

Instituto Tecnológico de Costa Rica
Carrera de Ingeniería en Computación,
Campus Tecnológico Local San Carlos

FungiDeTECtor: Sistema inteligente de reconocimiento
de hongos en plantas de piña del trópico húmedo de
Costa Rica

Práctica Profesional para optar por el título de
Ingeniero en Computación con el grado académico de
Bachiller Universitario

Warner Francisco Cortés Pérez

Costa Rica, 2021

Abstract - Object detection is one of the most basic and central tasks of computer vision. It is based on finding all the objects interested in an image and determining its category and location. In recent years, with the development of convolutional neural networks (CNN), significant advances have been made in object detection. The objective of this research is to build a model capable of detecting Fusarium and Phytophthora disease in pineapple plants from the Huetar Norte area. Neural networks of deep communication (CNN) fulfill the task of classifying the images in different conditions. In this document it will be carried out the creation of a model through platforms such as Tensorflow to carry out model training and classify healthy pineapples and pineapples with Fusarium.

Abstract - La detección de objetos es una de las tareas más básicas y centrales de la visión por computadora. Se basa en encontrar todos los objetos interesados en una imagen y determinar la categoría y ubicación de los mismos. En los últimos años, con el desarrollo de las redes neuronales convolucionales (CNN), se han logrado avances importantes en la detección de objetos. El objetivo de esta investigación consiste en construir un modelo capaz de detectar las enfermedades de Fusarium y Phytophthora en las plantas de piña de la zona Huetar Norte. Las redes neuronales de comunicación profunda (CNN) cumplen con la tarea de clasificar las imágenes en diferentes condiciones. En este documento se llevará a cabo la creación de un modelo por medio de plataformas como Tensorflow para realizar el entrenamiento del modelo y clasificar piñas sanas y piñas con alguna de las enfermedades mencionadas anteriormente.

Índice general

Introducción	1
Descripción de empresa	2
Contexto del proyecto	2
Problema	2
Objetivos	3
Objetivo general	3
Objetivos específicos	3
Justificación	3
Revisión de literatura	5
Marco teórico	5
Trabajos relacionados	5
Solución planteada	9
Propuesta	9
Involucrados	10
Procedimiento metodológico	11
Análisis de los riesgos	15
Requerimientos y Diseño	18
Definición de requerimientos	18
Requerimientos funcionales	18
Requerimientos no funcionales	19
Diseño de la plataforma de software	19
Plataforma de software	23
Aplicación	23
Desarrollo del modelo algorítmico	23
Desarrollo del servidor y aplicación web	25

Servidor web	25
Aplicación web	26
Evaluación	26
Conclusiones	27
Recomendaciones	27
Referencias	28

Índice de figuras

Figura 3.1: Propuesta del sistema
Figura 4.1: Diseño del modelo algorítmico
Figura 4.2: División del Dataset
Figura 4.3: Etiquetado de imagen
Figura 4.4: Aumentación de la data
Figura 4.5: Exportación del dataset
Figura 4.6: Diagrama del sistema
Figura 5.1: Importación de dependencias
Figura 5.2: Importación del dataset
Figura 5.3: Entrenamiento del modelo
Figura 5.4: Arquitectura de la plataforma
Figura 5.5: Servidor web
Figura 5.6: Aplicación web
Figura 5.7: Balanceo de Clases
Figura 5.8: Precisión
Figura 5.9: Reconocimiento de clases
Figura 6.0: Recall

Figura 6.1: Puntuación F1

Figura 6.2: Ejemplo resultado 1

Figura 6.3: Ejemplo resultado 2

Figura 6.4: Ejemplo resultado 3

Índice de tablas

Tabla 1: Matriz de los involucrados

Tabla 2: Matriz para el procedimiento metodológico

Tabla 3.1: Documentación de riesgos 1

Tabla 3.2: Documentación de riesgos 2

Tabla 3.3: Cronograma de trabajo

Capítulo I

1 Introducción

Este documento presenta el desarrollo del proyecto de práctica profesional, realizado como requisito final de graduación de la carrera de Ingeniería en Computación, el cual describe detalles del problema presente, la solución planteada para enfrentarlo y se complementa con otros apartados para mostrar la forma en que el mismo se aborda y el proceso que conlleva.

En el transcurso de este proyecto se desarrollará un nuevo conjunto de datos de imágenes de piñas para crear una herramienta que identifique o clasifique plantas de piñas sanas y plantas de piñas con las enfermedades de [Fusarium](#) y [Phytophthora](#), por lo que este proyecto se clasifica como un proyecto de innovación tanto para el sector agrícola costarricense como para el Instituto Tecnológico de Costa Rica.

El desarrollo de este proyecto trae consigo beneficios al desarrollo agro-tecnológico costarricense, pues además de servir como base para futuras implementaciones o investigaciones, también será de gran ayuda para el TEC en lo que a desarrollo de proyectos o investigaciones se refiere, tanto para los estudiantes como para los investigadores del mismo TEC, ya que actualmente el sector de San Carlos no cuenta con ningún recurso de este tipo y al ser la Piña un producto utilizado en esta zona, hay muchas personas interesadas el desarrollo de nuevos proyectos o investigaciones con esta fruta.

Este proyecto se desarrollará en conjunto con el Tecnológico de Costa Rica y contará con la participación de dos estudiantes de último semestre para su completo desarrollo.

1.1 Descripción de empresa

El Tecnológico de Costa Rica (TEC) es una institución nacional autónoma de educación superior universitaria, dedicada a la docencia, la investigación y la extensión de la tecnología y las ciencias conexas para el desarrollo de Costa Rica.

El desarrollo de este proyecto interno se llevará a cabo junto al Tecnológico de Costa Rica específicamente en su sede de San Carlos, esto dentro del esquema de investigación del laboratorio de PROTEC, en conjunto con la mentoría del profesor Rogelio González Quirós.

1.2 Contexto del proyecto

En este proyecto se desarrollará un nuevo set de datos de Piñas para la creación de una herramienta que identifique las enfermedades de “Fusarium” y “Phytophthora” utilizando técnicas de aprendizaje automático, ya que esta es una tarea que requiere de técnicas de laboratorio para llevarse a cabo, y estas representan un gran costo y dependencia de ojo experto, optar por estas nuevas técnicas de visión por computadora y aprendizaje automático, además de reducir en gran medida los costos, agilizará drásticamente el proceso.

Este proyecto se clasifica como un proyecto de innovación tanto para el sector agrícola costarricense como para el Instituto Tecnológico de Costa Rica, además no existen antecedentes o algún tipo de información relacionada con el tema a tratar.

1.3 Problema

Dificultad para identificar las enfermedades de “Fusarium” y “Phytophthora” en las frutas de piñas, así también como sus causas y efectos en la Zona Norte, ya que este proceso representa un gran costo y dependencia de ojo experto.

1.4 Objetivos

1.4.1 Objetivo general

Desarrollar una aplicación web que permita realizar la detección de las enfermedades Fusarium y Phytophthora en las plantas de piña mediante técnicas de aprendizaje automático y visión por computadora para promover el desarrollo tecnológico del sector piñero en la zona norte costarricense.

1.4.2 Objetivos específicos

1. Definir los requerimientos e historias de usuario para la elaboración del sistema.
2. Definir el diagrama de arquitectura del sistema.
3. Crear un conjunto de datos de fotografías de piña etiquetadas para alimentar un modelo de aprendizaje automático para la detección de las enfermedades “Fusarium” y “Phytophthora ” en las piñas.
4. Implementar un modelo de aprendizaje automático que permita detectar las enfermedades “Fusarium” y “Phytophthora” en las plantas de piñas a través de fotografías, mediante el uso de técnicas como “[Localización](#)”.
5. Diseñar y desarrollar una aplicación web para la detección de las enfermedades “Fusarium” y “Phytophthora” en las plantas de piña.

1.5 Justificación

Para nadie es un secreto que la elaboración de herramientas para la identificación de enfermedades en frutas es una tarea que requiere de técnicas de laboratorio para llevarse a cabo, por lo que optar por nuevas técnicas de visión por computadora y aprendizaje automático, además de reducir en gran medida los costos de estas investigaciones, también agilizarán drásticamente los tiempos de desarrollo.

El desarrollo de este proyecto trae consigo beneficios al desarrollo agro-tecnológico costarricense, pues además de servir como base para futuras implementaciones o investigaciones, también será de gran ayuda para el TEC en

lo que a desarrollo de proyectos o investigaciones se refiere, tanto para los estudiantes como para los investigadores del mismo TEC, ya que actualmente el sector de San Carlos no cuenta con ningún recurso de este tipo.

Capítulo II

2 Revisión de literatura

2.1 Marco teórico

En la actualidad técnicas de procesamiento de imágenes o de aprendizaje automático son muy utilizadas en procesos complejos y pesados como la clasificación de imágenes o reconocimiento preciso de ciertas características en imágenes. Existen muchos de algoritmos para llevar a cabo estos tipos de procesos, tales como las redes neuronales convolucionales (CNN) o el Support Vector Machine (SVM). Es importante mencionar que al ser algoritmos de alta complejidad es necesario utilizar lenguajes capaces de soportar dichos procesos, en esta ocasión se hará uso del lenguaje de programación de propósito general llamado Python.

Python, al ser Open Source posee una gran cantidad de mejoras continuas para resolver conflictos, y ofrece gran cantidad de librerías enfocadas en el machine learning como Numpy, utilizada como contenedor de datos genéricos; Pandas que suministra estructuras de datos y herramientas de analítica; Matplotlib la cual es una librería de trazado 2D, entre otras como OpenCV, Keras y TensorFlow.

2.2 Trabajos relacionados

Al realizar una revisión detallada de algunas de nuestras investigaciones o proyectos de referencia, entre cada subárea de categoría se mencionan técnicas utilizadas en otras investigaciones que pueden aportar información útil para cumplir con los objetivos de este proyecto, entre ellas se encuentran:

[\(Castro, 2020\)](#) en su tesis denominado DETECCIÓN Y CLASIFICACIÓN DE CÉLULAS NORMALES DE LA SANGRE PERIFÉRICA USANDO APRENDIZAJE PROFUNDO donde entreno modelos con redes neuronales convolucionales (CNN) y uso arquitecturas de CNN para la obtención de características de forma automática; logró entrenar una red desde cero para el clasificador con un accuracy de 0.862 y para el detector utilizó la arquitectura YOLO obteniendo un excelente desempeño.

[\(Garralda, 2018\)](#) en su tesis titulada LOCALIZACIÓN DE OBJETOS EN IMÁGENES MEDIANTE TÉCNICAS DE APRENDIZAJE PROFUNDO donde su objetivo es crear un modelo utilizando técnicas de aprendizaje profundo que sea capaz de determinar la posición de un violín en una imagen, ella lo realiza con el fin de contribuir para futuros 23 proyectos de su universidad donde concluye que su modelo óptimo fue menos del 65% e indica que en problemas de localización es difícil obtener una buena precisión.

[\(Lazo, 2019\)](#) en su trabajo de tesis titulado “ESPECTROSCOPIA CON INFRARROJO Y TÉCNICAS DE MACHINE LEARNING Y DEEP LEARNING PARA LA DETECCIÓN Y CLASIFICACIÓN DE FRUTAS PARA LA AGROINDUSTRIA: CASO ARÁNDANOS EMPRESA TAISA:2018”, que tiene como finalidad establecer una herramienta alternativa para poder hacer la clasificación de arándanos que permita reducir el costo el tiempo y optimizar el

proceso de detección y clasificación de frutas utilizando la librería keras con Tensorflow, encontró un modelo que clasifique los arándanos con una exactitud del 92%.

[\(Gutierrez, 2019\)](#) en su tesis titulado “RECONOCIMIENTO DE EVENTOS ANÓMALOS EN VIDEOS OBTENIDOS DE CÁMARAS DE VIGILANCIA, USANDO REDES CONVOLUCIONALES” tiene como objetivo detectar eventos anómalos en los videos de vigilancia y que no haya la necesidad de que intervenga alguna persona encargada para verificar dichos actos anómalos, para ello utiliza las arquitecturas de YOLO, Región Proposal + CNN(R-CNN) y el SSD encontrando al final un acierto del 86% de accuracy, 85% de precisión y 73% de recall en los videos además de ello indicando que YOLO es la arquitectura que procesa más rápido los videos en tiempo real.

3 Solución planteada

3.1 Propuesta

En la solución propuesta para este proyecto se plantea la creación desde cero de un dataset de Piña, en el cual las imágenes serán tomadas desde un punto de vista aéreo (**Drone**), esto con el objetivo de poder aprovechar posteriormente el dataset para otros fines además de este proyecto, además se desarrollará un modelo de Inteligencia artificial (IA) utilizando técnicas de aprendizaje automático y marcado como “**Localización**” para la detección de piñas sanas y piñas con las enfermedades de “**Fusarium**” y “**Phytophthora**”, todo esto para posteriormente desarrollar un servicio backend el cual cargará y utilizará dicho modelo entrenado para proveer la información necesaria al cliente en donde se podrán realizar las pruebas del modelo mencionado para mostrar la información correspondiente, dicho cliente será una aplicación web, utilizando tecnologías como ReactJS, Javascript (Frontend) entre otras como Python o NodeJS (Backend de manera auxiliar, no protagónico en el proyecto) como en la siguiente [figura \(3.1\)](#).

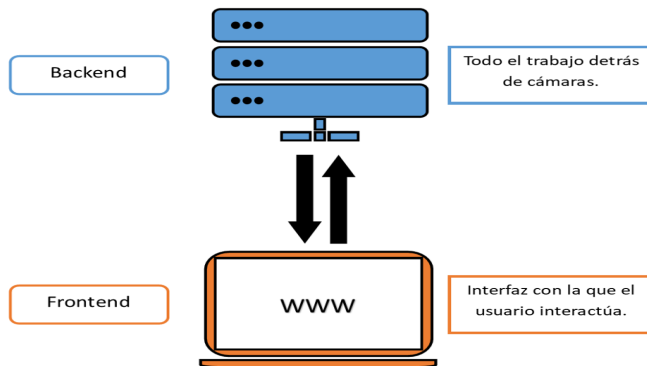


Figura 3.1: Propuesta del sistema

3.2 Involucrados

El personal involucrado en este proyecto se detalla en la siguiente tabla con su respectivo nombre, rol o responsabilidad que tendrá en el proyecto y una breve descripción del mismo.

Tabla 1: Matriz de los involucrados

Nombre	Rol	Descripción	Institución
Rogelio González Quirós	Líder del Proyecto	Se encargará de guiar desde un punto de vista técnico el desarrollo de cada etapa del proyecto.	Instituto Tecnológico de Costa Rica
Warner Francisco Cortés Pérez, Miguel Adán Rivas Méndez	Desarrolladores	Encargados de desarrollar el proyecto durante todas las etapas.	Instituto Tecnológico de Costa Rica
Arnoldo Gadea	Críticos	Encargados de verificar la calidad del producto y guiar en temas del agro.	Instituto Tecnológico de Costa Rica

Fuente: elaboración propia

3.3 Procedimiento metodológico

El trabajo a desarrollar se basa en la implementación de un dataset de plantas de piñas con las enfermedades de “Fusarium” y “Phytophthora”, el cual será el utilizado para la creación de un modelo algorítmico para la detección de las mismas enfermedades en dichas plantas para posteriormente ser utilizado o testeado en una aplicación web. Para lograr este objetivo se desarrollarán los siguientes cinco puntos:

Tabla 2: Matriz para el procedimiento metodológico

Objetivo Específico	Tarea	Meta	Indicador
Definir los requerimientos e historias de usuario para la elaboración del sistema.	<p>A. Entrevistar a los involucrados del proyecto para conocer qué funcionalidades esperar del proyecto.</p> <p>B. Definir una reunión con el líder técnico para conocer sobre los requerimientos no funcionales que se deben de satisfacer.</p> <p>C. Desarrollar las historias de usuario, los requerimientos funcionales y no funcionales.</p>	<p>A. Entregar un listado de historias de usuario.</p> <p>B. Entregar un listado de requerimientos funcionales.</p> <p>C. Entregar un listado de requerimientos no funcionales.</p>	<p>A. Documento de historias de usuario aprobado por el líder técnico del proyecto.</p> <p>B. Documento de requerimientos funcionales aprobado por el líder técnico del proyecto.</p> <p>C. Documento de requerimientos no funcionales aprobado por el líder técnico del proyecto.</p>

<p>Definir el diagrama de arquitectura del sistema.</p>	<p>A. Examinar la arquitectura del sistema.</p> <p>B. Diseñar un diagrama de la arquitectura que conlleve el completo desarrollo del sistema.</p>	<p>A. Entregar un diagrama de la arquitectura que conlleve el completo desarrollo del sistema.</p>	<p>A. Documento con el diagrama de arquitectura aprobado por el líder técnico del proyecto.</p>
<p>Crear un conjunto de datos de fotografías de piña etiquetadas para alimentar un modelo de aprendizaje automático para la detección de las enfermedades “Fusarium” y “Phytophthora” en la piña.</p>	<p>A. Planificación de día y lugar adecuado para tomar las fotografías.</p> <p>B. Planificar el modo adecuado para capturar las imágenes que serán utilizadas en el futuro modelo a entrenar, tales como tipo de ángulo, uso o no de drone, entre otros.</p> <p>C. Seleccionar y aplicar la técnica adecuada de etiquetado para cada una de las imágenes obtenidas anteriormente.</p>	<p>A. Entregar un dataset de piña para el desarrollo del modelo de aprendizaje automático.</p>	<p>A. Documento con la información de las imágenes de piña recolectadas en forma de dataset.</p>

	D. Aplicar distintas técnicas de limpieza o mejora de datasets para mejorar la calidad de su información.		
Diseñar e implementar un modelo de aprendizaje automático que permita detectar las enfermedades “Fusarium” y “Phytophthora” en las plantas de piñas a través de fotografías, mediante el uso de técnicas como “Localización”.	<p>A. Preparar la información necesaria para proceder a entrenar el modelo, tal como el tipo de algoritmo o técnicas a utilizar, tipo de modelo a entrenar, forma de consumir los datos entre otros.</p> <p>B. Entrenar el modelo planificado con los diferentes tipos de datos de entrenamiento definidos anteriormente.</p> <p>C. Evaluar el rendimiento obtenido para aplicar en caso de ser necesario técnicas para mejorar el rendimiento del modelo.</p> <p>D. Probar el modelo creado con diferentes tipos de datos para tener un acercamiento de cómo este funcionará en la aplicación web de pruebas.</p>	A. Generar un modelo de aprendizaje automático que permita detectar las enfermedades “Fusarium” y “Phytophthora” en las plantas de piñas a través de fotografías.	A. Modelo de Inteligencia Artificial capaz de detectar las enfermedades “Fusarium” y “Phytophthora” en piñas, validado por nuestros expertos en el área.

<p>Diseñar y desarrollar una aplicación web para la detección de las enfermedades “Fusarium” y “Phytophthora” en las plantas de piña.</p>	<p>A. Planificar el diseño de la aplicación web y la estructura del servidor web backend.</p> <p>B. Desarrollar la estructura del backend de acuerdo al diseño y la arquitectura planteada.</p> <p>C. Desarrollar la estructura del frontend de acuerdo al diseño y la arquitectura planteada e integrarlo con el backend.</p>	<p>A. Entregar un servicio web backend capaz de realizar el análisis de fotografías de piñas y detectar si estas se encuentran o no con las enfermedades de “Fusarium” o “Phytophthora”, utilizando el modelo desarrollado anteriormente.</p> <p>B. Entregar una aplicación web frontend en la que el usuario podrá interactuar y probar el modelo desarrollado cargando las imágenes deseadas de una manera intuitiva y rápida.</p>	<p>A. Aplicación web 100% funcional enfocada en realizar el análisis de fotografías de piñas y detectar si estas se encuentran o no con las enfermedades de “Fusarium” o “Phytophthora”.</p>
---	---	--	---

Fuente: elaboración propia

3.4 Análisis de los riesgos

En las siguientes tablas se muestran los riesgos que pueden afectar el desarrollo del proyecto.

Tabla 3.1: Documentación de riesgos 1.

Referencia	
Código	001
Nombre	Falta de datos
Categoría	Tecnológico
Causa	No tener una gran cantidad de imágenes de Piñas para generar un mejor rendimiento.
Impacto	Grave
Estrategia de evasión	–
Estrategia mitigación	Aplicar diversas técnicas de aumentación de los datos para multiplicar la cantidad de imágenes en diferentes formas.
Estrategia contingencia	–

Fuente: elaboración propia

Tabla 3.2: Documentación de riesgos 2.

Referencia	
Código	002
Nombre	Procesamiento de datos
Categoría	Tecnológico
Causa	Falta de cómputo para el alto procesamiento de datos
Impacto	Medio
Estrategia de evasión	–
Estrategia mitigación	Adquirir mejor poder computacional para procesamiento de los datos (AWS)
Estrategia contingencia	–

Fuente: elaboración propia

3.5 Tabla 3.3: Cronograma de trabajo

Objetivo	Fecha de inicio y finalización
Elaboración del Dataset	27/07/2021 - 02/09/2021
Preparaciones técnicas	27/07/2021 - 02/08/2021
Recolección de campo de los datos	03/08/2021 - 08/08/2021
Etiquetado de las imágenes	09/08/2021 - 25/08/2021
Saneamiento de los datos	26/08/2021 - 02/09/2021
Desarrollo del Modelo	03/09/2021 - 27/09/2021
Diseño del modelo	03/09/2021 - 16/09/2021
Entrenamiento del modelo	16/09/2021 - 23/09/2021
Evaluación del modelo	24/09/2021 - 27/09/2021
Desarrollo Aplicación Web	28/09/2021 - 10/11/2021
Definición de requerimientos	28/09/2021 - 04/10/2021
Diseño técnico de la aplicación	05/10/2021 - 11/10/2021
Desarrollo de la aplicación	12/10/2021 - 08/11/2021
Evaluación de la aplicación	09/11/2021 - 10/11/2021

Fuente: elaboración propia

4 Requerimientos y Diseño

4.1 Definición de requerimientos

A continuación, se muestra el proceso para la obtención de los requerimientos del sistema a desarrollar.

Se llevó a cabo el análisis de diferentes fuentes de información tales como papers en donde se analizó tanto técnicas como algoritmos para la identificación o clasificación de imágenes. Tras el análisis de la información y varias reuniones con los involucrados del proyecto se logró determinar los siguientes requerimientos:

4.1.1 *Requerimientos funcionales*

1. Al contar con una cantidad limitada de imágenes de piñas con las enfermedades de Fusarium y Phytophthora, y piñas en estado sano, el set de datos debe ser preprocesado antes del entrenamiento del algoritmo con técnicas de aumentación de la data como rotación, zoom, escala entre otras.
2. El modelo debe contar una gran cantidad de imágenes para su entrenamiento, las cuales se deben dividir en tres secciones, un setenta por ciento para el entrenamiento del algoritmo, un veinte por ciento para la validación de los datos y un diez por ciento para las pruebas.
3. El modelo debe llevar a cabo un proceso de obtención de características el cual deberá ser realizado por medio de una red neuronal convolucional (CNN) utilizando librerías como TensorFlow.
4. El modelo debe ser capaz de clasificar las imágenes de piñas en sanas o con la enfermedad de Fusarium o Phytophthora mediante el uso de la técnica de “Localización”.

4.1.2 Requerimientos no funcionales

1. El modelo para ser entrenado requiere de un conjunto de datos de imágenes de piñas con la enfermedad de Fusarium o Phytophthora y de imágenes de piñas en buen estado. Por lo que se necesita un gran set de datos que contengan ambas clases de imágenes para obtener un buen rendimiento.
2. Se debe hacer uso del lenguaje de programación llamado Python para la creación del algoritmo ya que este provee una gran cantidad de librerías potentes para el machine learning, tales como como OpenCV, Pandas o Tensorflow entre otras.

4.2 Diseño de la plataforma de software

En esta etapa se presenta la arquitectura conceptual de la solución planteada para la creación del modelo algorítmico y la plataforma de software.

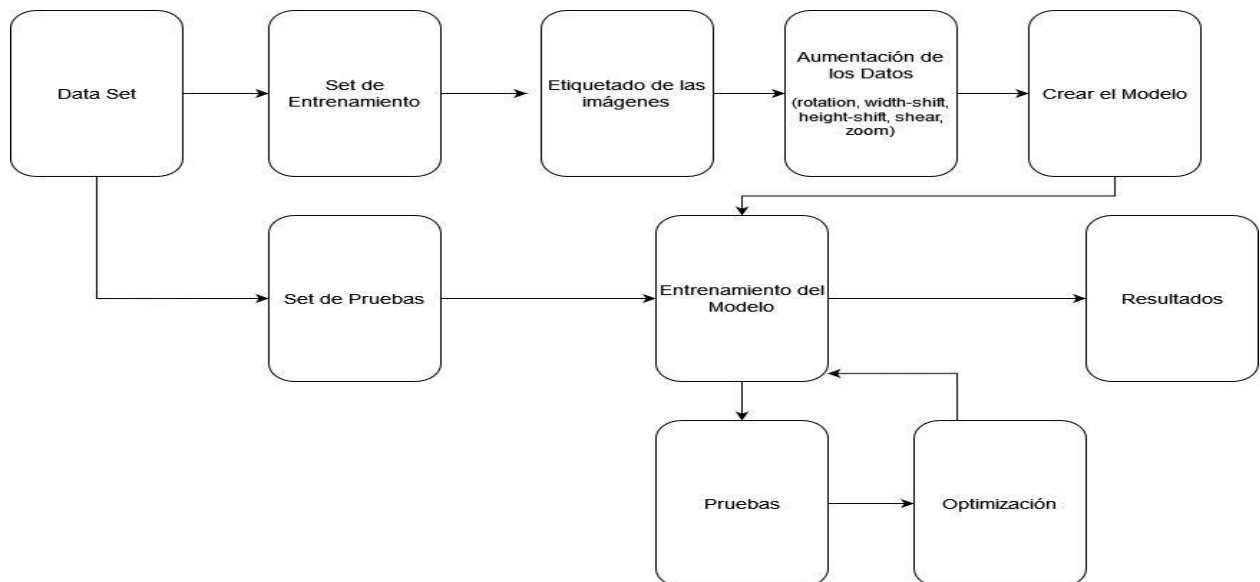


Figura 4.1: Diseño del modelo algorítmico

A continuación se procede a explicar las fases del diagrama mostrado en la [figura \(4.1\)](#).

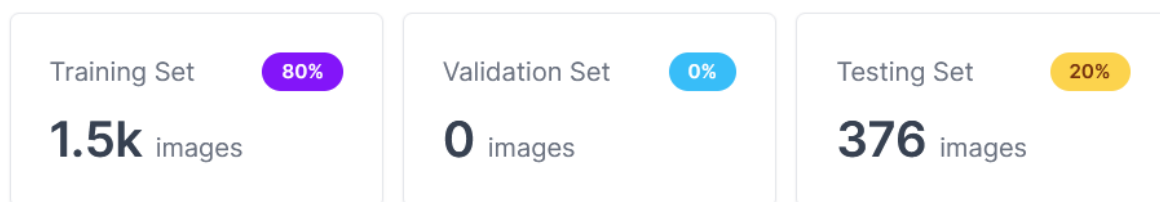


Figura 4.2: División del Dataset

Para hacer posible la correcta elaboración del modelo algorítmico es necesario dividir el set de datos en un set de entrenamiento y otro set de pruebas esto para probar posteriormente la eficiencia y precisión del modelo con los datos de prueba. Por lo que se procedió a definir un 80% de las imágenes para el set de entrenamiento y un 20% para el set de pruebas como lo muestra la siguiente imagen de [Roboflow](#) (Aplicación usada para el etiquetado y la aumentación de los datos).

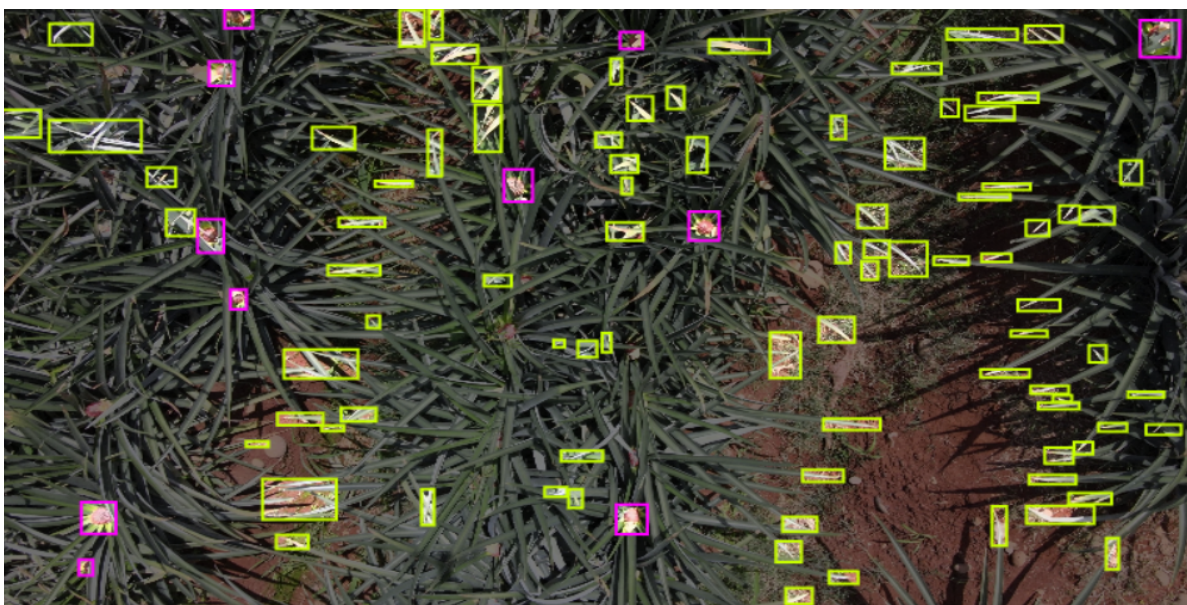


Figura 4.3: Etiquetado de imagen

Dicha herramienta llamada RoboFlow permite la anotación de imágenes por medio de rectángulos superpuestos en la zona que se quiere detectar, dándole una clase para que sea reconocida en el entrenamiento. En la [figura \(4.3\)](#) se muestra un ejemplo donde se etiquetan las plantas piñas con ambas clases “Fusarium” de color amarillo y “Phytophthora” con el color morado.

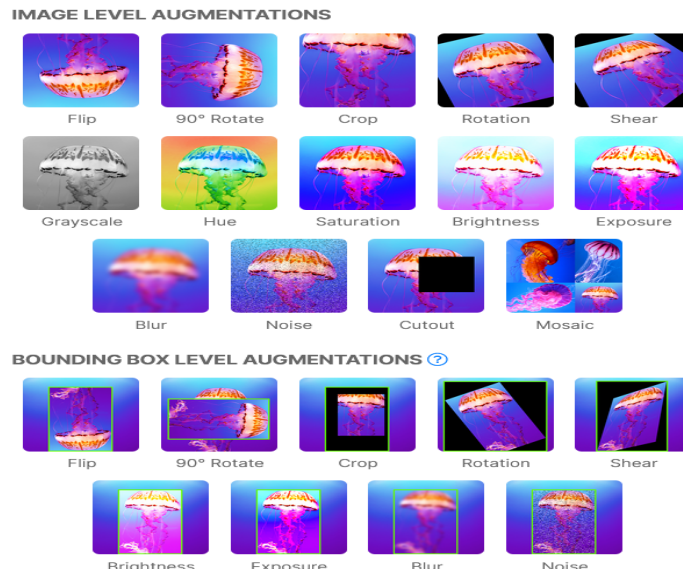


Figura 4.4: Aumentación de la data

Una vez terminado el proceso de etiquetado se procederá a realizar la aumentación de la data mediante diferentes técnicas como las mostradas en la [figura \(4.4\)](#) desde el mismo sistema de RoboFlow, con el objetivo de aumentar significativamente el volumen de datos para obtener un mejor rendimiento del modelo. **(En este caso se pasó de tener 1900 a 4500 imágenes aproximadamente)**

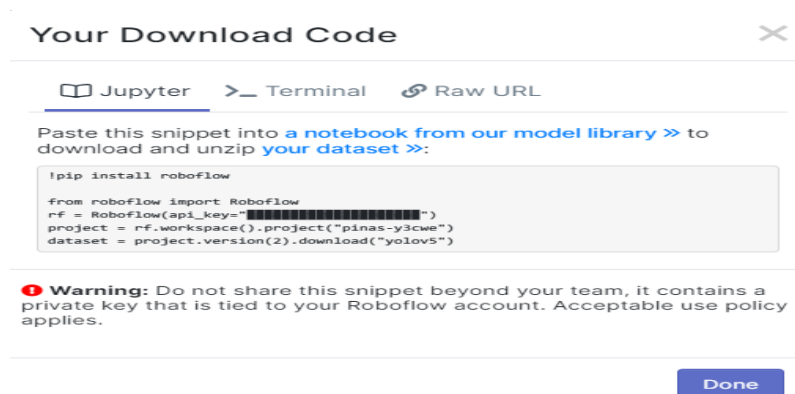


Figura 4.5: Exportación del dataset

Dicho modelo puede ser exportado como en la [figura \(4.5\)](#) desde RoboFlow en diferentes formatos, en este caso elegimos el formato compatible con [Yolov5](#), la cual será la herramienta que utilizaremos para realizar el entrenamiento y posteriormente analizar los resultados para buscar posibles mejoras. El proceso de optimización es uno de los más importantes, ya que una vez entrenado el modelo por primera vez y aplicadas las pruebas respectivas se puede tomar en cuenta los resultados obtenidos para proceder a reentrenar el modelo y así buscar los resultados óptimos para finalmente, al haber concluido con todas las optimizaciones posibles se llega a la etapa final donde se podrán ver resultados del rendimiento del modelo algorítmico mediante la aplicación web a desarrollar como lo muestra el siguiente diagrama de la [figura \(4.6\)](#).

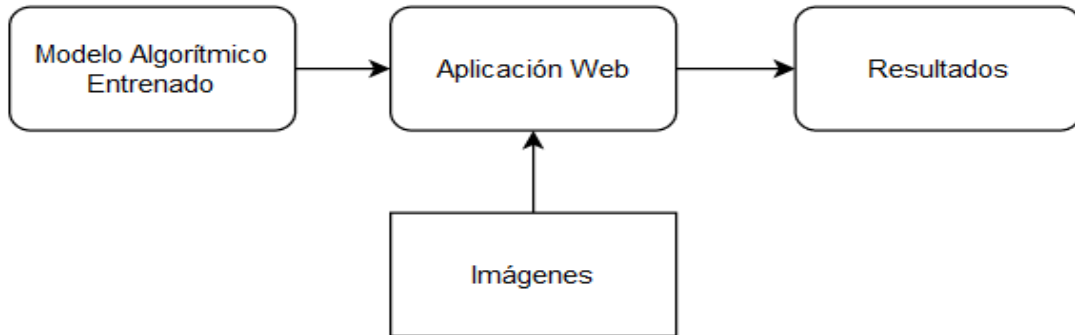


Figura 4.6: Diagrama del sistema

En la siguiente [figura \(4.6\)](#), del diagrama del sistema se puede observar claramente que la aplicación web hará uso del modelo algorítmico previamente entrenado para realizar las predicciones con las imágenes que sean cargadas en la misma aplicación web para posteriormente obtener los resultados y mostrarlos en la misma aplicación web.

5 Plataforma de software

5.1 Aplicación

Para lograr el desarrollo exitoso del sistema o plataforma de software se requiere la implementación de dos áreas específicas para cumplir de la mejor manera el desarrollo del proyecto, las cuales son:

- Desarrollo del modelo algorítmico.
- Desarrollo del servidor y aplicación web.

Desarrollo del modelo algorítmico

Como primer paso para implementar del modelo algorítmico se hará uso de la plataforma de software llamada [Google Colab](#), la cual es una plataforma que permite a cualquier usuario escribir y ejecutar código arbitrario de Python en el navegador, sin una configuración requerida, con acceso gratuito a GPU y con la facilidad para compartir tus proyectos.

```
#clone YOLOv5 and
!git clone https://github.com/ultralytics/yolov5 # clone repo
%cd yolov5
%pip install -qr requirements.txt # install dependencies
%pip install -q roboflow

import torch
import os
from IPython.display import Image, clear_output # to display images
```

Figura 5.1: Importación de dependencias

Como primer paso dentro de Google Colab, instalaremos algunas de las dependencias necesarias en el proyecto como lo son [Yolov5](#), [Roboflow](#) y [Torch](#), esta última es una biblioteca de código abierto para aprendizaje automático que posee una amplia gama de algoritmos de aprendizaje profundo que nos serán de gran ayuda.

```

from roboflow import Roboflow

# set up environment
os.environ["DATASET_DIRECTORY"] = ""/content/datasets""

rf = Roboflow(api_key="claveRepositorioRoboflow")
project = rf.workspace().project("pinas-y3cwe")
dataset = project.version(2).download("yolov5")

```

Figura 5.2: Importación del dataset

El siguiente paso es obtener el daset exportado anteriormente en la [figura \(4.5\)](#) desde la plataforma de Roboflow hacia nuestro ambiente de desarrollo en Colab, para el cual es necesario agregar las claves de nuestro repositorio.

```

!python train.py --img 512 --batch 16 --epochs 20 --data {dataset.location}/data.yaml --weights yolov5s.pt --cache

```

Figura 5.3: Entrenamiento del modelo

En este caso gracias a las facilidades que nos ofrece Yolov5, podemos entrenar nuestro modelo de una manera muy sencilla, simplemente se deben indicar algunas configuraciones como cantidad de épocas, resolución de la imagen o tamaño del batch entre otras.

El tiempo de entrenamiento del modelo dependerá de varios factores tales como el rendimiento del ambiente de Colab (se puede usar GPU o TPU), el tamaño del dataset (en este caso 4500 imágenes), la cantidad de épocas (número de veces que el algoritmo de aprendizaje ve el conjunto de datos completo, 20 en este caso) entre otros.

Este modelo será utilizado en el servidor web, el cual hablaremos a continuación:

Desarrollo del servidor y aplicación web

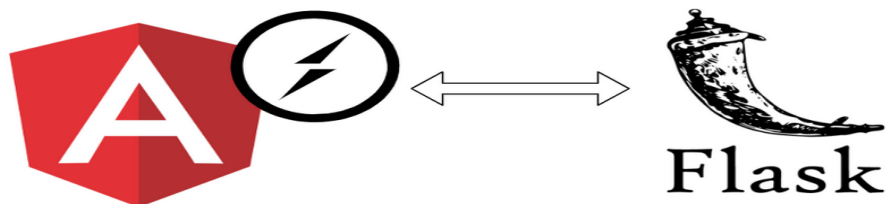


Figura 5.4: Arquitectura de la plataforma

En este caso la implementación del servidor web será mediante el lenguaje de programación Python, esto para aprovechar las facilidades que nos brinda el mismo en este mundo del aprendizaje profundo, junto a tecnologías como [Flask](#), [Socketio](#) y [Docker](#) entre otras del lado del servidor, por otra parte, el cliente o aplicación web será desarrollado en un framework de javascript llamado [Angular](#), seleccionado por su rapidez y agilidad para desarrollar aplicaciones de una sola página, como la requerida en este proyecto.

Servidor web

Es el encargado de cargar y utilizar el modelo desarrollado anteriormente mediante la ayuda del repositorio de Yolov5, el cual es necesario para ejecutar los archivos 'detect.py' y nuestro modelo llamado 'best.pt'. En la [figura \(5.5\)](#) podemos observar las funciones más significativas, llamadas 'recognition' la cual es la encargada de probar el modelo con el set de datos de prueba (Imágenes) y 'getImage', la cual obtiene los resultados de las pruebas realizadas y las envía como respuesta al cliente web (Aplicación web).

```
def recognition():
    process = subprocess.Popen("python "+pathDetect+" --weights "+pathBest+" --source "+pathTest, shell=True)
    process.wait()

def getImage():
    recognition()
    os.chmod("yolov5\\runs\\detect\\exp"+getLasFolder('yolov5/runs/detect')+'\\', 755)
    pathResults = os.path.join(os.path.dirname(__file__), "yolov5\\runs\\detect\\exp"+getLasFolder('yolov5/runs/detect')+'\\')
    resultsList=[]
    for filename in os.listdir(pathResults):
        with open(pathResults+filename, "rb") as f:
            im_b64 = base64.b64encode(f.read())
            data=im_b64.decode('utf-8')
            jsonData = {
                "img": data,
            }
            resultsList.append(jsonData)
    socketio.emit('getImage',resultsList, broadcast=True)
```

Figura 5.5: Servidor web

Aplicación web

Es la encargada de cargar los datos de prueba, ya sea cargar una imagen específica o bien cargar una carpeta con imágenes dentro, para posteriormente mostrar al usuario los resultados obtenidos desde el servidor web. Cabe recalcar que para una respuesta instantánea se hace uso de la librería [SocketIO](#), esto para ofrecer una mejor experiencia de usuario.

FungiDeTECtor



Seleccione una opción

Imagen Carpeta

Click para seleccionar una Imagen

Figura 5.6: Aplicación web

A continuación los resultados obtenidos tras el desarrollo del modelo algorítmico y la plataforma web con su respectiva evaluación.

5.2 Evaluación

Para determinar qué tan bien funciona el modelo desarrollado, es normal pensar que basta con calcular cuán exacto es el modelo prediciendo, para esto tenemos la exactitud o en inglés, el Accuracy. El cual mide el porcentaje de casos que el modelo ha acertado, pero para realizar este tipo de evaluaciones a modelos predictivos, el Accuracy no es la mejor opción ya que nos puede llevar al engaño, es decir, puede hacer que el modelo parezca mucho mejor de lo que realmente es, ya que el Accuracy solo es útil cuando se tienen clases balanceadas que cuenten con un número de ejemplos similares.



Figura 5.7: Balanceo de Clases

En este caso como podemos observar en la [figura \(5.7\)](#) obtenida de Roboflow que este no es el caso, pues la clase “Fusarium” tiene significativamente más peso o cantidad de imágenes que la clase “Phytophthora”. Según [Barrios \(2019\)](#) la mejor forma de analizar los modelos predictivos convolucionales es mediante las métricas de los gráficos de Precisión, Sensibilidad (Recall) y Puntuación F1.

Por lo que para realizar una correcta evaluación de la plataforma procederemos al proceso de evaluación con dichas métricas.

Precisión

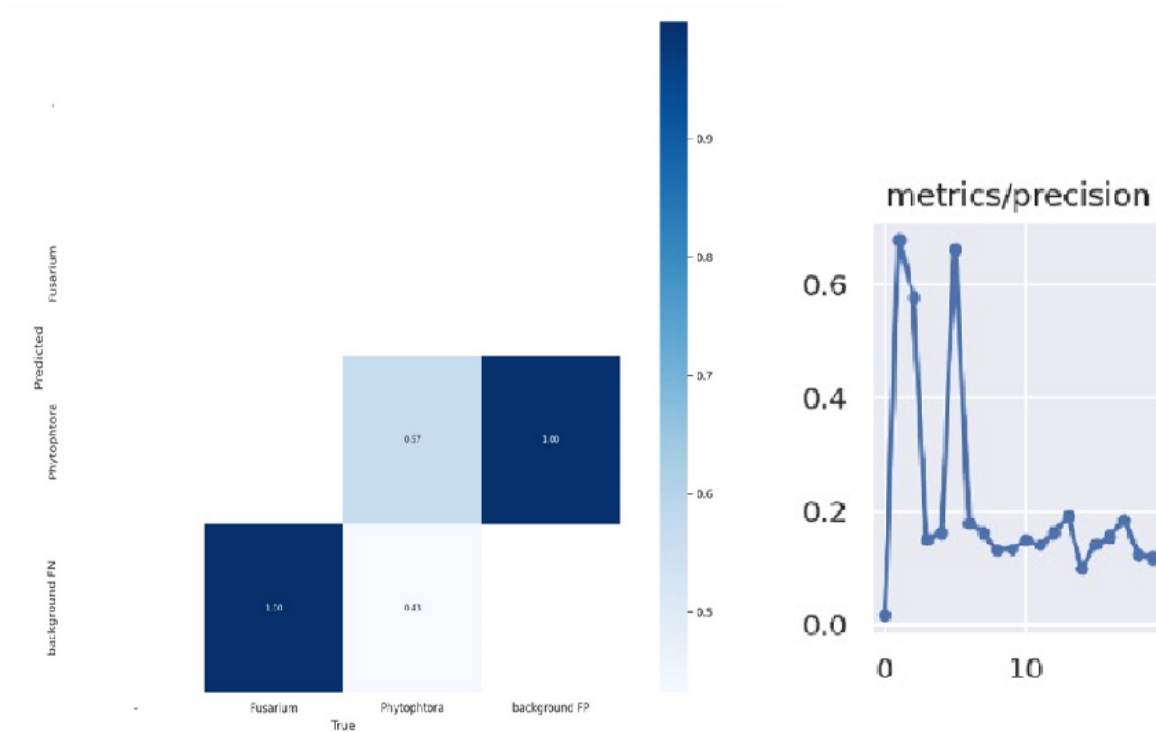


Figura 5.8: Precisión

En este caso con la métrica de precisión podemos medir la **calidad** del modelo de machine learning desarrollado para esta tarea de clasificación. Para este caso el modelo es capaz de predecir un 57% de las plantas con la enfermedad de “Phytophthora” y prácticamente no es capaz de predecir las plantas de piñas con la enfermedad de “Fusarium”, esto a pesar de que hayan muchas más imágenes de esta última.

Estos resultados pueden ser causados por factores como sobreentrenamiento, en donde a la hora de entrenar el modelo, el mismo no es capaz de estabilizarse para aprender de una buena manera. También hay otros factores como las mismas imágenes etiquetadas, pues para la enfermedad de “Fusarium” las imágenes son muy diversas ya que existen muchas formas en la que se puede encontrar o representar esta enfermedad, ya sea en las puntas de las plantas o

en la parte alta de la misma entre otras, además al ser este hongo de color blanco es posible que se confunda con la claridad del sol al reflejar objetos, por lo que a pesar de tener más cantidad de ejemplares no son fácilmente reconocidos por el modelo a diferencia de la enfermedad de “Phytophthora”, la cual es significativamente mas fácil de diferenciar, como se puede observar en la siguiente [figura \(5.9\)](#), en donde los cuadros amarillos se refieren a “Fusarium” y los morados a “Phytophthora”.



Figura 5.9: Reconocimiento de clases

Recall (Sensibilidad o Exhaustividad)

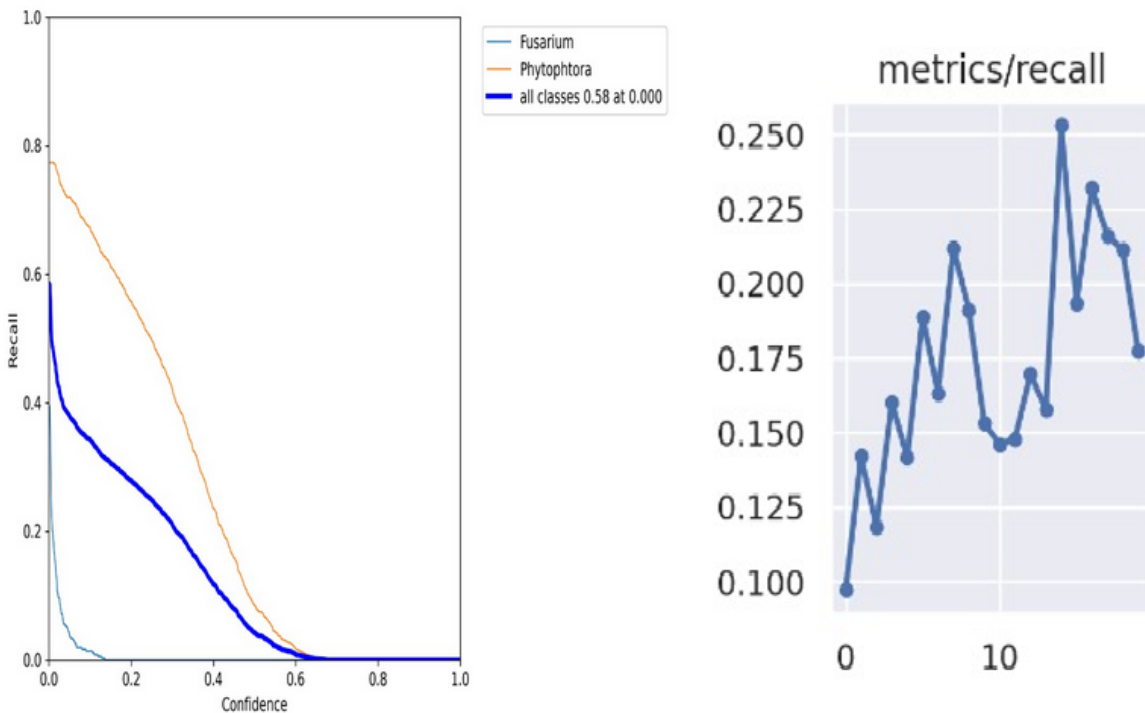


Figura 6.0: Recall

La métrica de sensibilidad o Recall nos va a informar sobre la proporción de casos positivos que fueron correctamente identificadas por el algoritmo. En este caso podemos observar que desde el inicio del entrenamiento el rendimiento del modelo se mantuvo empeorando a través cada una sus épocas, dado a la poca eficacia a la hora de identificar la enfermedad de “Fusarium”, esto explica el sobreentrenamiento innecesario lo cual hace que el modelo no aprenda de una manera eficiente.

Puntuación F1

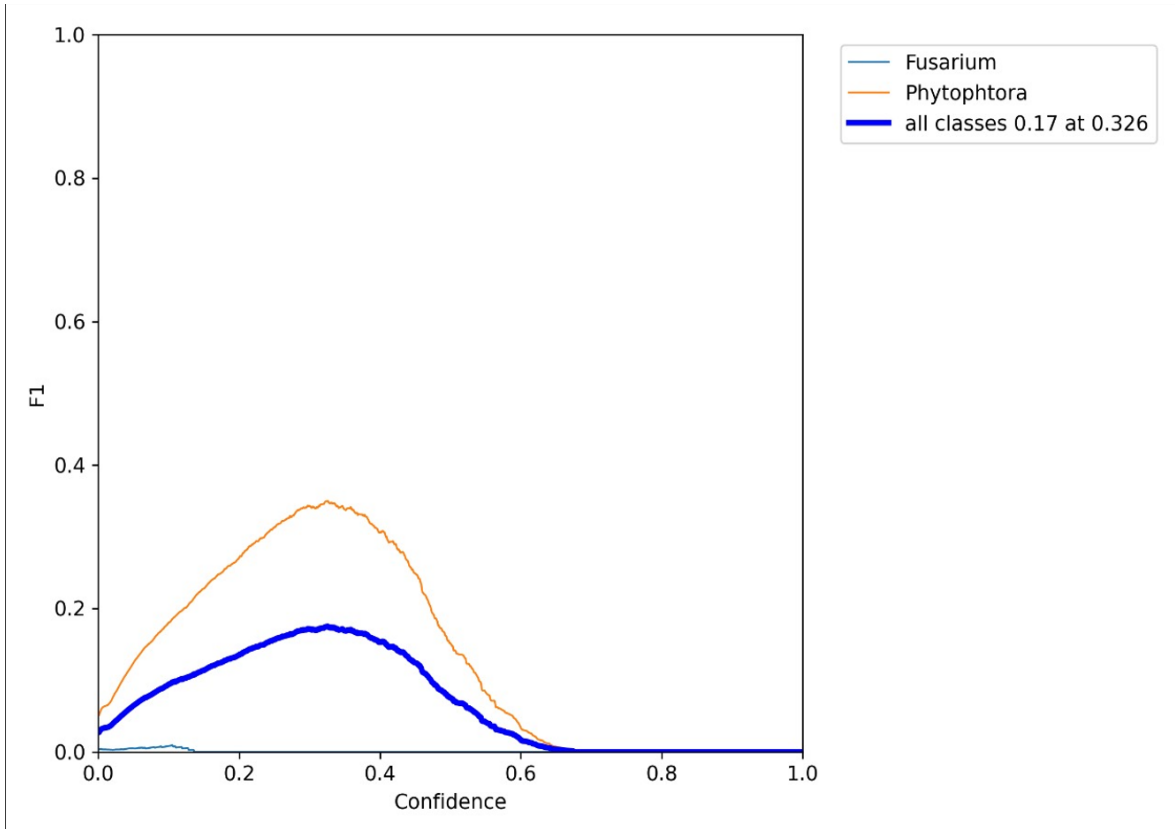


Figura 6.1: Puntuación F1

La puntuación F1 es una forma de combinar la precisión y el recall del modelo, y se define como el promedio armónico de la precisión y el recall, que para este caso tenemos un resultado de **0.326** lo cual deja mucho que desear pero nos brinda una mejor visión más realista del rendimiento del modelo algorítmico ya que como mencionamos anteriormente es la mejor manera de evaluar los modelos cuando se están usando **clases desbalanceadas**.

Resultados Obtenidos

A continuación una muestra de algunos de los resultados obtenidos tras probar la plataforma desarrollada.



Figura 6.2: Ejemplo resultado 1



Figura 6.3: Ejemplo resultado 2



Figura 6.4: Ejemplo resultado 3

En las anteriores figuras se puede observar que el rendimiento de modelo algorítmico no es el mejor, aunque de todas formas sí es posible reconocer las enfermedades de “Fusarium” y “Phytophthora” en las plantas de piñas, principalmente en las plantas con “Phytophthora” ya que se obtuvo un mejor rendimiento a la hora de entrenar el modelo con esta clase, no es el resultado deseado, puede mejorar, pero es un resultado razonable para la poca cantidad de imágenes de la misma en relación a la cantidad de imágenes de las plantas con la enfermedad de “Fusarium”, la cual fue la que tuvo los peores resultados.

Conclusiones

En conclusión se logró abarcar todos los requerimientos solicitados tras el desarrollo de la plataforma de software.

Se desarrolló el diagrama de arquitectura del sistema para una mejor comprensión de la plataforma a desarrollar, desde la creación del dataset, pasando por la creación del modelo y hasta su uso dentro de la plataforma web y todo lo que conlleva ello.

Se realizó de manera exitosa el desarrollo del conjunto de fotografías de piña etiquetadas (Dataset) para ser utilizado por un modelo de aprendizaje automático que logre la detección de las enfermedades “Fusarium” y “Phytophthora” en las plantas de piñas.

Se implementó un modelo de aprendizaje automático que permite detectar las enfermedades “Fusarium” y “Phytophthora” en las plantas de piñas a través de fotografías, mediante el uso de técnicas como “Localización”, ambos reconocimientos obtuvieron diferentes resultados, siendo la enfermedad de “Phytophthora” la más fácil de reconocer por el modelo de aprendizaje automático.

Se desarrolló de manera exitosa una plataforma web en la que los usuarios pueden interactuar de una manera sencilla para hacer pruebas del rendimiento del modelo algorítmico y obtener los resultados en la misma plataforma luego de realizado el análisis correspondiente.

Recomendaciones

Se recomienda definir más requerimientos para la continuación del proyecto a modo de hacerlo más escalable y aprovechar la plataforma desarrollada como guía o inspiración para la elaboración de futuros proyectos relaciones dentro de la zona norte costarricense para promover el avance o desarrollo tecnológico de la zona.

Para una mayor escalabilidad y eficiencia se recomienda en caso de continuar el proyecto, mantener una buena estructura en el diagrama de la arquitectura del sistema para una mejor comprensión de las nuevas funcionalidades a desarrollar.

Para el mejoramiento del dataset desarrollado, se recomienda agregar más imágenes de ambas categorías (entre más mejor) y su respectivo etiquetado, para tratar de balancear la cantidad de imágenes de cada tipo (ya que existe un desbalance) para apuntar a obtener mejores resultados en el desarrollo del modelo de aprendizaje automático o aplicar técnicas específicas para el balanceo de clases.

Para desarrollar una mejor versión del modelo algorítmico se recomienda de igual manera tratar de tener lo más equilibrado o balanceado posible el dataset a utilizar, aplicando diversas técnicas de aumentación de la data y probando nuevas librerías o alternativas a Yolov5 que tal vez pueden ofrecer un mejor rendimiento que el obtenido en este proyecto.

Finalmente se recomienda agregar una mayor accesibilidad a la plataforma web desarrollada con tecnologías compatibles con dispositivos utilizados para personas con diferentes capacidades como personas no videntes por ejemplo, así también como mejorar la experiencia de usuario de la página web en general.

Referencias

- Almansa, E. (2020, 18 septiembre). *¿Para qué se utilizan los drones con inteligencia artificial?* Canal Informática y TICS.
<https://revistadigital.inesem.es/informatica-y-tics/drones-con-inteligencia-artificial/>
- Angular. (2021). Angular Documentation. <https://angular.io/>
- Castro, W. A. Z. (2020). Detección y clasificación de células normales de la sangre periférica usando aprendizaje profundo. Universitat Oberta de Catalunya.
- Costa, C. D. (2020, marzo 24). Best python libraries for machine learning and Deep Learning. Towards Data Science. Recuperado el 18 de septiembre de 2021, de <https://towardsdatascience.com/best-python-libraries-for-machine-learning-and-deep-learning-b0bd40c7e8c>
- Barrios, J. (2019, 26 julio). *La matriz de confusión y sus métricas*. Juan Barrios.
<https://www.juanbarrios.com/la-matriz-de-confusion-y-sus-metricas/>
- Drenth, A., & Guest, D. (1). Phytophthora: la destructora de plantas. *Revista Palmas*, 34, 49-56. Recuperado a partir de <https://publicaciones.fedepalma.org/index.php/palmas/article/view/10666>

Empowering App Development for Developers. (2021). Docker.

<https://www.docker.com/>

Garralda, L. A. (2018). Localización de objetos en imágenes mediante técnicas de aprendizaje profundo [Universidad de la Rioja]. Recuperado el 18 de septiembre de 2021, de https://biblioteca.unirioja.es/tfe_e/TFE003098.pdf

Gutierrez, C. L. (2019). E . T . S . de Ingeniería Industrial , Informática y de Telecomunicación Detección de armas en vídeos mediante técnicas de Deep Learning Grado en Ingeniería Informática Trabajo Fin de Grado Resumen. Universidad Pública de Navarra.

Lazo, W. A. A. (2019). ESPECTROSCOPIA CON INFRARROJO Y TÉCNICAS DE MACHINE LEARNING Y DEEP LEARNING PARA LA DETECCIÓN Y CLASIFICACIÓN DE FRUTAS PARA LA AGROINDUSTRIA. CASO: ARÁNDANOS - EMPRESA TalSA -2018 ” [Universidad Privada Antenor Orrego]. Recuperado el 18 de septiembre de 2021, de http://www.gonzalezcabeza.com/documentos/CRECIMIENTO_MICROBIANO.pdf

Google Colaboratory. (2021a). Google Colaboratory.
<https://colab.research.google.com/>

Olivares, J. L. T. (2019). *Aprendizaje profundo para la detección de árboles en imágenes aéreas de un cultivo de higo capturadas con un dron.*

Recuperado el 18 de septiembre de 2021, de
<http://riaa.uaem.mx/handle/20.500.12055/1006>

Roboflow: Give your software the power to see objects in images and video. (s/f).

Roboflow.com. Recuperado el 18 de septiembre de 2021, de
<https://roboflow.com/>

Saha, S. (2018, diciembre 15). A comprehensive guide to convolutional neural networks — the ELI5 way. Towards Data Science. Recuperado el 18 de septiembre de 2021, de

<https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

Sharma, P. (2019, 21 agosto). *Image Classification vs. Object Detection vs. Image Segmentation.* Medium.

<https://medium.com/analytics-vidhya/image-classification-vs-object-detection-vs-image-segmentation-f36db85fe81>

Socket.IO. (2021). Socket.IO Documentation. <https://socket.io/>

torch. (2021, 21 octubre). PyPI. <https://pypi.org/project/torch/>

Urías, C. (2021). *Organismo Internacional Regional de Sanidad Agropecuaria.* oirsa. <https://www.oirsa.org/informacion.aspx?id=85>

Welcome to Flask. (2021). Flask Documentation (2.0.x).
<https://flask.palletsprojects.com/en/2.0.x/>

YOLOv5 Documentation. (2020). What Is YOLOv5. <https://docs.ultralytics.com/>