

**DESARROLLO DE UN PROTOTIPO FUNCIONAL DE SOFTWARE PARA
ESTIMAR LA PRODUCCIÓN DE CACAO, HACIENDO USO DE
HERRAMIENTAS DE APRENDIZAJE PROFUNDO Y VISIÓN POR
COMPUTADOR**

**KAROL DANIELA CALA PINZÓN
LISSETH ANDREA HERNÁNDEZ FLÓREZ
CRISTIAN DAVID PARRA MUÑOZ**

**UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA – UNAB
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
BUCARAMANGA
2021**

**DESARROLLO DE UN PROTOTIPO FUNCIONAL DE SOFTWARE PARA
ESTIMAR LA PRODUCCIÓN DE CACAO, HACIENDO USO DE
HERRAMIENTAS DE APRENDIZAJE PROFUNDO Y VISIÓN POR
COMPUTADOR**

**KAROL DANIELA CALA PINZÓN
LISSETH ANDREA HERNÁNDEZ FLÓREZ
CRISTIAN DAVID PARRA MUÑOZ**

**Proyecto de Trabajo de Grado para optar por el título de Ingeniero de
Sistemas**

**DIRECTOR:
Leonardo Hernán Talero Sarmiento**

**CODIRECTOR
Diana Teresa Parra Sánchez**

**ASESOR
Feisar Enrique Moreno Corzo**

**ASESOR
Néstor Edsgardo Nieves Peña**

**UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA – UNAB
FACULTAD DE INGENIERÍA
PROGRAMA DE INGENIERÍA DE SISTEMAS
BUCARAMANGA
2021**

AGRADECIMIENTOS

En primer lugar, queremos agradecerle a Dios por permitirnos llegar hasta este momento. A nuestro director Leonardo Talero, codirectora Diana Parra, asesor Feisar Moreno y al instructor del SENA Néstor Nieves por su amor a la investigación, por ser nuestros modelos a seguir como futuros profesionales y sobre todo por la paciencia y resistencia en el desarrollo de este proyecto de grado. A nuestros compañeros Juan Felipe Heredia, Sebastián Serrano y al grupo hermano Yeison Hernández y Juan Consuegra por compartir sus conocimientos, por hacernos reír en momentos de agobio y motivarnos durante este proceso. A nuestros familiares y amigos por apoyarnos y ser nuestros pilares en el desarrollo de este proyecto. En especial, queremos hacer mención a nuestros padres, que siempre estuvieron ahí para darnos palabras de apoyo y brindarnos sus consejos.

Muchas gracias a todos.

CONTENIDO

	pág.
1 INTRODUCCIÓN	13
2 PLANTEAMIENTO DEL PROBLEMA	14
3 OBJETIVOS	17
3.1 OBJETIVO GENERAL	17
3.2 OBJETIVOS ESPECÍFICOS	17
4 MARCO REFERENCIAL	18
4.1 MARCO CONCEPTUAL	18
4.1.1 Visión por Computador (Computer Vision)	18
4.1.2 Theobroma Cacao	18
4.1.3 Producción de cacao	18
4.1.4 Red Neuronal (Neural Network)	18
4.2 MARCO TEÓRICO	18
4.2.1 Periodo De Producción (Cacao)	18
4.2.2 Variedades De Cacao	19
4.2.3 Rendimiento De Cacao.	21
4.2.4 Espacios De Color	21
4.2.5 Métodos de preprocesamiento de imágenes	23
4.2.6 Representación De Imágenes Digitales	25
4.2.7 Operaciones Matriciales Y Vectoriales	26
4.2.8 Borde	26

4.2.9	Inteligencia Artificial	26
4.2.10	Data Augmentation	27
4.2.11	Aprendizaje Automático	30
4.2.12	Redes Neuronales	30
4.2.13	Aprendizaje Profundo	32
4.2.14	Redes Neuronales Convolucionales	32
4.2.15	YOLO	33
4.2.16	Colab	34
4.2.17	Android Studio	34
4.2.18	Labelimg	35
5	METODOLOGÍA	36
6	ANÁLISIS DEL ESTADO DEL ARTE	38
6.1	ANÁLISIS DETALLADO DEL ESTADO DEL ARTE	49
7	ADAPTACIÓN DEL MODELO DE APRENDIZAJE	53
7.1	ADQUISICIÓN DE DATOS	53
7.2	ANÁLISIS Y SELECCIÓN DEL CONJUNTO DE DATOS PARA EL ENTRENAMIENTO	55
7.3	LIMPIEZA Y ETIQUETADO	57
7.4	ELECCIÓN DEL MODELO DE APRENDIZAJE MÁQUINA	58
7.5	DESCRIPCIÓN DE YOLOv4.....	60
7.6	ENTRENAMIENTO.....	62
7.7	DEPENDENCIAS DEL ENTRENAMIENTO	64
7.8	CONFIGURACIÓN DEL ENTORNO DE EJECUCIÓN	65

7.9	CARGA Y PREPARACIÓN DEL CONJUNTO DE IMÁGENES.....	66
7.10	CONFIGURACIÓN DE LA RED NEURONAL.....	67
7.11	PESOS PREENTRENADOS.....	68
7.12	EJECUCIÓN DEL ENTRENAMIENTO.....	68
7.13	EVALUACIÓN DEL MODELO	69
7.14	CONVERSIÓN.....	71
7.15	ADAPTACIÓN DEL MODELO	72
8	DISEÑO Y DESARROLLO DEL PROTOTIPO	73
8.1	RECOLECCIÓN DE REQUISITOS	73
8.1.1	Requerimientos funcionales	73
8.1.2	Requerimientos no funcionales	74
8.2	DIAGRAMA DE COMPONENTES	74
8.3	DIAGRAMA DE CASOS DE USO.....	75
8.4	DIAGRAMA DE SECUENCIA	76
8.5	DISEÑO	77
8.6	DESARROLLO DEL PROTOTIPO.....	78
8.7	EVALUACIÓN DEL PROTOTIPO	78
9	DISCUSIÓN	79
10	CONCLUSIONES	81
11	REFERENCIAS	83
	ANEXOS	90

LISTADO DE TABLAS

	pág.
Tabla 1. Espacios de color	21
Tabla 2. Métodos fundamentales en el preprocesamiento de imágenes con sus respectivas operaciones.....	24
Tabla 3. Técnicas de data augmentation.	27
Tabla 4. Arquitectura de capas de redes neuronales.	32
Tabla 5. Criterios de búsqueda.....	39
Tabla 6. Elementos destacados de los de documentos recuperados.....	46
Tabla 7. Métricas para estimar la producción agrícola.....	51
Tabla 8. Conjuntos de datos resultados de la búsqueda.	53
Tabla 9. Características de los dispositivos usados en la toma de datos.	54
Tabla 10. Análisis de los conjuntos de datos.....	55
Tabla 11. Análisis de los conjuntos de datos por enfoques.	56
Tabla 12. Comparativa de los modelos más destacados en regresión y clasificación, evaluados con el dataset MS COCO.	59
Tabla 13. Descripción de los experimentos.....	63
Tabla 14. Resultados de los experimentos.....	69
Tabla 15. Análisis de varianza ANOVA	69

LISTADO DE FIGURAS

	pág.
Figura 1. Variedades de cacao.....	20
Figura 2. Variedades de cacao usadas en el proyecto.	20
Figura 3. El espacio de coordenadas cartesianas 2D de una imagen digital M x N	26
Figura 4. Estructura de capas presentes en una red neuronal.....	31
Figura 5. Arquitectura típica de una red neuronal convolucional.	33
Figura 6. Gráfica de documentos por año que cumplen con la ecuación de búsqueda establecida.	38
Figura 7. Características principales en la toma de datos presencial.	55
Figura 8. Muestra de etiquetado.	58
Figura 9. Representación gráfica de la arquitectura de YOLOv4.	60
Figura 10. Comparativa YOLOv4-Tiny.....	61
Figura 11. Flujo de trabajo en el entrenamiento de datos.....	64
Figura 12. Configuración del entorno de ejecución. Fuente: Los autores.....	65
Figura 13. Ejemplo de una imagen con data augmentation.....	66
Figura 14. Gráfico de efectos principales para mAP.	70
Figura 15. Modelo adaptado en Android Studio.....	72
Figura 16. Diagrama de componentes.	75
Figura 17. Diseño inicial.....	78

LISTA DE ANEXOS

	pág.
Anexo A. Matriz de casos de uso	90
Anexo B. Tabla de riesgos	91
Anexo C. Especificación de casos de uso	92
Anexo D. Diseño	101
Anexo E. Storyboarding	105
Anexo F. Pruebas de uso.	106

GLOSARIO

- ANN: Artificial Neural Network (Red Neuronal Artificial).
- AP: Average *Precision*.
- ASPP: Atrous Spatial Pyramid Pooling.
- CIELab: Espacio de color LAB.
- CNN: Convolutional Neural Network (Red Neuronal Convolutacional).
- CSP: Cross-stage partial connections (Conexiones parciales entre etapas).
- cuDNN: Librería CUDA Deep Neural Network proporcionada por NVIDIA.
- Detection Rate: Tasa de detección.
- F1 Score: Puntaje F1.
- GPS: Global Positioning System (Sistema de posicionamiento global).
- GPU: Graphics Processing Unit (Unidad de procesamiento gráfico).
- HSV: Modelo de color Hue, Saturation, Value. (Matiz, Saturación, Valor).
- IoU: intersección sobre la unión, métrica usada en detección de objetos.
- K-Means: K-Medias.
- *mAP*: Mean Average *Precision* (Media aritmética de precisión promedio).
- MiWRC: Multi-input Weighted Residual Connections (Conexiones residuales ponderadas de múltiples entradas).
- MLP: Multiscale Multilayer Perceptrons (Perceptrón multicapa).
- MSE: Mean Square Error (Error cuadrático medio).
- PAN (Path Aggregation Network).
- R-CNN: Region-based Convolutional Neural Network (Red neuronal convolutacional basado en regiones).
- ReLu: Rectified Linear Unit (Función de activación de unidad lineal rectificadora).
- RGB: Modelo de color Red, Green, Blue (Rojo, Verde y Azul).
- SSD: Single Shot Detector (Detector de disparo único).
- SVM: Support Vector Machine (Máquinas de vectores de soporte).
- TDR: True Detected Rate (Tasa real detectada).
- YOLO: You Only Look Once es un modelo de detección de objetos.

RESUMEN

Este proyecto, presenta el diseño y desarrollo de una aplicación móvil funcional capaz de estimar la producción de cacao, que propone la implementación de técnicas de visión por computador y aprendizaje profundo. Esto se debe a que la detección de objetos en la agricultura es importante para estimar la producción de un cultivo, porque incrementa la certeza en la toma de decisiones por parte de un agricultor, por consiguiente, el diseño propuesto realiza un conteo de las mazorcas de cacao que se encuentran en tres estados de sanidad, ya sea con presencia de monilia, fitóftora o completamente sanas. La aplicación planteada hace uso de una cámara de un dispositivo móvil y el sistema operativo Android. Los elementos presentes en el sistema, son un modelo de aprendizaje de máquina entrenado, un conjunto de datos, y tecnologías que apoyan el proceso de desarrollo de software. En primera instancia, se realiza una revisión de la literatura para profundizar sobre las técnicas, tecnologías, y métricas asociadas con visión artificial y que puedan ser aplicadas en el proyecto. Luego, se propone la selección de un conjunto de imágenes con *Theobroma cacao*. Asimismo, se plantea la adaptación de un modelo de aprendizaje profundo con una definición de parámetros e hiper parámetros, para posteriormente proponer un diseño y desarrollo de un prototipo móvil que detecta, clasifica y localiza las mazorcas de cacao con sus respectivos estados de sanidad, y a su vez estima la producción en términos de kilogramos de granos de cacao seco, teniendo en cuenta la variedad indicada por el usuario. Los resultados obtenidos dejan la evaluación de 8 modelos, en donde el mejor obtiene una mAP de 80.09% y se determina la incidencia de variables asociadas al balanceo sobre la precisión.

ABSTRACT

This project presents the design and development of a functional mobile application capable to estimate cocoa production based on the implementation of computer vision and deep learning techniques. Object detection in agriculture is important to estimate production in a crop because it increases the confidence in decision making by a farmer, therefore, the proposed design performs a count of cocoa pods that are in three sanitary states, either with the presence of monilia, phytophthora or completely healthy. The following application implements an Android mobile device camera. The elements existing in the system contain an object detection model, a dataset, and technologies that support the software development process. A literature review is done to explore techniques, technologies, and metrics associated with computer vision. Subsequently, a selection of an image dataset is done to train a deep learning model, setting up parameters and hyperparameters. Consequently, a design and development of a mobile prototype are proposed to detect, classify, and localize cocoa pods with their respective health status, and it returns the estimated value of production given in kilograms of dry cocoa beans, taking into account the variety indicated by the user. The results show the evaluation of 8 models, where the best one obtains a mAP of 80.09% and the incidence of variables associated with the balancing on the accuracy is determined.

1 INTRODUCCIÓN

La producción de cacao se ha efectuado manualmente, alejándose de herramientas tecnológicas que promuevan una mecanización generalizada (Voorra et. al, 2019). A nivel mundial, para 2025 se espera que los granos de cacao exportados alcancen los \$16,32 billones de dólares (Voorra et. al, 2019). Sin embargo, también se estima una escasez de la oferta a largo plazo, debido a la disminución de los rendimientos y precios que terminarían desanimando a los jóvenes trabajadores del sector (Gayi & Tsowou, 2017). A este se le debe añadir que los cacaocultores no siempre cuentan con los suficientes activos agrícolas, y deben lidiar con mercados poco transparentes, lo cual afecta a sus ingresos y calidad de vida. Es por ello por lo que, la estimación de la producción de un cultivo es una actividad de gran importancia para el agricultor, ya que, esta le permite conocer la cantidad de frutos, los recursos necesarios para mantener el cultivo y la productividad aproximada; sin embargo, es una actividad que consume demasiado tiempo y recursos que pueden ser destinados a otras tareas. Una de las vías para conseguir lo anterior mencionado radica en el avance de técnicas de visión por computador, el reconocimiento de objetos y el aprendizaje automático. Dichos avances han tenido un gran impacto en la agricultura, especialmente en la detección de frutos para detectar estado de madurez, enfermedades y en la automatización de actividades productivas. Dados en este punto, el presente documento muestra el desarrollo de un prototipo funcional de software para estimar la producción de cacao, haciendo uso de herramientas de aprendizaje profundo y visión por computador.

2 PLANTEAMIENTO DEL PROBLEMA

En el panorama global, el cacao, muestra ser uno de los cultivos más importantes para la economía, porque genera exportaciones, ingresos, y empleo que beneficia a muchas familias agricultoras (Gayi & Tsowou, 2017). Sin embargo, de acuerdo con *Cocoa Barometer* (2015) los cacaoteros reciben sólo el 6,6% del valor total agregado de una 1 tonelada de granos de cacao vendidos (Fountain & Hütz-Adams, 2015), esto hace que muchos cacaoteros vivan por debajo de la línea de pobreza, tendencia que se ha mantenido en los últimos años (*Cocoa Barometer*, 2018) (*Cocoa Barometer*, 2020). Los bajos ingresos traen como consecuencia una falta de interés en la industria del cacao, tal como ocurre con la población joven de agricultores que, prefieren optar por cultivos que generen mayores ingresos (ILRF, 2014). La labor de los pequeños agricultores es primordial en el mercado, gracias a ellos se aporta entre el 80% y 90% de la producción global de cacao (Gayi & Tsowou, 2017). Adicionalmente, su trabajo involucra tareas complicadas como: cultivar, cosechar, fermentar y secar granos de cacao. Aun así, los pequeños agricultores se enfrentan a diversos retos como la baja productividad, falta de transparencia en los mercados, una brecha en el acceso a las tecnologías de la información y la comunicación, y una carencia de ayudas financieras (Gayi & Tsowou, 2017). Estos retos hacen que los cacaocultores tengan dificultad al momento de establecer precios justos y, por lo tanto, deben limitarse sólo a la información que brindan los mercados.

Dentro de las variables que afectan los ingresos de los cacaoteros se encuentra el rendimiento por hectárea, el tamaño de las tierras, y el precio de sus cosechas. La estimación del rendimiento es importante en la agricultura, ya que permite identificar las necesidades y recursos que estarán presentes en el proceso de producción. Patel et al. (2013) afirmó que el conteo manual agrícola generó costos entre \$300,000 y \$500,000 dólares por año. Dichos conteos a pesar de que siguen vigentes se convierten en una técnica falible, debido a que es una actividad repetitiva ejercida por un humano durante horas. Por lo que, si se sobreestima la producción agrícola, se cae en el error derrochar dinero en servicios de transporte y distribución. Mientras que, si se subestima la cantidad agrícola, implicaría un déficit en la cantidad de trabajadores presentes en el proceso de producción, que a su vez conlleva a que no se satisfagan las necesidades del mercado. La subestimación puede traer en últimas como consecuencia una baja en los precios de los productos agrícolas. Por consiguiente, saber estimar el rendimiento de los cultivos de cacao puede ayudar a que los ingresos de los cacaoteros aumenten a la vez que incrementan la productividad. Sin embargo, se debe dejar en claro que un aumento del rendimiento no es una panacea que resolverá todos los problemas de los cacaoteros presentados en el contexto global, es necesario tener en cuenta los roles que ejercen los gobiernos y las empresas al momento de querer solventar algo tan complejo (*Cocoa Barometer*, 2018).

Desde otro punto de vista, Colombia a través de los años, ha demostrado ser un destacado productor de cacao fino y de aroma. Todo esto se debe a que, Colombia ha estado en la continua mejora de aspectos, tales como la calidad y productividad (Colombia Cacaotera, 2019); resultados esperados por el Plan Nacional de Desarrollo 2018 - 2022, que propone distintos objetivos, metas y estrategias que contemplan el progreso del país en distintas áreas, entre estas, se encuentra el apoyo a 550 mil productores con asistencia técnica agro y 300 mil con agricultura por contrato. Al igual que la erradicación de 280 mil hectáreas de cultivos ilícitos (Plan Nacional de Desarrollo, 2018), en el que se propone cultivar cacao para dar un nuevo uso a muchos de estos terrenos en varios sectores. Por otra parte, se plantea la participación en la Organización Internacional del Cacao (Redacción Economía, 2020), de manera responsable y proactiva, por medio del fortalecimiento de los productores, para que logren estándares de sanidad, inocuidad, y calidad (Plan Nacional de Desarrollo, 2018). No obstante, a pesar de que dichas iniciativas han promovido el aumento de la producción de cacao, el rendimiento por hectárea se ha mostrado bajo debido a factores como la incidencia de plagas y enfermedades, deficientes prácticas agrícolas, incompatibilidad genética, plantaciones envejecidas, cambio climático, y presencia de cadmio (metal con un alto nivel de toxicidad) (Sánchez, et al., 2019).

Algo excepcional de las plantaciones de cacao colombianas es que crecen bajo distintas condiciones climáticas y fisiográficas, en donde se destaca el departamento Santander (Sánchez, et al., 2019), reportando 59665 toneladas en el año 2019 (PDD-Santander, 2020). A nivel departamental, siguiendo los objetivos de desarrollo sostenible la Gobernación de Santander plantea la modernización tecnológica de trabajos para apoyar el objetivo de desarrollo sostenible de trabajo decente. De esta manera, con ayuda de la tecnología, se propone impulsar emprendimientos verdes que además de tener impacto ambiental positivo, generan economías cíclicas y buenas prácticas económicas. Esta tendencia se ratifica por el incremento en el enfoque de servicios de consumo y comida saludable. En donde, uno de los puntos del plan departamental de desarrollo es la iniciativa *Diamante Caribe y Santanderes*, que busca crear territorios inteligentes; uno de los objetivos de esta alianza es “Desarrollar nuevas oportunidades en el acceso a tecnologías fundamentales para la agricultura moderna con aplicaciones y plataformas tecnológicas aplicadas al sector” (PDD-Santander, 2020). No obstante, a pesar de que Santander es un departamento destacado en producción, enfrenta retos como la baja densidad de siembra, bajo rendimiento por hectárea, problemas fitosanitarios y un 25% de plantaciones envejecidas. (PDD-Santander, 2020).

La industria del cacao vista desde diferentes perspectivas enfrenta múltiples retos que terminan afectando la toma de decisiones, y se traducen en dificultades al momento de proyectar los cultivos a largo plazo en términos de objetivos e ingresos (Taher, 1996). Para ello, quienes trabajan en estos cultivos deben establecer estrategias para mantener la productividad, evitar las plagas, y prevenir las enfermedades. Esta variedad de responsabilidades y desventajas identificadas en

el contexto global, nacional y departamental, son el motivo de inspiración para desarrollar futuros trabajos. Sin embargo; cabe mencionar que el presente trabajo tiene como objetivo desarrollar un prototipo funcional de software, para estimar la producción de cacao, haciendo uso de herramientas de aprendizaje profundo y visión por computador.

Los algoritmos de visión por computador permiten obtener información sobre algunos parámetros físicos y químicos en los cultivos de cacao (Mite-Baidal et al., 2019). Obtener datos sobre los parámetros físicos ayuda a que se tenga más información del estado y la cantidad de los frutos. En vista de que tradicionalmente se siguen realizando conteos manuales que conllevan a una sobreestimación o subestimación de la producción agrícola, es una oportunidad para poner a prueba el potencial de la visión por computador para estimar la producción bajo condiciones ambientales no controladas. Investigaciones para contar o estimar la producción se han llevado a cabo en los últimos años, sin embargo, algunas de ellas son dependientes de sensores costosos o computadores (Gené-Mola et al., 2020) (Wang et al., 2013) (Koirala et al., 2019), mientras que otras dependen de la conectividad (Deng et al., 2020) (Ali Süzen et al., 2020). Por otra parte, hasta el momento, se han identificado pocos trabajos en la literatura para contar o estimar la producción de un cultivo utilizando un teléfono inteligente. Es por ello, que este proyecto es un paso para investigar y desarrollar un prototipo móvil que implemente un modelo de aprendizaje de máquina que será calibrado para lograr una labor de detección y conteo.

Lo anterior da origen a la siguiente pregunta de investigación *¿De qué manera se podría estimar la producción de Theobroma Cacao haciendo uso de herramientas de aprendizaje profundo y visión por computador?*

3 OBJETIVOS

3.1 OBJETIVO GENERAL

Desarrollar un prototipo funcional de software, para estimar la producción de cacao, haciendo uso de herramientas de aprendizaje profundo y visión por computador.

3.2 OBJETIVOS ESPECÍFICOS

- Analizar los principales elementos relacionados con la aplicación de técnicas de visión por computador para la medición y estimación de la producción agrícola.
- Adaptar un modelo de aprendizaje de máquina para la detección y conteo de mazorcas de cacao.
- Diseñar un prototipo móvil funcional que integre un algoritmo de aprendizaje de máquina para la identificación de mazorcas de cacao.

4 MARCO REFERENCIAL

4.1 MARCO CONCEPTUAL

El marco conceptual se desarrolla teniendo en cuenta los siguientes descriptores: *Computer Vision, Production, Theobroma Cacao, Neural Network*.

4.1.1 Visión por Computador (Computer Vision)

“La visión por computador es la habilidad [que tienen las máquinas] de percibir información visual, extraer datos de ella y analizarlos de manera similar a un cerebro humano” (Bhardwaj et al., 2019).

4.1.2 Theobroma Cacao

Theobroma cacao es una especie frutal de árbol diploide endémica de las selvas tropicales de América del Sur (Argout et al., 2011). Este se cultiva en los trópicos húmedos y es una fuente importante de moneda para los pequeños agricultores. Sus frutos (mazorcas) contienen las semillas (granos) que luego son procesadas por la industria del chocolate (Motamayor et al., 2008).

4.1.3 Producción de cacao

El término producción en el contexto de *Theobroma Cacao*, hace referencia al número de toneladas de granos de cacao que se generan en un determinado año (De la Cruz Medina et al., 2011).

4.1.4 Red Neuronal (Neural Network)

“Una red neuronal es una máquina que está construida para funcionar de manera similar a las operaciones del cerebro en la realización de una tarea o funcionamiento en particular” (Bhardwaj et al., 2019).

4.2 MARCO TEÓRICO

4.2.1 Periodo De Producción (Cacao)

Para comprender el periodo de producción de cacao, primero se deben plantear algunas consideraciones. Se estima que el tiempo de fecundación es de 36 horas.

De acuerdo a la variedad, el periodo entre la fertilización y la madurez de la mazorca comprende una duración entre 150 a 180 días (De La Cruz Medina et al., 2011). Por otra parte, la cosecha se realiza en un periodo entre 10 y 14 días. Los árboles cacao suelen tener dos periodos de fructificación, que dependen del clima y la variedad. Posterior a la cosecha, los frutos se almacenan durante máximo 4 días y se procede a destaparlos para empezar el proceso de fermentación de los granos. La fermentación tarda entre 5 y 7 días, pero cabe aclarar que depende de algunas variables ambientales como la temperatura. Finalmente, los granos de cacao son secados en un periodo variable entre 7 y 14 días, sujeto a condiciones climáticas y tecnológicas (Dostert et al., 2011).

4.2.2 Variedades De Cacao

Las variedades de cacao cuentan con particularidades en su sabor que se llegan a relacionar con las condiciones climáticas. Principalmente, se reportan tres variedades de cacao: Criollo, Forastero, y Trinitario (De La Cruz Medina et al., 2011). Sin embargo, algunas fuentes consideran que los híbridos crean nuevas variedades de cacao, y que la tradicional clasificación de Criollo, Forastero y Trinitario no representa todas las variedades existentes de la especie (Dostert et al., 2011). La clasificación de las variedades no es algo que preocupe a los agricultores. Dentro de las razones se tiene que, una gran parte de los cultivos son producto de una mezcla genética y, por lo tanto, hace difícil crear una clasificación objetiva de las variedades. Con base en lo anterior, sabiendo que es algo que no preocupa a los cacaoteros y con el objetivo de tener un punto de referencia claro, se pretende presentar a continuación las tres variedades comúnmente reportadas.

El cacao Criollo presenta mazorcas con cubiertas surcadas y delgadas; suelen tomar colores como el amarillo y rojo (o verde y blanco en el caso del Porcelana); sus granos son redondos y llenos, con colores que varían de púrpura claro a blanco. El Forastero muestra mazorcas con una cubierta relativamente lisa, bulbosa y dura, con colores como: rojo, amarillo, naranja y púrpura; sus granos son planos y púrpura oscuro. El trinitario es un híbrido entre las dos variedades ya mencionadas, con una genética heterogénea que complica su caracterización. Sin embargo, este último presenta algunos atributos que suelen ser comunes; por ejemplo, mazorcas que exhiben una cubierta relativamente suave, con granos planos y púrpura en el interior.



Figura 1. Variedades de cacao.

Fuente: Tomado de *CACAO: Operaciones Poscosecha* (De La Cruz Medina et al., 2011). *Hoja Botánica: Cacao*. (Dostert et al., 2011)

Las variedades con las que se cuenta para la realización del proyecto son las siguientes:

FSV41, FEC2, FEAR5 y CNN51.

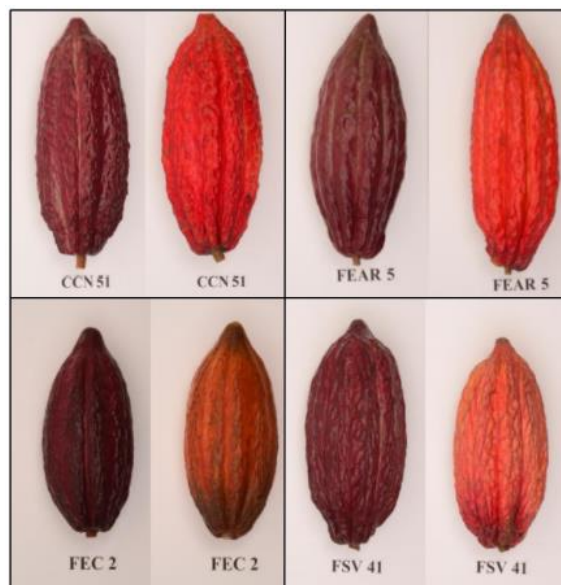


Figura 2. Variedades de cacao usadas en el proyecto.

Fuente: Adaptado de Aportes de la investigación de FEDECACAO –Fondo Nacional del Cacao al incremento de la productividad y reconocimiento de la calidad del cacao de Colombia (Martínez Guerrero, 2016)

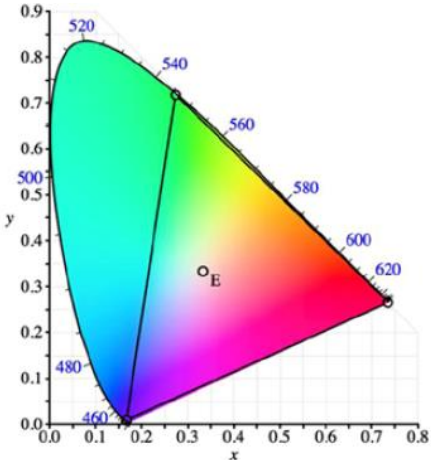
4.2.3 Rendimiento De Cacao.

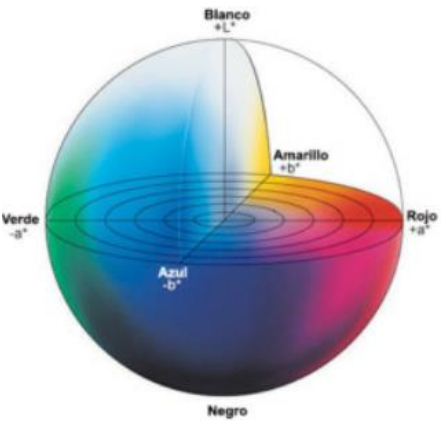
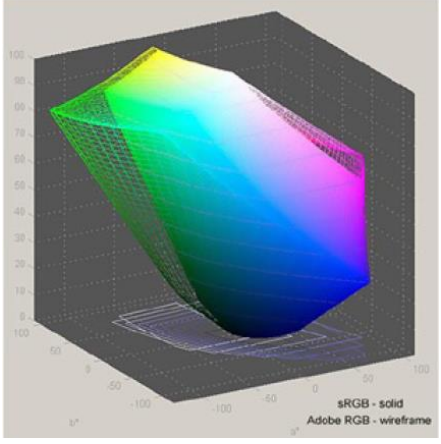
Corresponde a la masa que se da en una determinada área de un cultivo. En el caso del cacao, el rendimiento suele especificarse en hg (hectogramos) de granos de cacao/ ha(hectárea). En 2018, Colombia registró un valor de 3626 hg de granos de cacao/ha (De La Cruz Medina et al., 2011).

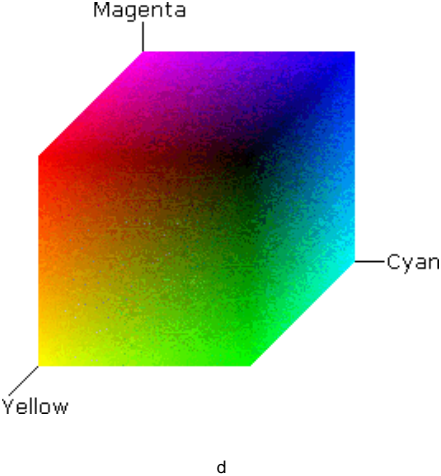
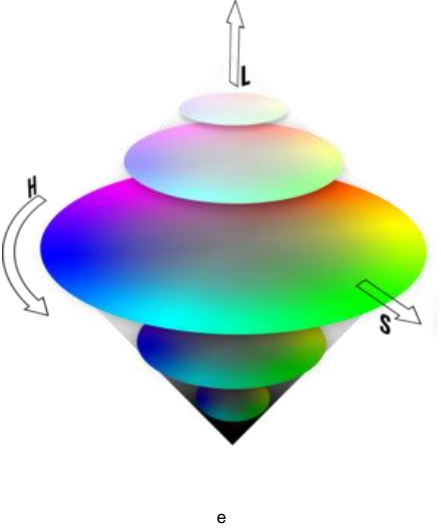
4.2.4 Espacios De Color

Los espacios, desde el punto de vista del procesamiento de imágenes, hacen referencia a la representación del color en relación con variables como la luminosidad, la saturación, y el tono. Lo cual sirve para hacer diferentes tipos de análisis y procesamiento en imágenes digitales (Krig, 2014). Sin embargo, definir un espacio de color no es tarea sencilla, debido a que las propiedades físicas de los dispositivos se pueden relacionar, dejando como resultado una subjetividad al momento de tratar espacios de color. A esto se le suma que el estudio de los colores, también conocido como ciencia del color, abarca muchas áreas de aplicación, con múltiples modelos, estándares, y conocimientos que pueden ser extensos de tratar (Burger & Burger, 2016). A continuación, se presentan algunos espacios de color representativos y usados en el procesamiento de imágenes.

Tabla 1. Espacios de color

Espacio de color	Dependiente de Hardware (DH)	Objetivo	Representación gráfica
CIE XYZ	No	Identificar tres colores primarios imaginarios X, Y, Z, de manera que todos los colores visibles puedan describirse como una suma de componentes positivos.	

			a
CIE LAB	No	Linealizar la representación con respecto a la percepción humana del color y simultáneamente crear un sistema de color más intuitivo.	 <p>Diagrama de un espacio de color esférico. El eje vertical representa el brillo, con 'Blanco +L*' en la parte superior y 'Negro' en la inferior. El eje horizontal a la izquierda es 'Verde -a*' y a la derecha es 'Rojo +a*'. El eje de profundidad es 'Amarillo +b*' y 'Azul -b*'. El interior del espacio está coloreado con un gradiente de colores.</p>
sRGB	Sí	Crear un espacio de color especificado con precisión, basado en asignaciones estandarizadas con respecto al espacio de color CIE XYZ	 <p>Diagrama 3D de un espacio de color. El eje vertical es 'L*' (0 a 100), el eje horizontal es 'a*' (-100 a 100) y el eje de profundidad es 'b*' (-100 a 100). Se muestra una superficie de color sólida y una versión de alambre. El texto 'sRGB - solid' y 'Adobe RGB - wireframe' está visible en la parte inferior derecha.</p>

CMY	Sí	Representar espacios de color a partir de los colores secundarios de la luz: cian, el magenta y el amarillo.	
HSI	Sí	Desacoplar el componente de intensidad de la información que lleva el color (tono y saturación) en una imagen en color.	

Fuente: Tomado de Digital image processing: an algorithmic introduction using Java (Burger & Burger, 2016), ^{a,c} *Computer Vision Metrics: Survey, Taxonomy, and Analysis* (Krig, 2014) ^b (Konica Minolta, 2020) ^d(White & Kennedy, 2018) ^e (*Colorizer*, 2020) Comparing deep-learning networks for apple fruit detection to classical hard-coded algorithms (Gonzalez & Woods, 2018)

4.2.5 Métodos de preprocesamiento de imágenes



Hace referencia al conjunto de métodos que permite una detección de características locales y globales de imágenes. Krig (2014) afirma que, “el pre-procesamiento de imágenes es análogo a la normalización matemática de un conjunto de datos”. De igual manera, menciona que el pre-procesamiento de

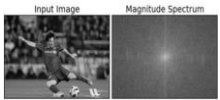

imágenes suele ser ignorado por algunos profesionales de la visión por computador, porque según ellos, se cambia la naturaleza propia de los datos. Sin embargo, el autor afirma que, un uso inteligente de dicho preprocesamiento puede dejar como resultado soluciones y beneficios que mejoran la extracción de características en las imágenes.

Krig (2014) propone cuatro métodos fundamentales para la descripción de características: Descriptores binarios locales, de espectro, de espacio básico, y de forma poligonal. A su vez, cada uno de estos métodos puede llegar a relacionarse con alguna de las siguientes Operaciones de preprocesamiento de imágenes: 1. Correcciones de iluminación; 2. Correcciones de enfoque y desenfoco; 3. Filtrado y eliminación de ruido; 4. Umbral; 5. Mejoras en los bordes; 6. Morfología; 7. Segmentación; 8. Procesamiento y filtros de región; 9. Procesamiento de puntos; 10. Procesamiento estadístico y matemático; 11. Conversiones de espacio de color.

A continuación, se presenta con más detalles los métodos y su relación las operaciones de preprocesamiento:

Tabla 2. Métodos fundamentales en el preprocesamiento de imágenes con sus respectivas operaciones.

Método	Descripción	Operaciones	Ejemplo	
Descriptores binarios locales	Se ocupa principalmente de las comparaciones de valores de intensidad de pares de puntos.	1,2,3	<i>LBP, ORB, FREAK</i>	 <p>(a) (b)</p> <p>a</p>
Descriptores de espectros	Utilizan una amplia gama de valores espectrales, como gradientes y promedios regionales.	1,2,3,5,8,9,10,11	<i>SIFT, SURF</i>	 <p>b</p>

<p>Descriptores de espacio básico</p>	<p>Estos métodos codifican un vector de características en un conjunto de funciones básicas, como la conocida serie de Fourier de magnitud y fase de seno y coseno.</p>	<p>1,2,3,8,11</p>	<p><i>FFT, Wavelets</i></p>	 <p>c</p>
<p>Descriptores de forma poligonal</p>	<p>Toman la forma de objetos medidos por métricas estadísticas, como área, perímetro y centroide. Las formas se extraen utilizando un canal de visión morfológica y algoritmos regionales.</p>	<p>1,2,3,4,5,6,7,8,9,10,11</p>	<p><i>Blob object area, perimeter, centroid</i></p>	 <p>d</p>

Fuente: Adaptado de ^a Local Derivative Pattern Versus Local Binary Pattern: Face Recognition With High-Order Local Pattern Descriptor (Zhang et al., 2010) ^b Introduction to SURF (Speeded-Up Robust Features).(OpenCV, 2015) ^c Fourier Transform (Mordvintsev & Abid K, 2013) ^d Computer Vision Metrics: Survey, Taxonomy, and Analysis (Krig, 2014)

4.2.6 Representación De Imágenes Digitales

Una imagen de dos dimensiones $I(m,n)$ representa valores en una serie de posiciones $(m = 1, 2, \dots, M; n = 1, 2, \dots, N)$ en un plano cartesiano de dos dimensiones. Los índices m y n representan las filas y las columnas de la imagen respectivamente y los elementos individuales de una imagen o píxeles se representan (m,n) de acuerdo a su índice en el plano cartesiano. Esto significa que la convención $I(m,n)$ denota un píxel ubicado en la m -ésima fila y n -ésima columna a partir de un origen de imagen superior izquierdo (Solomon & Breckon, 2011).



Figura 3. El espacio de coordenadas cartesianas 2D de una imagen digital $M \times N$
 Fuente: Adaptado de Fundamentals of Digital Image Processing (Solomon & Breckon, 2011)

4.2.7 Operaciones Matriciales Y Vectoriales

Una imagen puede ser vista como el equivalente a una matriz y mucha de la teoría matricial se aplica a las imágenes. Las operaciones entre imágenes se realizan entre sus respectivos píxeles (Gonzalez & Woods, 2018). Generalmente se realizan con Elementwise y las reglas de operaciones matriciales.

4.2.8 Borde

Cuando un conjunto de píxeles de una región R es cercano a un complemento de R , es denominado borde. Existen bordes internos y externos, los bordes externos corresponden al fondo de la imagen. Si R es una imagen completa, entonces se le llama borde a las primeras y últimas filas y columnas de la imagen. Comúnmente, la referencia a una región es una referencia a un subconjunto de una imagen. (Gonzalez & Woods, 2018)

4.2.9 Inteligencia Artificial



La inteligencia artificial es un campo de estudio que puede tener diferentes definiciones, esto dependiendo de la disciplina. Para los científicos de la computación, inteligencia artificial se refiere al desarrollo de programas que tengan un comportamiento inteligente como, por ejemplo, la traducción automática de sitios web. Para los ingenieros, la inteligencia artificial es la construcción de máquinas que realicen tareas usualmente hechas por humanos. Mientras que, para los científicos cognitivos este campo de estudio se refiere a la construcción de modelos de inteligencia humana para entender de mejor manera el comportamiento humano.







En resumen, la inteligencia artificial es un campo de estudio que desarrolla programas, máquinas y que intenta simular el comportamiento humano (Franceschetti, 2018).







4.2.10 Data Augmentation

El aumento de datos es un agrupamiento de técnicas utilizadas en aprendizaje profundo para incrementar el número de instancias en un conjunto de datos, con la idea de que, al aumentar los datos, se disminuirá el error tanto en el entrenamiento como en la validación. Para generar un aumento de datos, el nuevo conjunto se basa en uno ya existente, aplicando alteraciones. En el caso de la visión por computador, las alteraciones hacen referencia a cambiar el valor de los píxeles mediante operaciones matemáticas (Shorten & Khoshgoftaar, 2019). A continuación, se describen algunas de las técnicas más aplicadas en conjuntos de imágenes.

Tabla 3. Técnicas de *data augmentation*.

Imagen original	Técnica	Ejemplo
	<p><i>Flip</i>: consiste en invertir de manera horizontal o vertical una imagen, simulando el efecto de un espejo.</p>	 <p><i>Flip</i> horizontal.</p>

	<p><i>Rotation:</i> varía la inclinación de las imágenes, ya sea de manera horaria o antihoraria.</p>	 <p>Rotación a 30°.</p>
	<p><i>Cropping:</i> se basa en hacer recortes de manera aleatoria en un conjunto de imágenes, reduciendo el tamaño de la imagen generada.</p>	 <p>Cropping al 45%.</p>
	<p><i>Noise injection:</i> consiste en añadir una matriz de valores aleatorios, donde usualmente se usa una distribución gaussiana. Esto hace que los modelos entrenados sean más robustos.</p>	 <p>Ruido al 25%.</p>

	<p><i>Color space transformation:</i> altera la distribución de colores a partir de operaciones matriciales para que el color cambie sin afectar la forma de los objetos.</p>	
	<p><i>Mixed:</i> mezcla imágenes de manera aleatoria y promedia los valores de los píxeles en los canales RGB.</p>	
	<p><i>Random erasing:</i> Técnica usada para combatir problemas relacionados con superposición de objetos, generando borrados aleatorios. Esto obliga al modelo a aprender más características descriptivas de una imagen.</p>	
		<p>Borrado automático.</p>

Fuente: A survey on Image Data Augmentation for Deep Learning (Shorten & Khoshgoftaar, 2019)

4.2.11 Aprendizaje Automático

El aprendizaje automático, también conocido como *Machine Learning* en inglés, es una técnica de modelado que se basa en datos. De manera más detallada, el objetivo del aprendizaje automático es encontrar un modelo a partir de un conjunto de datos de entrenamiento (Kim, 2017). Desde otra perspectiva, el aprendizaje automático busca generalizar el conocimiento automáticamente a partir de la experiencia, con la finalidad de hacer una predicción ante una situación desconocida (Schuld & Petruccione, 2018). Dentro de las técnicas más conocidas en el aprendizaje automático destacan: Aprendizaje supervisado, Aprendizaje no supervisado, y Aprendizaje por refuerzo.

Kim (2013) propone que, en el aprendizaje supervisado, se tiene un conjunto de datos de tipo: {entradas, salidas}, donde las salidas también son conocidas como etiquetas. Las etiquetas son el producto de la supervisión humana en determinar a qué grupo pertenece una instancia de un conjunto de datos. El objetivo de este tipo de aprendizaje es crear un entrenamiento basado en datos etiquetados, para que en un futuro logre predecir la etiqueta de datos que no pasaron por supervisión humana. Mientras que, en el aprendizaje no supervisado, solo se cuenta con un conjunto de datos: {entradas}, que permite encontrar patrones sin el proceso de etiquetado. Por su parte, el aprendizaje por refuerzo tiene datos de tipo: {entradas, algunas salidas, recompensa}, donde busca maximizar el valor de la recompensa en cada iteración.

4.2.12 Redes Neuronales

Las redes neuronales artificiales o *ANN* por sus siglas en inglés (*Artificial Neural Networks*) son interconexiones entre unidades de procesamiento simples (Bhardwaj et al., 2019) que ayudan a representar funciones utilizando redes. Estas proporcionan un método para aproximar funciones de valores discretos, reales y vectoriales (Mitchell, 1997). Una red neuronal está compuesta por nodos conectados entre sí mediante enlaces que tienen un peso. Los pesos representan el aprendizaje y almacenamiento de la red neuronal. (Russell et al., 1995).

Los nodos presentes en una red neuronal se pueden ubicar en un lugar específico de profundidad, a este lugar se denomina capa, conocido en inglés como *layer*. De manera general, una red neuronal cuenta con una estructura de tres capas (Kim, 2017): capa de entrada, capas ocultas, y capa de salida, como muestra la siguiente figura.

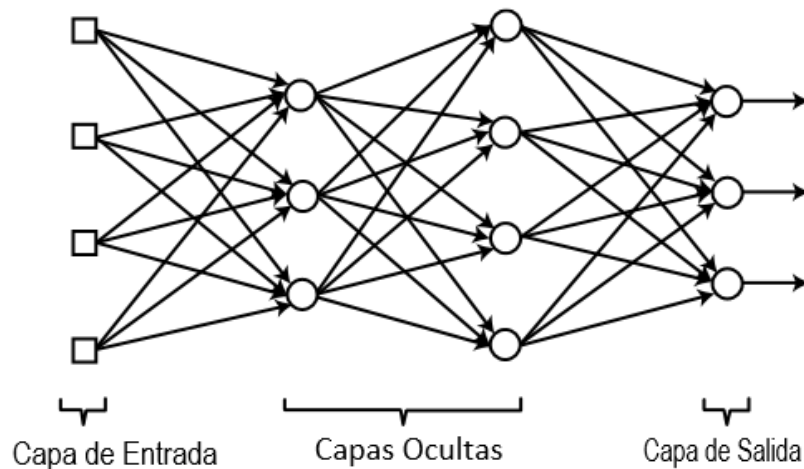


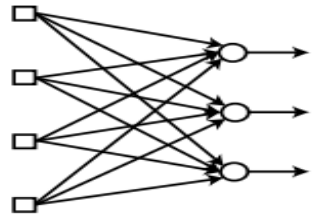
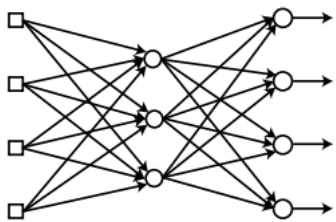
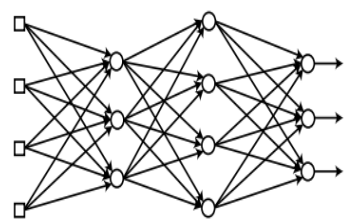
Figura 4. Estructura de capas presentes en una red neuronal.

Fuente: Tomado de *Deep Learning BT - MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence* (Kim, 2017)

El modelo matemático para el entrenamiento de una red neuronal puede ser representado como: unos datos de entrada para el entrenamiento X , unos pesos W , y unos sesgos (*bias*) b . Kim (2017), establece que la capa de entrada toma los datos de X y los pasa a la primera capa oculta sin efectuar una suma ponderada ni una función de activación. Esta primera capa oculta hace una suma ponderada $v = WX + b$ que posteriormente pasa por una función de activación $\varphi(v)$, y retorna las entradas que usará la siguiente capa oculta. De manera iterativa se calculan v y $\varphi(v)$ hasta llegar a la capa de salida. En la capa salida es común aplicar una función de activación diferente a la de las capas posteriores. Esto deja como resultado un $\tilde{y} = \varphi(v)$, que equivale a una predicción.

Por otra parte, las redes neuronales pueden clasificarse de acuerdo a la arquitectura de sus capas como se explica en la siguiente tabla.

Tabla 4. Arquitectura de capas de redes neuronales.

Arquitectura de capas		Estructura	Ejemplo
Red neuronal de una sola capa		Capa de entrada - Capa de salida	
Red neuronal de múltiples capas	Poco profunda	Capa de entrada - 1 Capa oculta - Capa de salida	
	Profunda	Capa de entrada - N Capas ocultas - Capa de salida	

Fuente: Adaptado de Deep Learning BT - MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence (Kim, 2017)

4.2.13 Aprendizaje Profundo

El aprendizaje profundo, conocido en inglés como *Deep learning*, es una técnica de aprendizaje automático que emplea una red neuronal multicapa, es decir, con dos o más capas ocultas (Kim, 2017). Kim (2017) afirma que el concepto de aprendizaje profundo se encuentra de alguna manera relacionado con el concepto de aprendizaje automático, debido a que cuenta con datos de entrada, datos de entrenamiento, y una salida.

4.2.14 Redes Neuronales Convolucionales

Las redes neuronales convolucionales, conocidas en inglés como *Convolutional Neural Network (CNN)*, o también como *ConvNet*, es una técnica que se desarrolló

entre 1980 y 1990. Pero, perdió relevancia durante los años posteriores debido a su bajo rendimiento en aplicaciones del mundo real, hasta que, en 2012 ostentaron su utilidad en el campo de la visión por computador (Kim, 2017).

Una red neuronal convolucional es una red profunda que imita el comportamiento de la corteza visual cerebral al momento de procesar y reconocer imágenes (Kim, 2017). Las *ConvNets* son útiles porque intentan dar una solución a un problema presente en las redes neuronales pre-alimentadas completamente conectadas. Dicho problema radica en que un gran número de neuronas puede impedir el desarrollo de una aplicación práctica (Habibi Aghdam & Jahani Heravi, 2017). Por lo tanto, gracias a las redes neuronales convolucionales se puede reducir la cantidad de parámetros de modo tal que, permita una red más profunda al mismo tiempo que evita el inconveniente de tener muchos parámetros (Kim, 2017).

La arquitectura de una red neuronal convolucional está constituida por dos partes, una red neuronal que extrae características de una imagen de entrada, y otra red neuronal que clasifica la imagen característica (Kim, 2017). La siguiente figura representa la arquitectura mencionada.

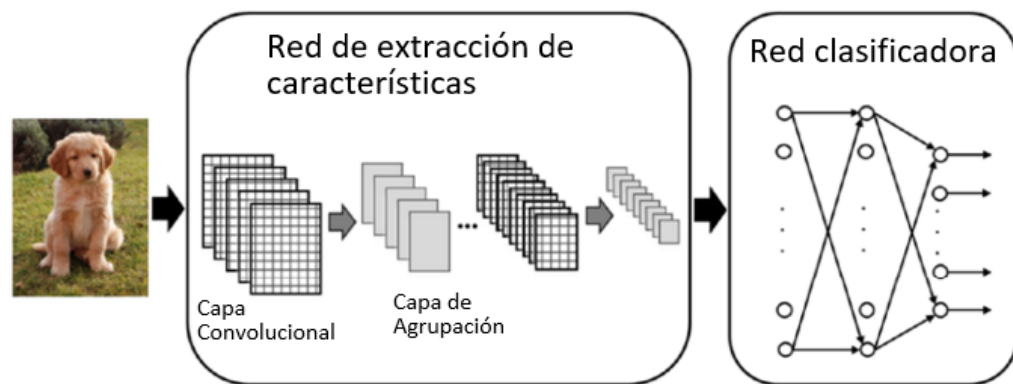


Figura 5. Arquitectura típica de una red neuronal convolucional.

Fuente: Tomado de Deep Learning BT - MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence (Kim, 2017)

4.2.15 YOLO

You Only Look Once (YOLO), es un modelo unificado que se considera el estado del arte para la detección de objetos en tiempo real. Dada una imagen, *YOLO* implementa en una red neuronal convolucional que divide la imagen en regiones para predecir múltiples cuadros delimitadores junto con sus respectivas

probabilidades de clase. Este avance ha dejado como resultado un sistema de detección en tiempo real más rápido y preciso que supera a otros sistemas del mismo tipo (Redmon, Divvala, Girshick, & Farhadi, 2018).

Redmon, Divvala, Girshick, & Farhadi (2018) plantea un modelo que divide la imagen en una cuadrícula de dimensiones $S \times S$. Donde cada celda de la cuadrícula predice B cuadros delimitadores con sus respectivas puntuaciones de confianza, y un conjunto de probabilidades C de que un objeto pertenezca a una determinada clase (Redmon, Divvala, Girshick, & Farhadi, 2018). A continuación, se brindan más detalles sobre las variables mencionadas:

$C = Pr(Clase | Objeto)$. Que resulta en una probabilidad de clases condicionales.

$B = [x, y, w, h, confianza]$. Cada cuadro delimitador contiene cinco atributos.

(x, y) : punto que representa el centro del cuadro delimitador

w : anchura del cuadro delimitador

h : altura del cuadro delimitador.

$confianza = Pr(Object) * IOU$, donde IOU es la intersección sobre la unión entre el cuadro predicho y el cuadro real.

4.2.16 Colab

Colab, también conocido como *Colaboratory*, es un producto de Google Research, que brinda un servicio basado en *Jupyter Notebook* (archivos con extensión .ipynb) alojado en la nube. *Colab* está pensado para trabajar en proyectos relacionados con aprendizaje automático, analítica de datos, y educación. De tal forma que, permite escribir y ejecutar código basado en python desde un navegador web y de forma gratuita. Adicionalmente, permite escoger entre dos tipos de entornos de ejecución: GPU y TPU. Todo lo anteriormente mencionado hace que se pueda programar sin la necesidad de contar con recursos de hardware muy potentes. (Google Colaboratory, 2020)

4.2.17 Android Studio

Android Studio es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) basado en *IntelliJ IDEA*, que sirve para desarrollar aplicaciones del sistema operativo Android (Andriod Developers, 2020). Además, es gratuito y permite el desarrollo de proyectos desde diversos sistemas operativos, como Microsoft

Windows, Mac OS, Linux, Chrome OS. En los sistemas operativos más populares, Android Studio demanda los siguientes requerimientos mínimos: memoria RAM de 4GB; almacenamiento en disco duro de 2GB; y resolución de pantalla de 1280 x 800. Android Studio cuenta con algunas características relevantes como: un sistema basado en Gradle, un emulador rápido, soporte C++ y NDK, soporte de Google Cloud Platform, y acceso a diversos entornos de trabajo como *Flutter*.

4.2.18 LabelImg

LabelImg es una herramienta gráfica de anotación de imágenes, que está escrita en *Python* y utiliza Qt para su interfaz gráfica. Las anotaciones de este, se guardan como archivos *Extensible Markup Language (XML)*, es decir, formato *PASCAL VOC*, el cual es utilizado por ImageNet, que es una base de datos de imágenes organizada. Además, también es compatible con el formato *YOLO* (Python Software Foundation, 2016).

5 METODOLOGÍA

Para el desarrollo del proyecto, se definen tres etapas principales, relacionadas estrechamente con los objetivos específicos del proyecto. A continuación, son presentadas las actividades asociadas con cada una de las etapas.

Etapas 1: Analizar los principales elementos relacionados con la aplicación de técnicas de visión por computador y la estimación de la producción agrícola. Se realiza la redacción del estado del arte que se basa en un enfoque centrado en conceptos. Las actividades que serán realizadas se presentan a continuación:

1. Búsqueda de documentos en bases de datos académicas con los conceptos: *Counting, Machine Learning, Deep learning, Computer Vision, Fruit, Cocoa, Agriculture, Production y Yield.*
2. Identificación de artículos que contienen técnicas y tecnologías relacionadas con la visión por computador que ayuden a establecer una medición y estimación de la producción agrícola.
3. Síntesis del estado del arte.
4. Análisis de los documentos recuperados, en el que se identifica el tipo de documento (Artículo científico, revisión del estado del arte, capítulo de libro), objetivo del documento, tecnologías y/o protocolos, y dispositivos de hardware.
5. Documentación de resultados.

Etapas 2: Adaptación de un modelo de Aprendizaje de Máquina para la detección y conteo de mazorcas de cacao.

Las actividades que serán realizadas se presentan a continuación:

6. Exploración de las diferentes bases de datos para la obtención del dataset, que contiene imágenes de mazorcas de cacao.
7. Etiquetado de las imágenes del dataset.
8. Entrenamiento del modelo de aprendizaje.
9. Evaluación del modelo generado.
10. Conversión del modelo a uno compatible con TensorFlow Lite.
11. Cargar el modelo al Interpretador.

Etapas 3: Diseño del prototipo móvil funcional que integre un algoritmo de aprendizaje de máquina para la identificación de mazorcas de cacao.

Las actividades que serán realizadas se presentan a continuación:

El modelo de prototipado utilizado se divide en 6 etapas, las cuáles son:

12. Recolección y refinamiento de requisitos: se establecen las limitaciones y alcances del proyecto.
13. Diseño rápido: se realizan los bocetos y modelos a seguir para tener en cuenta en la siguiente etapa. Todo esto, basado en lo establecido en la etapa de recolección y refinamiento de requisitos.
14. Construcción del prototipo: se materializan los bocetos y modelos realizados en la etapa de diseño rápido, haciendo uso de los programas ya determinados en la primera etapa.
15. Evaluación del prototipo por el cliente: el cliente revisa el prototipo construido y comunica mejoras a realizar, estas mejoras deben estar dentro de lo definido en la etapa de recolección y refinamiento de requisitos.
16. Refinamiento del prototipo: se revisa si hay mejoras indicadas por el cliente, si las hay, se procede a pasar nuevamente a la etapa de diseño rápido, para realizar las modificaciones desde los bocetos y modelos y así poder realizar nuevamente el ciclo de las etapas; si, por el contrario, no hay mejoras indicadas, se pasa a la etapa de producto de ingeniería.
17. Producto de ingeniería: se realiza la entrega del prototipo y se da por finalizado el proyecto.

6 ANÁLISIS DEL ESTADO DEL ARTE

La búsqueda de documentos e información para el desarrollo de este documento fue realizada utilizando las siguientes palabras clave: *Counting, Machine Learning, Deep learning, Computer Vision, Fruit, Cocoa, Agriculture, Production, y Yield*. La delimitación por tiempo en la búsqueda fue fijada en el período comprendido desde el año 2013 al 2020, dado que de la ecuación de búsqueda: ("Counting") AND ("Machine Learning" OR "Deep learning" OR "Computer Vision") AND ("Fruit" OR "Cocoa" OR "Agriculture") AND ("Production" OR "Yield"), se obtiene la siguiente tendencia de documentos por año, como se muestra en la siguiente figura (Figura 1).

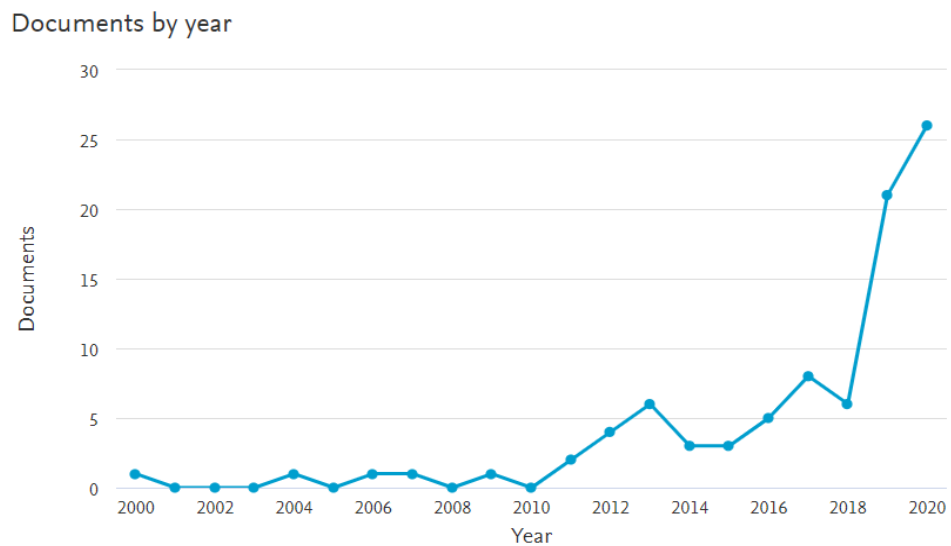


Figura 6. Gráfica de documentos por año que cumplen con la ecuación de búsqueda establecida. Fuente: Tomado de documentos por año (Scopus, 2020)

El promedio de documentos publicados por año entre 2000 y 2020 equivale a 4,2 documentos. Al observar el primer año que supera el promedio, se tiene que es 2013 con un total de 6 documentos. Por lo tanto, se justifica un aumento en documentos presentados a partir del año 2013. Esta búsqueda fue realizada a finales del 2020 en donde fueron seleccionados un total de 19 documentos, luego de hacer descartes por idioma y análisis de conceptos. La cantidad seleccionada es el punto de partida para el desarrollo del estado del arte del Proyecto de Investigación, en la Tabla 1 se presenta la información detallada.

Tabla 5. Criterios de búsqueda

Revisión de la literatura	
Palabras Clave	<i>Counting, Machine Learning, Deep learning, Computer Vision, Fruit, Cocoa, Agriculture, Production y Yield.</i>
Bases de datos Consultadas	IEEE Xplore, ScienceDirect, MDPI, Scopus.
Cantidad de Referencias Recuperadas	19 documentos recuperados
Fecha de Búsqueda	Desde agosto a diciembre del 2020
Criterios de Búsqueda	Rango de Fecha de la Búsqueda: entre 2013 y 2020.
	Pertinencia con el tema de investigación: <ul style="list-style-type: none"> • <i>Counting</i> • <i>Machine Learning</i> • <i>Deep learning</i> • <i>Computer Vision</i> • <i>Fruit</i> • <i>Cocoa</i> • <i>Agriculture</i> • <i>Production</i> • <i>Yield</i>
	Documentos en inglés, publicados en Revistas Científicas, libros o Actas de Conferencia.

A continuación, es presentada una síntesis de los documentos estudiados, los cuales fueron revisados en su totalidad por los autores del proyecto.

Grape Detection with Convolutional Neural Networks: el objetivo de este artículo es clasificar dos tipos de *Vitis Vinifera* (Albariño blancas, y Barbera rojas), implementando una arquitectura de redes neuronales convolucionales. A partir del modelo entrenado, se identifica el rendimiento y la precisión, teniendo en cuenta datos como: imágenes a color, a escala de grises, e histogramas de color. Posteriormente, se evalúan los parámetros relacionados con el tamaño de los bloques y *data augmentation*. El sistema planteado obtiene un alto rendimiento en el preentrenamiento con *ResNet*. Y propone que el *Transfer Learning* promueve a

que se alcance una mayor exactitud, sin la necesidad de entrenar por completo a la red neuronal (Cecotti, Rivera, Farhadloo, & Pedroza, 2020).

Application of Consumer RGB-D Cameras for Fruit Detection and Localization in Field: A critical review. Presenta una revisión de sistemas con sensores *RGB-D* y métodos en el procesamiento de imágenes, para detectar y localizar frutas. Para ello, se identifican los tipos de imágenes que se pueden obtener: *RGB*, de profundidad (imágenes que toma la distancia desde el sensor a cada punto de la escena como un valor de píxel), infrarrojas, y la fusión de varios tipos de imágenes. Se obtiene que las imágenes detectadas con infrarrojos tuvieron una mayor precisión, en términos de tasa de detección (*Detection Rate*), que sólo con las *RGB* (Particularmente, el *RGB* dio mejores resultados que las imágenes con profundidad). Mientras que, la combinación de múltiples tipos de imágenes ofrece mejores resultados para la detección. Finalmente, la detección y localización de frutas afronta grandes retos como superposiciones (cuando un objeto tapa parcialmente a uno que se está detectando y analizando), agrupamiento, y variabilidad en la iluminación y las condiciones de los cultivos (Fu et al., 2020).

Maize Tassels Detection: a Benchmark of the State of the Art. Este artículo presenta el caso de estudio de la detección de borlas de maíz, también conocidas como rastrojos. Los autores parten del estado del arte en el que se contemplan métodos basados en detección y regresión. Por lo que tienen en cuenta a modelos tales como: *Faster R-CNN*, *YOLOv3*, *FaceBoxes*, *RetinaNet*, y *TasselNet*. Al evaluar los modelos con datos propios obtienen que, los métodos de regresión son menos robustos que los de detección, como ocurre al evaluar *TasselNet*, que alcanza un *MSE* de 33.8 versus un 21.1 y 25.9 (*Faster R-CNN* y *RetinaNet* respectivamente). Pero, los métodos de regresión presentan una rápida convergencia e inferencia. Asimismo, los métodos basados en detección suelen subestimar la cantidad agrícola cuando se presentan superposiciones, mientras que los basados en regresión son recomendados por los autores ante esta situación. El estudio recomienda tener en cuenta la precisión, robustez, velocidad, y otros factores para elegir un método apropiado en el conteo de plantas (Zou, Lu, Li, Liu, & Cao, 2020).

Monocular Camera Based Fruit Counting and MAPping with Semantic Data Association. presenta el conteo de frutas con una cámara monocular, de esta manera el sistema de conteo logra mantener un bajo costo y rendimiento comparable a los sistemas que emplean más elementos como cámaras, sensores y *GPS*. El sistema funciona tomando dos fotografías del mismo árbol desde dos perspectivas; con una *CNN* se detecta el tronco del árbol y posteriormente se hace el reconocimiento del fruto, el rastreo también se basa en el filtro de Kalman (R. E. KALMAN, 1960). A diferencia de otros estudios, en este el problema de conteo se expande a secuencias de imágenes y no a una sola imagen (Liu et al., 2019).

Comparing Deep-Learning Networks for Apple Fruit Detection to Classical Hard-Coded Algorithms. en este artículo se comparan dos tipos algoritmos con el objetivo

de detectar manzanas en un árbol. El primero es el *Hard-Coded* (usado para la extracción de características) y el segundo, son las redes CNN. Para el algoritmo *Hard-Coded* se obtuvo que, con pocas imágenes se alcanza una precisión entre 0.6 y 0.9, mientras que para las CNN se necesitaba miles de imágenes para obtener una exactitud mayor al 50% en la detección. No obstante, cuando se integran imágenes diferentes de otras fechas o cultivos, el algoritmo *Hard-Coded* falló al detectar peras, y tuvo que ser reescrito para adaptarse a los cambios, cabe resaltar que dicha situación desfavorable no ocurre con las redes neuronales convolucionales. Por otra parte, gracias a técnicas como *Augmentation* las CNN pueden mejorar la precisión, logrando una mejor generalización del conocimiento, a la par de evitar el *overfitting* (Bresilla et al., 2020).

Faster R-CNN for Multi-Class Fruit Detection Using a Robotic Vision System: Este documento propone un marco de aprendizaje profundo para la detección de frutas con múltiples clases basado en *R-CNN* mejorado, es decir, más preciso y rápido. Se hace uso del modelo de *CNN VGG-166*, con 13 capas convolucionales, 13 capas *ReLU* y 4 capas de *pooling*. Para mejorar la detección y balancear el modelo se usa un método de regularización. Dos funciones de pérdida son añadidas para optimizar las capas convolucionales y de *pooling*. Las frutas usadas para probar el dataset son manzanas, naranjas y mango que fueron obtenidas del dataset Fruits 360, todas las imágenes son 100x100 pixels y tiene un fondo blanco; a los datos obtenidos se les realizó *data augmentation* con el fin de diversificar el dataset y mejorar la calidad de las imágenes. La evaluación muestra que, comparado con *YOLOv3* y *Faster R-CNN*, el modelo mejorado de *Faster R-CNN* obtuvo un mejor *mAP* de 86.41 comparado con 84.89 y 82.56 respectivamente, sin embargo, no presenta una mejor velocidad (47ms/imagen) comparado con *YOLOv3* que tiene 39ms/imagen. (Wan & Goudos, 2020).

Image Processing for Smart Farming: Detection of Disease and Fruit Grading: este trabajo demuestra la eficacia de los algoritmos que se utilizan como instrumento para vigilar las enfermedades de los frutos durante el cultivo y para el recuento de mangos. Por lo que proponen, vigilar la planta durante su período de crecimiento, así como en el momento de la cosecha. Para esto, se toman imágenes que se clasifican y se asignan a sus respectivas categorías de enfermedades, las cuales son tres: una de uva y dos de manzana. Estas imágenes se clasifican y se asignan a sus respectivas categorías de enfermedades con base en tres vectores de características a saber: color, textura y morfología. A partir de estos vectores de características, la morfología da un resultado correcto del 90%, que termina siendo mayor que otros dos vectores de características. La implementación final de las redes neuronales utiliza *MATLAB* (Jhuria, Kumar, & Borse, 2013).

Automated Crop Yield Estimation for Apple Orchards: en este estudio se realiza una estimación automática del rendimiento de cultivos de manzana rojas y verdes. Para esto se usa un automóvil que recorre el huerto en la noche y con ayuda de luz artificial toma las fotografías de ambos lados de los árboles. Se hace de noche para

que no intervenga la luz natural y existan cambios de luminosidad. Dado que se realiza la estimación del rendimiento de los cultivos, se debe hacer un reconocimiento y conteo de frutos de manera continua. El software usa posicionamiento global para registrar las manzanas detectadas, la detección se realiza con el tono, saturación y el valor de color *HSV*. Los errores de estimación tienen una media de -2.9% con una desviación estándar de 7.1%. (Wang et al., 2013).

A Framework for Measuring Infection Level on Cacao Pods. En esta investigación se realiza la detección y segmentación del nivel de infección de mazorcas de cacao basado en los algoritmos de *K-Means* y *SVM*; para realizar la detección se usaron los colores de la mazorca como guía. De acuerdo con los autores, el agrupamiento y la segmentación de los niveles de infección permite una mejor toma de decisiones en términos de tratamiento y gestión de enfermedades. La investigación hace uso de *SVM* porque comparado con *ANN* requiere de menos datos de entrenamiento y por lo tanto menos capacidad computacional. El algoritmo funciona con solo una mazorca por imagen, que muestra características externas de la infección. El proceso realizado para la detección comienza con tomar la fotografía. A continuación, el fondo de la mazorca es removido y se realiza a agrupamiento. Al final, se realiza la segmentación y se mide el nivel de infección del fruto (Tan et al., 2016).

Automatic Green Fruit Counting in Orange Trees Using Digital Images: El artículo propone un sistema de conteo de naranjas verdes, haciendo uso de imágenes digitales que fueron tomadas a 2 metros de distancia cada árbol, con una variación en el tiempo de toma y clima. Para hacer posible este sistema, los autores tienen en cuenta un modelo que contempla el número total de frutas detectadas en una imagen y el número total de la cantidad real presente en el árbol. Con este fin, desarrollan un método para la extracción de características que se basa en el análisis de texturas y *SVM*. Posteriormente, desarrollan un algoritmo para reconocer y contar las naranjas. El algoritmo fue evaluado, donde obtuvo una tasa de 3% en falsos positivos para la detección, y fue capaz de estimar el número medio de naranjas visibles en árboles con un error de 5% (al probarse con 46 imágenes). Finalmente, se demuestra que el conteo automático requiere de menos tiempo que el manual, con un promedio de 9.79 [s] y 64.9 [s] respectivamente (Maldonado & Barbosa, 2016)

Fast implementation of real-time fruit detection in apple orchards using deep learning: desarrolla un marco de implementación rápida de un detector de frutas basado en aprendizaje profundo *LeNet* en donde se contempla la generación automática de etiquetas. La adquisición de los datos se hace a través de un *Kinect v2* en donde se consideran distintas escenas y distancias de toma para posteriormente aplicar técnicas de *data augmentation*. La arquitectura propuesta consta de un generador de etiquetas que utiliza una pirámide multinivel y clusterización. Adicionalmente, cuenta con una red piramidal de características y un

módulo *ASPP* para una mejora en la extracción de características y la detección. Los resultados muestran que *LedNet* con *LW-net* obtiene un *F1 score* de 0.826 y un *IoU* de 86.3%, superando a *YOLOv3-Tiny* con resultados mayores en un 4.4% y 3.9% respectivamente. Por otra parte, *Faster R-CNN* resulta ser el enfoque más lento al considerar el tiempo computacional (145 ms), pero con un alto rendimiento en detección (0.818 en *F1 score*) (Kang & Chen, 2020).

Apple detection during different growth stages in orchards using the improved YOLO-V3 model: Propone una mejora de *YOLOv3* para detectar manzanas en diferentes etapas de crecimiento, teniendo en cuenta diversas condiciones de luminosidad y del cultivo. Para lograr esta diversidad se tomaron fotos en distintas horas del día simulando una iluminación frontal. La arquitectura propuesta implementa *YOLOv3* en combinación con *DenseNet* para mejorar la propagación de características y facilitar la reutilización y fusión de estas. Los resultados muestran que *YOLOv3-dense* obtuvo 0.817 en *F1 score*, 0.896 en *IoU*, y 0.304s en tiempo promedio de detección. En comparación con otros modelos, *YOLOv3-dense* supera a *YOLOv2*, *YOLOv3* y *Faster R-CNN VGG16 net*. Finalmente, se observa que los modelos basados en *YOLO* son más rápidos en tiempo promedio de detección, mientras que el basado en *Faster R-CNN* es el más lento (2.42 s), pero con un *F1 score* de 0.801. A pesar de que *Faster R-CNN* es más preciso que *YOLOv2* y *YOLOv3*, no se observa un aumento significativo respecto a este último, ya que *YOLOv3* cuenta con un *F1 score* de 0.793. (Tian et al., 2019).

Real-time kiwifruit detection in orchard using deep learning on Android™ smartphones for yield estimation: Se desarrolla una aplicación móvil en Android para detectar kiwi y posteriormente estimar el rendimiento usando *SSD* con *MobileNetV2* e *InceptionV3*. El conjunto de imágenes se tomó a través de un celular *Huawei P20* bajo diferentes condiciones de iluminación, y luego se aplicaron técnicas de *data augmentation*. La arquitectura para el entrenamiento del modelo de aprendizaje de máquina consta de una red para la extracción de características, una capa de características *SSD*, y una capa de detección. En el estudio se evaluaron *MobileNetV2*, *MobileNetV2* cuantificado, *InceptionV3*, e *InceptionV3* cuantificado para la red de extracción de características. Los resultados muestran que las tendencias de las curvas iteraciones vs pérdida son similares en los modelos cuantificados y aquellos que no lo son. Por otra parte, *MobileNetV2*, *MobileNetV2* cuantificado, *InceptionV3*, e *InceptionV3* cuantificado alcanzaron *TDRs* de 90.8%, 89.7%, 87.6%, y 72.8%, respectivamente. *MobileNet cuantificado* muestra la mayor rapidez con 103 ms/imagen y el menor tamaño en almacenamiento de modelo con 4.5 MB. La aplicación logra detectar las frutas mencionadas a través de imágenes; sin embargo, el rendimiento es insuficiente cuando se presentan superposiciones entre frutas (Zhou et al., 2020).

Deep Count: Fruit Counting Based on Deep Simulated Learning: este trabajo, hace uso de una red neuronal convolucional profunda simulada para la estimación del rendimiento, con el fin de conocer el número exacto de frutos, flores y árboles para

ayudar a los agricultores a tomar mejores decisiones sobre las prácticas de cultivo, la prevención de enfermedades de las plantas y el tamaño de la mano de obra para la cosecha. Así mismo, los resultados experimentales muestran una precisión media del 91% en imágenes reales y del 93% en imágenes sintética. Por otra parte, la red se implementó utilizando *TensorFlow* en una *GPU* NVidia 980Ti y para el entrenamiento se utilizaron 24.000 imágenes sintéticas y fue entrenada para contar tomates maduros y medio maduros. El algoritmo puede manejar la variación de la iluminación, el tamaño, la sombra y también las imágenes con frutos superpuestos y parcialmente ocluidos. Finalmente, el algoritmo se probó con 100 imágenes elegidas al azar y estos fueron comparados con varios métodos, como la técnica basada en el área, la red neuronal superficial y la red con la *Inception-ResNet* original (Rahnemoonfar & Sheppard, 2017)

Deep learning for real-time fruit detection and orchard fruit load estimation: benchmarking of 'MangoYOLO': en este trabajo se comparó el rendimiento de seis arquitecturas de aprendizaje profundo existentes (*Faster R-CNN-VGG*, *Faster R-CNN-ZF*, *YOLOv3*, *YOLOv2*, *YOLOv2-Tiny* y *SSD*) con una arquitectura nombrada *MangoYOLO* (basada en las características de *YOLOv3* y *YOLOv2-Tiny*) para la tarea de detección de frutos de mango en imágenes de copas de árboles. Se tuvieron en cuenta cinco huertos, a los cuales se les tomaron fotos adquiridas por la noche utilizando una cámara digital *RGB* de 5 megapíxeles. El modelo propuesto por los autores alcanza un *F1 score* de 0.968 y un *AP* de 0.983, superando a las otras arquitecturas de comparación. Además, el modelo mostró un alto rendimiento al probarse con imágenes de otras huertas y cultivos bajo distintos niveles de iluminación (Koirala et al., 2019)

Comparison of convolutional neural networks in fruit detection and counting: A comprehensive evaluation: en este trabajo, se implementa y evalúa el rendimiento de dos enfoques de *CNN* para resolver el problema del conteo de frutas, uno de estos es *Faster R-CNN* con *Inception V2* y *SSD* con *MobileNet*. Además, se propone un método de seguimiento multiobjeto basado en estimación Gaussiana para contar a partir de videos. Los datos contemplados se tomaron a partir de una cámara *RGB* en cultivos de aguacate Hass, limón chileno, y manzanas de California. El conjunto de datos de prueba muestra un *mAP* de 72% para el modelo basado en *Faster R-CNN*, y un 53% para el basado en *SSD*. Sin embargo, a *Faster R-CNN Inception v2* le toma 220 ms para procesar cada frame, mientras que a *SSD mobilenet* 60 ms. A pesar del alto rendimiento en detección, estos modelos son muy dependientes de un gran conjunto de datos, bajo diferentes condiciones ambientales (Vasconez et al., 2020).

Real-Time Plant Leaf Counting Using Deep Object Detection Networks: este artículo plantea el conteo, detección y localización de hojas de *Arabidopsis* en tiempo real, a partir de un conjunto de datos con 60 plantas a las cuales se les tomó dos fotos diarias a lo largo de un mes, con el fin de que estuviesen en diferente rotación, posición y etapa de crecimiento. Para este trabajo se optó por *YOLOv3-Tiny* porque

usa menos capas que *YOLO* y puede ser más rápida. Adicionalmente, este modelo representa una ventaja ya que, para el conteo de hojas, la pérdida en la eficiencia no es una gran preocupación debido a que solo hay una categoría de objetos y tienen formas similares. Otra ventaja de usar *YOLOv3-Tiny* es que requiere de menos hardware, garantizando más accesibilidad y versatilidad. Para entrenar la red y evaluarla se usó *Darknet* ya que presenta una mayor portabilidad. El modelo final fue probado en un teléfono móvil Honor 7x con sistema operativo Android en un ambiente controlado, teniendo un tiempo de inferencia de 5s. *YOLOv3-Tiny* obtuvo un tiempo de inferencia de 0.009s, una precisión de 0.8705 y un *F1 score* de 0.9446 comparado con *Faster R-CNN* que obtuvo 0.91s (100 veces menor a *YOLOv3-Tiny*), 0.9112 y 0.8986 respectivamente. Además, *YOLOv3-Tiny* presentó menor *mean-square error*, *accuracy* y menos ocurrencia de falsos positivos. (Buzzy et al., 2020).

*L*a*b*Fruits: A Rapid and Robust Outdoor Fruit Detection System Combining Bio-Inspired Features with One-Stage Deep Learning Networks*: este trabajo tiene como propósito la detección de fresas maduras en un ambiente real combinando el aprendizaje profundo teniendo en cuenta la teoría del color opuesto, de modo que buscan fusionar características en lugar de cambiar la arquitectura de la red. El dataset fue recolectado a lo largo de dos meses con diferenciación de 11 horas entre las capturas para garantizar diferentes niveles de luminosidad, tomando así mismo, datos del ambiente como temperatura y humedad. Luego, las imágenes fueron etiquetadas según su madurez (maduras e inmaduras). Posteriormente, se compara el enfoque propuesto por los autores con la implementación de otros modelos de color en términos de *F1 score*, donde se logró un aumento de 2.4% y 8.2% sobre *RGB* y *CIE Lab* solamente (Kirk et al., 2020).

Image Segmentation for Fruit Detection and Yield Estimation in Apple Orchards: en este artículo se presenta la segmentación de imágenes para la detección y estimación en un cultivo de manzanas. La segmentación utilizó dos algoritmos *MLP* y *CNN*. El dataset consta de alrededor de 8000 imágenes de 1232 x 1616, se eligió un conjunto de datos más pequeño que fue elegido al azar. Cada imagen de este subconjunto fue dividida en 32 sub-imágenes de 308 x 203 px. Finalmente se obtuvo un dataset con 1100 imágenes que fueron etiquetadas manualmente con las clases fruta o no fruta. La subdivisión de las imágenes hace el etiquetado más sencillo y con mayor varianza. Para realizar el entrenamiento el dataset fue dividido en 80% para el entrenamiento 10% para validación y 10% para test. La evaluación fue realizada con *F1 score*. *MLP* con metadata obtuvo un error de 13.3%, sin metadata de 12.2% y *CNN* de 10.48%. (Bargoti, S., & Underwood, J. P. (2017).

En la Tabla 2, es presentada información relevante de cada uno de los documentos estudiados, los cuales fueron desarrollados en escenarios relacionados con el cultivo de cacao.

Tabla 6. Elementos destacados de los de documentos recuperados.

Título	Tipo	Objetivo	Tecnologías / Protocolos	Dispositivos
<i>Comparing deep-learning networks for apple fruit detection to classical hard-coded algorithms</i> (Bresilla et al., 2020)	Artículo científico	Comparar los algoritmos clásicos para la detección de objetos, con el objetivo de detectar manzanas en un árbol	<i>hard-coded</i> , redes neuronales convolucionales, YOLOv2	webcam, cámara DSLR, teléfono inteligente
<i>Maize tassels detection: a benchmark of the state of the art</i> (Zou, Lu, Li, Liu, & Cao, 2020)	Revisión del estado del arte	Presentar el estado del arte en la detección de objetos y métodos de conteo	<i>R-CNN</i> , YOLOv3, <i>FaceBoxes</i> , <i>RetinaNet</i> , <i>Tassel Net</i>	Cámara digital (E450 Olympus)
<i>Grape detection with convolutional neural networks</i> (Cecotti, Rivera, Farhadloo, & Pedroza, 2020)	Artículo científico	Clasificar dos tipos de uva (Albariño blancas, y Barbera rojas)	<i>CNN</i>	<i>iPhone 8</i>
<i>Application of consumer RGB-D cameras for fruit detection and localization in field: A critical review</i> (Fu et al., 2020)	Revisión de la literatura	realizar una revisión de sistemas con sensores RGB y métodos en el procesamiento de imágenes, para detectar y localizar frutas	<i>Aprendizaje profundo</i> , <i>SVM</i> , <i>ANN</i>	Cámaras RGB-D, Kinect v1, Xtion PRO Live, Kinect v2, RealSense R200, RealSense D435
<i>Monocular Camera Based Fruit Counting and MAPping with Semantic Data Association</i> (Liu et al., 2019)	Revisión de la literatura	Propone conteo de alta efectividad con cámara monocular.	<i>Faster R-CNN</i> , <i>Semantic SfM</i>	Cámara monocular

<i>Faster R-CNN for multi-class fruit detection using a robotic vision system (Wan & Goudos, 2020)</i>	Revisión de la literatura	Este documento propone un marco de aprendizaje profundo para la detección de frutas de clases múltiples basado en un <i>R-CNN</i> mejorado y más rápido.	<i>R-CNN</i>	<i>Cámara, Robots</i>
<i>Image processing for smart farming: Detection of disease and fruit grading (Jhuria, Kumar, & Borse, 2013)</i>	Artículo científico	Este trabajo demuestra la eficacia de los algoritmos para la propagación de enfermedades y el recuento de mangos	<i>Redes Neuronales, MATLAB</i>	<i>Cámara</i>
<i>Automated Crop Yield Estimation for Apple Orchards (Wang et al., 2013)</i>	Capítulo de libro	En este estudio se realiza la estimación de rendimiento para un cultivo de manzanas rojas y verdes con visión artificial	<i>Detección de píxeles, POS LV (Position and Orientation System for Land Vehicles)</i>	<i>Equipo estéreo de dos cámaras, Vehículo autónomo, GPS</i>
<i>A framework for measuring infection level on cacao pods (Tan et al., 2016)</i>	Artículo científico	En esta investigación se realiza la detección y segmentación de los niveles de infección en mazorcas de cacao	<i>k-means, SMV</i>	<i>Cámara, computador.</i>
<i>Automatic green fruit counting in orange trees using digital images (Maldonado & Barbosa, 2016)</i>	Artículo científico	Propone un sistema de conteo de naranjas verdes, haciendo uso de imágenes digitales.	<i>OpenCV, modelo de color, umbralización, filtrado espacial con operadores de Laplace, Sobel y desenfoco gaussiano.</i>	<i>Cámara Sony DSC-W530</i>
<i>Fast implementation of real-time fruit detection in apple orchards using deep learning (Kang & Chen, 2020)</i>	Artículo científico	Propone un framework de etiquetado automático y aprendizaje profundo para detectar manzanas.	<i>Pirámide multinivel, clusterización, FPN, ASPP.</i>	<i>Kinect v2</i>

<i>Apple detection during different growth stages in orchards using the improved YOLO-V3 model (Tian et al., 2019).</i>	Artículo científico	Propone una mejora de YOLOv3 para detectar manzanas en diferentes etapas de crecimiento.	YOLO, DenseNet	Cámara de 3k x 4k píxeles de resolución.
<i>Real-time kiwifruit detection in orchard using deep learning on Android™ smartphones for yield estimation (Zhou et al., 2020)</i>	Artículo científico	Propone una aplicación móvil en Android para detectar kiwi y estimar el rendimiento usando SSD con MobileNetV2 e InceptionV3.	MobileNetV2, MobileNetV2 cuantificado, InceptionV3, e InceptionV3 cuantificado.	Huawei P20
<i>Deep Count: Fruit Counting Based on Deep Simulated Learning (Rahnemoonfar & Sheppard, 2017)</i>	Artículo científico	Este trabajo busca conocer el número exacto de frutos, flores y árboles utilizando TensorFlow.	TensorFlow	GPU NVidia 980Ti
<i>Deep learning for real-time fruit detection and orchard fruit load estimation: benchmarking of 'MangoYOLO' (Bargoti & Underwood, 2017)</i>	Artículo científico	En este trabajo se comparó el rendimiento de seis arquitecturas de aprendizaje profundo existentes para la tarea de detección de frutos de mango en imágenes de copas de árboles, haciendo uso de YOLOv3, YOLOv2, YOLOv2(tiny)	YOLOv3, YOLOv2, YOLOv2-Tiny	Cámara digital RGB de 5 megapíxeles y 720 W de iluminación LED
<i>Comparison of convolutional neural networks in fruit detection and counting: A comprehensive evaluation (Vasconez et al., 2020)</i>	Artículo científico	Este trabajo implementa y evalúa el rendimiento de dos enfoques de CNN para resolver el problema del recuento de frutas.	Faster R-CNN, SSD con MobileNet	Cámara comercial

<i>Real-Time Plant Leaf Counting Using Deep Object Detection Networks (Buzzy et al., 2020)</i>	Artículo científico	En este artículo se plantea el conteo, detección y localización de hojas de Arabidopsis en tiempo real.	<i>YOLOv3-Tiny Faster CNN</i>	<i>Cámara Canon Rebel XS. Robot autónomo. Honor X7.</i>
<i>L*a*b*Fruits: A Rapid and Robust Outdoor Fruit Detection System Combining Bio-Inspired Features with One-Stage Deep Learning Networks. (Kirk et al., 2020)</i>	Artículo científico	Propone la detección de fresas maduras en un ambiente real combinando el aprendizaje profundo con la teoría del color opuesto.	<i>espacios de color CIElab y RGB</i>	<i>Cámara. Plataforma robótica modular Thorvald.</i>
<i>Image Segmentation for Fruit Detection and Yield Estimation in Apple Orchards (Bargoti, S., & Underwood, J. P. (2017)</i>	Artículo científico	Se presenta la segmentación de imágenes para la detección de y estimación en un cultivo de manzanas.	<i>MLP, CNN</i>	<i>Cámara. Vehículo de investigación terrestre.</i>

6.1 ANÁLISIS DETALLADO DEL ESTADO DEL ARTE

A partir del planteamiento del estado del arte, se observan unos elementos en común que tienen los proyectos de visión por computador que pretenden estimar la producción agrícola. Por una parte, se observa que, al buscar en distintas fuentes académicas, los cultivos más destacados suelen ser los de uva, manzana, fresa, mango, kiwi, maíz, y naranja. Lo particular de dichos cultivos es que suelen tener frutos pequeños en comparación con las mazorcas de cacao. Son muy abundantes cuando es tiempo de cosecha, y esto significa un gran reto para las personas que tienen que contar a lo largo de muchas hectáreas de terreno.

También se observa que para llevar a cabo un proyecto de visión por computador se debe disponer de datos, un modelo de reconocimiento, y de tecnologías de desarrollo de software. Los datos suelen ser tomados tanto de repositorios abiertos como pagos, o por cuenta propia tomando fotos de manera presencial. La ventaja de contar con datos que no son tomados por cuenta propia es que estos pueden pasar por un proceso de limpieza y preprocesamiento, en donde se descartan imágenes repetidas, borrosas, y dependiendo del enfoque del problema, además, pueden venir etiquetados por expertos, por lo que disminuye costos. Sin embargo, la desventaja es que no se tiene un control de las variables de toma, por ejemplo,

en caso de querer detectar objetos lejanos con un conjunto de datos que solo tiene objetos cercanos, se corre el riesgo de que al final el modelo entrenado no resuelva el problema como se espera.

Por su parte, el enfoque de toma presencial implica que usen elementos de hardware para la toma de imágenes. Dentro de estos elementos se encuentran sensores, cámaras profesionales, celulares inteligentes, y la combinación de placas programables con componentes electrónicos. La ventaja de una autoría en los datos es que se tiene un control total de las variables de toma como son: la distancia entre la cámara y el objeto, la estabilización y posición de la cámara, el ángulo de toma, entre otras. Sin embargo, la desventaja radica en que se debe dedicar más tiempo y esfuerzo a tareas de captura, limpieza, y preprocesamiento de datos. Por lo tanto, dependiendo del problema, y los recursos con los que se cuente se puede optar por un conjunto de datos elaborado, o partir desde cero.

En cuanto a los modelos de reconocimiento, se observa que son muy variados, y dependientes tanto de los datos como del problema. Dentro de los artículos seleccionados, destacan los siguientes modelos: detección de objetos con redes neuronales convolucionales, máquinas de vectores de soporte (SVM), K Medias (K Means), filtrados basados en transformaciones de espacios de color, y segmentación de imágenes. En síntesis, las redes neuronales convolucionales para la detección y clasificación de objetos llegan a dar mejores resultados en términos de precisión. Además, no son dependientes de una sola técnica de extracción de características, porque en cada iteración aprende nuevos filtros que disminuyen el error en detección. Caso contrario ocurre con *K Means*, *SVM*, y las transformaciones de color, porque si se usan de manera individual lograrán resultados más rápidos, pero con menor precisión, sin llegar a generalizar el conocimiento como lo hacen las redes neuronales. A pesar de que las redes neuronales requieren de grandes cantidades de datos, su arquitectura es sencilla para modelar, y se cuenta con herramientas que pretenden optimizar el entrenamiento y desarrollo de soluciones tecnológicas.

Adicionalmente, se observa una implementación de tecnologías que ayudan a disminuir los tiempos de desarrollo, y la complejidad algorítmica. Dentro de las herramientas más mencionadas se contemplan librerías, frameworks, y entornos de desarrollo como: Opencv, Darknet, TensorFlow, Android Studio, y NDK (Native Development Kit). No obstante, los artículos científicos no suelen proveer mucha información sobre la metodología o los modelos de ciclo de vida de desarrollo, ni entregables basados en lenguaje unificado de modelado (UML, por sus siglas en inglés). Algunos suelen brindar pseudocódigos y diagramas de flujo sencillos que minimizan el funcionamiento de los modelos de visión por computador aplicados en múltiples plataformas. Por consiguiente, se hace difícil implementar modelos de detección cuando sólo se cuenta con un conocimiento basado en el componente teórico pero deficiente en la divulgación de la práctica.

Finalmente, con base en las técnicas relacionadas con la visión por computador, es necesario mencionar las métricas que suelen ser usadas para estimar o evaluar la producción. Para ello, se propone en primera instancia reiterar la definición de producción y rendimiento. Rendimiento se refiere a la producción cosechada por hectárea en un determinado cultivo. Mientras que la producción incluye unidades del peso del producto, en donde se pueden tener en cuenta pérdidas, cantidades consumidas en el mismo lugar de producción, y cantidades comercializadas (FAOSTAT, 2021). Con base en lo anterior, se observa que los resultados presentados en los artículos científicos toman como referencia un conteo hecho por el modelo de detección y es comparado con los datos denominados verdad fundamental, que hace referencia al número verdadero de frutos que hay en las imágenes de prueba, o en los terrenos. A continuación, se presentan algunas métricas para estimar o evaluar la producción agrícola identificadas en los documentos.

Tabla 7. Métricas para estimar la producción agrícola

Métrica	Ecuación	Descripción
Precisión porcentual (Rahnemoonfar & Sheppard, 2017)	$pa(\%) = \left[1 - \frac{ pc - ac }{ ac } \times 100 \right]$	<i>pc</i> : conteo predicho <i>ac</i> : conteo actual mide la precisión del conteo predicho.
Estimación de frutas (Koirala et al., 2019)	$E = C \times A$	<i>C</i> : conteo manual <i>A</i> : factor de conversión El factor de conversión ajusta la estimación a una más cercana con la realidad.
Error de estimación (Koirala et al., 2019)	$SE_{A \times B} = A \times B \times \sqrt{\left(\frac{a}{A}\right)^2 + \left(\frac{b}{B}\right)^2}$	<i>A</i> : factor de conversión <i>B</i> : conteo de máquina por árbol <i>a</i> : error estándar asociado <i>b</i> : distancia media cuadrática mínima asociada Determina el error en la estimación.
Diferencia de conteo (Buzzy et al., 2020)	$DiC = \frac{1}{N} \sum_{i=1}^N \epsilon_i$	<i>DiC</i> : la diferencia entre la verdad fundamental y las instancias estimadas <i>N</i> : número de muestras

	donde: $\epsilon_i = y_i - \hat{y}_i$	Promedia las diferencias de conteo
Diferencia absoluta en el conteo (Buzzy et al., 2020)	$(DiC) = \frac{1}{N} \sum_{i=1}^N \epsilon_i $	i : valor absoluto de la diferencia de conteo Aplica valor absoluto en caso de que la diferencia de un valor negativo
Error cuadrático medio (Buzzy et al., 2020)	$(MSE) = \frac{1}{N} \sum_{i=1}^N \epsilon_i^2$	i^2 : diferencia de conteo elevado al cuadrado Eleva al cuadrado la diferencia de conteo para determinar el error
Acuerdo porcentual (%) (Buzzy et al., 2020)	$\frac{1}{N} \sum_{i=1}^N 1[\epsilon = 0]$	1: función del indicador, que devuelve cero si el error 0, de lo contrario devuelve uno

Con el propósito de establecer una métrica para estimar la producción en los cultivos de cacao, el presente proyecto toma como referencia el índice de mazorca (IM). Dicho índice representa el número de mazorcas de cacao que se requieren para producir 1 kilogramo (Kg) de granos de cacao seco, y cambia dependiendo de la variedad de cacao (Fedecacao, 2005).

Dado un índice de mazorca IM , se requieren n mazorcas para producir 1 Kg de cacao seco:

$$IM = n$$

Por lo tanto, si se cuentan m mazorcas, entonces una estimación e usa la siguiente relación:

$$e = \frac{m}{IM}$$

7 ADAPTACIÓN DEL MODELO DE APRENDIZAJE

Con el objetivo de adaptar un modelo de aprendizaje de máquina para la detección y conteo de mazorcas de cacao es necesario disponer de un conjunto de datos consolidado que servirá para entrenar un modelo de detección de objetos. Por consiguiente, el presente capítulo describe la adquisición, selección, y preparación de los datos. Para después especificar el modelo de aprendizaje, el entrenamiento, la evaluación, y la conversión para dar cumplimiento con la adaptación.

7.1 ADQUISICIÓN DE DATOS

La adquisición del conjunto de imágenes consta de dos fases: búsqueda y toma de datos. Para la búsqueda se consultaron plataformas como *Kaggle*, *Github*, *Google Research Datasets*, y *Open Images Dataset v6*. Donde se usaron las siguientes palabras clave: *Theobroma cacao*, *cocoa ripeness*, *cocoa diseases*, *cocoa pods*, *cocoa tree*; la búsqueda se llevó a cabo tanto en inglés como en español, dando como resultado dos conjuntos de imágenes destacados que se describen a continuación.

Tabla 8. Conjuntos de datos resultados de la búsqueda.

Conjunto de imágenes	Variedades de cacao	Clases	Cantidad de imágenes por clase	Etiquetado	Tamaño de imagen (píxeles)
Cocoa Ripeness Dataset TCS 01 ^a	TCS01	inmadura, madura, sobremadura	263, 76, 137	No	505x505
Cocoa Diseases(YOLOv4) ^b	FEAR5, FEC, FSV41, CCN51	fitoftora, monilia, sana	100, 105, 107	Sí	3120x4160

Fuente: ^a *Cocoa Ripeness Dataset TCS 01* (Heredia Gómez & Talero Sarmiento, 2020) ^b *Cocoa Diseases (YOLOv4)* (Serrano & Heredia Gómez, 2020)

Debido a que se pretende contar mazorcas de cacao que se encuentran en los árboles, es necesario tener un conjunto de datos que tenga los frutos en su entorno natural. Con base en esta condición, se observa que el conjunto de imágenes relacionado con la madurez (Cocoa Ripeness Dataset TCS 01 (Heredia Gómez & Talero Sarmiento, 2020)) sólo cuenta con algunas fotos tomadas desde los árboles

(263 inmaduras, 26 maduras, 0 sobremaduras) mientras que el dataset de enfermedades (Cocoa Diseases(YOLOv4) (Serrano & Heredia Gómez, 2020)) tiene en su totalidad tomas hechas en el estado natural, que enfocan al árbol y considera múltiples mazorcas por toma.

Para la segunda fase, se lleva a cabo una toma de datos presencial a pesar de la actual pandemia COVID-19. Para ello, el día 24 de febrero del 2021 se viajó al municipio de El Playón, en el departamento de Santander (Colombia). Allí se visita la sede SENA Centro de Atención al Sector Agropecuario - Regional Santander, con la compañía del instructor e ingeniero agrónomo Néstor Edsgardo Nieves Peña. En este entorno se aprecian las siguientes condiciones en el cultivo de CNN51: la mayoría de las mazorcas se encuentran en una etapa inicial de crecimiento, es decir, son muy pequeñas mientras que las restantes, que están en una etapa de crecimiento avanzada, pertenecen a la cosecha intermedia del cultivo; además los árboles requieren de poda. A pesar de esto, las fotografías son tomadas allí ya que contaba con mayor extensión de cultivo.

La toma inicia aproximadamente a las 8 a.m donde es importante resaltar las siguientes restricciones. Cada imagen se toma a una distancia menor a 3 metros del árbol con la cámara posterior del celular; no se activa el flash; se mantiene una posición vertical en el teléfono; se evita el autoenfoco (AF); se mantienen los valores automáticos de ISO, balance de blancos (WB), y velocidad de obturación; se varía la posición, ángulo, y la trayectoria de la luz en cada toma; adicional a esto, cada toma debe capturar en su mayoría el área total del árbol para garantizar una detección de frutos lejanos. Para asegurar una variabilidad en la toma de datos, y prevenir a futuro el sobreentrenamiento, las fotos son tomadas desde tres celulares diferentes que a su vez toman fotos sin repetir árboles. Las características de las fuentes de toma, y las imágenes se describen a continuación.

Tabla 9. Características de los dispositivos usados en la toma de datos.

Celular	Resolución de la cámara trasera (Mpx)	Apertura focal	Resolución de toma (px)	Cantidad de fotos tomadas
Moto G5 plus	12	f/1.7	3024 x 4032	165
Moto One Action	12	f/1.8	3072 x 4096	270
Xiaomi Redmi Note 8	48	f/1.8	3000 x 3999	233

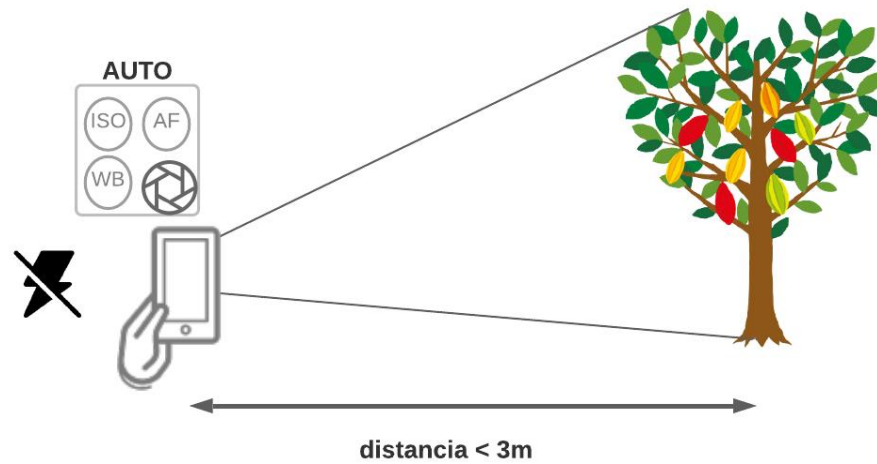


Figura 7. Características principales en la toma de datos presencial.

7.2 ANÁLISIS Y SELECCIÓN DEL CONJUNTO DE DATOS PARA EL ENTRENAMIENTO

Una vez se adquieren los datos, es necesario hacer una selección de cuáles conjuntos estarán presentes en el entrenamiento, en últimas se pretende usar los datos tomados de manera presencial y unirlos con otro conjunto para lograr un aumento de datos que presente una variabilidad. Al contemplar que se dispone de dos posibles conjuntos de datos es necesario analizarlos para tomar una decisión acertada. Como fue mencionado con anterioridad, para detectar y contar las mazorcas de cacao es necesario que estas se encuentren en su ambiente natural. En caso contrario, se podrían generar sesgos que afecten la precisión en detección. Asimismo, es importante seleccionar un conjunto de datos que en su mayoría considere múltiples frutos de cacao por toma, y desde distancias que alcancen a enfocar el árbol. En un inicio se plantea un análisis simple de datos que se muestra a continuación.

Tabla 10. Análisis de los conjuntos de datos.

Conjunto de datos	Etiquetas	Cantidad de fotos	Cantidad tomada desde los árboles	Proporción de utilidad
Cocoa Ripeness Dataset TCS 01	Inmadura	263	263	60,71%
	Madura	76	26	

	Sobremadura	137	0	
	Total	476	289	
Cocoa Diseases(YOLOv4)	Sana	100	100	
	Monilia	105	105	
	Fitoftora	107	107	
	Total	312	312	100%
CCN51	Sin definir	668	668	100%

En la anterior tabla se define como proporción de utilidad al resultado obtenido de dividir el total de fotos tomadas desde los árboles entre la cantidad total de fotos de cada conjunto de datos. A partir de lo anterior, se observa que todas las fotos de mazorcas sobremaduras fueron tomadas en un entorno diferente al natural, y de manera individual. Situación similar ocurre con la mayoría de las mazorcas maduras. Adicionalmente, se observa un alto desbalance de etiquetas, porque hay más fotos de la categoría inmadura en comparación con las de madura y sobremadura. Por otra parte, el conjunto relacionado con enfermedades tiene una proporción de utilidad del 100%, y conserva el balance de etiquetas. Al tener en cuenta sólo las condiciones de toma, el balance de etiquetas, y la proporción de utilidad, el conjunto relacionado con enfermedades llega a ser un posible candidato. No obstante, este análisis no es suficiente para tomar una decisión, porque se debe identificar el enfoque del etiquetado, grosso modo, se identifican dos enfoques para etiquetar las imágenes, uno por madurez y otro por sanidad. A continuación, se plantea el siguiente análisis por enfoques en dónde para calcular el tiempo estimado se plantea la relación 1 foto etiquetada/minuto

Tabla 11. Análisis de los conjuntos de datos por enfoques.

Conjunto de datos	Enfoque por madurez			Enfoque por sanidad		
	Etiquetado por madurez	Cantidad a etiquetar	Distintos tipos de madurez	Etiquetado por sanidad	Cantidad a etiquetar	Distintos tipos de sanidad
Cocoa Ripeness Dataset TCS 01	No	289	Sí	No	289	No
Cocoa Diseases(YOLOv4)	No	312	Algunos	Sí	0	Sí
CCN51	No	668	Algunos	No	668	Sí
Tiempo estimado	21.15 horas			15.95 horas		

Con base en lo anterior, es menos costoso elegir un enfoque basado en la sanidad, porque la mayoría de sus instancias están etiquetadas y requiere de menos tiempo. A esto se le debe agregar que la madurez es muy dependiente de la variedad, por lo que si se pretende contemplar este enfoque de datos en el entrenamiento, se debería considerar otras etiquetas diferentes a inmadura, madura o sobremadura, y optar por un modelo multi-etiqueta y multi-clase, teniendo que etiquetar a qué variedad pertenece la mazorca y su estado actual de madurez. Mientras que, con un enfoque basado en sanidad, la monilia y la fitóftora presentan los mismos patrones sin importar la variedad. Por lo tanto, a partir de la toma de las fotos, el balance de las etiquetas, la proporción de utilidad, y el tiempo estimado de etiqueta, se decide escoger un enfoque basado en sanidad que contemple el conjunto de de datos de enfermedades y el de CCN51 para que consoliden un único conjunto. Cabe aclarar que no se consideran las imágenes del conjunto TCS01 debido a que no cuenta con suficientes datos de la variedad bajo distintos estados de sanidad.

7.3 LIMPIEZA Y ETIQUETADO

Antes de etiquetar los objetos presentes en cada imagen del conjunto de datos CCN51 se contaba con 662 instancias. De las cuales se descartaron 379 imágenes, donde 76 pertenecían al Moto G5 Plus, 147 al Moto One Action, y 156 al Redmi Note 8; dando como resultado 289 instancias para etiquetar. Dentro de las razones para descartar las imágenes se tienen las siguientes: imágenes duplicadas, borrosas, y con un mayor número de mazorcas que se sospechaban enfermas. Cabe mencionar que para el etiquetado no se cuenta con ayuda profesional, por esta razón se decide no etiquetar mazorcas enfermas, y eliminar aquellas fotos con mazorcas que presentan indicios de enfermedad. En últimas, el objetivo de contemplar imágenes del dataset propio es agregar variabilidad en la categoría de sanas y aumentar la precisión en detección.

Para el proceso de etiquetado de imágenes se descarga e instala el programa *Labelimg versión 1.8.1* para windows. Gracias a esta herramienta se pueden etiquetar imágenes de formato JPG y generar automáticamente etiquetas en formato YOLO. Dicho formato se guarda con una extensión *txt*, que contiene una serie de parámetros por línea asociados al cuadro delimitador de la detección: clase, centroide x, centroide y, anchura, y altura. Posteriormente, se carga el conjunto de datos de CCN51, teniendo en cuenta las siguientes condiciones.

1. Crear el mapa de etiquetas: Fitóftora, Monilia, y Sana, que se representan como un conjunto categórico (0,1,2).
2. Establecer cuadros delimitadores estrechos, es decir, que capturen toda la forma de la mazorca, reduciendo en lo más posible el ruido.

3. En cuanto a las superposiciones, se deben etiquetar las mazorcas a pesar de un alto nivel de agrupamiento.
4. Etiquetar todos los objetos de interés visibles y fáciles de clasificar. De modo que, si un objeto se ve borroso o se confunde con otro objeto del entorno, se opta por no dibujar un cuadro delimitador.



Figura 8. Muestra de etiquetado.

El resultado final de la selección, etiquetado y limpieza dejan un conjunto de imágenes con 601 instancias que contiene 389 imágenes de mazorcas sanas, 107 de fitóftora y 105 de monilia.

7.4 ELECCIÓN DEL MODELO DE APRENDIZAJE MÁQUINA

Durante la revisión de antecedentes en la literatura se observó que las redes neuronales cuentan con un alto nivel de precisión al detectar y contar objetos en imágenes y videos. Adicionalmente, pueden ser implementadas en diversos entornos de desarrollo como los de escritorio, web, y móviles. No obstante, hasta la

fecha existen pocas investigaciones orientadas a detección de objetos en entornos móviles. Es por ello que, esta es una oportunidad para desarrollar y poner a prueba un modelo de detección de objetos que funcione sin conexión a internet y con limitados recursos de hardware. Para sacar provecho de los teléfonos inteligentes en tareas como las ya mencionadas, es importante elegir muy bien el modelo de aprendizaje profundo, porque de ello depende el tiempo de entrenamiento, la precisión en las detecciones, el tiempo de inferencia, y la complejidad del desarrollo de software.

En detección de objetos, se identifican dos tipos de modelos: los basados en regiones, y los basados en clasificación y regresión. Al tener en cuenta algunos resultados del estado del arte (Zou, Lu, Li, Liu, & Cao, 2020) (Kang & Chen, 2020) (Tian et al., 2019), se demuestra que los modelos basados en regiones llegan a ser muy precisos en términos de *AP*, pero son los más lentos en términos de tiempo de inferencia. Por otra parte, los basados en clasificación y regresión, son menos precisos (con una diferencia poco significativa), pero destacan en su rapidez y fácil implementación en entornos móviles. Por lo tanto, se decide orientar la elección a un modelo basado en regresión y clasificación. A continuación, se presenta una comparativa con los modelos más destacados en regresión y clasificación, evaluados con el dataset MS COCO con 30 clases, batch=1, sin usar tensor RT, y con una GPU Maxwell (Bochkovskiy et al., 2020).

Tabla 12. Comparativa de los modelos más destacados en regresión y clasificación, evaluados con el dataset MS COCO.

Modelo	Backbone	FPS	AP	AP50
LRF (Learning Rich Features)	ResNet-101	31.3	37.3%	58.5%
M2det	VGG-16	33.4	33.5%	52.4%
RefineDet	VGG-16	38.7	29.4%	49.2%
SSD (Single Shot Detector)	VGG-16	43	25.1%	43.1%
YOLOv4	CSPDarknet-53	31	43.0%	64.9%

Fuente: YOLOv4 / Scaled-YOLOv4 / YOLO - Neural Networks for Object Detection (Windows and Linux version of Darknet) (AlexeyAB, 2020)

En la anterior comparativa, se observa que YOLOv4 (considerado el estado del arte en detección de objetos) es el mejor en tiempo de inferencia, y precisión. Asimismo, se puede implementar en dispositivos Android gracias al framework de Tensorflow Lite. Por lo tanto, teniendo en cuenta la inferencia, el *AP*, la documentación, y la implementación en entornos móviles, se decide establecer a YOLOv4 como primer modelo detección candidato para abordar el problema planteado. En cuanto a las tecnologías, se decide usar Google Colab para el entrenamiento, Tensorflow Lite

como framework de desarrollo móvil, y Android Studio como entorno de desarrollo integrado. La justificación se debe a que son herramientas gratuitas, con constantes actualizaciones, y con módulos que disminuyen el tiempo de desarrollo.

7.5 DESCRIPCIÓN DE YOLOv4

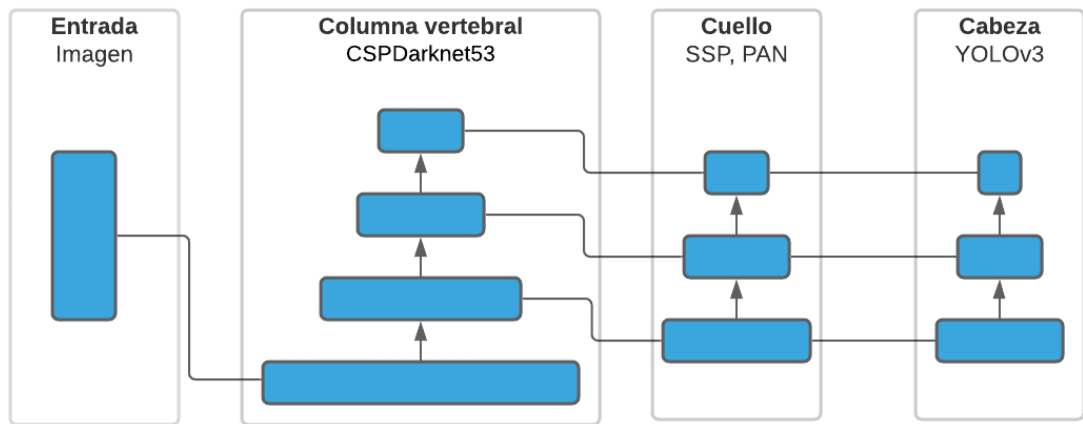


Figura 9. Representación gráfica de la arquitectura de YOLOv4.

Fuente: YOLOv4: Optimal Speed and Accuracy of Object Detection (Bochkovskiy et al., 2020)

YOLOv4 es el resultado de mejoras en versiones anteriores (YOLO, YOLOv2, YOLOv3), obteniendo una arquitectura que consta de una columna vertebral basada en CSPDarknet53 (Cross Stage Partial Network), un cuello con SPP y PAN, y una cabeza con YOLOv3 (Bochkovskiy et al., 2020). La columna vertebral de YOLOv4 se basa en una red neuronal convolucional preentrenada que busca disminuir la inferencia en los cálculos dada una imagen de entrada, dejando como resultado un modelo mejor optimizado (Wang et al., 2019). Por su parte, el cuello extrae mapas de características y gestiona la forma en la que se propaga la información en la red neuronal convolucional. Finalmente, la cabeza implementa un modelo de detección de objetos que toma los resultados obtenidos en el cuello y busca predecir los parámetros expuestos en el marco teórico (Bochkovskiy et al., 2020): probabilidad de clase, alto y ancho del cuadro delimitador, y coordenadas (x,y) del centro.

En adición a lo ya mencionado, YOLOv4 propone dos métodos nombrados como *Bag Of Freebies (BoF)* y *Bag Of Specials (BoS)*. El primero busca mejorar la precisión sin aumentar el costo de inferencia en un estado de producción, por ejemplo, cuando se busca implementar el modelo en una plataforma de desarrollo móvil. Mientras que el segundo busca mejorar significativamente la precisión en la detección a cambio de un aumento en el tiempo de inferencia. Lo formidable de *BoF* se encuentra en las técnicas de *data augmentation* implementadas, donde

destacan: *CutMix*, *Mosaic*, *MixUp*, *Blur*, *Flipping*, *Rotation*, *Cropping*, ajustes de brillo, contraste, tono, saturación y ruido. Por su parte, *BoS* enfatiza funciones de activación *Mish*, *CSP* y *MiWRC*.

Por otra parte, se debe mencionar que los modelos basados en YOLO cuentan con versiones ligeras que pueden ser implementadas en entornos de desarrollo con limitados recursos de hardware como pueden ser dispositivos móviles o placas programables como *Raspberry Pi* o *Jetson Nano*. Dentro de dichas versiones se encuentra YOLOv4-Tiny, que se basa en YOLOv4. Lo excepcional de YOLOv4-Tiny se encuentra en el tamaño de la red, con una columna vertebral CSP más reducida, y con 2 capas de YOLO, en lugar de 3 (Jiang et al., 2021) (Bochkovskiy et al., 2020). A pesar de dichas reducciones en las capas, YOLOv4-Tiny mantiene un alto rendimiento en detección y rapidez. Un punto de referencia para hacer una comparativa se encuentra en (Cai et al., 2021), donde demuestra que este modelo ligero es más rápido que modelos como SSD, YOLOv3, YOLOv4, y CenterNet. No obstante, es menos preciso que sus competidores, razón que tiene sentido al tener en cuenta su reducción en la arquitectura para que pueda ser ejecutada en dispositivos con bajos costos. Desde otro punto de vista, de acuerdo con con el repositorio oficial, YOLOv4-Tiny cuenta con un AP50 de 40.2, y una inferencia de 371 FPS al ser entrenado en una GPU GTX 1080 Ti, con el dataset MS COCO (30 clases), como se muestra a continuación.

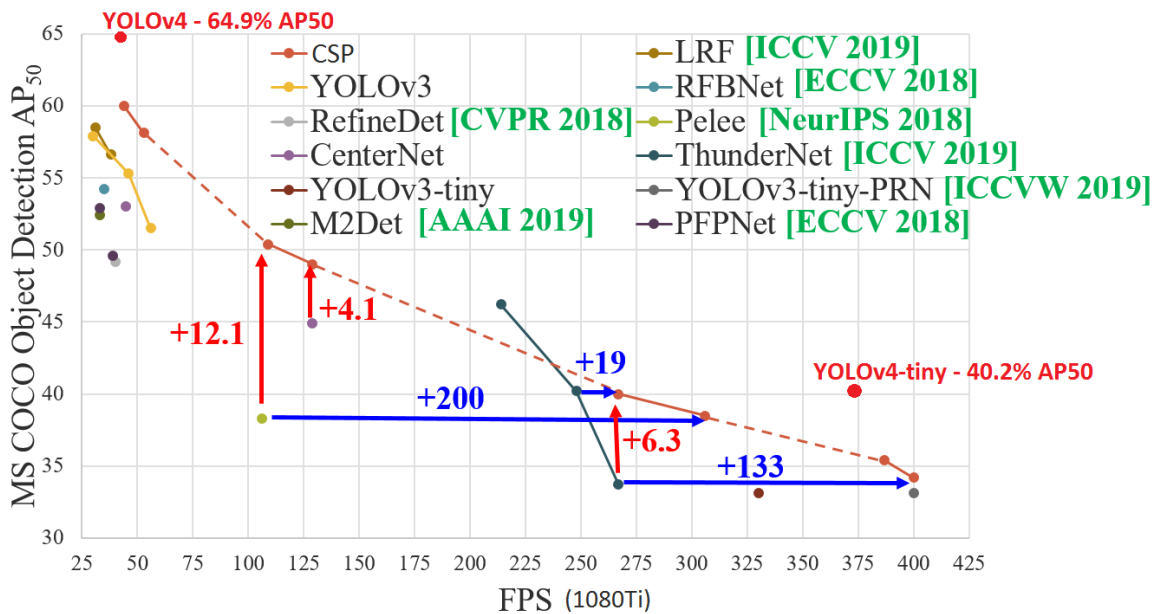


Figura 10. Comparativa YOLOv4-Tiny.
Fuente: YOLOv4 / Scaled-YOLOv4 / YOLO - Neural Networks for Object Detection (Windows and Linux version of Darknet) (AlexeyAB, 2020)

Finalmente, para escoger el modelo se hicieron dos entrenamientos rápidos, uno con YOLOv4, y otro con YOLOv4-Tiny. El entrenamiento se llevó a cabo en Google Colab, con las siguientes variables: `batch=64`, `subdivisions=16`, `max_batches=6000`, `random=1`. Cada una de las imágenes que entraba a la red se redimensionan a 416 x 416 píxeles. Adicionalmente, sólo se detectaron mazorcas de cacao sin identificar mazorcas enfermas, con una distribución por clase de 10-10-10. El objetivo de experimentar con estos entrenamientos es comprobar el rendimiento en detección y el tiempo de inferencia en un dispositivo móvil, para escoger un modelo que cumpla con los requerimientos móviles y sea viable para cumplir con el objetivo general. Los experimentos dejan como resultado que YOLOv4 obtiene un 76.76% de *mAP*, y YOLOv4-Tiny un 71,71%. Posteriormente, se hace una conversión e implementación en un dispositivo móvil con sistema operativo Android 7, en donde se observa que al modelo de YOLOv4 le toma aproximadamente 21 segundos en detectar mazorcas de cacao en una imagen, mientras que a la versión *tiny* le toma aproximadamente 2 segundos. Por lo tanto, a partir de la indagación de los modelos y la prueba mencionada, se decide usar YOLOv4-Tiny.

A partir de los datos obtenidos, se observa que YOLOv4 es el modelo más preciso en términos de *mAP*, superando a YOLOv4-tiny en 5.05%. No obstante, es el más lento en tiempo de inferencia cuando se prueba con imágenes individuales. Al tener en cuenta que el prototipo móvil funcional puede ser usado en una versión menor a Android 10, es necesario elegir un modelo que sea rápido y a la vez preciso. Es por ello, que, con base en los datos, se escoge a YOLOv4-Tiny como modelo para el entrenamiento de los próximos experimentos. A pesar de que dicha versión es un poco menos precisa, logra adaptarse mejor a las necesidades de dispositivos móviles con versiones previas de Android.

7.6 ENTRENAMIENTO

Una vez elegido YOLOv4-Tiny como modelo para el entrenamiento y la evaluación, se plantean 8 experimentos que buscan determinar si variables asociadas al equilibrio del conjunto de datos inciden sobre el *mAP*. De manera detallada, dichas variables aluden a: balanceo, *Data augmentation*, y Contador por clase. En síntesis, el balanceo hace referencia a una disminución en la cantidad de imágenes sanas para lograr un balance con las etiquetas restantes. Mientras que el uso de *Data augmentation*, utiliza imágenes generadas sintéticamente para aumentar el conjunto de datos de mazorcas enfermas, garantizando variabilidad y una disminución en el desbalance de etiquetas. Por su parte, el contador por clase es una variable presente en la red neuronal de YOLOv4-Tiny, nombrada originalmente como *counters_per_class*, que recibe n parámetros asociados al número de etiquetas por clase, donde n representa el número total de etiquetas; de acuerdo con la documentación de Darknet, dicha variable busca disminuir un poco el desbalance al tiempo que aumenta la precisión, sin embargo, no hay una evidencia clara sobre qué tanto incide en el entrenamiento.

Por consiguiente, se plantea el siguiente diseño factorial 2^k , donde k representa el número de factores a los que se les dará seguimiento en el entrenamiento para determinar si inciden en el mAP . Dado un número de factores $k = 3$, se plantean en total 8 experimentos que buscan dar respuesta a unas hipótesis, para posteriormente establecer una evaluación y elegir al mejor modelo de detección. El diseño factorial describe a continuación.

Tabla 133. Descripción de los experimentos

Balance	Data augmentation	Contador por clase
Sí	Sí	Sí
No	Sí	Sí
Sí	No	Sí
No	No	Sí
Sí	Sí	No
No	Sí	No
Sí	No	No
No	No	No

Con base en lo anterior se levantan las siguientes hipótesis:

1. La variable Balance incide en el mAP del modelo entrenado.
2. La variable *Data augmentation* incide en el mAP del modelo entrenado.
3. La variable contadores por clase incide sobre el mAP del modelo entrenado.

Para la fase de entrenamiento, conversión y adaptación se llevó a cabo el siguiente flujo de trabajo.

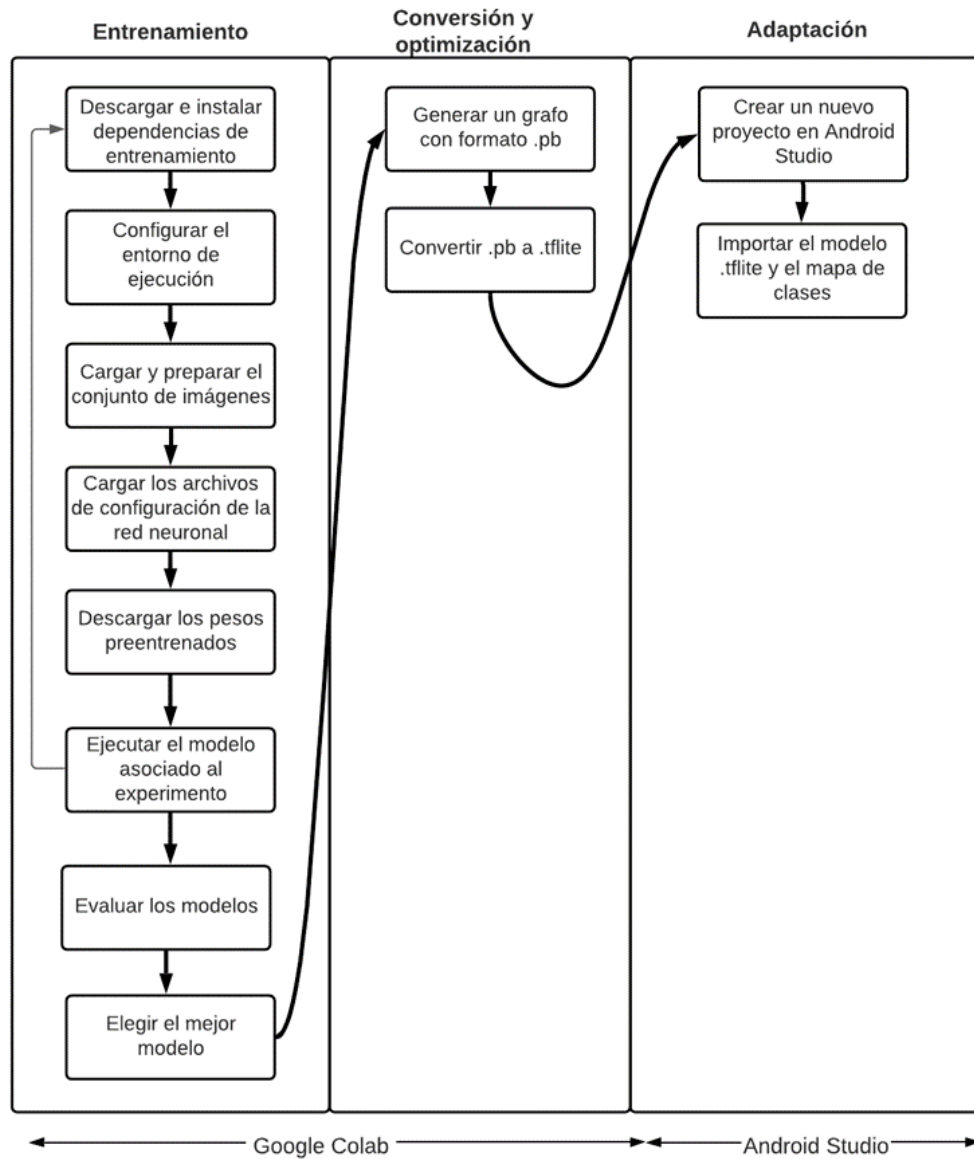


Figura 11. Flujo de trabajo en el entrenamiento de datos.

7.7 DEPENDENCIAS DEL ENTRENAMIENTO

Dentro de los requerimientos para ejecutar los entrenamientos en Google Colab, se debe contar con CMake, CUDA, OpenCV, cuDNN, y Darknet. CMake es un grupo de herramientas de código abierto y multiplataforma que sirve para controlar

automáticamente la compilación de software, generando un archivo denominado MakeFile que es utilizado para compilar el entorno de la máquina virtual (CMake, 2021). CUDA es una plataforma de computación paralela y un modelo de programación creados por NVIDIA que permiten acelerar las operaciones que se llevan a cabo en el proceso de entrenamiento (NVIDIA Developer, 2021), utilizando al máximo el potencial de la GPU remota proporcionada por Colab. NVIDIA también permite trabajar cuDNN, una librería enfocada en el entrenamiento de redes neuronales y que trabaja de forma conjunta con GPU y CPU, permitiendo un procesamiento y rendimiento de alta calidad, además permite realizar otras tareas como pooling, normalización, capas de activación entre otras (NVIDIA Developer, 2021). Otra de las principales dependencias del entrenamiento es OpenCV, una librería de código abierto que está compuesta por más de 2500 algoritmos, y que es el soporte fundamental en la aplicación de técnicas de visión por computador (OpenCV, 2021). Finalmente, Darknet es un framework de código abierto usado en redes neuronales, que permite el uso de GPU y CPU, con instrucciones detalladas para su implementación, desde la descarga de los pesos y las configuraciones respectivas de YOLOv4 y YOLOv4-Tiny, hasta la evaluación del modelo tanto en su rendimiento como en imágenes y videos (Bochkovskiy, 2021).

7.8 CONFIGURACIÓN DEL ENTORNO DE EJECUCIÓN

El entorno de ejecución proporcionado por Google Colab es muy variado, ya que dependiendo del tiempo en el que se ejecute o reinicie la máquina, este puede cambiar para ofrecer un servicio gratuito. Dentro de las GPU más usadas en los entrenamientos se encuentran: K80, T4, P4 y P100 de NVIDIA (Google Colaboratory, 2021). Para hacer uso de dicha GPU se accede a la sección de configuración de entorno y se procede a conectar la máquina. Posteriormente, se edita el Makefile para que reconozca la GPU, habilite cuDNN, y al tiempo adapte las librerías de OpenCV.

```
%cd /content/darknet/  
!sed -i 's/OPENCV=0/OPENCV=1/' Makefile  
!sed -i 's/GPU=0/GPU=1/' Makefile  
!sed -i 's/CUDNN=0/CUDNN=1/' Makefile  
!sed -i 's/CUDNN_HALF=0/CUDNN_HALF=1/' Makefile  
!sed -i 's/LIBSO=0/LIBSO=1/' Makefile
```

Figura 12. Configuración del entorno de ejecución.
Fuente: Los autores

7.9 CARGA Y PREPARACIÓN DEL CONJUNTO DE IMÁGENES

Los entrenamientos propuestos en el diseño factorial usan en su totalidad una partición de imágenes aproximadamente del 80% para el entrenamiento y 20% para la validación, haciendo referencia al conjunto de datos consolidado. En donde surgen dos tipos de datos cargados, balanceados y no balanceados. Los balanceados mantienen el número de imágenes por clase (87-83-90 para el entrenamiento y 20-22-22 para validación). Mientras que los desbalanceados contienen más imágenes de mazorcas sanas en comparación con enfermas (87-83-306 para entrenamiento y 20-22-83 para validación). A esto se le debe añadir que dependiendo del uso de *data augmentation*, el número de etiquetas aumentará para la categoría monilia y fitóftora mientras que la sana se mantiene. No obstante, las imágenes generadas con *data augmentation* no interfieren en la validación y se usan en segundo plano para disminuir el desbalance en el entrenamiento. Para detallar el proceso de *data augmentation*, se generaron 510 imágenes las cuales contenían mazorcas pertenecientes a las 3 categorías; sin embargo, en su mayoría eran de la categoría enferma. Dichas imágenes se redimensionan a 416 x 416 píxeles, y se les aplica las siguientes técnicas aleatoriamente: rotación a 90°, cropping 0-24%, rotación -15° grados a 15°, y ruido 0%-2%.



Figura 13. Ejemplo de una imagen con *data augmentation*.

7.10 CONFIGURACIÓN DE LA RED NEURONAL

El entrenamiento está compuesto por tres archivos principales: un `.data`, un `.names`, y un `.cfg`. Inicialmente el archivo `'obj.data'` contiene las rutas que describen dónde serán guardados los resultados de cada entrenamiento, los archivos usados para el `train` y para el `test`, el nombre de las clases y el número de clases a entrenar.

Por otra parte en el archivo `'obj.names'` se encuentra el mapa de clases, que contiene las clases, `fito` (haciendo referencia a `fitóftora`), `monilia` y `sana`, que fueron las etiquetas acordadas en el análisis de los conjuntos de datos disponibles.

Finalmente, el archivo `.cfg` que cuenta con 281 líneas de código dónde se encuentra la configuración del entrenamiento que contiene los parámetros e hiperparámetros asociados a la red neuronal convolucional. Esto incluye los filtros, el tamaño de las imágenes, el número de *batches*, el número de capas de YOLO (que en este caso son 2), el contador por clase, los pasos, las subdivisiones entre otros. Para garantizar una estandarización de los 8 entrenamientos, se definen los siguientes parámetros principales: `batch = 64`; `subdivisions = 16`; `max_batches = 6000`; `steps = 4800,5400`; `width=416`; `height=461`; `classes=3`; `random=1`.

A continuación, se presenta información detallada sobre algunos parámetros más destacados en la red neuronal (Bochkovskiy, 2021).

- *batch*: número de imágenes que serán procesadas en un lote (*batch*).
- *subdivisions*: número de mini *batches* en un *batch*.
- *mini_batch*: $\frac{batch}{subdivisions}$
- *width*: anchura de la imagen que entra a la red.
- *height*: altura de la imagen que entra a la red.
- *max_batches*: máximo número de iteraciones o *batches* que tendrá el entrenamiento.
- *steps*: número de iteraciones donde la tasa de aprendizaje varía.
- *classes*: número de etiquetas que contiene el entrenamiento (`fitóftora`, `monilia`, `sana`).
- *random*: variable que recibe un valor booleano (0,1), y cuando se activa redimensiona las imágenes a la dimensión especificada en los parámetros *height* y *width*.

7.11 PESOS PREENTRENADOS

Una vez terminada la configuración, se debe mencionar que los entrenamientos comenzaron basados en pesos preentrenados que provee el repositorio de Darknet. Dichos pesos reciben el nombre de *'yolov4-tiny.conv.29'* (Bochkovskiy, 2021). Estos hacen que el proceso de entrenamiento no sea tan extenso en comparación con los pesos inicializados aleatoriamente.

7.12 EJECUCIÓN DEL ENTRENAMIENTO

Con base en lo anteriormente descrito, se ejecuta la fase de entrenamiento con el siguiente comando: *'!./darknet detector train data/obj.data cfg/yolov4-tiny-custom.cfg yolov4-tiny.conv.29 -dont_show -mAP'* (Bochkovskiy, 2021). Donde *'./darknet'* representa un script que contiene las instrucciones que garantizan el ajuste de pesos y el aprendizaje automático. Dichas instrucciones reciben como entrada los archivos de configuración de la red neuronal y los pesos preentrenados. Adicionalmente, se especifican las banderas *'-dont_show'* y *'-mAP'*. La primera previene que surja un error debido a la generación automática de gráficas de manera iterativa, sin dicha bandera no es posible hacer el entrenamiento. Mientras que la segunda, ayuda a visualizar el *mAP* con cada iteración en la consola. El tiempo estimado de los entrenamientos varía dependiendo del conjunto de datos cargado y las máquinas que asigna Colab. Para datos balanceados y/o con *data augmentation* se estima un aproximado de 6.5 horas. Mientras que para datos desbalanceados y/o sin *data augmentation* el tiempo aumenta a 9 horas aproximadamente.

Los entrenamientos propuestos tienen definido un número máximo de *batches* *max_batches=6000*. Cada 1000 iteraciones se guarda una copia de respaldo que incluyen archivos con la extensión *.weights*. Adicionalmente, se debe tener en cuenta que se genera un archivo a partir de la iteración 100 que lleva el registro de los últimos pesos. En últimas, cada entrenamiento genera 9 copias de respaldo que contienen los pesos en una determinada iteración, donde 8 de ellas actúan como punto de comparación para que se genere un archivo denominado *'yolov4-tiny-custom_best.weights'* con los pesos que generan el mejor rendimiento en términos de *mAP*. Por lo tanto, los experimentos contemplados dejan como resultado un total de 72 archivos para comparar, donde finalmente se escogen a los 8 mejores.

7.13 EVALUACIÓN DEL MODELO

Para evaluar los resultados obtenidos, se plantea un análisis de varianza ANOVA que busca determinar si las variables Balance, *Data augmentation*, y Contadores por clase, inciden en la precisión medida como *mAP*. En la siguiente tabla se muestran los experimentos y sus respectivos resultados de *mAP*, *precision*, *recall* y *f1-score*.

Tabla 144. Resultados de los experimentos.

Experimento	Balance	<i>Data augmentation</i>	Contador por clase	<i>mAP_best</i>	<i>precision</i>	<i>recall</i>	<i>f1_score</i>
1	Sí	Sí	Sí	82,36	0,53	0,79	0,64
2	No	Sí	Sí	77,46	0,66	0,75	0,7
3	Sí	No	Sí	71,29	0,66	0,73	0,69
4	No	No	Sí	67,38	0,66	0,73	0,69
5	Sí	Sí	No	80,09	0,76	0,76	0,76
6	No	Sí	No	73,58	0,69	0,71	0,7
7	Sí	No	No	72,01	0,73	0,71	0,72
8	No	No	No	64,43	0,72	0,69	0,7

Basados en los resultados obtenidos y buscando dar respuesta a las hipótesis planteadas, se realiza un análisis de varianza cuyos resultados se muestran a continuación.

Tabla 155. Análisis de varianza ANOVA

Fuente	GL	SC Ajuste	MC Ajuste	Valor F	Valor P
Balance	1	65.551	65.551	44.17	0.003
Augmentation	1	184.128	184.128	124.06	0.000
Counters	1	8.778	8.778	5.91	0.072
Error	4	5.937	1.484		
Total	7	264.394			

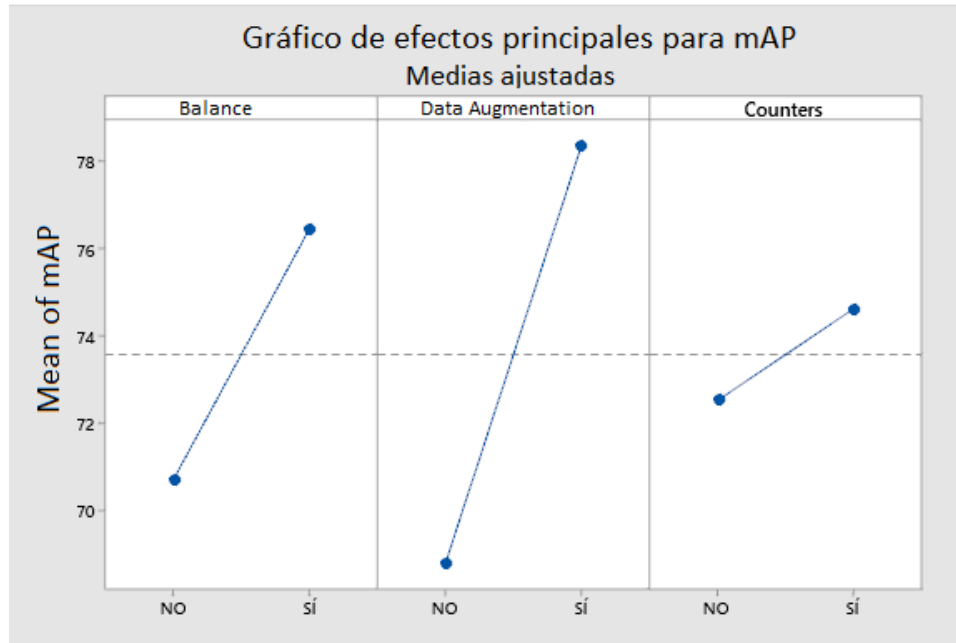


Figura 14. Gráfico de efectos principales para mAP .

Este análisis demuestra que las variables *Data augmentation* y *Balance* con sus respectivos resultados de valor $P(0.000$ y $0.003)$ tienen incidencia en el mAP , mientras que la variable *Contadores por clase no*. Con el propósito de detallar las variables que ayudan a evaluar al modelo elegido, se hace un planteamiento matemático de mAP , $recall$, y $precision$. Se tiene que $precision$ representada como $p(1)$, y $recall$ como $r(2)$ comprenden algunas de las variables presentes en una matriz de confusión (TP, FP, FN), donde TP hace referencia a cuadros delimitadores contados como verdaderos positivos, FP a los falsos positivos, y FN a los falsos negativos (Bochkovskiy, 2021). Asimismo, $AP(3)$ es una métrica que determina la precisión promedio por clase basada en las métricas de $recall$, $precision$, e IoU (Everingham et al., 2010). Esto da como resultado al mAP , una media aritmética de la precisión promedio (AP) (Beitzel et al., 2009). En esta evaluación, para calcular $recall$ y $precision$ se considera un umbral de 0.25. Mientras que para mAP se tiene en cuenta un IoU de 0.5.

$$p = \frac{TP}{TP + FP} \quad (1)$$

$$r = \frac{TP}{TP + FN} \quad (2)$$

$$AP = \frac{1}{n} \sum_{r \in \{0, 0.1, \dots, 1\}} P_{interpolada}^{(r)} \quad (3)$$

Donde, la precisión (p) en cada nivel de *recall* (r) se interpola tomando la precisión máxima medida para un método para el cual el *recall* correspondiente excede r :

$$p_{interpolada}^{(r)} = \max p(r) \quad r: r \geq r$$

Dado el cuadro delimitador predicho y el verdadero, se establece la siguiente relación entre la unión e intersección de los cuadros.

$$IoU = \frac{\text{área de la intersección}}{\text{área de la unión}} \quad (4)$$

$$mAP = \frac{1}{n} \sum_{i=1}^n AP_n \quad (5)$$

Finalmente, se reporta que el mejor modelo para convertir es el del experimento 5, con un *mAP* de 80.09%, 0.76 de *recall* y 0.76 de *precision*. Lo excepcional de este modelo además del *mAP*, es que reporta la menor tasa de falsos positivos y falsos negativos.

7.14 CONVERSIÓN

Dados en este punto, en el que se tiene al mejor modelo para detectar y contar mazorcas de cacao, se debe hacer una conversión de pesos con formato *.weights* a formato *.pb*. Y luego, convertir de formato *.pb* a *.tflite*. Esta conversión garantiza una compatibilidad con el *framework Tensorflow Lite* en un entorno de desarrollo móvil. Finalmente, a los pesos convertidos se les aplica las instrucciones expuestas en el repositorio de (Hùng, 2020) el cual provee un conversor ideal para modelos basados en YOLO. Dicha conversión garantiza la optimización del modelo y acelera la velocidad de inferencia en la implementación.

7.15 ADAPTACIÓN DEL MODELO

Finalmente, para adaptar el modelo de aprendizaje de máquina para la detección y conteo de mazorcas de cacao, se carga el modelo convertido a la carpeta *assets* en un proyecto de Android Studio. En donde se toma como punto de partida para el desarrollo del prototipo, el repositorio de (Hùng, 2020).

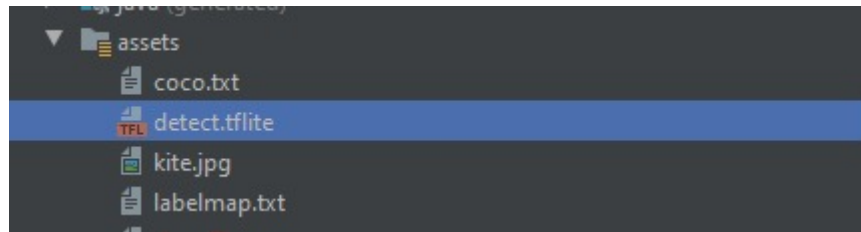


Figura 15. Modelo adaptado en Android Studio.

8 DISEÑO Y DESARROLLO DEL PROTOTIPO

Una vez terminados los procesos anteriores que involucran el análisis de las tecnologías y la adaptación del modelo de visión por computador, es necesario realizar el prototipo móvil que pretende reducir los costos y disminuir el grado de complejidad del desarrollo. A través de la implementación de un dispositivo móvil, se busca obtener mayor portabilidad y accesibilidad. Por lo que el presente capítulo, es una evidencia del proceso de prototipado rápido elegido para el desarrollo del prototipo funcional, el cual está compuesto de diferentes etapas como la recolección de requisitos, el diseño rápido, la construcción del prototipo y su respectiva evaluación. Cada una de estas etapas, se muestran a continuación con sus respectivos productos de software.

8.1 RECOLECCIÓN DE REQUISITOS

Para la recolección de requerimientos, se contó con la participación del instructor del SENA Néstor Edsgardo Nieves Peña, profesional en los cultivos de cacao en Santander, quien planteó los principales problemas que enfrentan los cacaoteros en las labores de conteo y estimación de la producción. Basados en esto y en el planteamiento del problema se identifican los siguientes requerimientos para el desarrollo de la aplicación en Android. De esta recolección surgen los anexos A, B y C.

8.1.1 Requerimientos funcionales

- **Indicar la variedad de cacao:** antes del proceso de detección, el usuario seleccionará la variedad de cacao para la cual quiere estimar la producción. De momento, sólo se contará con cuatro variedades: CCN51, FEC, FEAR 5, FSV41.
- **Cargar fotos:** la aplicación permitirá cargar fotos de mazorcas de cacao, ya sea desde la galería del usuario o la cámara trasera y predeterminada del celular.
- **Cargar fotogramas en tiempo real:** permitirá tomar un conjunto de fotogramas en tiempo real para llevar a cabo el proceso de estimación rápida.
- **Detectar mazorcas de cacao:** a partir de una imagen o un conjunto de fotogramas, la aplicación detectará las mazorcas de cacao mediante un proceso de clasificación y localización, que dejará como resultado una imagen con cuadros delimitadores dibujados en las respectivas mazorcas de cacao.

- **Guardar información de la detección:** la persona encargada de usar la aplicación, podrá indicar si quiere guardar la información obtenida de la detección (conteo detallado por sanidad), o si por el contrario no lo quiere.
- **Visualizar el conteo de mazorcas de cacao:** una vez detectadas las mazorcas de cacao, el usuario podrá visualizar un conteo total, y un conteo específico por sanidad.
- **Estimación individual por tipo de detección:** El usuario podrá informarse acerca de la estimación de granos de cacao seco que se calcula al operar el número de mazorcas sanas, con el respectivo índice de mazorcas, basado en el tipo de la variedad que ha escogido.
- **Visualizar historial e informe mensual de las detecciones:** el usuario podrá ver un historial con la información de las detecciones que ha realizado previamente, junto con una gráfica que muestra el informe mensual de la estimación de la producción, medida en Kilogramos de granos de cacao seco por variedad.

8.1.2 Requerimientos no funcionales

- **Tiempo de inferencia:** se espera hacer un conteo por imagen en el menor tiempo posible. Mientras que, para conteo rápido, se espera bajo tiempo de inferencia.
- **Aplicación móvil offline:** la aplicación funcionará sin conectividad a internet, usando sólo el almacenamiento local del dispositivo móvil.

8.2 DIAGRAMA DE COMPONENTES

Modelo Yolo: Es obtenido a partir de una conversión de formato *.weights* a *.tflite*, y de este depende el proceso de clasificación, detección y conteo en la Aplicación.

TensorflowLite: Es un framework de desarrollo móvil, que se encarga de retornar valores entendibles para el usuario del proceso de detección a partir del preprocesamiento y estandarización de las imágenes o fotogramas que pasan como entrada al modelo de detección de objetos.

Room: Es una librería de Base de Datos, que sirve en modo *offline* y se encarga de almacenar la información correspondiente a los datos de estimación.

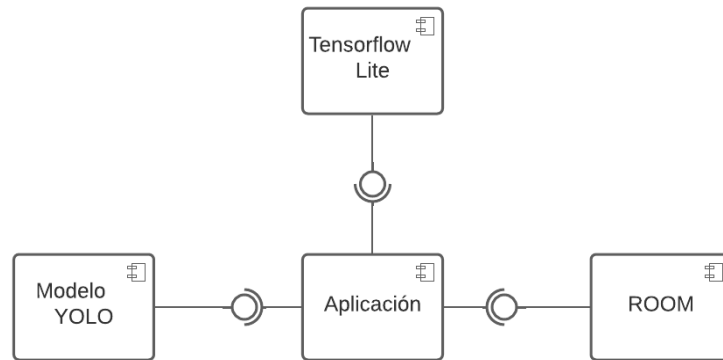


Figura 16. Diagrama de componentes.

8.3 DIAGRAMA DE CASOS DE USO

El planteamiento del siguiente diagrama busca explicar el funcionamiento del sistema, contemplando actores con sus respectivas relaciones entre casos de uso. En este proyecto, solo se cuenta con un actor que es usuario, quien puede elegir la variedad de cacao, detectar mazorcas ya sea a partir de imágenes o de un conjunto de *frames* (cuando se deja la cámara abierta en tiempo real), y estimar la producción agrícola. Para llevar a cabo el proceso de detección, es necesario que el sistema logre clasificar el estado de sanidad de las mazorcas, para posteriormente generar un conteo de los resultados obtenidos, y guardar la información asociada a la detección.

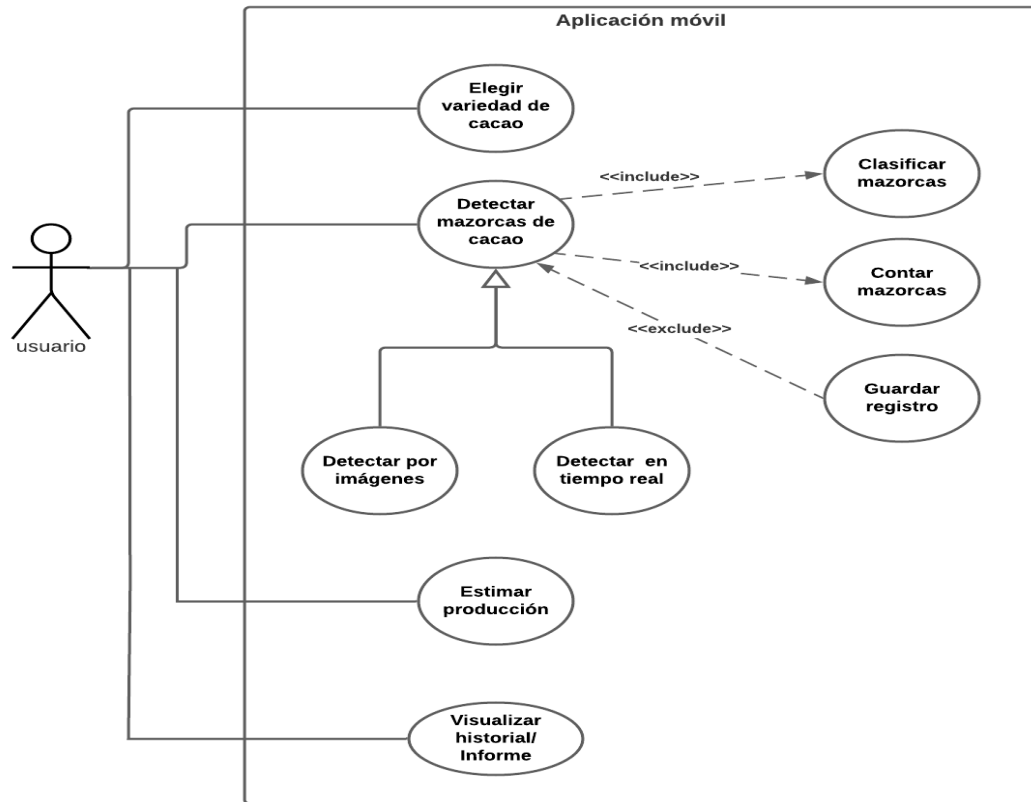


Figura 16. Diagrama de casos de uso.

8.4 DIAGRAMA DE SECUENCIA

El siguiente diagrama, pretende detallar los principales elementos presentes en la aplicación móvil y la manera en la que se relacionan. De modo que el usuario interactúa con una interfaz de captura de video o imagen, que a su vez se comunican con la API de *Tensorflow Lite*, un interpretador y un *Multibox Tracker*. Cabe resaltar, que cada una de las líneas horizontales y en bucle de la siguiente imagen, representan un flujo de datos o la declaración de un método.

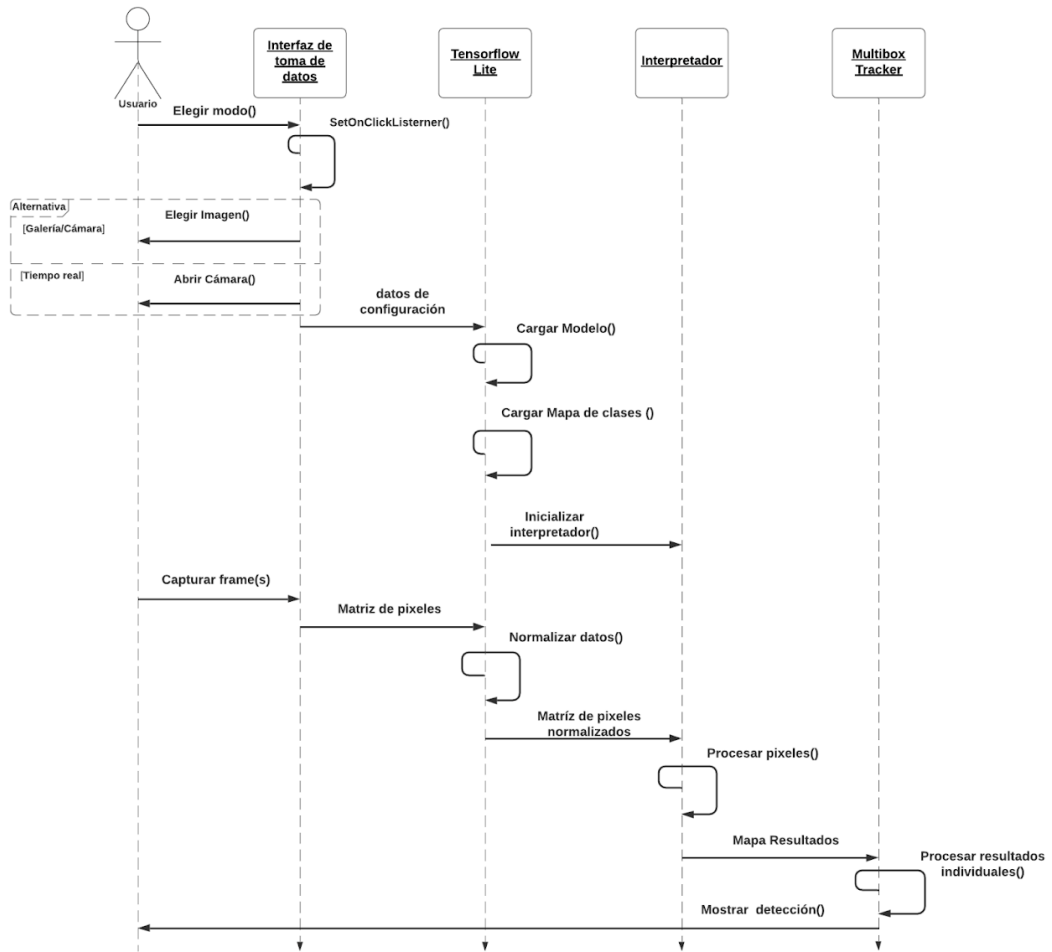


Figura 17. Diagrama de secuencia.

8.5 DISEÑO

Con base en la ingeniería de requerimientos, en el inicio se establece el siguiente diseño basado en un menú inferior, después en el refinamiento de prototipo, se plantea el diseño de botones que se encuentra en el anexo D para el desarrollo de la aplicación móvil.

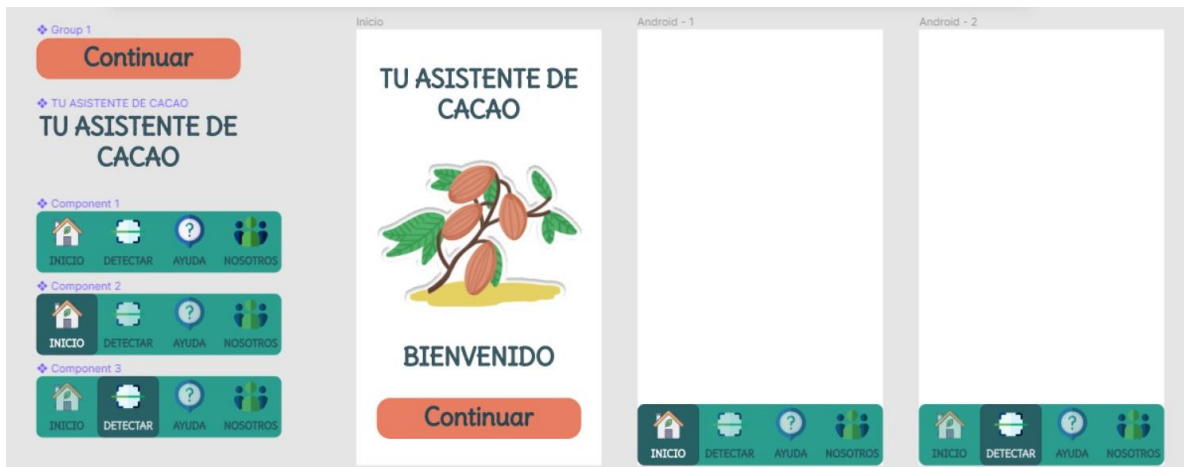


Figura 17. Diseño inicial.

8.6 DESARROLLO DEL PROTOTIPO

El prototipo móvil se desarrolla gracias al entorno de desarrollo Android Studio versión 4.1.2, lenguaje de programación Java, XML y SQL. Para llevar a cabo la implementación del almacenamiento local se usa la persistencia de datos *Room*, la cual garantiza un fácil manejo de datos de manera rápida, conservando la integridad. Dentro de las dependencias y recursos hace uso del permiso a la cámara y al almacenamiento. En adición, para mejorar la interacción en la muestra de resultados se implementa la librería *MPAndroidChart* que facilita el desarrollo y despliegue de recursos gráficos asociados con datos.

8.7 EVALUACIÓN DEL PROTOTIPO

Debido a la actual pandemia COVID-19 y a la situación social que vive Colombia en estos momentos, no fue posible realizar una evaluación presencial en el municipio de El Playón. No obstante, se plantean unos casos de pruebas presentes en el anexo F. Con base en el desarrollo y los casos de pruebas se entabla una comunicación de manera remota con Néstor Nieves, quien valida el diseño y la utilidad del prototipo.

9 DISCUSIÓN

A pesar de la cantidad de artículos centrados en la detección de frutos, pocos de los consultados contemplan el uso de un teléfono móvil, y por el contrario proponen el uso de robots o vehículos autónomos que agregan mayor costo y menor portabilidad al momento de realizar la detección, que contempla una clasificación, conteo y estimación. Es por ello que el presente proyecto es una vía para solucionar lo anteriormente mencionado a través de un teléfono celular que adapta un modelo de aprendizaje profundo. La implementación de una red neuronal convolucional con una arquitectura de YOLOv4-Tiny permite la obtención de una precisión promedio mayor al 50% sin requerir de miles de imágenes, contrario a lo propuesto por Bresilla et al. (2020) Sin embargo, se confirma que el uso de técnicas de *data augmentation* incrementa la precisión del modelo y diversifica el dataset, idea planteada por el mismo autor y Wan & Goudos (2020). Además, con el uso de un modelo basado en YOLO se rectifica las proposiciones realizadas por los autores Tian et al. (2019), Koirala et al. (2019), Kang & Chen (2020) y Wan & Goudos (2020) las cuales exponen principalmente que YOLO tiene un menor tiempo promedio de inferencia y un alto rendimiento ante imágenes desconocidas para el modelo.

Por otro lado, la toma de datos aplicando variación entre los puntos de toma y manteniendo las mismas características entre los diferentes dispositivos de captura permiten que las fotografías estén estandarizadas, pero con variabilidad en la toma. Por lo tanto, implica que se requiera de menos recursos tanto en tiempo de inferencia como en procesamiento de las imágenes, similar a lo planteado por Liu et al. (2019). A esto se le debe agregar que la variabilidad en la toma previene el sobreentrenamiento y logra generalizar de mejor manera el conocimiento.

Se verifica que el uso de YOLOv4-Tiny requiere de menos coste computacional y tiene mayor rapidez, accesibilidad y portabilidad como lo propone Buzzy et al. (2020) Adicionalmente el modelo tiene como ventaja menor tiempo de entrenamiento comparado con YOLOv4, lo cual permite ahorrar tiempo en dicha tarea y deja abierta la posibilidad de realizar un proceso de calibración a las variables de entrenamiento. A pesar de la diferencia notable entre el tiempo de entrenamiento de YOLOv4 y YOLOv4-Tiny, el mAP de este último no presenta una diferencia significativa y se adapta a los requerimientos del sistema.

En este trabajo también se analiza el papel de la variable *counters_per_class*, el uso de *data augmentation* y el balance de datos con sus respectivas incidencias en el mAP del modelo a través del análisis de varianza ANOVA. Los resultados de la evaluación demuestran que la variable *counters_per_class* no afecta significativamente los resultados del mAP en el modelo resultante. Mientras que el uso de *data augmentation*, y balanceo de imágenes logra incrementar la precisión y disminuir un poco el tiempo de entrenamiento.

También se debe mencionar que se identifican algunas limitaciones en el prototipo móvil funcional desarrollado. Por una parte, se debe considerar que el conjunto de datos propio no contiene instancias de mazorcas enfermas etiquetadas, debido a que no se contó con ayuda profesional. Esto dificulta que el modelo detecte mazorcas muy lejanas con un estado inicial de enfermedad o que no sea muy visible para la fuente de toma. Además, la funcionalidad de la aplicación nombrada como detección rápida cumple con su labor, pero es limitada, ya que como consecuencia de la última actualización de *Tensorflow Lite*, se han presentado problemas para hacer un multi seguimiento de objetos conocido como *MOT (Multiple Object Tracking)*. El uso de *MOT*, reduciría el flujo de datos, y podría automatizar el conteo.

10 CONCLUSIONES

En este trabajo se presenta un prototipo móvil que proporciona una herramienta de gran utilidad para el cultivo de cacao, ya que realiza una identificación, clasificación y conteo de mazorcas de cacao en distintos estados de sanidad, lo que contribuye a la mejora del proceso de toma de decisiones en cuanto a la sanidad, manejo del cultivo y buenas prácticas. Además, brinda una estimación de producción del cultivo que se traduce en kg de granos de cacao seco calculados a partir del índice de mazorca (*IM*). Esto permite el acceso a información de la cantidad de frutos con sus respectivos estados de sanidad. Dicha información le brinda al usuario una idea de lo que tiene y lo que tendrá siempre y cuando aplique buenas prácticas en sus plantaciones. Adicionalmente, la información proporcionada, podría ayudar al usuario a estimar por cuenta propia los insumos agropecuarios, mano de obra para la cosecha final, y ganancias a futuro.

Por otra parte, pese a que otras investigaciones suelen hacer un conteo simple para la estimación de la producción sin contemplar las variables de sanidad o madurez, este proyecto proporciona mayor información y utilidad para dicha estimación, porque contempla algunos problemas fitosanitarios a los cuales se enfrentan los agricultores en Colombia. El desconocimiento de las enfermedades en plantaciones de cacao termina generando un impacto negativo en la producción. Es por ello que conocer el estado de sanidad es importante, porque de nada sirve contar mazorcas y contemplarlas en el estado de producción si al final varias de ellas están enfermas y pueden afectar a futuras cosechas.

Asimismo, se demuestra que las versiones ligeras de los modelos YOLO pueden ser calibradas para alcanzar un alto grado de precisión al tiempo que garantiza una alta velocidad de inferencia. Esto quiere decir que, variando algunos parámetros e hiperparámetros de la red neuronal convolucional se puede mejorar la precisión sin importar la profundidad, pero la velocidad de inferencia suele mantenerse ya que es muy dependiente del número de capas y neuronas presentes en la red. A esto se le debe sumar que el uso de *Data augmentation*, y el control del balance de etiquetas pueden contribuir a un aumento en la precisión con menores tiempos de entrenamiento. Por lo tanto, el uso de YOLOv4-Tiny no representa una pérdida significativa en precisión, al contrario, es una ventaja que contempla los requerimientos de hardware de los dispositivos móviles, al tiempo que considera los casos de uso que involucran al usuario final.

Finalmente, el prototipo se caracteriza por su portabilidad, accesibilidad, bajos requerimientos de software y el no requerimiento de conectividad a internet, ya que la aplicación no requiere de actualizaciones ni servicios de almacenamiento

basados en la nube. Esto es importante para trabajos futuros, en donde la herramienta podría ser usada por cacaocultores que viven en zonas con baja conectividad a internet. Lo descrito permitiría reducir la brecha que enfrentan los cacaoteros en el acceso a tecnologías de la información y comunicación.

11 REFERENCIAS

- AlexeyAB/darknet: YOLOv4 / Scaled-YOLOv4 / YOLO - Neural Networks for Object Detection (Windows and Linux version of Darknet). (2020). Retrieved May 5, 2021, from <https://github.com/AlexeyAB/darknet>
- Ali Süzen, A., Gürfidan, R., Kayaalp, K., & Ali Şimşek, M. (2020). Information Technology and Computer Science. *Information Technology and Computer Science*, 2, 18–23. <https://doi.org/10.5815/ijitcs.2020.02.02>
- Android Developers. (2020). Android Studio. <https://developer.android.com/studio/>
- Android Studio Developer. (2020). Introducción a Android Studio. https://developer.android.com/studio/intro#top_of_page
- Argout, X., Salse, J., Aury, J.-M., Gultinan, M. J., Droc, G., Gouzy, J., Allegre, M., Chaparro, C., Legavre, T., Maximova, S. N., Abrouk, M., Murat, F., Fouet, O., Poulain, J., Ruiz, M., Roguet, Y., Rodier-Goud, M., Barbosa-Neto, J. F., Sabot, F., ... Lanaud, C. (2011). The genome of *Theobroma cacao*. *Nature Genetics*, 43(2), 101–108. <https://doi.org/10.1038/ng.736>
- Bargoti, S., & Underwood, J. P. (2017). Image Segmentation for Fruit Detection and Yield Estimation in Apple Orchards. *Journal of Field Robotics*, 34(6), 1039–1060. <https://doi.org/10.1002/rob.21699>
- Beitzel, S. M., Jensen, E. C., & Frieder, O. (2009). MAP. *Encyclopedia of Database Systems*. (2009). In *Encyclopedia of Database Systems*. Springer US. <https://doi.org/10.1007/978-0-387-39940-9>
- Bhardwaj, K. K., Banyal, S., & Sharma, D. K. (2019). Chapter 7 - Artificial Intelligence Based Diagnostics, Therapeutics and Applications in Biomedical Engineering and Bioinformatics (V. E. Balas, L. H. Son, S. Jha, M. Khari, & R. B. T.-I. of T. in B. E. Kumar (eds.); pp. 161–187). Academic Press. <https://doi.org/10.1016/B978-0-12-817356-5.00009-7>
- Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (n.d.). YOLOv4: Optimal Speed and Accuracy of Object Detection. Retrieved May 1, 2021, from <https://arxiv.org/abs/2004.10934>.
- Bresilla, K., Perulli, G. D., Boini, A., Morandi, B., Grappadelli, L. C., & Manfrini, L. (2020). Comparing deep-learning networks for apple fruit detection to classical hard-coded algorithms. *Acta Horticulturae*, 1279, 209–216. <https://doi.org/10.17660/ActaHortic.2020.1279.31>

- Burger, W., & Burge, M. J. (2016). Digital image processing: an algorithmic introduction using Java. Springer.
- Buzzy, M., Thesma, V., Davoodi, M., & Mohammadpour Velni, J. (2020). Real-Time Plant Leaf Counting Using Deep Object Detection Networks. In *Sensors* (Vol. 20, Issue 23). <https://doi.org/10.3390/s20236896>
- Cai, Y., Li, H., Yuan, G., Niu, W., Li, Y., Tang, X., & Wang, Y. (2021). YOLObile: Real-Time Object Detection on Mobile Devices via Compression-Compilation Co-Design. www.aaai.org
- Cecotti, H., Rivera, A., Farhadloo, M., & Pedroza, M. A. (2020). Grape detection with convolutional neural networks. *Expert Systems with Applications*, 159, 113588. <https://doi.org/10.1016/j.eswa.2020.113588>
- CMake. (2021). Retrieved May 5, 2021, from <https://cmake.org/>
- Colorizer. (2020). <http://colorizer.org/>
- CUDA Zone | NVIDIA Developer. (2021). Retrieved May 5, 2021, from <https://developer.nvidia.com/cuda-zone>
- De La Cruz Medina, J., Vargas Ortiz, M., Del Angel Coronel, O. (2011). CACAO: Operaciones Poscosecha. <http://www.fao.org/3/a-au995s.pdf>
- Deng, R., Jiang, Y., Tao, M., Huang, X., Bangura, K., Liu, C., Lin, J., & Qi, L. (2020). Deep learning-based automatic detection of productive tillers in rice. *Computers and Electronics in Agriculture*, 177, 105703. <https://doi.org/10.1016/j.compag.2020.105703>
- Dostert, N., Asunción Cano, José Roque., La Torre, María I., W. M. (2011, October). Hoja botánica: Cacao. http://www.botconsult.com/downloads/Hoja_Botanica_Cacao_2012.pdf
- Everingham, M., Luc, P., Gool, V., Christopher, D., Williams, K. I., Winn, J., Zisserman, A., Everingham, M., Van Gool, L., Leuven, K. U., Williams, B. C. K. I., Winn, J., & Zisserman, A. (2010). The PASCAL Visual Object Classes (VOC) Challenge. Retrieved May 5, 2021, from <http://www.flickr.com/>
- Factors influencing smallholder cocoa production: a management analysis of behavioural decision-making processes of technology adoption and application. (1996). Retrieved September 4, 2020, from <https://library.wur.nl/WebQuery/wurpubs/35987>
- FAOSTAT. (n.d.). Retrieved April 13, 2021, from <http://www.fao.org/faostat/en/#data/QC/metadata>

- FEDECACAO y los Cacaocultores, le apuestan a la certificación en Buenas Prácticas Agrícolas. (Abril de 2019). Colombia Cacaotera, p.6. Recuperado de www.fedecacao.com.co
- Fedecacao. (2005). CARACTERIZACIÓN FÍSICOQUÍMICA Y BENEFICIO DEL GRANO DE CACAO (*Theobroma cacao* L.) EN COLOMBIA. https://www.fedecacao.com.co/portal/images/recourses/pub_doctecnicos/fedecacao-pub-doc_09B.pdf
- Fountain, A., & Hütz-Adams, F. (2015). 2015 Cocoa Barometer (USA edition). Retrieved from Barometer Consortium website: <http://evols.library.manoa.hawaii.edu/handle/10524/48573>
- Fountain, A., & Hütz-Adams, F. (2018). Cocoa Barometer 2018. Retrieved from VOICE Network website: <https://www.voicenetwork.eu/wp-content/uploads/2019/07/2018-Cocoa-Barometer.pdf>
- Fountain, A., & Hütz-Adams, F. (2020). Cocoa Barometer 2020. Retrieved from VOICE Network website: <https://www.voicenetwork.eu/wp-content/uploads/2021/03/2020-Cocoa-Barometer-EN.pdf>
- Franceschetti, D. R. (Ed.). (2018). Principles of Robotics & Artificial Intelligence.
- Fu, L., Gao, F., Wu, J., Li, R., Karkee, M., & Zhang, Q. (2020, October 1). Application of consumer RGB-D cameras for fruit detection and localization in field: A critical review. *Computers and Electronics in Agriculture*, Vol. 177, p. 105687. <https://doi.org/10.1016/j.compag.2020.105687>
- Gayi, S. K., & Tsowou, K. (2017). Cocoa Industry. <https://doi.org/10.18356/cfb75b0e-en>
- Gené-Mola, J., Gregorio, E., Auat Cheein, F., Guevara, J., Llorens, J., Sanz-Cortiella, R., Escolà, A., & Rosell-Polo, J. R. (2020). Fruit detection, yield prediction and canopy geometric characterization using LiDAR with forced air flow. *Computers and Electronics in Agriculture*, 168. <https://doi.org/10.1016/j.compag.2019.105121>
- Gonzalez, R. C., & Woods, R. E. (2018). Digital Image Processing (Fourth).
- Google Colaboratory. (2020). Google Colaboratory. <https://colab.research.google.com/notebooks/intro.ipynb#recent=true>
- Habibi Aghdam, H., & Jahani Heravi, E. (2017). Guide to Convolutional Neural Networks. Springer International Publishing. <https://doi.org/10.1007/978-3-319-57550-6>

- Heredia Gómez, J. F., & Talero Sarmiento, L. H. (2020). Cocoa Ripeness Dataset TCS 01. <https://www.kaggle.com/juanfelipeheredia/cocoa-ripeness-dataset-tcs-01>
- Hùng, V. (2020). Tensorflow-yolov4-tflite. <https://github.com/hunglc007/tensorflow-yolov4-tflite>
- ILRF (2014). The fairness gap: Farmer incomes and solutions to child labor in cocoa. Washington, DC, International Labor Rights Forum.
- Jhuria, M., Kumar, A., & Borse, R. (2013). Image processing for smart farming: Detection of disease and fruit grading. 2013 IEEE Second International Conference on Image Information Processing (ICIIP-2013), 521–526. <https://doi.org/10.1109/ICIIP.2013.6707647>
- Jiang, Z., Zhao, L., Li, S., Jia, Y., & Lique, Z. (n.d.). Real-time object detection method for embedded devices.
- Kang, H., & Chen, C. (2020). Fast implementation of real-time fruit detection in apple orchards using deep learning. *Computers and Electronics in Agriculture*, 168, 105108. <https://doi.org/10.1016/j.compag.2019.105108>
- Kim, P. (2017). Deep Learning BT - MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence. Apress. https://doi.org/10.1007/978-1-4842-2845-6_5
- Kirk, R., Cielniak, G., & Mangan, M. (2020). L*a*b*Fruits: A Rapid and Robust Outdoor Fruit Detection System Combining Bio-Inspired Features with One-Stage Deep Learning Networks. In *Sensors* (Vol. 20, Issue 1). <https://doi.org/10.3390/s20010275>
- Koirala, A., Walsh, K. B., Wang, Z., & McCarthy, C. (2019). Deep learning – Method overview and review of use for fruit detection and yield estimation. In *Computers and Electronics in Agriculture* (Vol. 162, pp. 219–234). Elsevier B.V. <https://doi.org/10.1016/j.compag.2019.04.017>
- Konica Minolta. (2020). Entendiendo El Espacio de Color CIE L*A*B*. <https://sensing.konicaminolta.us/mx/blog/entendiendo-el-espacio-de-color-cie-lab/>
- Krig, S. (2014). Computer Vision Metrics: Survey, Taxonomy, and Analysis (Vol. 9781430259305). <https://doi.org/10.1007/978-1-4302-5930-5>
- Liu, X., Chen, S. W., Liu, C., Shivakumar, S. S., Das, J., Taylor, C. J., ... Kumar, V. (2019). Monocular Camera Based Fruit Counting and Mapping with Semantic

- Data Association. *IEEE Robotics and Automation Letters*, 4(3), 2296–2303. <https://doi.org/10.1109/LRA.2019.2901987>
- Maldonado, W., & Barbosa, J. C. (2016). Automatic green fruit counting in orange trees using digital images. *Computers and Electronics in Agriculture*, 127, 572–581. <https://doi.org/10.1016/j.compag.2016.07.023>
- Martínez Guerrero, N. (2016). Aportes de la investigación de FEDECACAO –Fondo Nacional del Cacao al incremento de la productividad y reconocimiento de la calidad del cacao de Colombia.
- Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill Science/Engineering/Math.
- Mite-Baidal, K., Solís-Avilés, E., Martínez-Carriel, T., Marcillo-Plaza, A., Cruz-Ibarra, E., & Baque-Bustamante, W. (2019). Analysis of Computer Vision Algorithms to Determine the Quality of Fermented Cocoa (*Theobroma Cacao*): Systematic Literature Review. *Advances in Intelligent Systems and Computing*, 901, 79–87. https://doi.org/10.1007/978-3-030-10728-4_9
- Mordvintsev, A., & Abid K. (2013). Fourier Transform. https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_transforms/py_fourier_transform/py_fourier_transform.html
- Motamayor, J. C., Lachenaud, P., da Silva e Mota, J. W., Loor, R., Kuhn, D. N., Brown, J. S., & Schnell, R. J. (2008). Geographic and Genetic Population Differentiation of the Amazonian Chocolate Tree (*Theobroma cacao* L). *PLoS ONE*, 3(10), e3311. <https://doi.org/10.1371/journal.pone.0003311>
- NVIDIA cuDNN | NVIDIA Developer. (2021). Retrieved May 5, 2021, from <https://developer.nvidia.com/cudnn>
- OpenCV - OpenCV. (2021). Retrieved May 5, 2021, from <https://opencv.org/>
- OpenCV. (2015). Introduction to SURF (Speeded-Up Robust Features). https://docs.opencv.org/master/df/dd2/tutorial_py_surf_intro.html
- Pacto por Colombia, pacto por la equidad. (2018). Plan Nacional de Desarrollo. Recuperado de <https://colaboracion.dnp.gov.co/CDT/Prensa/Resumen-PND2018-2022-final.pdf>
- Patel, H., Jiménez, A., Ceres, R., Annamalai, P., & Lee, W. (2013). A Survey of Computer Vision Methods for Counting Fruits and Yield Prediction.
- PDD-Santander. (2020). Santander siempre contigo y para el mundo Plan de Desarrollo Departamental 2020-2023. *Planeación Departamental*, 53(9), 1689.

- Python Software Foundation. (2016). Python Package Index. Labellmg. <https://pypi.org/project/labellmg/>
- Rahnemoonfar, M., & Sheppard, C. (2017). Deep Count: Fruit Counting Based on Deep Simulated Learning. *Sensors*, 17(4), 905. <https://doi.org/10.3390/s17040905>
- Redacción Economía. (20 de Agosto de 2020). El Espectador: Gobierno presentó proyecto de ley para incluir a Colombia en Organización Internacional del Cacao. Recuperado de <https://www.elespectador.com/>
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2018). You Only Look Once: Unified, Real-Time Object Detection. Retrieved from <http://pjreddie.com/yolo/>
- Russell, S. J., Norvig, P., Canny, J. F., Malik, J. M., Edwards, D. D., Jonathan, S. J. S., & Norvig, P. (1995). *Artificial Intelligence: A Modern Approach*. Alan Apt.
- Sánchez, V., Zambrano, J., & Iglesias, C. (2019). La cadena de valor del cacao en América Latina y el Caribe. Retrieved from: <http://repositorio.iniap.gob.ec/handle/41000/5382>
- Schuld, M., & Petruccione, F. (2018). Machine Learning. In *Supervised Learning with Quantum Computers* (pp. 21–73). Springer International Publishing. https://doi.org/10.1007/978-3-319-96424-9_2
- Serrano, S., & Heredia Gómez, J. F. (2020). Cocoa Diseases (YOLOv4). <https://www.kaggle.com/serranosebas/enfermedades-cacao-yolov4>
- Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1), 60. <https://doi.org/10.1186/s40537-019-0197-0>
- Solomon, C., & Breckon, T. (2011). *Fundamentals of Digital Image Processing*. <https://doi.org/doi:10.1002/9780470689776>.
- Tan, D. S., Leong, R. N., Laguna, A. F., Ngo, C. A., Lao, A., Amalin, D., & Alvindia, D. (2016). A framework for measuring infection level on cacao pods. 2016 IEEE Region 10 Symposium (TENSYMP), 384–389. <https://doi.org/10.1109/TENCONSpring.2016.7519437>
- Tian, Y., Yang, G., Wang, Z., Wang, H., Li, E., & Liang, Z. (2019). Apple detection during different growth stages in orchards using the improved YOLO-V3 model. *Computers and Electronics in Agriculture*, 157, 417–426. <https://doi.org/10.1016/j.compag.2019.01.012>
- Vasconez, J. P., Delpiano, J., Vougioukas, S., & Auat Cheein, F. (2020). Comparison of convolutional neural networks in fruit detection and counting: A

- comprehensive evaluation. *Computers and Electronics in Agriculture*, 173, 105348. <https://doi.org/10.1016/j.compag.2020.105348>
- Voora, V., Bermúdez, S., & Larrea, C. (2019). *Global Market Report: Cocoa*. International Institute for Sustainable Development.
- Wan, S., & Goudos, S. (2020). Faster R-CNN for multi-class fruit detection using a robotic vision system. *Computer Networks*, 168, 107036. <https://doi.org/10.1016/j.comnet.2019.107036>
- Wang, C.-Y., Liao, H.-Y. M., Yeh, I.-H., Wu, Y.-H., Chen, P.-Y., & Hsieh, J.-W. (2019). CSPNET: A NEW BACKBONE THAT CAN ENHANCE LEARNING CAPABILITY OF CNN A PREPRINT. <https://github.com/WongKinYiu/CrossStagePartialNetworks>.
- Wang, Q., Nuske, S., Bergerman, M., & Singh, S. (2013). Automated Crop Yield Estimation for Apple Orchards. In J. P. Desai, G. Dudek, O. Khatib, & V. Kumar (Eds.), *Experimental Robotics: The 13th International Symposium on Experimental Robotics* (pp. 745–758). Springer International Publishing. https://doi.org/10.1007/978-3-319-00065-7_50
- White, S., & Kennedy, J. (2018). CMY and CMYK Color Spaces. <https://docs.microsoft.com/en-us/windows/win32/wcs/cmy-and-cmyk-color-spaces>
- Zhang, B., Gao, Y., Zhao, S., & Liu, J. (2010). Local Derivative Pattern Versus Local Binary Pattern: Face Recognition With High-Order Local Pattern Descriptor. *IEEE Transactions on Image Processing*, 19(2), 533–544. <https://doi.org/10.1109/TIP.2009.2035882>
- Zhou, Z., Song, Z., Fu, L., Gao, F., Li, R., & Cui, Y. (2020). Real-time kiwifruit detection in orchard using deep learning on Android™ smartphones for yield estimation. *Computers and Electronics in Agriculture*, 179, 105856. <https://doi.org/10.1016/j.compag.2020.105856>
- Zou, H., Lu, H., Li, Y., Liu, L., & Cao, Z. (2020). Maize tassels detection: a benchmark of the state of the art. *Plant Methods*, 16(1), 1–15. <https://doi.org/10.1186/s13007-020-00651-z>

ANEXOS

Anexo A. Matriz de casos de uso

MATRIZ DE CASOS DE USO				
Número	Nombre del Caso de Uso	Actor que intervienen	Precedencia	Nivel de Complejidad
ECU001	Indicar la variedad de cacao	Usuario	Ninguna	Baja
ECU002	Detectar mazorcas de cacao	Usuario	ECU002	Alto
ECU003	Estimar producción de cacao	Usuario	ECU003 ECU004	Alto
ECU004	Guardar registro de detección	Usuario	ECU002	Medio
ECU005	Visualizar historial/informe	Usuario	ECU004	Medio

Anexo B. Tabla de riesgos

Ref. Riesgo	Riesgo	Categoría	Probabilidad %	Impacto	Plan RSG
R001	Planificación demasiado optimista	Proyecto	30	1	<p>Conocer los requerimientos funcionales y no funcionales del proyecto en detalle.</p> <p>Realizar una buena planeación de recursos, tareas y tiempo.</p> <p>Incluir en la planeación un tiempo racional por si ocurren imprevistos.</p>
R002	Los planes del proyecto se abandonan por la presión, llevando al caos y a un desarrollo ineficiente.	Proyecto	25	1	<p>Realizar una buena planificación de tareas con su correspondiente fecha</p> <p>Realizar un control de las tareas realizadas a la fecha.</p>
R004	Los desarrolladores necesitan un tiempo extra para acostumbrarse a trabajar con herramientas o entornos nuevos.	Empresarial	50	2	<p>Contemplar en el plan del proyecto tiempo extra para la capacitación de los desarrolladores sobre las herramientas a trabajar</p>
R005	El desarrollo va más lento de lo esperado.	Empresarial	70	1	<p>Asignar tareas semanales para ver avances en el proyecto y hacer consultas a personas de más experiencia para que las dudas sobre la herramienta del entorno se aclaren.</p>
R006	Un retraso en una tarea produce retrasos en cascada en las tareas dependientes.	Proyecto	75	1	<p>Hacer revisiones semanales sobre el avance del proyecto, dedicar tiempo extra a la actividad retrasada o asignar a otro miembro para que se realice la tarea, de esta manera el proyecto no se atrase en gran medida.</p>
R008	Los módulos propensos a tener errores necesitan más trabajo	Producto	30	2	<p>Asignar más tiempo a las tareas de alta complejidad y realizar revisiones constantes</p>

Anexo C. Especificación de casos de uso

ID	UC-001			
Nombre	Indicar la variedad de cacao			
Tipo	Indispensable			
Prioridad	Alta			
Actores involucrados	Usuario			
DESCRIPCIÓN DEL CASO DE USO				
El usuario puede seleccionar la variedad de cacao para la cual quiere estimar la producción. Donde contará con cuatro variedades: CCN51, FEC, FEAR 5, FSV41.				
ENTRADAS		SALIDAS		
Valor booleano asociados al botón de la variedad seleccionada.		Retorno de activity para la detección.		
PRECONDICIONES		POSTCONDICIONES		
<ul style="list-style-type: none"> Solo se puede elegir una variedad de cacao. 		<ul style="list-style-type: none"> No aplica. 		
DEPENDENCIA		EXTENSIÓN		
No aplica		No aplica		
FLUJO NORMAL DE EVENTOS				
No.	Acción del actor	Respuesta del sistema		
1	El usuario selecciona el tipo de detección.	El sistema retorna un activity con instrucciones y un botón de continuar.		
2	El usuario recibe instrucciones y oprime el botón continuar.	El sistema retorna un activity con botones de selección de variedad.		
3	El usuario selecciona una variedad de cacao para la detección.	El sistema actualiza la variable que lleva el control de selección.		
4	El usuario oprime el botón continuar.	El sistema guarda la el dato asociado a la variedad seleccionada.		
CAMINOS ALTERNATIVOS				
No. actividad en flujo normal		No.	Acción del actor	Respuesta del sistema
Se presenta la alternativa	Se integra al flujo normal			
Detección precisa	sí	4	El usuario selecciona el botón de continuar después de elegir la variedad.	El sistema retorna un activity con dos botones para cargar imagen, ya sea desde la galería o la cámara.
Detección rápida	sí	4	El usuario selecciona el botón de continuar después de elegir la variedad.	El sistema automáticamente abre la cámara y comienza a detectar en tiempo real.
CAMINOS DE EXCEPCIÓN				
No aplica				
PUNTOS DE EXTENSIÓN				
No aplica				
CRITERIOS DE ACEPTACIÓN				
Dado un usuario que selecciona el tipo de detección, cuando seleccione la variedad de cacao y oprima el botón de continuar, entonces navega a la página correspondiente de captura de imagen o fotogramas.				

REQUERIMIENTOS NO FUNCIONALES ASOCIADOS		
Conectividad offline.		
SEGUIMIENTO AL CASO DE USO		
Fecha	Descripción	Autores
19/04/2021	Indicar variedad de cacao	Karol Daniela Cala. Lisseth Andrea Hernández. Cristian David Parra.

ID	UC-002			
Nombre	Detectar mazorcas de cacao			
Tipo	Indispensable			
Prioridad	Esencial			
Actores involucrados	Usuario			
DESCRIPCIÓN DEL CASO DE USO				
El usuario puede visualizar la detección de mazorcas de cacao en una interfaz gráfica que clasifica y localiza.				
ENTRADAS			SALIDAS	
·Matriz de píxeles			·Matriz de píxeles con cuadros delimitadores dibujados	
PRECONDICIONES			POSTCONDICIONES	
<ul style="list-style-type: none"> Solo se puede hacer una detección individual 			<ul style="list-style-type: none"> La detección precisa retorna una imagen con las mazorcas clasificadas y localizadas La detección rápida retorna en tiempo real un conjunto de frames con cuadros delimitadores dibujados 	
DEPENDENCIA			EXTENSIÓN	
<ul style="list-style-type: none"> Modelo de detección con formato .tflite Mapa de clases 			·No aplica	
FLUJO NORMAL DE EVENTOS				
No.	Acción del actor		Respuesta del sistema	
1	El usuario selecciona la imagen o accede a la cámara.		El sistema procesa la matriz de píxeles y los envía como datos de entrada al modelo YOLOv4-Tiny.	
2	El usuario queda a la espera.		El sistema retorna gráficamente una detección que incluye mazorcas localizadas con cuadros delimitadores, y su respectiva etiqueta de clase. Asimismo, muestra información del conteo total de mazorcas detectadas, y el conteo específico que incluye el número de mazorcas sanas, con fitóftora y con monilia.	
CAMINOS ALTERNATIVOS				
No. actividad en flujo normal		No.	Acción del actor	Respuesta del sistema
Se presenta la alternativa	Se integra al flujo normal			
Detección precisa	sí	1	El usuario puede seleccionar entre cargar imagen desde la galería o desde cámara	El sistema retorna una imagen con mazorcas detectadas.
Detección rápida	sí	1	El usuario selecciona detección rápida	El sistema automáticamente toma el conjunto de frames y comienza a detectar en tiempo real.
CAMINOS DE EXCEPCIÓN				
<ul style="list-style-type: none"> El sistema informa si no se puede cargar la imagen o si hay problemas con la cámara trasera El sistema informa si se carga un archivo diferente a una matriz de píxeles El sistema no dibuja cuadros delimitadores si el número de mazorcas detectadas es igual a 0 El sistema informa si se acaba de cargar correctamente una imagen 				
PUNTOS DE EXTENSIÓN				
Guardar información asociada al conteo y detección				

CRITERIOS DE ACEPTACIÓN

Dado un usuario que selecciona el tipo de detección, cuando carga una imagen o abre la cámara, entonces recibe información gráfica de la detección y el conteo específico por sanidad.

REQUERIMIENTOS NO FUNCIONALES ASOCIADOS

- Conectividad offline
- Tiempo de inferencia

SEGUIMIENTO AL CASO DE USO

Fecha	Descripción	Autores
22/04/2021	Detectar mazorcas de cacao y visualizar información asociada al conteo	Karol Daniela Cala Liseth Andrea Hernández Cristian David Parra

ID	UC-003			
Nombre	Guardar registro de detección			
Tipo	Deseable			
Prioridad	Alta			
Actores involucrados	Usuario			
DESCRIPCIÓN DEL CASO DE USO				
Una vez hecha la detección (sin importar el tipo de detección), el usuario puede guardar un registro del conteo hecho junto con otros parámetros.				
ENTRADAS		SALIDAS		
<ul style="list-style-type: none"> • Conteo total • Conteo específico (incluye cantidad por fitóftora, monilia y sana) • Fecha y hora • Id de detección • Variedad de cacao • Índice de mazorca 		Validación del guardado		
PRECONDICIONES		POSTCONDICIONES		
<ul style="list-style-type: none"> • Solo se puede guardar registro si con anterioridad se ha realizado una detección 		<ul style="list-style-type: none"> • El registro se guarda en una base de datos local basada en SQLite 		
DEPENDENCIA		EXTENSIÓN		
<ul style="list-style-type: none"> • Librería de persistencia ROOM 		No aplica		
FLUJO NORMAL DE EVENTOS				
No.	Acción del actor	Respuesta del sistema		
1	El usuario oprime el botón de guardar	El sistema guarda una instancia automáticamente con los datos de entradas descritos previamente.		
CAMINOS ALTERNATIVOS				
No. actividad en flujo normal		No.	Acción del actor	Respuesta del sistema
Se presenta la alternativa	Se integra al flujo normal			
No aplica				
CAMINOS DE EXCEPCIÓN				
<ul style="list-style-type: none"> • El sistema informa si se guardó correctamente el registro. 				
PUNTOS DE EXTENSIÓN				
No aplica.				
CRITERIOS DE ACEPTACIÓN				
Dado un usuario que ejecuta un tipo de detección, cuando oprime el botón de guardar, entonces el sistema automáticamente guarda información asociada al conteo.				
REQUERIMIENTOS NO FUNCIONALES ASOCIADOS				
<ul style="list-style-type: none"> • Conectividad offline • Tiempo de inferencia 				
SEGUIMIENTO AL CASO DE USO				
Fecha	Descripción	Autores		
26/04/2021	Guardar registros asociados al conteo	Karol Daniela Cala Liseth Andrea Hernández Cristian David Parra		

ID	UC-004			
Nombre	Estimar producción de cacao			
Tipo	Indispensable			
Prioridad	Alta			
Actores involucrados	Usuario			
DESCRIPCIÓN DEL CASO DE USO				
Una vez hecha la detección y el guardado, el usuario puede estimar la producción de cacao con base en conteo y el índice de mazorca.				
ENTRADAS		SALIDAS		
<ul style="list-style-type: none"> Lista de conteos asociados a detecciones previas 		<ul style="list-style-type: none"> Valor numérico que representa la producción basado en el índice de mazorca. 		
PRECONDICIONES		POSTCONDICIONES		
<ul style="list-style-type: none"> Solo se puede estimar su con anterioridad ha hecho conteos. 		<ul style="list-style-type: none"> No aplica 		
DEPENDENCIA		EXTENSIÓN		
<ul style="list-style-type: none"> Librería de persistencia ROOM 		No aplica		
FLUJO NORMAL DE EVENTOS				
No.	Acción del actor		Respuesta del sistema	
1	El usuario oprime el botón de estimar		El sistema hace una consulta basada en la persistencia ROOM para obtener los registros correspondientes al id de detección y retorna una lista con instancias almacenados localmente.	
2	No aplica		El sistema procesa los datos asociados a detecciones previas, operando las cantidades con los índices de mazorca	
3	No aplica		El sistema retorna la información procesada, la cual incluye una gráfica con el conteo acumulado específico por sanidad y la estimación dada en kg	
4	El usuario puede visualizar información asociada a la producción estimada			
CAMINOS ALTERNATIVOS				
No. actividad en flujo normal		No.	Acción del actor	Respuesta del sistema
Se presenta la alternativa	Se integra al flujo normal			
No aplica				
CAMINOS DE EXCEPCIÓN				
<ul style="list-style-type: none"> El sistema informa si el número total de mazorcas es menor a 0. 				
PUNTOS DE EXTENSIÓN				
No aplica.				
CRITERIOS DE ACEPTACIÓN				
Dado un usuario que ejecuta un tipo de detección y conteo, cuando oprime el botón de estimar, entonces el sistema automáticamente estima la producción de cacao.				
REQUERIMIENTOS NO FUNCIONALES ASOCIADOS				
<ul style="list-style-type: none"> Conectividad offline 				
SEGUIMIENTO AL CASO DE USO				
Fecha	Descripción		Autores	

03/05/2021	Estimar la producción de cacao basado en la detección, conteo, e índice de mazorca.	Karol Daniela Cala Lisseth Andrea Hernández Cristian David Parra
------------	-------------------------------------------------------------------------------------	------------------------------------------------------------------------

ID	UC-005			
Nombre	Mostrar historial e informe mensual			
Tipo	Deseable			
Prioridad	Baja			
Actores involucrados	Usuario			
DESCRIPCIÓN DEL CASO DE USO				
Una vez registrada un número de número de detecciones, el usuario podrá visualizar el historial de las estimaciones realizadas y una gráfica de la función variedades de cacao vs estimaciones [kg]				
ENTRADAS		SALIDAS		
<ul style="list-style-type: none"> Lista de todas las estimaciones hechas 		Listado visible y entendible para el usuario con las estimaciones que ha realizado Gráfica de barras mensual de las variedades de cacao y el valor acumulado de estimaciones en kg.		
PRECONDICIONES		POSTCONDICIONES		
<ul style="list-style-type: none"> El tamaño de la lista de estimaciones debe ser mayor a 1. 		<ul style="list-style-type: none"> No aplica 		
DEPENDENCIA		EXTENSIÓN		
<ul style="list-style-type: none"> Librería de persistencia ROOM Librería MPAndroidChart 		No aplica		
FLUJO NORMAL DE EVENTOS				
No.	Acción del actor		Respuesta del sistema	
1	El usuario oprime el botón de registros		El sistema retorna un activity con los botones de historial e informe.	
2	El usuario escoge la información deseada		El sistema hace una consulta SQL con la persistencia ROOM y retorna un listado de estimaciones.	
4	El usuario puede visualizar información solicitada			
CAMINOS ALTERNATIVOS				
No. actividad en flujo normal		No.	Acción del actor	Respuesta del sistema
Se presenta la alternativa	Se integra al flujo normal			
Historial	sí	2	El usuario escoge ver el historial	El sistema tras hacer la Consulta SQL retorna un listado gráfico para el usuario con información asociada a cada una de las estimaciones que ha hecho.
Gráfica mensual	sí	2	El usuario escoge ver el informe mensual	El sistema tras hacer la Consulta SQL, filtra las instancias del último mes y retorna una gráfica con la representación de variedades vs kg de cacao estimados.
CAMINOS DE EXCEPCIÓN				
<ul style="list-style-type: none"> El sistema informa si el número total estimaciones es igual a 0 				
PUNTOS DE EXTENSIÓN				
No aplica.				
CRITERIOS DE ACEPTACIÓN				

Dado un usuario que consulta un tipo de registro, cuando oprime el botón de historial o informe, entonces el sistema automáticamente retorna el listado requerido de manera gráfica.

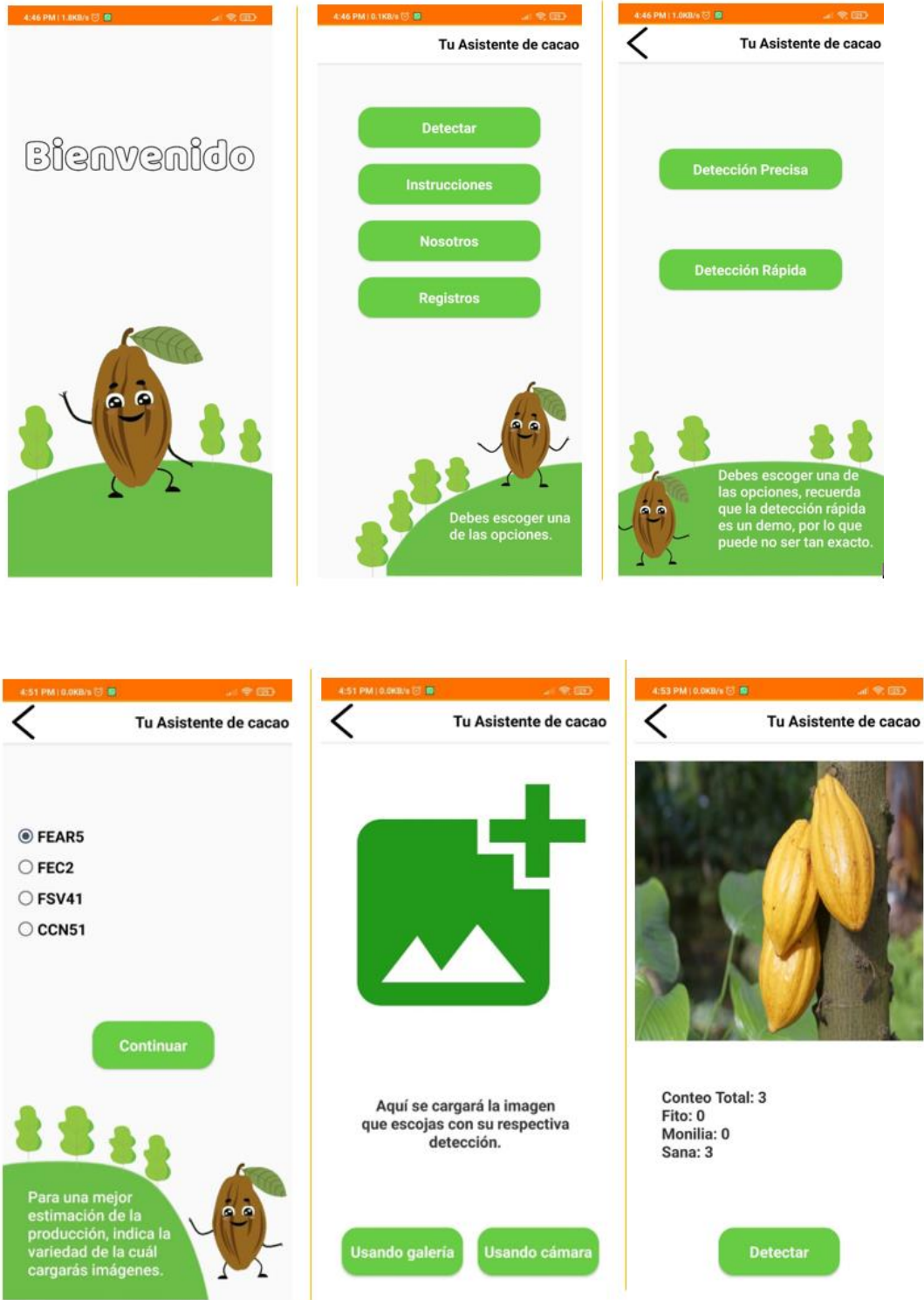
REQUERIMIENTOS NO FUNCIONALES ASOCIADOS

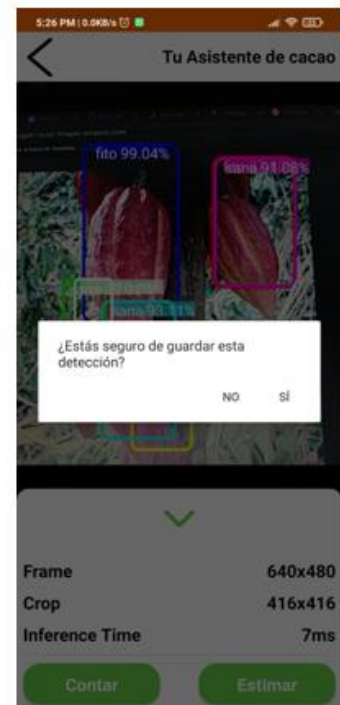
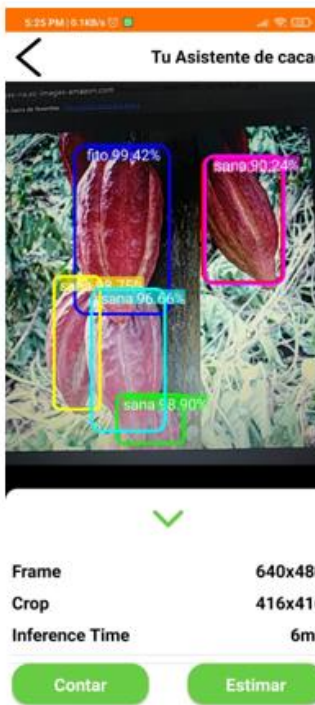
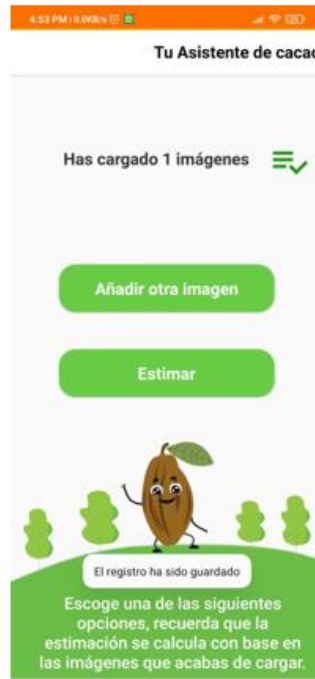
- Conectividad offline

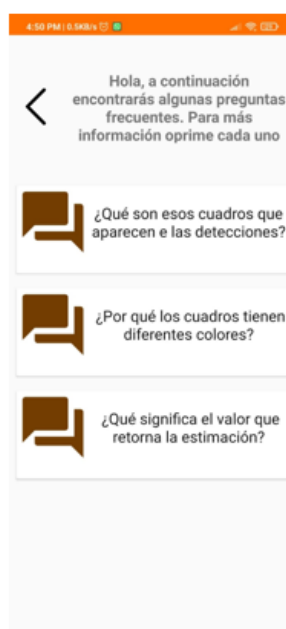
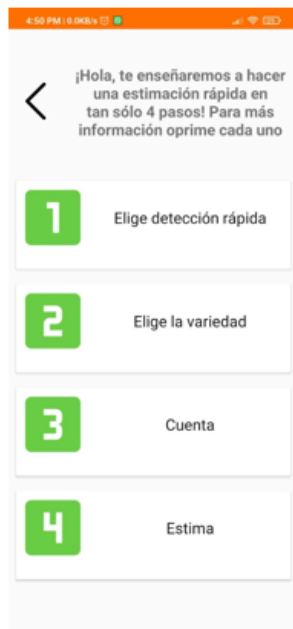
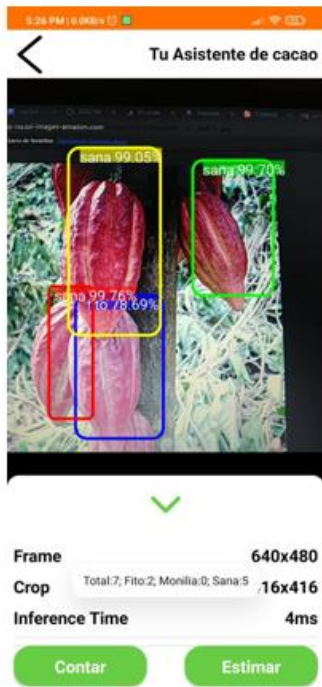
SEGUIMIENTO AL CASO DE USO

Fecha	Descripción	Autores
03/07/2021	Acceder al registro de las estimaciones hechas.	Karol Daniela Cala Lisbeth Andrea Hernández Cristian David Parra

Anexo D. Diseño







4:50 PM | 0.29x/s

Tu Asistente de cacao

Cristian David Parra Muñoz



Estudiante de Ingeniería de Sistemas
cparra569@unab.edu.co

Lisbeth Andrea Hernández Flórez



Estudiante de Ingeniería de Sistemas
lhernandez428@unab.edu.co

4:50 PM | 0.29x/s

Tu Asistente de cacao

Karol Daniela Cala Pinzón



Estudiante de Ingeniería de Sistemas
kcala461@unab.edu.co

Leonardo Hernán Talero Sarmiento



M.Sc, PhD(s)
ltalero@unab.edu.co

Nestor Edgardo Nieves Peña

4:50 PM | 0.00x/s

Tu Asistente de cacao

Leonardo Hernán Talero Sarmiento



M.Sc, PhD(s)
ltalero@unab.edu.co

Nestor Edgardo Nieves Peña




Ingeniero Agrónomo e Instructo Agrícola del SENA
nenieves64@misena.edu.co

7:41 PM | 0.00x/s

Tu Asistente de cacao

Historial

Reporte Mensual



Debes escoger una de las opciones.

7:41 PM | 0.00x/s

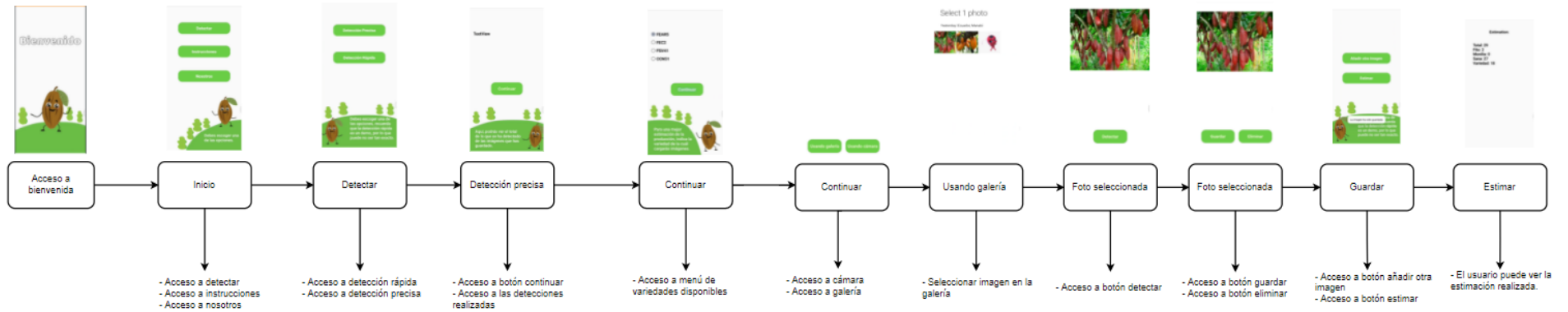
Tu Asistente de cacao

Aquí, se muestra el historial de todas las detecciones que has realizado.

May 16 2021	Varietal: FSV41	Producción: 0.00 Kg de cacao seco
May 16 2021	Varietal: FEARS	Producción: 0.00 Kg de cacao seco
May 16 2021	Varietal: FEARS	Producción: 0.17 Kg de cacao seco
May 16 2021	Varietal: FEC2	Producción: 0.00 Kg de cacao seco



Anexo E. Storyboarding



Anexo F. Pruebas de uso.

ID de caso de prueba	BU_001	Descripción de caso de prueba	Prueba de detección precisa		
Creado por	Cristian Parra	Revisado por	Liseth Hernández	Versión	Entregable

Nombre de quien ejecuta la prueba	Cristian Parra	Fecha de la prueba	10/05/2021	Caso de prueba (Aprobado/Fallido/No ejecutado)	Aprobado
------------------------------------------	----------------	---------------------------	------------	-------------------------------------------------------	----------

S #	Prerrequisitos:
1	Cargar imagen

S #	Datos de prueba
1	Bitmap = imagen_seleccionada_URI

Escenario de prueba Verificar que se puedan cargar imágenes y detectar mazorcas de cacao con una detección precisa.

Paso#	Detalles del paso	Resultados esperados	Resultados actuales	Aprobado/ Fallido/ No ejecutado/ Suspendido
1	Elegir detección precisa	Abrir vista de variedad	Como se esperaba	Aprobado
2	Elegir variedad	Abrir vista para cargar imagen	Como se esperaba	Aprobado
3	Cargar imagen	Mostrar la imagen seleccionada con las mazorcas localizadas y clasificadas	Como se esperaba	Aprobado

ID de caso de prueba	BU_002	Descripción de caso de prueba	Prueba de detección rápida		
Creado por	Liseth Hernández	Revisado por	Cristian Parra	Versión	Entregable

Nombre de quien ejecuta la prueba	Liseth Hernández	Fecha de la prueba	12/05/2021	Caso de prueba (Aprobado/Fallido/No ejecutado)	Aprobado
------------------------------------------	------------------	---------------------------	------------	-------------------------------------------------------	----------

S #	Prerrequisitos:
1	Enfocar las mazorcas de cacao

S #	Datos de prueba
1	byte [] rgbBytes= matriz_de_fotogramas

Escenario de prueba Verificar que se puedan enfocar y detectar mazorcas de cacao con una detección rápida.

Paso#	Detalles del paso	Resultados esperados	Resultados actuales	Aprobado/ Fallido/ No ejecutado/ Suspendido
1	Elegir detección rápida	Abrir vista de variedad	Como se esperaba	Aprobado
2	Elegir variedad	Abrir cámara para enfocar mazorcas de cacao	Como se esperaba	Aprobado
3	Enfocar mazorcas de cacao y dar click en contar	Mostrar la mazorca enfocada con las mazorcas localizadas, contadas y clasificadas	Como se esperaba	Aprobado

ID de caso de prueba	BU_003	Descripción de caso de prueba	Prueba de Historial		
Creado por	Karol Cala	Revisado por	Liseth Hernández	Versión	Entregable

Nombre de quien ejecuta la prueba	Karol Cala	Fecha de la prueba	14/05/2021	Caso de prueba (Aprobado/Fallido/No ejecutado)	Aprobado
------------------------------------------	------------	---------------------------	------------	-------------------------------------------------------	----------

S #	Prerrequisitos:
1	Guardar tomas y realizar estimaciones en Detección Precisa

S #	Datos de prueba
1	List<Estimacion> historial=estimacionDAO.obtenerEstimaciones()

Escenario de prueba	Verificar.
----------------------------	------------

Paso#	Detalles del paso	Resultados esperados	Resultados actuales	Aprobado/ Fallido/ No ejecutado/ Suspendido
1	Elegir Historial	Abrir vista de con información del historial de todas las detecciones realizadas	Como se esperaba	Aprobado

ID de caso de prueba	BU_004	Descripción de caso de prueba	Prueba de Reporte Mensual		
Creado por	Cristian Parra	Revisado por	Karol Cala	Versión	Entregable

Nombre de quien ejecuta la prueba	Cristian Parra	Fecha de la prueba	15/05/2021	Caso de prueba (Aprobado/Fallido/No ejecutado)	Aprobado
------------------------------------------	----------------	---------------------------	------------	-------------------------------------------------------	----------

S #	Prerrequisitos:
1	Guardar tomas y realizar estimaciones en Detección Precisa durante un mes.

S #	Datos de prueba
1	lastFecha=estimacionDAO.getLastEstimacion().get(0).getFecha().split(",")

Escenario de prueba Verificar.

Paso#	Detalles del paso	Resultados esperados	Resultados actuales	Aprobado/ Fallido/ No ejecutado/ Suspendido
1	Elegir Reporte Mensual	Abrir vista con una gráfica de todas las detecciones realizadas en el mes. Dicha gráfica debe mostrar una función de variedad vs	Como se esperaba	Aprobado

		estimación acumulada en el mes.		
--	--	------------------------------------	--	--