. Al igual que el RGB, el YCbCr es un espacio de color el cual ha tenido usos en los sistemas de televisión digital. Su primer componente “Y” hace referencia al brillo o intensidad de la luz que tiene la imagen, tanto el “Cb” como el “Cr” son el valor relativo de los colores azul y rojo respecto al color verde. Los dos últimos componentes son menos sensitivos al ojo humano, ya que este es más sensible a los cambios de la intensidad de la luz en comparación del matiz que pueda tener una imagen.

Preprocesamiento:

El orden que se siguió para utilizar este espacio de color fue el siguiente:

1. Las imágenes con las cuales se va a trabajar están en RGB.
2. Las imágenes se separan en carpetas mediante las cuales se van a clasificar.
3. Se convierte a YCbCr con la librería cv2 de Python.

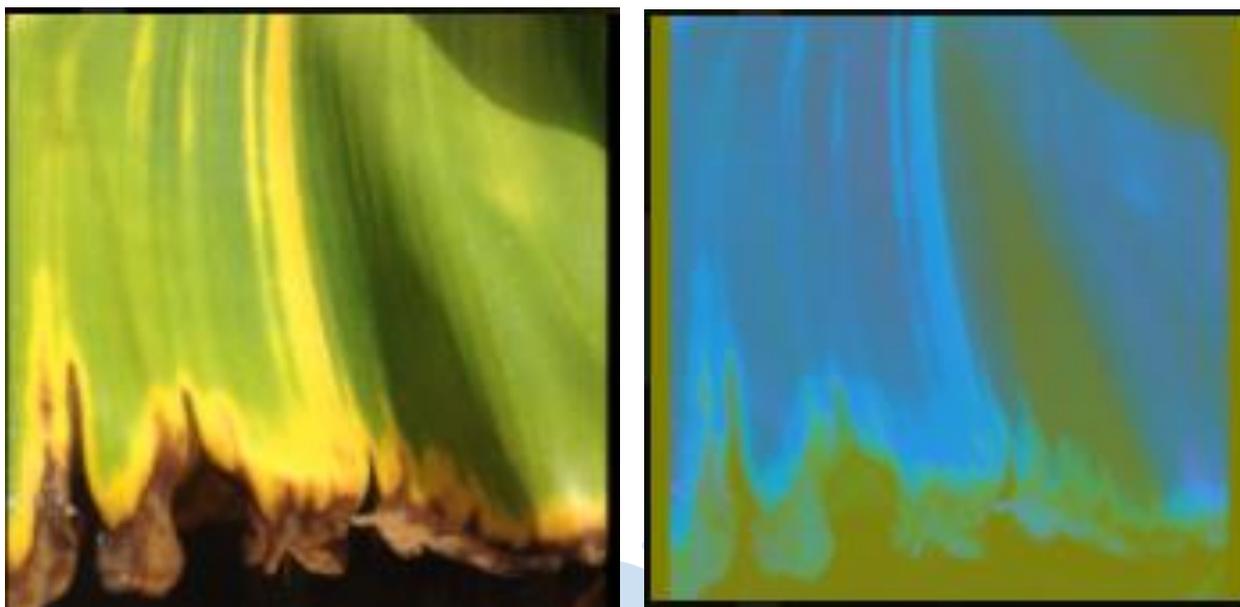


Figura 5. Imagen original (izquierda) y conversión al espacio de color YCbCr (derecha)

Fuente: Elaboración propia

4.1.3.4 Ecuación de histograma. El histograma nos permite representar la frecuencia de ocurrencias de los diferentes canales presentes en la imagen, en este caso tenemos tres canales los cuales son rojo, verde y azul o su abreviatura en inglés RGB (Zdonik et al., 2013).

La modelación del histograma es una poderosa técnica utilizada para la mejora de imágenes, haciendo variaciones en el contraste, permitiéndoles a las áreas de bajo contraste aumentar su valor de contraste, logrando así una mejora en la calidad de la imagen. Al resultado de la modelación del histograma se le llama proceso de ecualización de histograma el cual nos permite mejorar la ocupación del espacio de color. Otro punto bastante importante a señalar es que desde el punto de la fotografía un histograma perfecto es aquel que logra ascender en el comienzo y logra un pico y luego desciende hasta llegar a 0 (Zdonik et al., 2013).

A diferencia de otras operaciones disponibles para la mejora de imágenes en esta no depende de valores introducidos por el usuario. En las siguientes imágenes se podrá notar claramente el impacto positivo en la imagen que genera la ecualización de histograma (Zdonik et al., 2013).

4.1.4 Entrenamiento de CNN

Debido a la falta de un dataset lo suficientemente extenso para entrenar una red desde cero, utilizaremos Transfer Learning en una CNN VGG16 y utilizaremos los pesos de red previamente entrenados como inicializaciones para el entrenamiento específico de nuestra aplicación.

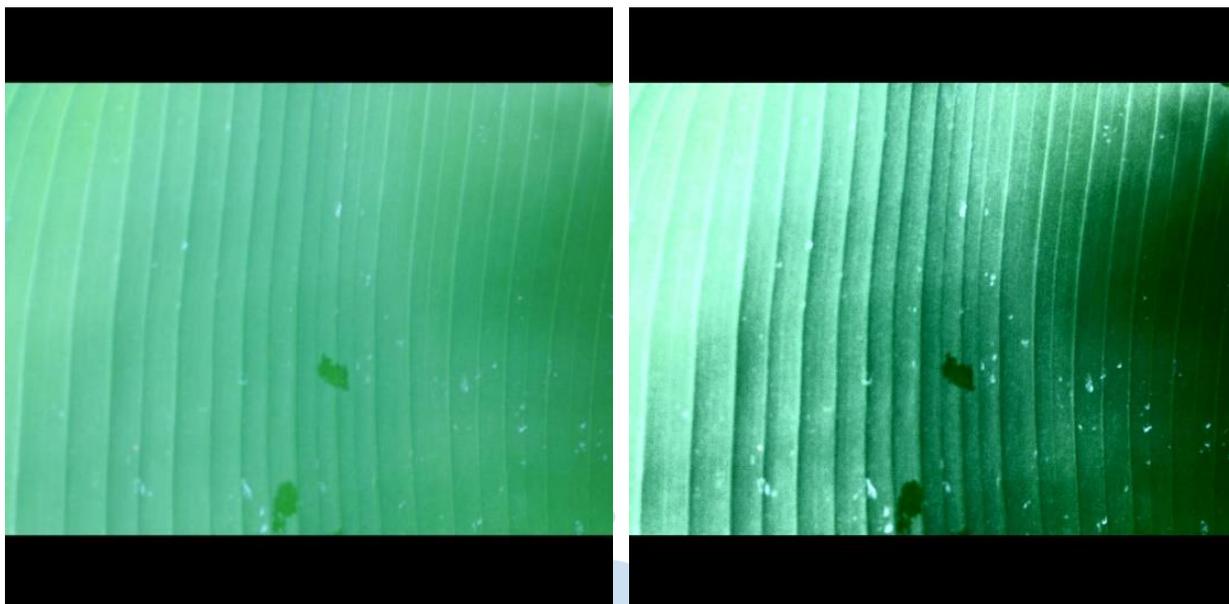


Figura 6. Imagen original (izquierda) y conversión al espacio de color (derecha)

Fuente: Elaboración propia

Además, se realizará data augmentation en el dataset original para aumentar la validación del modelo.

El entrenamiento de la red se realiza de la siguiente manera:

- Se realiza el proceso de data augmentation en el dataset
- Se realiza el preprocesamiento en diferentes espacios de color
- Se separa aleatoriamente el dataset aumentado y pre-procesado en tres grupos: entrenamiento, validación y pruebas.
- Se carga un modelo VGG-16 CNN previamente entrenado con en un gran conjunto de datos.
- Se congelan los pesos de las capas convolucionales inferiores del modelo.
- Todo hasta la penúltima capa es un transformador de características que se utilizará directamente.
- Agregamos nuestro propio clasificador personalizado en transformación de características y se trabaja con los optimizadores para obtener métricas óptimas

4.1.5 Interpretación y validación de resultados

Para saber si la red se está ejecutando y aprendiendo de tal forma que, se obtengan los resultados esperados, esta se mide con los siguientes parámetros:

Loss y val_loss: Es el cálculo del error del modelo que se obtiene en cada iteración mientras este se está ejecutando, por lo cual este valor debe ir disminuyendo a partir de la optimización del proceso que se va dando conforme se va avanzando en cada época. La

diferencia entre estas dos está en que la primera hace referencia al error al set de entrenamiento, mientras al segundo es del set de validación.

Accuracy y val_accuracy: Es el valor mediante el cual se sabe la exactitud que está teniendo el modelo para predecir o clasificar los datos con los cuales se está trabajando. Un indicativo que el programa está funcionando de manera correcta es que este valor vaya en aumento conforme se van ejecutando las épocas del modelo.

Matriz de confusión: Es una herramienta bastante utilizada para tener una visualización del desempeño del algoritmo que se utiliza, en este caso de estudio, para la clasificación de imágenes, aunque su uso está en distintas ramas de inteligencia artificial.

		Predicted	
		Negative	Positive
Actual	Negative	True Negative	False Positive
	Positive	False Negative	True Positive

Figura 7. Matriz de confusión.

Fuente: (Shung, 2018)

En la matriz se encuentra las columnas de los valores predichos y los valores reales y en la intersección de estos se tienen la cantidad de verdaderos negativos, falsos positivos, falsos negativos y verdaderos positivos.

Precisión: Es la relación que hay entre la cantidad de verdaderos positivos entre la cantidad total de positivos (suma de verdaderos positivos y falsos positivos) que el algoritmo detectó.

Recall: Es la relación que hay entre la cantidad de verdaderos positivos entre la cantidad total de valores positivos que realmente lo eran (suma de verdaderos positivos y falsos negativos).

F1 score: A partir de los del Recall y precisión se obtiene esta estimación, ya que se busca el balance entre estas dos métricas. El valor es igual al doble del producto de precisión y Recall sobre la suma de estos dos.

4.1.6 Detección de plagas utilizando TensorFlow

A diferencia de la clasificación de imágenes, en la detección de objetos no se obtiene una clase o etiqueta a la salida, sino que se obtienen las coordenadas de todos los objetos que se detectan en una imagen. En este trabajo se busca aplicar esta tecnología para la detección de plagas, se detallarán los procedimientos llevados a cabo.

En primer lugar, es necesario etiquetar los objetos, es decir, seleccionar manualmente los objetos que se desea detectar dentro de todo el conjunto de datos de entrenamiento. Se hizo uso de LabelImg, que es una herramienta que permite etiquetar fácilmente los objetos dentro de una imagen para posteriormente generar un archivo XML por cada imagen.

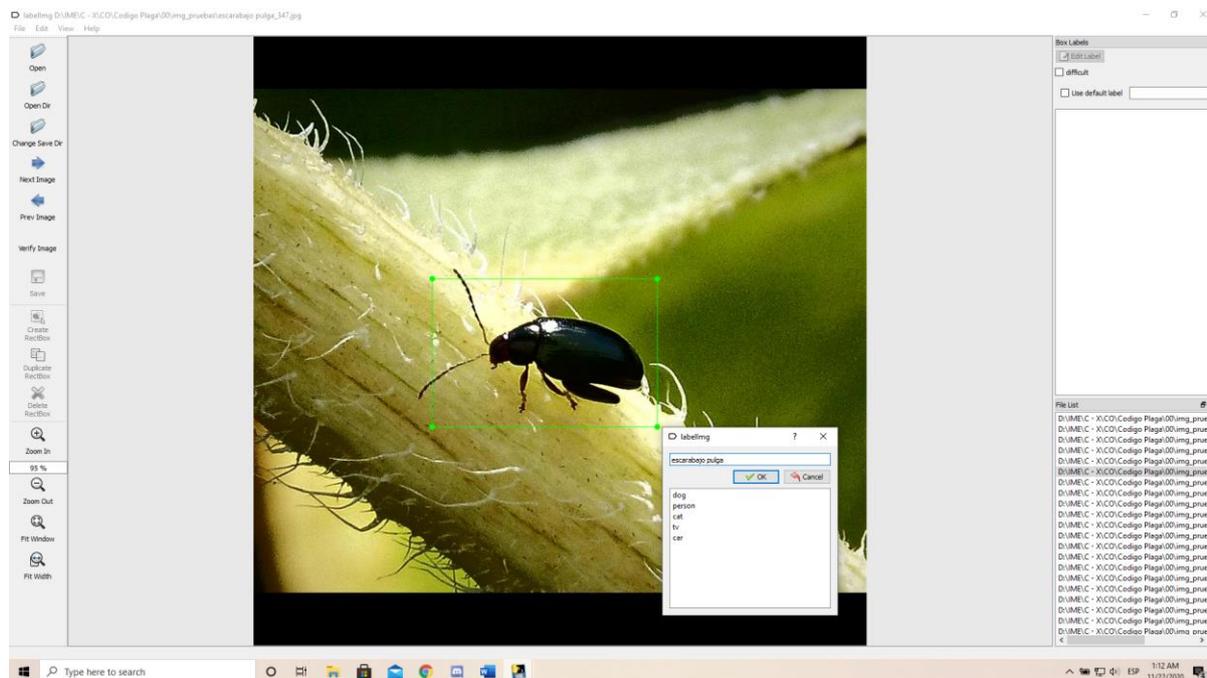


Figura 8. Etiquetado de objetos usando LabelImg

Fuente: (Wu et al., 2019)

Una vez finalizado el etiquetado, se procede a separar la data en tres subconjuntos: train, validation y test, separando un 70%, 15% y 15% de la data, respectivamente. Después, se hace uso del API de TensorFlow, descargando un algoritmo pre-entrenado por TensorFlow para detectar objetos.

En este caso se hace uso del siguiente modelo pre-entrenado: faster_rcnn_resnet101_coco. En él se debe detallar cuántos objetos se desea detectar, la ubicación de los archivos con las etiquetas, etc., se deben realizar las modificaciones pertinentes. Todas estas modificaciones se realizan en el archivo faster_rcnn_resnet101_coco.config.

```

model {
  faster_rcnn {
    num_classes: 1
    image_resizer {
      keep_aspect_ratio_resizer {
        min_dimension: 600

```

```
    max_dimension: 1024
  }
}
feature_extractor {
  type: 'faster_rcnn_resnet101'
  first_stage_features_stride: 16
}
first_stage_anchor_generator {
  grid_anchor_generator {
    scales: [0.25, 0.5, 1.0, 2.0]
    aspect_ratios: [0.5, 1.0, 2.0]
    height_stride: 16
    width_stride: 16
  }
}
first_stage_box_predictor_conv_hyperparams {
  op: CONV
  regularizer {
    l2_regularizer {
      weight: 0.0
    }
  }
}
initializer {
  truncated_normal_initializer {
    stddev: 0.01
  }
}
}
```

La parte inicial *image_resizer* entrega a la red cualquier tamaño de imagen deseado. En *first_stage_features_stride* se da un tamaño para el arreglo de pixeles que se irá desplazando a lo largo de toda la imagen para la detección de objetos: a objetos pequeños se debe colocar valores pequeños.

En *grid.anchor.generator* se crean muchas imágenes pequeñas dentro de la imagen donde se pueden encontrar características en distintas escalas.

```

first_stage_nms_score_threshold: 0.0
  first_stage_nms_iou_threshold: 0.7
  first_stage_max_proposals: 300
  first_stage_localization_loss_weight: 2.0
  first_stage_objectness_loss_weight: 1.0
  initial_crop_size: 14
  maxpool_kernel_size: 2
  maxpool_stride: 2
  second_stage_box_predictor {
    mask_rcnn_box_predictor {

```

En *weight* se pueden colocar pequeños incrementos si uno siente que la red va a sobreentrenarse.

En *truncated_normal_initializer* se inicia con desviación estándar porque se quiere crear números aleatorios para los pesos, que varíen entre 0 y 1.

En *first_stage_nms_score_threshold* se decide qué valor se debería eliminar en la primera etapa de convolución.

En *first_stage_max_proposals* se indica cuantas propuestas se desea guardar como máximo en la primera etapa. Se debe tener cuidado, porque si el número es muy alto se podría conservar propuestas de mala calidad.

```

  mask_rcnn_box_predictor {
    use_dropout: false
    dropout_keep_probability: 1.0
    fc_hyperparams {
      op: FC
      regularizer {
        l2_regularizer {

```

```

    weight: 0.0
  }
}
initializer {
  variance_scaling_initializer {
    factor: 1.0
    uniform: true
    mode: FAN_AVG
  }
}
}
}
}
}
second_stage_post_processing {
  batch_non_max_suppression {
    score_threshold: 0.0
    iou_threshold: 0.6
    max_detections_per_class: 100
    max_total_detections: 300
  }
  score_converter: SOFTMAX
}
second_stage_localization_loss_weight: 2.0
second_stage_classification_loss_weight: 1.0
}
}

```

En *second_stage_post_processing* se especifica lo que se desea conservar al final. En *max_detections_per_class* se especifica el máximo número de detecciones por cada clase: 100 para escarabajo pulga, 100 para araña roja y 100 para mosca blanca.

```
train_config: {
```

```

batch_size: 1
optimizer {
  momentum_optimizer: {
    learning_rate: {
      manual_step_learning_rate {
        initial_learning_rate: 0.0003
        schedule {
          step: 0
          learning_rate: .0003
        }
        schedule {
          step: 900000
          learning_rate: .00003
        }
        schedule {
          step: 1200000
          learning_rate: .000003
        }
      }
    }
  }
  momentum_optimizer_value: 0.9
}
use_moving_average: false
}
gradient_clipping_by_norm: 10.0
fine_tune_checkpoint: "modelo/model.ckpt"
from_detection_checkpoint: true

```

El *learning_rate* indica qué tan rápido se desea que el valor de la función de costo llegue al mínimo posible.

```

num_steps: 200000
data_augmentation_options {
  random_horizontal_flip {
  }
}
}

train_input_reader: {
  tf_record_input_reader {
    input_path: "TFRecords/entrenamiento.record"
  }
  label_map_path: "configuracion/label_map.pbtxt"
}

eval_config: {
  num_examples: 8000
  max_evals: 10
}

eval_input_reader: {
  tf_record_input_reader {
    input_path: "TFRecords/test.record"
  }
  label_map_path: "configuracion/label_map.pbtxt"
  shuffle: false
  num_readers: 1
  num_epochs: 1
}

```

En *from_detection_checkpoint*: true se toma el modelo pre-entrenado y se remueve la última capa y la reemplaza, utilizando nueva data.

4.2 Materiales

4.2.1 Dataset

Como parte de la preparación del set de datos se procedió en primer lugar a observar el conjunto de datos y como estaba conformado. Se observó que todas las imágenes estaban en 3 formatos distintos los cuales eran: “.jpg”, “.png”, “JPEG”. Por lo que se decidió pasar todas a un formato en común, en este caso se escogió el png.

Posteriormente se procedió a pasar a los diferentes espacios de color explicados anteriormente y en un caso se procedió a hacer ecualización de histograma.

Cabe resaltar que la composición de las subcarpetas era la siguiente:

- Healthy-60
- Nitrogen-220
- Phosphorus-140
- Potassium-110

4.2.2 Google Colab

Colaboratory, también llamado "Colab", permite ejecutar y programar en Python en el navegador con las siguientes ventajas (Colaboratory, n.d.):

- No requiere configuración
- Da acceso gratuito a GPUs
- Permite compartir contenido fácilmente

Los cuadernos de Colab permiten combinar código ejecutable y texto enriquecido en un mismo documento, además de imágenes, HTML, LaTeX y mucho más. Los cuadernos creados en Colab se almacenan en la cuenta de Google Drive del usuario. Los cuadernos de Google Colab se pueden compartir fácilmente, con capacidad de edición (Colaboratory, n.d.).

Colab permite importar datos de imágenes, entrenar clasificadores de imágenes con dicho conjunto de datos y evaluar el modelo con tan solo unas pocas líneas de código. Los cuadernos de Colab ejecutan código en los servidores en la nube de Google, lo que permite aprovechar la potencia del hardware de Google, incluidas las GPU y TPU, independientemente de la potencia del equipo del usuario. Lo único que se necesita es un navegador (Colaboratory, n.d.).

En general, es una herramienta muy utilizada en el área de aprendizaje automático. Estos son algunos ejemplos de las aplicaciones que tiene Colab (Colaboratory, n.d.):

- Dar los primeros pasos con TensorFlow
- Desarrollar y entrenar redes neuronales

- Experimentar con TPUs
- Divulgar datos de investigación sobre IA
- Crear tutoriales

4.2.3 Microsoft Visual Studio

Es una plataforma en la cual se puede codificar con diferentes lenguajes de programación como C+, Python, java, etc. Es usada para desarrollar aplicaciones para sistemas como Android o Windows. Además, se puede incluir las distintas bibliotecas con las cuales se trabajan en distintos proyectos. En este caso se utilizó una especial herramienta llamada kite, el cual permite al usuario poder autocompletar al llamar librerías o comandos a utilizar, además de la información referente a alguna herramienta que se esté usando.





Capítulo 5

Resultados

Para el entrenamiento y obtención de un modelo el cual pueda ser usado para subirlo a una plataforma, se recurrió al uso de las redes neuronales convolucionales pertenecientes al Deep Learning y de ellas aplicaremos “Transfer Learning” y posteriormente “Fine Tuning” con la finalidad de mejorar la predicción de nuestro modelo. Cabe resaltar que en cada uno de los pasos a realizar ya sea “Transfer Learning” o “Fine Tuning”, se modificarán sus hiperparámetros hasta que se obtenga el que nos permita obtener datos óptimos.

Al aplicar “Transfer Learning” lo que estamos realizando en breves palabras es reutilizar características extraídas de un modelo anterior en el que se buscaba solucionar algún problema, para resolver uno diferente, este modelo se caracteriza por ser el calentamiento previo a aplicar “Fine Tuning”, ya que lo que se ha realizado es el cargar los pesos de un modelo ya entrenado para ahora entrenar solo aquellas capas no entrenadas, pero con la data que se posee actualmente. Durante este entrenamiento solo se entrenarás las últimas capas del modelo de red neuronal escogido, en este caso se ha escogido el modelo VGG16. (Berzal, 2018)

Posterior a este método aplicado viene el método de “Fine Tuning” que consiste en el proceso de afinado del modelo al que se ha aplicado “Transfer Learning” para así mejorar su funcionamiento en nuestro problema a resolver, así mismo este método aplicado consiste en cargar los pesos ya entrenados durante el “Transfer Learning” y se reentrena la red, pero en este caso con una tasa de aprendizaje menor a la usada para el anterior entrenamiento. (Berzal, 2018)

Como parte de los hiperparámetros modificados tenemos el optimizador SGD que es el Gradiente Descendente Estocástico, este se encarga de calcular el gradiente de la función de costo y luego de ello irá actualizando los pesos y bias por cada lote analizado de la data set. Este método se caracteriza por ser un método inestable para tasas de aprendizaje pequeñas es por ello, que se escogen valores menores a 0.001 como tasa de aprendizaje para el optimizador a usar, logrando así datos bastante beneficiosos para el modelo. (Michelucci, 2018)

5.1 CIELAB

Los resultados obtenidos utilizando las imágenes resultantes del preprocesamiento en el espacio de color CIELAB fueron los siguientes:

Train acc: 0.9249, Train loss: 0.2324, Val accuracy: 0.9271, Val loss: 0.19

La grafica muestra el desempeño del modelo durante el primer entrenamiento

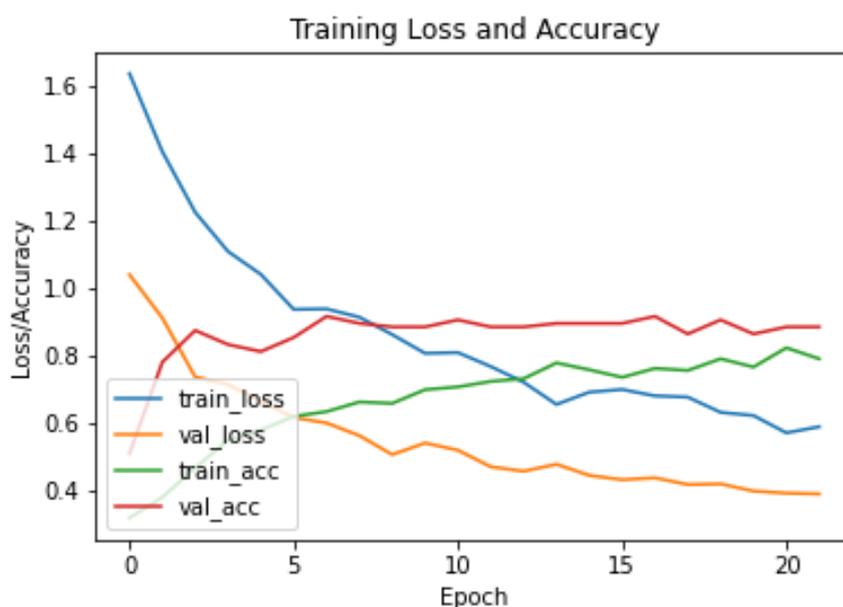


Figura 9. Training loss y accuracy del CIELAB durante el Transfer Learning

Fuente: Elaboración propia

```
[INFO] evaluating after fine-tuning network head...
      precision    recall  f1-score   support

Healthy Lab      0.86      0.91      0.88         33
Nitrogen Lab     0.80      0.75      0.77         32
Phosphorus Lab  0.71      1.00      0.83         29
Potassium Lab    1.00      0.40      0.57         20

   accuracy
macro avg      0.84      0.76      0.76         114
weighted avg   0.83      0.80      0.78         114
```

Figura 10. F1-Score, recall, precision, support obtenido durante Transfer Learning.

Fuente: Elaboración propia

Después del proceso de *fine tuning*, en el que se reentrenaron las capas finales, el modelo tuvo mejor desempeño:



Figura 11. Training loss y accuracy del CIELAB durante el Fine Tuning.

Fuente: Elaboración propia

```
[INFO] evaluating after fine-tuning network...
      precision    recall  f1-score   support

Healthy Lab      0.91      0.97      0.94         33
Nitrogen Lab     0.93      0.88      0.90         32
Phosphorus Lab  0.85      0.97      0.90         29
Potassium Lab   0.94      0.75      0.83         20

 accuracy                   0.90         114
 macro avg                  0.91      0.89      0.90         114
 weighted avg               0.91      0.90      0.90         114
```

Figura 12. F1-Score, recall, precision, support obtenido durante Fine Tuning.

Fuente: Elaboración propia

A continuación, se muestran las matrices de confusión normalizadas:

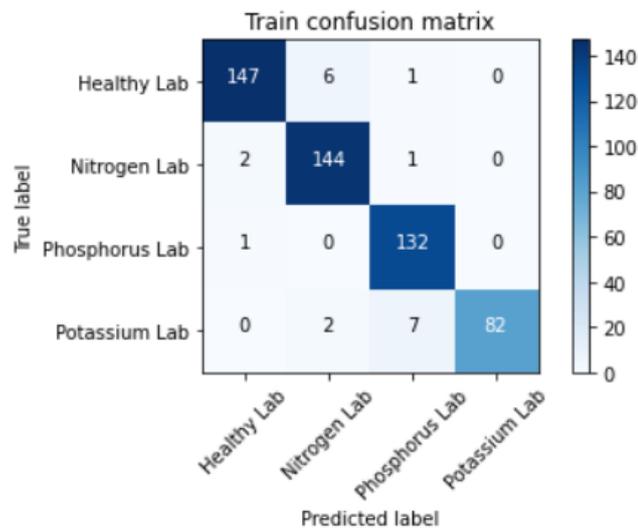


Figura 13. Matriz de confusión del conjunto del entrenamiento.

Fuente: Elaboración propia

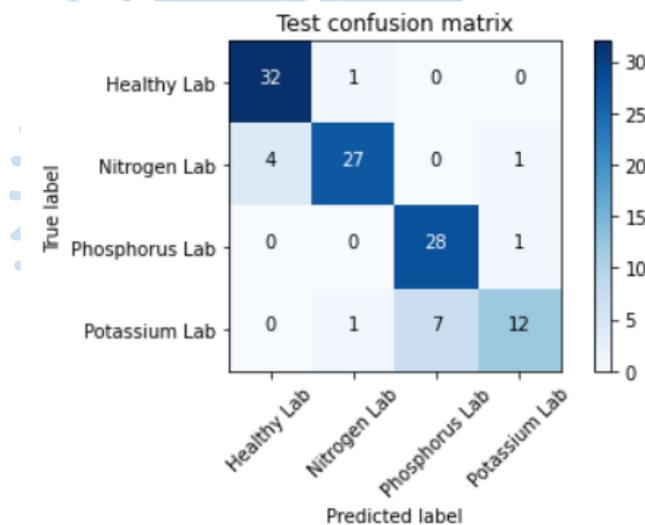


Figura 14. Matriz de confusión del conjunto del testeo.

Fuente: Elaboración propia

Como se puede observar de las matrices de confusión, el mayor error se obtuvo en la diferenciación de hojas deficientes en potasio y hojas deficientes en fósforo. El modelo “confunde las hojas deficientes en potasio como deficientes en fósforo, la situación inversa tiene menor margen de error, es decir que al clasificar las hojas de fósforo confunde una cantidad mucho menor como hojas deficientes en potasio.

En general es evidente como la actualización de los nuevos pesos en las últimas capas ha mejorado notablemente el desempeño del modelo. Esto se debe a que son estas últimas capas en las que confluyen las características genéricas, definidas y extraídas de las capas iniciales, para extraer los rasgos distintivos que terminan por definir la clasificación de una imagen.

5.2 HSV

Los resultados obtenidos utilizando las imágenes resultantes del preprocesamiento en el espacio de color HSV fueron los siguientes:

Train acc: 0.9552, Train loss: 0.1250, Val accuracy: 0.9375, Val loss: 0.2182

La grafica muestra el desempeño del modelo durante el primer entrenamiento

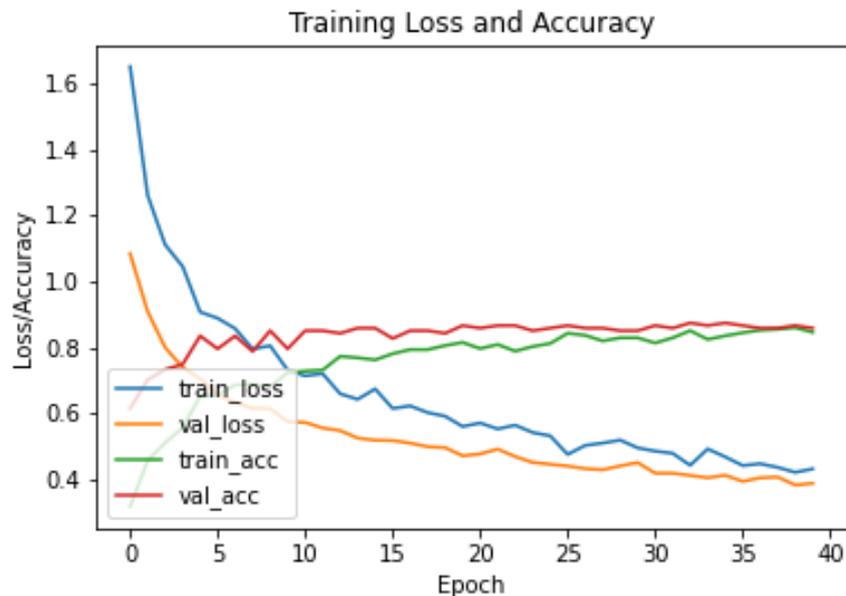


Figura 15. Training loss y accuracy del HSV durante el Transfer Learning

Fuente: Elaboración propia

```
[INFO] evaluating after fine-tuning network head...
      precision    recall  f1-score   support

 healthy         0.94     0.82     0.88         40
  nitrogen         0.76     0.94     0.84         33
 phosphorus       0.84     0.98     0.90         42
  potassium       1.00     0.69     0.81         32

 accuracy                   0.86         147
 macro avg                 0.88     0.86     0.86         147
 weighted avg              0.88     0.86     0.86         147
```

Figura 16. F1-Score, recall, precision, support obtenido durante Transfer Learning.

Fuente: Elaboración propia

Después del proceso de *fine tuning*, en el que se reentrenaron las capas finales, el modelo tuvo mejor desempeño:

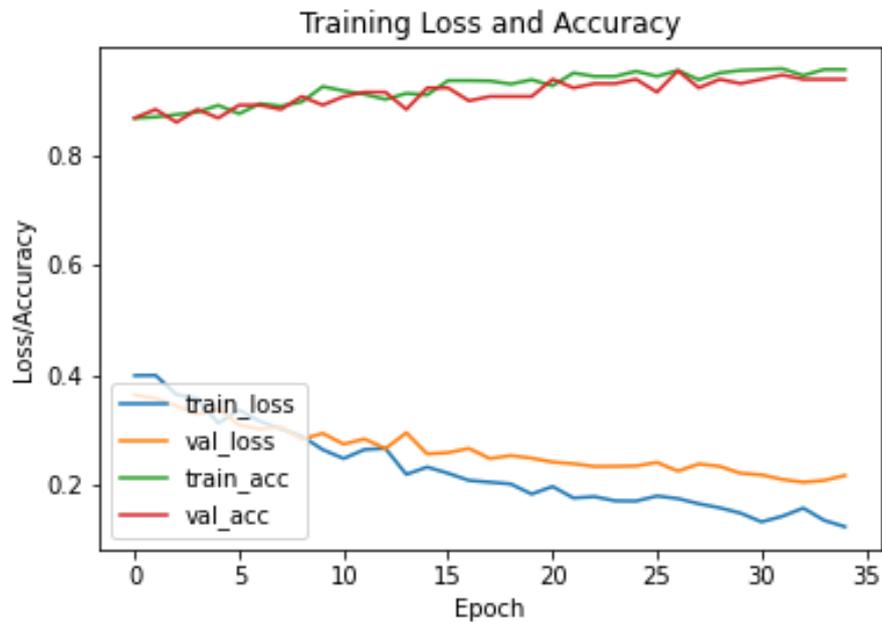


Figura 17. Training loss y accuracy del HSV durante el Fine Tuning.

Fuente: Elaboración propia

```
[INFO] evaluating after fine-tuning network...
      precision    recall  f1-score   support

 healthy         0.92     0.90     0.91         40
  nitrogen         0.91     0.88     0.89         33
 phosphorus         0.93     1.00     0.97         42
  potassium         0.97     0.94     0.95         32

 accuracy                   0.93         147
 macro avg                   0.93     0.93     0.93         147
 weighted avg                   0.93     0.93     0.93         147
```

Figura 18. F1-Score, recall, precision, support obtenido durante Fine Tuning

Fuente: Elaboración propia

A continuación, se muestran las matrices de confusión normalizadas:

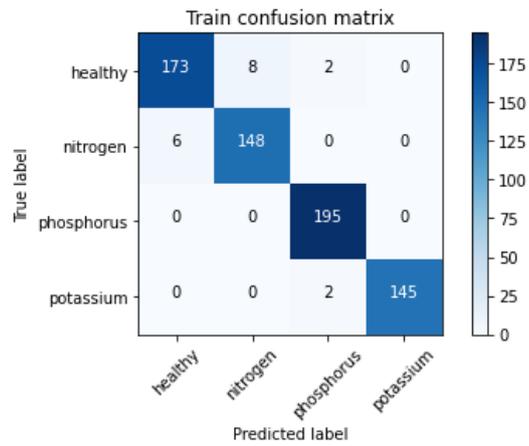


Figura 19. Matriz de confusión del conjunto del entrenamiento.

Fuente: Elaboración propia

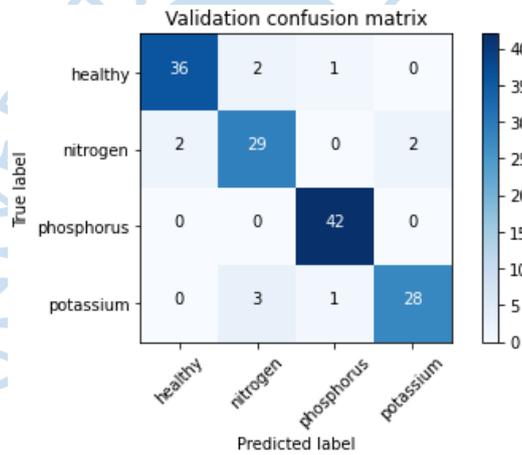


Figura 20. Matriz de confusión del conjunto de validación

Fuente: Elaboración propia

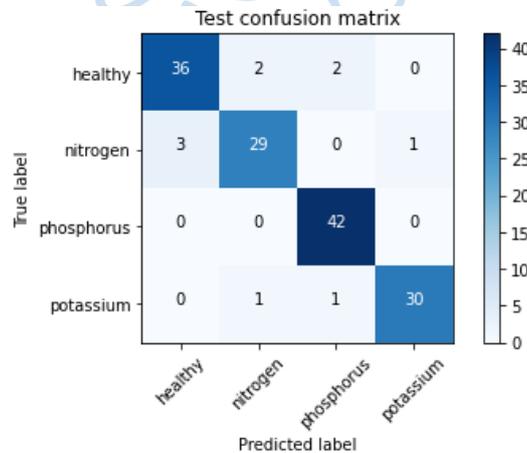


Figura 21. Matriz de confusión del conjunto del testeo.

Fuente: Elaboración propia

5.3 YCbCr

Utilizando el Fine tuning se obtiene un valor de:

Validation accuracy:0.9062

Validation los:0.2524

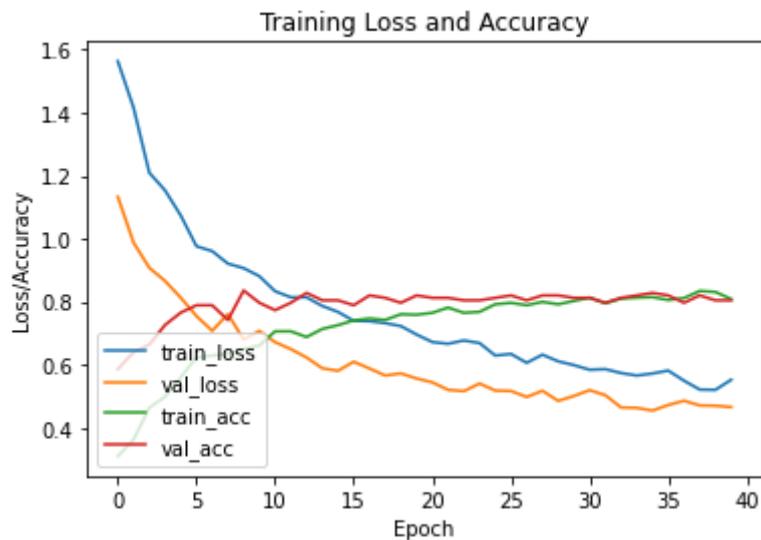


Figura 22 Training loss y accuracy del YCbCr durante el Transfer Learning.

Fuente: Elaboración propia

```
[INFO] evaluating after fine-tuning network head...
      precision    recall  f1-score   support

healthy ycbcr      0.97      0.80      0.88         40
nitrogen ycbcr     0.74      0.79      0.76         33
phosphorus ycbcr  0.82      1.00      0.90         42
potassium ycbcr   0.90      0.80      0.85         35

   accuracy
macro avg      0.86      0.85      0.85        150
weighted avg   0.86      0.85      0.85        150
```

Figura 23 F1-Score, recall, precision, support obtenido durante Transfer Learning.

Fuente: Elaboración propia

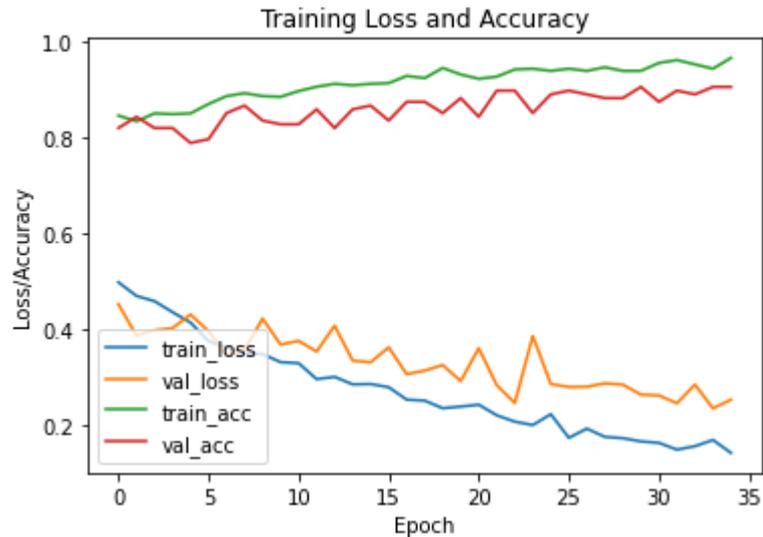


Figura 24 Training loss y accuracy del YCbCr durante el Fine Tuning.

Fuente: Elaboración propia

```
[INFO] evaluating after fine-tuning network...
```

	precision	recall	f1-score	support
healthy ybcr	1.00	0.90	0.95	40
nitrogen ybcr	0.92	1.00	0.96	33
phosphorus ybcr	0.93	1.00	0.97	42
potassium ybcr	0.97	0.91	0.94	35
accuracy			0.95	150
macro avg	0.95	0.95	0.95	150
weighted avg	0.96	0.95	0.95	150

Figura 25 F1-Score, recall, precision, support obtenido durante Fine Tuning

Fuente: Elaboración propia

A continuación, se muestran las matrices de confusión normalizadas:

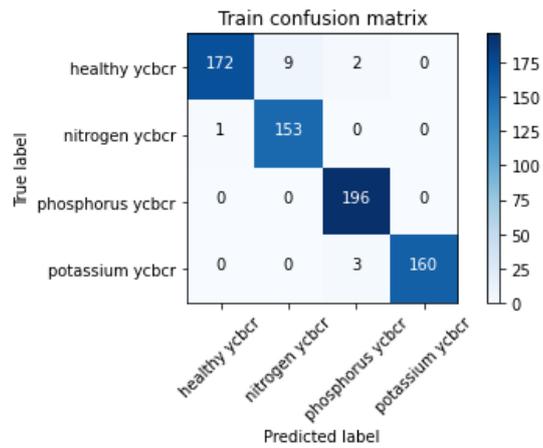


Figura 26 Matriz de confusión del conjunto del entrenamiento.

Fuente: Elaboración propia

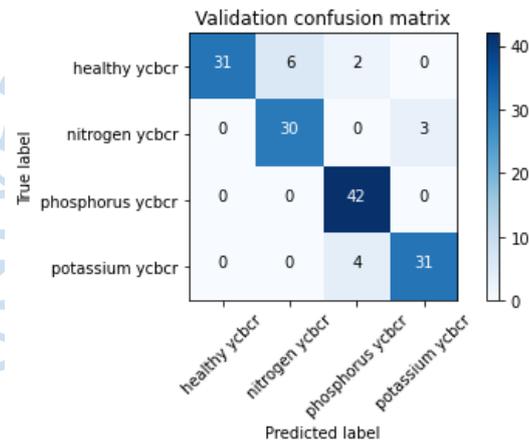


Figura 27 Matriz de confusión del conjunto de validación.

Fuente: Elaboración propia

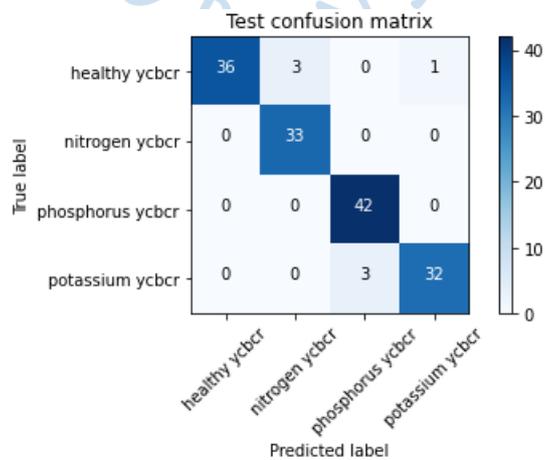


Figura 28 Matriz de confusión del conjunto del testeo.

Fuente: Elaboración propia

5.4 Ecuación de histograma

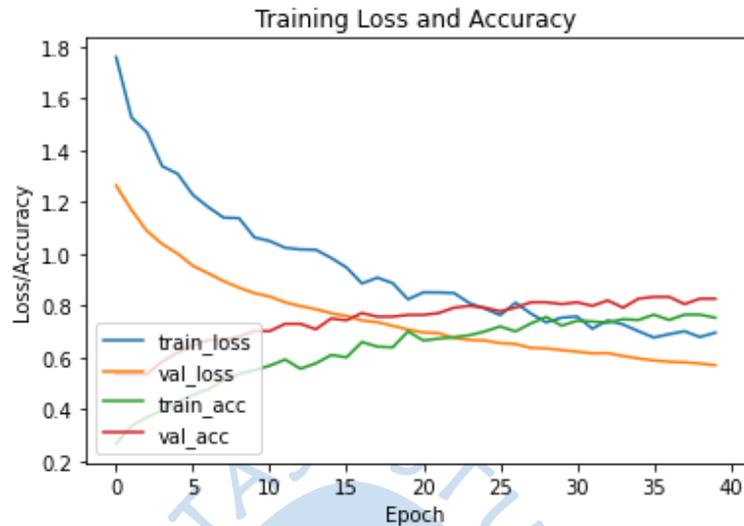


Figura 29 F1-Gráfica de Training Loss y Accuracy durante Transfer Learning

Fuente: Elaboración propia

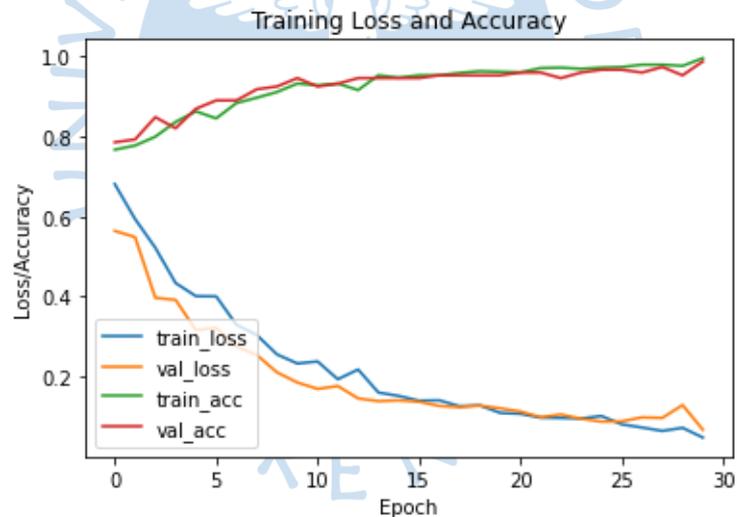


Figura 30 F1-Gráfica de Training Loss y Accuracy durante Fine Tuning

Fuente: Elaboración propia

Como parte del análisis inicial y haciendo una comparación entre ambas gráficas presentes en las figuras 1 y la figura 2, podemos notar claramente como ambas poseen curvas con una tendencia exponencial en el caso de las curvas de Loss ya sea en el entrenamiento o validación ambas tienen a decrecer exponencialmente y ambas llegando a valores muy cercanos en todo el recorrido lo cual nos muestra lo bien que está aprendiendo nuestro modelo así mismo podemos ver como la disminución de la tasa de aprendizaje durante el Fine Tuning ha causado un gran impacto durante esta parte del entrenamiento logrando valores

bastante satisfactorios, teniendo como resultado final en la exactitud de validación un valor de 98.61% siendo un valor bastante parecido al de entrenamiento que fue de 99.41% así mismo en la pérdida de validación obtuvimos 6.85% y en el de entrenamiento 4.9% lo cual indica que nuestro modelo ha logrado resultados bastante satisfactorios.

En las figuras 3 y 4 podemos observar que los valores de “precision”, “recall”, “f1-score” han mejorado en todos los casos llegando a valores satisfactorios. La figura 4 será analizada a profundidad con apoyo de la matriz de confusión para poder realizar un mejor análisis.

```
[INFO] evaluating after fine-tuning network head...
      precision  recall  f1-score  support
healthy      0.87    0.82    0.85     40
nitrogen     0.75    0.73    0.74     33
phosphorus  0.75    0.95    0.84     42
potassium    0.96    0.74    0.84     35

accuracy                    0.82    150
macro avg      0.83    0.81    0.82    150
weighted avg   0.83    0.82    0.82    150
```

Figura 31 F1-Score, recall, precision, support obtenido durante Transfer Learning

Fuente: Elaboración propia

```
[INFO] evaluating after fine-tuning network...
      precision  recall  f1-score  support
healthy      0.93    0.97    0.95     40
nitrogen     0.97    0.91    0.94     33
phosphorus  0.95    1.00    0.98     42
potassium    1.00    0.94    0.97     35

accuracy                    0.96    150
macro avg      0.96    0.96    0.96    150
weighted avg   0.96    0.96    0.96    150
```

Figura 32 F1-Score, recall, precision, support obtenido durante Fine Tuning

Fuente: Elaboración propia

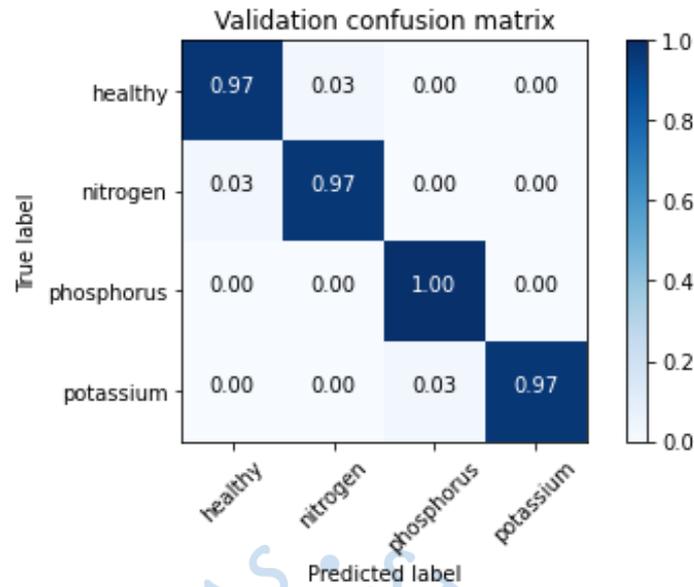


Figura 33 Matriz de confusión del conjunto de validación.

Fuente: Elaboración propia

Empezaremos el análisis de la matriz de confusión y posterior nos apoyaremos en esta para poder analizar los valores obtenidos en la Figura 4. Como ya sabemos el eje de “True Label” es lo que se tenía que predecir y el eje de “Predicted Label” es lo que el clasificador predijo.

Partiendo de lo mencionado anteriormente podemos introducir el significado de los verdaderos positivos que vendrían a ser cuando el clasificador acierta, luego el falso positivo es cuando el clasificador dice que es una clase, pero se equivoca, el verdadero negativo que es cuando el clasificador dice que no es una clase y verdaderamente no es esa clase y el falso negativo sería cuando dice que no es una clase, pero esta si es. Lo que se busca es que nuestro clasificador siempre acierte, pero esto solo es un ideal, pero claramente podemos observar que se logra una diagonal con valores muy cercanos a 1 lo cual es otro factor que nos indica de lo bien que ha aprendido nuestro modelo después de haberle realizado “Transfer Learning” y “Fine Tuning”.

Ahora relacionado la Figura 4 con la Figura 5 empezaremos analizando el “Accuracy”, esta tiene una estrecha relación con la matriz de confusión ya que el “Accuracy” es la cantidad de veces que nuestro modelo clasificador acertó una afirmación sobre el total de los datos de entrada, a partir de esto podemos observar que nuestra matriz de confusión y el reporte obtenido en cuanto a “Accuracy” mantiene una gran similitud lo cual reafirma el impacto positivo que ha causado el optimizador y el modelo usado para el diseño del clasificador.

Analizando “Precision” con un valor de 93% para Healthy, 97% para Nitrogen, 95% para Phosphorus y 100% para Potassium. Empezaremos entendiendo que este hace referencia a la cantidad de casos verdaderos positivos sobre toda la cantidad total de todo lo

que el clasificador era positivo, de otra manera se evalúa de todo lo que el clasificador predijo que era cierto que porcentaje de este es cierto, analizando la Figura 4 y 5 desde este punto de vista podemos observar que estos concuerdan mucho y hemos obtenido valores que tienen bastante similitud entre ellos lo cual reafirma el buen performance que ha realizado durante el entrenamiento nuestro modelo.

“Recall” o exhaustividad, este parámetro tiene una definición bastante parecida a la de “Precision”, pero esta se diferencia en que este compara todo lo que el clasificador afirmaba con lo que no dijo que era así. Interpretando este parámetro podemos decir en el caso de Healthy que el clasificador puede identificar el 97% de los casos que si es esta clase.

En el “F1-score”, se hace una combinación entre las medias de precisión y recall de un solo valor, esto resulta práctico porque hace más fácil el poder comparar el performance de la precisión y la exhaustividad entre varias soluciones. Interpretando el valor hallado que este dependerá de los valores de “Recall” y “Precision” y mientras estos valores sean más altos nuestro “F1-score” será más alto, en los resultados obtenido podemos observar que este valor corresponde a lo planteado anteriormente.

Como conclusión del análisis realizado podemos observar que para poder analizar el “F1-score”, ha sido necesario haber entendido y analizado los valores de “Precision” y “Recall”. Otra cosa importante es la estrecha relación que hay entre la matriz de confusión y las métricas analizadas ya que como hemos podido observar estas se validan mutuamente.

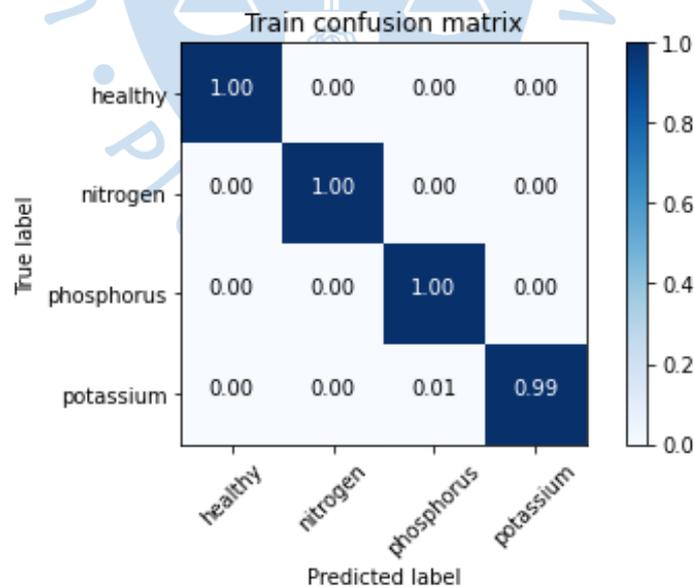


Figura 34 Matriz de confusión del conjunto de entrenamiento.

Fuente: Elaboración propia

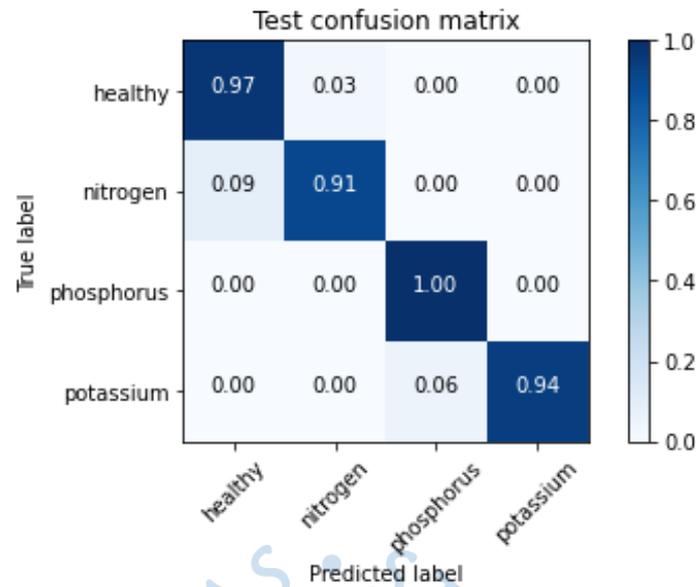


Figura 35 Matriz de confusión del conjunto de testeo.

Fuente: Elaboración propia

Como parte final del análisis podemos verificar el gran performance que ha tenido nuestro modelo al probarlo con imágenes que este no ha visto obteniendo valores bastante satisfactorios los cuales validan todo lo que ha sido identificado anteriormente, llegando la conclusión de la gran importancia e influencia de los hiperparámetros y así mismo del gran impacto del “Fine Tuning” en nuestro modelo.

```

43/43 [=====] - 10s 243ms/step - loss: 0.8092 - accuracy: 0.6853 - val_loss: 0.6668 - val_accuracy: 0.7986
Epoch 25/40
43/43 [=====] - 10s 241ms/step - loss: 0.7878 - accuracy: 0.7000 - val_loss: 0.6648 - val_accuracy: 0.7917
Epoch 26/40
43/43 [=====] - 10s 241ms/step - loss: 0.7629 - accuracy: 0.7191 - val_loss: 0.6548 - val_accuracy: 0.7778
Epoch 27/40
43/43 [=====] - 10s 240ms/step - loss: 0.8107 - accuracy: 0.7000 - val_loss: 0.6512 - val_accuracy: 0.7917
Epoch 28/40
43/43 [=====] - 10s 244ms/step - loss: 0.7689 - accuracy: 0.7309 - val_loss: 0.6363 - val_accuracy: 0.8125
Epoch 29/40
43/43 [=====] - 10s 243ms/step - loss: 0.7348 - accuracy: 0.7559 - val_loss: 0.6342 - val_accuracy: 0.8125
Epoch 30/40
43/43 [=====] - 11s 245ms/step - loss: 0.7531 - accuracy: 0.7221 - val_loss: 0.6277 - val_accuracy: 0.8056
Epoch 31/40
43/43 [=====] - 11s 244ms/step - loss: 0.7577 - accuracy: 0.7412 - val_loss: 0.6217 - val_accuracy: 0.8125
Epoch 32/40
43/43 [=====] - 10s 242ms/step - loss: 0.7106 - accuracy: 0.7382 - val_loss: 0.6152 - val_accuracy: 0.7986
Epoch 33/40
43/43 [=====] - 10s 242ms/step - loss: 0.7423 - accuracy: 0.7338 - val_loss: 0.6161 - val_accuracy: 0.8194
Epoch 34/40
43/43 [=====] - 11s 246ms/step - loss: 0.7269 - accuracy: 0.7471 - val_loss: 0.6048 - val_accuracy: 0.7917
Epoch 35/40
43/43 [=====] - 10s 243ms/step - loss: 0.7008 - accuracy: 0.7441 - val_loss: 0.5955 - val_accuracy: 0.8264
Epoch 36/40
43/43 [=====] - 11s 247ms/step - loss: 0.6763 - accuracy: 0.7647 - val_loss: 0.5876 - val_accuracy: 0.8333
Epoch 37/40
43/43 [=====] - 10s 244ms/step - loss: 0.6894 - accuracy: 0.7441 - val_loss: 0.5827 - val_accuracy: 0.8333
Epoch 38/40
43/43 [=====] - 11s 248ms/step - loss: 0.7006 - accuracy: 0.7647 - val_loss: 0.5805 - val_accuracy: 0.8056
Epoch 39/40
43/43 [=====] - 10s 244ms/step - loss: 0.6782 - accuracy: 0.7647 - val_loss: 0.5759 - val_accuracy: 0.8264
Epoch 40/40
43/43 [=====] - 11s 245ms/step - loss: 0.6951 - accuracy: 0.7529 - val_loss: 0.5693 - val_accuracy: 0.8264

```

Figura 36 Resultados durante el entrenamiento de la red, en el “Transfer Learning”.

Fuente: Elaboración propia

```

43/43 [=====] - 11s 248ms/step - loss: 0.5935 - accuracy: 0.7765 - val_loss: 0.5484 - val_accuracy: 0.7917
Epoch 3/30
43/43 [=====] - 11s 246ms/step - loss: 0.5220 - accuracy: 0.7985 - val_loss: 0.3973 - val_accuracy: 0.8472
Epoch 4/30
43/43 [=====] - 11s 250ms/step - loss: 0.4340 - accuracy: 0.8353 - val_loss: 0.3920 - val_accuracy: 0.8194
Epoch 5/30
43/43 [=====] - 11s 253ms/step - loss: 0.4014 - accuracy: 0.8618 - val_loss: 0.3168 - val_accuracy: 0.8681
Epoch 6/30
43/43 [=====] - 11s 257ms/step - loss: 0.4008 - accuracy: 0.8441 - val_loss: 0.3220 - val_accuracy: 0.8889
Epoch 7/30
43/43 [=====] - 11s 249ms/step - loss: 0.3300 - accuracy: 0.8824 - val_loss: 0.2755 - val_accuracy: 0.8889
Epoch 8/30
43/43 [=====] - 11s 248ms/step - loss: 0.3047 - accuracy: 0.8956 - val_loss: 0.2548 - val_accuracy: 0.9167
Epoch 9/30
43/43 [=====] - 11s 249ms/step - loss: 0.2561 - accuracy: 0.9099 - val_loss: 0.2114 - val_accuracy: 0.9236
Epoch 10/30
43/43 [=====] - 11s 246ms/step - loss: 0.2335 - accuracy: 0.9309 - val_loss: 0.1862 - val_accuracy: 0.9444
Epoch 11/30
43/43 [=====] - 11s 247ms/step - loss: 0.2386 - accuracy: 0.9265 - val_loss: 0.1702 - val_accuracy: 0.9236
Epoch 12/30
43/43 [=====] - 11s 247ms/step - loss: 0.1943 - accuracy: 0.9309 - val_loss: 0.1773 - val_accuracy: 0.9306
Epoch 13/30
43/43 [=====] - 11s 246ms/step - loss: 0.2186 - accuracy: 0.9147 - val_loss: 0.1465 - val_accuracy: 0.9444
Epoch 14/30
43/43 [=====] - 11s 246ms/step - loss: 0.1612 - accuracy: 0.9515 - val_loss: 0.1397 - val_accuracy: 0.9444
Epoch 15/30
43/43 [=====] - 11s 246ms/step - loss: 0.1528 - accuracy: 0.9456 - val_loss: 0.1416 - val_accuracy: 0.9444
Epoch 16/30
43/43 [=====] - 10s 243ms/step - loss: 0.1409 - accuracy: 0.9515 - val_loss: 0.1382 - val_accuracy: 0.9444
Epoch 17/30
43/43 [=====] - 10s 244ms/step - loss: 0.1418 - accuracy: 0.9515 - val_loss: 0.1277 - val_accuracy: 0.9514
Epoch 18/30
43/43 [=====] - 10s 242ms/step - loss: 0.1271 - accuracy: 0.9574 - val_loss: 0.1248 - val_accuracy: 0.9514
Epoch 19/30
43/43 [=====] - 11s 245ms/step - loss: 0.1291 - accuracy: 0.9618 - val_loss: 0.1287 - val_accuracy: 0.9514
Epoch 20/30
43/43 [=====] - 11s 245ms/step - loss: 0.1105 - accuracy: 0.9603 - val_loss: 0.1219 - val_accuracy: 0.9514
Epoch 21/30
43/43 [=====] - 10s 241ms/step - loss: 0.1082 - accuracy: 0.9588 - val_loss: 0.1138 - val_accuracy: 0.9583
Epoch 22/30
43/43 [=====] - 11s 248ms/step - loss: 0.0986 - accuracy: 0.9695 - val_loss: 0.0998 - val_accuracy: 0.9583
Epoch 23/30
43/43 [=====] - 10s 242ms/step - loss: 0.0978 - accuracy: 0.9706 - val_loss: 0.1064 - val_accuracy: 0.9444
Epoch 24/30
43/43 [=====] - 10s 240ms/step - loss: 0.0966 - accuracy: 0.9676 - val_loss: 0.0959 - val_accuracy: 0.9583
Epoch 25/30
43/43 [=====] - 10s 240ms/step - loss: 0.1027 - accuracy: 0.9706 - val_loss: 0.0888 - val_accuracy: 0.9653
Epoch 26/30
43/43 [=====] - 10s 239ms/step - loss: 0.0820 - accuracy: 0.9721 - val_loss: 0.0895 - val_accuracy: 0.9653
Epoch 27/30
43/43 [=====] - 10s 243ms/step - loss: 0.0740 - accuracy: 0.9779 - val_loss: 0.0993 - val_accuracy: 0.9583
Epoch 28/30
43/43 [=====] - 10s 240ms/step - loss: 0.0655 - accuracy: 0.9779 - val_loss: 0.0978 - val_accuracy: 0.9722
Epoch 29/30
43/43 [=====] - 10s 240ms/step - loss: 0.0732 - accuracy: 0.9750 - val_loss: 0.1300 - val_accuracy: 0.9514
Epoch 30/30
43/43 [=====] - 10s 240ms/step - loss: 0.0490 - accuracy: 0.9941 - val_loss: 0.0685 - val_accuracy: 0.9861

```

Figura 37 Resultados durante el entrenamiento de la red, en el “Fine Tuning”

Fuente: Elaboración propia

5.5 Análisis de Resultados

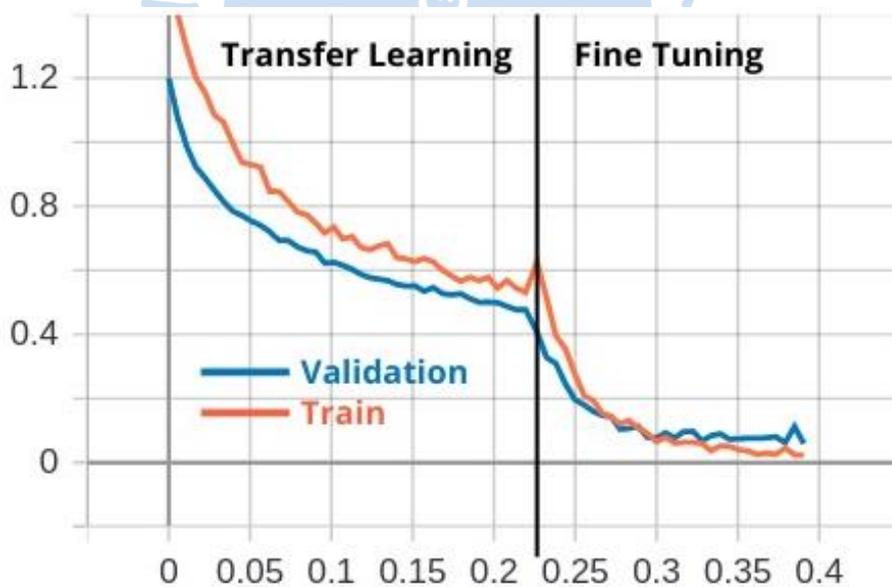
Analizando los cuatro métodos de preprocesamiento utilizados en esta investigación se observó que todos tienen una tendencia creciente en términos de precisión y una disminución del error. Sin embargo, la superioridad del ecualizador de histograma es evidente, con una curva suave que logra la mayor precisión y el mínimo error en comparación con los otros métodos.

Tabla 16 Comparación de los métodos de preprocesamiento

Preprocessing method	Train_acc	Train_loss	Val_acc	Val_loss
Histogram equalizer	0.9928	0.0231	0.9861	0.0593
<i>HSV</i>	0.9552	0.1250	0.9375	0.2182
<i>CIELAB</i>	0.9249	0.2324	0.9375	0.1631
<i>YCbCr</i>	0.9608	0.1439	0.8984	0.2424

Fuente: Elaboración propia

Además, es importante mencionar la gran contribución de Fine Tuning a los resultados finales obtenido en las Figuras 38 y 39, los cuales son gráficos de Tensor Board que muestran el impacto de FT, después de aplicar Transfer Learning.

**Figura 38** Valor de pérdida con Transfer Learning y Fine Tuning.

Fuente: Elaboración propia

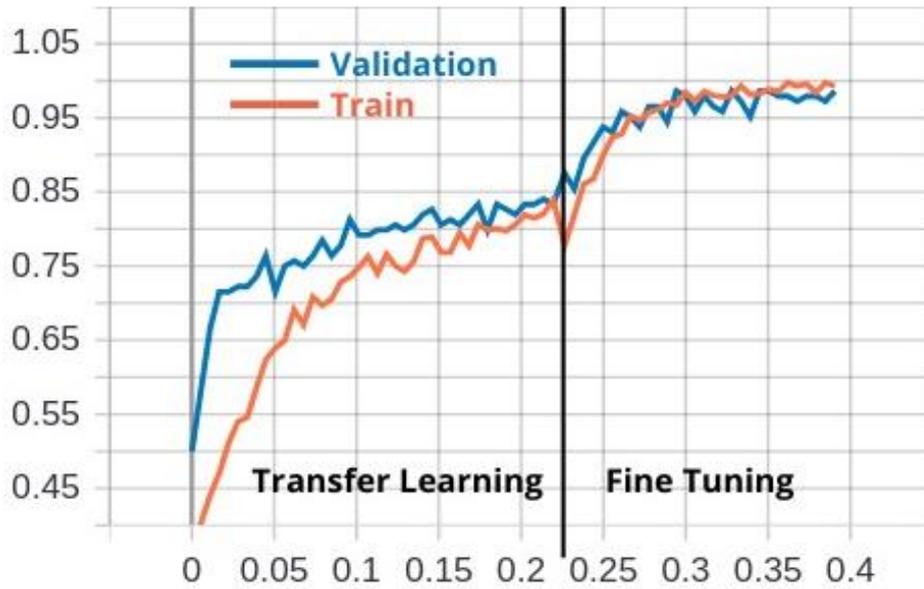


Figura 39 Valor de precisión con Transfer Learning y Fine Tuning.

Fuente: Elaboración propia

5.6 Plataforma

Se decidió diseñar un prototipo de plataforma online para presentarlo como una herramienta para detectar carencias de nutrientes en la planta de banano, con el fin de ser utilizado en el sector agrícola de nuestra región. Para el desarrollo de la plataforma, se utilizó un micro-framework llamado "flask". El flask es parte del backend de la plataforma, donde se ejecutan todas las operaciones. Incluye las etapas del preprocesamiento que se debe seguir la imagen para entrar en la red y posteriormente ser clasificado, permitiendo al usuario cargar una imagen en cualquier tamaño.

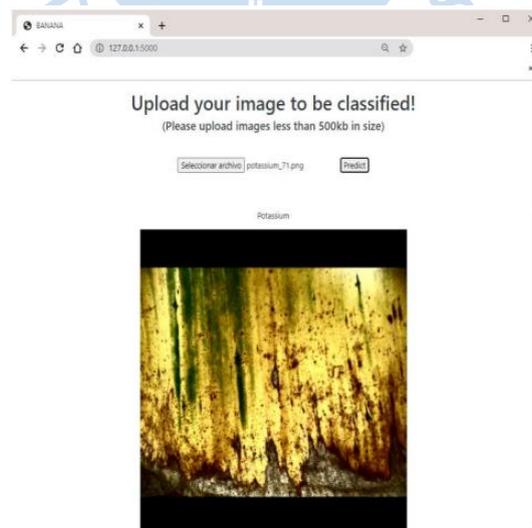


Figura 40 Prototipo de plataforma

Fuente: Elaboración propia

5.7 Plagas

Los resultados obtenidos en el desarrollo de la detección de plagas fueron considerablemente buenos, en los tres tipos de plagas que se escogieron para detectar. Esto se observa en los siguientes ejemplos de detección, en los cuales se presenta los insectos detectado con su porcentaje de exactitud.

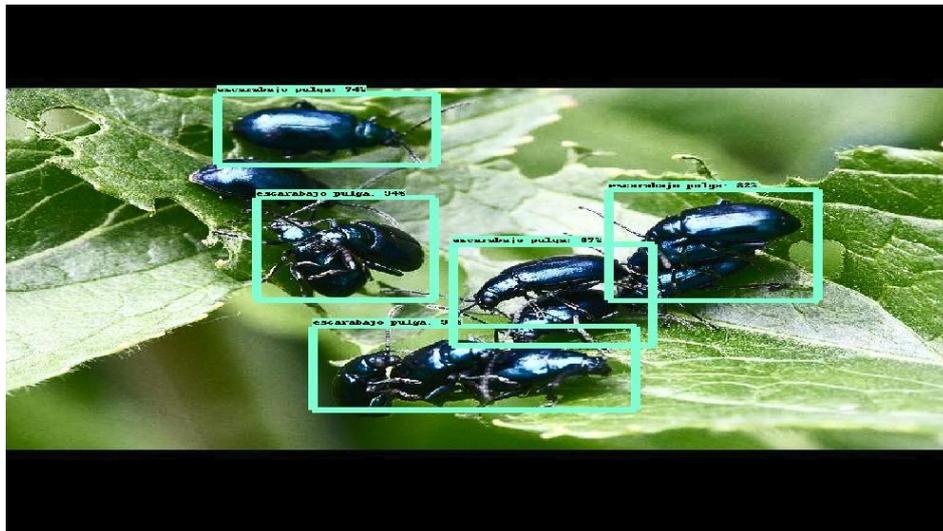


Figura 41_Detección del escarabajo pulga.

Fuente: Elaboración propia



Figura 42_Detección de mosca blanca.

Fuente: Elaboración propia

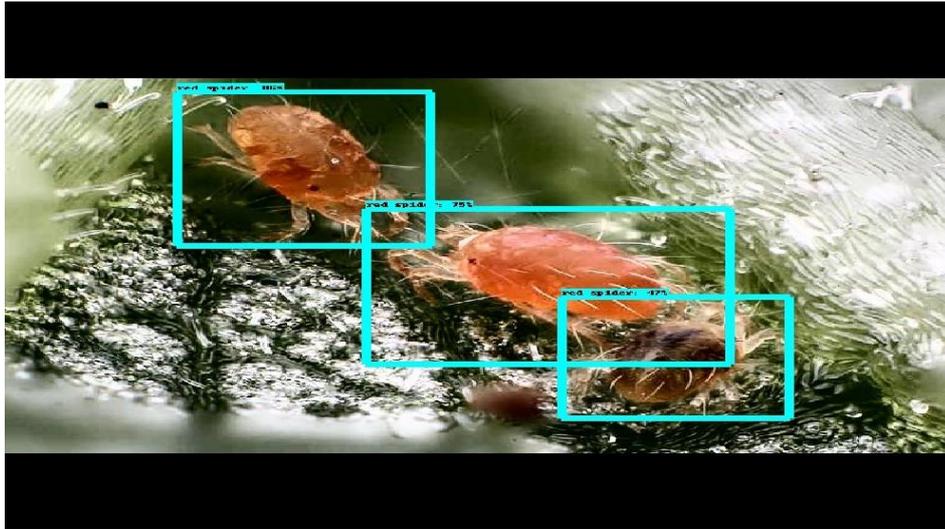


Figura 43 Detección de araña roja.

Fuente: Elaboración propia

Sin embargo, aún hace falta desarrollar el modelo, ya que no es del todo preciso a la hora de reconocer todos los insectos presentes en la imagen sobre todo en donde se presentan un amplio valor de insectos en la imagen y si estos se encuentran bastante acumulados en una zona.



Figura 44 Agrupaciones de mosca blanca en distintos puntos de la imagen

Fuente: Elaboración propia

Conclusiones

- Para aumentar el margen de ganancia y poder competir con productores extranjeros, se debe iniciar la implementación de tecnología la cual permita ahorrar o abaratar costos.
- El banano orgánico es un valioso producto de exportación, significa una fuente importante de ingresos en el norte del país. Utilizando *machine learning* podemos monitorear la deficiencia de nutrientes en la zona de cultivo y controlar plagas.
- La correcta aplicación de aditivos al banano orgánico es de gran importancia ya que permiten tener al cultivo en óptimas condiciones de crecimiento. Dentro de los aditivos más importantes usados en la fertilización por riego están: nitrógeno(N), fósforo(P), potasio(K), magnesio (Mg), calcio (Ca) y azufre(S).
- El sistema de detección e identificación de plagas ayuda al agricultor en el reconocimiento de estas, y la información recolectada podría ser almacenada en una base de datos para futuro análisis estadístico.
- Es posible generar una base de datos con la información recopilada. Se puede mostrar dicha data en una página web, de tal manera que cualquier usuario pueda acceder a ella de manera remota.
- Google Colab y Microsoft Visual Studio representan herramientas muy útiles para la programación en entornos de Python. Google Colab es de mayor utilidad para el entrenamiento de la red neuronal, mientras que Microsoft Visual Studio facilita el trabajo con servidores locales.
- El modelo HSV es una transformación no lineal del sistema RGB. Por lo tanto, es posible transformar directamente un color desde el sistema HSV al sistema RGB, y viceversa. Es ampliamente usado en sistemas de visión artificial.
- El espacio de color CIELAB se perfila como un espacio de color apropiado para manipular y segmentar las tonalidades características de las hojas de banano y contribuir a la detección de parámetros para la identificación de deficiencias.

- El uso del espacio de color YCbCr permite aprovechar su parámetro de luminosidad de tal forma que se denote diferencias entre las distintas categorías de planta que se desea clasificar.
- La ecualización de histograma ha tenido un fuerte impacto en el procesamiento de las imágenes ya que ha permitido obtener una mejora en el contraste debido al balance que realiza en los diferentes canales de color. Además, ha permitido resaltar los contornos de la imagen, lo que es una ventaja para el entrenamiento en la red neuronal.
- Es de gran importancia saber leer e interpretar las métricas obtenidas durante el entrenamiento y testeo, ya que estas nos permiten verificar si el modelo obtenido se ajusta adecuadamente a la realidad de una manera generalizada.
- Todos los métodos de preprocesamiento arrojaron resultados precisos, esto demuestra la gran capacidad de Fine Tuning para mejorar el rendimiento de un modelo previamente entrenado aplicando Transfer Learning. No obstante, esta investigación muestra que es aconsejable realizar una amplia gama de operaciones de preprocesamiento (cambios en el espacio de color, filtrado, ecualización son algunas de las más gratificantes), con el fin de encontrar la entrada más adecuada para un problema de clasificación específico. Para el propósito específico de esta investigación (detección de deficiencia de nutrientes en hojas de banano), el informe de clasificación y las matrices de validación y confusión de pruebas demostraron que la ecualización del histograma en el espacio de color YUV arrojó los mejores resultados. El modelo de clasificación final tiene una precisión muy alta y un error mínimo. La capacidad de clasificación del modelo, combinada con una plataforma en línea, lo convierte en una herramienta atractiva para ayudar en el desarrollo de la agricultura bananera.

Referencias bibliográficas

Ácaro rojo del plátano - EcuRed. (n.d.).

AGROBANCO. (2012). *Asistencia Técnica Dirigida En Cosecha Y Post Cosecha De Banano Orgánico*. 24.

Aruraj, A., Alex, A., Subathra, M. S. P., Sairamya, N. J., Thomas George, S., & Vinodh Edwards, S. E. (2019). Detection and Classification of Diseases of Banana Plant Using Local Binary Pattern and Support Vector Machine. *2nd International Conference on Signal Processing and Communication, ICSPC 2019 - Proceedings*, c, 231–235. <https://doi.org/10.1109/ICSPC46172.2019.8976582>

Berzal, F. (2018). Redes Neuronales & Deep Learning. *Departamento de Ciencias de La Computacion e IA*, 753.

César, R. (2014). *El Banano Peruano, Producto Estrella de Exportación*.

Colaboratory. (n.d.). *Te damos la bienvenida a Colaboratory - Colaboratory*. Te Damos La Bienvenida a Colaboratory Te. Retrieved October 2, 2020, from <https://colab.research.google.com/notebooks/welcome.ipynb?hl=es#scrollTo=-Rh3-Vt9Nev9>

Dinngo. (n.d.). *Design Thinking en Español*. Retrieved November 28, 2020, from <https://www.designthinking.es/inicio/index.php?success1=0>

Gavhale, K. R., Gawande, U., & Hajari, K. O. (2014). Unhealthy region of citrus leaf detection using image processing techniques. *2014 International Conference for Convergence of Technology, I2CT 2014*, 2–7. <https://doi.org/10.1109/I2CT.2014.7092035>

Guzmán Velásquez, P. (2014). Estudio experiential de la elaboración de puré de banano orgánico de la Región Piura. *Tesis de Pregrado No Publicado En Ingeniería Industrial y de Sistemas. Universidad de Piura. Facultad de Ingeniería. Programa Académico de Ingeniería Industrial y de Sistemas. Lima, Perú, 01, 91*.

- Haifa. (2014). *Recomendaciones nutricionales para Banana Musa acuminata y Musa balbisiana*. 72. https://www.haifa-group.com/sites/default/files/crop/Banana_Spanish_0.pdf
- Jaramillo, S. (2013). Manual de Aplicabilidad de Buenas practicas Agrícolas de banano. *Agencia Ecuatoriana de Aseguramiento de Las Calidad Del Agro(AGROCALIDAD)*, 1, 1–95. <http://www.agrocalidad.gob.ec/wp-content/uploads/pdf/inocuidad/manuales-aplicabilidad/manual-banano.pdf>
- Michelucci, U. (2018). Applied Deep Learning. In *Applied Deep Learning*. <https://doi.org/10.1007/978-1-4842-3790-8>
- Ministerio de Agricultura y Riego. (2018). *Nota Técnica.Situación Comercial del Banano Orgánico en el Mercado Europeo*. 3–6.
- Picudo negro del plátano - EcuRed*. (n.d.).
- Rosales, S. (2019). *Perú tiene 170,000 ha de plátano y banano orgánico en riesgo por plaga Fusarium*. Gestión.Pe.
- SEMINIS. (2016). *QUE ES LA MOSCA BLANCA?*
- Shung, K. P. (2018). *Accuracy, Precision, Recall or F1? | by Koo Ping Shung | Towards Data Science*.
- Tejada, J. A., & Gara, G. P. P. (2017). LeafCheckIT: A banana leaf analyzer for identifying macronutrient deficiency. *ACM International Conference Proceeding Series*, 458–463. <https://doi.org/10.1145/3162957.3163035>
- Tixier, P., Bugaud, C., Duguet, R., & Salmon, F. (2010). Effect of preharvest and postharvest application of calcium on banana green-life. *Fruits*, 65(4), 201–208. <https://doi.org/10.1051/fruits/2010016>
- Torres, S. (2012). Guía práctica para el manejo de banano orgánico en el valle del Chira. *Hidalgo Impresores E.I.R.L.*, 72. https://www.swisscontact.org/fileadmin/user_upload/COUNTRIES/Peru/Documents/Publications/manual_banano.pdf
- Vargas, J. C., & Rojas, J. (2013). “Manejo Integrado De Banano Organico.” In *Guía Técnica* (Vol. 1, p. 39). <https://doi.org/10.1017/CBO9781107415324.004>
- Wiwart, M., Fordoński, G., Zuk-Gołaszewska, K., & Suchowilska, E. (2009). Early diagnostics of macronutrient deficiencies in three legume species by color image analysis. *Computers and Electronics in Agriculture*. <https://doi.org/10.1016/j.compag.2008.08.003>
- Wu, X., Zhan, C., Lai, Y. K., Cheng, M. M., & Yang, J. (2019). IP102: A large-scale benchmark dataset for insect pest recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2019-June*, 8779–8788. <https://doi.org/10.1109/CVPR.2019.00899>

Zdonik, S., Ning, P., Shekhar, S., Katz, J., & Wu, X. (2013). Two-Dimensional Change Detection. In *Springer*. <https://doi.org/10.1007/978-1-4471-2179-4>





Apéndices





Apéndice A:

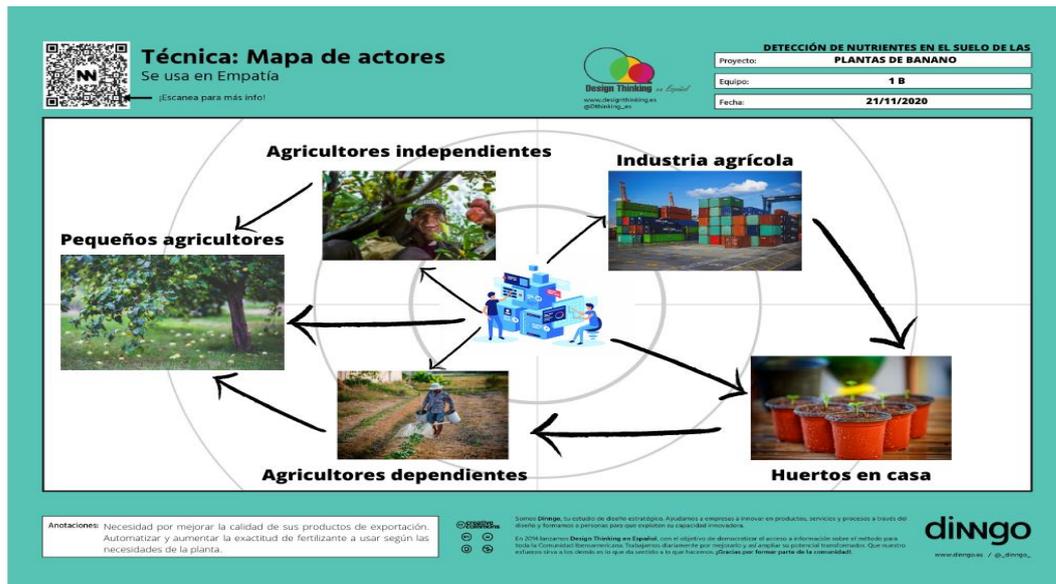


Figura 1_Empatiza, técnica: Mapa de actores.

Fuente: Elaboración propia

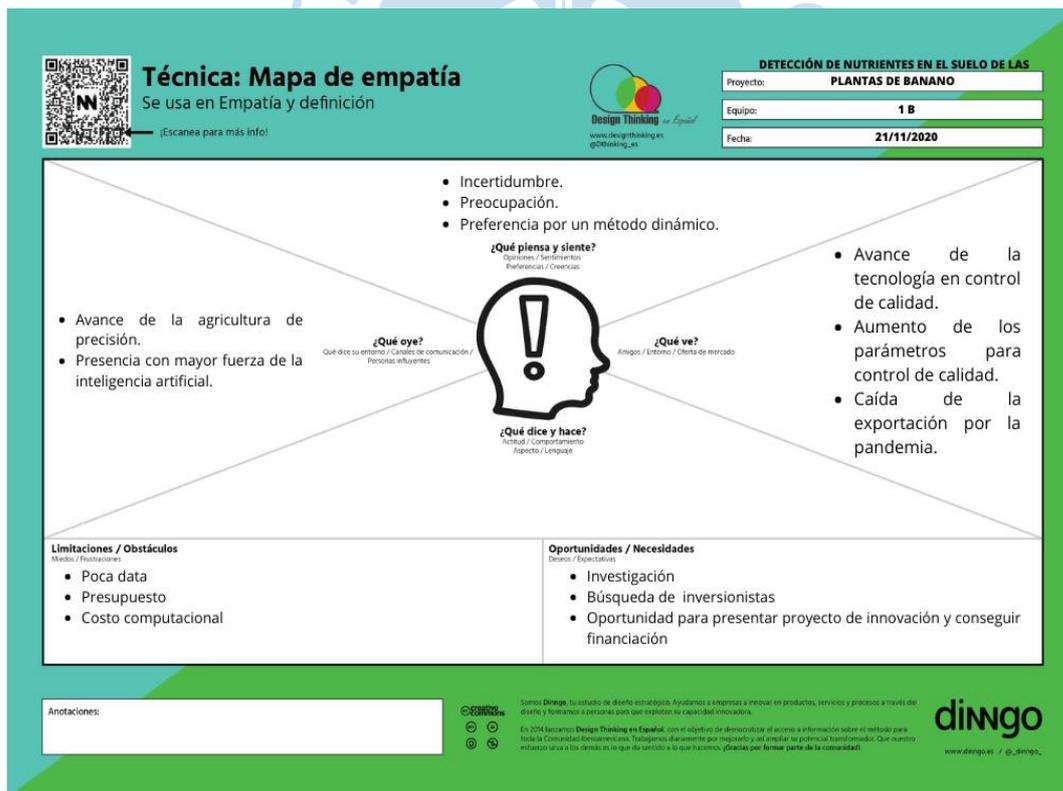


Figura 2 Definición, técnica: Mapa de empatía.

Fuente: Elaboración propia

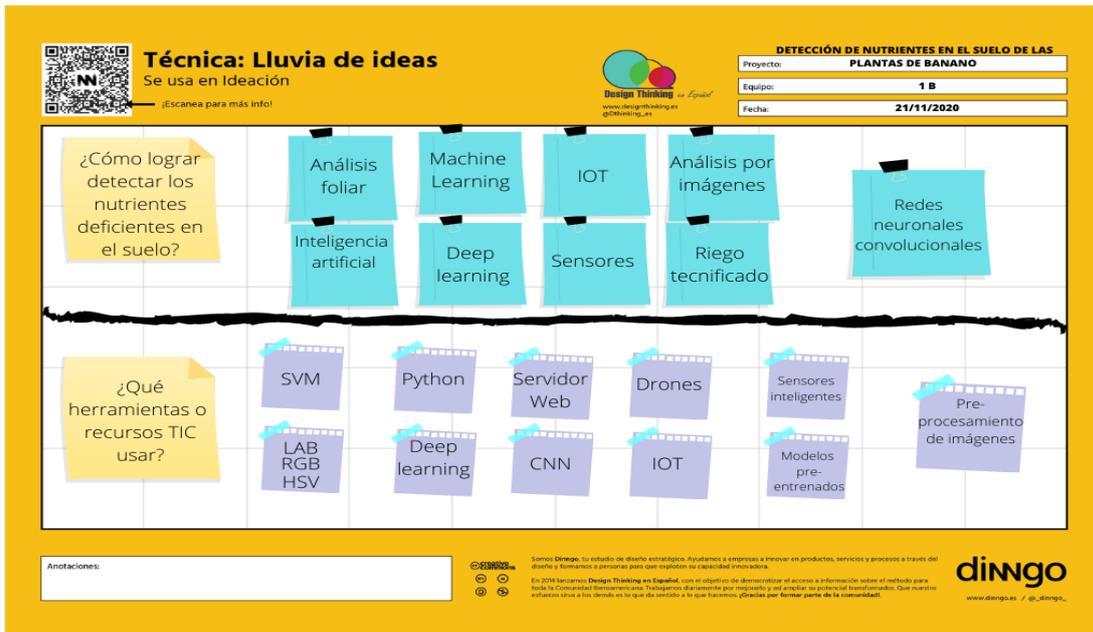


Figura 3 Ideación, técnica: Lluvia de ideas.

Fuente: Elaboración propia

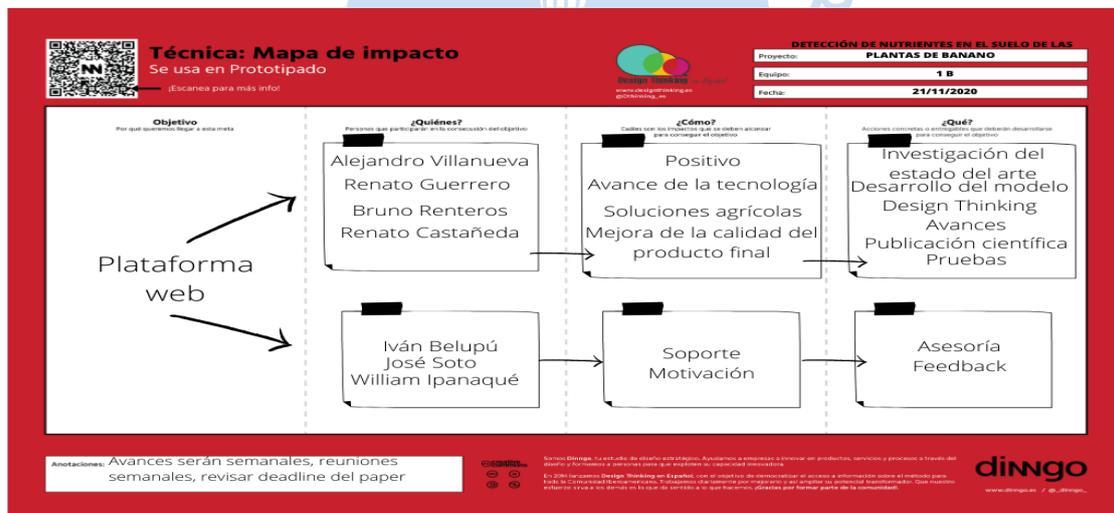


Figura 45 Prototipado, técnica: Mapa de impacto.

Fuente: Elaboración propia



Técnica: Apuntes de Testeo
Se usa en Testeo
¡Escanea para más info!



www.dingoinspanish.es
@DThinking_es

DETECCIÓN DE NUTRIENTES EN EL SUELO DE LAS PLANTAS DE BANANO

Proyecto: **PLANTAS DE BANANO**

Equipo: **1 B**

Fecha: **21/11/2020**

<p>Qué funciona</p> <ul style="list-style-type: none"> Plataforma Detecta deficiencias Detecta plagas Acepta y procesa las imágenes de cualquier tamaño 	<p>Qué se puede mejorar</p> <ul style="list-style-type: none"> Precisión de detección de plagas Plataforma Mejor presentación de la plataforma
<p>Preguntas</p> <ul style="list-style-type: none"> Se necesita registro ¿Qué plantas reconoce? ¿Qué plagas reconoce? ¿Qué requerimientos tiene? Tamaño máximo de las imágenes 	<p>Ideas</p> <ul style="list-style-type: none"> Aplicaciones con drones Contar número de objetos detectados Aumentar data Aplicación en Android

Anotaciones:

Somos Dingo, tu estudio de diseño estratégico. Ayudamos a empresas a innovar en productos, servicios y procesos a través del diseño y formamos a personas para que exploren su capacidad innovadora.

En 2014 fundamos Dingo Thinking en Español, con el objetivo de democratizar el acceso a información sobre el método para toda la Comunidad Bilingüe. Trabajamos de manera por nosotros y así mejorar su potencial bilingüe. Que nuestro esfuerzo sirva a los demás en lugar de cambiarse a lo que hacemos, gracias por formar parte de la comunidad!



www.dingo.es / @dingo_

Figura 5 Testeo, técnica: Apuntes de testeo

Fuente: Elaboración propia

