

Estudio comparativo de arquitecturas de CNNs en hojas de Pimiento Morrón infectadas con virus PHYVV o PEPGMV

Eduardo Díaz-Gaxiola, Zuriel E. Morales-Casas, Oscar Castro-López, Gerardo Beltrán-Gutiérrez, Ines F. Vega-López, Arturo Yee-Rendón

Universidad Autónoma de Sinaloa, Culiacán, Sinaloa, México
{eduardogaxiola, zurielernesto, oscarcastro, gerardo, ifvega, arturo.yee}@uas.edu.mx

Resumen. En este trabajo, presentamos un estudio comparativo de dos arquitecturas de Redes Neuronales Convolucionales (CNNs por sus siglas en inglés): MobileNet y VGG19. Estas arquitecturas de CNNs son utilizadas para construir modelos predictivos para clasificar hojas de pimiento morrón en tres clases: sana o infectada con virus PHYVV o PEPGMV a partir de imágenes digitales. La idea de este estudio es medir la *accuracy*, *precision* y *recall* de las arquitecturas de CNNs, más allá de las técnicas tradicionales de *Machine Learning*, para prevenir y detectar plagas y enfermedades en plantas de pimiento morrón. Los resultados obtenidos en los experimentos muestran una gran ventaja de las arquitecturas CNNs sobre las técnicas tradicionales de *Machine Learning*.

Palabras clave: arquitecturas de redes neuronales convolucionales (CNNs), MobileNet, VGG-19, técnicas tradicionales de machine learning, imágenes digitales de hojas de pimiento morrón.

A Comparative Study of CNNs Architectures in Bell Pepper Leaves Infected with PHYVV or PEPGMV

Abstract. In this paper, we present a comparative study of two Convolutional Neural Networks (CNNs) architectures: MobileNet and VGG-19. These CNNs architectures are used to build predictive models to classify bell pepper leaves into three different classes, healthy or infected with virus PHYVV or PEPGMV from digital images. The idea of this study is to measure the accuracy, precision and recall from CNNs architectures, beyond the traditional Machine Learning techniques, and to prevent and detect pests and diseases in bell peppers plants. The results obtained in the experiments show a great advantage of the CNNs over the traditional Machine Learning techniques.

Keywords: convolutional neural networks (CNNs) architectures, MobileNet, VGG-19, traditional machine learning techniques, bell pepper leaves.

1. Introducción

Las plagas y enfermedades de plantas son el principal problema en agricultura que podría desestabilizar la producción global de alimentos. El pimiento morrón es un cultivo de gran valor económico que se extiende globalmente. La producción y comercialización de pimiento morrón ha ido en crecimiento en los últimos años. Solo en México, la producción de 2016 alcanzó 104.4 miles de toneladas y se posiciona como el quinto producto agrícola más exportado [18].

En este trabajo, se lleva a cabo un estudio comparativo entre técnicas de aprendizaje profundo (*Deep Learning*) y técnicas tradicionales de *Machine Learning*. La idea es construir modelos predictivos para clasificar hojas de pimiento morrón en dos clases: sana o infectada. En este trabajo nos concentramos en cultivos de pimiento morrón que sean afectadas por el Virus Huasteco de Vena Amarilla del Chile (PHYVV) [13] o el Virus del Mosaico Dorado del Chile (PEPGMV) [14].

Los métodos tradicionales para análisis de cultivos de pimiento morrón en búsqueda de PHYVV y PEPGMV no son eficientes. Por ejemplo, los expertos en el área hacen uso de laboratorios especializados y el proceso es lento y costoso. Por otro lado, el análisis visual de plantas y hojas por parte de expertos es más rápido que los análisis de laboratorio. Sin embargo, incluso para los expertos, analizar grandes áreas de cultivos se vuelve impráctico. Por estos motivos proponemos automatizar la detección de los virus PHYVV y PEPGMV en hojas de pimiento morrón a partir de imágenes digitales, utilizando técnicas de *Deep Learning* y *Machine Learning*.

La identificación de especies de plantas a partir de imágenes es un reto muy grande en el área de la visión por computadora. En particular, es necesario tratar con formas y texturas irregulares, así como una gran cantidad de variabilidad intra-clase y pequeñas diferencias inter-clase [22]. La literatura reporta un gran número de enfoques para la identificación de plantas a partir de imágenes de sus hojas [8,10,26,27]. Todos estos enfoques tienen excelentes resultados con técnicas tradicionales de *Machine Learning*, pero utilizan conjunto de datos (*datasets*) limitados y con imágenes tomadas bajo condiciones bien controladas (fondo, iluminación, orientación, etc.)

Cuando utilizamos técnicas de *Machine Learning* para reconocer objetos a partir de imágenes, es necesario en primer lugar encontrar una representación formal y estructurada que caracterice los atributos importantes para ayudar al proceso de clasificación. Usualmente, esta caracterización se presenta como un vector, llamado vector de características (*feature vector*). Hasta hace unos años, el proceso de extracción de características era realizado por humanos expertos. Recientemente, este proceso se ha delegado a los algoritmos de *Machine Learning* para identificar las características relevantes para el aprendizaje. Las técnicas de *Machine Learning* que utilizan este método de caracterización se denominan *Deep Learning* [2]. Para diferenciar estos enfoques, llamaremos técnicas tradicionales de *Machine Learning* a aquellas que requieran de una fase de caracterización por parte de humanos expertos.

En este trabajo, presentamos un estudio de las arquitecturas de CNNs MobileNet y VGG-19, y las técnicas tradicionales de *Machine Learning*, Regresión Logística (LR), Red Neuronal Artificial (ANN por sus siglas en inglés) y Máquinas de Soporte Vectorial (SVMs por sus siglas en inglés) sobre un *dataset* de imágenes de hojas de pimiento morrón infectadas con virus PHYVVV o PEPGMV. Las imágenes de las hojas se encuentran etiquetadas como una de las tres clases: sanas o infectada con virus PHYVVV o PEPGMV.

El resto de este trabajo se encuentra organizado de la siguiente manera. En la sección 2, presentamos una revisión de la literatura de enfoques existentes en identificación de plantas a partir de imágenes basados en técnicas tradicionales de *Machine Learning*, así como técnicas de *Deep Learning*. En la sección 3, describimos las arquitecturas de CNNs: MobileNet y VGG-19. En la sección 4, presentamos la adquisición de datos y preprocesamiento de imágenes para la extracción de características para técnicas tradicionales de *Machine Learning*. La sección 5 presenta los resultados de una serie de experimentos detallados basados en simulación, los cuales muestran la precisión de los modelos predictivos. Por último, en la sección 6 presentamos nuestras conclusiones y dirección de investigación futura.

2. Antecedentes y trabajo relacionado

Existen muchos enfoques que se centran en la detección de plantas usando técnicas tradicionales de *Machine Learning*. Por ejemplo, el proyecto LeafSnap [10] identifica especies de árboles a partir de fotografías de sus hojas. Para hacer esto, utiliza técnicas de segmentación y extracción de características, apoyándose principalmente en medidas de la curvatura de las hojas. Así mismo, ApLeaf [27] hace uso de extracción de características (principalmente de hojas) para identificar especies vegetales del mediterráneo Francés.

Jamit et al. [8] reportan un análisis de sensibilidad de varios métodos de caracterización y concluyeron que, para la clasificación de plantas a partir de imágenes de las hojas, las características basadas en textura llevan a modelos más precisos, alcanzando valores de 94 % usando una base de datos controlada de imágenes de hojas cuidadosamente seleccionadas, maduras, y libres de enfermedades o parásitos. Las hojas son colocadas sobre un fondo plano y fotografiadas en un entorno controlado. El experimento incluye solamente siete especies. Los valores de precisión y el hecho de que las características de textura resultan en mejores modelos son consistentes con los resultados reportados por Zhong-Qiu Zhao et al. (2015) [26] con 4,870 imágenes de hojas extraídas de la base de datos del reto ImageCLEF en 2012, correspondiendo a 128 especies.

Sfar et al. (2015) [19] proponen un algoritmo basado en *confidence sets* para clasificar especies de plantas a partir de imágenes de sus hojas. Los resultados reportados corresponden a experimentos llevados a cabo con conjuntos de datos que varían en tamaño y calidad. En las imágenes, las hojas aparecen cuidadosamente colocadas en un fondo blanco. La precisión lograda con esta propuesta es muy buena para imágenes tomadas en condiciones ideales (97 %

para el conjunto de datos Swedish Leaf), pero es drásticamente afectada por dos factores: cuando las imágenes son tomadas en condiciones reales (condiciones de campo) y cuando el número de especies se incrementa (53% para el conjunto de datos ImageCLEF).

El reto ImageCLEF para la identificación de plantas a partir de imágenes es bastante más complejo que la identificación de especies usando solamente imágenes de hojas. Hasta 2011, los investigadores en esta área basaban sus enfoques en técnicas tradicionales de *Machine Learning*. Sin embargo, desde 2012 las técnicas de *Deep Learning* han mostrado ser superiores para identificar plantas a partir de imágenes [4,23]. De hecho, en el reto de ImageCLEF 2016, todos los participantes utilizaron técnicas basadas en *Deep Learning* [5]. El mejor enfoque se basó en Redes Neuronales Convolucionales (CNNs) [11], alcanzando una precisión de 82.7%.

Un enfoque presentado en el reto ImageCLEF 2015 por Reyes et al. (2015) [16] diseñaron un sistema de identificación de plantas utilizando *Deep Learning* que no requiere la caracterización explícita de las imágenes. El sistema propuesto se basa en observaciones de 3 a 5 imágenes del mismo espécimen y no en imágenes únicas para realizar la clasificación. El modelo es un proceso de dos fases. En la primera fase, llamada pre-entrenamiento, 1.2 millones de imágenes fueron utilizadas, donde aparecen objetos (muestras) de 1,000 categorías distintas. Estas categorías corresponden a objetos que un humano puede reconocer con facilidad, sin ser experto en algún área en particular. Estas imágenes se obtuvieron del reto ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [17], en su edición de 2012. En la segunda fase, refinamiento, se utilizaron 91,759 imágenes correspondientes a 13,887 plantas de 1,000 especies diferentes, incluyendo árboles, hierbas, helechos, entre otros. El objetivo de esta fase es aprender a distinguir las categorías de interés de un tema específico a partir de un modelo entrenado para reconocer objetos genéricos (a esto se le conoce como *transfer learning*). Con este modelo, Reyes et. al. obtuvieron una precisión en la identificación de plantas del 65%. Un año después, Lee et. al. (2016) [11] usaron Redes Neuronales Convolucionales (CNNs) para la propuesta que presentaron al reto ImageCLEF. Adicionalmente, utilizaron un proceso conocido como *data augmentation* para enriquecer en tamaño y variedad su conjunto de entrenamiento. Este proceso consiste en aplicar transformaciones aleatorias a las imágenes del *dataset* inicial, tales como cambios en luminosidad, orientación, fondo, oclusiones parciales, entre otras. Con este modelo, Lee et. al. lograron una precisión de 74.2%.

Las técnicas de *Deep Learning* han sido utilizadas exitosamente para detectar plagas en algunos cultivos a partir del análisis de fotografías de sus hojas. Por ejemplo, Sladojevic et. al. (2016) [21] utilizaron CNNs para clasificar 13 enfermedades que ocurren en cinco especies de cultivos pera, cereza, durazno, manzana y uva. En este trabajo, los autores utilizan *Deep Learning* para generar un modelo predictivo que está entrenado con 30,800 imágenes. La base de datos propuesta contiene solamente 4,483 imágenes, pero los autores usan *data augmentation* para incrementar el número de imágenes. La precisión de este enfoque es de 96% en Top-1 y 99.99% en Top-5. Una respuesta de Top-K retorna las K

categorías a las que el objeto es más probable que pertenezca de acuerdo al modelo predictivo. Si la categoría correcta se encuentra en cualquiera de las K propuestas, la respuesta se considera correcta.

Los enfoques más recientes en materia de identificación de plantas a partir de imágenes utilizando Redes Neuronales Convoluciones (CNNs) se describen a continuación.

Lee et. al. [11] utilizaron una arquitectura de red VGG16 modificada para el reto LifeCLEF 2016, con la meta de identificar 1000 especies de plantas y, al mismo tiempo, rechazar clases desconocidas, obteniendo resultados limitados. Hang et. al. [6] utilizaron una red VGG16 modificada (denominada Bluefield) para el reto LifeCLEF 2016. La red fue modificada removiendo la última capa de *Pooling* con una capa *Spatial Pyramid Pooling* y cambiando la función de activación estándar ReLU por la variante paramétrica, alcanzando un *Mean Average Precision* (MAP) de 0.827, el más alto de entre todos los participantes. Mehdipour et. al. [12] usaron dos arquitecturas de CNNs, en concreto GoogLeNet y VGGNet, pre-entrenadas con el *dataset* ILSVRC 2012 y refinadas con el *dataset* LifeCLEF 2015. Adicionalmente, entrenaron GoogLeNet para rechazar imágenes que no contuvieran plantas. El sistema alcanzó una calificación final de 0.738.

Toma et. al. [24] emplearon una red AlexNet pre-entrenada con el dataset PlantCLEF 2017, alcanzando un *Mean Reciprocal Rank* de 0.361 en su mejor corrida. Pawara et al. [15] utilizaron AlexNet y GoogleNet para clasificar clases de los *datasets* Folio, AgrilPlant y Swedish leaf, usando a su vez una serie de técnicas de *data augmentation* para incrementar la eficacia de los modelos. Barre et al. [1] diseñaron una arquitectura de CNN orientada a la clasificación de hojas, denominada LeafNet. Esta demostró tener un desempeño superior a métodos tradicionales para la clasificación de imágenes en los datasets Foliage, LeafSnap y Swedish Leaf.

Carpentier et al. [3] entrenaron una red con arquitectura ResNet para identificar especies de árboles nativos de Canadá a partir de imágenes de cortezas, logrando un nivel de confianza máximo de 97.81 %. Los resultados obtenidos por Waldchen et al. [25] en su revisión del estado del arte en materia de identificación de plantas demuestran que las CNNs se desempeñan consistentemente mejor que los métodos de clasificación tradicionales.

3. Descripción formal de arquitecturas de Redes Neuronales Convolucionales (CNNs)

La Red Neuronal Convolutiva (CNN) es una clase de red neuronal profunda que se utiliza en tareas de visión por computadora. Fue popularizada por Krizhevsky et al. en 2012 [9]; los autores presentaron la arquitectura de CNN AlexNet, la cual utiliza capas convolucionales consecutivas, y una función de activación ReLU. Esta arquitectura fue la ganadora del reto ImageNet en 2012.

En contraste con las técnicas tradicionales de *Machine Learning*, las CNNs poseen la ventaja de ser capaces de descubrir de forma autónoma las caracte-

terísticas necesarias para discriminar entre las clases que se le presenten. Esto elimina la necesidad de desarrollar vectores de características especializados y les permite ser utilizadas en diferentes tareas de clasificación.

El obstáculo principal cuando se utilizan las CNNs es la vasta cantidad de datos requeridos para su entrenamiento. Sin embargo, proyectos tales como ImageNet han contribuido de gran manera con datasets que cubren la alta diversidad de plantas en el mundo, mientras que proyectos como Pl@ntNet y iNaturalist han construido comunidades de voluntarios para obtener y clasificar imágenes de plantas alrededor del mundo.

3.1. Arquitectura MobileNet

La primer version de la arquitectura MobileNet [7] fue propuesta por Google, para aplicaciones de visión por computadora móviles y embebidas. MobileNet se basa en una arquitectura simplificada que utiliza convoluciones separables en profundidad, o *depthwise-separable convolutions*, para construir redes neuronales profundas y ligeras. La convolución separable en profundidad consiste de dos operaciones de convolución: la convolución en profundidad, o *depthwise convolution*, y la convolución por punto, o *pointwise convolution*. La convolución en profundidad aplica convoluciones por separado a cada canal del tensor de entrada, esto es, una convolución tradicional de $n \times m$ en una imagen a color. Posteriormente, los mapas de activación producto de las operaciones de convolución se concatenan sobre el eje de profundidad. Una convolución tradicional de 1×1 es entonces aplicada al tensor resultante, (*pointwise convolution*), lo cual combina los canales de los mapas de activación concatenados.

Por ejemplo, la convolución bidimensional tradicional opera sobre una ventana de $N \times M$. Sin embargo, también opera sobre todos los canales del tensor de entrada; esto significa que si el tensor de entrada es una imagen RGB (3 canales de profundidad), un kernel de convolución de 5×5 en realidad tendría la forma $5 \times 5 \times 3$. Así, cada operación de convolución realiza $5 \times 5 \times 3 = 75$ operaciones. Usando la *depthwise-separable convolution*, el número de operaciones se reduce a $5 \times 5 + 3 = 28$, lo cual es significativamente más eficiente.

La arquitectura de la red MobileNet se basa en las *depthwise-separable convolution*, con la excepción de su primera capa, la cual es una capa de convolución tradicional. Todas las capas son seguidas de *batch normalization* y cuentan con la función de activación ReLU, excepto la última capa *fully-connected*. La arquitectura completa se describe en la Tabla 1.

3.2. Arquitectura VGG-19

Las arquitectura VGG fue propuesta en 2014 por Simonyan y Zisserman [20]. VGG consiste de *stacks* lineales de bloques conformados por una cierta cantidad de capas convolucionales, una función de activación no lineal y una capa *Max Pooling*, seguidos por 3 capas *fully-connected* y una capa Softmax al final.

La arquitectura VGG19 posee cinco de estos bloques. Los dos primeros contienen dos capas convolucionales de 64 y 128 filtros respectivamente; el bloque

Tabla 1. Arquitectura MobileNet.

Tipo / Stride	Forma de filtro	Tamaño de input
Conv / s2	3 x 3 x 3 x 32	224 x 224 x 3
Conv dw / s1	3 x 3 x 32 dw	112 x 112 x 32
Conv / s1	1 x 1 x 32 x 64	112 x 112 x 32
Conv dw / s2	3 x 3 x 64 dw	112 x 112 x 64
Conv / s1	1 x 1 x 64 x 128	56 x 56 x 64
Conv dw / s1	3 x 3 x 128 dw	56 x 56 x 128
Conv / s1	1 x 1 x 128 x 128	56 x 56 x 128
Conv dw / s2	3 x 3 x 128 dw	56 x 56 x 128
Conv / s1	1 x 1 x 128 x 256	28 x 28 x 128
Conv dw / s1	3 x 3 x 256 dw	28 x 28 x 256
Conv / s1	1 x 1 x 256 x 256	28 x 28 x 256
Conv dw / s2	3 x 3 x 256 dw	28 x 28 x 256
Conv / s1	1 x 1 x 256 x 512	14 x 14 x 256
Conv dw / s1	3 x 3 x 512 dw	14 x 14 x 512
Conv / s1	1 x 1 x 512 x 512	14 x 14 x 512
Conv dw / s2	3 x 3 x 512 dw	14 x 14 x 512
Conv / s1	1 x 1 x 512 x 1024	7 x 7 x 512
Conv dw / s2	3 x 3 x 1024 dw	7 x 7 x 1024
Conv / s1	1 x 1 x 1024 x 1024	7 x 7 x 1024
Avg Pool / s1	Pool 7 x 7	7 x 7 x 1024
Fully-Conn / s1	1024 x 1000	1 x 1 x 1024
Softmax / s1	Clasificador	1 x 1 x 1000

intermedio contiene tres capas convolucionales de 256 filtros, y los dos últimos contienen ambos tres capas convolucionales de 512 filtros cada uno. El 19 representa la cantidad de capas entrenables contenidas en la arquitectura: 16 capas convolucionales y 3 capas *fully-connected*.

Las capas convolucionales cuentan con un campo receptivo de 3×3 , *stride* de 1×1 y *padding* de 1 pixel. Las operaciones de Max Pooling se realizan con un *kernel* de 2×2 y un *stride* de 2×2 . Cada capa oculta de la red cuenta con la función de activación ReLU. La arquitectura VGG-19 se resume en la Tabla 2.

4. Adquisición y pre-procesamiento de datos

Se recolectaron diferentes hojas de Pimiento Morrón (bell pepper), las cuales se encontraban dentro de un invernadero experimental. De acuerdo a las especificaciones del experto en fitopatología, una manera de distinguir una hoja sana de una hoja infectada es enfocándose en los niveles de colores verde y amarillo presentes en la hoja. En Fig. 1, mostramos algunos ejemplos de hojas utilizadas en este estudio.

Reunimos 103 imágenes de hojas sanas, 103 imágenes de hojas infectadas con el virus PHYVV y 104 imágenes de hojas infectadas con el virus PEPGMV.

Tabla 2. Arquitectura VGG-19.

Tipo	N filtros/parámetros
Conv2D	64
Conv2D	64
Max Pool	N/A
Conv2D	128
Conv2D	128
Max Pool	N/A
Conv2D	256
Conv2D	256
Conv2D	256
Conv2D	256
Max Pool	N/A
Conv2D	512
Conv2D	512
Conv2D	512
Conv2D	512
Max Pool	N/A
Conv2D	512
Conv2D	512
Conv2D	512
Conv2D	512
Max Pool	N/A
Fully-Conn	4096
Fully-Conn	4096
Fully-Conn	1000
Softmax	N

Siendo un total de 310 imágenes reunidas. Las imágenes fueron tomadas utilizando un iPhone 6s (dispositivo móvil) con una cámara de 12 MP. Con la idea de resaltar las características de las hojas y mantener enfocada y estable la imagen, se utilizó un dispositivo para estabilizar y enfocar el dispositivo móvil a una altura de 30cm. Cada hoja fue colocada mostrando su área frontal. Además, por cada hoja se realizaron dos fotografías, una de ellas con fondo blanco y la siguiente con fondo negro. El propósito de los fondos diferentes fue obtener las características de las hojas en diferentes entornos, obteniendo así un total de 620 imágenes.

4.1. Pre-procesamiento de datos para la extracción de características

Para las técnicas tradicionales de *Machine Learning*, necesitamos construir vectores de características. Un vector de características describe la información relevante sobre la forma de la hoja contenida en un patrón, así que para las imágenes de hojas de Pimiento Morrón necesitamos obtener la información más

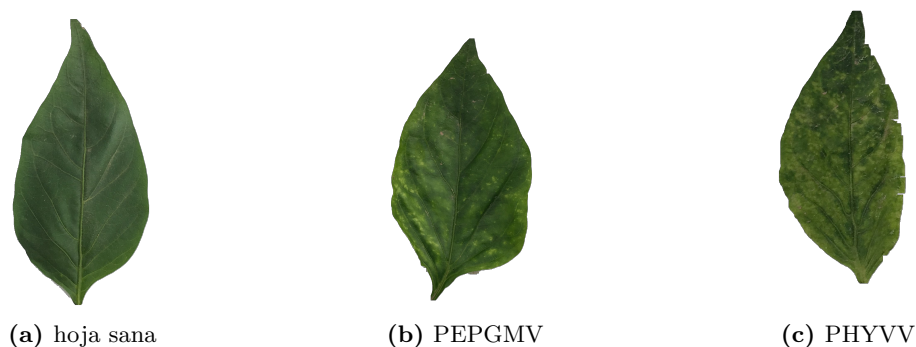


Fig. 1. Ejemplos de imágenes, (a) una hoja sana, (b) una hoja de PEPGMV y(c) una hoja de PHYVV.

relevante y representar dicha información en un espacio de baja dimensión. Utilizamos la biblioteca OpenCV para el procesamiento de las imágenes y resaltar los niveles de verde y amarillo en las hojas de Pimiento Morrón. La imagen original en formato RGB fue convertida a formato HSV. El formato HSV nos permite manipular el brillo y aislar los colores de una mejor manera para el fondo blanco y negro de las imágenes. A partir de la imagen HSV, se removió el fondo y se aplicó un proceso de binarización para resaltar los colores verde y amarillo (obteniendo dos imágenes, una para cada color). Se asignó el valor de 1 para el pixel detectado por color verde/amarillo y 0 al resto de los pixeles, ver Fig 2.

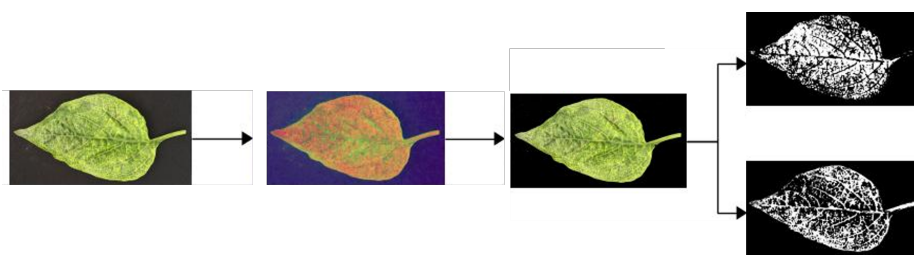


Fig. 2. Flujo de trabajo para el procesamiento de imágenes para resaltar los niveles verde y amarillo en las hojas.

Una vez generadas las imágenes en el pre-procesamiento, obtuvimos las siguientes características a partir de ellas (ver Tabla 3). Cada característica fue calculada tanto para color amarillo como para color verde. Un objeto es un conjunto de pixeles conectados, como resultado del pre-procesamiento de imágenes, estos pixeles adyacentes comparten un porcentaje de color similar.

Tabla 3. Características extraídas de las imágenes de hojas de Pimiento Morrón.

Feature	Description
PP	número de píxeles detectados en la hoja
NO	número de objetos detectados en la hoja
NOG	número de objetos detectados cuyo tamaño es mayor al 1 % de la imagen
TPO	tamaño promedio de los objetos, representado en píxeles
TPOG	tamaño promedio de los objetos filtrados, representado en píxeles.
STO	estadística que mide la variación en el tamaño de los objetos.
STOG	estadística que mide la variación en el tamaño de los objetos filtrados.

Para cada imagen, fue generado un vector de 14 variables (características) y fue etiquetado como clase sana / infectada para indicar la ausencia o presencia de virus (PHYVV o PEPGMV). En total se generaron 620 vectores de características.

5. Experimentos y resultados

Los experimentos fueron realizados utilizando dos computadoras con las siguientes características en hardware/software. Procesadores Intel Xeon W-2133, 32 GB de RAM, y una tarjeta NVIDIA GTX 1080 cada una. Sistema operativo Ubuntu 18.04, CUDA toolkit 10.0.

Para las técnicas tradicionales de *Machine Learning*, se utilizaron el lenguaje R en la plataforma Rstudio. Para la regresión logística, se utilizaron las funciones *train* del paquete *caret* y el modelo *lm*. Se utilizó el paquete *nnet* para las redes neuronales artificiales y el paquete *e1071* con la función *svm* para las máquinas de soporte vectorial.

Las arquitecturas de CNNs, MobileNet y VGG-19 fueron diseñadas utilizando Keras version 2.2.4, con Tensorflow 1.13.1 como *backend*, y ejecutándose dentro de un entorno virtual de Anaconda 4.6.7 con Python version 3.7.8.

En las técnicas tradicionales de *Machine Learning* se consideraron como el conjunto de datos un total de 620 vectores de características. Para el caso de MobileNet y VGG-19, se consideraron un total de 121 imágenes sin fondo como el conjunto de datos. Para ambos casos, seleccionamos de manera aleatoria el 75 % de los datos para el entrenamiento y el resto fue usado para pruebas.

5.1. Clasificación en dos clases: sana e infectada

Los resultados obtenidos en los experimentos son reportados en términos de **accuracy**, **precision** y **recall**.

Verdadero sana (VS), Falso sana (FS), Falso infectado (FI) y Verdadero Infectado (VI). La **accuracy** es calculada a través de la ecuación (1). La **precision** para la clase sana (2) y para la infectada (3). El **recall** para la clase sana (4) y para la infectada (5).

$$Accuracy = \frac{VS + VI}{VS + FS + FI + VI}, \quad (1)$$

$$PrecisionClaseS = \frac{VS}{VS + FS}, \quad (2)$$

$$PrecisionClaseI = \frac{VI}{VI + FI}, \quad (3)$$

$$RecallClaseS = \frac{VS}{VS + FI}, \quad (4)$$

$$RecallClaseI = \frac{VI}{VI + FS}. \quad (5)$$

Tabla 4. Matriz de confusión de regresión logística.

	clase sana (actual)	clase infectada (actual)
clase sana	32 (VS)	5 (FS)
clase infectada	13 (FI)	71 (VI)
Acc	85.12 %	

Tabla 5. Matriz de confusión de redes neuronales artificiales.

	clase sana (actual)	clase infectada (actual)
clase sana	32 (VS)	1 (FS)
clase infectada	13 (FI)	75 (VI)
Acc	88.42 %	

Tabla 6. Matriz de confusión de máquinas de soporte vectorial.

	clase sana (actual)	clase infectada (actual)
clase sana	32 (VS)	5 (FS)
clase infectada	13 (FI)	71 (VI)
Acc	85.12 %	

Tabla 7. Matriz de confusión de la arquitectura VGG-19.

	clase sana (actual)	clase infectada (actual)
clase sana	37 (VS)	8 (FS)
clase infectada	6 (FI)	70 (VI)
Acc	88.42 %	

Tabla 8. Matriz de confusión de la arquitectura MobileNet.

	clase sana (actual)	clase infectada (actual)
clase sana	43 (VS)	2 (FS)
clase infectada	3 (FI)	73 (VI)
Acc	95.86 %	

Los resultados de las técnicas tradicionales de *Machine Learning* fueron los siguientes. En Tabla 4, presentamos los resultados de la regresión logística (LR), el **accuracy** de LR fue **85.12 %**, la *precision* de la clase sana fue 86.48 %, y para la infectada fue 84.52 %. El *recall* de la clase sana fue de 71.11 % y para la clase infectada fue de 93.42 %. En Tabla 5, presentamos los resultados de ANN, el

accuracy de ANN fue **88.42%**, la **precision** de la clase sana fue de 96.96%, y de la clase infectada fue de 85.22%. El **recall** de la clase sana fue 71.11% y de la clase infectada fue 98.68%. En Tabla 6, presentamos los resultados de SVMs, el **accuracy** de SVMs fue **85.12%**, la **precision** de la clase sana fue 86.48%, y de la clase infectada fue 84.52%. El **recall** de la clase sana fue 71.11% y de la clase infectada fue 93.42%.

Los resultados experimentales para las arquitecturas de CNNs fueron los siguientes. En Tabla 7, presentamos los resultados de VGG-19. El **accuracy** de VGG-19 fue **88.42%**, la **precision** de la clase sana fue 82.22%, y de la clase enferma fue 92.10%. El **recall** de la clase sana fue 86.04% y de la clase infectada fue 89.74%. En Tabla 8, presentamos los resultados de MobileNet. El **accuracy** de MobileNet es **95.86%**, la **precision** de la clase sana es 95.55%, y de la clase infectada es 96.05%. El **recall** de la clase sana es 93.47% y de la clase infectada es de 97.33%.

El mejor **accuracy** fue el obtenido por **MobileNet (95.86%)**, seguido por VGG-19 y RNA (88.42%).

5.2. Clasificación en tres clases: sana, PHYVV y PEPGMV

En los siguientes experimentos, solamente utilizamos las arquitecturas de CNNs. Incrementando el número de clases, la complejidad del proceso de clasificación también aumenta. Para eso, las técnicas tradicionales de *Machine Learning* no pueden reconocer adecuadamente entre los virus PHYVV y PEPGMV. Por ende, decidimos solamente realizar el estudio comparativo entre VGG-19 y MobileNet.

Tabla 9. Matriz de confusión de arquitectura VGG-19.

	clase sana (actual)	clase PHYVV (actual)	clase PEPGMV (actual)
clase sana	37	3	5
clase PHYVV	1	26	13
clase PEPGMV	5	3	26
Acc	75.20%		

Tabla 10. Matriz de confusión de arquitectura MobileNet

	clase sana (actual)	clase PHYVV (actual)	clase PEPGMV (actual)
clase sana	43	0	2
clase PHYVV	3	26	11
clase PEPGMV	0	2	34
Acc	85.12%		

En Tabla 9, presentamos los resultados de VGG-19. El **accuracy** de VGG-19 fue **75.20%** y en Tabla 10, presentamos los resultados de MobileNet, el **accuracy** de MobileNet es **85.12%**.

6. Conclusión

En este artículo se llevó a cabo un estudio comparativo entre las técnicas tradicionales de *Machine Learning* y *Deep Learning* redes neuronales convolucionales (CNNs) utilizadas en el reconocimiento de hojas de Pimiento Morrón infectadas con el virus PHYVV o PEPGMV en imágenes digitales. Tenemos dos etapas experimentales. En la primer etapa, el objetivo es el determinar si una hoja de Pimiento Morrón en una imagen está infectada o no, i.e., clasificar en dos clases: sana o infectada con virus PHYVV o PEPGMV. De acuerdo con nuestros resultados, la arquitectura de CNN, MobileNet, alcanzó el mejor **accuracy** con 95.86%. Para la segunda etapa, solamente usamos CNNs con tres diferentes clases, llamadas sana, PHYVV y PEPGMV. En este caso MobileNet alcanzó el mejor **accuracy** con 85.12%. Como reportamos en los resultados experimentales, las arquitecturas CNNs tienen una clara ventaja sobre las técnicas tradicionales de *Machine Learning* en el proceso de reconocimiento de virus PHYVV o PEPGMV en imágenes digitales de hojas de Pimiento Morrón. Como trabajo a futuro, tenemos la intención de usar otras arquitecturas de CNNs y realizar un estudio comparativo.

Agradecimientos. Los autores desean agradecer el financiamiento proporcionado para esta investigación al Consejo Nacional de Ciencia y Tecnología (CONACyT) y a la Universidad Autónoma de Sinaloa (UAS).

Referencias

1. Barré, P., Stöver, B.C., Müller, K., Steinhage, V.: LeafNet: A computer vision system for automatic plant species identification. *Ecological Informatics* 40, 50–56 (2017)
2. Bengio, Y., Courville, A., Vincent, P.: Representation Learning: A Review and New Perspectives. *IEEE Transactions on Software Engineering* 35(8), 1798–1828 (2013)
3. Carpentier, M., Giguere, P., Gaudreault, J.: Tree Species Identification from Bark Images Using Convolutional Neural Networks. arXiv preprint arXiv:1803.00949 (2018)
4. Goeau, H., Pierre, B., Alexis Joly.: LifeCLEF Plant Identification Task 2015. In: CLEF: Conference and Labs of the Evaluation forum, Toulouse, France, Working Notes of CLEF (2015)
5. Goeau, H., Pierre, B., Alexis Joly.: Plant Identification in an Open World. In: CLEF: Conference and Labs of the Evaluation forum, Evora, Portugal, Working Notes of CLEF (2016)
6. Hang, S.T., Tatsuma, A., Aono, M.: Bluefield (KDE TUT) at LifeCLEF 2016 Plant Identification Task. CLEF Working Notes (2016)

7. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017)
8. Jamil, N., Hussin, N., Nordin, S., Awang, K.: Automatic Plant Identification: Is Shape the Key Feature? *Procedia Computer Science* 76(5), 436–442 (2015)
9. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: 25th International Conference on Neural Information Processing Systems, pp. 1097–11052, Curran Associates Inc., Lake Tahoe, Nevada (2012)
10. Kumar, N., Belhumeur, P., Biswas, A., Jacobs, D., Kress, W., Lopez, I., Soares, J.: Leafsnap: A Computer Vision System for Automatic Plant Species Identification. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C.(eds.) *Computer Vision ECCV 2012*, LNCS, vol. 7573, pp. 502–516, Springer, Heidelberg. https://doi.org/10.1007/978-3-642-33709-3_36 (2012)
11. Lee, S.H., Chang, Y.L., Chan, C.S., Remagnino, P.: Plant Identification System based on a Convolutional Neural Network for the LifeClef 2016 Plant Classification Task. *CLEF Working Notes*, pp. 502–510 (2016)
12. Mehdipour Ghazi, M., Yanıkoğlu, B., Aptoula, E.: Open-set plant identification using an ensemble of deep convolutional neural networks. *CLEF* (2016)
13. Melendrez-Bojorquez, N., Magallanes-Tapia, M.A., Armenta-Anaya, C., Camacho-Beltrán, E., Rodríguez-Negrete, E.A., Leyva-López, N.E., Méndez-Lozano, J.: Pepper huasteco yellow vein virus Associated to Sweet Pepper Disease in Sinaloa, Mexico. *Plant Disease* 100(11), 2338–2338 (2016)
14. Méndez-Lozano, J., Torres-Pacheco, I., Fauquet, C.M., Rivera-Bustamante, R.F.: Interactions Between Geminiviruses in a Naturally Occurring Mixture: Pepper huasteco virus and Pepper golden mosaic virus. *Phytopathology* 93(3), 270–277 (2013)
15. Pawara, P., Okafor, E., Schomaker, L., Wiering, M.: Data augmentation for plant classification. In: *International Conference on Advanced Concepts for Intelligent Vision Systems*, pp. 615–626, Springer (2017)
16. Reyes, A.K., Caicedo, J.C., Camargo, J.E.: Fine-tuning Deep Convolutional Networks for Plant Recognition. *CLEF Working Notes* (2015)
17. Russakovsky, O., Deng, J., Su, H., Krause, j., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision* 115(3), 211–252 (2015)
18. Secretaría de Agricultura y Desarrollo Rural: <https://www.gob.mx/sader/prensa/produccion-nacional-de-chile-alcanza-2-3-millones-de-toneladas>. Last accessed April 1, 2019
19. Sfar, A.R., Nozha, B., Donald G.: Confidence Sets for Fine-Grained Categorization and Plant Species Identification. *International Journal of Computer Vision* 111(3), 255–275 (2015)
20. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
21. Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., Stefanovic, D.: Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification. *Computational Intelligence and Neuroscience* (2016)
22. Sulc, M., Matas, J.: Texture-Based Leaf Identification. In: Agapito L., Bronstein M., Rother C. (eds.) *Computer Vision - ECCV 2014 Workshops*, LNCS, vol. 8928, pp. 185–200, Springer, Heidelberg. https://doi.org/10.1007/978-3-319-16220-1_14 (2015)

23. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston Ma, USA (2015)
24. Toma, A., Stefan, L.D., Ionescu, B.: UPB HES SO @ PlantCLEF 2017: Automatic Plant Image Identification using Transfer Learning via Convolutional Neural Networks. Working Notes of CLEF (2017)
25. Wäldchen, J., Rzanny, M., Seeland, M., Mäder, P.: Automated plant species identification—Trends and future directions. PLoS computational biology 14(4), 50–56 (2018)
26. Zhao, C., Chan, S.F., Cham, W.K., Chu, L.M.: Plant identification using leaf shapes—A pattern counting approach. Pattern Recognition 48(10), 3203–3215 (2015)
27. Zhao, Z.Q., Ma, L.H., Cheung, Y.M., Wu, X., Tang, Y., Chen, C.L.P.: ApLeaf: An efficient android-based plant leaf identification systems. Neurocomputing 151, 1112–1119 (2015)