Congr. Int. en Ing. Electrónica. Mem. ELECTRO, Vol. 41, pp. 125-129, Oct 2019, Chihuahua, Chih. México
http://electro.itchihuahua.edu.mx/memorias_electro/MemoriaElectro2019.zip
ISSN 1405-2172

CLASIFICACIÓN DE FLORES EN IMÁGENES DIGITALES UTILIZANDO CNNs

Eduardo Diaz-Gaxiola, Zuriel E Morales-Casas, Jose A Berger-Castro[1], Arturo Yee-Rendon, Ines F Vega-Lopez.
Universidad Autónoma de Sinaloa
Ciudad Universitaria, 80040 Culiacán, Sinaloa.
{eduardogaxiola, zurielernesto, arturo.yee, ifvega}@uas.edu.mx, ja.berger15@info.uas.edu.mx[1]

RESUMEN.

En este trabajo, presentamos un estudio comparativo de diferentes técnicas de aprendizaje profundo (*deep learning*) para la clasificación de flores de plantas usando imágenes digitales. Las arquitecturas de redes neuronales convolucionales (CNNs por sus siglas en inglés) *DenseNet* e *InceptionResNet* fueron utilizadas para construir modelos predictivos para la clasificación de 102 especies de flores de plantas del conjunto de datos (*dataset*) de flores de *Visual Group Geometry* (VGG). Los resultados obtenidos muestran que la arquitectura *InceptionResNet* alcanzó una precisión de 72.65% en top 1, 86.47% en top 3 y 90.59% en top 5.

Palabras Clave: Redes neuronales convolucionales (CNNs), DenseNet, InceptionResNet, flores.

ABSTRACT.

In this paper, we present a comparative study of different Deep Learning technics for plant flower species identification from digital images. Convolutional neural networks architectures (CNNs) *DenseNet* and *InceptionResNet* was used to develop predictive models for the classification of 102 different flower species of the dataset of *Visual Group Geometry* (VGG). The paper aims to realize a study of different CNNs in plant flower species identification. The results obtained show *InceptionResNet* architecture an accuracy of 72.65% in top 1, 86.47% in top 3 and 90.59% in top 5.

Keywords: Convolutional neural networks (CNNs), DenseNet, InceptionResNet, flowers.

1. INTRODUCCIÓN

La clasificación de objetos en imágenes digitales es un problema que genera un gran interés en el área de visión por computadora. Por ejemplo, el poder clasificar diferentes tipos de autos que pasan por una calle, diferentes tipos de suelos en imágenes satelitales o diferentes tipos de especies de plantas.

El poder clasificar diferentes flores de plantas implica un análisis extenso de diferentes factores, por ejemplo, el color o el tamaño de los pétalos de la flor. Distinguir un girasol de una rosa a simple vista es algo sencillo, ya que las características como el color y la forma de ambas flores son muy distintas, pero no en todas las especies las diferencias son tan notorias. El problema es que las similitudes entre especies son tan sutiles que son difíciles de diferenciar a simple vista, por ejemplo, si los bordes de los pétalos terminan en esquina o curvatura, o si las hojas son más delgadas en una especie respecto a otra.

Figura 1. Imágenes de especies de flores del conjunto de VGG, a la izquierda la especie *barbeton daisy* y a la derecha la especie *common dandelion*.

Como podemos observar en la Figura 1, para alguien que no sea experto en el área de la biología puede clasificar ambas flores en una misma especie, pero no lo son. Algunos factores distintivos son: el centro de la flor, el grosor de los pétalos, el distinto tono de amarillo, etc.

Para poder realizar una clasificación correcta entre especies de plantas, necesitamos la ayuda de un taxónomo experto, una guía de flores detallada o un sitio donde especifiquemos las características de las flores que queremos reconocer.

Una manera de dar solución al problema de clasificación de especies de flores de plantas es mediante el uso de diferentes técnicas de aprendizaje máquina (*machine learning*). Para poder utilizar estas técnicas, es necesario tener bien definidos y estructurados los atributos distintivos que ayuden a clasificar las diferentes especies de flores de plantas (tamaño de la flor, tamaño de la hoja, color, etc.) [1, 2]. Las técnicas tradicionales de aprendizaje máquina obtienen buenos resultados cuando la caracterización de estos atributos es posible y pueden ser representadas en valores numéricos. La situación se complica cuando estos atributos son difíciles de caracterizar.

El uso de técnicas de aprendizaje profundo ayuda a resolver la caracterización de los atributos de las diferentes especies de flores de plantas en imágenes digitales, utilizando algoritmos para caracterizar los atributos más relevantes para el aprendizaje. Las redes neuronales convolucionales (CNNs por sus siglas en inglés) realizan el procesamiento de las imágenes aplicando diferentes filtros, encontrando los patrones distintivos de cada clase [3].

Actualmente, los buenos resultados obtenidos por las CNNs han propiciado el incremento en el uso de estas técnicas en problemas de clasificación de imágenes digitales.
Las CNNs utilizan filtros de convolución para resaltar atributos de los objetos contenidos en imágenes digitales, pasando esta información a las capas siguientes, para obtener las características distintivas.
En este trabajo, presentaremos un estudio comparativo entre las arquitecturas de CNNs *DenseNet* e *InceptionResNet*, para la clasificación de especies de flores en imágenes digitales, mostrando su precisión en top 1, top 3 y top 5. Nos referiremos como top a los mejores *n* valores tomados en cuenta para decidir si se acertó a la predicción, tomando entonces el mejor, los 3 mejores y los 5 mejores valores.

2. TRABAJOS RELACIONADOS

El desafío de reconocimiento visual a gran escala (por sus siglas ILSVRC) es un reto utilizado como punto de comparación para la detección y clasificación de objetos en miles de imágenes [4]. En sus primeros años, los ganadores utilizaron técnicas tradicionales de aprendizaje máquina, siendo Lin et al. [5] los ganadores en el 2010, proponiendo un esquema *Hadoop* para la extracción de características y máquina de soporte vectorial (SVM por sus siglas en inglés) para el entrenamiento. En el 2011, Perronnin et al. [6] utilizaron un *Fisher Kernel* y SVM para el entrenamiento.
Al año siguiente, en el 2012, Krizhevsky et al. [7] propusieron CNNs para clasificar un total de 1000 diferentes clases de objetos dentro de 1.2 millones de imágenes, obteniendo porcentajes de error del 37.5% y 17.0% para top 1 y top 5, respectivamente. A partir de ese año, en el reto ILSVRC ha prevalecido el uso de técnicas de aprendizaje profundo, viéndose rezagadas las técnicas de aprendizaje máquina tradicionales.
S. H. Lee [8] en él 2015, propone un esquema de técnicas no supervisadas utilizando CNNs para extraer las características de 44 diferentes especies de plantas recolectadas en Inglaterra, encontrando que la estructura de la venación en las hojas es una característica relevante. El esquema obtuvo una precisión del 99.5%.

3. PROPUESTAS.

En esta sección se describe las arquitecturas de *DenseNet* e *InceptionResNet*.

3.1. DenseNet.

La arquitectura *DenseNet* fue propuesta en 2017 por investigadores de la Universidad de Cornell, de la Universidad de Tsinghua y del departamento de Inteligencia Artificial de Facebook [9]. Lo que distingue a este tipo de CNN es la forma en la que se conectan sus capas convolucionales.
En una CNN tradicional, los datos de entrada siguen una ruta lineal a través de la red. Los datos de salida de la capa L serán los datos de entrada de la capa siguiente $L + 1$, mientras que los datos de salida de esta a su vez serán los datos de entrada de la capa $L + 2$, y así sucesivamente.
La arquitectura *DenseNet* se conforma principalmente de los llamados bloques densos (*dense blocks*) un ejemplo de estos bloques densos se ilustra en la Figura 2. Cada capa L dentro de estos bloques recibe como entrada los datos de todas las capas que le preceden, haciendo uso de concatenación por canales. De esta forma, cada capa dentro de un bloque denso tiene acceso a toda la información de las capas anteriores, que le permite reutilizar características previas según sea necesario.

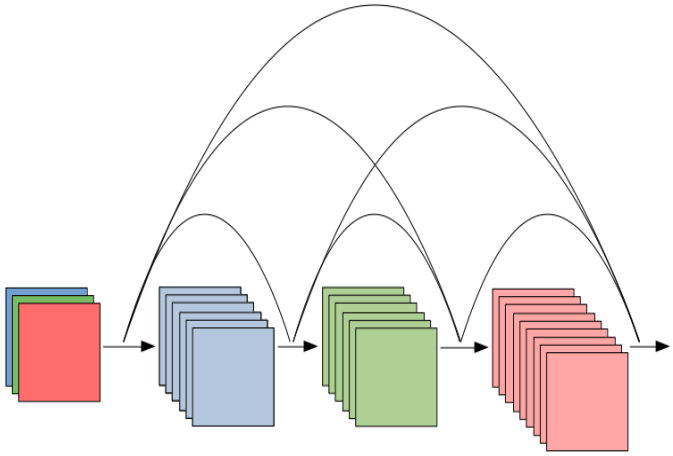

Figura 2. Bloque denso de cinco capas.

3.2. Inception.

Google presentó la primera versión de la arquitectura *Inception* en 2015, bajo el nombre de GoogLeNet [10]. En su trabajo, los investigadores señalan el problema de seleccionar un tamaño de *kernel* correcto, debido a la potencial variación en el tamaño y localización de las áreas de interés en las imágenes.
Nos referimos como *kernel* de convolución a un filtro digital que se aplica en una región de una matriz de entrada (imagen) para la extracción de características distintivas.
Para solucionar esto, los investigadores diseñaron el módulo *Inception* (Figura 3) que consiste de una combinación de capas convolucionales paralelas, cuyos mapas de activación se concatenan al final.
Un mapa de activación consiste en una matriz resultante de aplicar una serie de operaciones en cada una de las posiciones posibles en una matriz de datos.
En concreto, el módulo *Inception* posee tres canales, cuyas capas convolucionales poseen filtros de 1×1, 3×3 y 5×5, así como un canal de valor máximo (*max pooling*). Todas las variantes de la arquitectura *Inception* consisten en diversas configuraciones de este módulo.

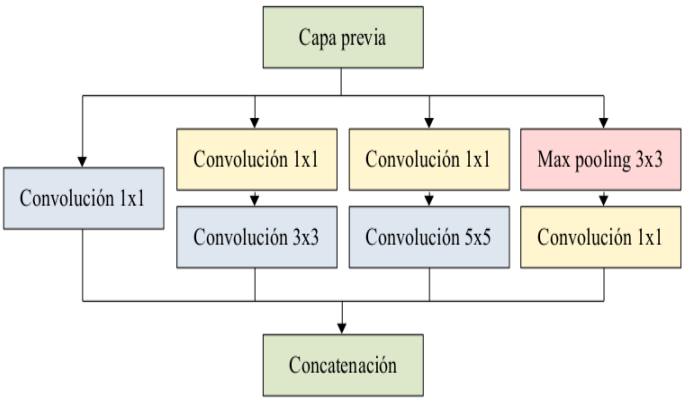

Figura 3. Módulo de *Inception* clásico.

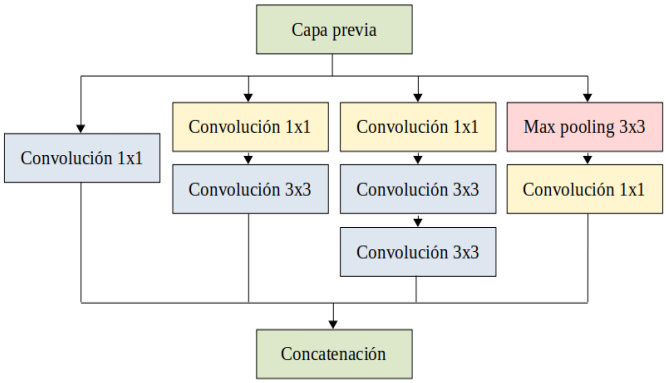

Figura 4. Módulo de *Inception* v2, con la convolución de 5×5 refactorizada en dos de 3×3.

La principal ventaja del módulo *Inception* es permitir a la red decidir el tamaño de filtro más adecuado para aprender, ya que utiliza filtros de distintos tamaños en sus capas convolucionales. Con el propósito de mitigar el incremento en costo computacional, el módulo *Inception* hace uso de convoluciones de 1×1 para reducir la dimensionalidad de los datos antes de enviarlos a las capas de convolución de 3×3 y 5×5. De esta forma es posible crear una red profunda sin sacrificar demasiado en desempeño.

Adicionalmente, la arquitectura cuenta con dos clasificadores auxiliares, cuyo propósito es mitigar el problema de desvanecimiento de gradiente (*vanishing gradient*), el cual ocurre cuando los gradientes se vuelven demasiado pequeñas para tener algún impacto en el entrenamiento de la red. En resumen, estos clasificadores producen dos valores de perdida (*loss*) auxiliares, los cuales se suman parcialmente al valor de perdida real durante el entrenamiento.

La segunda versión de esta arquitectura se presentó en conjunto con la versión 3. Los cambios más significativos consisten en la factorización de las convoluciones de 5×5 a dos convoluciones de 3×3 para reducir el costo computacional ver Figura 4. Adicionalmente, las convoluciones de $N \times N$ son reemplazadas por dos convoluciones con la forma $1 \times N$ y $N \times 1$ ver Figura 5. Este método resultó ser hasta 33% más eficiente que las convoluciones tradicionales [10].

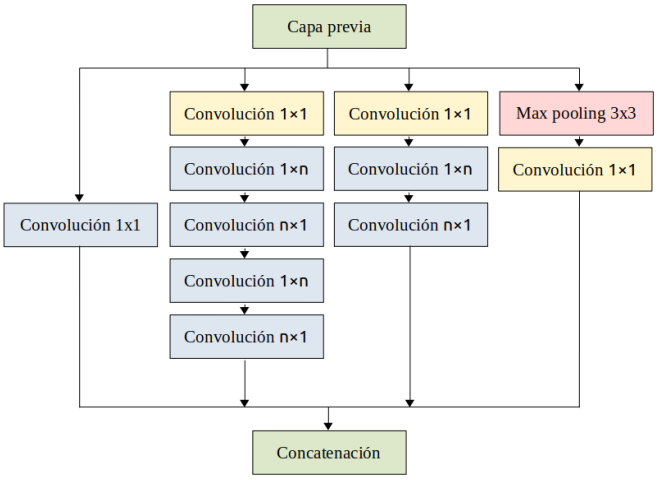

Figura 5. Factorización de capas convolucionales en *Inception* v2

Además de las mejoras presentes en la versión 2, *Inception* v3 posee convoluciones de 7×7 factorizadas, normalización por lotes (*batch normalization*) en los clasificadores auxiliares y hace uso de suavizado de etiquetas (*label smoothing)*.

La normalización por lotes es una técnica que normaliza los datos de salida de las capas de una red neuronal, mientras que el suavizado de etiquetas asigna pesos a las etiquetas de verdad (*ground truth*) en lugar de 0s y 1s con el fin de ayudar a la red a generalizar mejor.

3.3. ResNet.

La arquitectura *ResNet* fue presentada por He et al. [11] en 2015. La idea detrás de esta arquitectura es resolver el problema de degradación de desempeño en redes neuronales profundas. En su trabajo, los investigadores destacan como el desempeño de una red neuronal de una cierta profundidad se degrada conforme se incrementa el número de capas. La causa detrás de esto es un tema de debate, pero los investigadores denotan que no se debe

al problema de desvanecimiento de gradiente sino a una incapacidad de los modelos para aprender la función identidad. Para solucionar este problema, He et al. [11] en el 2015 proponen el aprendizaje residual. Como se aprecia en la Figura 6, la conexión residual se salta las capas subsecuentes. La información que las capas reciben como datos de entrada se suman a sus datos de salida. La hipótesis detrás de esto es que es más sencillo para las capas de la red aprender una función residuo en lugar de la función identidad.

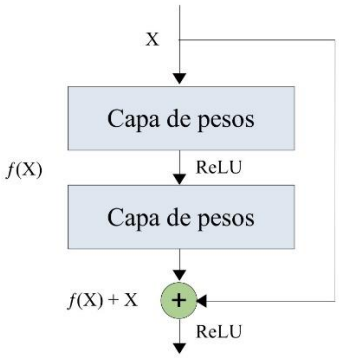

Figura 6. Diagrama de una conexión residual.

3.4. InceptionResNet

Luego de la introducción de las conexiones residuales por parte de He et al. [11] en el 2015, los investigadores responsables de la arquitectura *ResNet* decidieron combinar la idea con el módulo *Inception*. El estudio, llevado a cabo en 2016, dio como resultado la arquitectura *InceptionResNet* [12]. La arquitectura posee tres módulos *Inception* rediseñados, menos costosos que sus antecesores. Como se puede apreciar en la Figura 7, posee de igual forma una conexión residual que omite las capas intermedias del módulo.

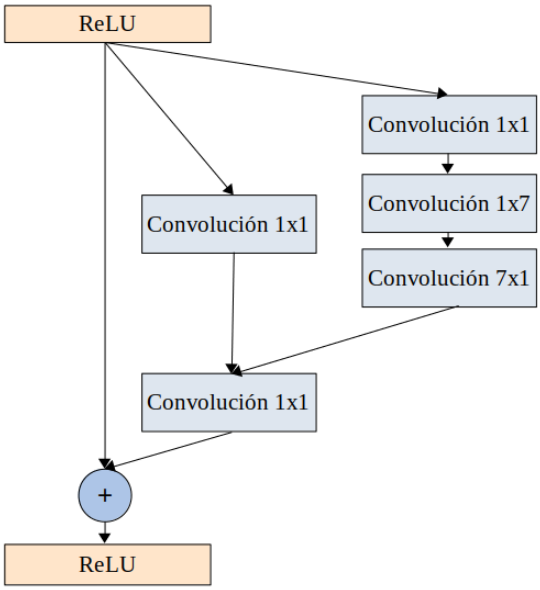

Figura 7. Módulo *InceptionResNet* B.

4. EXPERIMENTOS Y RESULTADOS

Los experimentos fueron realizados utilizando una computadora con las siguientes características. Procesador Intel Xeon W-2133, 16GB de RAM y una tarjeta gráfica NVIDIA GTX 1080. Sistema operativo Ubuntu 16.04 y *CUDA toolkit* 10.0.

Utilizamos el conjunto de datos de flores de plantas del Visual Geometry Group [13], que contiene 8189 imágenes divididas en 102 clases diferentes de flores, originarias del Reino Unido.

Para desarrollar las arquitecturas de CNNs de *DenseNet* e *InceptionResNet* utilizamos Keras versión 2.2.4 y Tensorflow 1.13.1, en un entorno virtual creado en Anaconda 4.6.7 y Python 3.7.8.

Las arquitecturas de CNNs fueron entrenadas con 200 épocas y tamaño de lote (*batch size*) de 16. El conjunto de datos fue dividido de manera aleatoria, un 75% de las imágenes se utilizó para el proceso de entrenamiento y el 25% restante para el proceso de validación.

Tabla 1. Comparativa de precisiones de arquitecturas CNNs

Top	Precisión *DenseNet*	Precisión *InceptionResNet*
1	63.14%	72.65%
3	79.02%	86.47%
5	85.19%	90.59%

En la Tabla 1 se resumen los resultados de desempeño de las arquitecturas *DenseNet* e *InceptionResNet*. *InceptionResNet* fue la arquitectura con mejores resultados con un Top 1 de 72.65%, un Top 3 de 86.47% y un Top 5 de 90.59%.

5. CONCLUSIONES

En este trabajo presentamos un estudio comparativo de técnicas de aprendizaje profundo, especialmente redes neuronales convolucionales (CNNs) para el proceso de clasificación de especies de flores de plantas en imágenes digitales. Utilizamos el conjunto de flores de VGG con 102 especies distintas, para el entrenamiento de las arquitecturas *DenseNet169* e *InceptionResNet*. *InceptionResNet* obtuvo el mejor desempeño, obteniendo 90.59% de precisión en validación top 5. Consideramos que el desempeño superior de *InceptionResNet* se debe a su capacidad de seleccionar los tamaños de filtros más adecuados durante el entrenamiento. Como trabajo a futuro, tenemos la intención de entrenar otras arquitecturas de CNNs y poder realizar un estudio comparativo mayor.

6. REFERENCIAS

[1] T. Rumpf, A.-K. Mahlein, U. Steiner, E.-C. Oerke, H.-W. Dehne y L. Plümer, «Early detection and classification of plant diseases with Support Vector Machines based on hyperspectral reflectance,» *Computers and Electronics in Agriculture*, vol. 74, pp. 91-99, 10 2010.

[2] S. Wu, F. Bao, E. Xu, Y.-X. Wang, Y.-F. Chang y Q.-L. Xiang, «A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network,» *IEEE International Symposium on Signal Processing and Information Technology*, pp. 11-16, 8 2007.

[3] Y. LeCun, Y. Bengio y G. Hinton, «Deep Learning,» *Nature*, vol. 521, pp. 436-44, 5 2015.

[4] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg y F. F. Li, «ImageNet Large Scale Visual Recognition Challenge,» *International Journal of Computer Vision*, vol. 115, pp. 211-252, 9 2014.

[5] T.-N. Doan y F. Poulet, «Large Scale Image Classification: Fast Feature Extraction, Multi-codebook Approach and Multi-core SVM Training,» *Advances in Knowledge Discovery and Management*, vol. 4, pp. 155-172, 1 2014.

[6] F. Perronnin, J. Sánchez y T. Mensink, «Improving the Fisher Kernel for Large-Scale Image Classification,» *Proceedings of IEEE European Conference on Computer Vision, 2010*, vol. 6314, pp. 143-156, 9 2010.

[7] A. Krizhevsky, I. Sutskever y G. E. Hinton, «ImageNet Classification with Deep Convolutional Neural Networks,» *Neural Information Processing Systems*, vol. 25, pp. 84-90, 1 2012.

[8] S. H. Lee, C. S. Chan, P. Wilkin y P. Remagnino, «Deep-plant: Plant identification with convolutional neural networks,» *IEEE International Conference on Image Processing (ICIP)*, pp. 452-456, 9 2015.

[9] G. Huang, Z. Liu, L. Maaten y K. Weinberger, «Densely Connected Convolutional Networks,» *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261-2269, 7 2017.

[10] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens y Z. B. Wojna, «Rethinking the Inception Architecture for Computer Vision,» *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818-2826, 6 2016.

[11] K. He, X. Zhang, S. Ren y J. Sun, «Deep Residual Learning for Image Recognition,» *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770-778, 6 2016.

[12] C. Szegedy, S. Ioffe y V. Vanhoucke, «Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning,» *AAAI Conference on Artificial Intelligence*, pp. 4278-4284, 2 2016.

[13] M.-E. Nilsback y A. Zisserman, «Automated Flower Classification over a Large Number of Classes,» *Sixth Indian Conference on Computer Vision, Graphics Image Processing*, pp. 722-729, 12 2008.