

**PROTOTIPO DE APLICACIÓN MÓVIL PARA LA IDENTIFICACIÓN DE
MAZORCAS DE CACAO ENFERMAS HACIENDO USO DE VISIÓN POR
COMPUTADORA Y APRENDIZAJE DE MÁQUINA**

**JUAN SEBASTIÁN SERRANO ARENAS
CAMILO ANDRES TORRES VILLAMIZAR**

**UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
GRUPO DE INVESTIGACIÓN: GTI
TELEMÁTICA
PROYECTO DE GRADO II
BUCARAMANGA, SANTANDER
2020**

**PROTOTIPO DE APLICACIÓN MÓVIL PARA LA IDENTIFICACIÓN DE
MAZORCAS DE CACAO ENFERMAS HACIENDO USO DE VISIÓN POR
COMPUTADORA Y APRENDIZAJE DE MÁQUINA**

**JUAN SEBASTIÁN SERRANO ARENAS
CAMILO ANDRES TORRES VILLAMIZAR**

TRABAJO DE GRADO II

**DIRECTOR:
LEONARDO HERNÁN TALERO SARMIENTO M.Sc.**

**CODIRECTOR:
FEISAR ENRIQUE MORENO CORZO M.Sc.**

**ASESOR:
NÉSTOR EDGARDO NIEVES PEÑA**

**UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
GRUPO DE INVESTIGACIÓN: GTI
TELEMÁTICA
PROYECTO DE GRADO II
BUCARAMANGA, SANTANDER
2020**

AGRADECIMIENTOS

Esta tesis no habría sido posible sin la colaboración, ayuda y conocimiento que brindaron de una u otra manera, los directores Leonardo Hernán Talero Sarmiento y Feisar Enrique Moreno Corzo, el grupo de investigación GTI y la línea Telemática de la Universidad Autónoma de Bucaramanga, en especial a los padres y familiares por proporcionar ese amor por la educación, por su trabajo y constancia. A ellos, a los amigos que alentaron y estuvieron siempre al lado, y a los compañeros con los que se compartieron grandes momentos en este último año, dedico este resultado y este hermoso viaje.

“Se que la duda es uno de los nombres de la inteligencia
Mi semblante de estudiante en esencia es ser feliz
Siendo el eterno postulante, el eterno aprendiz.”

- Roberto Musso

CONTENIDO

1. INTRODUCCIÓN	10
2. PLANTEAMIENTO DEL PROBLEMA	11
1.1. PLANTEAMIENTO DEL PROBLEMA	11
1.2. JUSTIFICACIÓN	13
3. OBJETIVOS	15
3.1. OBJETIVO GENERAL	15
3.2. OBJETIVOS ESPECÍFICOS	15
3.3. RESULTADOS ESPERADOS	15
4. REVISIÓN DE LITERATURA	16
4.1. Ecuación de búsqueda y artículos	16
4.2. Antecedentes	17
4.3. Conclusión	20
5. MARCO TEÓRICO	22
5.1. Agricultura y Cacao	22
5.1.1. Cacao – <i>Theobroma Cacao L</i>	22
5.1.2. Cacao en Santander	23
5.1.3. Enfermedades del Cacao	24
5.2. Procesamiento de Imágenes	26
5.2.1. Imágenes	26
5.2.2. Herramientas	26
5.2.3. Red Neuronal Convolucional	27
5.2.4. Segmentación de imágenes	28
5.3. Inteligencia Artificial	29
5.3.1. Aprendizaje automático	29
5.3.2. Aprendizaje Profundo	30
6. DISEÑO METODOLÓGICO	31
6.1. Levantamiento del conjunto de datos	31
6.2. Diseño y desarrollo del prototipo funcional	31
6.3. Entrenamiento y calibración de la máquina de aprendizaje.	38

6.3.1. Evaluación y calibración.....	39
7. RESULTADOS.....	40
7.1. Diseño del prototipo de aplicación móvil para la identificación de mazorcas de cacao enfermas con Monilia y/o Fitóftora:	40
7.1.1. Requerimientos	40
7.1.2. Casos de uso	41
7.1.3. Diagramas de actividades	42
7.1.4. Diagrama de secuencia.....	42
7.1.5. Diagrama de clases.....	43
7.2. Conjunto de datos consolidado.....	44
7.2.1. Visita y tipos de cacao.....	44
7.2.2. Captura de imágenes con mazorcas enfermas y sanas.....	45
7.2.3. Proceso para etiquetar (LabelImg)	45
7.2.4. Conjunto de datos consolidado y Kaggle	47
7.3. Máquina de detección para la identificación de mazorcas de cacao enfermas con Monilia y/o Fitóftora.....	48
7.3.1. Darknet.....	49
7.3.2. Google Colab	49
7.3.3. Datos y etiquetas.....	50
7.3.4. Entrenamiento	50
7.3.5. Modelo entrenado	52
7.4. Prototipo de aplicación funcional para la detección de mazorcas de cacao enfermas con Monilia y/o Fitóftora.....	53
7.4.1. Repositorio TensorFlow (YOLOv4-tflite).....	54
7.4.2. Modelo y conversión de pesos (YOLO v4 a TensorFlow Lite)	55
7.4.3. Configuración etiquetas y modelo personalizado	55
7.4.4. Prototipo de aplicación móvil.....	56
8. DISCUSIONES.....	58
9. CONCLUSIONES	59
10. TRABAJOS FUTUROS	60
11. REFERENCIAS	61
12. ANEXOS.....	67

LISTA DE TABLAS

Tabla 1 Requerimientos funcionales y no funcionales	40
Tabla 2 Caso de uso para iniciar la cámara.....	68
Tabla 3 Caso de uso para detectar la mazorca.	69
Tabla 4 Caso de uso para Modelo YOLO v4 Entrenado	69

LISTA DE FIGURAS

Figura 1	Árbol de problemas.....	13
Figura 2	Ecuación de búsqueda	16
Figura 3	Diagrama criterios de inclusión y exclusión	17
Figura 4	Ejemplo del árbol y mazorca de cacao	22
Figura 5	Banco nacional de germoplasma de cacao - AGROSAVIA	24
Figura 6	Ejemplo de la mazorca de cacao con Monilia – AGROSAVIA.....	25
Figura 7	Ejemplo de la mazorca de cacao con Fitóftora – AGROSAVIA	25
Figura 8	Ejemplo de la red neuronal convolucional	27
Figura 9	Evaluación de perdidas familia Crystal	32
Figura 10	Proceso de desarrollo	36
Figura 11	Arquitectura de YOLOv4	39
Figura 12	Diagrama de casos de uso para la identificación de mazorcas de cacao enfermas.....	41
Figura 13	Diagrama de actividades para la identificación de mazorcas de cacao enfermas.....	42
Figura 14	Diagrama de secuencia para la identificación de mazorcas de cacao enfermas.....	43
Figura 15	Diagrama de clases para la identificación de mazorcas de cacao enfermas.....	44
Figura 16	Captura repositorio Labellmg.....	46
Figura 17	Captura configuración y ejecución Labellmg	46
Figura 18	Etiquetando mazorca con herramienta Labellmg	47
Figura 19	Cocoa Diseases (YOLOv4) Dataset Kaggle	48
Figura 20	Detección de objetos modelo entrenado con el MSCOCO dataset	49
Figura 21	Detección de objetos comando de descarga pesos pre-entrenados	51
Figura 22	Detección de objetos entrenando modelo Enfermedades del cacao.	52
Figura 23	Backup de modelos o pesos generados en el entrenamiento.	52
Figura 24	Resultado mAP del entrenamiento	53
Figura 25	Ejemplo de detección con Tflite y el conjunto de datos MSCOCO	54
Figura 26	Comando para guardar YOLOv4 – Tflite	55
Figura 27	Comando para convertir YOLOv4 – Tflite	55
Figura 28	App Enfermedades Cacao.....	56
Figura 29	Prototipo de aplicación móvil funcional para detectar mazorcas	57
Figura 30	Modelo de la red neuronal de enfermedades del cacao Tflite.	70

RESUMEN

Según fuentes de Fedecacao, para el 2018 se determinó que la producción total de cacao *Theobroma cacao L* en Colombia superó las 56 mil toneladas, la segunda más alta en toda la historia puesto que hubo una reducción de la producción del 6% en comparación al 2017, al pasar de 60 mil a 56 mil toneladas (Fedecacao, 2018). La reducción se dio debido a los flujos de floración, al incremento de la enfermedad llamada Monilia y a las inundaciones causadas por las precipitaciones de comienzos de año. Teniendo en cuenta que la inspección semanal es un procedimiento confiable para evaluar el grado de infección en los cultivos, el presente proyecto desarrolla un prototipo de aplicación móvil que permita identificar mediante técnicas de inteligencia artificial y análisis de imágenes las mazorcas de cacao enfermas. Mediante el levantamiento de datos esta investigación realiza la consolidación de un conjunto de datos que contiene las enfermedades más preocupantes como son la Fitóftora y la Monilia. Para mejorar el procedimiento de inspección se entrena una máquina de aprendizaje con YOLOv4 obteniendo un 60% de precisión en la detección de mazorcas de cacao. Los resultados prometen una herramienta móvil útil a los agricultores e investigadores agrícolas para la toma de decisiones, permitiendo evaluar y realizar con precisión el proceso de inspección de las mazorcas sin la necesidad de un experto capacitado en el manejo fitosanitario en los cultivos de cacao.

Palabras clave: *Theobroma cacao L.*, mazorcas de cacao, enfermedades del cacao, inteligencia artificial, análisis de imágenes y manejo Fitosanitario

ABSTRACT

According to sources from Fedecacao, by 2018 it was determined that the total production of *Theobroma cacao* L in Colombia exceeded 56 thousand tons, the second highest in history since there was a reduction in production of 6% compared to 2017, from 60 thousand to 56 thousand tons (Fedecacao, 2018). The reduction was due to the flowering, the increase of the disease called *Monilia* and the flooding caused by the rainfall at the beginning of the year. Considering that the weekly inspection is a reliable procedure to evaluate the degree of infection in the crops, the present project develops a prototype of mobile application that allows to identify by means of artificial intelligence techniques and image analysis the sick cocoa ears. By means of data collection, this research carries out the consolidation of a set of data that contains the most worrying diseases such as *Phytophthora* and *Monilia*. To improve the inspection procedure, a training machine is trained with YOLOv4, obtaining 60% accuracy in the detection of cocoa beans. The results promise a mobile tool useful to farmers and agricultural researchers for decision making, allowing to evaluate and perform accurately the inspection process of the ears without the need of an expert trained in phytosanitary management in cocoa crops.

Keywords: *Theobroma cacao* L., cocoa pods, cocoa diseases, artificial intelligence, image analysis and Phytosanitary management.

1. INTRODUCCIÓN

El cacao crece en regiones tropicales con un clima húmedo y una buena cantidad de lluvia. Santander, Colombia pertenece a una de las regiones donde las condiciones climáticas y las características del suelo son apropiadas para el cultivo y producción de cacao. De hecho, el departamento de Santander, Colombia es actualmente el mayor productor de cacao a nivel nacional con una producción anual aproximada de 25.158 toneladas la cual representa una participación del 42,1% para el 2019, siendo durante la última década el primer productor de cacao a nivel nacional (Fedecacao, 2019). Sin embargo, las plagas y enfermedades como la Monilia y la Fitóftora afectan considerablemente las producciones de cacao, llegando a perder hasta el 40% de la producción anual.

Debido a la afectación de las enfermedades en la producción de cacao, la mayoría de los agricultores y técnicos agrícolas realizan manualmente la inspección visual semanal (RE-SE) de los cultivos de cacao con el fin de identificar las enfermedades a tiempo con el propósito de intervenir el cultivo para evitar que la enfermedad se propague o genere pérdidas en la producción. En este sentido, realizar las respectivas inspecciones semanales termina siendo una tarea repetitiva que consume tiempo y energía y aumenta los costos de producción, de igual forma esta labor cumple un papel esencial en la detección de enfermedades de manera oportuna en mazorcas de cacao para poder contribuir con un volumen de producción competitivo en el mercado internacional para Colombia y en este sentido, en el presente proyecto se plantean una serie de objetivos los cuales se centran en realizar un diseño de un prototipo de aplicación móvil y respectivamente consolidar un banco de imágenes que permitan construir una aplicación móvil para la detección de enfermedades en la mazorca de cacao como la Monilia y/o Fitóftora que funcione de forma offline.

Por lo tanto en este proyecto, se integra en la construcción del prototipo de aplicación móvil el modelo de aprendizaje profundo YOLOv4 (Bochkovskiy et al., 2020a), mejorado para detectar enfermedades como la monilia y/o fitóftora en superficies de mazorcas de cacao.

2. PLANTEAMIENTO DEL PROBLEMA

1.1. PLANTEAMIENTO DEL PROBLEMA

Para el departamento de Santander, Colombia, el cultivo de cacao representa el 41,45% de la participación en la producción nacional y tiene un impacto considerable en el sustento de las familias de la región. “La producción actual anual en Santander es de unas 23.574 toneladas.” (Fedecacao, 2019). No obstante, al comparar la producción anual de 2017 que alcanzó un récord con 60.535 toneladas a nivel nacional con la de 2018 que tuvo un total de 56.867 toneladas; se encontró una disminución significativa (Fedecacao, 2018). La reducción se debe a un aumento de mazorcas enfermas en los cultivos de cacao, “El comportamiento del clima sigue siendo una variable que incide en la producción siendo el caso del primer semestre del año 2018, el cual se caracterizó por intensas lluvias que afectaron los cultivos al aumentar la presencia de Moniliasis, daños en la floración y en el cuajamiento de frutos” (Fedecacao, 2019). Enfermedades como la Moniliasis no son las únicas que crean daños en la floración, una bastante similar es la Fitóftora que junto a la Monilia puede reducir hasta un 50% de la producción (ICA, 2012). Ambas enfermedades generan una pudrición en las mazorcas de cacao. La monilia por un lado causada por *Moniliophthora roreri* (Cif y Par) Evans *et al.* (Aime & Phillips-Mora, 2005) patógeno hemi-biótrofo conocido como un ascomicete anamórfico, perteneciente a la familia Marasmiaceae (Aime & Phillips-Mora, 2005), cuando esta enfermedad se manifiesta tiene una estructura infectada por esporas de color blanco que se pueden expandir por toda la mazorca, estas esporas se liberan por corrientes de aire y vibraciones de las plantas durante la cosecha (Evans, 1981) haciendo que se reproduzca la enfermedad en otras mazorcas. Por otro lado la Fitóftora que es una enfermedad causada por especies del oomicete *Phytophthora*, cuenta con las mayores pérdidas en producción mundial de cacao causadas por *P. palmivora*, *P. megakarya* y *P. citrophthora* (McMahon P & Purwantara A, 2004). La estructura de infección se asemeja a manchas con un color marrón que también se pueden expandir por toda la mazorca dañando la pulpa. La dispersión de esta enfermedad y el desarrollo de la epidemia ocurre al salpicar agua lluvia sobre los esporangios, encontrados en el suelo o en plantas infectadas, y por insectos vectores (McMahon P & Purwantara A, 2004).

Con el fin de identificar las enfermedades a tiempo con el propósito de intervenir el cultivo, se encuentra imprescindible que las mazorcas de cacao tengan una revisión semanal manual (RE-SE) por parte del agricultor o encargado del cultivo pues se necesita estar al tanto de las condiciones en que se encuentran las plantas. Las mazorcas por su parte presentan cambios diarios debido al clima y el manejo fitosanitario que se le da al cultivo. También es importante resaltar que la producción

de los cultivos colombianos por hectárea es muy reducida. Los cultivos en Colombia no superan producción de entre 300 y 400 kilos por hectárea. Mientras que en otros países como México el promedio es de 499 kilos y en países como Tailandia e Indonesia los cultivos alcanzan una producción de 1090 y 813 kilos por hectárea respectivamente. (Espinosa-García et al., 2015)

Teniendo en cuenta lo anterior, se concluye que alcanzar un volumen de producción en función del área (toneladas / hectárea) competitivo, depende en gran medida de la identificación y el manejo correcto del cultivo de cacao, el presente proyecto propone construir un prototipo de aplicación que identifique en un estado temprano de maduración o crecimiento, las mazorcas de cacao que se encuentran enfermas con el fin de permitirle al agricultor tomar decisiones relacionadas con la intervención del proceso productivo del caca, con esto se espera como futuro impacto aumentar la productividad y conseguir mejores resultados el rendimiento agrícola.

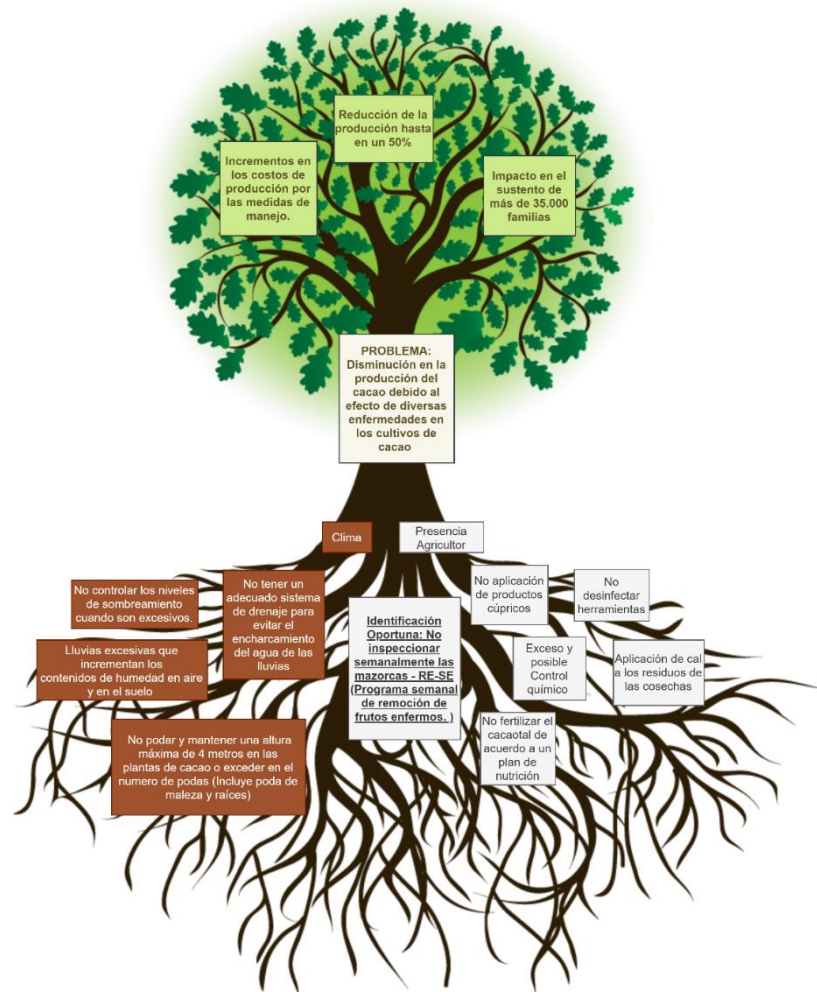


Figura 1 Árbol de problemas

Fuente: Adaptado de (Seekpng, 2020) por Los autores

1.2. JUSTIFICACIÓN

La identificación oportuna de las enfermedades en las mazorcas de cacao impacta positivamente en el rendimiento agrícola del país, pues evita posibles pérdidas en la producción de cacao, estas pérdidas se deben en aproximadamente un 50% los casos a la presencia de enfermedades como la Moniliasis y la Fitóftora (ICA, 2012). Hay que resaltar que la dispersión de las dos enfermedades Fitóftora y Monilia depende también de las características intrínsecas de susceptibilidad de la variedad de la planta de cacao donde se pueden encontrar algunos clones que logran tener más susceptibilidad en algunas enfermedades. Es el caso de los clones ICS95 y CCN51 pues se reportan como susceptible y altamente susceptible a *Phytophthora spp.* respectivamente, donde el uso de este último clon se incrementa cada vez más en el establecimiento de nuevas plantaciones a nivel nacional (Polanco et al., 2015).

En general, la detección de enfermedades de manera oportuna en mazorcas de cacao es necesaria por lo que se necesita de una revisión semanal (RE-SE) de las condiciones en que se encuentra el cultivo para trabajar y ayudar a lograr un volumen de producción competitivo en el mercado internacional para Colombia y en este sentido, las técnicas de visión artificial permiten la portabilidad y la automatización de procesos como la inspección semanal, en el caso agrícola. Por lo tanto, este proyecto se centra en el desarrollo de un prototipo de aplicación móvil que implemente técnicas de visión artificial existentes dentro de una máquina de aprendizaje entrenada con el algoritmo de código abierto YOLOv4, para poder reconocer visualmente y responder a tiempo a posibles enfermedades que generan una pudrición en las mazorcas de cacao como los son la Fitoftria y la Moniliasis.

Esta aplicación tiene el potencial de brindar a los agricultores e investigadores una herramienta confiable que facilite la identificación de mazorcas de cacao enfermas, con lo cual será posible tomar medidas fitosanitarias en los cultivos de manera oportuna, con lo que se esperaría aumentar la competitividad del sistema productivo del cacao. Esta herramienta no solo contará con un soporte para el manejo y el aumento y/o pérdidas en la producción sino, también funciona como apoyo a investigaciones sobre las variedades de clones que pueden resistir a varias cepas. Por ejemplo en plantaciones de cacao en Colombia, donde se han reportado clones como el ICS95, ya que resiste a varias cepas de *M. roleri* (Aime & Phillips-Mora, 2005; Jaimes & , Aranzazu Fabio, 2011) que es el hongo causante la enfermedad de Moniliasis y al clon CCN51 que resistente al mismo patógeno en tres regiones del país, incluyendo la Orinoquía colombiana (Arauca) (Jaimes & , Aranzazu Fabio,

2011), sin embargo estos dos clones como se mencionó al principio son susceptibles a Phytophthora.

3. OBJETIVOS

3.1. OBJETIVO GENERAL

Desarrollar un prototipo de aplicación móvil para la identificación de mazorcas de cacao enfermas con Monilia y/o Fitóftora, haciendo uso de visión por computadora y aprendizaje de máquina.

3.2. OBJETIVOS ESPECÍFICOS

- Diseñar un prototipo de aplicación móvil para la identificación de mazorcas de cacao enfermas con Monilia y/o Fitóftora.
- Consolidar un conjunto de datos que permitan calibrar una máquina de aprendizaje para la identificación de mazorcas de cacao enfermas con Monilia y/o Fitóftora, mediante la toma de imágenes.
- Determinar la presencia de mazorcas de cacao enfermas con Monilia y/o Fitóftora en cultivos sembrados de una región de Santander, mediante la implementación y calibración de una máquina de aprendizaje.
- Construir un prototipo funcional que integre un algoritmo de aprendizaje de máquina para la identificación de mazorcas de cacao enfermas con Monilia y/o Fitóftora.

3.3. RESULTADOS ESPERADOS

- Resultado esperado 1: Un diseño del prototipo de aplicación móvil para la identificación de mazorcas de cacao enfermas con Monilia y/o Fitóftora.
- Resultado esperado 2: Conjunto de datos consolidado el cual contenga los datos para entrenamiento y prueba de la máquina de aprendizaje.
- Resultado esperado 3: Máquina para la detección de mazorcas de cacao enfermas con Monilia y/o Fitóftora.
- Resultado esperado 4 Prototipo de aplicación funcional para la detección de mazorcas de cacao enfermas con Monilia y/o Fitóftora.

4. REVISIÓN DE LITERATURA

La revisión de la literatura presentada analiza y estudia las técnicas de aprendizaje de máquina, aprendizaje profundo, procesamiento de imágenes, visión por computadora, clasificación de imágenes y reconocimiento de imágenes. La búsqueda se realizó en la base de datos científica SCOPUS a partir de una ecuación en la que se busca principalmente artículos de proyectos que involucran enfermedades en las mazorcas de cacao como la Monilia y la Fitóftora. La revisión de la literatura está dividida en: 1) La ecuación de búsqueda, sus criterios de inclusión y exclusión, y los artículos seleccionados. 2) La conclusión de los artículos seleccionados.

4.1. Ecuación de búsqueda y artículos

Durante la construcción de la ecuación que se muestra en la Figura 2, se encontraron los siguientes criterios de inclusión: el cacao y sus variantes, el árbol de cacao conocido como Theobroma Cacao, las enfermedades y sus variantes en inglés, aprendizaje de máquina, procesamiento de imágenes, visión por computadora, clasificación de imágenes, reconocimiento de imágenes, detección de objetos, aprendizaje profundo, imágenes, aplicaciones, Android y herramientas como YOLO v2, Tensorflow y DarkFlow.

```
( TITLE-ABS-KEY ( "cacao" OR "Cocoa" OR "Theobroma Cacao" ) AND TITLE-ABS-KEY ( "diseases" OR "sickness" OR " issues" OR "illness" ) AND TITLE-ABS-KEY ( "machine learning" OR "image processing" OR "Computer vision" OR "Image Classification" OR "Image Recognition" OR "Object Detection" OR "Deep learning" OR "Image" OR "App" OR "Android" OR "YOLO v2" OR "TensorFlow" OR "DarkFlow" ) )
```

Figura 2 Ecuación de búsqueda

Fuente: Los autores

Con el fin de analizar los artículos estrechamente relacionados con el análisis de enfermedades en el cultivo de cacao, se tienen en consideración los siguientes criterios de exclusión que se muestran a continuación y en la Figura 3

- Filtro de los artículos que no están relacionados completamente con el cacao, su producción y enfermedades
- Filtro de los artículos repetidos

Una vez seleccionados los artículos, estos fueron revisados en orden cronológico para buscar las posibles técnicas de machine learning, lenguajes de programación y herramientas que son tendencia en el desarrollo de trabajos relacionados al análisis de enfermedades en el cultivo de cacao. Se seleccionaron artículos desde el año 2016 al año 2019.

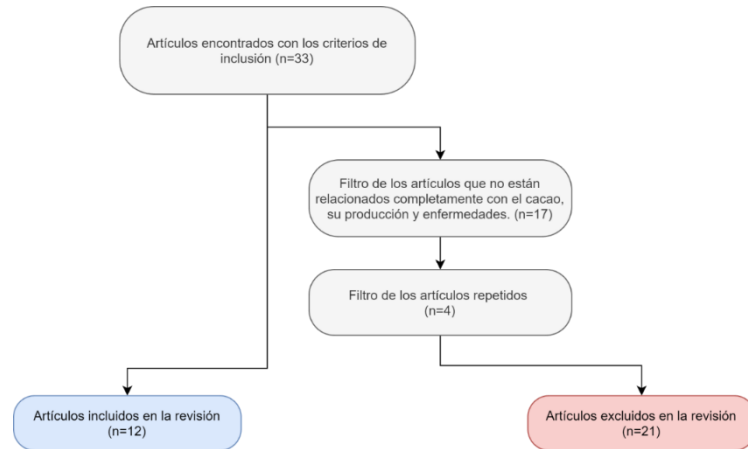


Figura 3 Diagrama criterios de inclusión y exclusión

Fuente: Los autores

4.2. Antecedentes

En 2016, Tan Daniel Stanley y otros desarrollan un trabajo titulado “*A framework for measuring infection level on cacao pods*” (Tan et al., 2016). Desarrollan un marco para la medición automática y objetiva del nivel de infección en las mazorcas de cacao, se basan principalmente en cuatro pasos los cuales son la recolección de imágenes y pre-procesado; agrupación de imágenes basada en colores; clasificación de grupos sanos o infectados y computación del nivel de infección. Utilizan algoritmos de agrupamiento (K-means) y clasificación (SVM) para identificar y segmentar las secciones infectadas en la mazorca de cacao en 35 imágenes recolectadas de las cuales 30 fueron para entrenamiento y 5 para test.

Posteriormente en 2018, Tan Daniel Stanley y otros desarrollan un trabajo titulado “*AuToDiDAC: Automated Tool for Disease Detection and Assessment for Cacao Black Pod Rot*” (Tan et al., 2018). Proporcionan a los agricultores en Filipinas una herramienta para monitorear e informar objetivamente el nivel de infección de la enfermedad al proporcionar una aplicación móvil “AuToDiDAC” que aísla la mazorca de cacao negro de su fondo y permite medir el nivel de infección BPR (causada por hongos *Phytophthora*) con mayor precisión. Para el entrenamiento y calibración de

la maquina se seleccionó al azar 50 imágenes de las 60 que recolectaron. El algoritmo utiliza un algoritmo de agrupamiento (K-means) que agrupa por objetos basándose en sus características y uno de clasificación (SVM9, al final el algoritmo determina por etiquetas si esta "infectado" o "no infectado".

Luego en 2018, Bayu Taruna Widjaja Putra y otros desarrollan un trabajo titulado "*Using information from images for plantation monitoring: A review of solutions for smallholders*" (Putra et al., 2018). En donde plantean un protocolo para el procesamiento de imágenes y un DSS (Un Sistema de Soporte a la Decisión) como solución para el uso de los pequeños productores en indonesia que tienen limitaciones y desean seguir monitoreando las actividades agrícolas como las plagas, enfermedades, nutrientes e información que tienen en las plantaciones mediante uso de cámaras digitales y sensores ópticos. Utiliza MANET que es una red de dispositivos conectados por conexión inalámbrica y que proporciona propiedades que dan movilidad.

Después en 2018, De Oliveira Juliana Rodrigueiro C.P y Romero Roseli Ap Francelin desarrollan un trabajo titulado "*Transfer Learning Based Model for Classification of Cocoa Pods*" (De Oliveira & Romero, 2018). Construyen una red neuronal para la clasificación de mazorcas de cacao (Maduro o Inmaduro) en San Carlos, Brasil. Hacen uso de aprendizaje profundo y aprendizaje de transferencia. La red de aprendizaje profundo utiliza la biblioteca código abierto Tensorflow y se basa en Inception-Resnet-Modelo v2 para clasificar las imágenes tomadas de una base de datos llamada ImageNet que contiene 1243 imágenes de mazorcas de cacao maduro e inmaduro. Durante el entrenamiento, se aplicó una tasa de abandono del 80%, que es una técnica común de sobreajuste y normalización para redes neuronales. El modelo de red neuronal construido presentó una precisión de 0,90.

También en 2018, Nembot Christian y otros desarrollan un trabajo titulado "*Modeling the temporal evolution of cocoa black pod rot disease caused by Phytophthora Megakarya*" (Nembot et al., 2018). Plantean un modelo epidemiológico en Camerún que incorpora una función de tasa de natalidad periódica para las mazorcas de cacao y analiza la evolución temporal que tiene la enfermedad de la mazorca de cacao negra. Evaluaron el impacto de la extracción de mazorcas negras. El manejo fitosanitario en la cosecha puede reducir la incidencia de la mazorca negra entre un 22% a 31% y entre un 35% a 65% por *P. palmivora*. Muestran con datos obtenidos que sí la extracción de mazorcas se hace más intensa y regular (cada cuatro días), se puede conseguir reducir aún más la incidencia

Después en 2018, Lomotey Richard K y otros desarrollan un trabajo titulado “*Mobile technology for smart agriculture: Deployment case for cocoa production*” (Lomotey et al., 2018). Realizan entrevistas donde identifican una necesidad de tecnología móvil en los agricultores del cacao y para ello desarrollan una aplicación móvil llamada CocoaGuard. La aplicación se desarrolló en Canadá y da información sobre control de plagas, fertilizantes, costo de producción y estimación de ganancias. También integra conexión con redes sociales y un foro de discusión. La aplicación da acceso continuo a funciones / servicios en donde no se verá obstaculizado nada debido a pérdida de conectividad o desconexiones esporádicas. Se realizó en Xamarin utilizando el lenguaje de programación C#.

Para el 2019, Jaimes Yeirme Y. y otros desarrollan un trabajo titulado “*Factors affecting the dynamics of frosty pod rot in the main cocoa areas of Santander State, Colombia*” (Jaimes et al., 2019). Por medio de árboles de regresión ampliados modelan la dinámica de las mazorcas de cacao enfermas en 4 lugares donde realizan pruebas. Se utilizó la distribución de Poisson para modelar el número de mazorcas de cacao enfermas por árbol para cada 14 días donde solo seleccionaron árboles con mazorcas de cacao enfermas mayores o iguales a 4 para relacionar la enfermedad con otras variables.

Posteriormente en el año 2019, Fuentes Sigfredo y otros desarrollan un trabajo titulado “*Spatial variability of aroma profiles of cocoa trees obtained through computer vision and machine learning modelling: A cover photography and high spatial remote sensing application*” (Fuentes et al., 2019). Proporcionan una herramienta para los productores en Nueva Zelanda, que contribuye con la adaptación de estrategias para el manejo de los cultivos de cacao. El modelo de aprendizaje automático utiliza datos de arquitectura extraídos de imágenes aéreas para obtener el dosel y los bloques en donde están los árboles de cacao. Los datos se obtienen mediante GPS para considerar la densidad de la plantación a través de un código que fue escrito en Matlab, la aplicación informática se llama VitiCanopy y se encarga de validar la información extraída de las imágenes para obtener los parámetros de plantación de los cultivos. Esta información se puede obtener en tiempo real mediante un dispositivo móvil.

Luego en 2019, Gamboa Andrea A y otros desarrollan un trabajo titulado “*Predictive model for cocoa yield in Santander Using Supervised Machine Learning*” (Gamboa et al., 2019). Presentan una metodología para implementar modelos con herramientas de aprendizaje automático para la predicción del rendimiento de los cultivos de cacao en Santander, Colombia. Hacen uso del lenguaje de programación Python y su función “GridSearch”, en donde desarrollaron modelos (GLM) que es un Modelo lineal generalizado Y Máquinas de vectores de soporte

(SVM). Para ajustar la maquina e implementarla se dividió el conjunto de datos por la regla del 80/20. El 80% de los datos para entrenar los algoritmos y el otro 20% restante para la prueba de estos. Las variables que se analizaron fueron Diámetro del tronco, Fósforo (P), Magnesio (Mg), % Arena, % Hum/Grav, Radiación, Temperatura, Humedad y Lluvias acumuladas que son las que más afectan en el rendimiento de los cultivos de cacao en La Suiza en Rio Negro, Santander.

Después en el año 2019, Lagos Ortiz Katty y otros desarrollan un trabajo titulado “*Decision Support System for the Control and Monitoring of Crops*” (Lagos-Ortiz et al., 2019). Presentan un desarrollo de un DSS (Un sistema de soporte de decisiones) para apoyar a los agricultores e investigadores ecuatorianos en la identificación de cultivos que tengan enfermedades, plagas y malezas. Se utilizaron una serie de experimentos para validar la efectividad de los resultados obtenidos mediante imágenes de cultivos de arroz, café y cacao afectados. La efectividad se mide utilizando métricas de precisión, recuperación y F-metric, que está asociado con las condiciones respecto a una enfermedad determinada.

Unos meses después en 2019, Basri, Harli y otros desarrollan un trabajo titulado “*Image Processing System for Early Detection of Cocoa Fruit Pest Attack*” (Basri et al., 2019). En donde desarrollan un sistema de detección temprana basado en procesamiento de imágenes en síntomas de ataques de plagas en mazorcas de cacao, para ello utilizan muestras de imágenes de la cuales están divididas en 30 imágenes de cacao sano y 50 imágenes de mazorcas de cacao podridas por plagas, de cada conjunto de imágenes deciden utilizar 10 imágenes para test. Los autores desarrollan este sistema basados en una metodología de desarrollo en cascada.

Para finalizar en 2019, Ariandi Vicky y otros desarrollan un trabajo titulado “*Expert system for disease diagnosis in cocoa plant using android-based forward chaining method*” (Ariandi et al., 2019). En el desarrollan una aplicación móvil programada en Android Studio con el fin de identificar cultivos de cacao afectados por enfermedades o plagas, para ello hacen uso de los sistemas expertos con el método de encadenamiento hacia adelante.

4.3. Conclusión

En síntesis, los artículos encontrados se centran en la producción de los cultivos, resaltando aspectos como la capacidad, las enfermedades en los frutos y las plagas que poseen las plantas. Referente a las aplicaciones, están optan por modo offline, debido a la escasa conectividad que prevalece en las zonas alejadas de los países como lo es el campo agrícola. Además, se evidencia que existen tendencias en el ámbito de procesamiento de imágenes para poder entrenar la maquina y que se utiliza la regla del 80/20 en la consolidación de los datos adquiridos, el 80% se utiliza para calibrar la maquina y el 20% restante para hacer pruebas. Por otro lado,

también se puede evidenciar que en la mayoría de los proyectos optan por realizar el tratamiento de imágenes por medio de tecnologías alternativas. Estas son bibliotecas para lenguajes como JAVA, Python, o MATLAB. Esta última, MATLAB, posee una caja de herramientas poderosa pero no es una herramienta de código abierto y es importante que lo sea ya que pueden ser fácilmente implementadas en aplicaciones móviles.

Teniendo en cuenta lo anterior, se propone una aplicación para dispositivos móviles que no necesiten tener conectividad y/o acceso a internet en respuesta a que la mayoría de las fincas cacaoteras en Santander no poseen conectividad en áreas de cultivo. También se propone realizar la selección de un conjunto de datos estandarizados para poder trabajar con la arquitectura de red que ya trae preestablecida YOLO, separando por la regla del 80/20 el conjunto de datos para entrenar y calibrar la máquina. Por último, se él desarrolla de un prototipo software de aplicación móvil que le permita al cacaotero identificar las mazorcas de cacao enfermas por Monilia y Fitóftora, haciendo uso de visión por computadora y aprendizaje de máquina.

5. MARCO TEÓRICO

5.1. Agricultura y Cacao

5.1.1. Cacao – *Theobroma Cacao L*

El nombre científico del árbol que da el cacao es *Theobroma Cacao L*, este nombre científico proviene del griego y significa alimento de los dioses. La planta de hoja perenne de la familia *Malvaceae* tiene su origen hace miles de años y se remonta a la familia mixe-zoque que hablaban los olmecas. Los olmecas quienes fueron los primeros en cultivar la planta.



Figura 4 Ejemplo del árbol y mazorca de cacao

Fuente: Los Autores

El árbol de cacao alcanza una altura de hasta ocho metros y la mayoría de las variedades cultivadas son injertadas (híbridas o clones). La propagación vegetativa artificial ha hecho que al realizar el injerto el tejido de la planta se asiente y es por eso que rara vez sobrepasan los cuatro metros de altura. La raíz del árbol de cacao se hunde verticalmente en la tierra y llega a alcanzar hasta 1,5 metros de profundidad. Horizontalmente las raíces del árbol se pueden expandir hasta unos tres metros de radio. En la ramificación del árbol de cacao, se forma de una manera natural y comienzan a partir del metro, y metro y medio de altura. Se forman entre 3 a 5 ramas que son llamadas mesa, horqueta o verticilo. Las ramas pueden ser de dos tipos, unas de son de crecimiento vertical y otras son de crecimiento orto trópico,

que hacen parte del tallo y/o chupones, es decir ramas mal dirigidas. Otras ramas son de crecimiento oblicuo porque crecen hacia afuera y son denominadas ramas de crecimiento plagio trópico. Las hojas del cacao dependen de las variedades del árbol de cacao, pero casi siempre poseen un color verde bastante variable. Pueden ser color café claro pasando por morado, rojizo y hasta verde pálido. El árbol de cacao produce flores, estas son pequeñas y su color va del blanco al rosado y se forman en la madera más antigua del tronco y de las ramas más viejas del árbol.

La parte más importante son los frutos del cacao, es una baya denominada mazorca y son de aproximadamente de 20 a 40 centímetros de largo y de 10 a 15 centímetros de diámetro. Por lo general, el cacao es verde o rojo cuando aún no ha madurado lo suficiente, pero al final, cuando ya está madura casi siempre se torna de un color amarillo o naranja dependiendo de la variedad, Al interior del fruto se pueden encontrar 25 y 40 granos o semillas que al fermentar y secar se convierten en la materia prima para la elaboración del chocolate y sus derivados.

5.1.2. Cacao en Santander

Santander se encuentra encabezando la lista de los productores de cacao dentro de los principales departamentos de Colombia, cuenta con un 41,45% (2018) de la producción nacional, seguido por Arauca y Antioquia. Obtiene los mayores volúmenes de grano en periodos de noviembre-diciembre y mayo-junio. “Santander es el principal productor de cacao en Colombia con una participación del 38% del mercado, cuenta con 25.700 productores de cacao con cédula cacaotera, los cuales facturan aproximadamente 200 mil millones de pesos al año” (Santander, 2019).

En Santander se encuentra la Corporación Colombiana de Investigación Agropecuaria AGROSAVIA (anteriormente CORPOICA) desde hace varios años trabaja en nuevos diseños bajo este sistema y ha incorporado maderas finas tropicales abarco (*Cariniana pyriformis*), teca (*Tectona grandis*) o caucho (*Hevea brasilienses*) como sombrío. Así mismo, la Corporación hizo el lanzamiento de dos nuevas variedades de cacao, la TCS 13 (Theobroma Corpoica La Suiza 13) y TCS 19 (Theobroma Corpoica La Suiza 19), adaptadas a los departamentos de Santander y el Magdalena medio, con un buen nivel de fecundación y con óptimos niveles de crecimiento.



Figura 5 Banco nacional de germoplasma de cacao - AGROSAVIA

Fuente: Los autores

5.1.3. Enfermedades del Cacao

Las principales enfermedades que afectan la producción de cacao en Colombia y en específico en el departamento de Santander son las siguientes:

5.1.3.1. La moniliasis del Cacao

Esta enfermedad puede ser conocida de muchas maneras dependiendo de las regiones donde se cultive el cacao, dentro de las más conocidas están: pudrición acuosa, pasmo, nieva, hielo, ceniza y pringue. “La moniliasis del cacao es producida por el hongo *Moniliophthora roreri*, que se alimenta de los frutos del cacao y, por tanto, los daña” (ICA, 2012). La enfermedad muestra síntomas diversos estados a la hora de ser atacado. En todas las regiones donde se cultiva cacao en Colombia, causa daños y pérdidas considerables en la producción. Este hongo produce millones de esporas de color blanco, que se multiplican rápidamente cuando el cacao no está siendo bien manejado y el ambiente es favorable para la reproducción del *Moniliophthora Roreri*. “La moniliasis es causada por el hongo *Moniliophthora roreri* que infecta exclusivamente frutos jóvenes de cacao y de otras especies afines y destruye las semillas que son el producto comercial.” (Yanet et al., 2013)



Figura 6 Ejemplo de la mazorca de cacao con Monilia – AGROSAVIA
Fuente: Los Autores

5.1.3.2. Mazorca negra o Fitóftora

“Es una enfermedad causada por el hongo *Phytophthora* sp. Ataca raíces, hojas, tallos, frutos y ramas del cacao. En cacao se han reportado siete especies patógenas: *P. palmivora*, *P. megakarya*, *P. capsici*, *P. citrophthora*, *P. nicotianae* var. *Parasitica*, *P. megasperma*, y *P. arecae*. El género *Phytophthora* se encuentra distribuido en todo el mundo; predominan diferentes especies de acuerdo con la zona geográfica y el hospedero.”(ICA, 2012)



Figura 7 Ejemplo de la mazorca de cacao con Fitóftora – AGROSAVIA

Fuente: Los autores

5.2. Procesamiento de Imágenes

5.2.1. Imágenes

5.2.1.1. Imagen Digital

Es la representación óptica de uno o más objetos iluminados por una o más fuentes de radiación. se puede definir mediante una matriz bidimensional dispuesta específicamente en filas y columnas (Alegre et al., 2016). La imagen digital se compone de un número finito de elementos, cada uno de los cuales tiene un valor particular en una ubicación particular estos elementos se denominan elementos de imagen, y píxeles

5.2.2. Herramientas

5.2.2.1. Tensorflow

Es una interfaz para expresar algoritmos de aprendizaje automático y una implementación para ejecutar dichos algoritmos. Un cálculo expresado con Tensorflow se puede ejecutar con poco o ningún cambio en una amplia variedad de sistemas heterogéneos, desde dispositivos móviles como teléfonos y tabletas hasta sistemas distribuidos a gran escala de cientos de máquinas y miles de dispositivos computacionales como tarjetas de procesamiento gráfico (*Actualidad/tensorflow-dev-summit-2020--dias-11-y-12-de-marzo-en-en-sunnyvale-california*, n.d.). El sistema es flexible y se puede utilizar para expresar una amplia variedad de algoritmos, incluidos los algoritmos de entrenamiento e inferencia para modelos de redes neuronales profundas, y se ha utilizado para realizar investigaciones y desplegar sistemas de aprendizaje automático en producción en más de una docena de áreas de ciencias de la computación y otros campos, incluyendo reconocimiento de voz, visión por computadora, robótica, recuperación de información, procesamiento de lenguaje natural, extracción de información geográfica y descubrimiento computacional de drogas (Martín Abadi et al., 2015).

5.2.2.2. CUDA

Es una plataforma de computación paralela y modelo de programación inventados por la NVIDIA. Esta permite aumentos significativos de rendimiento computacional al aprovechar la potencia de la unidad de procesamiento gráfico (GPU). La plataforma CUDA es poderosa por habilitar el acceso directo a los elementos computacionales paralelos y al conjunto de instrucciones virtual de la CPU (C, 2010) (Dias Rafa, 2016).

5.2.2.3. YOLO

El algoritmo (*You Only Look Once*, YOLO), es un sistema de código abierto del estado del arte para detección de objetos en tiempo real, el cual hace uso de una única red neuronal convolucional para detectar objetos en imágenes (Redmon & Farhadi, 2018; Valencia Hugo, 2019).

5.2.2.4. Python

Es un lenguaje de programación interpretado cuya filosofía hace hincapié en la legibilidad de su código. Es un lenguaje de alto nivel ya que contiene implícitas algunas estructuras de datos como listas, diccionarios, conjuntos y tuplas, que permiten realizar algunas tareas complejas en pocas líneas de código y de manera legible (Holguín et al., n.d.).

5.2.2.5. LabelImg

Es una herramienta de anotación gráfica de imágenes y cuadros de límite de objetos de etiquetas en imágenes el cual está escrito en Python y usa Qt para su interfaz gráfica, las anotaciones se guardan como archivos XML en formato pascal VOC, esta herramienta es compatible con el formato que utiliza YOLO (Tzutalin, 2015).

5.2.3. Red Neuronal Convolucional

Una red neuronal convolucional (CNN) es un tipo de red neuronal artificial utilizada en el reconocimiento y procesamiento de imágenes que está específicamente diseñada para procesar datos de píxeles (Bagnato, 2018). Para esto, la CNN emplea una operación matemática llamada convolución en al menos una de sus capas. Los tipos de capa de red neuronal convolucional incluyen principalmente tres tipos, a saber, capa convolucional, capa de agrupación y capa totalmente conectada, como se muestra en la Figura 8. (Institute of Electrical and Electronics Engineers, 2017).

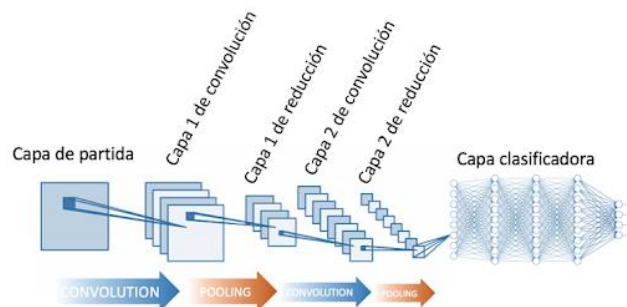


Figura 8 Ejemplo de la red neuronal convolucional
Fuente: (Diego Calvo, 2017)

5.2.3.1. Capa Convolutiva

La capa convolutiva es la parte central de la red neuronal convolutiva, que tiene conexiones locales y pesos de características compartidas. El objetivo de la capa convolutiva es aprender representaciones de características de las entradas.

5.2.3.2. Capa de Agrupamiento

La capa de agrupación tiene el efecto de la extracción de rasgos secundarios, puede reducir las dimensiones de los mapas de rasgos y aumentar la robustez de la extracción de rasgos. Normalmente se coloca entre dos capas convolutivas.

5.2.3.3. Capa totalmente Conectada

En general, el clasificador de la red neuronal convolutiva es una o más capas completamente conectadas. Toman todas las neuronas de la capa anterior y las conectan a cada una de las neuronas de la capa actual, la última capa totalmente conectada es seguida por una capa de salida.

5.2.4. Segmentación de imágenes

La segmentación de la imagen puede describirse generalmente como la subdivisión de una imagen en varias regiones con características de imagen que tenga cierta correlación. Como objetivo la segmentación propone simplificar la manera en cómo se representa una imagen para que su posterior análisis sea más fácil. La segmentación de la imagen es un paso clave en muchas aplicaciones de reconocimiento de patrones, visión por computador y la comprensión de la imagen para permitir una mayor explotación del contenido de la imagen de manera eficiente (Frucci & di Baja, 2008).

5.2.4.1. Visión por computador

Es la ciencia y la tecnología que permite programar un computador para simular los procesos fisiológicos detrás de la percepción visual en los humanos, y para clonar el comportamiento humano de desempeño en la inspección de color, contenido, forma y textura (Du & Cheng, 2014). Este campo interdisciplinario simula y automatiza estos elementos de los sistemas de visión humana utilizando sensores, computadoras y algoritmos de aprendizaje automático. La visión por computadora es la teoría que se encuentra implícita en la capacidad de los sistemas de inteligencia artificial para ver y comprender su entorno, permite adquirir, procesar, analizar y comprender las imágenes con el fin de producir información numérica o simbólica utilizada para la toma de decisiones tanto en tiempo real y en posteriores análisis (Li, 2008).

5.3. Inteligencia Artificial

Es la capacidad de las máquinas de demostrar inteligencia o imitar capacidades cognitivas propias de seres inteligentes. Estas capacidades incluyen razonar, representar conocimiento, aprender en base a evidencias, planear en base a objetivos, tener curiosidad, y entender el lenguaje natural. La implementación de inteligencia artificial se basa en métodos de optimización matemática, inferencia estadística, y abstracción computacional. Actualmente la Inteligencia Artificial (IA) recibe influencia de diversos campos incluyendo lingüística, biología, psicología, economía, y muchos otros. (Omar U. Florez, PhD)

5.3.1. Aprendizaje automático

El aprendizaje automático es una aplicación de inteligencia artificial (IA) que proporciona a los sistemas la capacidad de aprender y mejorar automáticamente a partir de la experiencia sin ser programado explícitamente. El aprendizaje automático se centra en el desarrollo de programas informáticos que pueden acceder a los datos y utilizarlos para aprender por sí mismos (Experts System Team, 2017).

El proceso de aprendizaje comienza con observaciones o datos, como ejemplos, experiencia directa o instrucción, para buscar patrones en los datos y tomar mejores decisiones en el futuro en función de los ejemplos que proporcionamos (El Naqa & Murphy, 2015). El objetivo principal es permitir que las computadoras aprendan automáticamente sin intervención o asistencia humana y ajustar las acciones en consecuencia

5.3.1.1. Aprendizaje Supervisado

Es un modelo de aprendizaje creado para hacer predicciones, dada una instancia de entrada imprevista. Un algoritmo de aprendizaje supervisado toma un conjunto conocido de conjunto de datos de entrada y sus respuestas conocidas a los datos salida para aprender el modelo de regresión / clasificación (Shobha & Rangaswamy, 2018).

5.3.1.2. Aprendizaje no supervisado

Es una técnica de aprendizaje automático donde no hay necesidad de supervisar el modelo, en gran parte por que este tipo de técnica permite que el modelo este en la capacidad de descubrir información. Esta técnica encuentra y analiza patrones en los datos etiquetados o sin etiquetar. Tales patrones pueden incluir las medidas de similitud, densidad, asociación, o jerarquía entre las observaciones (Florez, 2018).

5.3.2. Aprendizaje Profundo

Es una técnica de aprendizaje automático que enseña a los ordenadores a hacer lo que resulta natural para las personas: aprender mediante ejemplos. Es una función de inteligencia artificial que imita el funcionamiento del cerebro humano al procesar datos y crear patrones para su uso en la toma de decisiones. El aprendizaje profundo es un subconjunto del aprendizaje automático en inteligencia artificial que atiende redes capaces de aprender sin supervisión a partir de datos no estructurados o sin etiquetar. También conocido como aprendizaje neuronal profundo o red neuronal profunda (Chang, n.d.).

6. DISEÑO METODOLÓGICO

El diseño metodológico se encuentra dividido en tres frentes de trabajo importantes para el cumplimiento de los objetivos planteados:

6.1. Levantamiento del conjunto de datos

Para consolidar y estandarizar la base de datos que permite seguir con las actividades planteadas de calibración, entrenamiento y desarrollo, se realizará el levantamiento del conjunto de datos en el que se ejecuta una recolección de imágenes por etapas. Mediante un convenio con el SENA (Servicio Nacional de Aprendizaje) y se visita el centro de atención al sector agropecuario del Playón, ubicado en la subselección Aguas Calientes, El playón, Santander. Una vez en el lugar se procede a capturar las imágenes de manera estándar. Los pasos para la estandarización son los siguientes:

1. Preparar cámara con las siguientes características fijas dependiendo de la cámara: Punto Focal, Tiempo de exposición, Velocidad ISO, Compensación de la exposición, Distancia focal, Apertura máxima, Modo de medición: Foco, Modo de flash: sin flash, y longitud focal.
2. Capturar las imágenes bajo la asesoría del experto.
3. Organizar las imágenes: Darles orientación a las imágenes procurando que las mazorcas queden de manera vertical.
4. Etiquetar y codificar las imágenes de mazorcas de cacao enfermas y no enfermas haciendo uso de la herramienta Labellmg.
5. Consolidar el conjunto de imágenes de mazorcas de cacao enfermas y no enfermas, para posteriormente darle uso en el entrenamiento de la máquina de aprendizaje automático.

6.2. Diseño y desarrollo del prototipo funcional

La metodología de desarrollo de este proyecto está definida por la metodología ágil iterativa Crystal. “Esta maneja iteraciones cortas con retroalimentación frecuente por parte de los usuarios/clientes, minimizando de esta forma la necesidad de productos intermedios” (Calderón & Valverde, Rebaza, 2007, p. 23). La metodología Crystal es una familia de metodologías que dispone de algunas variantes, estas poseen un código de color que marca la complejidad de una metodología. Cuanto más oscuro es un color más pesado y crítico es el método. En la Figura 9 se muestra una evaluación de las pérdidas que puede ocasionar la falla de un sistema y el método requerido. Los parámetros son: Comodidad (C), Dinero Discrecional (D),

Dinero Esencial (E) y Vidas (L). Los números del cuadro indican el número de personas afectadas en un proyecto. (Calderón & Valverde, Rebaza, 2007, p. 24)

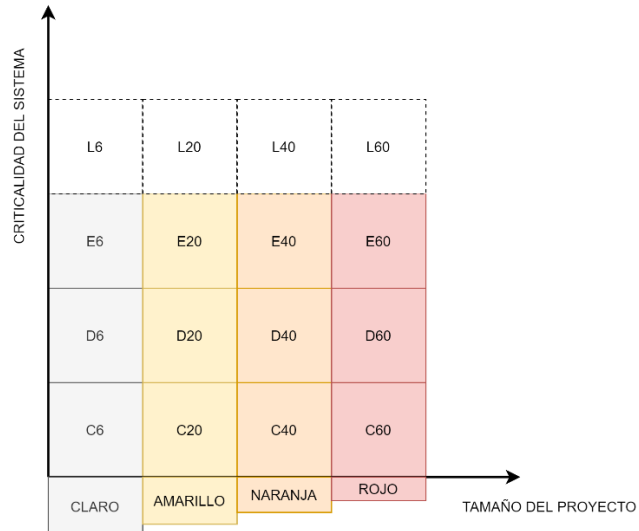


Figura 9 Evaluación de pérdidas familia Crystal

Fuente: Adaptado de título (Calderón & Valverde, Rebaza, 2007) Los autores

La variante de metodología a usar para este proyecto es Crystal Clear (“Claro como cristal”). La metodología está diseñada para equipos de ocho o menos integrantes, por lo que es justa para los cuatro integrantes que conforman el equipo de desarrollo de este proyecto. Crystal Clear cuenta con unos valores, técnicas y procesos que son esenciales para cumplir con la metodología de desarrollo. (Duarte & C, 2008)

Los siete valores o propiedades de la metodología Crystal Clear son:

- Entrega frecuente: “Consiste en entregar software a los clientes con frecuencia, no solamente en compilar el código.” (Calderón & Valverde, Rebaza, 2007, p. 25). Los tiempos de entrega dependen de lo establecido en el proyecto, estas pueden ser diarias, semanales, mensuales o lo necesario.
- Comunicación Osmótica: “Todos juntos en el mismo cuarto. Una variante especial es disponer en la sala de un diseñador senior; eso se llama Experto al Alcance de la Oreja. Una reunión separada para que los concurrentes se concentren mejor es descripta como El Cono del Silencio” (Calderón & Valverde, Rebaza, 2007, p. 25).
- Mejora reflexiva: Esta propiedad trata de darse un pequeño tiempo, el equipo debe tomar unas pocas horas a la semana para reflexionar y discutir aspectos.

- Seguridad Personal: “Hablar cuando algo molesta: decirle amigablemente al mánager que la agenda no es realista, o a un colega que su código necesita mejorarse, o que sería conveniente que se bañase más seguido.” (Calderón & Valverde, Rebaza, 2007, p. 25). Este valor es muy importante para poder lograr con las actividades y el cronograma, aquí se genera la confianza con los miembros del equipo.
- Foco: “Saber lo que se está haciendo y tener la tranquilidad y el tiempo para hacerlo.” (Calderón & Valverde, Rebaza, 2007, p. 25). Primero debe existir comunicación y segundo se debe discutir la dirección y lo más importante a ejecutar.
- Fácil acceso a usuarios expertos: Es importante tener comunicación con los que están más capacitados para dar soluciones, por lo que es bueno hacer un encuentro telefónico semanal. “Otra variante es que los programadores se entrenen para ser usuarios durante un tiempo. El equipo de desarrollo, de todas maneras, incluye un Experto en Negocios.” (Calderón & Valverde, Rebaza, 2007, p. 26).
- Ambiente técnico con prueba automatizada, management de configuración e integración frecuente: “Microsoft estableció la idea de los builds cotidianos, y no es una mala práctica. Muchos equipos ágiles compilan e integran varias veces al día.” (Calderón & Valverde, Rebaza, 2007, p. 26).

Crystal Clear no necesita de estrategias, pero teniendo en cuenta que muchas metodologías ágiles utilizan estrategias es útil tener unas cuantas a mano para empezar. Unas de las estrategias más comunes son:

- Exploración de 360°: “Verificar o tomar una muestra del valor de negocios del proyecto, los requerimientos, el modelo de dominio, la tecnología, el plan del proyecto y el proceso.” (Duarte & C, 2008, p. 7).
- Victoria temprana: “Es mejor buscar pequeños triunfos iniciales que aspirar a una gran victoria tardía.” (Duarte & C, 2008, p. 7).
- Esqueleto ambulante: “Es una transacción que debe ser simple pero completa.” (Duarte & C, 2008, p. 7).
- Re-arquitectura incremental: “Se ha demostrado que no es conveniente interrumpir el desarrollo para corregir la arquitectura. Más bien la arquitectura debe evolucionar en etapas, manteniendo el sistema en ejecución mientras ella se modifica.” (Duarte & C, 2008, p. 7).
- Radiadores de información: “Es una lámina pegada en algún lugar que el equipo pueda observar mientras trabaja o camina. Tiene que ser comprensible para el

observador casual, entendida de un vistazo y renovada periódicamente para que valga la pena visitarla.” (Duarte & C, 2008, p. 7).

Las técnicas de Crystal Clear son:

- Entrevistas de proyectos: “Se suele entrevistar a más de un responsable para tener visiones más ricas”. (Duarte & C, 2008, p. 7).
- Talleres de reflexión: Se parece mucho al valor o propiedad de mejora reflexiva, el propósito es unirse en la semana y discutir de los acontecimientos.
- Planeamiento Blitz: Esta técnica suele representarse como un juego, adaptado por el juego del planeamiento de XP que es una de las metodologías de desarrollo. Trata de unas tarjetas indexadas en una mesa que poseen una historia de usuario o función visible. Por ejemplo, “Los programadores escriben en cada tarjeta el tiempo estimado para desarrollar cada función. El patrocinador del usuario escribe la secuencia de prioridades, teniendo en cuenta los tiempos referidos y el valor de negocio de cada función. Las tarjetas se agrupan en períodos de tres semanas llamados iteraciones que se agrupan en entregas, usualmente no más largas de tres meses.” (Duarte & C, 2008, p. 7).
- Estimación Delphi con estimaciones de pericia: “En el proceso Delphi se reúnen los expertos responsables y proceden como en un remate para proponer el tamaño del sistema, su tiempo de ejecución, la fecha de las entregas según dependencias técnicas y de negocios y para equilibrar las entregas en paquetes de igual tamaño.” (Duarte & C, 2008, p. 7 y 8)
- Encuentros diarios de pie: Es enfocarse en identificar problemas, de cinco a diez minutos como máximo diariamente.
- Miniatura de procesos: “Una forma de presentar Crystal Clear puede consumir entre 90 minutos y un día. La idea es que la gente pueda “degustar” la nueva metodología.” (Duarte & C, 2008, p. 8)
- Gráficos de quemado: “Su nombre viene de los gráficos de quemado de calorías de los regímenes dietéticos; se usan también en Scrum.” (Duarte & C, 2008, p. 8). Esta técnica trata de descubrir mediante una graficación las demoras y los problemas que se dan en el proceso. La idea es evitar que se descubran problemas en un futuro; cuando este más avanzado el proyecto.
- Programación lado a lado: Establece que cada programador se enfoca en su trabajo asignado, pero estando atento al desarrollo de los otros programadores. “Esta es una ampliación de la Comunicación Osmótica al contexto de la programación” (Duarte & C, 2008, p. 8)

La metodología Crystal Clear tiene ocho roles, a continuación, se explica cada uno de ellos, sus responsabilidades y características:

- Patrocinador: “Produce la Declaración de Misión con prioridades de compromiso (Tradeoff). Consigue los recursos y define la totalidad del proyecto.” (Duarte & C, 2008, p. 8)
- Usuario Experto: “Junto con el Experto en Negocios produce la Lista de Actores-Objetivos y el Archivo de Casos de Uso y Requerimientos. Debe familiarizarse con el uso del sistema, sugerir atajos de teclado, modos de operación, información a visualizar simultáneamente, navegación.” (Duarte & C, 2008, p. 8)
- Diseñador Principal: “Produce la Descripción Arquitectónica. Se supone que debe ser al menos un profesional de Nivel 3. (En Metodologías Ágiles se definen tres niveles de experiencia: Nivel 1 es capaz de “seguir los procedimientos”. Nivel 2 es capaz de “apartarse de los procedimientos específicos” y encontrar otros distintos. Nivel 3 es capaz de manejar con fluidez, mezclar e inventar procedimientos). El Diseñador Principal tiene roles de coordinador, arquitecto, mentor y programador más experto.” (Duarte & C, 2008, p. 8)
- Diseñador-Programador: “Produce, junto con el diseñador principal, los borradores de pantallas, el modelo común de dominio, las notas y diagramas de diseño, el código fuente, el código de migración, las pruebas y el sistema empaquetado. un programa en CC es “diseño y programa” ; sus programadores son diseñadores-programadores. En CC un diseñador que no programe no tiene cabida.” (Duarte & C, 2008, p. 8)
- Experto en Negocios: “Junto con el usuario experto produce la lista de actores-objetivos y el archivo de casos de uso y requerimientos. Debe conocer las reglas y políticas del negocio.” (Duarte & C, 2008, p. 8)
- Coordinador: “Con la ayuda del equipo, produce el Mapa de Proyecto, el Plan de Entrega, el Estado del Proyecto, la Lista de Riesgos, el Plan y Estado de Iteración y la Agenda de Visualización.” (Duarte & C, 2008, p. 8)
- Verificador: “Produce el Reporte de Bugs. Puede ser un programador en tiempo parcial, o un equipo de varias personas.” (Duarte & C, 2008, p. 8)
- Escritor: “Produce el Manual de Usuario. El equipo como grupo es responsable de producir la estructura y convenciones del equipo y los resultados del taller de reflexión.” (Duarte & C, 2008, p. 8)

“En cuanto al proceso con Crystal Clear este se enfatiza con un conjunto de ciclos anidados. En los procesos se realizan siete ciclos: “(1) el proyecto, (2) el ciclo de entrega de una unidad, (3) la iteración (nótese que CC requiere múltiples entregas por proyecto, pero no muchas iteraciones por entrega), (4) la semana laboral, (5) el período de integración, de 30 minutos a tres días, (6) el día de trabajo, (7) el episodio

de desarrollo de una sección de código, de pocos minutos a pocas horas” (Calderón & Valverde, Rebaza, 2007, p. 27)

El proceso de este desarrollo está dividido en 8 entregas compuestas de 2 iteraciones como se muestra en la Figura 10. En cada una de las entregas se hace un proceso de trabajo similar donde se debe cumplir con la entrega de los requisitos planificados. En cada iteración el equipo evoluciona el producto y a partir de los resultados en cada entrega se obtiene retroalimentación. Un aspecto importante para guiar la metodología Crystal Clear, es priorizar los objetivos/requisitos para ofrecer valor y aportar con cada entrega a la idea.

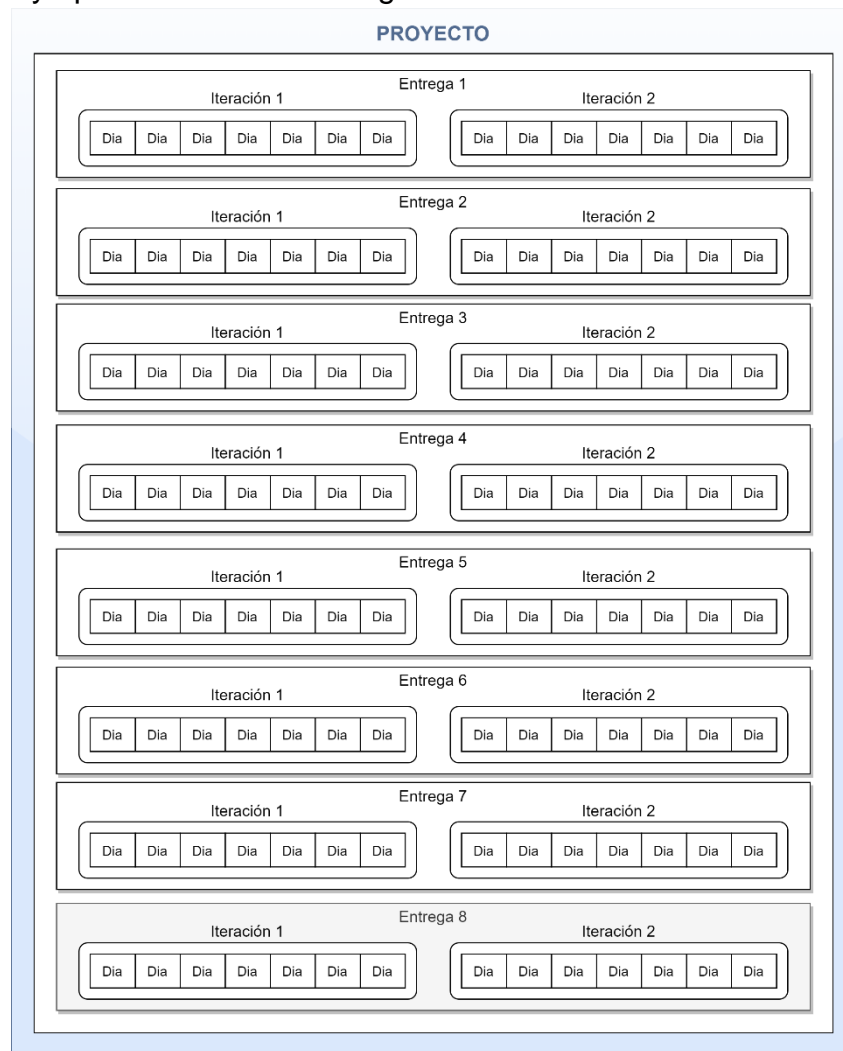


Figura 10 Proceso de desarrollo

Fuente: Adaptado de Ciclos anidados de Crystal Clear (Calderón & Valverde, Rebaza, 2007, p. 26) Los autores

La metodología Crystal Clear, define unas condiciones respecto al tiempo que se debe manejar en cada entrega. Cada una de estas entregas está compuesta de dos iteraciones realizadas por semana y una retroalimentación cada 15 días. A continuación, se explican los resultados esperados en cada una de las entregas para dar claridad en el desarrollo del proyecto:

- Primera Entrega - Concebir la idea de investigación: Esta entrega de dos semanas se enfoca en la idea de investigación que a su vez se centra en la construcción de una aplicación para la identificación temprana de enfermedades en las mazorcas de cacao. Esta fue concebida por el director y de la mano de otro equipo de investigación enfocado en la identificación de la madurez en las mazorcas de cacao. La propuesta de los proyectos es enfocarse en una estrategia agronómica y añadir componentes tecnológicos para ofrecer soluciones que apoyen en los procesos de cosecha al agricultor y a su vez le proporcionen herramientas útiles.
- Segunda Entrega - Planteamiento del problema, justificación y objetivos del proyecto: Se establece el objetivo general y los objetivos específicos del proyecto de investigación. También se formula una pregunta de investigación a partir del problema determinado. Se redacta la justificación y la forma en que se dará solución al problema. Tercera Entrega - Marco Teórico y Revisión de la literatura: Se hace una búsqueda de conceptos, términos, técnicas, herramientas e información sobre la idea de proyecto y se elabora el marco teórico con toda la información recolectada. Se construye las ecuaciones de búsqueda para la revisión de la literatura, donde se aplica un criterio de exclusión e inclusión de términos para poder filtrar los artículos más interesantes que muestren las tendencias. Se hace una descripción de cada uno de los artículos seleccionados y para finalizar una conclusión con la propuesta detallada.
- Cuarta Entrega - Definición del tipo de investigación: Esta investigación involucra riesgo, perseverancia y predisposición. El tema planteado puede ser abordado de diferentes maneras debido a que posee un carácter libre para la detección de las enfermedades: Monilia y/o Fitóftora. Al final de la investigación se realizará un prototipo aplicativo que sirva de apoyo para la toma de decisiones de los agricultores y cacaotero. Mediante los datos obtenidos se preparará el terreno para nuevos estudios (observatorio). El costo y el tiempo que se tiene para llevar a cabo el proyecto es corto.
- Quinta Entrega – Definición de la hipótesis. Como proposición se espera que la máquina de inteligencia artificial sea capaz de discriminar mazorcas sanas de

enfermas, por consiguiente, desde el punto de vista de hipótesis nula se plantea: a partir de la metodología propuesta el algoritmo de aprendizaje automático no puede diferenciar mazorcas enfermas de mazorcas sanas, y como hipótesis alterna: a partir de la metodología propuesta el algoritmo de aprendizaje automático sí puede diferenciar mazorcas sanas de enfermas.

- Sexta Entrega – Conjunto de datos estandarizados y diseño del prototipo de aplicación móvil. Desarrollar diagramas de secuencia y construcción de diagramas de clases. Terminar las visitas a campo, la transformación y la estandarización del conjunto de datos. Comenzar la codificación, etiquetado y orientación de las imágenes. Por último, consolidar el conjunto de imágenes de entrenamiento, y prueba y Configurar la maquina personal para trabajar con las herramientas.
- Séptima Entrega – Maquina de aprendizaje calibrada y con análisis estadísticos. Comenzar a entrenar la máquina de aprendizaje con el conjunto de datos obtenido e implementarla mediante calibración y evaluación de esta. Comenzar a desarrollar la capa de acceso a los datos (Back-end) con la máquina de aprendizaje.
- Octava Entrega – Prototipo de aplicación funcional para la detección de mazorcas de cacao enfermas con Monilia y/o Fitóftora. Desarrollar la interfaz del prototipo de aplicación que permita integrar la máquina de aprendizaje. Por último, realizar pruebas del prototipo para obtener resultados y conclusiones.

6.3. Entrenamiento y calibración de la máquina de aprendizaje.

El entrenamiento de la máquina inicia con la selección del conjunto de datos estandarizados los cuales han pasado por un proceso de etiquetado que consiste en indicar dentro de cada imagen el objeto y asignarle la clase enferma o no enferma. Para esto se va a hacer uso de la herramienta LabelImg la cual guarda las anotaciones en archivos XML en formato PASCAL VOC, además es compatible con el formato YOLO, y luego se realiza la división de datos entre datos de entrenamiento y datos de prueba donde el 80% del conjunto de datos va a ser destinado para el entrenamiento del algoritmo y el 20% restante para hacer pruebas.

Una vez el conjunto de datos ha sido seleccionado y preparado se procede a trabajar con la arquitectura de red que ya trae preestablecida YOLOv4 la cual usa una variante de Darknet-53 que tiene una red de 53 capas convolucionales entrenada en Imagenet, adicionalmente se apilan 53 capas convolucionales para la tarea de detección lo que da una arquitectura subyacente totalmente convolucional de 106 capas para YOLOv4 (Bochkovskiy et al., 2020b). Posteriormente se le pasará como entrada el conjunto de datos etiquetado. Seguidamente se definen los

parámetros de entrenamiento como el número de épocas, el batch y la tasa de aprendizaje. Ya terminada la etapa de entrenamiento se procede a evaluarla.

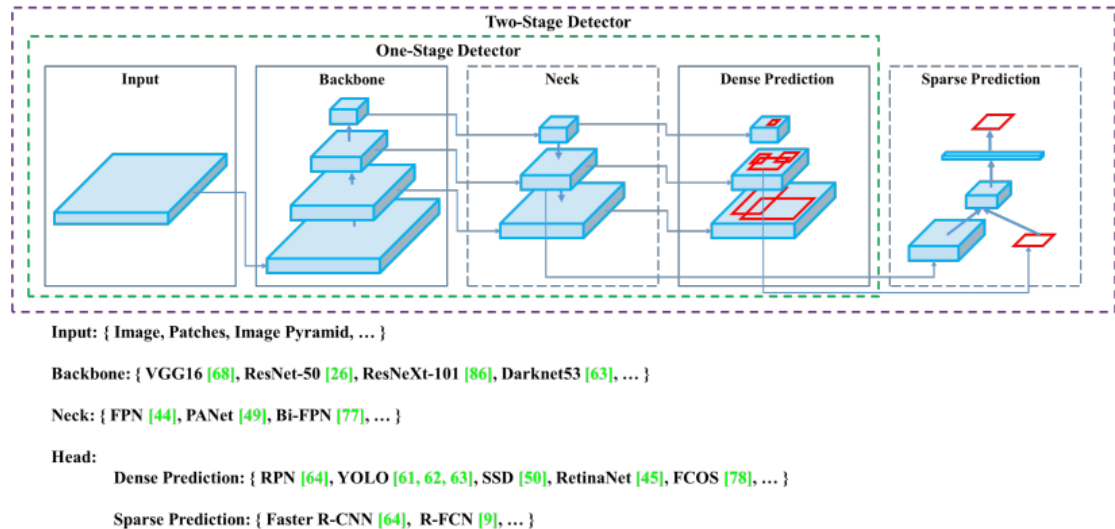


Figura 11 Arquitectura de YOLOv4
Fuente: (Bochkovskiy et al., 2020b)

6.3.1. Evaluación y calibración

En esta etapa se pone a prueba la información que la red ha obtenido del entrenamiento en la etapa anterior. Se evalúa la precisión del modelo en sus predicciones y llegado el caso que el rendimiento no sea adecuado, se realizara de nuevo la etapa de entrenamiento y se modificaran algunos parámetros hasta lograr una rendimiento y eficacia aceptable. Este proceso consiste en pasar los datos destinados a hacer pruebas por la red, para obtener los resultados de precisión y detección dados por la red y posteriormente compararlos. Finalmente se tomará la decisión si el modelo es adecuado o no para el problema planteado el cual es identificar las mazorcas de cacao enfermas con Monilia o Fitóftora.

7. RESULTADOS

7.1. Diseño del prototipo de aplicación móvil para la identificación de mazorcas de cacao enfermas con Monilia y/o Fitóftora:

El diseño del prototipo de aplicación móvil para la identificación de mazorcas de cacao enfermas con Monilia y/o Fitóftora tiene las siguientes características, las cuales tienen como actividades principales la recopilación de los requerimientos de software, la construcción de casos de usos, diagramas de secuencia y diagramas de clases, que son importantes para representar el uso de la aplicación, su funcionamiento y las distintas conexiones que debe realizar para conectarse con el modelo de aprendizaje de máquina resultante del posterior proceso de entrenamiento y calibración para realizar una detección por parte de la aplicativo móvil.

7.1.1. Requerimientos

Para la recopilación de los requerimientos de software se tienen en cuenta los siguientes requerimientos funcionales y no funcionales que se muestran en la Tabla 1.

FUNCIONALES	NO FUNCIONALES
Acceso a la cámara	Compatibilidad con Android 9.0 (Pie) y Herramienta offline
Reconocer la mazorca de cacao	Tiempo de inferencia

Tabla 1 Requerimientos funcionales y no funcionales

Fuente: Los autores

7.1.1.1. Funcionales

- **Acceso a la cámara:** La aplicación móvil tiene que contar con acceso a la cámara interna de un smartphone: El usuario tiene que dar permisos de control de cámara a la aplicación una vez esta inicie.
- **Reconocer la mazorca de cacao:** La máquina de aprendizaje debe identificar si la mazorca de cacao capturada esta sana o si por lo contrario tiene alguna de

las dos enfermedades previstas Monilia y Fitóftora, todo por medio del modelo de aprendizaje de maquina entrenado.

7.1.1.2. No funcionales

- **Compatibilidad con Android 9.0 (Pie) y herramienta offline:** La aplicación móvil es compatible con el sistema operativo Android desde la versión del kit de desarrollo de software de Android 9.0 (Pie). Permite el acceso y uso de la aplicación sin contar con una conexión a internet.
- **Tiempos de inferencia:** La aplicación móvil presenta unos tiempo de inferencia que pueden ir desde 100 ms hasta 600 ms

7.1.2. Casos de uso

En la Figura 12, se representan los casos de uso para identificar mazorcas de cacao por medio de un actor agricultor o persona a cargo de un cultivo. El actor se encarga de iniciar la cámara mediante el botón llamado “cámara” y orientar el dispositivo para que la mazorca pueda ser detectada por el algoritmo de TensorFlow y YoloV4. El algoritmo a su vez es el encargado de extender del modelo de aprendizaje de maquina resultante del proceso de entrenamiento y calibración.

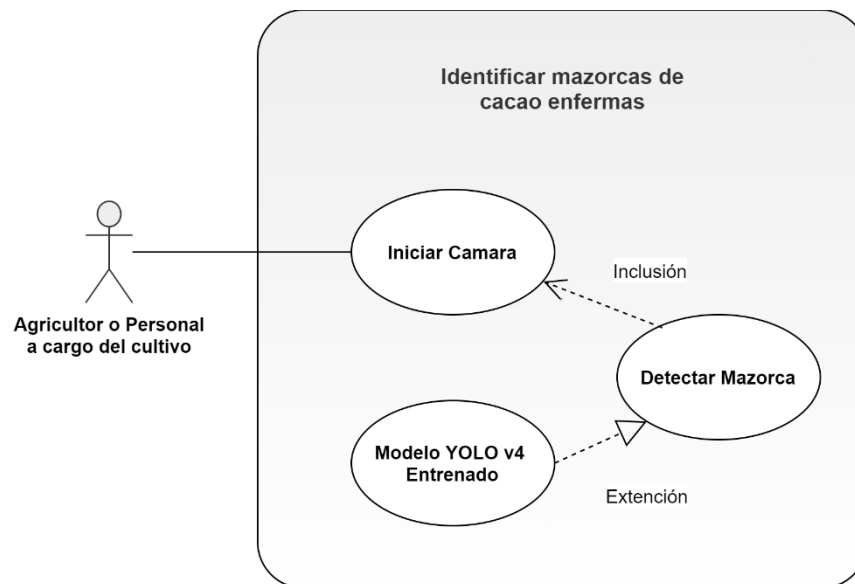


Figura 12 Diagrama de casos de uso para la identificación de mazorcas de cacao enfermas.

Fuente: Los autores

Para más información sobre los casos de uso, al final del documento en el apartado de anexos se encuentran organizadas las especificaciones de cada uno de los casos de uso, previamente mencionados en la Figura 12.

7.1.3. Diagramas de actividades

En Figura 13 se encuentra el diagrama de actividades que muestra el flujo de cada una de las actividades que realizaría la aplicación una vez se captura a tiempo real la imagen.

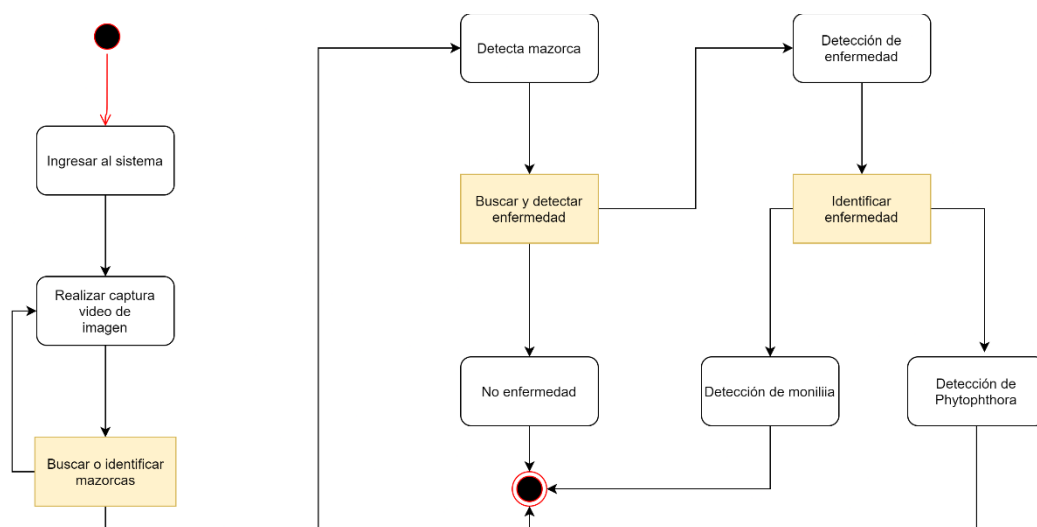


Figura 13 Diagrama de actividades para la identificación de mazorcas de cacao enfermas.

Fuente: Los autores

7.1.4. Diagrama de secuencia

En la Figura 14 se puede ver el diagrama de secuencia que muestra las capas que usa el prototipo de aplicación móvil para la detección de las mazorcas de cacao enfermas. El agricultor debe realizar una aproximación a la mazorca, una vez la interfaz de captura de videocámara a tiempo real este activa. Una vez se encuentre la mazorca la información es comunicada y traducida por el modelo de YOLOv4 para generar una inferencia, la cual dice si la mazorca se encuentra Sana o Enferma ya sea con Monilia o Fitóftora.

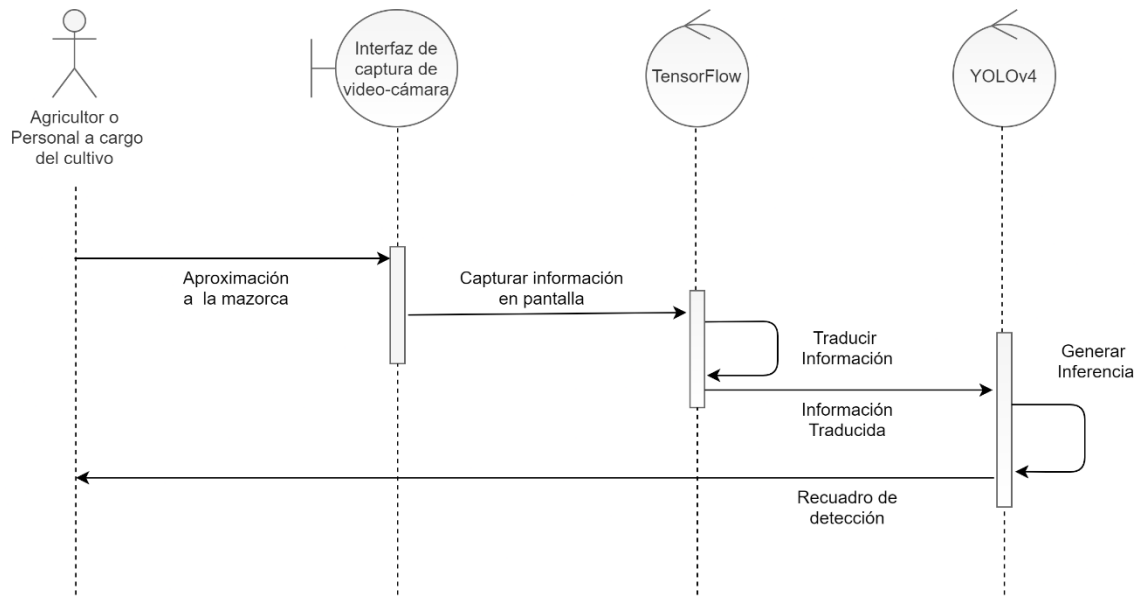


Figura 14 Diagrama de secuencia para la identificación de mazorcas de cacao enfermas.

Fuente: Los autores

7.1.5. Diagrama de clases

En la Figura 15 se pueden ver las clases de la aplicación Android que permiten la detección y la extensión con el modelo entrenado de YOLOv4. La clase “IniciarCamara” contiene un método incluido en TensorFlow que se encarga de hacer uso de la cámara del dispositivo móvil. Mientras que la clase de detección que en este caso es “DetectarMazorca” contiene el clasificador que extiende de la clase “Main” donde se inicializa el ModeloYOLOv4 entrenado y el archivo de etiquetas “coco.txt”

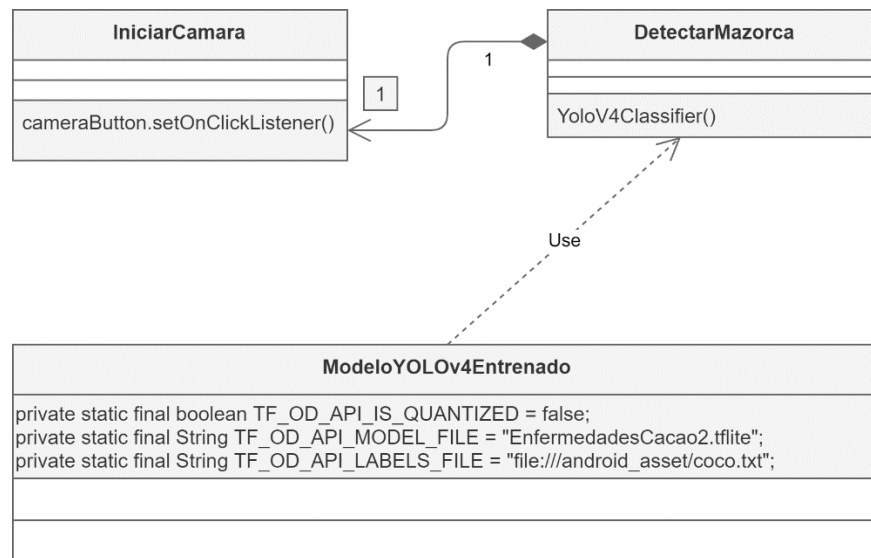


Figura 15 Diagrama de clases para la identificación de mazorcas de cacao enfermas.

Fuente: Los autores

7.2. Conjunto de datos consolidado

Para la consolidación del conjunto de datos se sigue una serie de pasos importantes que van desde la visita a la finca con plantaciones de cacao donde se capturan las imágenes hasta el proceso de etiquetado que cuenta con un ordenamiento especial para cada tipo de enfermedad contemplada en el proyecto. A continuación, se encuentran cada uno de los pasos que permiten la consolidación:

7.2.1. Visita y tipos de cacao

El levantamiento del conjunto de datos se realizó durante una visita al centro de atención al sector agropecuario del Playón, ubicado en la subsede Aguas Calientes, El playón, Santander. De la mano del encargado de los cultivos e instructor escalafonado y no escalafonado Néstor Edgardo Nieves Peña especialista en gestión ambiental y con conocimientos en ingeniería agronómica del SENA (Servicio Nacional de Aprendizaje), se recorrieron distintas plantaciones de mazorca de cacao en el centro agropecuario del playón. A medida que recorríamos cada plantación el encargado iba señalando distintas mazorcas de cacao enfermas cada una con inicios de Monilia o Fitóftora, así como también con estas enfermedades en un estado más avanzado. Entre los distintos cultivos de cacao se encontraron variedades de clones representativas de (Fedecacao, 2018) de los que se capturaron imágenes hay que resaltar los clones: FEAR 5, FSV41, FEC y CCN51.

7.2.2. Captura de imágenes con mazorcas enfermas y sanas

En el total de imágenes capturadas (500), son seleccionadas 300 según disponibilidad de información relacionada a la variedad y el tipo de enfermedad fitosanitaria entre Monilia, Fitóftora y Sana con la siguiente cantidad de elementos por categoría: Enferma con Monilia (100), enferma con Fitóftora (100) y mazorcas sanas (100). Las imágenes son tomadas con un celular Vivo Y50 el cual tiene las siguientes características: cámara con sensor principal de 13MP en lente angular con apertura focal f/2.2; sensor 8MP con apertura f/2.2; sensor de profundidad 2MP con apertura de f/2.4 y un sensor de 2MP con apertura f/2.4. Con la cual se obtienen imágenes con una dimensión de 3120x4160 pixeles, a una resolución de 13 mega pixeles, en un modelo de color sRGB y la categoría asignada por un experto.

7.2.3. Proceso para etiquetar (Labellmg)

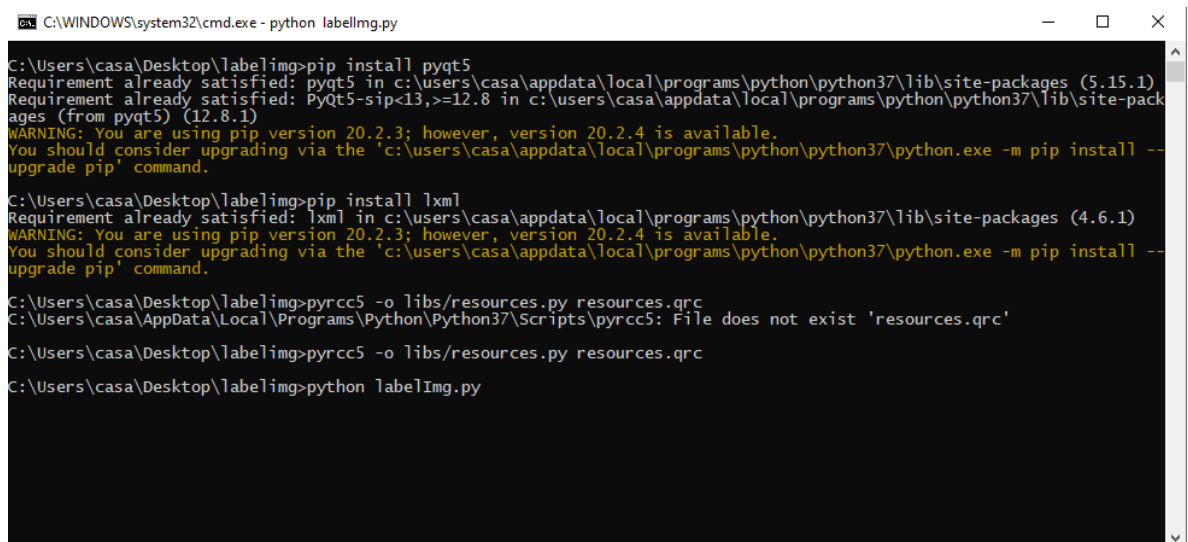
Con la finalidad de estandarizar las imágenes para el entrenamiento de la máquina de aprendizaje, se procede hacer uso de la herramienta Labellmg (Tzutalin, 2015). Como primer paso se procede a descargar la herramienta por medio de la siguiente url: “<https://github.com/tzutalin/Labellmg>”, para hacer uso de esta herramienta hay que tener instalado y actualizado Python 3.

The screenshot shows the GitHub repository page for 'tzutalin/labellmg'. The repository is on the 'master' branch, with 4 branches and 23 tags. A 'Clone' dropdown menu is open, showing options for cloning via HTTPS (GitHub CLI), cloning with Git or SVN, opening with GitHub Desktop, and downloading a ZIP file. The repository contains several folders and files, including 'build-tools', 'data', 'demo', 'libs', 'requirements', 'resources', 'tests', '.gitignore', '.travis.yml', and 'CONTRIBUTING.rst'. The 'About' section on the right describes Labellmg as a graphical image annotation tool and label object bounding boxes in images. It also includes links to a YouTube video, a README file, and a MIT License.

Figura 16 Captura repositorio Labellmg

Fuente: Los autores captura del repositorio web (Tzutalin, 2015)

Una vez se descarga el repositorio desde GitHub se procede a extraer la carpeta del archivo comprimido y ubicarla en el escritorio. Luego se abre la consola de comandos de Windows 10 (cmd.exe) y se navega para encontrar la carpeta Labellmg e instalar las librerías de PyQt5 y lxml por medio del sistema de gestión de paquetes de Python (Pip). Posteriormente se ejecuta el PyQt5 y se mueven los archivos resources.py y resources.qrc a la carpeta libs de la carpeta Labellmg.



```
C:\WINDOWS\system32\cmd.exe - python labellmg.py
C:\Users\casa\Desktop\labellmg>pip install pyqt5
Requirement already satisfied: pyqt5 in c:\users\casa\appdata\local\programs\python\python37\lib\site-packages (5.15.1)
Requirement already satisfied: PyQt5-sip<13,>=12.8 in c:\users\casa\appdata\local\programs\python\python37\lib\site-packages (from pyqt5) (12.8.1)
WARNING: You are using pip version 20.2.3; however, version 20.2.4 is available.
You should consider upgrading via the 'c:\users\casa\appdata\local\programs\python\python37\python.exe -m pip install --upgrade pip' command.
C:\Users\casa\Desktop\labellmg>pip install lxml
Requirement already satisfied: lxml in c:\users\casa\appdata\local\programs\python\python37\lib\site-packages (4.6.1)
WARNING: You are using pip version 20.2.3; however, version 20.2.4 is available.
You should consider upgrading via the 'c:\users\casa\appdata\local\programs\python\python37\python.exe -m pip install --upgrade pip' command.
C:\Users\casa\Desktop\labellmg>pyrc5 -o libs/resources.py resources.qrc
C:\Users\casa\AppData\Local\Programs\Python\Python37\Scripts\pyrc5: File does not exist 'resources.qrc'
C:\Users\casa\Desktop\labellmg>pyrc5 -o libs/resources.py resources.qrc
C:\Users\casa\Desktop\labellmg>python labellmg.py
```

Figura 17 Captura configuración y ejecución Labellmg

Fuente: Los autores

Al abrir la herramienta Labellmg se procede a cargar imagen por imagen para su respectivo etiquetado ya sea de Monilia, Fitóftora o Sana.

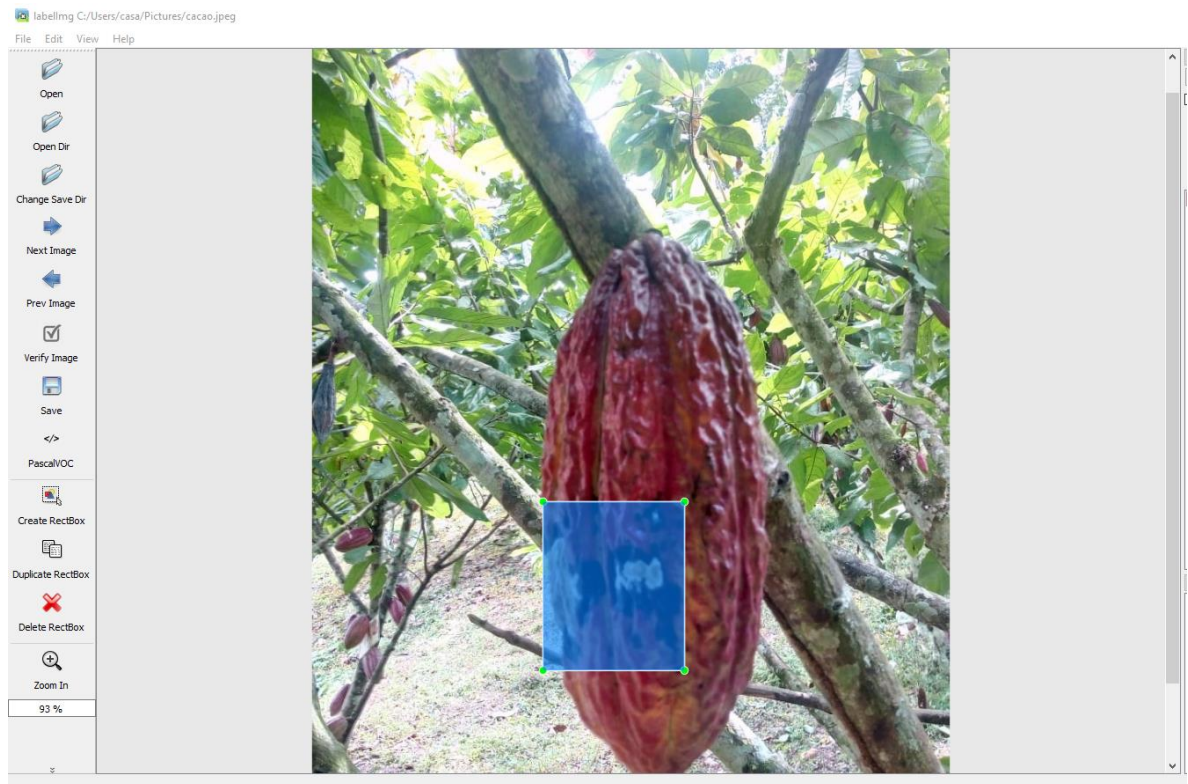


Figura 18 Etiquetando mazorca con herramienta LabelImg

Fuente: Los autores

Para hacer uso de la herramienta LabelImg se etiquetan, transforman y organizan las imágenes procurando que las mazorcas queden orientadas de manera vertical. Se procede a etiquetar 100 imágenes por categoría: enferma con Monilia (100), enferma con Fitóftora (100) y Sana (100) para continuar con el respectivo entrenamiento de la máquina de aprendizaje.

7.2.4. Conjunto de datos consolidado y Kaggle

Finalmente, queda consolidado el dataset de imágenes de mazorcas de cacao para dar cumplimiento al objetivo, este conjunto se puede encontrar en la plataforma Kaggle que es una comunidad en línea compuesta por científicos de datos y profesionales del aprendizaje automático. Por medio del nombre Cocoa Diseases YOLOv4 (Serrano et al., 2020) se podrá acceder al sitio, en el que se puede realizar la descarga del dataset.

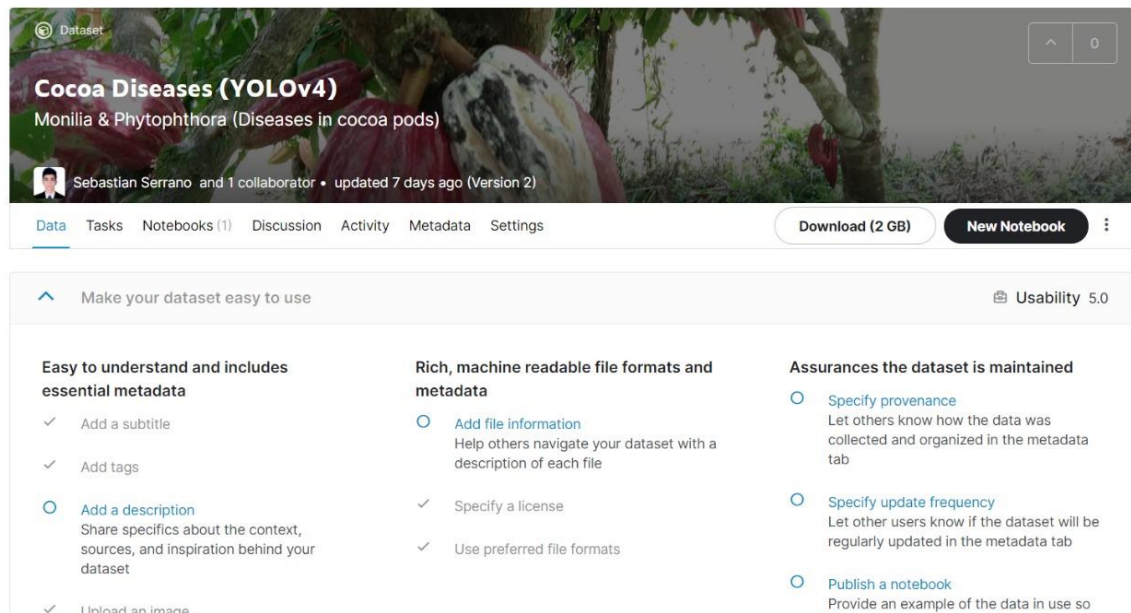


Figura 19 Cocoa Diseases (YOLOv4) Dataset Kaggle
Fuente: Los autores (Serrano et al., 2020)

7.3. Máquina de detección para la identificación de mazorcas de cacao enfermas con Monilia y/o Fitóftora.

Para implementar y calibrar la máquina de aprendizaje se usa un repositorio llamado Darknet, este emplea la tecnología YOLOv4 para la detección de objetos. Una vez el conjunto de datos está listo, se descarga el repositorio y con el conjunto de datos se sube a una carpeta drive donde se continua con el proceso mediante Google Colab, el servicio Cloud que permite el uso gratuito de GPUs y TPUs de Google (Colaboratory, 2020) Después de esto se hace la declaración de las variables que en este caso son las listas de datos de imágenes, con sus localizaciones teniendo en cuenta la división del conjunto de datos respecto a las etiquetas de detección y por último se despliega el entrenamiento con la máquina de Google que se encarga de crear unos pesos para que el modelo funcione correctamente.

A continuación, se muestra en detalle cada uno de los pasos para la calibración de la máquina de aprendizaje:

7.3.1. Darknet

El primer paso es buscar una fuente que tenga información del correcto uso de YOLOv4 para la detección de objetos. Se encuentra entonces que una de las mejores opciones es el repositorio o directorio llamado Darknet. Este permite ser clonado mediante la plataforma GitHub, pertenece al señor Aleksey Bochkovskiy el directorio de archivos de código abierto especialmente programados para usar la tecnología YOLOv4 y a su equipo quien fueron los encargados de documentar la literatura de YOLOv4 (Bochkovskiy et al., 2020b). La tecnología mencionada también conocida por ser una de las mejores redes neuronales a tiempo real que existen para un conjunto de datos, permite usar un detector de última generación que es más rápido (FPS) y preciso (MS COCO AP50.95 and AP50) que los detectores alternativos disponibles. El detector se puede entrenar y usar en una GPU convencional con 8 a 16 Gb-VRam lo que hace posible un amplio uso. (Colaboratory, 2020)



Figura 20 Detección de objetos modelo entrenado con el MSCOCO dataset
Fuente: (Bochkovskiy, 2020)

7.3.2. Google Colab

En el segundo paso se usa una alternativa de herramienta llamada Google Colaboratory, (Colaboratory, 2020) con la que el grupo de investigación GTI y la línea de Telemática de la Universidad Autónoma de Bucaramanga esta trabajado. Es importante debido a que no se cuenta con una máquina con las características necesarias (GPU) para hacer un entrenamiento de YOLOv4. Colaboratory, permite el uso gratuito de GPUs y TPUs de Google (Colaboratory, 2020). Además, Google Colab contiene librerías de Python imprescindibles para utilizar el repositorio Darknet. Una vez en la notebook de Google Colab se siguen los pasos y criterios para hacer el entrenamiento completo.

7.3.3. Datos y etiquetas

El conjunto de datos MSCOCO que usa Darknet para entrenar y mostrar que tan eficiente es la detección de personas, carros, animales, entre una gran variedad de objetos (Lin et al., 2014). Sin embargo, para construir y entrenar un detector de objetos personalizado se debe recopilar un conjunto de datos personalizados y realizar una configuración del entrenamiento YOLOv4. Es aquí donde se tiene que resaltar lo esencial de los datos y el trabajo realizado con la herramienta Labelimg.

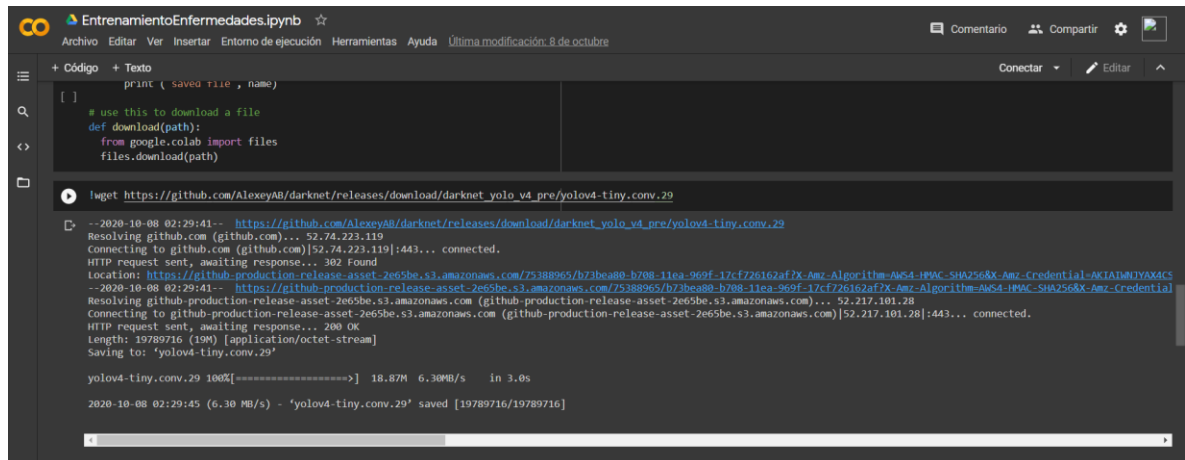
Para empezar, se debe declarar en un archivo de texto con las rutas de cada uno de los archivos de imágenes del conjunto de datos. Se construyen dos archivos, donde se resalta la regla del 80-20, que divide los datos de la siguiente manera: el 80% para datos de entrenamiento y el 20% restante para datos de prueba. Con el objetivo de una mayor facilidad, los datos se subieron en una carpeta de Drive compartida y se obtuvieron las rutas desde el notebook de Google Colab para completar el archivo. También se construye un archivo de nombres (.names) con las etiquetas utilizadas previamente en el proceso de etiquetar, las cuales son los tres estados previstos en este proyecto para una mazorca de cacao: Fitóftora, Monilia y Sana. Luego de esto se declaran los datos de los archivos de texto y el archivo de nombres en un archivo de datos (.data) para usarlo con el comando especial.

Antes de comenzar el entrenamiento se debe editar el archivo (.cfg) para ajustarlo al conjunto de datos personalizado y a las etiquetas establecidas para la detección. Se edita la parte de las clases que son el número de etiquetas y se realiza un cálculo para obtener el número de lotes máximo, los pasos y los filtros que debe realizar el entrenamiento.

7.3.4. Entrenamiento

El entrenamiento de detección de objetos YOLOv4 con Darknet en la nube de Google Colab inicia con la habilitación de la GPU que proporciona Google, entre las que se encuentran disponibles están las tesla K80, T4, P4 y P100 de NVIDIA (Colaboratory, 2020). Una vez se habilita la GPU para acelerar el entrenamiento se obtiene el repositorio Darknet mediante el comando Git o permitiéndole acceso al drive donde se encuentran los archivos editados para el entrenamiento. Para que el entrenamiento funcione se cambian configuraciones para tener GPU y OPENCV activados. También se verifica la versión de CUDA que es la Arquitectura Unificada de Dispositivos de Cómputo, este es una plataforma de computación paralela y un modelo de programación inventado por NVIDIA, permite aumentos drásticos en el rendimiento informático aprovechando la potencia de la unidad de procesamientos

gráficos GPU (Ghorpade, 2012). Lo siguiente es descargar unos pesos pre-entrenados para las capas convolucionales de la red YOLOv4, el uso de estos pesos ayuda a que el detector de objetos personalizado sea mucho más preciso y no tenga que entrenar tanto tiempo pero no es obligatorio.



```
EntrenamientoEnfermedades.ipynb
Archivo Editar Ver Insertar Entorno de ejecución Herramientas Ayuda Última modificación: 8 de octubre
Comentario Compartir
+ Código + Texto
print ( saved file , name)
[ ]
# use this to download a file
def download(path):
    from google.colab import files
    files.download(path)

!wget https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v4_pre/yolov4-tiny.conv.29

-2020-10-08 02:29:41-- https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v4_pre/yolov4-tiny.conv.29
Resolving github.com (github.com)... 52.74.223.119
Connecting to github.com (github.com)[52.74.223.119]:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://github-production-release-asset-2665be.s3.amazonaws.com/75388965/b72bea88-b708-11ea-969f-17cf726162af?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIA1M1YVX4CS-2020-10-08-02:29:41--https://github-production-release-asset-2665be.s3.amazonaws.com/75388965/b72bea88-b708-11ea-969f-17cf726162af?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=github-production-release-asset-2665be.s3.amazonaws.com (github-production-release-asset-2665be.s3.amazonaws.com)... 52.217.101.28
Connecting to github-production-release-asset-2665be.s3.amazonaws.com (github-production-release-asset-2665be.s3.amazonaws.com)[52.217.101.28]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 19789716 (19M) [application/octet-stream]
saving to: 'yolov4-tiny.conv.29'

yolov4-tiny.conv.29 100%[=====] 18.87M 6.30MB/s in 3.0s

2020-10-08 02:29:45 (6.30 MB/s) - 'yolov4-tiny.conv.29' saved [19789716/19789716]
```

Figura 21 Detección de objetos comando de descarga pesos pre-entrenados
Fuente: Los autores

Después de lo anterior por fin está preparado, entonces se ejecuta el comando que se muestra en la Figura 22 para iniciar el entrenamiento del detector de objetos YOLOv4. Se puede ver que para entrenar se usa el archivo .data que especifica las rutas del conjunto de datos, como las etiquetas, también el archivo de configuraciones .cfg y por ultimo los pesos pre-entrenados para que el modelo converja y sea preciso. En el comando se ve la bandera -dont_show que evita que el grafico aparezca en Google Colab y la bandera -map es una superposición que elabora una precisión promedio en el gráfico. El comando se encarga de mostrar la información de cada de los lotes donde se ve las pérdidas y el porcentaje de éxito en cada detección, también se muestra las iteraciones y promedio de las horas que dura el entrenamiento. Para el entrenamiento de la maquina detectora de mazorcas de cacao enfermas se llevó algo más de 15 horas y se realizaron 6000 iteraciones.

```

total_bbox = 856563, rewritten_bbox = 0.131572 %
v3 (iou loss, Normalizer: (iou: 0.07, cls: 1.00) Region 30 Avg (IOU: 0.915908, GIoU: 0.915052), Class: 0.999972, Obj: 0.996995, No Obj: 0.008313)
v3 (iou loss, Normalizer: (iou: 0.07, cls: 1.00) Region 37 Avg (IOU: 0.875760, GIoU: 0.873214), Class: 0.999432, Obj: 0.908972, No Obj: 0.002913)
total_bbox = 856563, rewritten_bbox = 0.131572 %
v3 (iou loss, Normalizer: (iou: 0.07, cls: 1.00) Region 30 Avg (IOU: 0.870397, GIoU: 0.867339), Class: 0.989581, Obj: 0.938311, No Obj: 0.007613)
v3 (iou loss, Normalizer: (iou: 0.07, cls: 1.00) Region 37 Avg (IOU: 0.802676, GIoU: 0.795330), Class: 0.999024, Obj: 0.698282, No Obj: 0.002613)
total_bbox = 856584, rewritten_bbox = 0.131569 %
v3 (iou loss, Normalizer: (iou: 0.07, cls: 1.00) Region 30 Avg (IOU: 0.892847, GIoU: 0.890604), Class: 0.999476, Obj: 0.998599, No Obj: 0.005513)
v3 (iou loss, Normalizer: (iou: 0.07, cls: 1.00) Region 37 Avg (IOU: 0.843176, GIoU: 0.838391), Class: 0.998801, Obj: 0.972007, No Obj: 0.002613)
total_bbox = 856603, rewritten_bbox = 0.131566 %
v3 (iou loss, Normalizer: (iou: 0.07, cls: 1.00) Region 30 Avg (IOU: 0.864282, GIoU: 0.863899), Class: 0.991901, Obj: 0.830342, No Obj: 0.003213)
v3 (iou loss, Normalizer: (iou: 0.07, cls: 1.00) Region 37 Avg (IOU: 0.795416, GIoU: 0.787612), Class: 0.995807, Obj: 0.683433, No Obj: 0.001613)
total_bbox = 856616, rewritten_bbox = 0.131564 %
v3 (iou loss, Normalizer: (iou: 0.07, cls: 1.00) Region 30 Avg (IOU: 0.877153, GIoU: 0.874358), Class: 0.999243, Obj: 0.892273, No Obj: 0.007013)
v3 (iou loss, Normalizer: (iou: 0.07, cls: 1.00) Region 37 Avg (IOU: 0.861466, GIoU: 0.859576), Class: 0.999820, Obj: 0.838476, No Obj: 0.001713)
total_bbox = 856631, rewritten_bbox = 0.131562 %

6000: 0.264453, 0.223745 avg loss, 0.000026 rate, 1.297626 seconds, 384000 images, 0.266480 hours left
Saving weights to backup/yolov4-tiny-obj_6000.weights
Saving weights to backup/yolov4-tiny-obj_last.weights
Saving weights to backup/yolov4-tiny-obj_final.weights
If you want to train from the beginning, then use flag in the end of training command: -clear

Evaluar

[ ] imshow('chart.png')

```

Figura 22 Detección de objetos entrenando modelo Enfermedades del cacao.

Fuente: Los autores

7.3.5. Modelo entrenado

El modelo final es guardado antes que la ejecución o entrenamiento del detector de objetos YOLOv4 esté terminada. En la Figura 23 se muestran los modelos que contiene la carpeta backup, son un tipo de archivo. (.weights) o pesos, estos pesos son generados durante la ejecución del entrenamiento, cada 1000 iteraciones se crean uno nuevo. También se generan a su vez unos pesos de respaldo para contener al mejor peso y al último que fue generado.

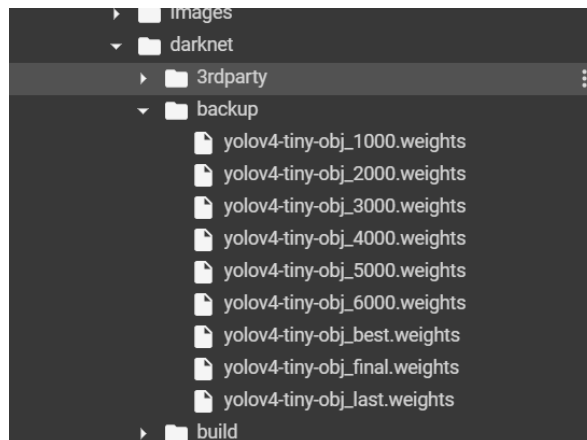


Figura 23 Backup de modelos o pesos generados en el entrenamiento.

Fuente: Los autores

El indicado es el .weights final, este contiene el mejor modelo después de las 6000 iteraciones. Este modelo denominado también máquina de aprendizaje es el encargado de detectar enfermedades en las mazorcas de cacao. En la Figura 24 se recuperó el mAP promedio de un 60% de las primeras y últimas 3000 iteraciones

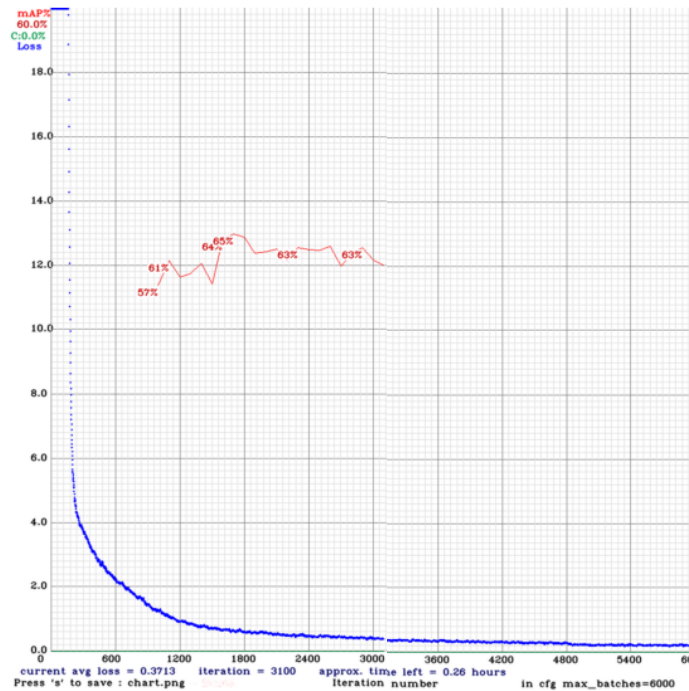


Figura 24 Resultado mAP del entrenamiento

Fuente: (Bochkovskiy et al., 2020b) Los autores

7.4. Prototipo de aplicación funcional para la detección de mazorcas de cacao enfermas con Monilia y/o Fitóftora.

La construcción del prototipo de aplicación funcional para la detección de mazorcas de cacao enfermas con Monilia y/o Fitóftora también presenta una serie de pasos con las que se resalta el uso de un repositorio de GitHub perteneciente a la cuenta hunglc007 que contiene la herramienta TensorFlow como aplicativo móvil especializado para la detección de objetos por medio de un modelo entrenado. (Hunglc007, 2020)

7.4.1. Repositorio TensorFlow (YOLOv4-tflite)

El repositorio de TensorFlow YOLOv4-tflite se obtiene clonando desde la plataforma GitHub, pertenece a la cuenta hunglc007 (Hunglc007, 2020). Este repositorio permite convertir el modelo. weights entrenado con YOLOv4 a un formato. tflite TensorFlow Lite para poder ejecutarlo como una aplicación móvil (Lee et al., 2019). La aplicación es desarrollada para Android por lo que es necesario tener Android Studio para hacer los cambios y modificaciones. La aplicación se encarga de analizar los frames obtenidos a tiempo real desde el acceso a la cámara e incluye un detector elaborado por TensorFlow que extiende del modelo previamente entrenado con YOLOv4 y Darknet para detectar mazorcas de cacao sanas como también enfermas con Monilia y Fitóftora. Esta herramienta tiene distintas versiones, se puede ejecutar la versión demo, TensorRT y la Tflite que permite modelos pequeños y es elaborada para dispositivos móviles pues permite portabilidad desde cualquier dispositivo y detección a tiempo real.

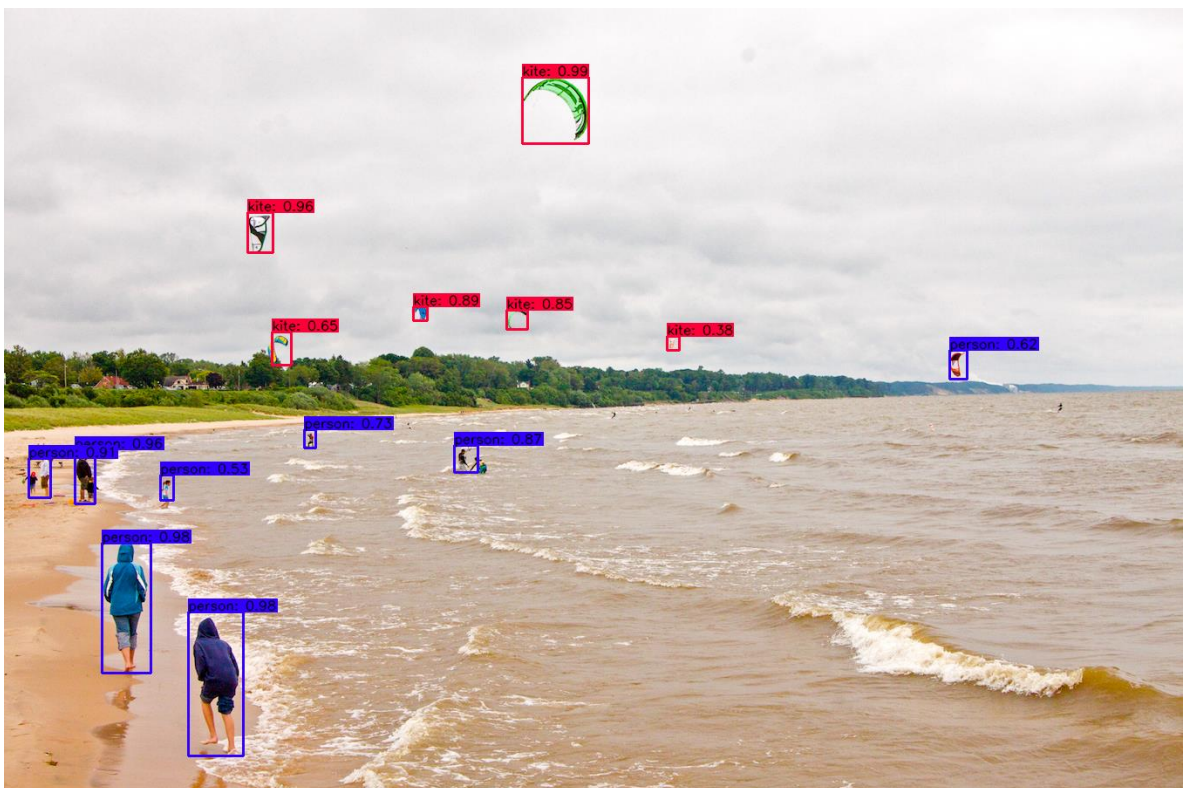


Figura 25 Ejemplo de detección con Tflite y el conjunto de datos MSCOCO
Fuente: (Hunglc007, 2020)

7.4.2. Modelo y conversión de pesos (YOLO v4 a TensorFlow Lite)

En cuanto al modelo hay que convertirlo para eso hay que entrar en la ruta donde se encuentra el repositorio y utilizar el comando de Python para la conversión del modelo que se encuentra en el repositorio web de GitHub:

```
python save_model.py --weights yolov4-tiny-obj_final.weights --output
./checkpoints/EnfermedadesCacao2 --input_size 416 --model yolov4 --framework
tflite --tiny
```

Figura 26 Comando para guardar YOLOv4 – Tflite

Fuente: Los autores

Después de guardar el modelo y especificar el nombre de salida se realiza la conversión con otro comando Python obtenido también del repositorio web de GitHub.

```
python convert_tflite.py --weights ./checkpoints/EnfermedadesCacao2 --output
./checkpoints/EnfermedadesCacao2.tflite
```

Figura 27 Comando para convertir YOLOv4 – Tflite

Fuente: Los autores

Una vez ejecutados estos dos comandos el modelo. tflite está listo para configurar desde Android Studio con las etiquetas y el modelo personalizado que se ha creado en la carpeta checkpoints.

7.4.3. Configuración etiquetas y modelo personalizado

La configuración desde Android Studio consiste en el traslado del modelo .tflite previamente convertido que se encuentra en la carpeta checkpoints a la carpeta assets en Android Studio, después solo queda inicializar las variables, usar rutas de los archivos, como lo son: el modelo. tflite y las etiquetas que originalmente se encuentran en el archivo coco.txt. El archivo contiene las etiquetas usadas para el conjunto de datos MSCOCO, por lo que hay que reformar estos archivos por las etiquetas usadas a lo largo del proyecto (Fitóftora, Monilia y Sana). Las clases en donde se remplazan las rutas del modelo son el MainActivity.java y el DetectorActivity.java. Hay que asegurarse de que no existan errores a la hora de ejecutarlo y convertirlo en formato .app para el dispositivo Android, una vez configurado se ejecuta y se obtiene una app como la de la Figura 28.



Figura 28 App Enfermedades Cacao

Fuente: Los autores

7.4.4. Prototipo de aplicación móvil

La aplicación desarrollada con el kit de desarrollo de software de Android 9.0 (Pie) cuenta con un botón para acceder a la cámara y comenzar la detección a tiempo real. El acceso a la cámara es uno de los requerimiento más importantes, este permite realizar el procesado mediante el componente para realizar una inferencia. Al abrir por primera vez la aplicación, esta solicita permisos. Es importante aceptar las condiciones para que la aplicación pueda tener acceso a la cámara. A continuación, se puede ver una imagen en donde la aplicación está haciendo uso de la detección de las mazorcas de cacao y muestra en pantalla mediante las etiquetas usadas que tipo de enfermedad tiene: Fitóftora, Monilia o si por lo contrario la mazorca se encuentra sana. Los resultados obtenidos se pueden ver en la Figura 29

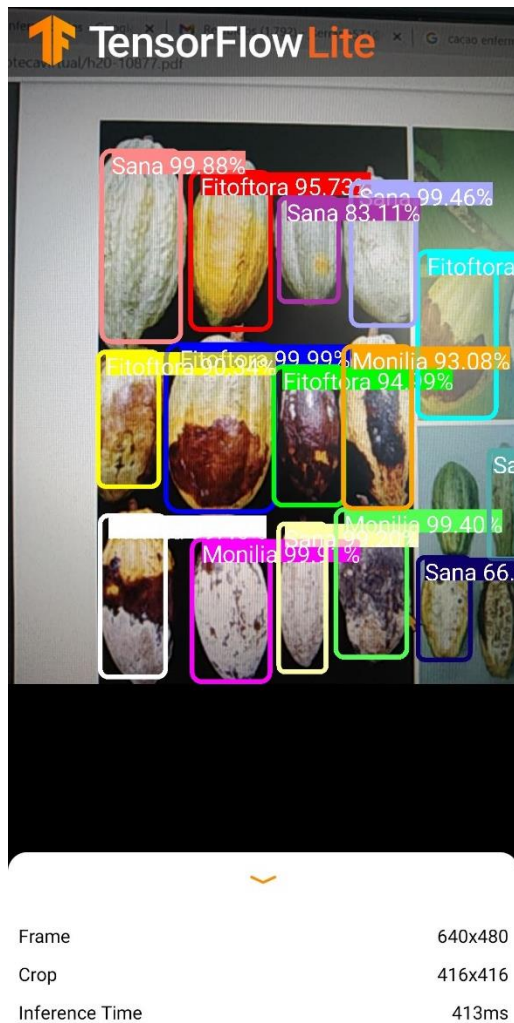


Figura 29 Prototipo de aplicación móvil funcional para detectar mazorcas
 Fuente: Los autores

Además del reconocimiento, en la Figura 29 la aplicación muestra en pantalla el tiempo de inferencia que transcurre mientras el detector reconoce la mazorca, como también el tamaño de los marcos que captura a tiempo real la cámara.

8. DISCUSIONES

En las distintas investigaciones encontradas y consignadas en los antecedentes del presente proyecto, se presenta una afinidad por abordar la detección de enfermedades provocadas por la Fitóftora en mazorcas de cacao. La manera en la que cada autor trabaja una solución a la detección de enfermedades en mazorcas de cacao va desde la implementación de algoritmos como el K-means con máquinas de soporte vectorial en el caso del autor Tan Daniel Stanley para medir el nivel de infección en mazorcas de cacao (Tan et al., 2016) hasta el uso de filtros de labor y redes convolucionales por parte de Basri Harli Indrabayu que permite identificar las características de los pixeles en las imágenes capturadas a las mazorcas de cacao (Basri et al., 2019). Siguiendo esta línea tecnológica el presente proyecto trabajo una solución haciendo uso de una librería de detección de objetos YOLOv4 (Lin et al., 2014). Esto implica que existe un problema de clasificación y segmentación por tanto fue necesario hacer uso de otra tecnología llamada Labellmg (Tzutalin, 2015). Este ejercicio de segmentación se logró mediante la identificación manual de distintas enfermedades en 500 imágenes, estas imágenes están compuestas por imágenes de mazorcas de cacao con monilia, mazorcas de cacao con Fitóftora y mazorcas de cacao sanas. En comparación a la cantidad de imágenes estandarizadas y utilizadas en promedio en la literatura, es de resaltar que el actual proyecto uso una mayor cantidad de imágenes gracias al trabajo de campo realizado en la consolidación del conjunto de datos y respecto al proyecto de nuestro apreciado compañero del grupo de investigación GTI y la línea de investigación de telemática Felipe Heredia y su equipo de proyecto de investigación. (Heredia Gómez et al., 2020).

En cuanto al conjunto de datos consolidado este se encuentra guardado en la comunidad en línea Kaggle donde esta a disponibilidad del público. Las imágenes contienen las etiquetas generadas con Labellmg (Tzutalin, 2015) de forma que sirva como apoyo para futuras investigaciones y trabajos. Por ejemplo con otras propuestas de servicios externos que permitan hacer fortalecer los procesos de producción del cacao. Este prototipo es un prototipo funcional, lo que implica que la verificación de las características de la herramienta es principalmente técnica y por consiguiente es importante seguir trabajando esta línea de proyectos buscando identificar diferentes elementos o barreras tecnológicas que permitan desarrollar un prototipo que pueda ser llevado hasta un usuario final: un agricultor de cacao o un científico enfocado a investigar el cacao.

9. CONCLUSIONES

El proyecto realizado tiene el potencial de contribuir de manera importante al fortalecimiento del sector agricultor colombiano. Puesto que su diseño está enfocado en apoyar los procesos para la producción de cacao nacional, consiguiendo ser una herramienta útil, con la que se espera promover el uso tecnológico en las distintas fincas cacaoteras permitiendo a agricultores e investigaciones agropecuarias mejorar los procesos fitosanitarios. Cabe aclarar que, la implementación abarcó un pequeño campo de las necesidades reales que tiene el sector agricultor colombiano. El uso de tecnologías como la arquitectura YOLOv4 para la máquina de aprendizaje y la construcción del aplicativo móvil offline con TensorFlow para la detección de mazorcas de cacao que fue adaptado al problema abordado en el proyecto, puede ser usada de diversas maneras en cultivos y lugares de investigación. No solo identificando mazorcas de cacao, sino enfocándose en distintas variedades de frutos permitiendo fortalecer una gran cantidad de procesos que impulsen al país a nuevos retos globales.

10. TRABAJOS FUTUROS

Una de las dificultades del proyecto se centra en la identificación de las mazorcas enfermas, pues muchas de ellas germinan demasiado rápido y adquieren Fitóftora o Monilia en una etapa más bien tardía. Se prevé que la identificación de mazorcas enfermas en futuros trabajos pueda centrarse en la captura de datos donde las mazorcas capturadas tengan una calidad fotográfica relevante. Permitiendo analizar a menor escala las mazorcas y su capa exterior. Según el instructor escalafonado y no escalafonado Néstor Edgardo Nieves Peña especialista en gestión ambiental y con conocimientos en ingeniería agronómica del SENA (Servicio Nacional de Aprendizaje) las enfermedades se pueden identificar por abultamientos o inflamaciones por parte de la mazorca, que puede ser vistas mediante una inspección más especializada. Lo que proporciona una oportunidad a la captura de datos con más calidad y enfoque permitiendo que herramientas fortalezcan la evaluación tanto de las enfermedades como de los procesos agrícolas en los cultivos de cacao.

En este sentido es importante resaltar que en la limitación de este trabajo no existe un sistemas de actualización, es decir la aplicación al ser offline utiliza la red con la cual se entrenó. Por consiguiente todo el proceso investigativo puede ser mejorado mediante cambios en su alcance, ya sea aumentando la precisión o considerando etapas de la producción donde el cacao se encuentre en una edad más temprana donde sea necesario hacer un proceso completo desde el inicio centrándose en la calidad del conjunto de datos para el entrenamiento del modelo YOLO v4. Otra limitación importante para resaltar es que este proyecto se enfocó en determinar e identificar si una mazorca se encontraba enferma o no, sin embargo en las implicaciones practicas del presente proyecto se encontró que una identificación en una edad temprana de las mazorcas enfermas permitiría detectar el nivel de infección que posee cada mazorca analizada.

11. REFERENCIAS

- Alegre, E., Pajares, G., & De La Escalera, A. (2016). *Conceptos y Métodos en Visión Por Computador*.
- Bagnato, J. I. (2018, November 29). *Convolutional Neural Networks: La Teoría explicada en Español | Aprende Machine Learning*.
<https://www.aprendemachinelearning.com/como-funcionan-las-convolutional-neural-networks-vision-por-ordenador/>
- Bancolombia. (2018). *Guía completa sobre el cultivo de cacao*.
- C, N. C. (2010). Nvidia cuda. *Compare A Journal Of Comparative Education*.
- Chang, D. T. (n.d.). *Concept-Oriented Deep Learning*.
- Clúster Santander - Cluster Cacao. (n.d.). Retrieved April 14, 2020, from
<http://clustersantander.com/bienvenido-cluster-cacao/>
- Corpoica, T., & Suiza, L. (2019). *Tcs 01*.
- Cowley, J. (2018). Redes neuronales convolucionales. *Ibm*, 1.
<https://www.ibm.com/developerworks/ssa/library/cc-convolutional-neural-network-vision-recognition/index.html>
- El Naqa, I., & Murphy, M. J. (2015). What Is Machine Learning? In *Machine Learning in Radiation Oncology* (pp. 3–11). Springer International Publishing.
https://doi.org/10.1007/978-3-319-18305-3_1
- Española, R. A., & Española, R. A. (n.d.-a). -cultura. *Diccionario de La Lengua Española*. Retrieved February 21, 2020, from
<http://buscon.rae.es/draeI/SrvltGUIBusUsual?origen=RAE&LEMA=-cultura>
- Española, R. A., & Española, R. A. (n.d.-b). agro-. *Diccionario de La Lengua Española*. Retrieved February 21, 2020, from
<http://buscon.rae.es/draeI/SrvltGUIBusUsual?origen=RAE&LEMA=agro->
- Espinosa-garcía, J. A., Uresti-gil, J., Vélez-izquierdo, A., Moctezuma-lópez, G., & Daniel, H. (2015). Productividad y rentabilidad potencial del cacao (*Theobroma cacao L.*) en el trópico mexicano * Productivity and profit potential of cocoa (*Theobroma cacao L.*) in the Mexican tropics Resumen Introducción. *Revista Mexicana de Ciencias Agrícolas*, 6, 1051–1063.
- Experts System Team. (2017). *What is Machine Learning? A definition - Expert System*. <https://expertsystem.com/machine-learning-definition/>
- Fedecacao. (2018). Federacion Nacional De Cacaoteros Fondo Nacional Del Cacao. *On Line*, 53.
https://www.fedecacao.com.co/portal/images/MEMORIAS_V_SEMINARIO/001_Dr_Eduard_Baquero_-_Fedecacao.pdf
- Fedecacao. (2019). *Informe de ejecución plan de ingresos, inversiones y gastos del fondo nacional del cacao durante el cuarto trimestre y consolidado vigencia 2018*. 1–228.
https://www.fedecacao.com.co/portal/images/INFORME_2018.pdf
- ICA. (2012). *Manejo fitosanitario del cultivo del cacao (Theobroma cacao L): medidas para la temporada invernal*.
- Institute of Electrical and Electronics Engineers. (2017). *2017 IEEE 2nd*

- International Conference on Big Data Analysis (ICBDA 2017) : March 10-12, 2017, Beijing, China. 721–724.*
- Lutz, M. (2007). Learning Python. In *Icarus*. [https://doi.org/10.1016/0019-1035\(89\)90077-8](https://doi.org/10.1016/0019-1035(89)90077-8)
- Martín Abadi, A. A. P. B. E. B., Zhifeng Chen, C. C. G. S. C. A. D., Jeffrey Dean, M. D. S. G. I. G., Andrew Harp, G. I. M. I. R. J. Y. J., Lukasz Kaiser, M. K. J. L. D. M. M. S., Rajat Monga, S. M. D. M. C. O. J. S., Benoit Steiner, I. S. K. T. P. T., Vincent Vanhoucke, V. V. F. V., Oriol Vinyals, P. W. M. W. M. W., & Yuan Yu y Xiaoqiang Zheng. (2015). *TensorFlow White Papers*. <https://www.tensorflow.org/about/bib>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016-Decem*, 779–788. <https://doi.org/10.1109/CVPR.2016.91>
- Redmon Joseph, & Farhadi, A. (2018). *YOLO: detección de objetos en tiempo real*. <https://pjreddie.com/darknet/yolo/>
- Society, N. G. (2019). *The Development of Agriculture | National Geographic Society*. Aug. 19. <https://www.nationalgeographic.org/article/development-agriculture/12th-grade/>
- Yanet, L., Contreras, S., Luz, A., & Riaño, R. (2013). *Aislamiento de microorganismos para control biológico de Moniliophthora roreri Isolation of microorganisms for biological control of Moniliophthora roreri*.
- Agrotrading. (2016). *Tipos de Cacao*. http://agrotradingvenezuela.com.ve/cw_site/1/link_2.php?id=26&op=menu
- Ariandi, V., Kurnia, H., Heriyanto, & Marry, H. (2019). Expert system for disease diagnosis in cocoa plant using android-based forward chaining method. *Journal of Physics: Conference Series*, 1339(1). <https://doi.org/10.1088/1742-6596/1339/1/012009>
- Basri, Harli, Indrabayu, Areni, I. S., & Tamin, R. (2019). Image Processing System for Early Detection of Cocoa Fruit Pest Attack. *Journal of Physics: Conference Series*, 1244(1). <https://doi.org/10.1088/1742-6596/1244/1/012003>
- Bowers, J. H., Bailey, B. A., Hebbbar, P. K., Sanogo, S., & Lumsden, R. D. (2001). The Impact of Plant Diseases on World Chocolate Production. *Plant Health Progress*, 2(1), 12. <https://doi.org/10.1094/php-2001-0709-01-rv>
- Calderón, A., & Valverde, Rebaza, S. (2007). *Universidad Nacional de Trujillo Metodologías Ágiles*. 37. https://s3.amazonaws.com/academia.edu.documents/43784053/METODOLOGIAS_AGILES.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1555814033&Signature=21xq%2F4v4%2BMjp6HMPN7iEd5iZpVc%3D&response-content-disposition=inline%3B filename%3DUniversidad_Nacional_de_
- Castillo, J. A. (2019). *RGB qué es esto y para qué se utiliza en informática*. <https://www.profesionalreview.com/2019/01/20/rgb-que-es/>
- De Oliveira, J. R. C. P., & Romero, R. A. F. (2018). Transfer Learning Based Model for Classification of Cocoa Pods. *Proceedings of the International Joint Conference on Neural Networks, 2018-July*, 1–6.

- <https://doi.org/10.1109/IJCNN.2018.8489126>
- Diego Calvo. (2017, July 20). <http://www.diegocalvo.es/red-neuronal-convolucional/>
- Duarte, A. O., & C, M. R. (2008). Las Metodologías de Desarrollo Ágil como una Oportunidad para la Ingeniería del Software Educativo. *Revista Avances En Sistemas e Informática*, 5(2), 159–171.
- Figueroa, G. (2005). *Convolución y transformadas*.
<https://tecdigital.tec.ac.cr/revistamatematica/cursos-linea/EcuacionesDiferenciales/EDO-Geo/edo-cap5-geo/laplace/node7.html>
- Florez, O. U. (2018). *Diccionario Inglés-Español de Términos Técnicos en Inteligencia Artificial (IA)*. c, 1–20.
https://github.com/capitalone/AI_Dictionary_English_Spanish
- Folch, S. (2010). *Modos o modelos de color HSB (o HSV) y códigos hexadecimales: qué son y usos específicos*. <https://www.comunicacion-multimedia.info/2010/05/modos-o-modelos-de-color-hsb-o-hsv-y.html>
- Frucci, M., & di Baja, G. S. (2008). From Segmentation to Binarization of Gray-Level Images. *Journal of Pattern Recognition Research*, 3(1), 1–13.
<https://doi.org/10.13176/11.54>
- Fuentes, S., Chacon, G., Torrico, D. D., Zarate, A., & Viejo, C. G. (2019). Spatial variability of aroma profiles of cocoa trees obtained through computer vision and machine learning modelling: A cover photography and high spatial remote sensing application. *Sensors (Switzerland)*, 19(14).
<https://doi.org/10.3390/s19143054>
- Gamboa, A. A., Cáceres, P. A., Lamos, H., Zárata, D. A., & Puentes, D. E. (2019). Predictive model for cocoa yield in Santander using Supervised Machine Learning. *2019 22nd Symposium on Image, Signal Processing and Artificial Vision, STSIVA 2019 - Conference Proceedings*.
<https://doi.org/10.1109/STSIVA.2019.8730258>
- ICA. (2012). *Manejo fitosanitario del cultivo del cacao (Theobroma cacao L): medidas para la temporada invernal*.
- Jaimes, Y. Y., Ribeyre, F., Gonzalez, C., Rojas, J., Furtado, E. L., & Cilas, C. (2019). Factors affecting the dynamics of frosty pod rot in the main cocoa areas of Santander State, Colombia. *Plant Disease*, 103(7), 1665–1673.
<https://doi.org/10.1094/PDIS-10-18-1761-RE>
- Joshua Lanzarini. (2018, July 5). *Organic Farming - Person holding leafed plant photo – Free Food Image on Unsplash*.
<https://unsplash.com/photos/FGvQKMP-iXY>
- Lagos-Ortiz, K., Medina-Moreira, J., Alarcón-Salvatierra, A., Morán, M. F., del Cioppo-Morstadt, J., & Valencia-García, R. (2019). Decision Support System for the Control and Monitoring of Crops. In *Advances in Intelligent Systems and Computing* (Vol. 901, pp. 20–28). https://doi.org/10.1007/978-3-030-10728-4_3
- Lomotey, R. K., Mammay, A., & Orji, R. (2018). Mobile technology for smart agriculture: Deployment case for cocoa production. *International Journal of Sustainable Agricultural Management and Informatics*, 4(2), 83–97.

- <https://doi.org/10.1504/IJSAMI.2018.094810>
- Nembot, C., Takam Soh, P., ten Hoopen, G. M., & Dumont, Y. (2018). Modeling the temporal evolution of cocoa black pod rot disease caused by *Phytophthora megakarya*. *Mathematical Methods in the Applied Sciences*, 41(18), 8816–8843. <https://doi.org/10.1002/mma.5206>
- Putra, B. T. W., Soni, P., Marhaenanto, B., Pujiyanto, Sisbudi Harsono, S., & Fountas, S. (2018). Using information from images for plantation monitoring: A review of solutions for smallholders. *Information Processing in Agriculture*, 7(1), 109–119. <https://doi.org/10.1016/j.inpa.2019.04.005>
- RAE. (2014a). *agricultura* | Definición | Diccionario de la lengua española | RAE - ASALE. <https://dle.rae.es/agricultura>
- RAE. (2014b). *agro, agra* | Definición | Diccionario de la lengua española | RAE - ASALE. <https://dle.rae.es/agro?m=form>
- RAE. (2014c). *cultura* | Definición | Diccionario de la lengua española | RAE - ASALE. <https://dle.rae.es/cultura>
- Santander, C. (2019). *Clúster Santander - Cluster Cacao*. <http://clustersantander.com/bienvenido-cluster-cacao/>
- Seekpng. (n.d.). *Arbol Raices Borges - Tree And Roots Drawing PNG Image | Transparent PNG Free Download on SeekPNG*. Retrieved May 6, 2020, from https://www.seekpng.com/ipng/u2w7q8o0t4t4q8o0_arbol-raices-borges-tree-and-roots-drawing/
- Shobha, G., & Rangaswamy, S. (2018). Machine Learning. *Handbook of Statistics*, 38, 197–228. <https://doi.org/10.1016/BS.HOST.2018.07.004>
- Stanford, L. (2015). *Introduction to Computer Vision*. <https://ai.stanford.edu/~syeyung/cvweb/tutorial1.html>
- Tan, D. S., Leong, R. N., Laguna, A. F., Ngo, C. A., Lao, A., Amalin, D., & Alwindia, D. (2016). A framework for measuring infection level on cacao pods. *Proceedings - 2016 IEEE Region 10 Symposium, TENSymp 2016*, 384–389. <https://doi.org/10.1109/TENCONSpring.2016.7519437>
- Tan, D. S., Leong, R. N., Laguna, A. F., Ngo, C. A., Lao, A., Amalin, D. M., & Alwindia, D. G. (2018). AuToDiDAC: Automated Tool for Disease Detection and Assessment for Cacao Black Pod Rot. *Crop Protection*, 103(October 2017), 98–102. <https://doi.org/10.1016/j.cropro.2017.09.017>
- Tzutalin. (2015). *Labellmg*. In *Labellmg*. *Actualidad/tensorflow-dev-summit-2020--dias-11-y-12-de-marzo-en-en-sunnyvale-california*. (n.d.).
- Aime, M. C., & Phillips-Mora, W. (2005). The causal agents of witches' broom and frosty pod rot of cacao (*Theobroma cacao*) form a new lineage of Marasmiaceae. *Mycologia*, 97(5), 1012–1022. <https://doi.org/10.1080/15572536.2006.11832751>
- Basri, Harli, Indrabayu, Areni, I. S., & Tamin, R. (2019). Image Processing System for Early Detection of Cocoa Fruit Pest Attack. *Journal of Physics: Conference Series*, 1244(1). <https://doi.org/10.1088/1742-6596/1244/1/012003>
- Bochkovskiy, A. (2020). *YOLOv4 — the most accurate real-time neural network on MS COCO dataset*. | by Aleksey Bochkovskiy | Medium. 2020.

- <https://medium.com/@alexeyab84/yolov4-the-most-accurate-real-time-neural-network-on-ms-coco-dataset-73adfd3602fe>
- Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020a). *YOLOv4: Optimal Speed and Accuracy of Object Detection*. <http://arxiv.org/abs/2004.10934>
- Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020b). *YOLOv4: Optimal Speed and Accuracy of Object Detection*.
- Colaboratory. (2020). *Colaboratory – Google*.
<https://research.google.com/colaboratory/faq.html#resource-limits>
- Dias Rafa. (2016, January). *¿Qué es la tecnología CUDA?*
- Espinosa-García, J., ... J. U.-G.-R. mexicana, & 2015, U. (2015). Productividad y rentabilidad potencial del cacao (*Theobroma cacao* L.) en el trópico mexicano. *Scielo.Org.Mx*. http://www.scielo.org.mx/scielo.php?pid=S2007-09342015000500012&script=sci_arttext
- Evans, H. . (1981). *Pod rot of cacao caused by Moniliophthora (Monilia) roreri*.
<https://www.cabdirect.org/cabdirect/abstract/19811373782>
- Fedecacao. (2018). Federación Nacional De Cacaoteros Fondo Nacional Del Cacao. *On Line*, 53.
https://www.fedecacao.com.co/portal/images/MEMORIAS_V_SEMINARIO/001_Dr_Eduard_Baquero_-_Fedecacao.pdf
- Fedecacao. (2019). *Informe de ejecución plan de ingresos, inversiones y gastos del fondo nacional del cacao durante el cuarto trimestre y consolidado vigencia 2018*. 1–228.
https://www.fedecacao.com.co/portal/images/INFORME_2018.pdf
- Ghorpade, J. (2012). GPGPU Processing in CUDA Architecture. *Advanced Computing: An International Journal*, 3(1), 105–120.
<https://doi.org/10.5121/acij.2012.3109>
- Heredia Gómez, J. F., Rueda Gómez, J. P., & Ramírez Acuña, J. S. (2020). *APLICACIÓN PARA ESTIMAR EL NIVEL DE MADUREZ EN LAS MAZORCAS DE CACAO HACIENDO USO DE VISIÓN POR COMPUTADOR Y APRENDIZAJE DE MÁQUINA “DELECO”*. Universidad Autónoma de Bucaramanga.
- Holguín, C., Díaz-Ricardo, Y., & Antonio Becerra-García, R. (n.d.). *Ciencias Holguín, Revista trimestral, Año XX, abril-junio 2014 El lenguaje de programación Python/The programming language Python Ivet Challenger-Pérez*.
- Hunglc007. (2020). *GitHub - hunglc007/tensorflow-yolov4-tflite: YOLOv4, YOLOv4-tiny, YOLOv3, YOLOv3-tiny Implemented in Tensorflow 2.0, Android. Convert YOLO v4 .weights tensorflow, tensorrt and tflite*. 2020.
<https://github.com/hunglc007/tensorflow-yolov4-tflite#readme>
- Jaimés, Y., & , Aranzazu Fabio, R. E. y M. N. (2011). *Behavior of introduced regional clones of Theobroma cacao toward the infection Moniliophthora roreri in three different regions of Colombia*.
http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0120-99652011000200002
- Lee, J., Chirkov, N., Ignasheva, E., Pisarchyk, Y., Shieh, M., Riccardi, F., Sarokin,

- R., Kulik, A., & Grundmann, M. (2019). *On-Device Neural Net Inference with Mobile GPUs*. <http://arxiv.org/abs/1907.01989>
- Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8693 LNCS(PART 5), 740–755. https://doi.org/10.1007/978-3-319-10602-1_48
- Lutz Roeder. (2020). *Netron*. <https://netron.app/>
- McMahon P, & Purwantara A. (2004). Diversity and management of Phytophthora in Southeast Asia. *Diversity and Management of Phytophthora in Southeast Asia*.
- Polanco, E. R., Gicela, A., & Rodríguez Bióloga, V. (2015). *Identificación y manejo de la pudrición parda de la mazorca (Phytophthora sp.) en cacao*. www.corpoica.org.co
- Redmon, J., & Farhadi, A. (2018). *YOLOv3: An Incremental Improvement*.
- Seekpng. (2020). *Arbol Raices Borges - Tree And Roots Drawing PNG Image | Transparent PNG Free Download on SeekPNG*. https://www.seekpng.com/ipng/u2w7q8o0t4t4q8o0_arbol-raices-borges-tree-and-roots-drawing/
- Serrano, S., Torres, C., & Talero, L. (2020). *Cocoa Diseases (YOLOv4) | Kaggle*.
- Tan, D. S., Leong, R. N., Laguna, A. F., Ngo, C. A., Lao, A., Amalin, D., & Alwindia, D. (2016). A framework for measuring infection level on cacao pods. *Proceedings - 2016 IEEE Region 10 Symposium, TENSymp 2016*, 384–389. <https://doi.org/10.1109/TENCONSpring.2016.7519437>
- Tzutalin. (2015). *LabelImg*. In *LabelImg*.
- Valencia Hugo. (2019, February). *Entrenando redes neuronales: identificación de objetos*.

12. ANEXOS

Anexo A: Especificación casos de uso

Iniciar cámara – ECU 001

RF- <ECU 001>	Iniciar Cámara	
Versión	1.0 28/08/2020	
Autores	Camilo Andrés Torres - Juan Sebastian Serrano	
Usuario	Agricultor	
Precondición	Cualquier tipo de usuario debe ingresar a la aplicación.	
Descripción		
Secuencia normal	Pas o	Acción
	1	Inicia
	2	El usuario ingresa a la aplicación
	3	La aplicación muestra una pantalla de inicio con un botón para acceder a la cámara
	4	El usuario pulsa el botón para acceder a la cámara
	5	La aplicación ejecuta la detección a tiempo real mediante la cámara
Postcondición		
Excepciones	Pas o	Acción
	1	Excepción inicializando la cámara (paso 4)
	E1	El sistema informa de lo que ha sucedido: "La cámara no pudo ser inicializada".
Historial de Cambio		
Inclusión	El usuario debe ver en pantalla la detección a tiempo real obtenida por la aplicación de Tensorflow. (Paso 4)	
Extensión		
Urgencia	Inmediatamente	
Comentarios		

Tabla 2 Caso de uso para iniciar la cámara.

Fuente: Los autores

Detectar Mazorca – ECU 002

RF- <ECU 002>		Detectar Mazorca	
Versión	1.0 28/08/2020		
Autores	Camilo Andrés Torres - Juan Sebastian Serrano		
Usuario	Agricultor		
Precondición	Cualquier tipo de usuario debe ingresar a la aplicación.		
Descripción			
Secuencia normal	Pas o	Acción	
	1	Inicia	
	2	La aplicación ejecuta la detección a tiempo real mediante la cámara	
	3	La aplicación analiza a tiempo real los frames y compara con el modelo de YOLOv4	
	4	La aplicación al detectar el objeto mazorca muestra en pantalla un cuadro con la etiqueta asignada por medio del modelo de YOLOv4.	
Postcondición	El sistema redirige al tipo de usuario con respecto a la vista programada para él.		
Excepciones	Pas o	Acción	
		1	Excepción inicializando el clasificador
	E1	El sistema informa de lo que ha sucedido (paso 2): “El clasificador no pudo ser inicializado”.	
Historial de Cambio			
Inclusión			
Extensión	La aplicación TensorFlow debe poder conectarse con el modelo de YOLO v4 al iniciar el clasificador		

	pues el modelo contiene la información de la máquina entrenada. (Paso 3)
Urgencia	Inmediatamente
Comentarios	

Tabla 3 Caso de uso para detectar la mazorca.

Fuente: Los autores

Modelo YOLOv4 Entrenado – ECU 003

RF- <ECU 003>	Modelo YOLO v4 Entrenado	
Versión	1.0 28/08/2020	
Actores	Tipo de Usuario (Agricultor)	
Precondición	Cualquier tipo de usuario debe ingresar a la aplicación.	
Descripción		
Secuencia normal	Paso	Acción
	1	Inicia
	2	La clase pasa los parámetros
	3	Recibe una respuesta
Postcondición	El sistema redirige al tipo de usuario con respecto a la vista programada para él.	
Excepciones	Paso	Acción
Historial de Cambio		
Inclusión		
Extensión		
Urgencia	Inmediatamente	
Comentarios		

Tabla 4 Caso de uso para Modelo YOLO v4 Entrenado

Fuente: Los autores

Anexo B: Modelo entrenado con YOLOv4

A continuación, en la Figura 30 se puede ver el modelo final obtenido después de la conversión y previo entrenamiento. Se consiguió usando la herramienta Netron que es un visor de modelos de redes neuronales, aprendizaje profundo y aprendizaje automático (Lutz Roeder, 2020)

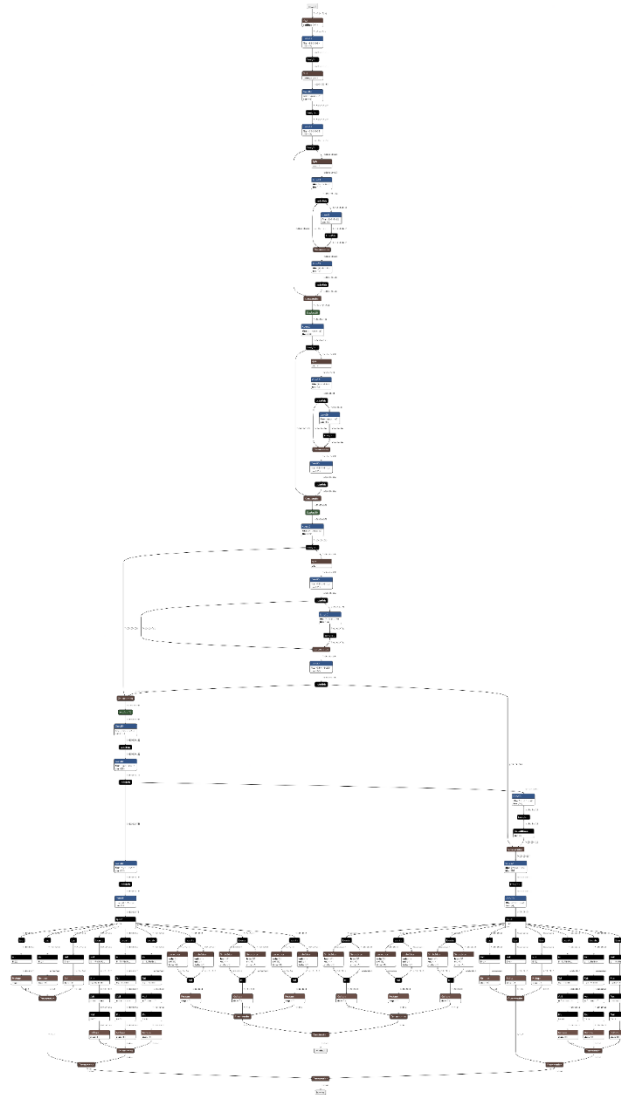


Figura 30 Modelo de la red neuronal de enfermedades del cacao Tflite.
Fuente: Adaptado con la herramienta (Lutz Roeder, 2020)