Deep Learning Model Based on ResNet9 for Early Blight and Late Blight Disease Detection in Potato

Mario A. Quispe-Layme, Estudiante en ingeniería de sistemas e informática [1], Jeremy J. Ccoyuri-Apaza, Estudiante en ingeniería de sistemas e informática [2]
[1,2]Universidad Tecnológica del Perú, Perú, U20222228@utp.edu.pe, U19207262@utp.edu.pe

Abstract*– The present study addresses the need to detect potato diseases such as early blight, late blight. The objective is to evaluate the performance and accuracy of the Deep Learning technique using in particular the ResNet-9 model convolutional neural network architecture for the identification of diseases on potato leaves. A dataset from the PlantVillage repository on Kaggle was used, which includes 12,009 images of potato leaves classified into three categories: early blight, late blight, and healthy leaves. The model was trained on Google Colab, achieving an accuracy of 99.55% at the final stage. The research highlights the importance of early disease detection to reduce losses in agricultural production, especially in a context where the world population is increasing and the demand for food is growing. Traditional diagnostic techniques, such as visual inspection, are limited and often inaccurate, making the use of Deep Learning a promising solution. This approach not only improves the accuracy of disease identification, but also offers an affordable alternative for low- income farmers who lack the resources for expensive diagnostic techniques. The study concludes that the implementation of Deep Learning algorithms, such as ResNet-9, can be highly effective for disease detection in crops, contributing to more efficient and sustainable agriculture.

Keywords*-- Deep learning, Potato, Plant diseases, Image analysis, Disease detection.

Modelo de Deep Learning basado en ResNet9 para la detección de la enfermedad de Tizón Temprano y Tizón Tardío en la papa

Mario A. Quispe-Layme, Estudiante en ingeniería de sistemas e informática [1], Jeremy J. Ccoyuri-Apaza, Estudiante en ingeniería de sistemas e informática [2]
[1,2]Universidad Tecnológica del Perú, Perú, U20222228@utp.edu.pe, U19207262@utp.edu.pe

Resumen*– El presente estudio abarca la necesidad de detectar las enfermedades de la papa como tizón temprano, tizón tardío. El objetivo es evaluar el desempeño y la precisión de la técnica de Deep Learning utilizando en particular la arquitectura de red neuronal convolucional modelo ResNet-9 para la identificación de enfermedades en las hojas de papa. Se utilizó un conjunto de datos del repositorio PlantVillage en Kaggle, que incluye 12,009 imágenes de hojas de papa clasificadas en tres categorías: tizón temprano, tizón tardío y hojas sanas. El modelo fue entrenado en Google Colab, logrando una precisión del 99.55% en la etapa final. La investigación destaca la importancia de la detección temprana de enfermedades para reducir pérdidas en la producción agrícola, especialmente en un contexto donde la población mundial está en aumento y la demanda de alimentos crece. Las técnicas tradicionales de diagnóstico, como la inspección visual, son limitadas y a menudo imprecisas, lo que hace que el uso de Deep Learning sea una solución prometedora. Este enfoque no solo mejora la precisión en la identificación de enfermedades, sino que también ofrece una alternativa accesible para agricultores de bajos ingresos que carecen de recursos para técnicas de diagnóstico costosas. El estudio concluye que la implementación de algoritmos de Deep Learning, como ResNet-9, puede ser altamente efectiva para la detección de enfermedades en cultivos, contribuyendo a una agricultura más eficiente y sostenible.

***Palabras clave*-- Deep learning, Potato, Plant diseases, Image analysis, Disease detection.**

I. Introducción

La detección temprana de enfermedades en tubérculos, como el tizón temprano y tardío en la papa, es crucial para reducir pérdidas y facilitar el proceso de cultivo y recolección [1]. La papa (Solanum Tuberosum) es uno de los cultivos más populares y vitales a nivel mundial [2]. La papa ocupa el tercer lugar entre los cultivos más producidos y consumidos, siendo consumida por miles de millones de personas en diversas formas [3]. La FAO predice que para el 2050 la población mundial alcanzará los 9.1 mil millones de personas, lo que incrementará significativamente el consumo de alimentos, mientras que las enfermedades agrícolas podrían destruir cosechas y causar hambruna, especialmente en países en desarrollo [4]. Los métodos tradicionales de diagnóstico, como la inspección visual, dependen en gran medida de la experiencia del agricultor, lo que a menudo lleva a resultados imprecisos. Para superar estas limitaciones, se han desarrollado técnicas avanzadas como el espectrómetro, que mejora la precisión en la identificación de hojas sanas e infectadas [5]. La evaluación manual de cultivos es un proceso arduo y poco fiable. En este contexto, el uso de técnicas de Deep Learning para el análisis de enfermedades en cultivos de tubérculos mediante imágenes digitales se presenta como una solución prometedora [6]. El tizón temprano, una enfermedad común en los cultivos de papa causada por Alternaria Solani, afecta principalmente a las hojas más viejas antes de propagarse por toda la planta, acelerando la senescencia. Se caracteriza por la aparición inicial de pequeñas manchas oscuras que crecen y desarrollan anillos concéntricos en condiciones favorables. En Pakistán, su control se ha basado en el uso indiscriminado de fungicidas, sin considerar su distribución geográfica, lo que genera impactos negativos en el medio ambiente y eleva los costos de producción [7], [8]. Las redes neuronales convolucionales, es un tipo de técnica de aprendizaje profundo, se han consolidado como métodos avanzados para el reconocimiento de patrones, especialmente en la identificación de enfermedades en plantas. Estas redes crean representaciones visuales organizadas en diferentes niveles que se adaptan a tareas específicas, como la clasificación de imágenes, logrando modelos precisos y consistentes. Las técnicas de Deep Learning se entrenan utilizando conjuntos de datos específicos, ajustando sus parámetros para alcanzar el objetivo deseado. Una de sus mayores ventajas es su capacidad para analizar nuevos datos y extraer características relevantes con poca intervención humana [9]. Por ejemplo, en Bangladesh, las papas son un producto agrícola significativo con un volumen de exportación considerable, enfrentan desafíos debido a enfermedades destructivas como el tizón tardío, la enfermedad fúngica más común y devastadora que puede reducir los rendimientos de cultivo hasta en un 57% [6]. Las enfermedades de las plantas son responsables de alrededor del 25% de las pérdidas de cultivos, lo que equivale a una cantidad de alimentos que podría alimentar a aproximadamente 600 millones de personas. Entre estas, las enfermedades que afectan las hojas juegan un rol crucial en la reducción de la productividad agrícola, particularmente en cultivos importantes como la papa, que es esencial tanto a nivel local como global para el consumo y la exportación [3].

El objetivo de esta investigación fue evaluar el desempeño y precisión de la técnica de Deep Learning con el modelo ResNet-9 para la identificación de la enfermedad del tizón temprano y tizón tardío en la papa adquiriendo un dataset para la clasificación de enfermedades.

II. Estudio de Literatura

Las enfermedades que afectan a la papa, especialmente el tizón temprano y tardío, generan un impacto considerable tanto en la calidad como en la cantidad de la producción global de este cultivo [4]. Según un estudio de la FAO, estas enfermedades pueden reducir la producción anual entre un 9% y un 11%, lo que representa uno de los principales obstáculos para el crecimiento de este cultivo [10].

En el campo de la detección de enfermedades en las hojas de los cultivos, se han desarrollado diversos enfoques basados en imágenes, entre los que destacan las técnicas de visión por computadora. En los últimos años, el uso de redes neuronales convolucionales (CNN) ha sido crucial para avanzar en esta área. Modelos como LeNet, AlexNet, VGGNet, GoogLeNet, ResNet y DenseNet han demostrado ser altamente efectivos en el reconocimiento y clasificación de imágenes de hojas afectadas por enfermedades [4].

El tizón temprano, causado por Alternaria Solani, se manifiesta principalmente en las hojas de papa y tomate mediante manchas oscuras con anillos concéntricos. Para su control, se utilizan fungicidas como los inhibidores de quinona (QoI) y succinato deshidrogenasa (SDHI), considerados los más efectivos, además de la desmetilación (DMI). Junto con el uso de fungicidas, la aplicación de biofumigantes, semillas libres de patógenos y prácticas que eviten el estrés en las plantas son medidas preventivas efectivas, ya que las plantas estresadas son más susceptibles a esta enfermedad [1].

Por otro lado, Phytophthora Infestans, el agente causal del tizón tardío, provoca manchas marrones en hojas húmedas. En condiciones de alta humedad y temperaturas moderadas, las lesiones pueden desarrollar un crecimiento mohoso gris o blanco, lo que acelera la propagación de la enfermedad. En su etapa avanzada, las hojas amarillean, se vuelven marrones, se marchitan y mueren. Este patógeno fue responsable de la hambruna irlandesa del siglo XIX y sigue siendo una amenaza para los cultivos de papa. El uso de productos a base de neem ha mostrado ser efectivo para controlar el tizón tardío, además de la aplicación de fungicidas protectores y penetrantes, que si se utilizan a tiempo, pueden frenar la enfermedad [1], [11].

Debido a que gran parte de la agricultura es realizada por personas de bajos ingresos, los agricultores a menudo no tienen acceso a técnicas costosas para la detección de enfermedades en los cultivos [4]. Sin embargo, el uso de técnicas de Deep Learning (DL), que imitan el proceso de reconocimiento de objetos del cerebro humano, ha revolucionado la evaluación multiespectral en la investigación agrícola. Redes neuronales convolucionales como GoogLeNet, DenseNet, VGG y ResidualNet han demostrado ser efectivas en la medición de granos, la identificación de plantas, la cuantificación de frutos y el diagnóstico de enfermedades en cultivos, ofreciendo resultados confiables y accesibles para los agricultores [4].

III. METODOLOGÍA

En este estudio, se implementó el algoritmo ResNet9 para la detección de enfermedades en hojas de papa, específicamente tizón temprano, tizón tardío y hojas sanas, a través de imágenes. El proceso involucra varias etapas clave, desde el preprocesamiento de las imágenes hasta la evaluación del modelo y el análisis de casos problemáticos. Las imágenes fueron preprocesadas para eliminar ruido y mejorar la precisión del modelo. Posteriormente, se entrenó el modelo utilizando un conjunto de datos etiquetado, y se probó su capacidad predictiva sobre nuevas imágenes. A continuación, se detalla el flujo de trabajo utilizado:

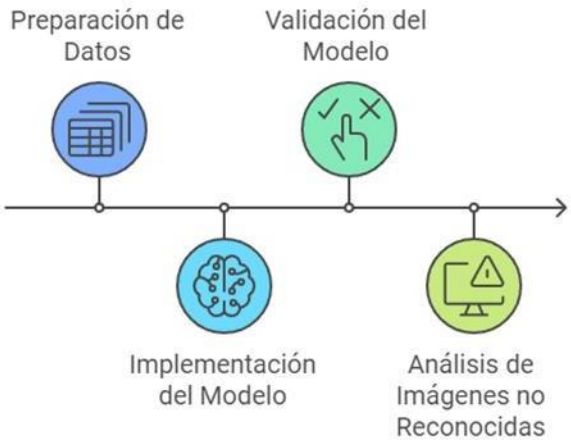

Fig. 1 Flujo de Trabajo de Entrenamiento y Evaluación del Modelo.

A. *Preparación de datos*

Para el entrenamiento del modelo, se empleó un subconjunto específico del conjunto de datos PlantVillage, disponible en Kaggle, y publicado por Alberta Odamea Anim-Ayeko [12] Este conjunto incluye imágenes clasificadas en tres categorías principales:

- Early Blight: 3,996 imágenes
- Healthy: 4,011 imágenes
- Late Blight: 4,002 imágenes

El modelo se entrenó utilizando la arquitectura de red neuronal convolucional ResNet-9. El proceso de entrenamiento se llevó a cabo en Google Colab, implementando un cuaderno interactivo basado en el repositorio de GitHub de Alyeko [13]. El archivo correspondiente, 1-Potato&Tomato-blight-disease-detection.ipynb, fue adaptado para facilitar el entrenamiento del modelo.

Una vez completado el proceso, el modelo entrenado fue almacenado en Google Drive bajo el nombre resnet9- mdlsd.pth para su posterior utilización.

El entrenamiento se realizó en un entorno Google Colab con las siguientes especificaciones técnicas:

- Unidad de Procesamiento Gráfico (GPU): backend de Google Compute Engine

- Versión de Python: 3
- Memoria RAM: 12.67 GB (uso máximo: 1.53 GB)
- Almacenamiento: 112.64 GB (uso máximo: 32.59 GB)

Se utilizaron diversas bibliotecas esenciales para la implementación y análisis del modelo, como torch y torchvision para el manejo de datos y redes neuronales, shap para interpretabilidad del modelo, y optuna para la optimización de hiperparámetros. Además, se emplearon herramientas de visualización como matplotlib y seaborn para generar gráficos y analizar la distribución de datos.

El conjunto de datos está organizado en dos carpetas principales: train y valid. La carpeta train contiene las imágenes destinadas al entrenamiento del modelo, mientras que la carpeta valid alberga las imágenes para validar su desempeño. Para confirmar la organización y el contenido de este directorio, se utilizó el comando os.listdir(), que permite listar los archivos y carpetas presentes. Este análisis inicial aseguró que los datos estuvieran estructurados de manera adecuada para su uso en el proceso de modelado.

B. Implementación del Modelo ResNet-9

Esta red cuenta con cuatro bloques convolucionales, dos bloques residuales (cada uno formado por dos capas residuales) y un clasificador, sumando un total de nueve capas.

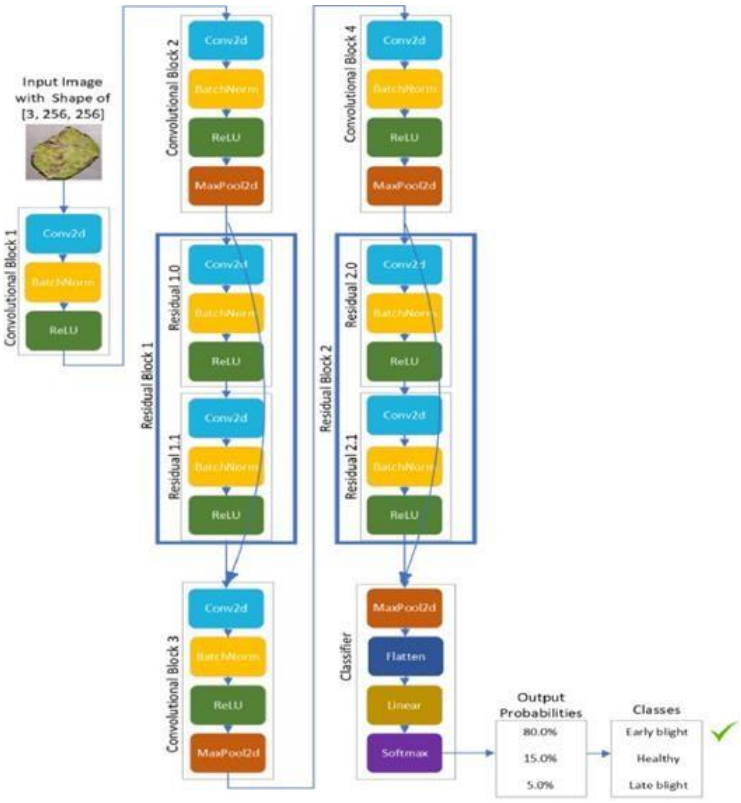

Fig. 2 Capas del modelo Rest9 [1].

La entrada consiste en una imagen de una hoja de papa de tamaño [3, 256, 256], con 3 canales en formato RGB.

1. Bloque Convolucional 1

Este bloque inicial realiza la extracción de características básicas de la imagen de entrada.

- Conv2d: Aplica una convolución 2D, utilizando filtros para extraer características fundamentales de la imagen como bordes y texturas. En esta implementación, los filtros se optimizan mediante capas Depthwise Convolutional Eigen-Filter (DeCEF), una técnica de rango bajo que reduce significativamente el número de parámetros y operaciones necesarias [14].
- BatchNorm: Normaliza las activaciones del lote, acelerando el entrenamiento y reduciendo la sensibilidad a la inicialización de pesos.
- ReLU: Introduce no linealidad en las activaciones, asegurando que el modelo pueda aprender representaciones complejas.

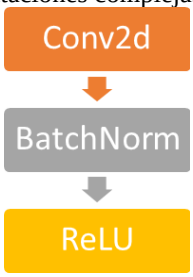

Fig. 3 Bloque Convolucional 1.

2. Bloque Convolucional 2

Refina las características extraídas en el primer bloque mientras reduce la resolución espacial.

- Conv2d: Realiza una segunda convolución para detectar patrones más avanzados combinando características simples.
- BatchNorm: Mejora el entrenamiento de redes neuronales profundas, como las CNN, mediante un paso de normalización junto con parámetros de cambio y escala entrenables. Aunque el mecanismo exacto de mejora no es completamente claro, se ha demostrado empíricamente que BatchNorm aumenta el rendimiento, la estabilidad y la precisión del modelo, posiblemente debido a su impacto en la inicialización de pesos [15].
- ReLU: Se aplica a los nodos en las redes neuronales, excepto en la capa de salida. Este tipo de activación ha demostrado superar a funciones como la sigmoide en redes profundas, logrando resultados sobresalientes en problemas de referencia en diversos campos [16].

- MaxPool2d: Reduce la dimensionalidad espacial, disminuyendo el tamaño de los mapas de características y enfocándose en las áreas más importantes.

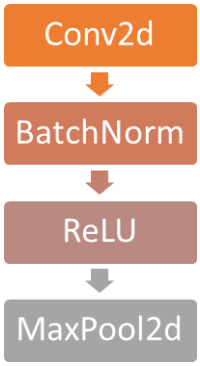

Fig. 4 Bloque Convolucional 2.

3. Bloque Residual 1.0

El primer bloque residual introduce conexiones residuales para mitigar problemas de gradientes desaparecidos.

- Conv2d: Extrae características utilizando filtros más profundos.
- BatchNorm: Ees una técnica ampliamente empleada en las CNN modernas, conocida por su capacidad para actuar como regularizador, acelerar la convergencia y facilitar el entrenamiento de redes más profundas [17].
- ReLU: Aumenta la capacidad de aprendizaje no lineal. Conexión Residual: La salida de este bloque se suma

a la entrada original para mantener información previa y aprender nuevas características en capas más profundas.

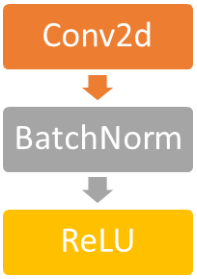

Fig. 5 Bloque Residual 1.0.

4. Bloque Residual 1.1

Complementa al Bloque Residual 1.0 refinando aún más las características aprendidas.

- Conv2d: Aplica otra convolución para identificar patrones más complejos.
- BatchNorm: Ajusta las activaciones para estabilizar el entrenamiento.
- ReLU: Es una función de activación no lineal que activa las neuronas solo cuando su entrada es mayor que cero, permitiendo que el modelo aprenda representaciones complejas y no lineales. Se ha demostrado que mejora el rendimiento de redes neuronales profundas en diversas tareas de aprendizaje automático [18].
- Conexión Residual: Similar al Bloque Residual 1.0, permite la suma de características originales y transformadas.

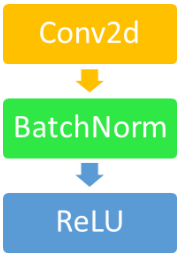

Fig. 6 Bloque Residual 1.1.

5. Bloque Convolucional 3

Avanza hacia características más abstractas de nivel intermedio.

- Conv2d: Usa filtros más especializados para extraer patrones jerárquicos.
- BatchNorm: Normaliza activaciones para mejorar la generalización.
- ReLU: Aumenta las propiedades no lineales del modelo.
- MaxPool2d: Reduce la dimensionalidad espacial para minimizar el costo computacional.

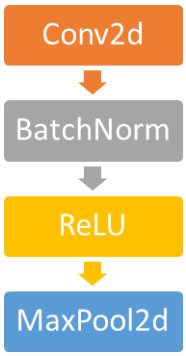

Fig. 7 Bloque Convolucional 3.

6. Bloque Convolucional 4

Consolida las características avanzadas que servirán como entrada para el siguiente bloque.

- Conv2d: Detecta patrones específicos de alto nivel como texturas complejas.

- BatchNorm: Garantiza una propagación estable de activaciones.
- ReLU: Activa las neuronas solo cuando la entrada supera el valor cero. Este enfoque permite que el modelo capture patrones no lineales de manera eficiente, lo que mejora su capacidad para aprender representaciones complejas en diversas aplicaciones [18].
- MaxPool2d: Reduce el tamaño de los mapas de características para enfocarse en información clave.

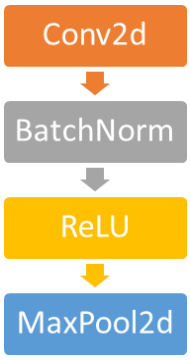

Fig. 8 Bloque Convolucional 4.

7. Bloque Residual 2.0

Introduce conexiones residuales adicionales para aprender características avanzadas manteniendo la información original.

- Conv2d: Aplica filtros especializados en niveles superiores.
- BatchNorm: Suaviza las salidas para evitar oscilaciones en las activaciones.
- ReLU: Activa las características útiles para las capas posteriores.
- Conexión Residual: Garantiza que el modelo conserve la información útil de las entradas previas.

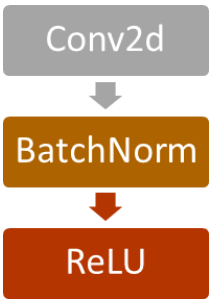

Fig. 9 Bloque Convolucional 2.0.

8. Bloque Residual 2.1

Refuerza el aprendizaje de características complejas a partir del bloque anterior.

- Conv2d: Detecta patrones específicos para la clasificación.
- BatchNorm: Ajusta las activaciones para lograr mayor estabilidad en la optimización.
- ReLU: Permite modelar relaciones no lineales.
- Conexión Residual: Asegura la retención de información importante mientras se aprenden nuevas características.

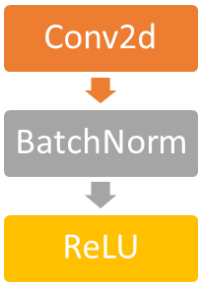

Fig. 10 Bloque Convolucional 2.1.

9. Clasificador

La etapa final del modelo, encargada de realizar la predicción de clases.

- MaxPool2d: Resume las características aprendidas de manera compacta.
- Flatten: Convierte las características espaciales 2D en un vector unidimensional.
- Linear: Mapea las características a las probabilidades de cada clase.
- Softmax: Es una función de activación comúnmente empleada en redes neuronales profundas, que activa las neuronas solo cuando la entrada supera el valor cero. Este enfoque permite que el modelo capture patrones no lineales de manera eficiente, lo que mejora su capacidad para aprender representaciones complejas en diversas aplicaciones [19].

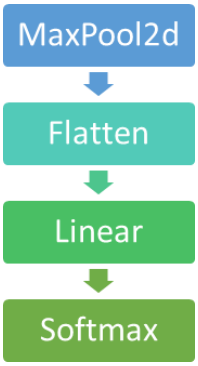

Fig. 11 Clasificador

Para las salidas del modelo, el clasificador genera probabilidades asociadas a las tres categorías:

- Early Blight (Tizón Temprano)
- Healthy (Saludable)

- Late Blight (Tizón Tardío)

C. Evaluación de modelos

Se emplean las métricas de exactitud, precisión, exhaustividad y puntuación F1 en el conjunto de prueba para evaluar los modelos ResNet-9 y VGG-16. Asimismo, se utilizan las curvas de pérdida para analizar el proceso de entrenamiento del modelo.

$$\text{Accuracy} = 2 \text{ x } \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (1)$$

$$\text{Precision} = 2 \text{ x } \frac{\text{TP}}{\text{TP} + \text{FP}} \quad (2)$$

$$\text{Recall} = 2 \text{ x } \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (3)$$

$$\text{F1} - \text{Score} = 2 \text{ x } \frac{\text{Precisión} \times \text{Recall}}{\text{Precisión} + \text{Recall}} \quad (4)$$

Las métricas de evaluación consideran una clase a la vez:

- Verdadero Positivo (TP): ocurre cuando el modelo predice correctamente que pertenece a la clase considerada.
- Verdadero Negativo (TN): ocurre cuando el modelo predice correctamente que no pertenece a la clase considerada.
- Falso Positivo (FP): ocurre cuando el modelo predice incorrectamente que pertenece a la clase considerada.
- Falso Negativo (FN): ocurre cuando el modelo predice incorrectamente que no pertenece a la clase considerada.

Para las métricas de Precisión, Exhaustividad y Puntuación F1, los puntajes finales se calculan promediando los resultados obtenidos en todas las clases.

D. Validación del Modelo

Para la validación del modelo entrenado, se utilizó otro cuaderno de Google Colab, 2-using-the-already-trained-resnet9-model.ipynb [13], que permitió importar el modelo entrenado y probarlo con datos nuevos.

Como parte del proceso de validación, se emplearon tres carpetas del conjunto de datos de Kaggle proporcionado por Samir Bhattarai [20], que contienen imágenes de hojas de papa clasificadas en:

- Early_blight (485 imágenes)
- Late_blight (485 imágenes)
- Healthy (201 imágenes)

El modelo entrenado fue sometido a pruebas con estas imágenes para evaluar su precisión en la detección de tizón temprano y tizón tardío en las hojas de papa.

El tiempo total de ejecución del entrenamiento fue de aproximadamente 1 hora y 18 minutos, utilizando tanto CPU como GPU para acelerar el proceso.

E. Análisis de Imágenes no Reconocidas

Durante el análisis de los resultados, se identificaron diversas imágenes problemáticas que impactaron negativamente en el desempeño del modelo. Algunas imágenes mostraban un alto nivel de brillo (Fig. 13), lo que dificultaba la identificación de las características esenciales de las hojas. Otras imágenes no estaban bien encuadradas (Fig. 12), lo que complicaba la clasificación correcta.

Fig. 12 Flujo de Trabajo de Entrenamiento

Fig. 13 Evaluación del Modelo.

La limpieza de datos resultante permitió eliminar estos ejemplos problemáticos, lo que se tradujo en una mejora significativa del rendimiento del modelo. Este análisis destaca la importancia de la calidad de los datos en proyectos de clasificación de imágenes, ya que las imágenes mal representadas pueden inducir al modelo a realizar predicciones erróneas.

IV. Resultados

Una evaluación exhaustiva del rendimiento del modelo requiere medir su desempeño no solo en términos de precisión, sino también en métricas complementarias como la pérdida y la capacidad del modelo para generalizar con datos no vistos. Las predicciones y el cálculo de métricas se realizaron en un conjunto de prueba independiente que contenía imágenes organizadas de la misma manera que el conjunto de entrenamiento. Las métricas clave que se calcularon incluyen la precisión global, la matriz de confusión y el análisis de los errores cometidos por el modelo.

A. Entrenamiento del Modelo

Este modelo fue ajustado para las tareas específicas del proyecto. Se emplearon 22 épocas de entrenamiento, con un enfoque de ciclo de aprendizaje mediante la técnica fit OneCycle, que ajusta dinámicamente la tasa de aprendizaje durante el entrenamiento. Esto permite que el modelo aprenda de manera eficiente, mejorando su rendimiento sin caer en el sobreajuste.

En la tabla I, se observa una tendencia general de disminución en las pérdidas de entrenamiento y validación del modelo ResNet-9 entre las épocas 0 y 21. Al inicio en la época 0, el modelo muestra un ajuste insuficiente con pérdidas de entrenamiento superiores a la validación. Entre las épocas 1 y 12 las pérdidas de validación fluctúan, aumentando y disminuyendo mientras que las pérdidas de entrenamiento siguen una tendencia descendente. A partir de la época 13 hasta la 21 la perdida de entrenamiento y validación se mantienen bajas.

TABLA I
ÉPOCA TASA DE APRENDIZAJE PÉRDIDA DE ENTRENAMIENTO
PRECISIÓN DE ENTRENAMIENTO

Época	Tasa de Aprendizaje	Pérdida de Entrenamiento	Precisión de Entrenamiento
0	0.00102	0.5758	0.9914
1	0.00265	0.5670	0.9919
2	0.00496	0.5668	0.9909
3	0.00744	0.5648	0.9909
4	0.00954	0.5606	0.9935
5	0.01079	0.5571	0.9965
6	0.01098	0.5551	0.9976
7	0.01078	0.5540	0.9981
8	0.01035	0.5557	0.9972
9	0.00973	0.5538	0.9986
10	0.00893	0.5524	0.9994
11	0.00799	0.5520	0.9996
12	0.00694	0.5520	0.9996
13	0.00584	0.5520	0.9996
14	0.00472	0.5520	0.9996
15	0.00363	0.5519	0.9996
16	0.00262	0.5519	0.9996
17	0.00173	0.5519	0.9996
18	0.00100	0.5519	0.9996
19	0.00045	0.5519	0.9996
20	0.00011	0.5519	0.9996
21	0.00000	0.5520	0.9994

B. Evaluar Resultados del Entrenamiento

El proceso de evaluación del entrenamiento se realizó utilizando las métricas y precisión tanto para el conjunto de entrenamiento como de validación. La Tabla I muestra los resultados por cada época, evidenciando una mejora constante en la pérdida de entrenamiento, que comenzó en 0.5758 y disminuyó a 0.5519 en la última época. La precisión de entrenamiento mostró un desempeño elevado.

En cuanto a la validación, la precisión inicial fue de 94.20%, la cual aumentó progresivamente hasta llegar a un máximo de 99.78% en la época 20. Sin embargo, se observaron fluctuaciones en la pérdida de validación, lo que sugiere posibles indicios de sobreajuste en etapas intermedias, pero que fueron mitigados por las técnicas de regularización aplicadas, como el decaimiento de peso.

C. Predicciones

Después de entrenar el modelo, se procedió a evaluar su desempeño utilizando el conjunto de datos de prueba. Se recorrieron las imágenes del conjunto y se calcularon las etiquetas estimadas por el modelo. Estas salidas se compararon con las etiquetas reales para medir su rendimiento. Los resultados se representaron en una matriz de confusión, lo que permitió un análisis detallado de los errores y la precisión por clase.

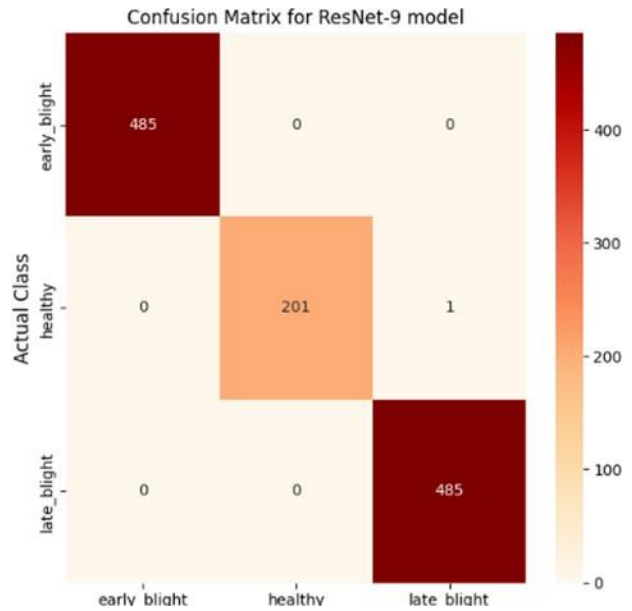

Fig. 14 Flujo de Trabajo de Entrenamiento y Evaluación del Modelo.

El proceso de predicción demostró que el modelo era capaz de identificar con alta precisión las categorías de las hojas de papa, aunque se identificaron ciertos problemas relacionados con la calidad de algunas imágenes.

D. Evaluar Rendimiento del Modelo

La evaluación del modelo mostró resultados prometedores en cuanto a precisión y robustez. En una primera evaluación, utilizando 1,426 imágenes, el modelo logró una precisión de prueba del 94.11%. No obstante, se identificaron problemas en la calidad de las imágenes, como brillo excesivo y encuadres incorrectos, lo que afectó negativamente el rendimiento.

Tras realizar una limpieza del conjunto de datos, el modelo fue reevaluado con 1,172 imágenes de mejor calidad, y la precisión de prueba aumentó a 99.55%, subrayando la importancia de contar con datos de alta calidad. Este análisis destacó la relación directa entre la calidad de los datos y el rendimiento del modelo, siendo crucial contar con imágenes representativas y libres de ruido para garantizar predicciones fiables.

E. Precisión del Modelo en el Conjunto de Prueba

En una primera evaluación con 1426 imágenes del conjunto de prueba, el modelo logró una precisión del 94.11%. Esta cifra, aunque prometedora, se vio afectada por la presencia de imágenes problemáticas en el conjunto de datos.

Después de una limpieza del conjunto de prueba, eliminando imágenes con brillo excesivo o encuadres incorrectos, la precisión mejoró significativamente, alcanzando un 99.55%. Esta mejora subraya la importancia de utilizar datos de alta calidad para obtener resultados óptimos.

Como resultado de este proceso de depuración, ahora el conjunto de datos contiene 1,172 imágenes.

F. Análisis del Desempeño

El modelo fue particularmente efectivo en la clasificación de imágenes de las clases healthy y late_blight. Sin embargo, algunas imágenes de early_blight presentaron errores en la clasificación, probablemente debido a características visuales ambiguas que las hacían parecerse a otras categorías.

La matriz de confusión reveló que la mayoría de los errores ocurrieron en imágenes mal encuadradas o sobreexpuestas, lo que llevó al modelo a confundirse entre las clases early_blight y late_blight.

G. Calidad de las Imágenes

La eliminación de imágenes con problemas de calidad, como las que presentaban un alto nivel de brillo o estaban mal centradas, permitió una mejora notable en el rendimiento del modelo. Este hallazgo refuerza la necesidad de realizar un control de calidad riguroso en los conjuntos de datos de entrenamiento y prueba.

H. Curva de Aprendizaje

La curva de aprendizaje del modelo, observada a través de la disminución en la pérdida de entrenamiento y el aumento de la precisión de validación, mostró una tendencia positiva a medida que se avanzaba en las épocas. Esta curva confirma que el modelo estaba aprendiendo de manera eficiente y que las técnicas de regularización fueron efectivas para evitar el sobreajuste.

V. Discusión y Conclusiones

En la Tabla II, se presenta un análisis comparativo entre el trabajo propuesto y otros enfoques de aprendizaje profundo, con el objetivo de destacar la investigación en relación con ellos. Para ello, se seleccionaron diversos marcos de aprendizaje profundo conocidos como ResNet-9 [1], PLDPNet [3] EfficientRMT-Net [4], CNN [6]. Se compararon las arquitecturas de Deep Learning desde la precisión y el modelo. Los resultados se muestran en la tabla 2. En precisión del modelo, el resultado más bajo en rendimientos lo tiene CNN, con 96.09%. El segundo rendimiento más bajo lo tiene EfficientRMT-net, con un 97.65%. En semejanza, la investigación esta con una puntuación de precesión del 99.55%, visiblemente la comparación de estos enfoques muestra un promedio del 97.99%, y el modelo ResNet-9 tiene 99.55%. Este destacado rendimiento se puede deber al uso de su arquitectura optimizada que utiliza bloques residuales para el entrenamiento más efectivo, el aumento de datos y calidad de imagen.

TABLA II
COMPARACIÓN DE LOS MODELOS HALLADOS

Nro.	Modelo	Precisión
Propuesto	ResNet-9	99.55%
[1]	ResNet-9	99.25%
[3]	PLDPNet	98.66%
[4]	EfficientRMT-Net	97.65%
[6]	CNN	96.09%

VI. CONCLUSIONES

El modelo ResNet-9 se presenta como una solución efectiva para identificar y clasificar enfermedades en las hojas en cultivos de papa, incluyendo el tizón temprano y tardío. Basado en la arquitectura el uso bloques residuales y por etapas, se destaca por su alta clasificación, precisión y rapidez en la detección. Los experimentos realizados evidencian su superioridad, alcanzando tasas de precisión del 94,11% en el conjunto de prueba inicial y 99,55% en la etapa final con el conjunto de datos, superando a otros enfoques existentes.

Esta investigación no tomó en cuenta factores externos como la temperatura del ambiente, las cantidades de nutrientes y minerales del suelo debido a que no teníamos el control del ambiente en el data set obtenido. No obstante, se centró en y evaluar un modelo de aprendizaje profundo capaz de clasificar nuevas imágenes de plantas, empleando fotografías de hojas de papas disponibles en dataset.

La detección anticipada de las enfermedades en las plantas es esencial para proteger los cultivos y contribuir a una mejor calidad de cosecha y disminuyendo los costes de producción y aumentando la calidad de la papa. Este estudio propone un enfoque basado en Deep Learning para predecir enfermedades en las hojas de papa, dentro de un esquema de clasificación múltiple: sana, tizón temprano y tizón tardío.

Instaurar este modelo de aprendizaje en dispositivos más portátiles como smartphones o tablets permitiría a los agricultores y usuarios tener un mayor acceso de uso. La creación del modelo con una interfaz gráfica amigable, donde el usuario pueda tomar la fotografía del cultivo enfermo seria de utilidad.

REFERENCIAS

[1] A. O. Anim-Ayeko, C. Schillaci, and A. Lipani, "Automatic blight disease detection in potato (Solanum tuberosum L.) and tomato (Solanum lycopersicum, L. 1753) plants using deep learning," Smart Agricultural Technology, vol. 4, Aug. 2023, doi: 10.1016/j.atech.2023.100178.

[2] Deepkiran, M. Pandey, and L. Singh, "Spectral Segmentation Augmented with Normalized Cuts for Detection of Early Blight Disease in Potato," in Procedia Computer Science, Elsevier B.V., 2024, pp. 1034–1043. doi: 10.1016/j.procs.2024.03.292.

[3] F. Arshad et al., "PLDPNet: End-to-end hybrid deep learning framework for potato leaf disease prediction," Alexandria Engineering Journal, vol. 78, pp. 406–418, Sep. 2023, doi: 10.1016/j.aej.2023.07.076.

[4] K. Shaheed et al., "EfficientRMT-Net—An Efficient ResNet-50 and Vision Transformers Approach for Classifying Potato Plant Leaf Diseases," Sensors, vol. 23, no. 23, Dec. 2023, doi: 10.3390/s23239516.

[5] J. Rashid, I. Khan, G. Ali, S. H. Almotiri, M. A. Alghamdi, and K. Masood, "Multi-level deep learning model for potato leaf disease recognition," Electronics (Switzerland), vol. 10, no. 17, Sep. 2021, doi: 10.3390/electronics10172064.

[6] J. Akther, A. A. Nayan, and M. Harun-Or-roshid, "Potato Leaves Blight Disease Recognition and Categorization Using Deep Learning," Engineering Journal, vol. 27, no. 9, pp. 27–38, Sep. 2023, doi: 10.4186/ej.2023.27.9.27.

[7] A. Abbas, U. Maqsood, S. Ur Rehman, K. Mahmood, T. Alsaedi, and M. Kundi, "An Artificial Intelligence Framework for Disease Detection in Potato Plants," Engineering, Technology and Applied Science Research, vol. 14, no. 1, pp. 12628–12635, 2024, doi: 10.48084/etasr.6456.

[8] T. Nazir, M. M. Iqbal, S. Jabbar, A. Hussain, and M. Albathan, "EfficientPNet—An Optimized and Efficient Deep Learning Approach for Classifying Disease of Potato Plant Leaves," Agriculture (Switzerland), vol. 13, no. 4, Apr. 2023, doi: 10.3390/agriculture13040841.

[9] F. Leiva, F. Abdelghafour, M. Alsheikh, N. E. Nagy, J. Davik, and A. Chawade, "ScabyNet, a user-friendly application for detecting common scab in potato tubers using deep learning and morphological traits," Sci Rep, vol. 14, no. 1, Dec. 2024, doi: 10.1038/s41598-023-51074-4.

[10] K. Shaheed et al., "EfficientRMT-Net—An Efficient ResNet-50 and Vision Transformers Approach for Classifying Potato Plant Leaf Diseases," Sensors, vol. 23, no. 23, Dec. 2023, doi: 10.3390/s23239516.

[11] A. O. Anim-Ayeko, C. Schillaci, and A. Lipani, "Automatic blight disease detection in potato (Solanum tuberosum L.) and tomato (Solanum lycopersicum, L. 1753) plants using deep learning," Smart Agricultural Technology, vol. 4, Aug. 2023, doi: 10.1016/j.atech.2023.100178.

[12] "pt-leaf-data." Accessed: Dec. 03, 2024. [Online]. Available: https://www.kaggle.com/datasets/alyeko/pt-leaf-data

[13] "GitHub - Alyeko/potato-tomato-blight-disease-detection: Detection of blight disease in potato and tomato plant leaf images." Accessed: Dec. 03, 2024. [Online]. Available: https://github.com/Alyeko/potato-tomato-blight-disease-detection

[14] D. Oppenheim, G. Shani, O. Erlich, and L. Tsror, "Using deep learning for image-based potato tuber disease detection," Phytopathology, vol. 109, no. 6, pp. 1083–1087, 2019, doi: 10.1094/PHYTO-08-18-0288-R.

[15] Y. Peerthum and M. Stamp, "An empirical analysis of the shift and scale parameters in BatchNorm," Inf Sci (N Y), vol. 637, Aug. 2023, doi: 10.1016/j.ins.2023.118951.

[16] B. T. Agyeman, J. Liu, and S. L. Shah, "ReLU surrogates in mixed- integer MPC for irrigation scheduling," Chemical Engineering Research and Design, vol. 211, pp. 285–298, Nov. 2024, doi: 10.1016/j.cherd.2024.10.005.

[17] D. Rivoir, I. Funke, and S. Speidel, "On the pitfalls of Batch Normalization for end-to-end video learning: A study on surgical workflow analysis," Med Image Anal, vol. 94, May 2024, doi: 10.1016/j.media.2024.103126.

[18] P. Hu, L. Sun, C. Hu, L. Dai, S. Guo, and M. Yu, "DReP: Deep ReLU pruning for fast private inference," Journal of Systems Architecture, vol. 152, p. 103156, Jul. 2024, doi: 10.1016/J.SYSARC.2024.103156.

[19] X. Ye, J. Zhao, J. Qian, and Y. Li, "SoftmaxU: Open softmax to be aware of unknowns," Eng Appl Artif Intell, vol. 133, p. 108594, Jul. 2024, doi: 10.1016/J.ENGAPPAI.2024.108594.

[20] "New Plant Diseases Dataset." Accessed: Dec. 03, 2024. [Online]. Available: https://www.kaggle.com/datasets/vipoooool/new-plant-diseases-dataset/data