81 Ciencia Nicolaita

Diseño de una arquitectura de Red Neuronal Convolucional para la clasificación de objetos

Moisés García Villanueva y Leonardo Romero Muñoz

Facultad de Ingeniería Eléctrica, UMSNH

Resumen

Uno de los problemas más importantes y fundamentales en el área de Visión Computacional es la detección de objetos. Existe una gran cantidad de aplicaciones que requieren encontrar objetos en una escena y entonces clasificarlos, considerando la complejidad existente cuando se presentan varias categorías de objetos. El surgimiento de las técnicas de aprendizaje profundo ha llegado como una estrategia muy poderosa en la extracción automática de características a partir de las imágenes, provocando mejoras importantes en la problemática asociada a la detección de objetos. El objetivo de este artículo es presentar el diseño de una arquitectura de Red Neuronal Convolucional adecuada para clasificar 6 categorías diferentes de objetos comunes: cama, escalera, mesa, puerta, silla y sofá. Los resultados obtenidos indican una precisión superior al 90% en los experimentos realizados.

***Palabras clave*:** Clasificación de objetos, red neuronal convolucional, visión computacional, aprendizaje profundo.

Abstract

One of the most important and fundamental problems in the area of Computer Vision is object detection. There are a large number of applications that require seek objects in a scene and then classifying them, considering the complexity that exists when there are several categories. The deep learning techniques have emerged as a very powerful strategy in the automatic feature extraction from images, causing significant improvements in the general problem of object detection. The goal of this article is to present the design of a Convolutional Neural Network architecture suitable for classifying 6 different categories of common objects: bed, stairs, table, door, chair and sofa. The obtained results indicate a precision greater than 90% in the experiments carried out.

***Keywords*:** Object classification, convolutional neural network, computer vision, deep learning.

1. Introducción

El problema de detección de objetos es fundamental en el área de Visión Computacional, es complejo, desafiante e involucra el manejo de grandes volúmenes de datos. Se ha mantenido como un tópico activo de investigación por varias décadas (Fischler y Elschlager, 1973; Khan y col., 2020). Como piedra angular en la interpretación de una imagen y en visión por computadora, la detección de objetos forma la base para resolver tareas de visión complejas o de alto nivel, tales como segmentación, clasificación de escenas, seguimiento de objetos, subtítulos de imágenes, detección de eventos y reconocimiento de actividades. La detección de objetos admite una amplia gama de aplicaciones, incluida la visión de robots, la electrónica de consumo, seguridad, conducción autónoma, interacción humano-computadora, recuperación de imágenes basada en contenido, video vigilancia inteligente y realidad aumentada (Liu y col., 2020). El objetivo de la detección de objetos es determinar si hay instancias de objetos de categorías preestablecidas (algunas de ellas como: humanos, muebles, automóviles, bicicletas, perros o gatos) en una imagen y, si está presente, devolver la ubicación espacial y la extensión de cada objeto identificado (Everingham et al. 2010; Russakovsky et al. 2015).

Durante la última década el aprendizaje profundo se ha convertido en la "joya de la corona" de la inteligencia artificial y el aprendizaje automático (LeCun y col., 2015),

mostrando un rendimiento superior en diferentes áreas, tales como: la acústica (Hinton y col., 2012), el procesamiento de imágenes (Krizhevsky y col., 2012), el procesamiento de lenguaje natural (Bahdanau y col., 2014), por mencionar algunas. El prominente poder del aprendizaje profundo para extraer patrones complejos de datos subyacentes es bien reconocido (Cohen et al, 2016; Zhang y col., 2020). Ese gran éxito en una variedad de problemas de aprendizaje automático se encuentra en las redes neuronales convolucionales profundas (CNN por sus siglas en inglés de *Convolutional Neural Network*). Las CNNs son consideradas como las técnicas dominantes del aprendizaje profundo (LeCun y col., 2015). Sin embargo, el rendimiento de las CNNs depende en gran medida de sus arquitecturas (Krizhevskyy col., 2012; Simonyan y col., 2014). Para lograr el mejor rendimiento, los modelos de CNNs del estado del arte, como GoogleNet (Szegedy y col., 2015), ResNet (He y col., 2016) y DenseNet (Huang y col., 2017), están diseñadas manualmente por expertos que tienen gran conocimiento del dominio de los datos a investigar y el desarrollo de modelos CNN. Sin embargo, los modelos de estos autores requieren de un gran poder de cómputo y miles de imágenes por categoría para lograr el rendimiento publicado, limitando su uso en otras aplicaciones. Esto significa que un área activa de investigación se encuentra en el diseño de nuevas arquitecturas de CNN, especializadas en nuevas aplicaciones de detección de objetos.

La contribución de este trabajo es el diseño de una arquitectura CNN que permite la clasificación de seis objetos diferentes en el dominio del hogar: cama, silla, mesa, sofá, puerta y escalera. Para justificar la selección de los objetos y el dominio seleccionado, los autores actualmente están desarrollando mejoras al prototipo "Escáner Láser Parlante para Guiar a Personas Ciegas" (Romero y col., 2019), una de las propuestas es la implementación de un sistema de visión por computadora que permita la identificación de objetos, entonces describir la escena por voz, señalando la existencia y distancia a la que se encuentran los elementos identificados. Se presentan los experimentos que llevaron a desarrollar la arquitectura final, permitiendo con esto mostrar un procedimiento alternativo para crear CNNs que se adapten a aplicaciones con características específicas. La precisión obtenida supera el 90% para un conjunto de datos de prueba creado para una aplicación. Adicionalmente se proporciona el conjunto de datos para futuras investigaciones.

2. Metodología

La convolución es el bloque principal de construcción de una CNN. El término convolución se refiere a la combinación matemática de dos funciones para producir una tercera, es decir, fusiona dos conjuntos de información. En el caso de imágenes, se cuenta con una imagen de entrada de dimensiones alto (H) por ancho (W) y la cantidad de canales que representan el color C=3 (rojo, verde y azul), de tal forma que se tiene $\mathrm{I} \in \mathbb{R}^{\mathrm{H}\times\mathrm{W}\times\mathrm{C}}$. Si se considera un banco de filtros de cantidad D, se tiene entonces que $K \in \mathbb{R}^{k_1\times k_2\times C\times D}$, en donde $k_1 \times k_2$ se refiere a las dimensiones (renglones, columnas) de cada filtro, adicionando un valor de sesgo $\mathrm{b} \in \mathbb{R}^{\mathrm{D}}$ para cada filtro. La salida del procedimiento de convolución es el indicado en la ecuación (1) (Rosebrock, 2017).

$$(\mathrm{I} * \mathrm{K})_{\mathrm{i,j}} = \sum_{\mathrm{m}=0}^{\mathrm{k}_1-1} \sum_{\mathrm{n}=0}^{\mathrm{k}_2-1} \sum_{\mathrm{p}=0}^{\mathrm{C}} \mathrm{K}_{\mathrm{m,n,p}} \cdot \mathrm{I}_{\mathrm{i+m,j+n,c}} + \mathrm{b} \qquad (1)$$

Los modelos de redes neuronales convolucionales profundas tienen una serie de ventajas sobresalientes: a) una estructura jerárquica para aprender representaciones de datos con múltiples niveles de abstracción; b) la capacidad de aprender funciones muy complejas; y c) aprender representaciones de características directas en forma automática a partir de datos con un conocimiento mínimo del dominio (Liu y col., 2020).

Las CNNs están compuestas de diferentes filtros/núcleos, los cuales constituyen un conjunto de parámetros entrenables y que pueden convolucionar espacialmente una imagen dada para detectar características como bordes y formas. Este alto número de filtros esencialmente aprende a capturar características espaciales de la imagen en función de los pesos aprendidos a través de la propagación hacia atrás. Las capas apiladas de filtros se pueden usar para detectar formas complejas a partir de las características espaciales en cada nivel posterior. Por lo tanto, pueden reducir con éxito una imagen dada en una representación altamente abstracta que es fácil de predecir. Las arquitecturas CNNs se pueden encontrar con diversas variantes, sin embargo, en general se componen de capas convolucionales y de submuestreo que se encuentran agrupadas en módulos o bloques convolucionales. Otro elemento de este tipo de modelos son las capas completamente conectadas (densas), tal y como se conocen en las redes neuronales de retroalimentación hacia

adelante, colocadas después de los módulos que agrupan varias operaciones de las CNNs. Cuando estos módulos son apilados uno sobre otro da origen a lo que se conoce como un modelo profundo (Rawat y Wang, 2017). La Figura 1 muestra una arquitectura típica de una CNN en la tarea de clasificación de imágenes, en la que se indica los diferentes bloques que la conforman.

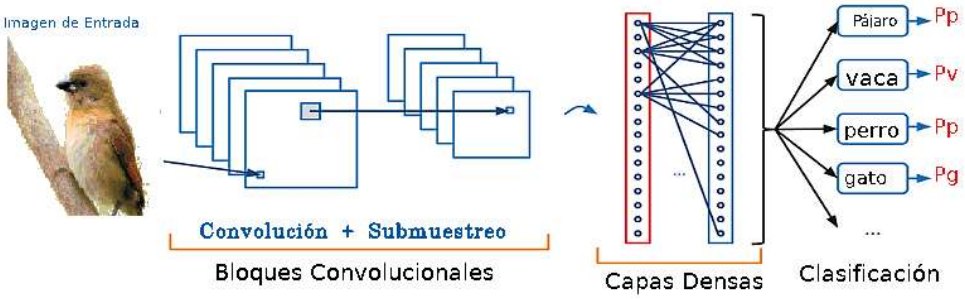

Figura 1.- Arquitectura típica de una Red Neuronal Profunda en la clasificación de objetos.

La práctica común en la mayoría de los desarrollos recientes de redes profundas es implementar modelos más grandes y profundos para lograr un mejor rendimiento. A medida que el modelo se hace más grande y profundo, los parámetros de la red se incrementan dramáticamente. Como resultado, el modelo se vuelve más complejo de entrenar y en consecuencia es más costoso computacionalmente. Por lo tanto es muy importante diseñar una arquitectura que proporcione un mejor rendimiento utilizando un número razonablemente menor de parámetros de la red (Alom y col., 2018), principalmente si éste será utilizado en una aplicación que requiere respuestas en tiempo real o se pretende implementar en algún sistema embebido.

3. Conjunto de datos

Los conjuntos de datos han jugado un rol clave a través de la historia en la investigación de reconocimiento de objetos, no solamente como una base común para medir y comparar el desempeño de algoritmos, también empujando a esta área de investigación hacia problemas cada vez más complejos y desafiantes. El acceso a una gran cantidad de imágenes en internet permite construir conjuntos de datos completos para capturar una gran riqueza y diversidad de los elementos que componen una escena, lo que permite un rendimiento sin precedentes en el reconocimiento de objetos (Liu y col., 2020). Aun cuando se cuenta con una facilidad

increíble para obtener imágenes de objetos a través de Internet, es necesaria una especial atención en la construcción del conjunto de datos etiquetado a gran escala. Desafíos como la diversidad de formas de los objetos, la posición en la escena, diferentes ubicaciones de un objeto, la oclusión, ruido en las imágenes, etcétera, deben ser claramente considerados e identificados en los diversos escenarios de la aplicación en particular que se va a implementar. En este trabajo se construyó un conjunto de datos de seis categorías: cama, escalera, mesa, puerta, silla y sofá. La Figura 2 contiene algunos ejemplos de los objetos en cada una de las categorías. La cantidad de elementos que contiene cada clase y su partición en datos para entrenar, validar y probar se indican en la Tabla 1. Se utilizó la interfaz de programación de aplicaciones (API del inglés Application Programming Interface) de Flickr (Mignon 2018) para obtener imágenes relacionadas mediante la consulta de los objetos en el idioma inglés y español. Flickr es una red social en línea para compartir fotografías, para el 2009 contaba con más de dos billones de fotos y 2.3 millones de usuarios (Cha y col., 2009). El preprocesamiento de los datos obtenidos consistió en recortar lo más posible los objetos de la imagen, considerando mantenerlos en forma individual, es decir, no incluir objetos de dos o más categorías en una imagen.

Figura 2.- Ejemplos de imágenes de las diferentes categorías que conforman nuestro conjunto de datos.

TABLA 1

Cantidad de imágenes en las seis categorías, subdivididos en tres diferentes subconjuntos de datos.

Categoría	Cantidad de imágenes			Total
	Entrenar	Validar	Probar	
Cama	900	100	33	1033
Escalera	905	285	51	1241
Mesa	1471	141	54	1666
Puerta	1675	281	41	1997
Silla	1044	207	82	1333
Sofá	900	100	21	1021
Total	6895	1114	282	8292

Las instancias del subconjunto Probar (aquellas imágenes que no se utilizaron en el proceso de entrenamiento de las CNN's), se obtuvieron mediante consultas realizadas directamente en el buscador de imágenes de Google.

4. Diseño de la arquitectura CNN

Múltiples opciones se tienen para crear una arquitectura que logre el mayor desempeño en la solución del problema que se pretende resolver. Se puede iniciar con las implementaciones más populares de redes neuronales convolucionales: VGG (Simonyan y Zisserman, 2014), Resnet (He y col., 2016), Inception (Szegedy y col., 2015) y Xception (Chollet 2017). En este trabajo se explora un modelo con características similares a VGG, por su facilidad de implementación. Se inicia con varias capas convolucionales y se culmina con una o más capas densas. Se procedió a realizar diferentes pruebas e ir evaluando el desempeño de los mejores modelos generados durante el proceso de entrenamiento. Las pruebas consistieron en a) incrementar la cantidad de capas convolucionales en la arquitectura; b) emplear diferentes cantidades de neuronas en las capas densas; y c) utilizar diferentes tamaños de las imágenes de entrada o dimensiones de la capa inicial.

Para llegar a la arquitectura propuesta, se estimó el rendimiento del modelo empleando la métrica de precisión (***E***), definida por la ecuación (2).

$$E = \frac{P_C}{N_T} \qquad (2)$$

En donde P_c es la cantidad de predicciones correctas del modelo y N_T es el número total de predicciones.

a. Cantidad de Capas Convolucionales

Se estableció la dimensión de la capa de entrada de 150x150 (imágenes de 150x150 pixeles) y a partir de 2 capas convolucionales se procedió a entrenar y verificar el desempeño de los mejores modelos obtenidos por dicha arquitectura. Al ir incrementando la cantidad de capas en un modelo, se espera de una mayor precisión en los resultados, sin embargo, esto dependerá de los datos de aplicación. En la Figura 3 se muestran los resultados de la arquitectura al ir incrementando la cantidad de capas convolucionales y manteniendo 2 capas densas al final de la CNN con diferentes tamaños, en ella se observa el comportamiento de clasificación de los mejores modelos obtenidos durante el proceso de entrenamiento de la arquitectura. Se procedió de igual forma con una capa densa en la arquitectura, observando el mejor desempeño con cuatro capas convolucionales, logrando un 90.07% de precisión en los datos de prueba.

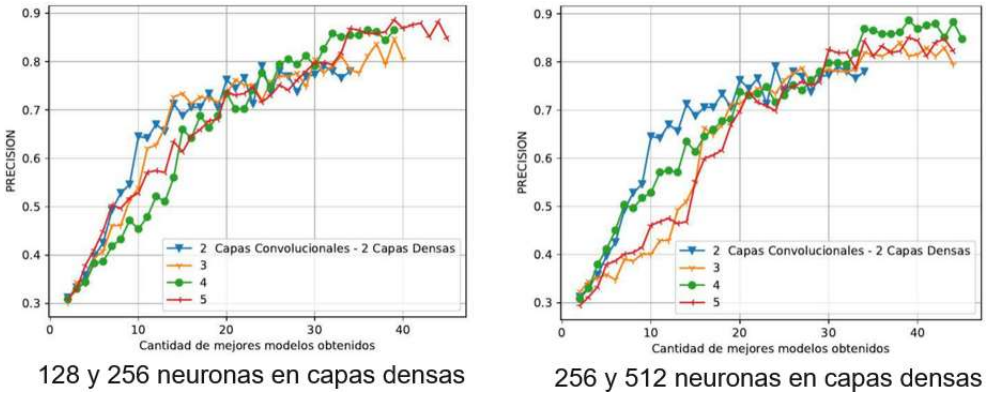

Figura 3.- Resultados de clasificar los datos de prueba con los mejores modelos obtenidos durante el proceso de entrenamiento y al incrementar la cantidad de capas convolucionales en la arquitectura, el mejor desempeño se observa con los modelos de 4 capas.

b. Cantidad de neuronas en las capas densas

Para establecer la cantidad de neuronas en cada una de las capas densas, se realizaron pruebas utilizando cantidades para 64, 128, 256, 512 y 1024 neuronas en una capa, mientras que para dos capas se establecieron las cantidades 64 - 128, 128 - 256, 256 - 512 y 512 -1024 respectivamente. Las cantidades de neuronas por capa pueden ser cualquier valor, incluso puede realizarse con incrementos de uno en uno, lo que hace el desarrollo e implementación poco práctico, por la familiaridad con los valores en potencias de 2, en el presente trabajo se optó establecer dichas cantidades. Las Figuras 4 y 5 presentan los resultados de desempeño con el subconjunto de datos Probar, para los mejores modelos obtenidos durante el proceso de entrenamiento de la arquitectura tanto para una y dos capas densas, respectivamente. El mejor resultado obtenido con dos capas densas se obtuvo en las cantidades de neuronas 256 - 512, logrando un 89.71% de precisión. En las pruebas con una capa densa con 256 neuronas, se logró obtener una precisión del 90.07%.

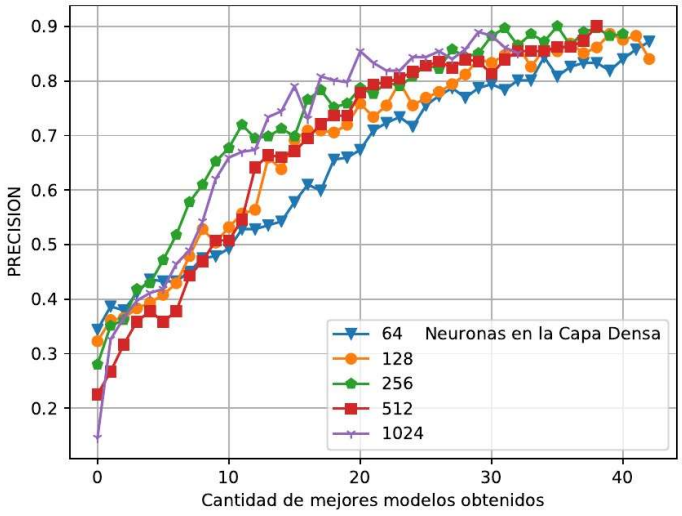

Figura 4.- Desempeño con el subconjunto de datos Probar, utilizando los mejores modelos obtenidos a diferentes cantidades de neuronas en la arquitectura con una capa densa.

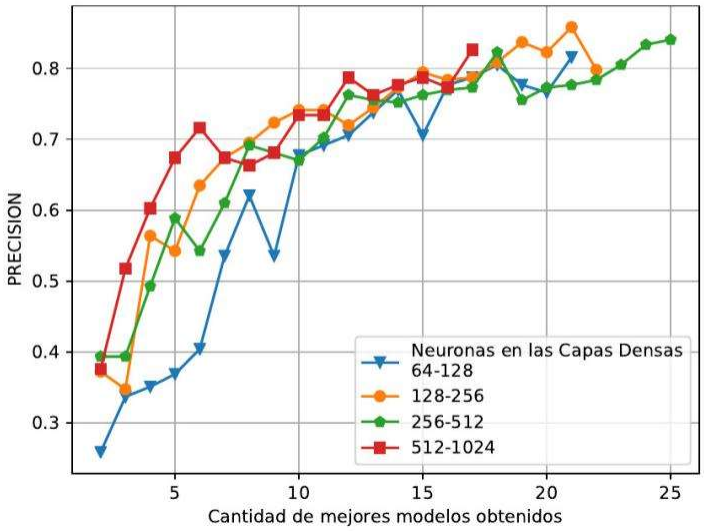

Figura 5.- Desempeño con el subconjunto de datos Probar, utilizando los mejores modelos obtenidos a diferentes cantidades de neuronas en la arquitectura con dos capas densas.

c. Dimensión de la capa de entrada

Determinar el tamaño de la capa de entrada o dimensiones de la imagen, también afecta el rendimiento de una CNN. Se realizaron pruebas para las dimensiones de 100x100 a 225x225, ancho y alto de las imágenes para la capa de entrada. Dimensiones por arriba de estos valores implica: a) un mayor tiempo de entrenamiento, b) el tamaño del modelo ocupa más espacio de memoria, y c) se incrementa el tiempo de procesamiento al momento de aplicar el modelo en un sistema real. Por otro lado, dimensiones inferiores no nos permiten incrementar el tamaño de las capas convolucionales en la arquitectura, debido a la reducción producida por las operaciones inherentes en las CNNs.

La Figura 6 muestra en forma gráfica el detalle de la arquitectura propuesta que obtuvo los mejores resultados. En ella se observan 4 bloques convolucionales que consisten en una capa convolucional y una capa de máximo muestreo (*Max-Pool*) e indicando la cantidad de filtros en cada bloque, una capa densa de 256 neuronas y la capa que proporciona las probabilidades de la clase a la cual pertenece la imagen de entrada.

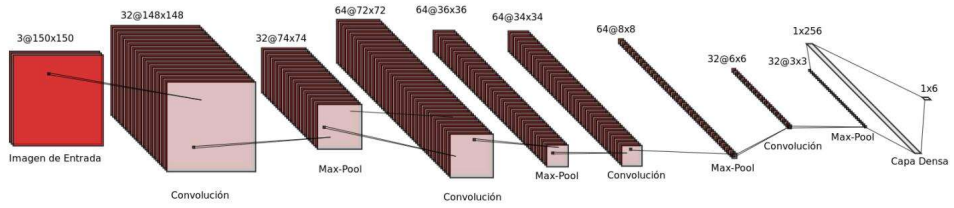

Figura 6.- Forma gráfica de la arquitectura propuesta con 4 capas convolucionales, las operaciones de máximo muestreo, una capa densa de 256 y la capa de clasificación para 6 clases.

5. Comparación de resultados

En las arquitecturas de aprendizaje profundo, las redes denominadas Inicio (Inception) han sido consideradas el estado del arte en la solución de problemas que identifican y reconocen imágenes. Se considerará como línea base para comparar el desempeño de la arquitectura propuesta el modelo de Inicio inspirado en (Szegedy y col., 2015) y que aplicó en su trabajo (Mollahosseini y col., 2016), debido a que no se cuenta con la infraestructura para implementar el modelo original. Los resultados del desempeño al entrenar las arquitecturas a diferentes dimensiones de la imagen de entrada se muestran en la Tabla 2. Se observa que los modelos con 3 y 4 capas convolucionales obtienen los mejores rendimientos, superando la línea base establecida por la arquitectura Inception. La arquitectura propuesta se desempeña mejor para dimensiones de 100x100 y 150x150 (ancho x alto de la imagen), logrando superar la línea base en al menos 4 ocasiones en cada tamaño preestablecido. Se observa además que incrementar la dimensión en la capa de entrada, no afecta en forma proporcional el desempeño de la arquitectura, debido a que los mejores resultados no se encuentran en las pruebas con la dimensión de capa de entrada mayor. El incrementar la cantidad de capas, tanto convolucionales como densas, tampoco garantiza el incremento de desempeño, lo que hace evidente que para obtener el modelo final se convierte en una actividad a prueba y error que puede resultar muy laboriosa. Para una aplicación de un modelo de aprendizaje profundo en un sistema embebido, se requiere que éste ocupe poco espacio de memoria, los modelos complejos o de una gran cantidad de parámetros de entrenamiento (pocas capas convolucionales), logran tamaños de cientos de megabytes, por lo que requieren de una gran cantidad de recursos de cómputo en la aplicación. Para la dimensión de capa de entrada 225x225, en el modelo con 2 CNN y una capa densa el tamaño es de 365 MB, mientras que para la arquitectura

con 4CNN y una capa densa es de 2.2 MB, la arquitectura de línea base ocupa 1.1 GB de espacio en el disco duro bajo está condición.

TABLA 2

Precisión de la arquitectura propuesta y la línea base que implementa una arquitectura del tipo Inicio. CC significa capa convolucional y CD capa densa.

Arquitectura	Dimensiones				
	100 x 100	150 x 150	175x175	200 x 200	225x225
Inception (Mollahosseini y col., 2016)	82.97	85.10	82.62	87.58	85.81
Propuesta CNN (4CC - 1 CD 256)	84.39	**90.07**	89.00	86.52	88.65
(3CC - 1 CD 256)	87.23	85.10	84.75	86.87	84.04
(2CC - 1 CD 256)	82.62	82.98	81.20	83.68	79.07
(5CC - 1 CD 256)	-	-	84.75	86.52	85.81
(5CC - 2 CD 256 - 512)	-	-	88.29	85.10	87.58
(4CC - 2 CD 256 - 512)	81.20	89.00	**89.36**	89.00	84.39
(3CC - 2 CD 256 - 512)	**89.71**	87.58	87.23	86.87	86.52
(2CC - 2 CD 256 - 512)	84.04	81.56	83.33	82.97	79.78

El tiempo promedio de entrenamiento por época de las arquitecturas que se implementaron, empleando diferentes dimensiones de imágenes de entrada, utilizando una computadora con procesador Intel core i5 a 3.1 GHz y 8GB de memoria RAM, se indica en la Tabla 3. Para la dimensión de 100x100, la arquitectura Inception consume 44% más de tiempo que la arquitectura propuesta,

en la dimensión 150x150 es un 29% y para la dimensión 225x225 es un 18% que equivale a 7.44 horas. Considerando 200 épocas de entrenamiento y la dimensión más pequeña de la capa de entrada, la arquitectura propuesta requiere de 7.05 horas para su entrenamiento, mientras que el modelo de Inception se entrena en 10.22 horas.

TABLA 3

Tiempo promedio en segundos por época que utilizó cada arquitectura durante el proceso de entrenamiento a diferentes dimensiones de imágenes de entrada.

Arquitectura	Dimensiones				
	100 x 100	150 x 150	175x175	200 x 200	225x225
Inception (Mollahosseini y col., 2016)	184	351	428	544	725
Propuesta CNN	127	247	353	425	591

6. Conclusiones

Se ha presentado una arquitectura de una red neuronal convolucional para la clasificación de objetos en el dominio del hogar. Por las pruebas obtenidas se ha logrado un desempeño del 90% en la tarea de clasificación de seis categorías de objetos, superando la CNN de Inception (Mollahosseini y col., 2016) que se estableció como línea base para comparar los resultados. En relación con el tiempo de entrenamiento de la arquitectura propuesta, se observó que es de al menos el 18% inferior para la dimensión mayor de la capa de entrada respecto al tiempo de entrenamiento de la CNN base. A mayor dimensión de la capa de entrada, el incremento de tiempo para entrenar la red neuronal puede ser de varias decenas de horas, en el caso de la dimensión 225 la arquitectura propuesta requiere de 32.8 horas aproximadamente, para la misma condición la arquitectura de línea base requirió 40.27 horas. Se señala que los mejores modelos obtenidos se encontraron en la segunda mitad de las épocas del proceso de entrenamiento, siempre por arriba de las 100. Dado que en una red neuronal se tiene una gran cantidad de parámetros

a modificar para el diseño de un modelo, este trabajo es una evidencia empírica que se puede emplear como punto de partida en la creación de nuevos modelos, al reportar las pruebas: cantidad de capas convolucionales y densas, dimensión de la capa de entrada y de las capas densas. Se tiene entonces como trabajo futuro la exploración de otro subconjunto de parámetros.

Gracias a la disponibilidad de grandes volúmenes de imágenes en la Web y a las herramientas de desarrollo de redes neuronales, es posible crear conjuntos de datos que hacen posible la implementación y diseño en forma rápida de arquitecturas CNN's para aplicaciones específicas.

Agradecimientos

Agradecemos a la Facultad de Ingeniería Eléctrica de la Universidad Michoacana de San Nicolás de Hidalgo, por el apoyo y facilidades para llevar a cabo este trabajo.

Referencias

Fischler, M. A., y Elschlager, R. A. (1973). The representation and matching of pictorial structures. IEEE Transactions on computers, 100(1), 67-92.

Khan, A., Sohail, A., Zahoora, U., y Qureshi, A. S. (2020). A survey of the recent architectures of deep convolutional neural networks. Artificial Intelligence Review, 1-62.

Romero Muñoz, L., García Villanueva, M. y Ramos Badillo, J., 2019. Escáner Láser Parlante para Guiar a Personas Ciegas. In: M. García Trillo, L. Márquez Pérez and W. Jacinto Díaz, ed., Diseño de prototipos para la inclusión de personas con discapacidad, 1st ed. Morelia Michoacán México: Universidad Michoacana de San Nicolás de Hidalgo, pp.176-192.

Rosebrock, A. (2017). Deep Learning for Computer Vision with Python: Starter Bundle. PyImageSearch.

Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., y Pietikäinen, M. (2020). Deep learning for generic object detection: A survey. International journal of computer vision, 128(2), 261-318.

Mignon, A. (2018). Flickr API. Version 0.6.1. URL: https://github.com/alexis-mignon/python-flickr-api.

Cha, M., Mislove, A., y Gummadi, K. P. (2009, April). A measurement-driven analysis of information propagation in the flickr social network. In Proceedings of the 18th international conference on World wide web (pp. 721-730).

Everingham, M., Gool, L. V., Williams, C., Winn, J., y Zisserman, A. (2010). The pascal visual object classes (voc) challenge. IJCV, 88(2), 303–338.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). ImageNet large scale visual recognition challenge. IJCV, 115(3), 211–252.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. (2015). Deep learning. Nature, 521(7553), 436–444.

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. R., Jaitly, N., ... y Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal processing magazine, 29(6), 82-97.

Krizhevsky, A., Sutskever, I., y Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105). Bahdanau, D., Cho, K., y Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

Cohen, N., Sharir, O., y Shashua, A. (2016, June). On the expressive power of deep learning: A tensor analysis. In Conference on Learning Theory (pp. 698-728).

Zhang, Z., Cui, P., y Zhu, W. (2020). Deep learning on graphs: A survey. IEEE Transactions on Knowledge and Data Engineering.

Simonyan, K., y Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... y Rabinovich, A. (2015). Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1-9).

He, K., Zhang, X., Ren, S., y Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

Huang, G., Liu, Z., Van Der Maaten, L., y Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).

Rawat, W., y Wang, Z. (2017). Deep convolutional neural networks for image classification: A comprehensive review. Neural computation, 29(9), 2352-2449.

Alom, M. Z., Hasan, M., Yakopcic, C., Taha, T. M., y Asari, V. K. (2018). Improved inception-residual convolutional neural network for object recognition. Neural Computing and Applications, 1-15.

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1251-1258).

Mollahosseini, A., Chan, D., y Mahoor, M. H. (2016, March). Going deeper in facial expression recognition using deep neural networks. In 2016 IEEE Winter conference on applications of computer vision (WACV) (pp. 1-10). IEEE.