

Mobile system for the identification of pest in the lucuma leaf based on artificial intelligence in Villa El Salvador

Huarote Zegarra Raúl Eduardo, Maestro¹ 

¹*Universidad Nacional Tecnológica de Lima Sur, Perú, rhuarote@untels.edu.pe*

Abstract— The present research focuses on applying artificial vision and the artificial neural network, for the identification of pests in lucuma leaves (SP, PG, ER) in the field of the district of Villa El Salvador through a mobile system with 4500 images obtained in the field, managing to do it from a mobile application being easy to use, as well as the results will be reliable, since the results are statistically demonstrated (confusion matrix), where 99.32% of accuracy, an f1-score of 99%, 100% and 0.99% for SP, ER and PG disease respectively. Thus, it was also possible to achieve an effectiveness rate of 99.32%, implying that the adjustments in the configurations necessary for the SOM neural network model are ideal and the algorithms based on computer vision used (sobel, medfield and orientation) are ideal, therefore the first specific objective is met, in the same way it was possible to reduce the pest per crop area by 1.5% (visually) due to the care taken in identifying the pest on the leaf, therefore, it can be said that Considering the limitations of nature, it meets the second specific objective. Considering the level of simplicity of using the Smartphone, where 33% are extremely satisfied, 45% are very satisfied, 22% are somewhat satisfied and 0% are not so satisfied with not at all satisfied.

Keywords— Smartphone, artificial vision, SOM neural network, lucuma leaf, leaf pests.

Sistema móvil para la identificación de plaga en la hoja de lúcuma basado en inteligencia artificial en el sembrío del distrito de Villa el Salvador

Huarote Zegarra Raúl Eduardo, Maestro¹ 

¹Universidad Nacional Tecnológica de Lima Sur, Perú, ruarote@untels.edu.pe

Resumen— La presente investigación se centra en aplicar visión artificial y la red neuronal artificial, para la identificación de plagas en las hojas de lúcuma (SP, PG, ER) en el sembrío del distrito de villa el salvado por medio de un sistema móvil con 4500 imágenes obtenidas en campo, logrando realizarlo a partir de un aplicativo móvil siendo sencillo de usar, así como los resultados serán confiables, ya que queda demostrado estadísticamente los resultados (matriz de confusión), donde se obtiene un 99.32% de accuracy, un f1-score de 99%, 100% y 0.99% para la enfermedad SP, ER y PG respectivamente. Así también se logró alcanzar una tasa de efectividad de 99.32% implicando que los ajustes en las configuraciones necesarias para el modelo de la red neuronal SOM son las ideales y los algoritmos basados en visión computacional usados (sobel, medfield y orientación) son los idóneos, por tanto se cumple el primer objetivo específico, de la misma manera se logró disminuir la plaga por área de sembrío en un 1.5% (de manera visual) por el cuidado tenido al identificar la plaga en la hoja, por tanto, se puede decir que considerando las limitaciones propias de la naturaleza cumple el segundo objetivo específico. Dando cuenta el nivel de sencillez del uso del Smartphone, donde el 33% está extremadamente satisfecho, 45% está muy satisfecho, 22% algo de satisfacción y 0% para no tan satisfecho con nada satisfecho.

Palabras clave— Smartphone, visión artificial, SOM red neuronal, hoja de lúcuma, plagas de hoja.

I. INTRODUCCIÓN

Al igual que en otros países, en Perú también existen problemas de plagas y enfermedades que disminuyen la producción hasta en un 80%, tal es el caso de la plaga del gorgojo, enfermedad de la ranca, entre otras. Siendo Para DGDA en [1] La ranca y la bacteria son las enfermedades que más se presentan en los cultivos, en muchos casos se llega a perder el 70% de la producción por descuido o desconocimiento del agricultor.

Sin embargo, los cultivos de esta zona se ven afectados continuamente por plagas y enfermedades que provocan grandes problemas en la producción, ocasionándoles gastos importantes para la fumigación, por otro lado, en cuanto a la mano de obra se ven en la necesidad de contratar personas que se encarguen de la fumigación del cultivo.

La enfermedad que más ataca sus cultivos es la ranca, que puede llegar a presentarse hasta en un 1% en el cultivo de lúcuma, sin embargo, se llega a proliferar en cuestión de días, debido a que se expande rápidamente. Esta enfermedad ocasiona que las hojas del cultivo tomen un color negro,

llegando a tener un aspecto a quemado, lo cual ocasiona que la planta se debilite por ende disminuya su producción.

Así también para DICTA en [2], la plaga de la mosca minadora. Esta mosca en su etapa adulta se encuentra reposando en las hojas, iniciando daño en las hojas, debido a que de adultas se alientan del mesófilo de las hojas, presentándose como minas de puntos blancos dentro de la hoja, disminuyendo su capacidad fotosintética, al inicio se puede presentar en un 10% de afectación de la hoja, sin embargo, sino se controla llega a expandirse al 100% afectando la producción.

La presente investigación pretende es cubrir la necesidad de implementar un sistema móvil en el cual identifique el estado situacional de una hoja de planta lúcuma, dicha planta es usado como insumo para la producción en el distrito de Villa el Salvador, el fruto de esta planta es comercializado por los agricultores en los distritos vecinos y poder exportar a otros países como Suiza, Estados Unidos entre otros, por tal motivo es necesario contar con un sistema que logre identificar si las hojas están sanas o con tipo de plaga del gorgojo, o plaga la ranca, para su tratamiento adecuado y a tiempo, esto de la mano con la tecnología móvil, ya que estos dispositivos cuentan con una cámara y es de fácil disponibilidad de los agricultores.

En base a la problemática presentada por INIA (Instituto Nacional de Innovación Agraria), donde expone que los frutos de lúcuma se ven afectados por las plagas existentes, donde la afectación es por la plaga el gorgojo, la ranca, etc. Así también lo refiere en la Resolución directoral N° 0007-2021-INIA-DGIA de Lima la 8 de julio donde refiere una necesidad de realizar una modificación genética de la Poutería lúcuma para mitigar este problema, generando perdidas en la producción de este importante fruto para la producción y exportación tanto a otros distritos y al extranjero, por tanto, en campo los agricultores quisieran conocer de manera temprana y oportuna si su sembrío se ve afectado por estas plagas.

Investigaciones vinculadas al presente contexto encontramos a N. Zhao en [3], donde implementa un sistema cuantitativo completo para aportar mucha información, como el nivel de desastre de plagas, el consumo de plaguicidas y las estrategias de control de plagas, basándose de las imágenes obtenidas de campo para ser aplicada en una red

convolucional deformable para debilitar la influencia de los cambios de forma de las plagas, logrando obtener una precisión de 93.2%.

Desde el punto de vista internet de las cosas por R. Priya en [4], donde en la fase de crecimiento de las plantas, toma información general y predice la infestación de plagas de cultivo, donde utiliza para mejorar la predicción inferencia difusa optimizada. Logrando identificar el momento adecuado para la realización del plantío de cultivo con el fin de que la ocurrencia de la plaga sea la mínima posible. Así también existe una investigación del presente autor en [5] donde se aplica Deep Learning para el reconocimiento de plagas en papas en la sierra de Cajamarca – Perú, logrando un 99.42% en la precisión de la detección de plaga. También en aplicaciones móviles para la clasificación de imágenes encontramos usando red neuronal convolucional en la fuente de Abhinav en [6]. También encontramos la fuente de Piscocya en [7] donde ha obtenido un 82.6% de efectividad en la identificación de plagas y enfermedades del cultivo de sandía en el distrito de Ferreñafe, tomando como fuentes para su entrenamiento 200 imágenes digitales. También encontramos a Córdova en [8] aplicó el aprendizaje profundo para la detección y clasificación automática de insectos agrícolas en trampas pegantes usando CNN y Yolo obteniendo una tasa de efectividad de 94.06% y 95.82 respectivamente usando 500 imágenes de trampas pegantes para la extracción de características (con los algoritmos Otsu, Canny y SLIC).

II. MARCO TEÓRICO

A. Inteligencia Artificial

Para Mora en [9] considera que la IA puede ser una herramienta valiosa para desarrollar herramientas educativas rentables e innovadoras que puedan implementarse en países de bajos ingresos y afectados por crisis. También Haddock en [10] considera que se debe usar la inteligencia artificial para facilitar la aplicación de la simulación al diseño y análisis de sistemas de fabricación. También Dubravova en [11] considera que la IA podría aportar un valor significativo, como el análisis de datos y la asistencia en la producción de documentos. Sin embargo, cabe destacar que esta tecnología todavía se encuentra en sus primeras etapas de desarrollo y su implementación requeriría un enfoque cuidadosamente considerado que implique la colaboración interdisciplinaria y más investigaciones para poner a prueba los supuestos teóricos presentados en este estudio. Para Eloisa en [12] considera que las herramientas basadas en IA mejoran la comprensión de los datos para obtener más conocimientos, pero los desafíos incluyen la falta de alineación entre la academia y la industria, muestras industriales limitadas para la aplicación de la IA, brechas en la capacitación académica, complejidad de los algoritmos y mala interpretación de la toma de decisiones. Tomando en cuenta el comentario de Vasiliki en [13] donde en las últimas décadas se ha escrito mucho sobre cada uno de estos avances y su impacto desde una perspectiva legal. Sin embargo, la "cooperación" de estas nuevas tecnologías aún no

ha sido objeto de debate, aunque su interacción ya se ha hecho evidente.

Por tanto, la inteligencia artificial emula en lo posible todas las capacidades propias del humano plasmadas en algoritmos, agenciándose de tecnologías de información (software y hardware) para su implementación.

1) *Visión artificial*: Para Batista en [14] permite que las computadoras y los sistemas obtengan información significativa de imágenes digitales, videos y otras entradas visuales, y tomen medidas o hagan recomendaciones basadas en esa información. Por tanto, la visión artificial se basa extraer información a partir de una escena.

Para el caso en particular, se tiene usar esta herramienta para poder resaltar los patrones característicos de la hoja de lúcuma con las plagas o enfermedades si las tuviese.

En la publicación de Kun en [15] ha formalizado en la ecuación 1 para encontrar la neurona vencedora ya se realiza por un método competitivo, la ecuación 2 refleja la actualización de pesos y la ecuación 3 para encontrar el radio. Este modelo de clasificación por medio de la red SOM se puede aplicar en la clasificación plagas a partir de las imágenes que son obtenidas en campo, considerando los parámetros de entrada posterior a la obtención del representativo para poder utilizar la red neuronal.

$$G_x(i, j) = \sum_{u=-1}^1 \sum_{v=-1}^1 I(i+u, j+v) \cdot S_x(u, v), \quad (1)$$

$$G_y(i, j) = \sum_{u=-1}^1 \sum_{v=-1}^1 I(i+u, j+v) \cdot S_y(u, v), \quad (2)$$

donde $I(i, j)$ es la intensidad en la posición (i, j) de la imagen, y S_x, S_y son las máscaras de Sobel. Luego, se aplicó un filtrado mediano para reducir el ruido.

2) *Red neuronal SOM*: Creado por Kohonen donde menciona en [16] que ha mostrado el modelo de red neuronal autoorganizativo, en una red en el cual no se considera un tutor en el proceso de aprendizaje, para lograr ello se basó en la manera como la información es captada a través de los órganos sensoriales de los animales (específicamente en el córtex superior), donde las zonas detectoras se ordenan y representan internamente formas de mapas bidimensionales (la capa competitiva), donde compiten y se agrupan según las características resaltantes de cada imagen ingresadas. También para Zi en [17] considera una red neuronal de mapeo de características autoorganizada en un algoritmo típico de red neuronal no supervisada, que a menudo se utiliza para el análisis de agrupamiento y la compresión de datos.

B. Servicios en la nube

Para C&W en [18] tener una estrategia de computación en la nube para toda la empresa se vuelven de vital importancia en el mundo cada vez más digital. La adopción de la nube ha

estado en expansión rápidamente durante varios años, por lo que ha surgido una transformación digital. La computación en la nube ganó aún más importancia por el impacto de la pandemia, ya que las empresas tuvieron que reemplazar los procesos de negocio heredados y adaptarse a una fuerza laboral distribuida. Para ello se va a considerar el servicio de Firebase [19], Colab [20], y Hugging Face [21].

1) *Diseño de Base de datos:* Para el proceso de selección, ha sido necesariamente usar servicios Real time database para poder almacenarlos, tanto los resultados (diagnostico) como la URL de las imágenes tomadas, esto es necesario ya que por que las imágenes tienen diferentes dimensiones y tamaños y en conjunto requiere de mayor capacidad, por tanto, es necesario que sea almacenado en un servicio.

2) *Diseño de almacenamiento de archivos:* Para el proceso de subir imagen inmediatamente la imagen, que posteriormente (inmediatamente después) se envía la URL de dicha fuente a otro servicio, como hugging face entre otros.

C. Aplicación móvil

Para Forbes Business Council en [22] considera que crear una aplicación móvil exitosa implica una cuidadosa planificación, ejecución y mejora continua. Desde la definición del alcance y las funciones hasta el soporte posterior al lanzamiento, cada paso juega un papel vital en el éxito de su aplicación.

III. JUSTIFICACIÓN

El resolver el problema mediante un sistema integrado con servicios se justifica desde el punto de vista social, pretendiendo tener más personas inmersas en el agro de este fruto, teniendo en cuenta que puede generar nuevas fuentes laborales de Villa el Salvador. Desde el punto de vista económico lo que se pretende lograr es que los productores agrícolas tengan un control e información del estado de su sembrío, por ende, beneficios económicos para todos los involucrados. Así también desde el punto de vista científico, de la mano de la Universidad Nacional Tecnológica de Lima Sur, aporta con el conocimiento basado en Inteligencia Artificial y las áreas que las comprende para lograr mitigar cada una de las problemáticas de las plagas.

El beneficiario directo es el agricultor de Villa el Salvador, específicamente en el conocimiento del estado del fruto (Poutería lúcuma) por medio de las hojas en campo, en el sentido de la evaluación de un posible tratamiento adecuado en función de la calidad del fruto y de mejor calidad, con el fin futuro de ser parte del grupo exportador.

IV. DESARROLLO

A. Diseño del sistema

La realización del sistema ha sido implementada usando dos servicios, una con Hugging Face (para la funcionalidad del Python el cual esta lo aprendido de la red neuronal

artificial) y otro servicio con Firebase (para el almacenamiento de la imagen y el diagnostico). Cabe resaltar que lo aprendido en la red neuronal ha sido realizado en un escenario Colab con la librería keras, El aplicativo móvil se ha realizado en Android Studio Kotlin, tal como se muestra en la siguiente Fig. 1.

En la gráfica se identifica la funcionalidad del aplicativo, donde el entrenamiento lo realiza en el entorno colab (red neuronal) y a partir de ello genera dos archivos somlucuma.pkl y matrizMM.txt, donde uno es lo aprendido de la red neuronal som y otro matriz de índices respectivamente. Esto dos archivos se suben al servicio de Hugging Face, para que esté disponible quien lo desee utilizar.

La imagen para analizar se va al Storage del Firebase, dándome una url de dicha imagen, que será enviado al hugging face para que lo tome en el proceso de diagnóstico y me retorne una de las respuestas del diagnóstico aprendido. Dichos diagnósticos serán almacenados en una base de datos de tipo Real Time Database. Este proceso se puede realizar en paralelo desde varios celulares, y ser almacenados en la misma fuente de datos de imágenes (Storage de Firebase).

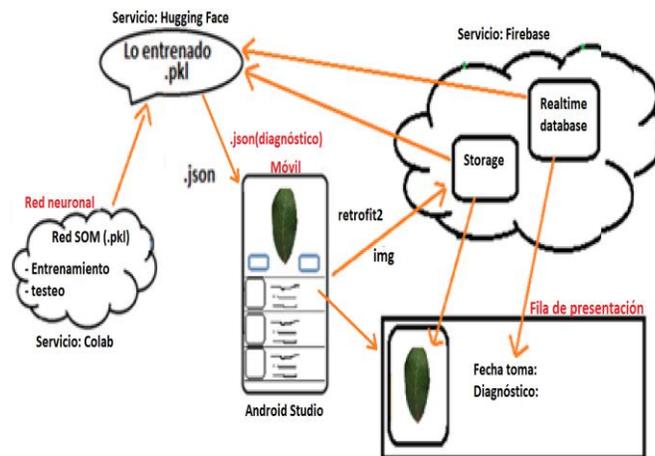


Fig 1. Diseño Back End del sistema

B. Desarrollo de Visión artificial para la extracción de patrones característicos

Para la extracción de patrones característicos primero se obtiene la imagen (de una fuente en el drive), paso siguiente se aplica la orientación (para obtener vectorialmente la representación de las hojas), a su vez se ha a convertir de matriz a vector. Dentro de la función de orientación se considera internamente aplicar la función sobel y medfilt2, para poder obtener lo característico de cada una de las hojas, esto es un paso necesario para poder apoyar a la de una red neuronal a clasificar y diferenciarlo (vectorialmente) de cada uno de los tipos de hojas. Donde n vienen a ser las 4500 imágenes de hojas obtenidas en laboratorio y listas para ser procesadas digitalmente. Tal como se muestra en la Fig 2.

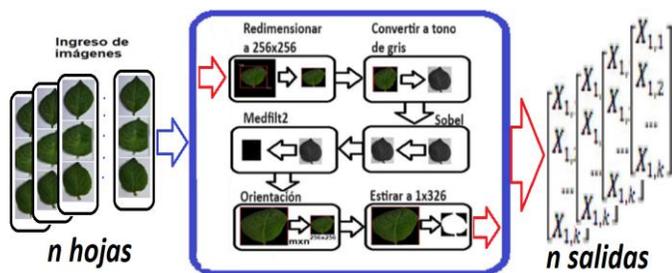


Fig 2. Proceso para la extracción de patrones característicos a partir de la hoja

Para que la imagen llegue como patrón de entrenamiento a la red neuronal artificial de tipo SOM, es necesario hacer un conjunto de procesamiento (aplicar filtros) usando la librería Open Cv, el siguiente código fuente de elaboración propia:

```

from scipy.ndimage import median_filter
from scipy.signal import convolve2d
def sobel(patron):
    gx = np.array([[ -1,  0,  1], [ -2,  0,  2], [ -1,  0,
1]], dtype=np.float32)
    gy = np.array([[ 1,  2,  1], [ 0,  0,  0], [ -1, -2, -
1]], dtype=np.float32)
    Gx = convolve2d(patron, gx, mode='valid')
    Gy = convolve2d(patron, gy, mode='valid')
    return Gx, Gy
def medfilt2(G, d=3):
    return median_filter(G, size=d)
def orientacion(patron, w):
    Gx, Gy = sobel(patron)
    Gx = medfilt2(Gx)
    Gy = medfilt2(Gy)
    m, n = Gx.shape
    mOrientaciones = np.zeros((m // w, n // w),
dtype=np.float32)
    for i in range(m // w):
        for j in range(n // w):
            Gx_patch = Gx[i*w:(i+1)*w, j*w:(j+1)*w]
            Gy_patch = Gy[i*w:(i+1)*w, j*w:(j+1)*w]
            YY = np.sum(2 * Gx_patch * Gy_patch)
            XX = np.sum(Gx_patch**2 - Gy_patch**2)
            mOrientaciones[i, j] = (0.5 *
np.arctan2(YY, XX) + np.pi / 2.0) * (18.0 / np.pi)
    return mOrientaciones
X_filtrado = []
for imagen in X:
    imagen = cv2.cvtColor(imagen,
cv2.COLOR_BGR2GRAY)

```

```

orientaciones = orientacion(imagen, w=14)
orientaciones = orientaciones.reshape(-1)
features =
np.concatenate([orientaciones.ravel()])
X_filtrado.append(features)
# Convertir a un array NumPy
X = np.array(X_filtrado)

```

C. Desarrollo del aprendizaje de enfermedades en las hojas

Para el aprendizaje de las enfermedades de las hojas, así también las hojas sanas, se realiza la siguiente secuencia previa para el ingreso de patrones característicos de la red neuronal artificial de tipo SOM. Para el entrenamiento es necesario obtener los datos en un formato idóneo para el ingreso a las redes neuronales artificiales a partir de las imágenes de las hojas, teniendo en cuenta que cada una de las imágenes tienen un fondo, dicho de otra manera, cada imagen es obtenida en laboratorio (no en campo). Este proceso se muestra en la Fig 3. Estas clases se pueden representar de esta manera según las etiquetas de la Tabla I.

TABLA I
ETIQUETAS DE LOS TIPOS DE ENFERMEDADES DE LA HOJA

Clase	Etiqueta	Detalle
0	SP	En la imagen identificado sin plaga.
1	PG	En la imagen identificado con la plaga del gorgojo.
2	ER	En la imagen identificado con la enfermedad de la Rancho.

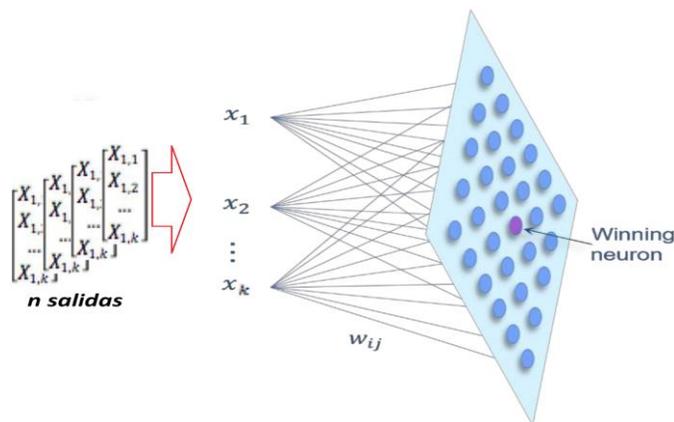


Fig 3. Patrones característicos (de imágenes) a la red neuronal SOM.

Dentro del proceso de aprendizaje el modelo de tipo red neuronal SOM (o mapa autoorganizativo) es un modelo en el cual se realiza solo buscando de manera vectorial los valores angulares de similares características, motivo por el cual hace más sencilla su implementación, así como para el proceso de

aprendizaje de la red, la estructura de la red SOM refleja en el siguiente código fuente:

```
max_iter = 100000
som = MiniSom(x=50, y=50, input_len= X.shape[1],
sigma=1.0,
learning_rate=0.5,activation_distance='euclidean',
topology='hexagonal',
neighborhood_function='gaussian')
som.random_weights_init(X)
# Encontrar la precision del modelo
precision = 0
for i in range(len(y)):
    if y[i]==Y_simulado[i]:
        precision = precision + 1
precision = precision/float(len(y))
print(str(100*precision)+"%")
99.31773879142301%
```

A partir de estos datos de configuración para la red neuronal SOM, donde se ha aplicado estos parámetros adecuados, genera una matriz de índices tal como se visualiza en la Figura 4.

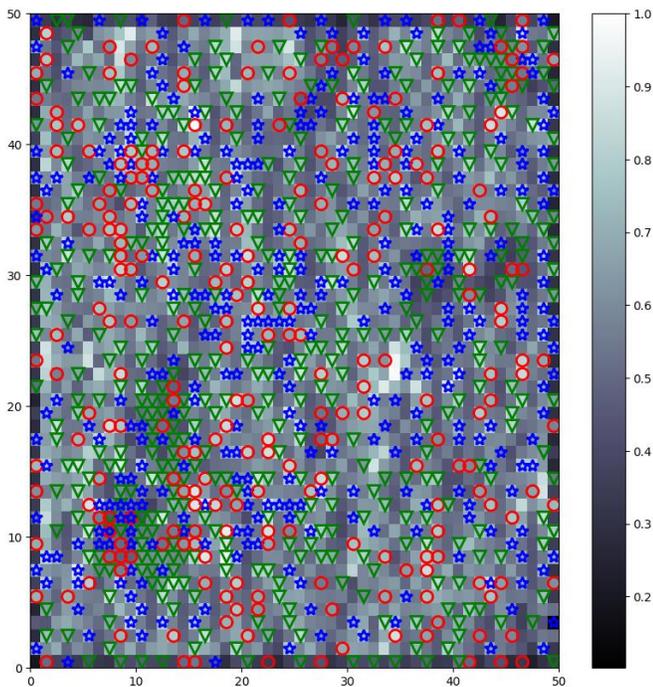


Fig 4. Matriz de índices donde refleja el agrupamiento de datos.

Es necesario grabar lo aprendido por la red neuronal SOM, para ello se almacena en un solo archivo, para su fácil portabilidad, tal como se muestra en la Fig 5.

```
# Guardar lo aprendido
import pickle
with open('modelo_som.pkl','wb') as outfile:
    pickle.dump(som,outfile)
```

Figura 5. Código en Python fuente que me permite grabar lo aprendido de una red som.

D. Desarrollo del diseño de requisitos

De acuerdo con la Fig. 6 se muestra los siguientes datos:

Agricultor: Es la persona que va a tomar foto y va a conocer el diagnóstico de cada una de las hojas.

Tomar fotos: El dispositivo tiene la parte de tomar foto o seleccionar una foto de la galería de fotos que puede tener el dispositivo.

Realizar diagnóstico: Es la parte del móvil, donde va a recepcionar el diagnóstico a partir del envío de la foto a un servidor.

Diagnóstico con la red neuronal: La red neuronal aprendida (en formato .pkl) se va a almacenar en un servidor gratuito llamado Hugging Face, donde tan solo va a requerir solo una imagen para poder realizar la comprobación de lo aprendido, dicha comprobación será el diagnóstico.

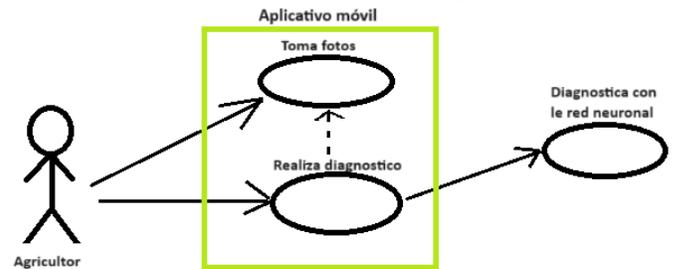


Fig 6. Funcionalidad del sistema con el usuario (el agricultor)

E. Desarrollo del aplicativo móvil (Front End)

La realización de la interfaz ha sido implementada en una sola actividad en Kotlin en Android Studio en su versión actual, el cual contendrá las acciones de realizar la selección de fotos a partir de una galería de fotos que puede tener el dispositivo móvil, con la intención de poder realizar las pruebas no necesariamente en campo, facilitando al usuario agricultor para realizar las pruebas en cualquier momento.

Un botón (button) “Cargar”, en el cual al clickar se habilitará la lista de galería de imágenes de su dispositivo, el cual podrá seleccionar la hoja que desee evaluar.

Un botón (button) “Enviar”, el cual permitirá subir a un servicio donde esta lo aprendido por la red neuronal, para realizar la evaluación de la imagen y a la vez le responderá el diagnóstico. En la parte inferior se ve una lista de imágenes (Listview) que ha tomado con su diagnóstico y fecha cuando ha sido tomado.

Por tanto, para que el usuario le sea sencillo el uso del sistema para el diagnóstico de la hoja de lúcumo, es necesario colocar solo estas acciones para el usuario.

Donde contempla una pantalla (image view) en el cual se mostrará la imagen que ha seleccionado. La interfaz esta realizado de la siguiente forma en la Fig. 7.

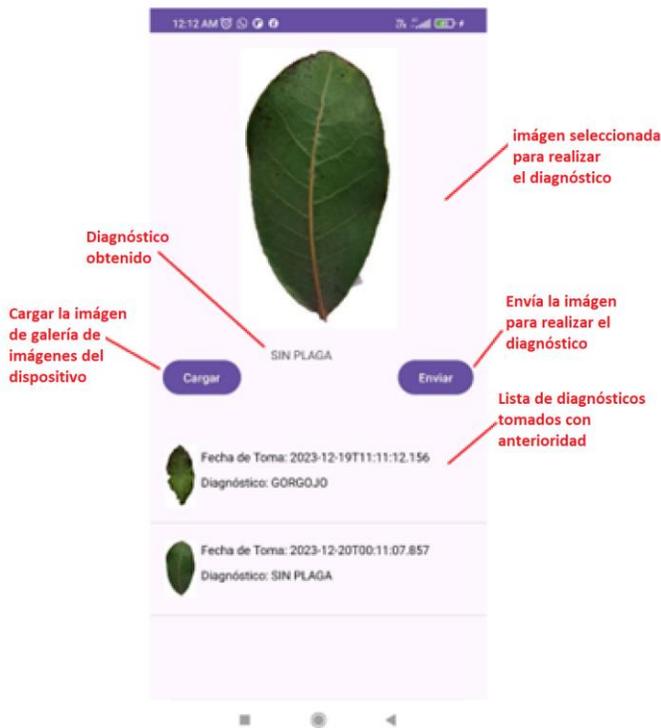


Fig 7. Diseño del aplicativo móvil (Front End)

Así para el proceso de espera de la respuesta que de un servicio aplicando la red SOM y mostrar el resultado se muestra el siguiente código fuente kotlin:

```

if (response.isSuccessful) {
    val responseData = response.body()
    responseData?.let { it: ResponseData
        val prediction = it.prediction.toInt()
        var diagnostico :String= "UNKNOWN"
        when(prediction) {
            0-> diagnostico = "SIN PLAGA"
            1-> diagnostico = "GORGJO"
            2-> diagnostico = "RANCHA"
            else-> diagnostico = "UNKNOWN"
        }
        binding.lbldiagnostico.setText(diagnostico)
    }
}

```

E. Desarrollo del funcionamiento de los servicios (Front End y Back End)

En la Fig 8 se detalla la funcionalidad con 2 servicios (Hugging Face y Firebase), y el medio de comunicación que me permite acceder a los datos propios del Firebase que es por medio de un archivo google-service.json, el cual este archivo es generado cuando crea un servicio de tipo Firebase, como canal de conexión entre la parte móvil y los servicios.

Cabe resaltar que la base de datos esta implementado en una base de datos no relacional, por tanto, no se considera un diagrama de entidad relación.

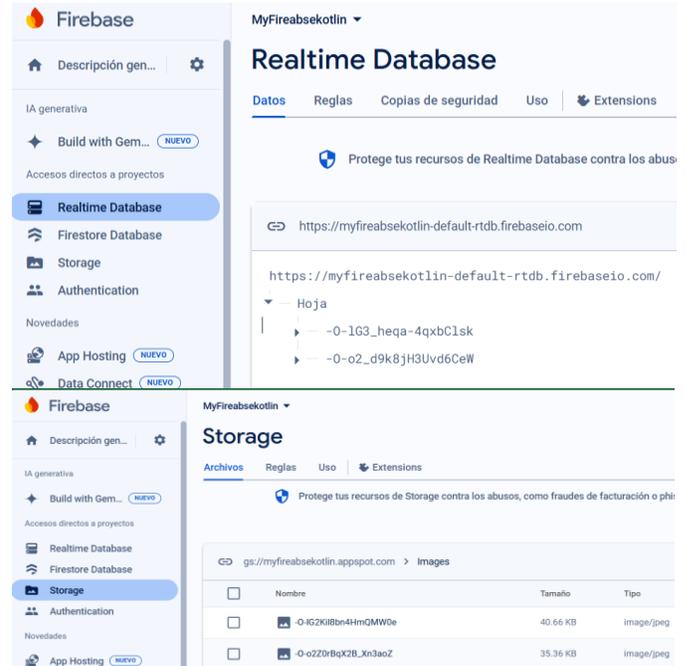


Fig 8. Servicio Realtime database y Storage de Firebase para el sistema

El servicio utilizado es el Hugging Face en la Fig. 9. Este servicio ofrece adicionalmente al almacenamiento de datos de tipo Python, a soluciones en tiempo real.

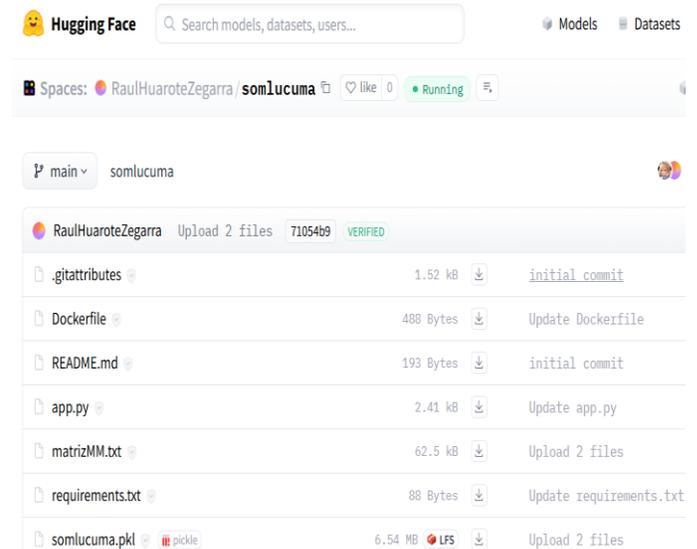


Fig 9. Servicio Hugging Face para el sistema

Implementación en Hugging Face: El archivo entrenado se aloja en un espacio el hugging_face, con nombre que considere, pero se recomienda que sea referencial, para ser invocado, para su uso en la aplicación móvil. Esto permite

realizar solicitudes mediante un API desde el dispositivo móvil. Para ello ha sido necesario considerar 3 archivos esenciales, como es el Dockerfile, requirements.txt y app.py tal como se muestra en la Figura 10, 11 y 12 respectivamente lo que contiene cada uno de estos archivos.

```

Dockerfile
1 # Usa una imagen base de Python
2 FROM python:3.9
3 # Establece el directorio de trabajo
4 WORKDIR /code
5
6 # Copia los archivos necesarios al contenedor
7 COPY ./requirements.txt /code/requirements.txt
8 RUN pip install --no-cache-dir -r /code/requirements.txt
9 RUN apt-get update && apt-get install ffmpeg libsm6 libxext6 -y
10
11 COPY app.py modelo_som.pkl ./
12 COPY matrizMM.txt ./
13
14 COPY . .
15
16 RUN chmod -R 777 /code
17
18 # Comando para ejecutar la aplicación
19 CMD ["uvicorn", "app:app", "--host", "0.0.0.0", "--port", "7860"]

```

Fig 10. Código en dockerfile para los requerimientos de software del funcionamiento del servicio en Hugging Face.

```

requirements.txt
1 scikit-learn
2 fastapi
3 numpy
4 opencv-python
5 setuptools
6 minisom
7 requests
8 pydantic
9 uvicorn
10 opencv-python
11 matplotlib
12 python-multipart

```

Fig 11. Lista de requerimientos de librerías de Python para el funcionamiento del servicio en Hugging Face.

```

28 @app.post('/predict/')
29 async def predict(file: UploadFile = File(...)):
30     try:
31         image = Image.open(BytesIO(await file.read()))
32         image = np.asarray(image)
33
34         prediction = analizarenfermedad(image)
35
36         return {"prediction": prediction}
37     except Exception as e:
38         raise HTTPException(status_code=500, detail=str(e))

```

Fig 12. Código fuente en Python en el app.py en Hugging Face.

V. CONCLUSIONES

Al aplicar la secuencia propuesta para esta investigación basada en visión artificial y aplicar la red neuronal artificial SOM para el proceso de aprendizaje de enfermedades de plagas, se obtuvo como resultado en la clasificación de plagas en las hojas de lúcuma (SP, PG, ER) con 4500 imágenes obtenidas en laboratorio, logrando realizarlo a partir de un aplicativo móvil siendo sencillo de usar, así como los resultados serán confiables, ya que queda demostrado estadísticamente los resultados (matriz de confusión en Fig 13), donde se obtiene un 99.32% de accuracy, un f1-score de

99%, 100% y 0.99% para la enfermedad SP, ER y PG respectivamente.

Matriz de Confusión - DATOS ORIGINALES:

```

[[246  0  2]
 [ 2 421  2]
 [ 1  0 352]]

```

Métricas de Matriz de Confusión - DATOS ORIGINALES:

	precision	recall	f1-score	support
0	0.99	0.99	0.99	248
1	1.00	0.99	1.00	425
2	0.99	1.00	0.99	353
accuracy			0.99	1026
macro avg	0.99	0.99	0.99	1026
weighted avg	0.99	0.99	0.99	1026

Fig 13. Resultado de la Matriz de confusión con pruebas de 1026 imágenes que no han sido parte del aprendizaje

Estos porcentajes alentadores se logró por los ajustes en las configuraciones del modelo de la red neuronal SOM, siendo los ideales (ver apartado IV.C) y los algoritmos basados en visión computacional usados (sobel, medfield y orientación) son los idóneos, por tanto, se cumple con la “Identificación de plaga en la hoja de lúcuma basado en inteligencia artificial con una tasa de efectividad mayor a 80% en el sembrío del distrito de villa el salvado por medio de un sistema móvil”.

Se logró disminuir la plaga por área de sembrío en un 1.5% (de manera visual) por el cuidado tenido al identificar la plaga en la hoja, por tanto, se puede decir que considerando las limitaciones propias de la naturaleza (espera de temporada) cumple con la “Identificación de plaga en la hoja de lúcuma basado en inteligencia artificial con una disminución de plagas por área formada por los agricultores en el sembrío del distrito de villa el salvado por medio de un sistema móvil”.

Se logró encontrar el nivel de sencillez, en la pregunta 5, donde el 33% está extremadamente satisfecho, 45% está muy satisfecho, 22% algo de satisfacción y 0% para no tan satisfecho con nada satisfecho. (de la encuesta realizada a los 09 agricultores).

VI. DISCUSIÓN

Para la identificación de plagas a la hoja de lúcuma en visión computacional con los modelos propuestos (sobel, medfield y orientación) con la red neuronal SOM en la presente investigación indica que es un modelo idóneo para el reconocimiento de plagas, a comparación de Piscocoya en [7] usó el modelo Orbit y Surf con la red neuronal convolucional, ya que la presente investigación ha obtenido un 16.72% de mejora.

En este sentido con los modelos propuestos (sobel, medfield y orientación) con la red neuronal SOM en la presente investigación indica que es un modelo idóneo para el reconocimiento de plagas, a comparación Córdova en [8] donde usó el modelo Otsu, Canny y SLIC con la red neuronal convolucional (CNN) y Yolo, ya que en la presente

investigación se ha obtenido una mejora de 5.26% y 3.5% respectivamente.

Cabe resaltar que se ha usado estos modelos y estrategias específicamente para hojas de planta para identificar las plagas presentes (si los hubiera) en la hoja de lúcuma, mas no para otros tipos de imágenes diferentes a las hojas, ya que para otros casos (diferentes a las hojas) se tendría que analizar otros detalles dependiendo el contenido de la imagen.

VII. RECOMENDACIONES

Tomar como fuente bibliográfica la presente investigación, ya que permitirá a futuras investigaciones tomar en cuenta los algoritmos usados, tanto para la extracción de características basados en visión computacional, así como el modelo basado en red neuronal y la manera como interactuar ambas áreas de la Inteligencia artificial.

Se recomienda aplicar estos modelos para la identificación de plagas en hojas, ya que da buen resultado en el accuracy, frente a otras imágenes, que necesariamente se tiene que hacer un estudio adicional.

Se recomienda que el fondo de las imágenes de las hojas sea estándar (fondo blanco de preferencia), ya que esto ayuda a la identificación de las características, ya que puede darse errores en la segmentación de la hoja en una escena.

AGRADECIMIENTO

Agradecimiento a la Universidad Nacional Tecnológica de Lima Sur, y su área de investigación por el apoyo para lograr concretar la presente investigación.

REFERENCIAS

- [1] Agroecología en Perú, "ASPECTOS RELEVANTES EN LA FASE DEL CULTIVO: CASO PAPA", 2022.
- [2] Dirección General de Desarrollo Agrícola y Dirección de Ciencia y Tecnología Agropecuaria (DICTA) - Honduras, "Manejo de la mosca minadora (*Liriomyza huidobrensis*) en el cultivo de la papa", 2019.
- [3] N. Zhao, L. Zhou, T. Huang, M. Farag Taha, H. Yong and Q. Zhengjun, "Development of an automatic pest monitoring system using a deep learning model of DPeNet", *Measurement*, Volume 203, 15 November 2022.
- [4] R. Priya Sharma, R. Dharavath and R. Damodar, "IoFT-FIS: Internet of farm things based prediction for crop pest infestation using optimized fuzzy inference system", *Internet of Things*, Volume 21, April 2023.
- [5] Huarote R, Cabrera Elis & Llanos Katherine, "Deep Learning to support the recognition of pests and diseases in Yungay potato crops in the province of Cutervo, Angurra hamlet, Perú", *LACCEI*, 2023.
- [6] A. Kumar, A. Sharma, V. Bharti, A. K. Singh, S. K. Singh and S. Saxena, "MobiHisNet: A Lightweight CNN in Mobile Edge Computing for Histopathological Image Classification," in *IEEE Internet of Things Journal*, vol. 8, no. 24, pp. 17778-17789, 15 Dec.15, 2021, doi: 10.1109/JIOT.2021.3119520.
- [7] Piscocoya J., "Sistema De Visión Artificial Para Apoyar En La Identificación De Plagas Y Enfermedades Del Cultivo De Sandía En El Distrito De Ferreñafe", 2019, url: https://tesis.usat.edu.pe/bitstream/20.500.12423/2356/1/TL_PiscocoyaFerre%C3%B1anJesus.pdf
- [8] Córdova C., "Aplicación de aprendizaje profundo para la detección y clasificación automática de insectos agrícolas en trampas pegantes: una revisión de literatura", 2021, URL: <https://tesis.pucp.edu.pe/repositorio/handle/20.500.12404/18092>
- [9] Andrés L Mora & Erin Debiasi, "LEVERAGING ARTIFICIAL INTELLIGENCE FOR DEVELOPMENT OF A COST-EFFECTIVE BRONCHOSCOPY SIMULATOR FOR RESOURCE-CONSTRAINED SETTINGS", *CHEST*, 2023.
- [10] Jorge Haddock & Robert M. O'Keefe, "Using artificial intelligence to facilitate manufacturing systems simulation", *Computers & Industrial Engineering*, 1990
- [11] Hana Dubravova, Eno Cap, Kristyna Holubova & Lukas Hribnak, "Artificial Intelligence as an Innovative Element of Support in Policing", *Procedia Computer Science*, 2024.
- [12] Eloisa Bagnulo, Giulia Strocchi, Carlo Bicchi & Erica Liberto, "Industrial food quality and consumer choice: Artificial intelligence-based tools in the chemistry of sensory notes in comfort foods (coffee, cocoa and tea)", *Trends in Food Science & Technology*, 2024.
- [13] Vasiliki Papadouli & Vagelis Papakonstantinou, "A preliminary study on artificial intelligence oracles and smart contracts: A legal approach to the interaction of two novel technological breakthroughs", *Computer Law & Security Review*, 2023.
- [14] J. Batista Miranda, J. Quinteiro, M. López de Mesa, A. Bassas-Parga, J. Monzón Falconi, L. Hernández-Acosta & D. Quinteiro-Donaghy, "07 - ARTIFICIAL INTELLIGENCE TO READ URODYNAMIC TRACINGS: COULD WE SKIP A HUMAN READING?", *Continence*, 2024.
- [15] Kun Zhang, Yuming Zhang, Pu Wang, Ye Tian & Jun Yang, "An Improved Sobel Edge Algorithm and FPGA Implementation", *Procedia Computer Science*, 2018.
- [16] T. Kohonen, *Self-Organizing Maps*, Springer, 2001
- [17] Zi Ye, Kai Yu, Gong-De Guo & Song Lin, "Quantum self-organizing feature mapping neural network algorithm based on Grover search algorithm", *Physica A: Statistical Mechanics and its Applications*, 2024.
- [18] C&W Business, "Cloud computing in 2024: Trends, challenges, and opportunities", 2024.
- [19] Firebase Documentation. <https://firebase.google.com/docs>
- [20] Colab Documentation. <https://colab.research.google.com/github/jakevdp/PythonDataScienceHandbook/blob/master/notebooks/01.01-Help-And-Documentation.ipynb>
- [21] Firebase Documentation. <https://huggingface.co/docs>
- [22] Forbes Business Council, "From Concept To App Store: 10 Steps For Mobile App Development Success In 2024", 2024.