



REVISTA INGENIO

Inteligencia Artificial: Machine Learning, para Detección Temprana de Plagas y Enfermedades de Cultivos Básicos, Nicaragua

Artificial Intelligence: Machine Learning for Early Detection of Pests and Diseases in Basic Crops, Nicaragua

Saira María Urbina Cienfuegos | Universidad Nacional Autónoma de Nicaragua- UNAN, Managua-Nicaragua
Jazcar Josué Bravo Rivas | Universidad Nacional Autónoma de Nicaragua- UNAN, Managua-Nicaragua

Recibido: 23/7/2024
Recibido tras revisión: 15/10/2024
Aceptado: 11/11/2024
Publicado: 03/01/2025

PALABRAS CLAVE

Machine learning, plagas, enfermedades, cultivos.

KEY WORDS

Machine learning, pests, diseases, crops

RESUMEN

El presente artículo, muestra aspectos relevantes del proceso de desarrollo de la aplicación móvil que incorpora técnicas de Machine Learning para detectar de forma temprana plagas y enfermedades en cultivos de granos básicos como maíz, frijol y sorgo, estos son indispensables para el consumo humano en Nicaragua. Se utilizó metodología de desarrollo ágil Scrum, se adoptaron tecnologías como Android Studio, lenguaje de programación Java, Google Teachable Machine para entrenamiento del modelo de aprendizaje automático y TensorFlow Lite para incorporar modelo en la aplicación móvil. Los resultados muestran un Sprint con sus historias de usuarios, estas se convirtieron en funcionalidades que incluyen el modelo para el reconocimiento de imágenes con precisión de 95.8% utilizando un conjunto de datos de 252 imágenes de cultivos sanos y enfermos. La metodología indica organización de la programación según patrón Modelo – Vista – Controlador y métricas utilizadas por el modelo. Las conclusiones hacen énfasis en detalles de los resultados obtenidos en Sprint#1. Al final, también se mencionan retos a superar al aplicar aprendizaje automático en el sector agrícola.

ABSTRACT

The present article highlights relevant aspects of the development process of the mobile application that incorporates Machine Learning techniques to early detect pests and diseases in staple grain crops such as corn, beans, and sorghum, which are essential for human consumption in Nicaragua. Agile development methodology Scrum was used, technologies such as Android Studio, Java programming language, Google Teachable Machine for training the machine learning model, and TensorFlow Lite for incorporating the model into the mobile application were adopted. The results show a Sprint with its user stories, which were turned into functionalities that include the model for image recognition with an accuracy of 95.8% using a dataset of 252 images of healthy and diseased crops. The methodology indicates the organization of programming according to the Model-View-Controller pattern and the metrics used by the model. The conclusions emphasize details of the results obtained in Sprint#1. In the end, challenges to overcome in applying machine learning in the agricultural sector are also mentioned.

I. INTRODUCCIÓN

La agricultura es un sector fundamental en la economía nicaragüense porque constituye la fuente principal de alimentos básicos para el consumo humano. Sin embargo, cada año se ve afectada debido a que los cultivos son atacados por plagas y enfermedades [1]. Es complicado controlarlas y evitar que se propaguen de forma exponencial, generando pérdidas considerables, debido a que se reduce el volumen de producción previsto lo que repercute de manera significativa en la economía y seguridad

alimentaria del país. Los agricultores pueden hacer frente a esta problemática en etapa temprana, apoyándose en tecnologías de precisión y monitoreo, que les permitan optimizar el uso de recursos y mejorar la eficiencia en la gestión de cultivos [2].

En el año 2022, la FAO (Organización de las Naciones Unidas para la Alimentación y la Agricultura) estimó que las plagas y enfermedades de las plantas provocan pérdidas de hasta el 40 % de los cultivos alimentarios, y los daños que generan en

la agricultura agravan el problema existente del aumento del hambre mundial y amenazan los medios de vida rurales [3]. Sin embargo, en Nicaragua, por ahora no se dispone de estadísticas sobre las pérdidas ocasionadas por plagas y enfermedades en los granos básicos. Asimismo, la falta de datos y las pocas herramientas e instrumentos apropiados para la recolección de información dificulta la elaboración de planes de contingencia efectivos por parte de agricultores y autoridades.

El avance significativo de la agricultura de precisión y el desarrollo digital ha hecho posible el monitoreo en tiempo real haciendo uso de imágenes satelitales, maquinaria autónoma, drones y sensores, asimismo, el análisis oportuno de datos y la aplicación precisa de insumos en los cultivos [4]. De igual forma, la aplicación de técnicas de Inteligencia Artificial (IA), tales como el aprendizaje automático (Machine Learning) con capacidad de analizar y clasificar grandes volúmenes de datos utilizando algoritmos como las Redes Neuronales Convulsionales (Convolutional Neural Networks - CNN) que se caracterizan por resolver problemas asociados con la clasificación y reconocimientos de imágenes y audio [5], por esta razón, se utilizan para identificación temprana de daños en cultivos causados por plagas o enfermedades, de esta forma apoyar a los agricultores con la gestión eficiente de sus recursos.

El uso de la tecnología representan un avance crucial en la gestión agrícola y es posible incorporarla en aplicaciones que se convierten en agentes que monitorean cultivos, proporcionando respuestas oportunas para minimizar daños y proteger el rendimiento de los mismos [6]. En diversas partes del mundo se han creado soluciones acertadas a problemas agrícolas y coinciden en que la introducción de la IA y el aprendizaje automático en la agricultura tiene el potencial de generar avances previamente inalcanzables en la optimización del rendimiento, la detección de enfermedades y la gestión de cultivos [7]. Los estudios consultados desarrollaron modelos entrenados de aprendizaje automático con CNN para identificar plagas y enfermedades en variedad de cultivos, en su mayoría de consumo humano frecuente, como arroz, frijol y hortalizas [5].

Ante esta problemática aquí descrita y aprovechando las bondades de las tecnologías actuales, este estudio tiene como objetivo el desarrollo de una aplicación móvil que incorpora un modelo de aprendizaje automático (Machine Learning) entrenado con imágenes de cultivos (maíz, frijol y sorgo) sanos y enfermos para la detección de plagas y enfermedades. Estos en la actualidad enfrentan amenazas importantes, como el gusano cogollero en maíz, la antracnosis en frijol y la mosca del sorgo

que reducen su productividad y afectan la seguridad alimentaria del país [8].

2. MÉTODO

El tipo de investigación es mixta debido a la utilización de enfoques cuantitativos y cualitativos. De acuerdo al estudio a través del conocimiento, es descriptiva debido a que se detallan nombres, características y otros elementos referentes a cultivos, plagas y enfermedades, lo que permitió especificar requerimientos a cumplir en el software, enfocado en el problema identificado [9].

Por otra parte, es un estudio retrospectivo, debido a que la identificación de requerimientos para la aplicación móvil se llevó a cabo durante el año 2023; y según su cobertura es de corte transversal.

Este estudio está enfocado en identificar seis plagas y tres enfermedades comunes en tres cultivos, maíz, frijol y sorgo. Por ello, se recolectaron imágenes de cultivos sanos, las cuales fueron tomadas directamente en el campo en compañía de agricultores. También se obtuvieron imágenes que muestran afectaciones de estas plagas y enfermedades, estas fueron facilitadas por técnicos de campo, en total se utilizaron como muestra 252 imágenes. Además, se consultó manual del Tecnológico Nacional (INATEC) para obtener características que tienen cultivos afectados por plagas y enfermedades.

2.1 TECNOLOGÍAS SELECCIONADAS

Según las funcionalidades a implementar se seleccionaron las tecnologías:

- Android SDK como Entorno de Desarrollo Integrado (IDE) para aplicaciones móviles [10].
- Java, lenguaje de programación orientado a objetos, con bibliotecas dispuestas para su uso [11].
- Google Teachable Machine, herramienta basada en la web para entrenamiento de modelos de Machine Learning, utiliza Redes Neuronales Convulsionales (CNN), estas aprenden estructuras jerárquicas, niveles de representación y abstracción para comprender los patrones de datos que provienen de una fuente, en este caso imágenes. Utiliza aprendizaje supervisado, los datos facilitados para el entrenamiento incluyen la solución deseada llamada etiqueta [12], [13].
- TensorFlow Lite, facilita la ejecución de modelos de aprendizaje automático para el despliegue del modelo de Machine Learning en dispositivos móviles, permitiendo trabajar sin conexión a Internet [14].
- Cloud Firestore y Realtime Database, servicios de bases de datos en la nube, que permiten almacenamiento y sincronización de datos en tiempo real [15].

La selección de estas tecnologías se realizó con base en la experiencia de uso y la curva de aprendizaje del equipo de trabajo.

2.2 ENTRENAMIENTO DEL MODELO MACHINE LEARNING

Se consideraron características relevantes para la selección de la tecnología utilizada en el entrenamiento del modelo de Machine Learning (ML), que permite reconocer imágenes referentes a plagas y enfermedades en los cultivos básicos. Se optó por Google Teachable Machine; tiene una interfaz visual, flexible y accesible, no requiere experiencia en ML o codificación, facilitando el entrenamiento e implementación de modelos rápidamente [16]. En [17] se indica que esta tecnología puede generar un modelo de aprendizaje automático con una tasa de precisión de prueba de hasta el 100%, lo que fue considerado importante por el equipo de trabajo.

El proceso de entrenamiento del modelo que utiliza Red Neuronal Convulsional (CNN) en Google Teachable Machine, enfocado en la identificación de plagas y enfermedades en cultivos de maíz, frijol y sorgo consistió en realizar un pilotaje con 252 imágenes clasificadas en 12 clases (neuronas), a cada clase se asignaron 21 muestras, para mantener un número equilibrado de imágenes, evitar sesgos y asegurar la precisión del modelo. Las imágenes fueron preprocesadas por la herramienta a tamaño uniforme de 224x224 píxeles. La tabla 1 muestra detalles de la cantidad de imágenes seleccionadas por clase.

Para que el modelo pueda identificar correctamente enfermedades o plagas, se seleccionaron imágenes claras y de alta calidad, enfocando de forma directa en una muestra del daño en la planta o plaga que la afecta, hacer tomas con escasa iluminación, bajo lluvia o condiciones de neblina, puede afectar la calidad de las imágenes. En la tabla 2 (Anexo tabla2) se muestran detalles que presentan los cultivos según enfermedad o plaga que los afecta.

Para medir el rendimiento del modelo al clasificar las muestras en Google Teachable Machine se utiliza:

- Matriz de confusión, muestra el grado de precisión de las predicciones, consiste en una tabla que resume el número de predicciones correctas e incorrectas realizadas por el modelo [18]. La tabla 3, muestra variables utilizadas por la matriz que permiten calcular otras métricas como precisión y exactitud (accuracy) del modelo.
- Precisión por clase, se calcula usando muestras de prueba (15% de los datos), Se dividen los verdaderos positivos (VP) entre la suma de VP y los Falsos Positivos (FP) [18].

$$\text{Precisión} = \frac{\text{Clasificaciones correctas}}{\text{Clasificaciones positivas}} = \frac{VP}{VP+FP} \quad (1)$$

- Exactitud (Accuracy) del modelo al clasificar, corresponde a proporción de predicciones correctas positivas y negativas [18], la fórmula utilizada es la siguiente:

$$\text{Exactitud} = \frac{\text{Clasificaciones correctas}}{\text{Total de clasificaciones}} = \frac{VP + VN}{VP + VN + FP + FN} \quad (2)$$

- Precisión de época, indica el porcentaje de clasificaciones que un modelo acierta durante la preparación.

2.3 Desarrollo de aplicación

Se utilizó Metodología Ágil (Scrum) para organizar el desarrollo de la aplicación móvil; esta ayuda a personas, equipos y organizaciones a generar valor a través de soluciones adaptativas para problemas complejos, contiene artefactos que representan trabajo o valor, de esta forma, Product Backlog representa el objetivo del producto, Sprint Backlog es el objetivo del Sprint y el Increment corresponde a las funcionalidades terminadas [19].

Los requerimientos del proyecto se escribieron como historias de usuarios (HU) y se organizaron por prioridad, de esta forma se obtuvo el Producto Backlog como una lista emergente y ordenada, posteriormente, se dividió en Sprints, generando subproductos terminados. Para cada Sprint se define el Sprint Backlog, este es un plan de trabajo realizado por el equipo de desarrollo, se pueden crear varios incrementos dentro de un Sprint, estos son considerados pequeños avances hacia el objetivo del producto [19].

Para estructurar el código de la aplicación móvil en Android, se seleccionó el patrón Modelo-Vista- Controlador (MVC), que permite organizar el código de la vista y el acceso a datos de forma separada. Esto divide conceptualmente la aplicación en tres elementos y facilita hacer cambios en cualquiera de las partes [20].

3. RESULTADOS Y DISCUSIÓN

La identificación de requerimientos para el desarrollo de la aplicación móvil se realizó utilizando la metodología ágil Scrum. En este proyecto se crearon 18 historias de usuarios, las cuales se organizaron en 5 Sprint, cada uno generó un entregable correspondiente a las funcionalidades completadas en un periodo de tiempo de 3 semanas. Durante esta fase, el equipo de desarrollo trabajó estrechamente con el usuario final (dueño del producto, técnicos de campo y agricultores) quien tiene conocimiento de las necesidades que deben abordarse a través de las funcionalidades del software [21].

En este artículo, se hace referencia a Sprint #1, porque contienen las historias de usuario (HU) particularmente valiosas para el proyecto. Los elementos que tienen may-

Tabla 1.

Clasificación de imágenes por clase

Clasificación	Etiquetado de clases	Cantidad
Cultivo Sano	Frijol	21
	Maíz	21
	Sorgo	21
Enfermedades	Mosaico dorado (BGYMV)	21
	Roya (Puccinia sorghi)	21
	Mancha zonada (Gloeocercospora sorghi)	21
Plagas	Mosca blanca (Bemisia tabaci)	21
	Chicharrita verde (Empoasca kraemer)	21
	Gallina ciega (Phyllophaga sp)	21
	Cogollero (Spodoptera frugiperda)	21
	Mosquita del sorgo (Contarinia sorghicola)	21
	Barrenador mayor del tallo (Diatraea saccharalis)	21
Total		252

Tabla 3.

Variables de matriz de confusión

	Positivo real	Negativo real
Predicción positiva	Verdadero positivo (VP): casos positivos, clasificado correctamente como positivos.	Falso positivo (FP): casos negativos, pero el modelo predijo como positivo.
Predicción negativa	Falso negativo (FN): casos positivos, pero el modelo predijo como negativo.	Verdadero negativo (VN): casos negativos y el modelo predijo como negativos.

or valor para el cliente deben priorizarse al inicio, con la expectativa de ser trabajados de primero [22].

RESULTADOS SPRINT #1

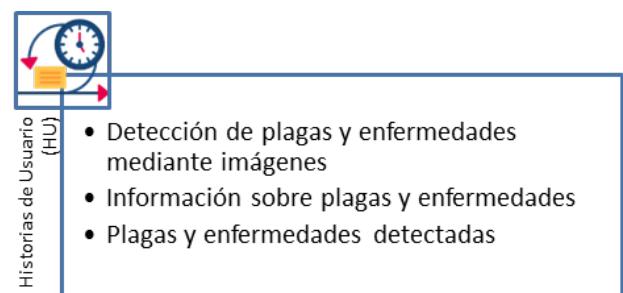
La figura 1 muestra Sprint Backlog del Sprint #1. Se establecieron 3 historias de usuarios, cada una se desglosó en tareas pequeñas. Se completó en un periodo de 3 semanas y se logró el primer subproducto funcional del proyecto.

En el primer Sprint planificado, se obtuvo como resultado un modelo de Machine Learning entrenado que permite detectar plagas y enfermedades específicas de los cultivos: maíz, frijol y sorgo, las principales tareas realizadas fueron:

- Recopilación y clasificación de datos
- Creación del proyecto de aprendizaje automático (ML) en Google Teachable Machine.
- Etiquetado de 12 clases con 21 imágenes para cada una.
- Establecimiento de parámetros en Google Teachable Machine: 21 épocas (Cantidad de veces que se prueban los datos de muestra), 16 lotes (predeterminado) y una tasa de aprendizaje 0.0001 (predeterminado).

Figura 1.

Sprint Backlog #1

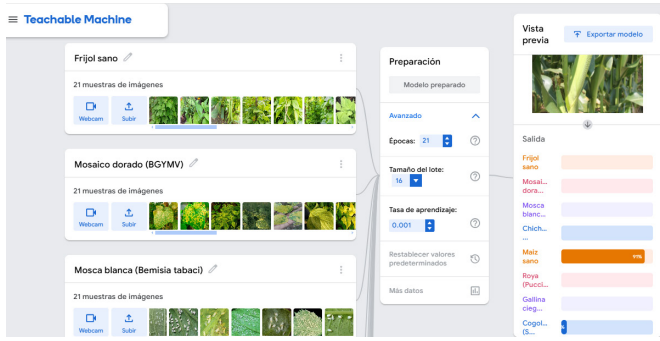


- Preparación y prueba del modelo en Google Teachable Machine.
- Exportación del modelo a TensorFlow Lite
- Incorporación del modelo entrenado a la APP móvil
- Diseño de interfaz de usuario para información y detección de plagas y enfermedades desde la APP móvil.

La Figura 2, muestra el proceso de entrenamiento del modelo de aprendizaje automático (ML) en Google Teachable Machine, subproducto obtenido en Sprint #1.

Figura 2.

Entrenamiento del Modelo ML – Entregable Sprint #1



Como resultado del entrenamiento del modelo de aprendizaje automático (ML) en Google Teachable Machine, la red neuronal ajustó sus pesos y sesgos a lo largo de 21 épocas, procesando 252 imágenes divididas en lotes de 16, lo que resultó en 15 lotes por época. Este ajuste de parámetros permitió que la red aprendiera de la mejor manera posible. La tasa de aprendizaje, considerado un hiperparámetro crítico, se seleccionó cuidadosamente para equilibrar la rapidez del entrenamiento y la estabilidad del modelo. La configuración avanzada en Google Teachable Machine permitió establecer estos parámetros para optimizar el rendimiento.

Durante el proceso de entrenamiento del modelo, se obtuvieron dos resultados relevantes:

- Precisión de las 12 clases igual a 1
- Precisión de 10 clases igual a 1 y 2 clases con precisión 0.75

En ambos procesos de entrenamiento se utilizaron la misma cantidad de clases, cantidad de imágenes, y la misma configuración, sin embargo, refleja resultados diferentes en la precisión por clase. Esto se debe a que el algoritmo selecciona muestras aleatorias para la división de datos del modelo y puede generar leves variaciones. La figura 3 muestra la precisión por clase obtenida en cada proceso, para ambos el algoritmo ha tomado 4 imágenes de las 21 facilitadas por clase como muestra para validar el modelo.

Los modelos de aprendizaje automático en Google Teachable Machine utilizan 85% de los datos para entrenamiento y 15% de datos para validación, aspecto esencial para evaluar y ajustar el modelo. De esta manera, se asegura no solo que aprenda a clasificar correctamente durante el entrenamiento, sino que también generalice bien con datos nuevos.

Como puede observarse en la Figura 3, uno de los resultados presenta una exactitud de 1.00 para todas las clases durante la validación, mientras que el otro muestra variación en dos clases: “Sorgo sano” y “Mosquita del sorgo”, obteniendo en las mismas 0.75 de precisión en ambas. Esto indica que el 83% de las clases las puede predecir correctamente, sin embargo, el modelo presenta dificultades en la clasificación del 17% restante, sugiriendo que podría estar confundiéndolas con otras. Las características

Figura 3.

Precisión por clase

CLASS	ACCURACY	# SAMPLES
Frijol sano	1.00	4
Mosaico dorado (BG...	1.00	4
Mosca blanca (Bemi...	1.00	4
Chicharrita verde ...	1.00	4
Maíz sano	1.00	4
Roya (Puccinia sor...	1.00	4
Gallina ciega (Phy...	1.00	4
Cogollero (Spodopt...	1.00	4
Sorgo sano	1.00	4
Mancha zonada (Glo...	1.00	4
Mosquita del sorgo...	1.00	4
Barrenador mayor d...	1.00	4

CLASS	ACCURACY	# SAMPLES
Frijol sano	1.00	4
Mosaico dorado (BG...	1.00	4
Mosca blanca (Bemi...	1.00	4
Chicharrita verde ...	1.00	4
Maíz sano	1.00	4
Roya (Puccinia sor...	1.00	4
Gallina ciega (Phy...	1.00	4
Cogollero (Spodopt...	1.00	4
Sorgo sano	0.75	4
Mancha zonada (Glo...	1.00	4
Mosquita del sorgo...	0.75	4
Barrenador mayor d...	1.00	4

Nota. Resultados en Google Teachable Machine

similares afectan en gran medida el reconocimiento de objetos, por lo que el valor de precisión puede disminuir y se dividirá en otras muestras de clase [23].

La matriz de confusión obtenida del entrenamiento del modelo que generó diferencias en la precisión de dos clases, se muestra en la Figura 4. Esta tabla tiene un tamaño de 12 filas por 12 columnas, las clases reales previamente etiquetadas en Google Teachable Machine se encuentran en las filas y en las columnas se ubican las predicciones realizadas. En la diagonal principal resaltada en color azul se visualizan las clasificaciones correctas, los valores fuera de la diagonal indican predicciones incorrectas.

El modelo utilizó 4 muestras por clase para la validación. De las 12 clases evaluadas, 10 clasificaron correctamente todas las muestras, sin embargo, las clases “Sorgo sano” y “Mosquita del sorgo” presentaron errores con una muestra al predecirla como “Maíz sano”. En este caso la precisión fue de 0.75 para las dos clases con predicción incorrecta.

Se identificaron en la matriz de confusión valores incorrectos correspondientes a las clases “Sorgo sano” y “Mosquita del sorgo” obtenida producto del entrenamiento:

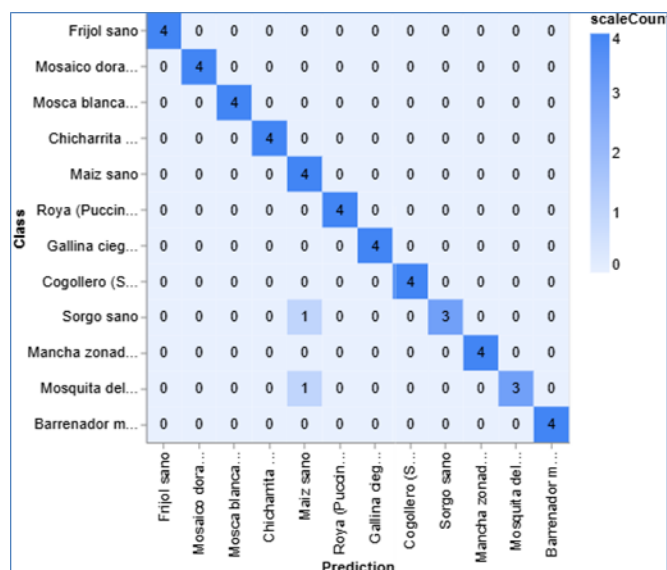
- El modelo clasificó una muestra como “Maíz sano”, pero pertenece a “Sorgo sano”, la precisión para “Sorgo sano” es de 0.75.
- El modelo clasificó una muestra como “Maíz sano”, pero pertenece a plaga “Mosquita del sorgo”, la precisión para “Mosquita del sorgo” es de 0.75.

La tabla 4, muestra el resultado de precisión del modelo por clase, según datos obtenidos de matriz de confusión. Se dividen los verdaderos positivos (VP) entre la suma de los VP y los Falsos Positivos (FP).

De acuerdo a los datos facilitados por la matriz de confusión, se calculó la exactitud (Accuracy) del modelo

Figura 4.

Matriz de confusión



Nota. Resultado de entrenamiento en Google Teachable Machine al realizar las clasificaciones de las muestras, esta métrica corresponde a la proporción de predicciones correctas tanto positivas como negativas. El modelo alcanzó 0.958 de exactitud al clasificar muestras de las clases etiquetadas, en la tabla 5 se muestra el cálculo correspondiente.

La precisión por época del modelo entrenado, obtenida en Google Teachable Machine se puede corroborar en la gráfica mostrada en la figura 5, la línea naranja indica la precisión al realizar las pruebas de validación, esta línea se acerca ligeramente a precisión 1.

3.1.1 RESULTADOS DE PRUEBA PARA MODELO CON PRECISIÓN 1

En la fase de prueba, cuando se alcanzó precisión 1 en todas las clases, el modelo fue evaluado con nuevos datos. Los resultados mostraron variaciones en la precisión dependiendo de la familiaridad del modelo con las clases, la figura 6 muestra prueba realizada con una imagen de la plaga “falsa mancha angular del frijol”, para la cual no se incluyeron datos en el modelo. Se obtuvo la siguiente respuesta:

- Frijol sano: 45%,
- Enfermedad Mosaico dorado: 19%, y
- Plaga “Mosca blanca” 26%.

Estos datos indican que existe cierta similitud en las características de la imagen facilitada y las clases a las que el modelo les está asignando un valor. No obstante, la clase que corresponde a enfermedad “Roya”, para la cual el

Tabla 4.

Cálculo de precisión por clase

Métrica / Fórmula	Clases	Resultado precisión por clase
$Precisión = \frac{VP}{VP + FP}$	1. Sorgo Sano	$Precisión = \frac{3}{3+1} = 0.75$
	2. Mosquita del sorgo	$Precisión = \frac{3}{3+1} = 0.75$
	3. Frijol sano	$Precisión = \frac{4}{4+0} = 1$
	4. Mosaico dorado	
	5. Mosca blanca	
	6. Chicharrita del maíz	
	7. Maíz sano	
	8. Roya	
	9. Gallina ciega	
	10. Cogollero	
	11. Mancha zonada	
	12. Barrenador mayor de tallo alto	

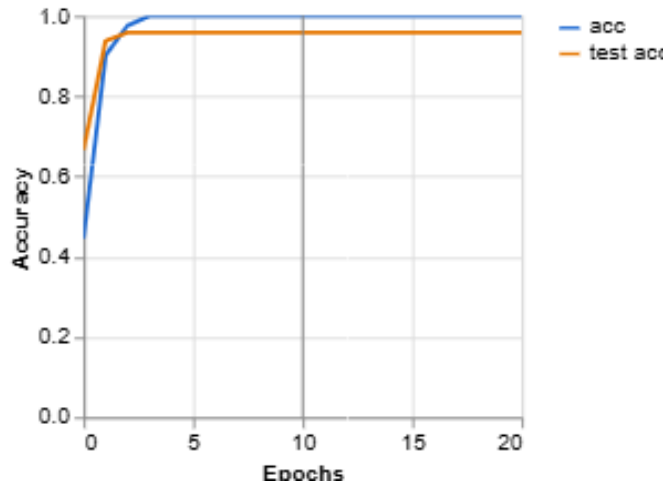
Tabla 5.

Cálculo de exactitud del modelo

Métrica / Fórmula	Accuracy (Exactitud) del modelo
$Exactitud = \frac{VP+VN}{VP+VN+FP+FN}$	$Exactitud = \frac{4 + 4 + 4 + 3 + 4 + 4 + 4 + 4 + 4 + 4 + 4 + 3}{4 + 4 + 4 + 3 + 4 + 4 + 4 + 4 + 4 + 4 + 3 + 1 + 1} = 0.958$

Figura 5.

Precisión por Época



Nota. Resultado de entrenamiento en Google Teachable Machine

modelo tenía datos de entrenamiento, alcanzó una precisión del 100%.

Aunque la precisión del modelo es de 1 en todas las clases, según los resultados obtenidos, se puede inferir que no ha aprendido las características generales de las clases, ya que no generaliza bien con nuevos datos. Es posible que haya memorizado los patrones de las imágenes utilizadas para el entrenamiento debido al pequeño conjunto de datos facilitados. Esto puede indicar que el modelo está sobreajustado; ha aprendido a clasificar las muestras de manera exhaustiva, impidiendo clasificar correctamente las muestras de prueba [24].

3.1.2 RESULTADOS DE PRUEBA PARA MODELO CON PRECISIÓN 0.958

Se realizaron pruebas con el modelo cuando obtuvo precisión de 0.75 para las clases “Sorgo sano” y “Mosquita del sorgo”, se le proporcionaron imágenes de “Mosquita del sorgo” y “Sorgo sano” que no estaban incluidas en las muestras de entrenamiento (Figura 7). Los resultados fueron los siguientes:

- **Muestra de plaga Mosquita del sorgo:** 80% para “Sorgo Sano”
- **Muestra de Sorgo Sano:** 69 % para “Sorgo Sano” y 27% para “Maíz sano”

Aunque el modelo alcanzó una exactitud de 0.958 en general, como se puede apreciar en la figura 7, las clases que obtuvieron precisión 0.75 tienen dificultades al clasificar de forma correcta, esto puede estar asociado a la similitud de características que poseen las imágenes y al

reducido conjunto de datos que ha utilizado el modelo para su entrenamiento.

Durante el proceso de prueba del modelo con imágenes nuevas para las clases “Mosca blanca” y “Gallina ciega” quienes alcanzaron precisión de 1 durante el entrenamiento (figura 8), se obtuvieron los siguientes resultados:

- **Muestra de “Mosca blanca”:** 93% Mosca blanca
- **Muestra de “Gallina ciega”:** 99% Gallina ciega

Para las clases que alcanzaron precisión 1 durante el entrenamiento del modelo, al mostrarles nuevos datos responden de forma acertada a la predicción, logrando más del 90% de exactitud, acercándose a la precisión general de 0.958 obtenida en el modelo.

Al finalizar el proceso de entrenamiento del modelo para este proyecto, se exportó en formato TensorFlow Lite, lo que permitió su implementación en la aplicación móvil. Incorporar Machine Learning como herramienta tecnológica en el sector agrícola, facilita de forma directa en el campo la identificación temprana de plagas y enfermedades optimizando el proceso de monitoreo y control de las mismas.

Por otra parte, la implementación del patrón MVC en la arquitectura de la aplicación permitió una clara separación de las responsabilidades, apartar los datos de la aplicación, la interfaz de usuario y la lógica de negocios, facilita el mantenimiento y la extensibilidad del software por parte del equipo de trabajo [25]. Por tal razón, dentro del Modelo se incluyeron las clases en lenguaje Java, que permiten la representación de los datos, en la Vista se ubicaron todas las interfaces, en Android se definen como Layout. En el Controlador se incorporaron Actividades (Activity) y Fragmentos (Fragment) que gestionan la interacción del usuario entre el Modelo y la Vista. A la estructura del proyecto Android se incorporó una carpeta ML que contiene el modelo entrenado para el reconocimiento de plagas y enfermedades que utiliza la aplicación.

3.1. DISCUSIÓN

Estudios similares que utilizan Red Neuronal Convulsional (CNN) demuestran un alto porcentaje de precisión en el proceso de clasificación de imágenes referente a cultivos afectados por plagas o enfermedades, en este sentido, se ha observado resultados de algoritmo implementado utilizando Red Neuronal Convulsional Modificada (ModCNN) para la detección de enfermedad enrollamiento en hojas de cultivo de tomates y virus mosaico dorado en cultivo de frijol, para los cuales utilizaron 596 imágenes de hojas de frijol (213 sanas y 283 enfermas) y 447 imágenes de tomate (302 sanas y 145 enfermas) alcanzando durante el proceso de prueba una precisión de 0.9769, esto demuestra la robustez de este tipo de tec-

nologías y el aporte significativo que ofrecen al sector agropecuario [5].

Otro estudio sobre clasificación de imágenes para la predicción de enfermedades en hojas del cultivo de arroz utilizando Redes Neuronales Convulsionales (CNN) ha obtenido 95.83 de precisión al realizar validaciones con datos obtenidos del directorio de aprendizaje automático de UCI (Machine Learning Repository / Universidad de California en Irvine), este modelo entrenado se ha incorporado a una interfaz gráfica para ser utilizado por agricultores como usuarios finales [7].

De igual forma, la implementación en Tensorflow de algoritmo con modelo entrenado para detectar plaga mosca blanca en cultivos de hortalizas obtuvo resultados entre 0.81 y 0.94 de precisión, en las distintas pruebas realizadas lo que indica que el algoritmo tiene un alto porcentaje de precisión [6].

Para el proyecto abordado en este artículo se ha utilizado Google Teachable Machine que implementa CNN para la clasificación de imágenes, el modelo entrenado para que identifique plagas y enfermedades en cultivos de maíz, frijol y sorgo ha obtenido una precisión de 0.958, en comparación con modelos similares (tabla 6) se observa que, el modelo con Modificación de Redes Neuronales Convulsionales (ModCNN) alcanzó el valor más alto de precisión, cabe mencionar que el conjunto de datos utilizados en ese proyecto es superior al estudio aquí realizado.

Machine Learning destaca como una tecnología versátil debido a que puede ser aplicada en diferentes sectores y permite desarrollar para cada problema un algoritmo de predicción para un caso de uso particular. Lo interesante de estos algoritmos es que aprenden de los datos con la finalidad de encontrar patrones o tendencias para comprender qué nos dicen estos datos y, a partir de ellos, se construye un modelo que es capaz de predecir o clasificar los elementos. Sin embargo, existen factores relevantes para el buen desempeño del modelo, que se refieren a la calidad y disponibilidad de los datos, el algoritmo puede ser potente, pero no es suficiente si no se le facilitan los datos indicados para entrenarlo [12].

En este estudio se ha identificado como limitante la cantidad de datos reducida utilizada para el entrenamiento del modelo y se requiere un conjunto de datos más amplio de al menos 600 imágenes a más por clase para que pueda generalizar bien con datos nuevos como ha sido comprobado por [5] con ModCNN, para mejorar el rendimiento del modelo, se recolectarán nuevas muestras por clases (cultivos sanos y enfermos) bajo la asesoría de agricultores y técnicos de campo, de igual forma se indagará a cerca del acceso a repositorios de Machine Learning para fortalecer el modelo con datos suficientes, como lo hizo [7] con el proyecto que predice enfermedades en hojas del cultivo de arroz utilizando CNN.

Para reducir el sobreajuste de modelo entrenado, [26] sugiere mejorar la cantidad y calidad de los datos de entrenamiento aplicando métodos comúnmente utilizados, como la transformación de traslación y rotación de las imágenes, se considerará como una acción importante a tomar en cuenta para la optimización de modelo entrenado.

La tecnificación y el control integrado de plagas y enfermedades, contribuye a la producción y productividad del cultivo de maíz [27], aspectos que también pueden aplicarse a otros cultivos básicos. Por ello, utilizar modelo de aprendizaje automático en una aplicación móvil, para identificar daños causados por plagas y enfermedades en los cultivos es una buena estrategia, debido a que, es una tecnología que permiten agilizar el reconocimiento de imágenes correspondiente a afectaciones, de esta forma la atención y el manejo a problemas puede tratarse en fases iniciales, evitar que se extienda y compartir información al respecto con la comunidad, asimismo se minimizan los daños y se contribuye al buen rendimiento de los cultivos.

Incorporar tecnologías en el desarrollo de actividades agrícolas facilita el trabajo y mejora la productividad, sin embargo, la presencia de recurso humano con conocimientos técnicos sigue siendo valioso para asegurar la aplicación adecuada y oportuna.

4. CONCLUSIONES

Los requerimientos relevantes para este proyecto se detallan en 3 historias de usuarios prioritarias ubicadas en un Sprint, este generó un entregable terminado con el modelo entrenado en Google Teachable Machine para detección de plagas y enfermedades específicas en cultivos de maíz, frijol y sorgo.

El modelo entrenado con 252 imágenes como prueba piloto, se incorporó en la aplicación móvil utilizando TensorFlow Lite, en función de facilitar el trabajo a agricultores y técnicos de campo que cosechan, supervisan y ejecutan acciones para controlar el deterioro en cultivos provocados por plagas y enfermedades.

La precisión general alcanzada del modelo es de 0.958 en la detección de plagas y enfermedades de cultivos de maíz, frijol y sorgo, influye en el resultado la forma de capturar la imagen, capacidad de la cámara del dispositivo y la variedad de imágenes que dispone el modelo para clasificar.

El patrón MVC utilizado para estructurar la aplicación permitió obtener una arquitectura organizada, escalable y de fácil manejo del código, aplicar patrones de diseño y arquitectónicos constituye una buena práctica de programación.

A pesar de que el aprendizaje automático es un buen aliado para solventar problemas en el sector agrícola, aún existen muchos retos por superar, como, la aceptación

y disponibilidad de nuevas tecnologías por parte de los agricultores, la adquisición de recursos necesarios para su implementación, la clasificación, calidad y cantidad de imágenes que debe proveerse al modelo para que los resultados sean acertados.

Es importante mencionar que, las aplicaciones desarrolladas para detectar daños en cultivos con Machine Learning deben ser utilizadas por personas que se enfrentan de forma cotidiana a estas dificultades, ellos son los que pueden validar el verdadero aporte de estas. Aunque este tipo de tecnologías facilita la identificación temprana de plagas y enfermedades, el acompañamiento de los técnicos de campo siempre será importante y se debe tomar en cuenta sus recomendaciones.

REFERENCIAS

- [1] O. F. C. Martínez, «Situación del cultivo de maíz en Guatemala: principales limitaciones en la productividad», *ACI Av. En Cienc. E Ing.*, vol. 14, n.º 1, Art. n.º 1, abr. 2022, doi: 10.18272/aci.v14i1.2596.
- [2] N. C. Eli-Chukwu, «Applications of Artificial Intelligence in Agriculture: A Review», *Eng. Technol. Appl. Sci. Res.*, vol. 9, n.º 4, Art. n.º 4, ago. 2019, doi: 10.48084/etasr.2756.
- [3] FAO, «La FAO acoge con satisfacción la decisión de conmemorar cada año el Día Internacional de la Sanidad Vegetal el 12 de mayo», Newsroom. Accedido: 20 de octubre de 2024. [En línea]. Disponible en: <https://acortar.link/uiuR5M>
- [4] R. R. Hernández, «La Agricultura de Precisión. Una necesidad actual», *Ing. Agríc.*, vol. 11, n.º 1, Art. n.º 1, 2021, Accedido: 7 de julio de 2024. [En línea]. Disponible en: <https://www.redalyc.org/articulo.oa?id=586269368010>
- [5] F. Nihar, N. N. Khanom, S. S. Hassan, y A. K. Das, «Plant Disease Detection through the Implementation of Diversified and Modified Neural Network Algorithms», *J. Eng. Adv.*, vol. 2, n.º 01, pp. 48-57, mar. 2021, doi: 10.38032/jea.2021.01.007.
- [6] O. López Gorozabel, R. Malla Valdiviezo, G. Morejón López, y M. León Bravo, «Sistema de detección de plagas en los cultivos», *593 Digit. Publ. CEIT*, vol. 9, n.º 1, pp. 128-137, 2024, Accedido: 10 de julio de 2024. [En línea]. Disponible en: <https://dialnet.unirioja.es/servlet/articulo?codigo=9262986>
- [7] B. Paneru, B. Paneru, y K. B. Shah, «Analysis of Convolutional Neural Network-based Image Classifications: A Multi-Featured Application for Rice Leaf Disease Prediction and Recommendations for Farmers», 17 de septiembre de 2024, *arXiv: arXiv:2410.01827*. doi: 10.48550/arXiv.2410.01827.
- [8] A. Flores, «INTA presenta nuevas variedades de sorgo rojo de alta calidad industrial», *El 19 Digital*. Accedido: 25 de junio de 2023. [En línea]. Disponible en: <https://bit.ly/3rB7iB5>
- [9] Hernández-Sampieri, Roberto y Mendoza Torres, Christian Paulina, *Metodología de la Investigación: las rutas cuantitativa, cualitativa y mixta*. México: McGRAW-HILL, 2018.
- [10] «Android Studio», Android Developers. Accedido: 26 de junio de 2023. [En línea]. Disponible en: <https://acortar.link/JMuOhb>
- [11] J. Sánchez Allende, *Java 2: iniciación y referencia*. McGraw-Hill España, 2005. Accedido: 7 de julio de 2024. [En línea]. Disponible en: <https://elibro.net/es/ereader/unanmanagua/50290?page=20>
- [12] J. Torres, *Python Deep Learning: Introducción práctica con Keras y TensorFlow 2*. Alpha Editorial, 2020.
- [13] Google, «Teachable Machine». Accedido: 8 de junio de 2023. [En línea]. Disponible en: <https://teachablemachine.withgoogle.com/>
- [14] Google, «TensorFlow Lite», TensorFlow. Accedido: 7 de julio de 2024. [En línea]. Disponible en: <https://www.tensorflow.org/lite/guide?hl=es-419>
- [15] Google Cloud, «Bases de datos de Google Cloud», Google Cloud. Accedido: 10 de junio de 2023. [En línea]. Disponible en: <https://acortar.link/dyoTk5>
- [16] M. Carney *et al.*, «Teachable Machine: Approachable Web-Based Tool for Exploring Machine Learning Classification», en *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, en CHI EA '20. New York, NY, USA: Association for Computing Machinery, abr. 2020, pp. 1-8. doi: 10.1145/3334480.3382839.
- [17] D. Agustian, P. P. G. P. Pertama, P. N. Crisnapati, y P. D. Novayanti, «Implementation of Machine Learning Using Google's Teachable Machine Based on Android», en *2021 3rd International Conference on Cybernetics and Intelligent System (ICORIS)*, oct. 2021, pp. 1-7. doi: 10.1109/ICORIS52787.2021.9649528.
- [18] «Umbrales y matriz de confusión | Machine Learning | Google for Developers». Accedido: 27 de octubre de 2024. [En línea]. Disponible en: <https://acortar.link/UB-PrcS>
- [19] K. S. & J. Sutherland, «La Guía Definitiva de Scrum: Las Reglas del Juego», nov. 2020, Accedido: 25 de junio de 2023. [En línea]. Disponible en: <https://repositorio.uvm.edu.ve/handle/123456789/59>
- [20] J. M. V. Gertrudix, *Java 17 Programación Avanzada*. Rama Editorial, 2022.
- [21] D. S. Menchaca, «Calidad en la especificación de requerimientos de software aplicado en metodologías ágiles», *INF-FCPN-PGI Rev. PGI*, pp. 61-64, nov. 2021, Accedido: 25 de junio de 2023. [En línea]. Disponible en: https://ojs.umsa.bo/ojs/index.php/inf_fcpn_pgi/article/view/49
- [22] R. Herranz, *Despegar con Scrum*. Lulu.com, 2016. [En línea]. Disponible en: <https://acortar.link/ZqhItg>
- [23] E. A. U. Malahina, R. P. Hadjon, y F. Y. Bisilisin, «Teachable Machine: Real-Time Attendance of Students Based on Open Source System», *IJICS Int. J. Inform. Comput.*

Sci., vol. 6, n.º 3, Art. n.º 3, nov. 2022, doi: 10.30865/ijics.v6i3.4928.

- [24] Google, «Glosario sobre aprendizaje automático | Machine Learning | Google for Developers». Accedido: 27 de octubre de 2024. [En línea]. Disponible en: <https://acortar.link/saYFPI>
- [25] «Marco de Trabajo para Seleccionar un Patrón Arquitectónico en el Desarrollo de Software». Accedido: 1 de julio de 2024. [En línea]. Disponible en: <https://dspace.tdea.edu.co/handle/tdea/2670>
- [26] Jiang Huixian, «The Analysis of Plants Image Recognition Based on Deep Learning and Artificial Neural Network | IEEE Journals & Magazine | IEEE Xplore», vol. 8. Accedido: 9 de octubre de 2024. [En línea]. Disponible en: <https://ieeexplore.ieee.org/document/9062591/similar#similar>
- [27] M. Caviedes-Cepeda, F. Carvajal-Larenas, y J. L. Zambrano-Mendoza, «Generación de tecnologías para el cultivo de maíz (Zea mays. L) en el Ecuador», *ACI Av. En Cienc. E Ing.*, vol. 14, n.º 1, Art. n.º 1, abr. 2022, doi: 10.18272/aci.v14i1.2588.
- [28] Tecnológico Nacional, «Manual Granos Básicos opt. pdf». Accedido: 9 de julio de 2024. [En línea]. Disponible en: https://www.tecnacional.edu.ni/media/Manual_Granos_B%C3%A1sicos_opt.pdf

ANEXOS A

Figura 6.

Predicción con imagen que no se usó / Predicción con imagen que se usó en el entrenamiento

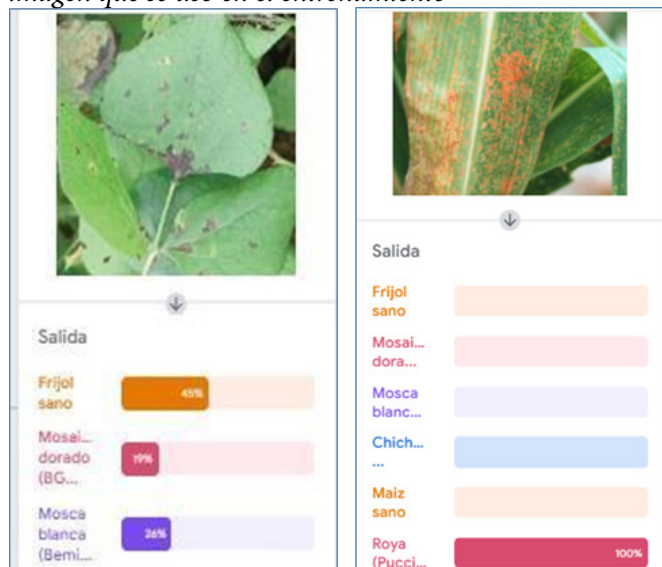


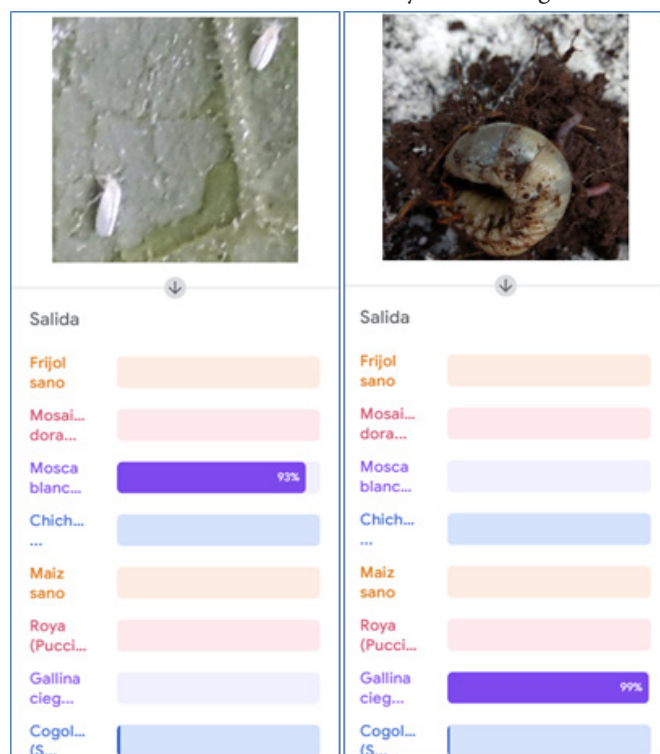
Figura 7.

Predicción errónea de Mosquita del sorgo y Sorgo sano



Figura 8.

Predicción correcta de Mosca blanca y Gallina ciega



ANEXOS B**Tabla 2.**

Afectaciones de plagas y enfermedades en cultivos

Clasificación	Nombres	Afectaciones en cultivos.
Enfermedad	Mosaico dorado (BGYMV)	Las hojas tienen un moteado de tonos amarillos, con venas más blancas de lo normal.
	Roya (Puccinia sorghi)	Pústulas alargadas de color herrumbroso oscuro, pústulas oscuras casi negras, presentes en el haz y envés de la hoja, solo presente en hojas.
	Mancha zonada (Gloeocercospora sorghi)	Lesiones elípticas y largas de color café claro en el centro con márgenes oscuras. Cuando la infestación es severa el área foliar queda con una apariencia quemada.
Plagas	Mosca blanca (Bemisia tabaci)	Hojas se muestran amarillas. Aparece un insecto pequeño de color blanco sobre las hojas del cultivo.
	Chicharrita verde (Empoasca kraemer)	Lesiones en las hojas, se desarrolla el hongo negro cubre la hoja. Insecto diminuto, de color verde en el cultivo.
	Gallina ciega (Phyllophaga sp)	Larvas blancas en las raíces del cultivo, afecta desarrollo de la planta. En etapa adulta es un escarabajo de color oscuro.
	Cogollero (Spodoptera frugiperda)	Raspadura de la epidermis en las hojas, agujeros grandes e irregulares, excrementos como huella. Larvas verde o café / en etapa adulta es polilla de hojas café.
	Mosquita del sorgo (Contarinia sorghicola)	Hojas enrolladas, floración y granos en formación afectados. Larvas amarillas / etapa adulta mosca pequeña con alas transparentes.
	Barrenador mayor del tallo (Diatraea saccharalis)	Tejidos de las plantas, cogollo y vaina de hojas carcomidos. Larvas amarillo claro / etapa adulta larva con alas café claro.

Nota. Información obtenida de Tecnológico Nacional [28]

Tabla 6.

Comparación de métodos utilizados y precisión obtenida de estudios relacionados

Referencia	Tecnologías / Métodos	Datos	Precisión de prueba
[5]	ModCNN	Tomate y Frijol	0.9769
[7]	CNN	Arroz	0.9583
[6]	Tensorflow /CNN	Hortaliza	0.81 - 0.94