Clasificador de objetos en MATLAB® con redes neuronales de aprendizaje profundo

Allison Guzmán Lembo, Carlos Daniel Mayorga Alvarado, Jimena Fernanda Dávila Vázquez, Jonathan Martínez Reyna, Angel Rodriguez-Liñan, Luis M. Torres-Treviño
Universidad Autónoma de Nuevo León,
Facultad de Ingeniería Mecánica y Eléctrica
angel.rodriguezln@uanl.edu.mx,

RESUMEN

En este trabajo, de manera introductoria se ilustra la implementación de tres redes neuronales preentrenadas con el paradigma de aprendizaje profundo en el software MATLAB®, que pueden reconocer objetos en imágenes capturadas por una cámara. Mediante experimentos para reconocer objetos, se determinó cuál de estas redes tuvo mejor desempeño, aprovechando una base de datos estándar de imágenes. Dichos resultados se ilustran con ejemplos del uso del software y con datos comparativos de los aciertos.

PALABRAS CLAVE

Red neuronal artificial, aprendizaje profundo, AlexNet, GoogLeNet, VGG-16, reconocimiento de imágenes.

ABSTRACT

In this work, an introduction to the implementation of three pre-trained neural networks with Deep Learning is illustrated in MATLAB®, for recognition of objects from images acquired by a camera. Through experiments to recognize objects, it was determined which of these networks had better performance, taking advantage of a standard database of images. These results are illustrated with examples with the software and with comparative data.

KEYWORDS

Artificial neural network, deep learning, AlexNet, GoogLeNet, VGG-16, image recognition.

INTRODUCCIÓN

Uno de los principales objetivos y preocupaciones de la humanidad a lo largo de la historia, ha sido el poder diseñar y construir máquinas capaces de realizar diversas actividades con cierta autonomía. De los avances alcanzados en este sentido, se han llegado a definir métodos para diseñar sistemas, que tomen decisiones o realicen acciones (por ejemplo, encender/apagar iluminación, control automático de válvulas en la industria, o el reconocer un objeto en una imagen), de acuerdo con los requerimientos de una actividad. En general este proceso de acciones puede verse como un sistema que procesa variables de entrada, para generar variables de salida. [1]

Este proceso para toma de decisiones, acciones de control o procesamiento de la información, dependiendo del caso, puede realizarse con diversas herramientas como algoritmos de procesamiento de señales y datos, esquemas convencionales de control automático, y técnicas de inteligencia artificial (como las redes neuronales artificiales), entre otras. Las redes neuronales artificiales tienen la ventaja de ofrecer posibles soluciones a problemas que no pueden ser resueltos por esquemas de procesamiento o de control convencional, ante una gran cantidad de variables de entrada o de la complejidad del problema. Las redes neuronales han sido un importante avance en la inteligencia artificial y se han hecho cada vez más accesibles al público en general, ya que tienen aplicaciones en videojuegos, asistentes virtuales, servicios financieros, agentes autónomos, entre otros.

Por otro lado, para reconocimiento de una imagen se identifican varias etapas, que consisten en la adquisición de la imagen, un preprocesamiento de la imagen, extracción de características y finalmente se comparan sus características con las de imágenes conocidas, logrando entonces el reconocimiento de un objeto. [1]

Particularmente, las redes neuronales son unas de las herramientas que han tenido más avance y desarrollo para ejecutar estas etapas, y por fin, reconocer imágenes. Por ejemplo, gracias a ello se ha logrado una mejor precisión para detección en sistemas de vigilancia, mejora en entrenamientos basados en imágenes y video, o para diagnóstico de tumores. [2, 3]

Considerando la literatura científica en esta área, en el trabajo de Coronel Tobar [1] se utilizó una red neuronal con una capa de entrada de 10304 nodos. Las ventajas de este trabajo es su fácil implementación que no requiere cálculos complejos y que las variaciones de iluminación no son muy significativas, ya que su procesamiento está en términos de los niveles de gris de la imagen original.

Otro sistema que permite reconocer diferentes figuras geométricas adquiere imágenes a color con una cámara web con una resolución de 480 filas por 640 columnas, y luego genera una imagen en blanco y negro de 340 píxeles4. La red de clasificación de color tiene 3 entradas: la componente roja, la componente azul y la componente verde, todas con un rango de 0 a 255. El número total de capas de la red es 4, una capa de entrada que tiene tres neuronas, dos capas ocultas y una capa de salida lineal que tiene 2 neuronas. La función de activación para las 2 capas ocultas es la función tangente sigmoidea. [4]

Otro trabajo, consiste en el desarrollo de un sistema de reconocimiento de imágenes con dispositivos móviles, el cual debe ser capaz de funcionar utilizando una función de GaussianBlur, para el algoritmo de suavizado en reconocimiento de hojas de plantas, ya que tiene en cuenta el peso de los píxeles más cercanos que los alejados. Utiliza una base de datos formada por 25 imágenes de 12 tipos de plantas distintas y un tipo de red neuronal de retro propagación, con arquitectura de 7 neuronas de entrada y 2 neuronas en la capa oculta. [5]

Entre las ventajas de las redes de retro propagación, es que aprovechan la naturaleza paralela de las redes neuronales para reducir el tiempo requerido por un procesador secuencial para determinar la correspondencia entre unos patrones dados.

En el presente trabajo, se introduce al uso de un toolbox para reconocimiento de objetos a partir de su imagen mediante una red neuronal. La red neuronal debe ser primeramente entrenada con un conjunto de imágenes y posteriormente se prueba

la eficiencia del sistema, utilizando una base de datos de imágenes estándar. Además, se calcula el porcentaje de aciertos del sistema de reconocimiento.

El resto del documento se organiza de la siguiente manera: Primero se explican los conceptos principales de las redes neuronales, luego se explica la metodología que se utiliza para la implementación y ajuste de las redes neuronales preentrenadas en MATLAB®, posteriormente se explica la base de datos que se utilizó para entrenamiento y pruebas. Se ilustran los resultados del reconocimiento con tablas comparativas. Finalmente, se presentan las conclusiones obtenidas.

REDES NEURONALES

Una neurona artificial posee diversas entradas ponderadas, un bloque sumador, una función de activación y su respectiva salida. [4] En la figura 1 se muestra la estructura de una neurona artificial.

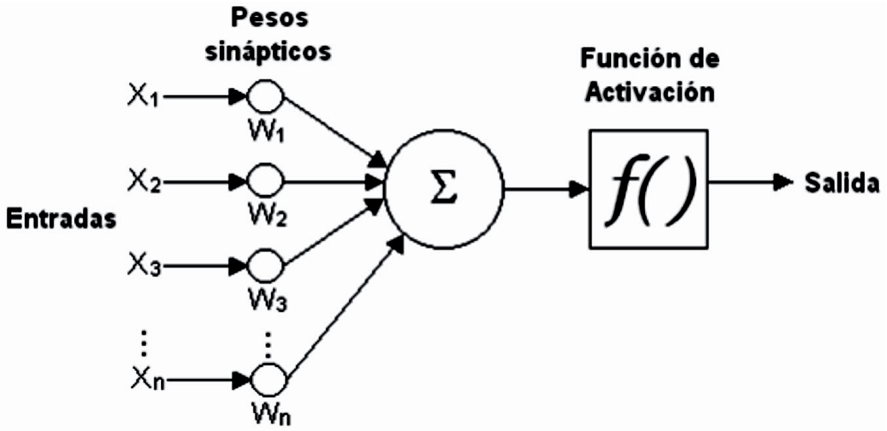

Fig. 1. Estructura de una neurona artificial [imagen obtenida de [4]].

Como se ilustra en la figura 2, las redes neuronales están formadas por un vector o capa de entrada, capas ocultas y capa de salida. El vector de entrada contiene las señales provenientes de las fuentes externas; las capas ocultas se encuentran entre el vector de entrada y la capa de salida, el número de capas ocultas puede ser desde cero hasta un número elevado; la capa de salida es la que transmite la respuesta de la red al medio externo. [5]

En una red de retro propagación, el error se propaga de manera inversa al funcionamiento normal de la red, a esto también se le llama método del gradiente descendiente. A continuación, se describe el algoritmo de propagación hacia atrás o regla delta generalizada.[6]

1. Se inicializan los valores de los pesos sinápticos de toda la red; por lo general se asignan valores aleatorios.
2. Se ingresa un ejemplo de entrada para entrenamiento de la red (XP= [XP1, …, XPN]), junto con su salida esperada (YP = [YP1…YPM]).
3. Se calcula la entrada total de cada neurona en cada capa oculta, mediante la ecuación
   $entrada_{pj}^{h} = \sum_{i=1}^{N} 1\, W_{pj}^{h} * x_{pi} + w_{j^0}^{h}.$

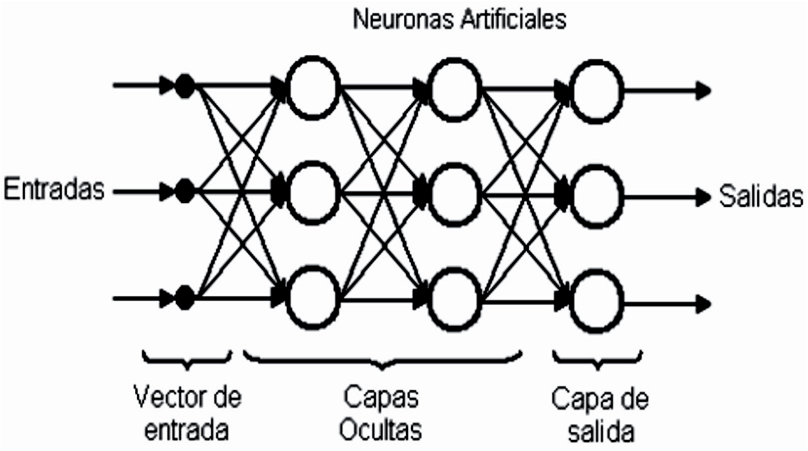

Fig. 2. Esquema de red neuronal artificial [imagen obtenida de [5]].

4. Se calculan las salidas de la capa oculta, mediante
   $i_{Pj} = f^h(entrada^h_{pj})$
5. Se calcula la entrada total de cada neurona de la capa de salida, con
   $$entrada^0_{pk} = \sum_{J=1}^{L} 1\, W^0_{kj} i_{pj} + w^0_{k^0}$$
6. Se calculan las salidas de la red:
   $O_{Pk} = f^0(entrada^0_{pk})$
7. Se calculan los términos de error para las neuronas de la capa de salida:
   $\delta^0_{pk} = (y_{Pk} - 0_{Pk}) * f^0(entrada^0_{pk})$
8. Se calculan los términos de error para las neuronas de la capa oculta:
   $$\delta^h_{pj} = f^h\left(entrada^h_{pj}\right) * \sum_{K=1}^{M} 1\, \delta^0_{pk} * w^0_{kj}$$
9. Se actualizan los pesos sinápticos de las neuronas de la capa de salida, como
   $$w^0_{kjnuevo} = w^0_{kjactual} + \eta * \delta^0_{pk} * i_{pj}$$
10. Se actualizan los pesos sinápticos de la capa oculta, con
    $$w^h_{jinuevo} = w^h_{jiactual} + \eta * \delta^h_{pj} * x_{pi}$$
11. Si no existen más ejemplos de entrenamiento, se calcula el error cuadrático medio de todos los ejemplos, mediante la ecuación
    $\text{Error} = \sum_p \frac{1}{2} \sum_{K=1}^{M} ((y_{Pk} - 0_{Pk}))^2$.

Aprendizaje profundo (*Deep Learning*)

Una red neuronal de aprendizaje profundo consiste en unir varias capas de neuronas; la diferencia con las redes neuronales tradicionales radica en la forma de conectarse entre las capas y en la función de activación. Las capas

internas tienen la función de crear atributos de manera automática, una tarea que usualmente se realizaba 'a mano'. Esta característica es lo que ha impulsado el uso de estas redes de manera masiva, sobre todo para el procesamiento de imágenes.

Las formas de conectividad entre capas son básicamente de dos tipos: (a) convolución (conv), (b) conexión total, y (c) agrupación (Pooling). La conectividad de convolución realiza una lectura focalizada llamada kernel en un área específica de la capa de neuronas, que se desplaza por todas las neuronas de la capa. Todas las señales del kernel son entradas de una neurona artificial, que se ponderan con pesos ajustables y se aplica la función de activación muy fácil de implementar llamada ReLu, además de las funciones tangenciales, Sigmoidales o Gaussianas. En la conexión total, cada entrada se pondera y se le aplica una función de activación para enviar la señal de salida a la siguiente capa, ésta es la forma tradicional de conectar las redes neuronales entre capas. En la operación de agrupamiento se reduce el número de neuronas requeridas en la siguiente capa, aplicando una función de agrupamiento (Pool) o de máximo (softmax) al kernel.

El aprendizaje profundo es una tecnología clave en los vehículos autónomos, que les permite distinguir entre un señalamiento de 'Alto', un peatón o un semáforo. Con el Aprendizaje Profundo, un modelo informático aprende a realizar tareas de clasificación directamente a partir de imágenes, texto o sonido. Los modelos de aprendizaje profundo pueden alcanzar una precisión de vanguardia que, en ocasiones, supera el desempeño humano. Los modelos se entrenan mediante un amplio conjunto de datos etiquetados y arquitecturas de redes neuronales que contienen muchas capas.

En algunos paquetes de software, existen herramientas o *Toolboxes* que facilitan a los usuarios programar y ajustar algoritmos de redes neuronales, como en MATLAB® algunos son los *Toolboxes* AlexNet, [7, 8], GoogLeNet, [9, 10] y VGG-16. [11]

AlexNet [7, 8] es una red neuronal convolucional que se entrena con más de un millón de imágenes. La red tiene 8 capas de profundidad y puede clasificar imágenes en 1000 categorías de objetos, como teclado, *mouse*, lápiz y muchos animales. Como resultado, la red ha aprendido ricas representaciones de características para una amplia gama de imágenes. La red tiene un tamaño de entrada de imagen de 227 por 227 píxeles. La red toma una imagen como entrada y genera una etiqueta para el objeto en la imagen junto con las probabilidades para cada una de las categorías de objetos. El aprendizaje por transferencia se usa comúnmente en aplicaciones de aprendizaje profundo. Puede tomar una red pre-entrenada y usarla como punto de partida para aprender una nueva tarea. Para un análisis más profundo de la red neuronal, MATLAB® pone a su disposición una vista que permite observar el comportamiento de toda la red, donde muestra las entradas, pesos, funciones de activación, convolución, secuencia, agrupación, normalización, utilidad y salidas, como se ilustra en las figuras 3 y 4.

GoogLeNet [9, 10] es una red neuronal circumvolucional preentrenada de 22 capas profundas, ha sido entrenada sobre 1 millón de imágenes y puede clasificar imágenes en categorías de 1000 objetos (tales como teclado, taza de café,

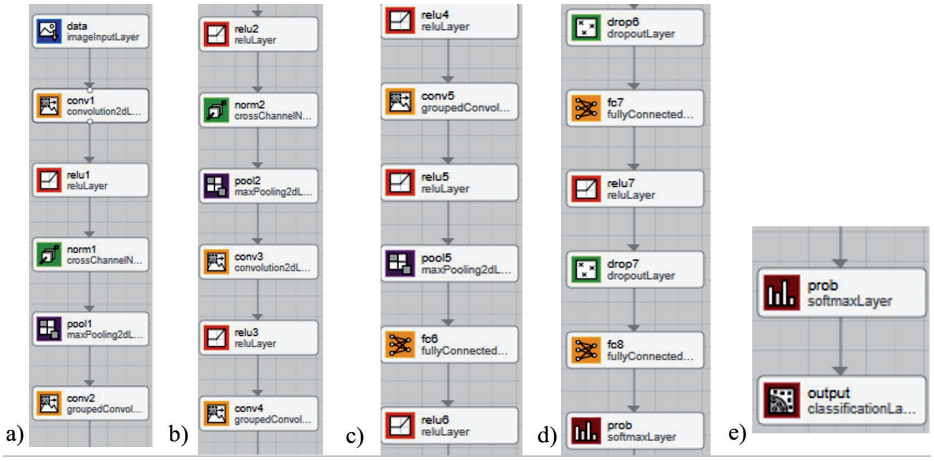

Fig. 3. Red Neuronal Alex Net en MATLAB®.

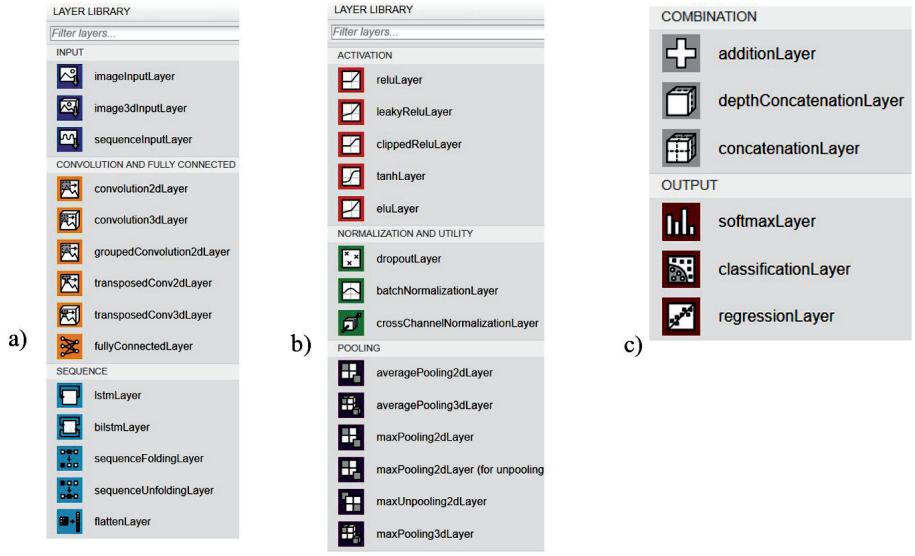

Fig. 4. Capas que Componen la Red Neuronal AlexNet.

lápiz y muchos animales). La red ha aprendido abundantes representaciones para una amplia gama de imágenes. La red tiene una imagen como entrada y genera la salida como una etiqueta o nombre del objeto de la imagen, junto con las probabilidades para cada una de las categorías de objeto. Su arquitectura consistía en 22 capas de profundidad, pero redujo el número de parámetros de 60 millones (en AlexNet) a 4 millones. MATLAB® permite observar completamente la red y el cómo está constituida. Tiene las entradas, salidas y pesos, funciones de activación, secuencia, agrupación, etcétera. Se pueden modificar los valores preestablecidos de las funciones de activación, pesos, convolución, secuencia, agrupación, normalización y utilidad, que se muestran las figuras 5 y 6.

VGG-168, [11] es una red neuronal convolucional que está entrenada por más de 1 millón de imágenes. La red es de 16 capas de profundidad y puede clasificar

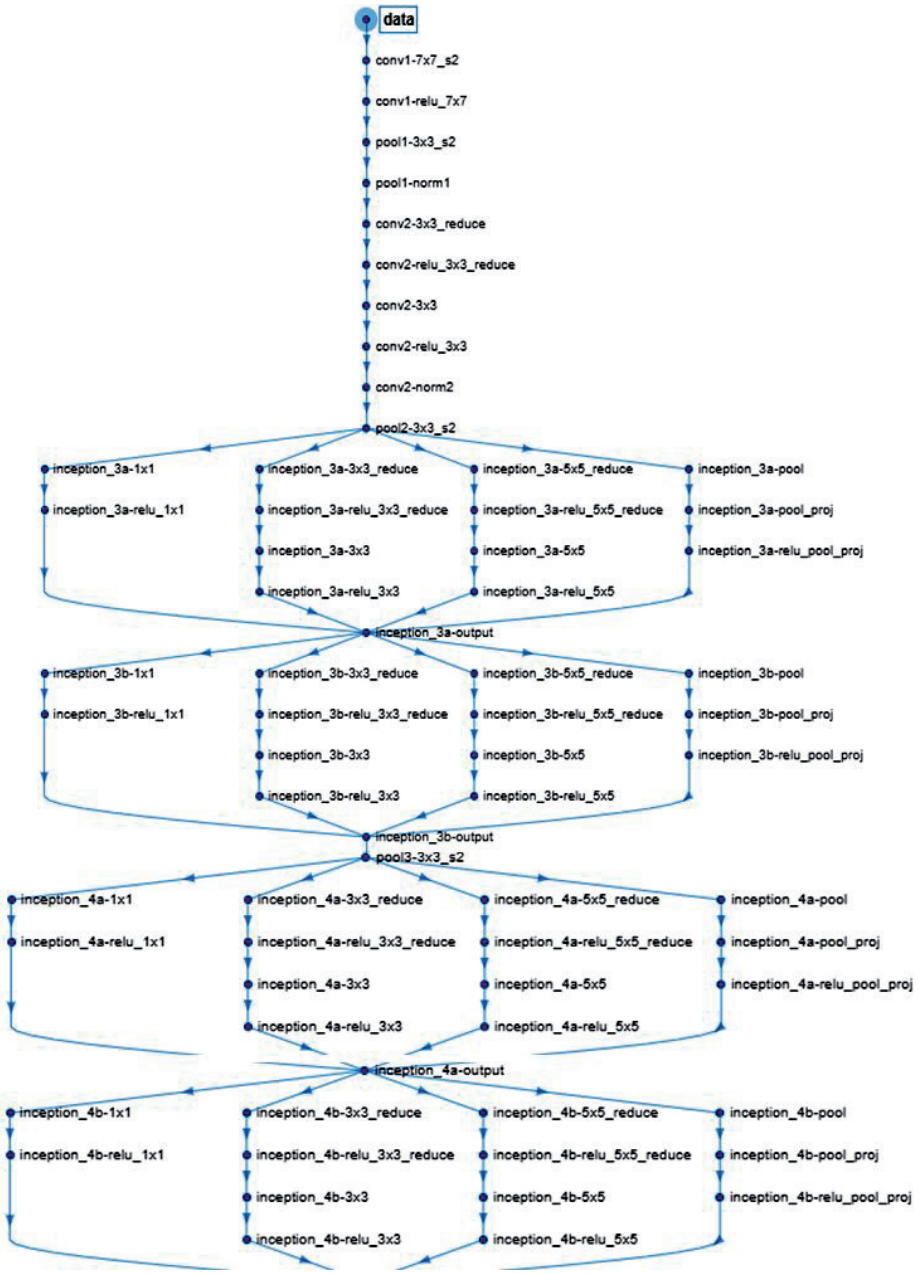

Fig. 5. Red neuronal GoogLeNet en MATLAB® (parte 1).

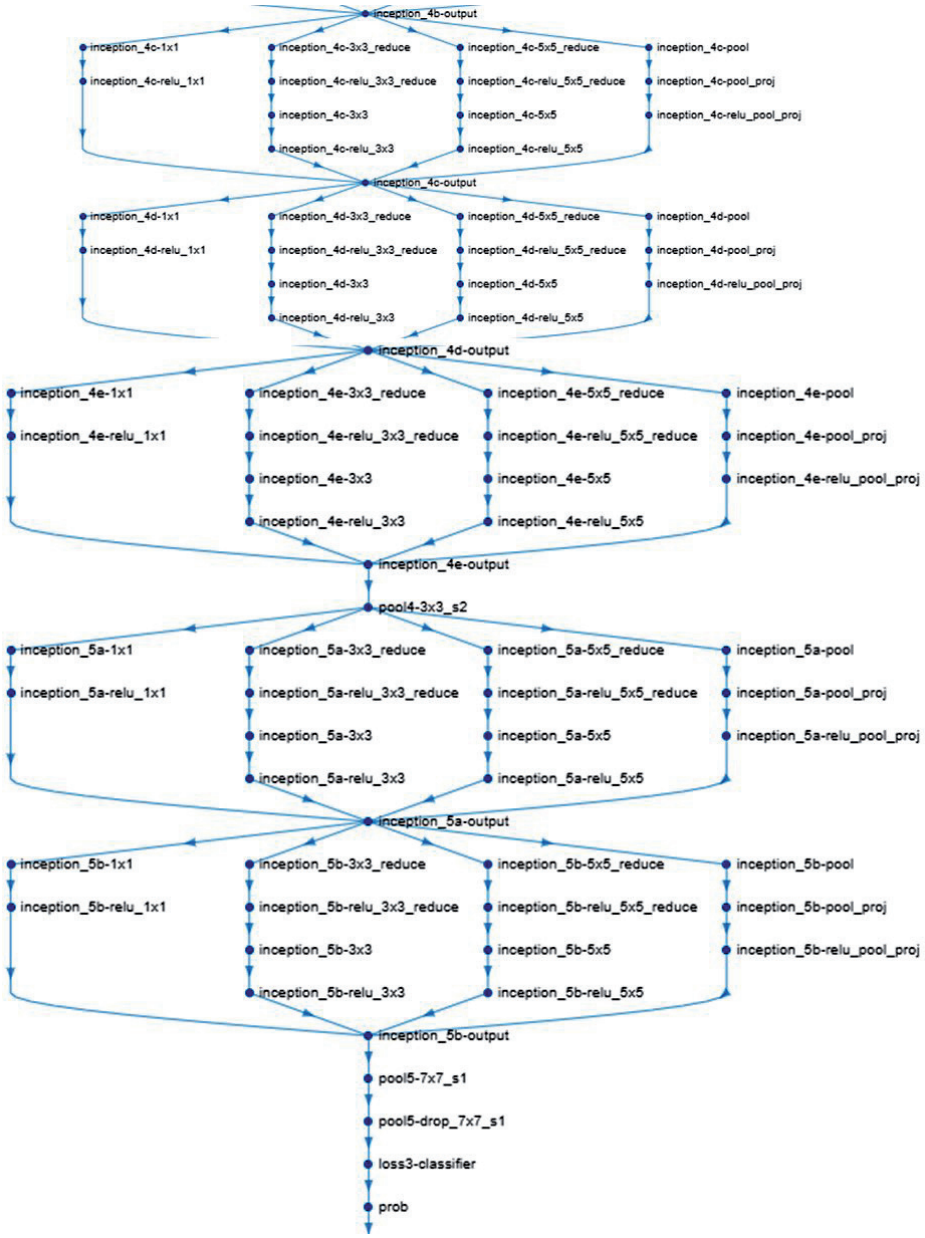

Fig. 6. Red neuronal GoogLeNet en MATLAB® (parte 2).

imágenes en categorías de 1000 objetos. Como resultado, la red ha aprendido representaciones abundantes de una amplia gama de imágenes. La red tiene un tamaño de imagen la entrada de 224 de 224 píxeles. Es muy atractiva debido a su muy uniforme arquitectura. Similar a AlexNet, sólo maneja convoluciones 3x3, pero muchos filtros. Actualmente es la opción más preferida en la comunidad para extraer características de las imágenes. Sin embargo, VGG-16 consta de 138 millones de parámetros, lo que puede ser un poco difícil de manejar. Para un mayor análisis de la red neuronal, MATLAB® pone a su disposición una vista que permite observar el comportamiento de toda la red, y permite modificar los valores preestablecidos de las funciones de activación, pesos, convolución, secuencia, agrupación, normalización y utilidad. En la figura 7 se muestra la vista de la red neuronal VGG-16 en MATLAB®.

En el trabajo de Siddharth Das, [12] se determinó que el mayor porcentaje de acierto de estas 3 redes neuronales preentranadas es la VGG-16, con un 73.8%. Mientras que GoogLeNet resultó con un 68.8% y AlexNet con un 52.2%.

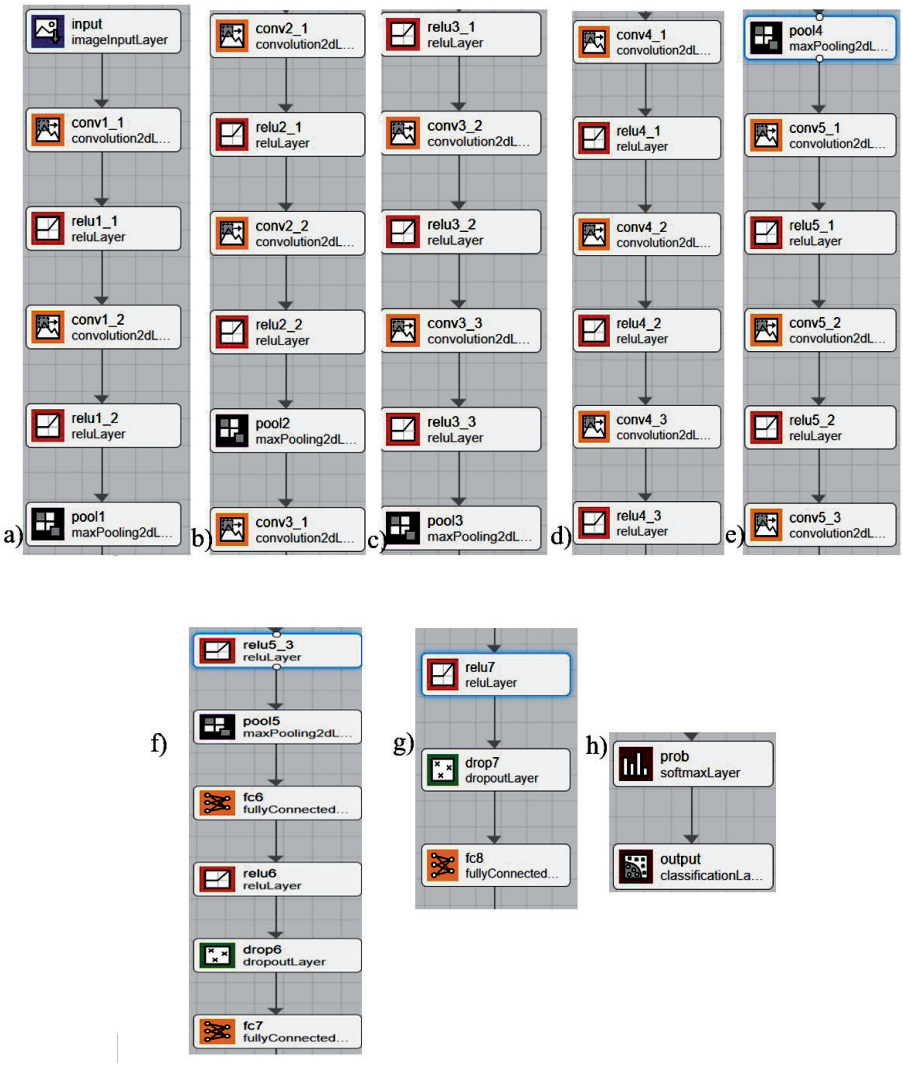

Fig. 7. De a) a h) se muestran las partes de la red neuronal VGG-16 en MATLAB®.

Uso del toolbox de MATLAB®

Luego de descargar estos 3 toolbooxes de la página de MathWorks [9, 11, 13] (sólo es posible en las versiones MATLAB® R2018a en delante), los pasos que se requieren para utilizarlos, a partir de la captura de imágenes con una cámara web, son los siguientes:

1. Se activa la cámara web mediante el comando.
   camera = webcam
2. Para elegir la red preentrenada, se usa alguno de los 3 siguientes comandos:
   net = googlenet
   net = alexnet
   net = vgg16
3. Se obtiene el tamaño de la imagen:
   inputSize = net.Layers(1).InputSize(1:2)
4. Para hacer la captura de una imagen con la cámara web, se usan los comandos:
   figure;
   im = snapshot(camera);
5. Para la clasificación de objeto se necesita los comandos:
   image(im);
   im= imresize(im,inputSize);
   [label,score] = classify(net,im);
6. La etiqueta del objeto reconocido y el porcentaje de acierto se pueden escribir como título en la imagen:
   title({char(label),num2str(max(score),2)});
   title({char(label),num2str(max(score),2)});

Con estos comandos, el resultado del reconocimiento se mostrará mediante una leyenda con la clase de objeto reconocido, con el por ciento de acierto y mostrando la captura de la imagen.

RESULTADOS DEL RECONOCIMIENTO DE OBJETOS MEDIANTE LAS REDES NEURONALES

A continuación, se ilustra el desempeño de los toolboxes explicados (AlexNet, GoogLeNet y VGG-16), para reconocimiento de objetos. Para ello se ejecutó un programa con los comandos de la sección anterior, colocando 50 objetos distintos ante la cámara web para su reconocimiento.

Las figuras 8a, 8b, 8c y 8d muestran los resultados al intentar reconocer algunos objetos ante la cámara web utilizando el toolbox de Alex Net.

Las figuras 9a, 9b, 9c y 9d muestran los resultados al intentar reconocer algunos objetos ante la cámara web utilizando el toolbox de GoogLeNet.

Las figuras 10a, 10b, 10c y 10d muestran los resultados al intentar reconocer algunos objetos ante la cámara web utilizando el toolbox de VGG-16.

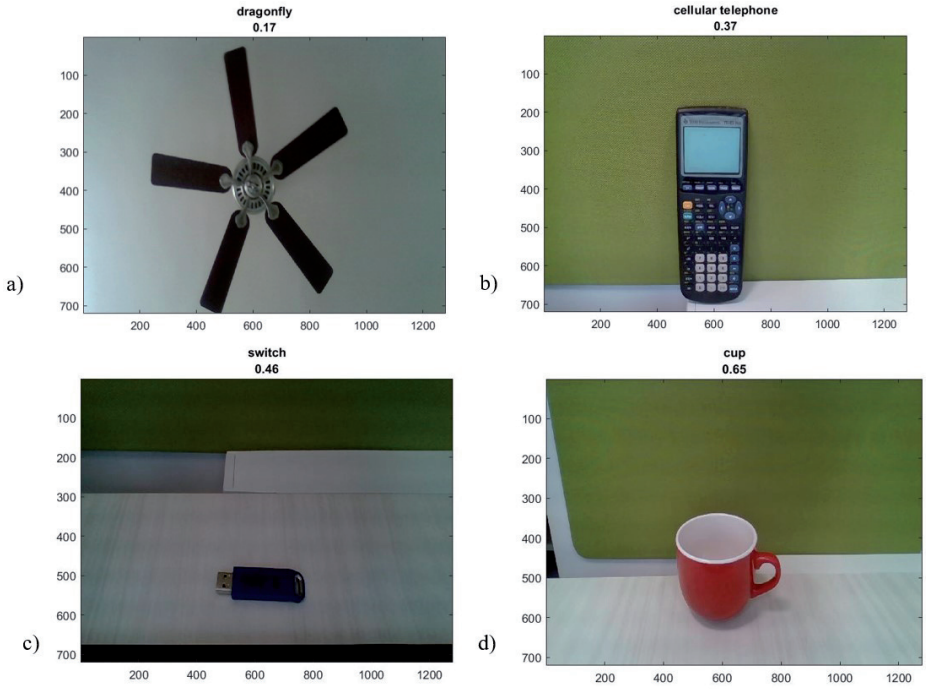

Fig. 8. Resultados de reconocimiento usando Alex Net.

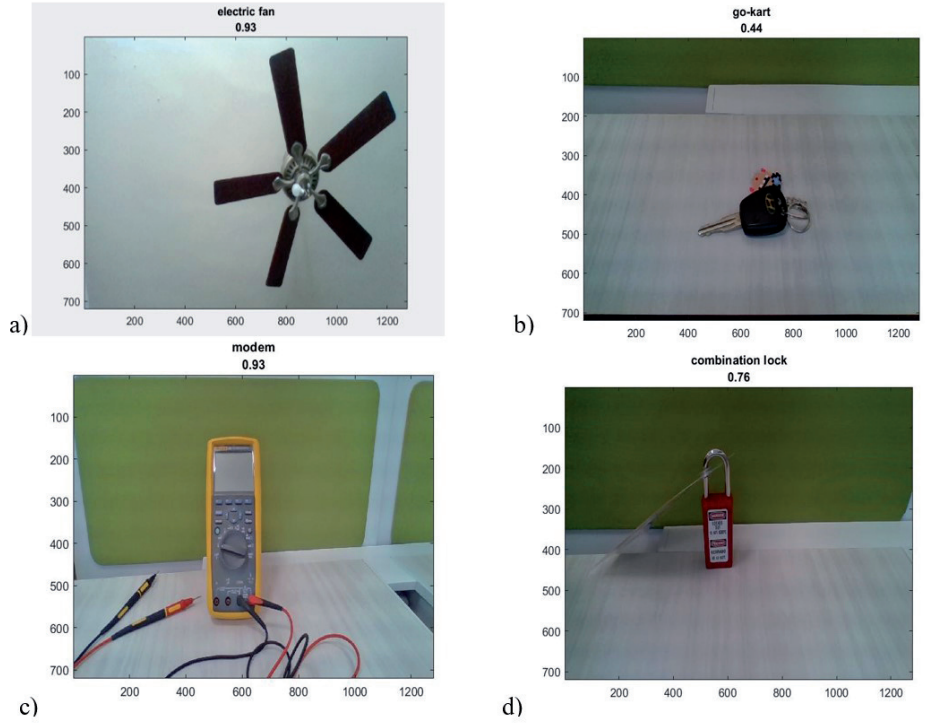

Fig. 9. Resultados de reconocimiento usando GoogLeNet.

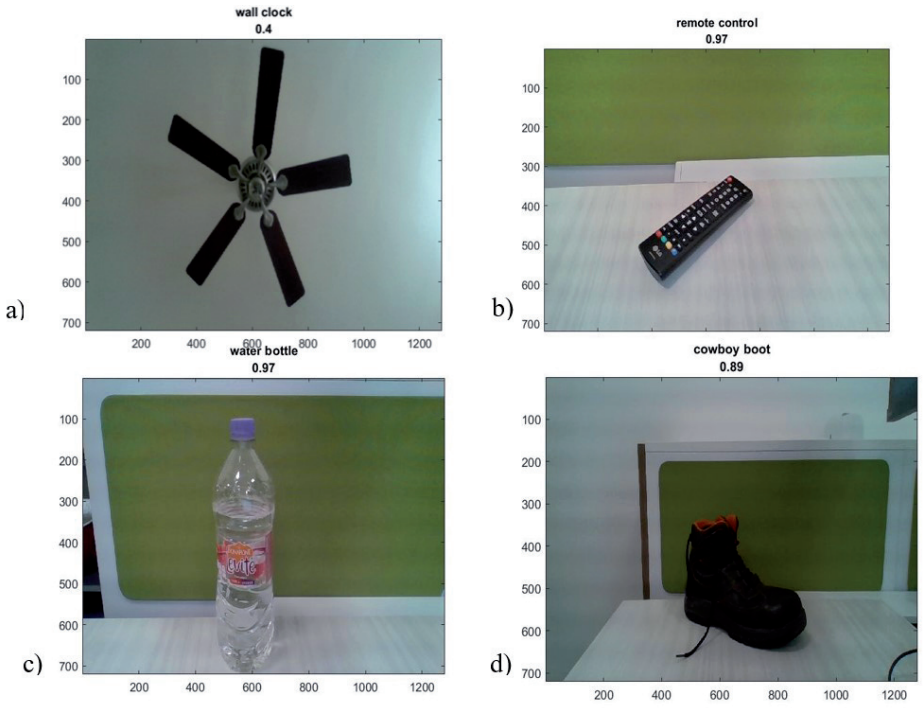

Fig. 10. Resultados de reconocimiento usando VGG-16.

Posteriormente, pueden modificarse los parámetros en los bloques de las figuras 3, 4 y 7 de las redes AlexNet y VGG-16, con el propósito de mejorar el desempeño de la red para el reconocimiento. Las variables que se pueden modificar son las que están marcadas como 'fc' (como se muestra en la figura 11), las cuales indican las conexiones que se hacen de las capas anteriores.

En el caso de las figuras 11 y 12, se pueden modificar los valores que están encerrados en color rojo. Sin embargo, estas redes de aprendizaje profundo ya están preentrenadas para un alto desempeño, por lo que ante algunas modificaciones arbitrarias de los parámetros podría disminuir el porcentaje de acierto.

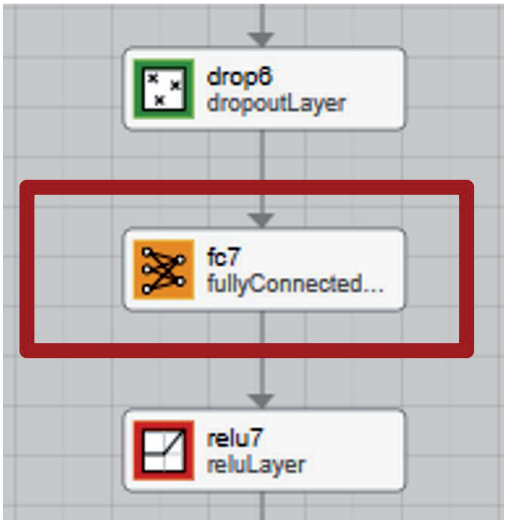

Fig. 11. Capa de Conexiones.

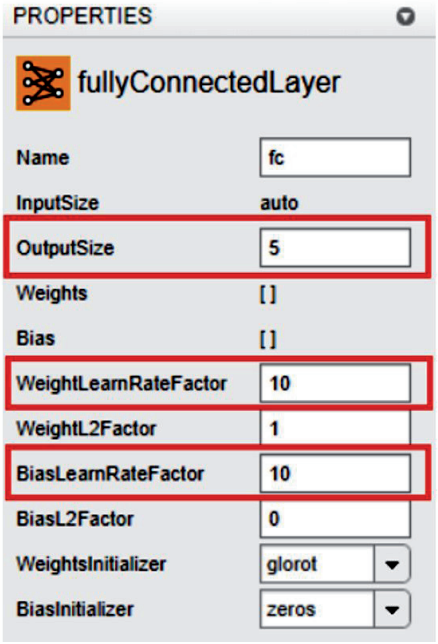

Fig. 12. Propiedades de la capa de conexiones.

Ante las pruebas realizadas con 50 imágenes, el porcentaje de aciertos con el toolbox AlexNet con sus parámetros por defecto fue del 48%, es decir, acertando en reconocer 24 de los 50 objetos.

El porcentaje de aciertos con el toolbox GoogLeNet con sus parámetros por defecto fue del 66%, es decir, acertando en reconocer 33 de los 50 objetos.

El porcentaje de aciertos con el toolbox VGG-16 con sus parámetros por defecto fue del 74%, es decir, acertando en reconocer 37 de los 50 objetos. Estos resultados se resumen en la figura 13.

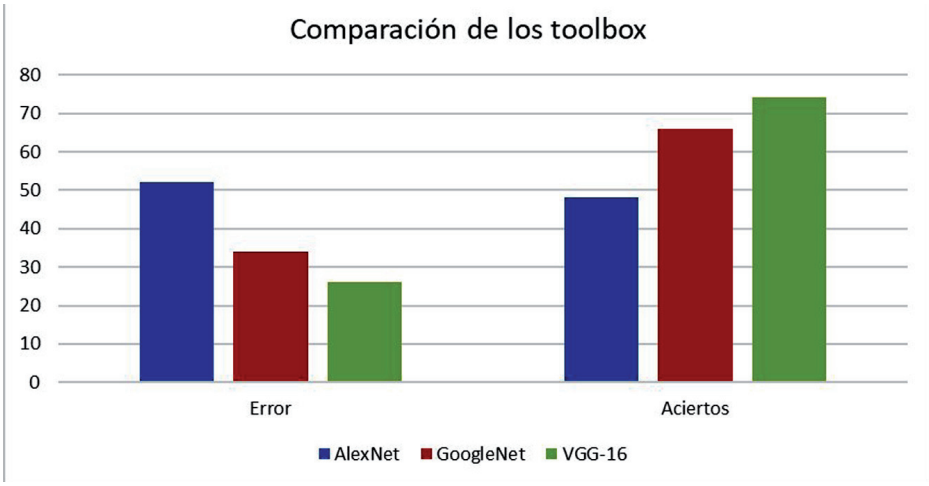

Fig. 13. Comparativa de porcentajes de acierto y error de cada red neuronal en las pruebas experimentales.

CONCLUSIONES

De los resultados de implementación y prueba de los tres toolboxes de redes neuronales, se concluye que la más amigable para utilizar y que cuenta con mayor porcentaje de aciertos es la VGG-16.

REFERENCIAS

1. Coronel Tobar, Hernán Fabricio. (2007). Reconocimiento de rostros utilizando redes neuronales. Tesis, Escuela Politécnica Nacional, Quito, Ecuador.
2. Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., & Thrun, S. (2017). Dermatologist-level classification of skin cancer with deep neural networks. Nature, 542(7639), 115–118.
3. Yun Liu, Krishna Gadepalli, Mohammad Norouzi, George E. Dahl, Timo Kohlberger, Aleksey Boyko, Subhashini Venugopalan, Aleksei Timofeev, Philip Q. Nelson, Gregory S. Corrado, Jason D. Hipp, Lily Peng, Martin C. (2017). Detecting Cancer Metastases on Gigapixel Pathology Images. CoRR abs/1703.02442.
4. Ramírez González, D., Pulido Sarmiento, G., Gerardino Arévalo, B., Cruz Romero, J., Estupiñán Escalante, E., & Cancino Suárez, S. (2009). Adquisición y reconocimiento de imágenes por medio de técnicas de visión e inteligencia artificial. ITECKNE, 6(1), 5-13.
5. García García, Pedro Pablo. (2013). Reconocimiento de imágenes utilizando redes neuronales artificiales. Tesis de Maestría, Universidad Complutense de Madrid, España.
6. Castro García, José Francisco. (2006). Fundamentos para la implementación de red neuronal perceptrón multicapa mediante software. Tesis, Universidad de San Carlos, Guatemala.
7. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks." Advances in neural information processing systems. 2012.
8. Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei. (2015). ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision, 115(3), 211-252.
9. Pretrained GoogLeNet convolutional neural network. http://la.mathworks.com/help/deeplearning/ref/googlenet.html
10. Zhou, B., Khosla, A., Lapedriza, À., Torralba, A., & Oliva, A. (2017). Places: An Image Database for Deep Scene Understanding. CoRR, abs/1610.02055.
11. Pretrained VGG-16 convolutional neural network. http://la.mathworks.com/help/deeplearning/ref/vgg16.html
12. Siddharth, Das. (2017) CNN Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet.
13. Pretrained AlexNet convolutional neural network. http://la.mathworks.com/help/deeplearning/ref/alexnet.html