ISSN 1870-4069

Red neuronal convolucional con extracción de características multi-columna para clasificación de imágenes

Fidel López Saca, Andrés Ferreyra Ramírez, Carlos Avilés Cruz, Juan Villegas Cortéz

Universidad Autónoma Metropolitana, Unidad Azcapotzalco, Ciudad de México, México
fidelosmcc@gmail.com, {fra, caviles, juanvc}@correo.azc.uam.mx

Resumen. En los últimos años, las redes neuronales convolucionales han traído una evolución significativa en la clasificación de imágenes, estos métodos poderosos de aprendizaje automático extraen características de nivel inferior y las envían a las siguientes capas para identificar características de nivel superior que mejoran el rendimiento. En el presente artículo se propone una arquitectura de una red convolucional con un enfoque que logra incrementar el rendimiento de clasificación, es una estructura jerárquica que es fácil de construir, es adaptable y fácil de entrenar con un buen rendimiento en tareas de clasificación de imágenes. En la arquitectura propuesta, la imagen ingresa a la red por tres secciones de extracción de características que utilizan diferentes filtros de convolución, las características extraídas son combinadas y enviadas a capas totalmente conectadas para realizar la clasificación. Los resultados experimentales muestran que la arquitectura propuesta tiene un rendimiento superior a diferentes redes convolucionales tradicionales tales como AlexNet, GoogleNet y ResNet 152.

Palabras clave: redes neuronales convolucionales, clasificación, extracción de características.

Convolutional Neural Network with Extraction of Multi-Column Features for Image Classification

Abstract. Due to the immense amount of information available on the Internet, it causes users to feel overwhelmed with so much information, making it difficult to search for products and/or services that suit the tastes and needs of each user. For this reason the development of intelligent tools has become indispensable as are the Recommendation Systems, where its main objective is to help users find information of products and/or services in a better way filtering all the available information thus achieving a better use of it. In the present research work a Hybrid Recommendation Algorithm is designed and developed to create a list of recommended items (restaurants) for users (consumers), merging the

algorithms: Collaborative and Content Based Filter, using a Bayesian Classifier with techniques of Natural Language Processing. Besides, the user experience is improved by applying your GPS location as a filter to the recommendations. To measure the performance of the proposed system, we experimented with a set of data extracted from the Foursquare and TripAdvisor Websites.

Keywords: convolutional neural network, classification, feature extraction.

1. Introducción

En las últimas décadas, el crecimiento constante de imágenes digitales -como fuente principal de representación de la información para aplicaciones científicas- ha hecho de la clasificación de imágenes una tarea desafiante. Con el fin de alcanzar rendimientos de clasificación altos, se han propuesto diferentes técnicas de reconocimiento de patrones, entre las que se encuentran los métodos de aprendizaje profundo que hoy en día son un foco de estudio en el procesamiento de imágenes y visión por computadora. En este enfoque, la arquitectura más popular para la tarea de clasificación de imágenes son las redes neuronales convolucionales (CNNs, por sus siglas en inglés); una red construida de múltiples capas y en donde cada capa modela un campo receptivo de la corteza visual lo que la hace mucho más efectiva en tareas de visión artificial [11].

Las CNN combinan las características de bajo nivel dentro de características abstractas de alto nivel con transformaciones no lineales, lo que le permite tener la capacidad de aprender la representación semántica de las imágenes. Estas redes extraen características generalmente útiles de los datos con o sin etiquetas, detectan y eliminan redundancias de entrada y preservan solo aspectos esenciales de los datos en representaciones sólidas y discriminativas [2]; pueden capturar las características mas obvias de los datos [19], por lo que podrían lograr mejores resultados en varias aplicaciones. A diferencia de las características creadas a mano, como SIFT [12] y HOG [3]; las características extraídas por la CNN se generan de extremo a extremo, lo que elimina la intervención humana. Las CNN tienen menos conexiones y parámetros lo que favorece que la extracción de características sea más eficiente.

La mayoría de arquitecturas de redes convolucionales tales como AlexNet [10], GoogleNet [16], VGG [15], ResNet 152 [8], y muchas otras [7], [18], [13], [6], [5], utilizan el mismo concepto para producir mapas de características en las capas de convolución, seguidas de capas de 'pooling' para reducir la dimensión de los mapas y conforme profundizan en la arquitectura, duplican el número de filtros para compensar la reducción a la mitad del tamaño de los mapas de características posteriores. La profundidad de la red favorece el rendimiento de clasificación y evita la desaparición de los gradientes mediante el uso de la inferencia de clases en capas de convolución consecutivas y la capa de pooling máximo, o el uso de capas softmax que realza el desvanecimiento de los gradientes [16], [15],[19]. Algunas de estas arquitecturas, utilizan nuevas funciones

de activación, métodos de regularización de actualización de pesos, inferencias de clase, entrenamiento previo por capas en enfoques supervisados; mostrado resultados muy prometedores [18], [5].

Aumentar el número de capas de una CNN significa aumentar la profundidad y el número de parámetros de la red; lo que complica el entrenamiento y reduce considerablemente el rendimiento, sobretodo con bases de datos pequeñas. Por otra parte, debido a la falta de características únicas, la fusión de características es cada vez mas importante para tareas como la clasificación y la recuperación de imágenes. Estas son técnicas que simplemente concatenan un par de características diferentes o utilizan métodos basados en el análisis de correlación canónica para la reducción de la dimensionalidad conjunta en el espacio de características.

En este artículo, se propone una red neuronal convolucional con extracción de características multi-columna para clasificación de imágenes. La red integra la capacidad de abstracción de las redes neuronales profundas y la capacidad de concatenar diferentes características. La red crece tanto en profundidad como en amplitud, la imagen a clasificar ingresa a través de tres secciones diferentes de extracción de características con diferentes filtros en las operaciones de convolución, las características extraídas son entonces concatenadas e ingresadas a capas totalmente conectadas que realizan la etapa de clasificación.

Los resultados muestran que la red tiene alto rendimiento en conjunto de imágenes como Oliva & Torralba [14], Stanford Dogs [9] y Caltech 256 [1]. Se realiza una comparación de la red propuesta con las redes existentes: AlexNet [10], GoogleNet [16] y ResNet [8], cuyas características están descritas en [4].

El documento está estructurado de la siguiente manera: En la sección 2 se describe el diseño de la red. La sección 3 analiza los conjuntos de datos empleados para los experimentos. En la sección 4 se describen los parámetros de entrenamiento. La sección 5 afirma la validez de la arquitectura propuesta mediante los experimentos realizados y los resultados experimentales. Finalmente en la sección 6, se discuten las conclusiones y el trabajo futuro.

2. Arquitectura propuesta

Aunque una sola CNN puede trabajar bien en problemas de clasificación de imágenes, puede presentar problemas para alcanzar precisiones altas en la fase de predicción. En la etapa de entrenamiento la red puede presentar un problema estadístico ya que el algoritmo de aprendizaje está buscando un espacio de parámetros (pesos y bias) que puede ser demasiado grande con respecto a la cantidad de datos de entrenamiento. En estos casos, puede haber muchos conjuntos de parámetros diferentes que produzcan la misma precisión, por lo que el algoritmo elije una de estas opciones; sin embargo, se corre el riesgo de que los parámetros elegidos no tengan buenas predicciones con las datos de prueba. En muchas aplicaciones, el algoritmo de aprendizaje de la CNN, no puede garantizar encontrar el mejor conjunto de pesos y bias; por lo que se produce un problema computacional, ya que se tiene que jugar con el ajuste de las variables de entrenamiento de la red, tales como: número de épocas, tasa

inicial de aprendizaje, tamaño del lote, etc. Por otra parte, cuando el espacio de soluciones no contiene un conjunto de parámetros que sea una aproximación correcta a la función objetivo verdadera, la red puede tener problemas en la representación de las características de las imágenes lo que dificulta aun más la clasificación y predicción. Los problemas mencionados anteriormente se pueden superar resolviendo el mismo problema de clasificación con varias CNN con arquitectura diferente y combinando los resultados. Esta expectativa nos llevó a proponer una arquitectura que en lugar de estar compuesta por diferentes CNN, esta diseñada con diferentes secciones de extracción de características; las cuales se combinan para realizar la clasificación. La arquitectura propuesta se muestra en la figura 1, está compuesta de 18 capas, las primeras quince son para la extracción de características y las últimas tres son para la clasificación. A continuación se hace una descripción de las capas de la red:

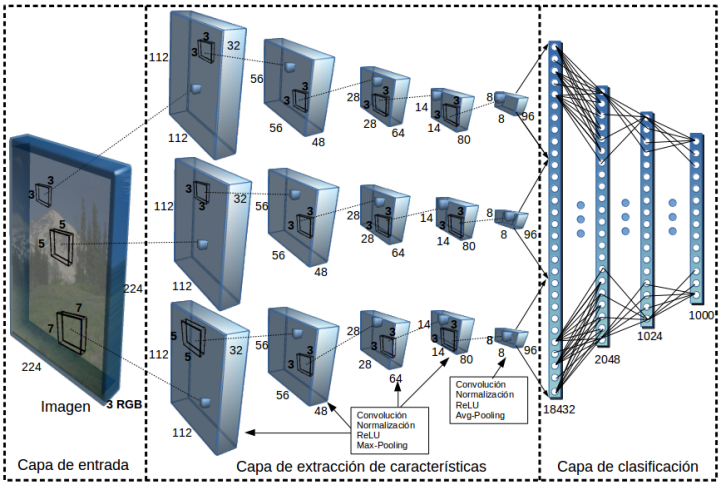

Fig. 1. Arquitectura general de la red neuronal propuesta.

- **Capa de entrada de la imagen**: esta capa establece una etapa de preprocesamiento de las imágenes de entrada a la red. En esta capa las imágenes se pueden redimensionar, rotar e incluso se pueden tomar pequeñas muestras al azar. Esta capa esta diseñada para que la red acepte imágenes con dimensiones 224×224 con una profundidad de tres, ya que la imagen ingresa con el formato RGB (ver figura 1, parte izquierda).
- **Capas de extracción de características**: la imagen ingresa por tres diferentes secciones de extracción de características, cada una extrae diferentes características utilizando filtros de diferentes tamaños. La primera sección utiliza un filtro de 3×3 en sus cinco capas. La segunda sección utiliza en la primera capa un filtro de 5×5 y en las siguientes cuatro, filtros de 3×3.

En la tercera sección, la primera capa utiliza un filtro de 7×7 en la segunda uno de 5×5 y en las últimas tres, filtros de 3×3. Las primeras cuatro capas de cada sección de extracción utilizan convolución, normalización, ReLU y max-pooling con un filtro de 3×3 y paso de 2 con relleno. En la última capa de cada sección se utilizan convolución, normalización, ReLU y avg-pooling con un filtro de 7×7, un paso de uno sin relleno (ver figura 1, parte central).

- **Clasificación**: la salida de cada sección de extracción de características se concatena para generar un vector unidimensional. Así se puede continuar con las capas totalmente conectadas para realizar la clasificación, como se describe a continuación (ver figura 1, parte derecha).
  - **Concatenación de las salidas de las 3 secciones de extracción de características**: la clasificación inicia con la concatenación de la salida de cada sección de extracción, se tendrá un vector con dimensiones 18432×1 (características de la imagen).
  - **Primera capa totalmente conectada**: tiene una profundidad de 2048 neuronas y es seguida de una capa de ReLU y una capa de Dropout.
  - **Segunda capa totalmente conectada**: consta de 1024 neuronas y es seguida de una capa de ReLU y una capa de Dropout.
  - **Tercera capa totalmente conectada**: esta capa es utilizada para ajustar la red convolucional a cada uno de los conjuntos de entrenamiento utilizados ya que es necesario igualar el número de neuronas de salida al número de clases de cada conjunto de entrenamiento. Esta capa es seguida de una capa de Softmax, una función de regresión, que ayuda a clasificar múltiples categorías.
- **Capa de salida**: es la capa final que se encarga de mostrar el porcentaje de éxito de clasificación.

Una red convolucional puede tener la tendencia a memorizar datos de entrenamiento, fenómeno conocido como sobreajuste, presentando porcentajes bajos de generalización. Para evitar esto, la red propuesta utiliza una capa de regularización comúnmente llamada "Dropout" en las primeras dos capas complemente conectadas. Las capas de Dropout hacen que la red sea más robusta a los datos de entrada imprevistos, y solo están activas durante la etapa de entrenamiento de la red, es decir, no están presentes durante la etapa de predicción.

3. Conjuntos de datos utilizados

En este trabajo se utilizaron 3 conjuntos de datos diferentes, los cuales se describen brevemente a continuación:

- **Oliva & Torralba** [14]: este conjunto de datos está compuesto por 2,688 imágenes de escenas a color que pertenecen a la misma categoría semántica. La base de datos tiene un total de 8 categorías y las imágenes fueron obtenidas de diferentes fuentes: bases de datos comerciales, sitios web y cámaras digitales.

- **Stanford Dogs** [9]: este conjunto de datos consta de 20,580 imágenes a color, pertenecientes a 120 clases o razas de perros de todo el mundo. Este conjunto de datos se ha creado utilizando imágenes y anotaciones de ImageNet para la tarea de categorización de imágenes de grano fino.
- **Caltech 256** [1]: este conjunto de datos consta de 30,607 imágenes a color pertenecientes a 256 categorías más una de nombre "clutter" que contiene múltiples escenas. Cada categoría contiene de 80 a 827 imágenes y la mayoría de las categorías tiene alrededor de 100 imágenes.

En la tabla 1 se muestran algunas estadísticas de los conjuntos de datos.

Tabla 1. Conjuntos de datos utilizados y sus estadísticas.

Conjunto de datos	Clases	Dimensiones			No. imágenes		
		Ancho	Alto	Prof	Total	Min x clase	Max x clase
Oliva & Torralba	8	256	256	3	2,688	260	410
Stanford Dogs	120	200 − 500	150 − 500	3	20,580	148	252
Caltech 256	257	300	200	3	30,607	80	827

4. Parámetros de entrenamiento

Tanto la red propuesta como las redes con las que se hace la comparación: AlexNet, GoogleNet y ResNet, fueron entrenadas utilizando el algoritmo de optimización ADAM (adaptive moment estimation), con un tamaño de lote de $\beta = 32$ imágenes y un decaimiento de pesos (factor de regularización) de $\lambda = 0{,}0005$. Los pesos iniciales en cada una de las capas fueron inicializados con una distribución gaussiana con una media de 0 y una desviación estándar de 0.01. Los umbrales de activación en cada una de las capas fueron inicializados a cero. Se inicio con una tasa de aprendizaje de $\mu = 0{,}001$ la cual se disminuyó en un factor de 10 después de cada 50 épocas, para tener cambios de aprendizaje más específicos en 250 épocas de entrenamiento. Las redes fueron entrenadas en 4 GPUs NVIDIA GTX 1080, con 8 GB de memoria RAM y 2560 núcleos, con sistema operativo Linux Ubuntu 16.04, Linux kernel 4.12, Python 2.7, TensorFlow 1.12, NVIDIA CUDA®8.0, NVIDIA cuDNN v5.1.

La falta de una cantidad suficiente de imágenes de entrenamiento, el desequilibrio de imágenes por clase y el balance desigual de clases dentro de cada conjunto de datos, puede provocar que una red se sobreajuste. Para hacer a las redes más robustas e invariantes a transformaciones en los datos, mejorar la eficiencia del proceso de entrenamiento e incrementar la generalización; utilizamos el método de aumento de datos como una forma de crear imágenes nuevas con diferentes orientaciones [17].

Ya que las bases de datos contienen imágenes de diferentes tamaños, se hace un recorte aleatorio de la imagen de entrenamiento, el recorte es del mismo

Fig. 2. Ejemplo de imágenes modificadas para el ingreso a la CNN.

tamaño que acepta la capa de entrada de la red; es decir, de $224 \times 224 \times 3$. La imagen de entrada es reflejada de manera horizontal, de izquierda a derecha, con un 50 % de probabilidad. El brillo y el contraste de la imagen se ajustan de manera aleatoria en intervalos del 63 % y $0{,}2 - 1{,}8$, respectivamente. Finalmente la imagen es normalizada para que la red tenga cierta independencia de las propiedades de la imagen. En la figura 2, se muestran algunos ejemplos de imágenes modificadas con el aumento de datos.

5. Experimentos y resultados

Para evaluar el rendimiento, las redes se entrenaron desde cero con cada una de las bases de datos y se utilizó aumento de datos. Los conjuntos de entrenamiento fueron formados con el 70 % de las imágenes de cada base de datos y el 30 % restante se utilizó para formar los conjuntos de prueba; la selección de las imágenes se realizó de manera aleatoria. Para separar los conjunto de entrenamiento y prueba, se utilizó el toolbox desarrollado en [4], para crear archivos de tipo tfrecord para evitar el desbordamiento de la memoria al realizar el entrenamiento. En la tabla 2 se muestran las características de cada conjunto de datos utilizadas para realizar los experimentos en este proyecto.

Tabla 2. Conjuntos de imágenes separadas en entrenamiento y pruebas, utilizando el 70 % de imágenes por clase para entrenamiento y el 30 % para pruebas.

Conjunto de datos	**Clases**	**Cantidad de imágenes**		
		Entrenamiento	**Prueba**	**Total**
Oliva & Torralba	8	1,879	809	2,688
Stanford Dogs	120	14,358	6,222	20,580
Caltech 256	257	21,314	9,293	30,607

Para describir la precisión de las redes en la tarea de clasificación, utilizamos los términos **top-1** y **top-5**. El número **top-1** es la cantidad de veces que la

etiqueta correcta tiene la probabilidad más alta predicha por la red. El número **top-5** es la cantidad de veces que la etiqueta correcta está dentro de las 5 clases principales predichas por la red.

5.1. Evaluación de la red propuesta

En la figura 3 se muestran los resultados de entrenamiento de la red propuesta, la cual a partir de este momento llamaremos ToniNet para diferenciarla de las otras redes. Note que la red llega a una exactitud del 100 % para el conjunto de entrenamiento Oliva & Torralba, mientras que para los conjuntos Stanford Dogs y Caltech 256 alcanza solo el 98 %.

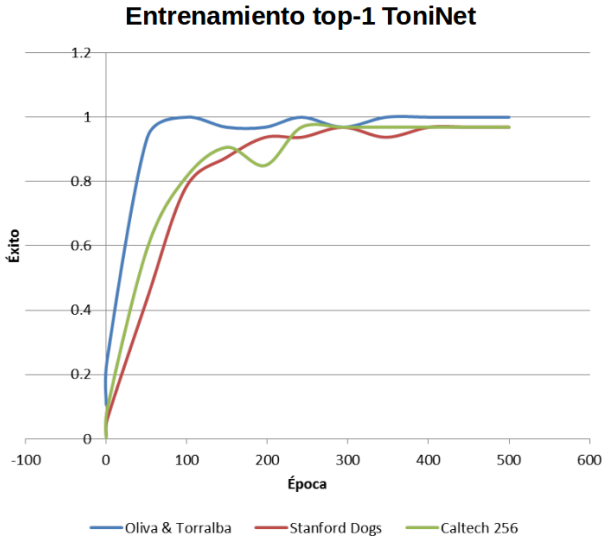

Fig. 3. Resultados de precisión top-1 en el entrenamiento de la red ToniNet.

Para evitar el sobre ajuste de la red ToniNet, se incluyeron capas de regularización (Dropout) en las dos primeras capas completamente conectas y se utilizó el aumento de datos. En la figura 4, se muestra los resultados obtenidos en la evaluación del sobreajuste de la red. Note que el error en la fase de entrenamiento disminuye de manera constante conforme aumenta el número de épocas, lo que refleja el crecimiento constante del rendimiento (Exactitud) de aprendizaje de la red.

5.2. Comparación

En la tabla 3 se muestra un resumen de los resultados para la etapa de prueba de las redes AlexNet, GoogleNet, ResNet 152 y ToniNet. Note que

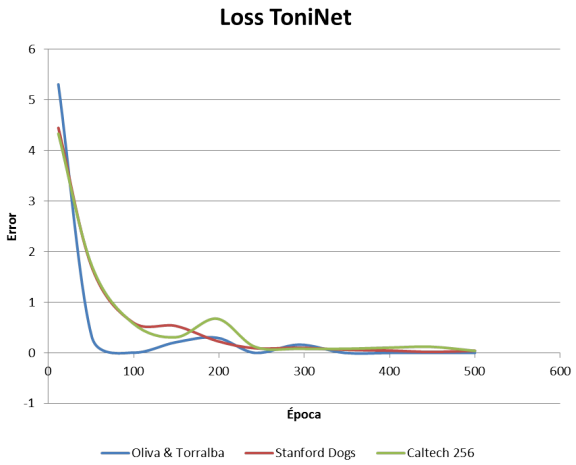

Fig. 4. Error de entrenamiento para la red ToniNet.

ToniNet supera a AlexNet y a GoogleNet tanto en top-1 como en top-5 para los 3 conjuntos de prueba; sin embargo, ToniNet es superada por ResNet 152 solo con el conjunto de prueba Caltech 256. GoogleNet supera en precisión a AlexNet ya que es una red con mayor profundidad; es decir, es una red que cuenta con un número mayor de capas de extracción; por la misma razón, ResNet 152 supera a GoogleNet. Sin embargo, aunque ResNet 152 también tiene una profundidad mayor a ToniNet, está solo supera a ToniNet en precisión con Caltech 256; lo que enfatiza tanto la ventaja como la importancia de tener tres secciones de extracción de características en paralelo idénticas pero con filtros de diferentes tamaños.

Tabla 3. Resultados y comparación de redes

CNN	**Oliva & Torralba**			**Stanford Dogs**			**Caltech 256**		
	Minutos	**Top-1**	**Top-5**	**Minutos**	**Top-1**	**Top-5**	**Minutos**	**Top-1**	**Top-5**
AlexNet	19	90,4 %	99,8 %	175	43,2 %	70,8 %	213	57,5 %	74,7 %
GoogleNet	14	91,5 %	99,8 %	164	51,9 %	80,2 %	196	60,6 %	79,0 %
ResNet 152	85	92,8 %	99,8 %	538	53,6 %	82,7 %	739	64,7 %	81,8 %
ToniNet	34	94,6 %	100 %	255	55,2 %	84,7 %	371	62,5 %	80,8 %

En la tabla 3 también se muestran los tiempos de entrenamiento de las redes. GoogleNet es la red con el menor tiempo de entrenamiento seguida de AlexNet, ToniNet y ResNet 152. Tomando como referencia el mayor tiempo de entrenamiento, que corresponde a ResNet 152, para las tres bases de datos podemos comentar que en promedio: ToniNet tiene un tiempo de entrenamiento del 18.03 % mayor a AlexNet y un 21.3 % mayor a GoogleNet, pero un 54.13 % menor a ResNet 152.

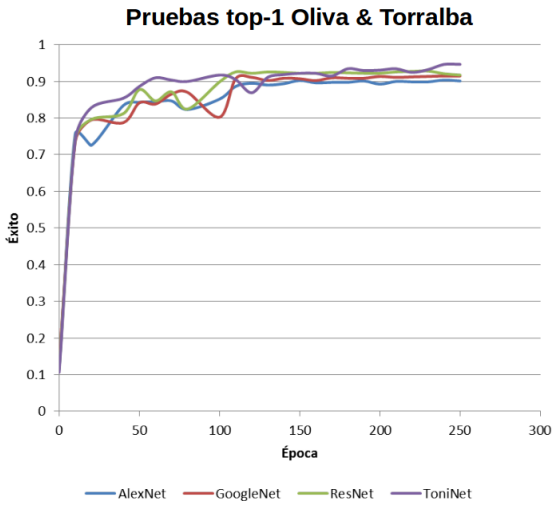

Fig. 5. Resultados de precisión top-1 (Oliva & Torralba).

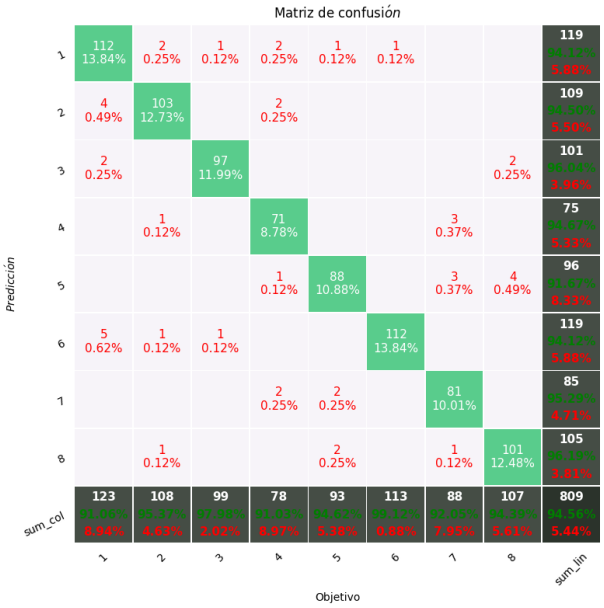

Fig. 6. Matriz de confusión de la red ToniNet (Oliva & Torralba).

- **Resultados con Oliva & Torralba:** Esta es la base de datos relativamente más simple que se utilizo, consta de 1,879 imágenes de entrenamiento y 809 de prueba pertenecientes a 8 clases. Analizando el comportamiento de

las redes en la fase de prueba, en la figura 5, podemos observar que las 4 redes aprenden muy bien y alcanzan una buena precisión top-1 en tan solo 150 épocas. Sin embargo, vale la pena mencionar que ToniNet muestra una mejor generalización durante toda la fase de aprendizaje y obtiene el mejor rendimiento con un 94,6 % de éxito en la época 250. Con esta base de datos ToniNet supera en 4,2 % a AlexNet, en 3,1 % a GoogleNet y en 1,8 % a ResNet 256.

Para visualizar mejor el desempeño de ToniNet, en la figura 6 se muestra la matriz de confusión con $C1, C2, ..., C8$ que corresponden a las clases *Opencountry, Coast, Forest, Highway, Inside_city, Mountain, Street* y *Tallbuilding*, respectivamente. En la diagonal principal se observa que la red obtiene un 94,56 % de éxito con 765 imágenes correctas de 809.

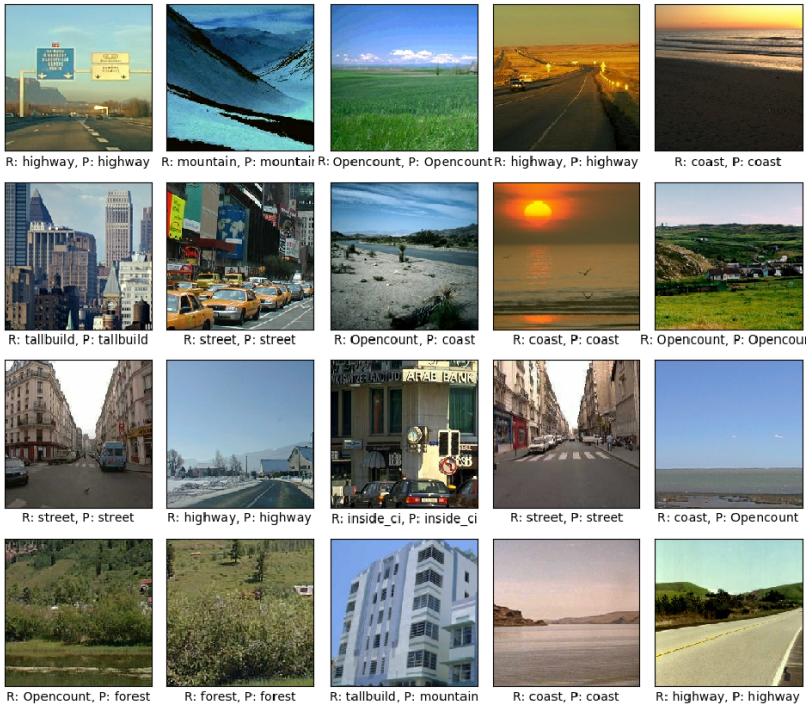

Fig. 7. Prueba con Oliva & Torralba, la letra R significa que es la clasificación real, la letra P significa que es la clasificación dada por la red.

La red tiene la mejor predicción con la clase *Tallbuilding* con el 96,19 % seguida de *Forest* con el 96,04 %.

La clase con menor éxito de predicción es *Inside_city* con 91,67 %, confundiéndose con *Street* y *Tallbuilding*. La clase real *Highway* es la más confundida por la red con *Opencountry*, *Coast* y *Street*, también *Opencountry*

es confundida con *Coast*, *Forest* y *Mountain*. *Mountain* es la clase que mejor aprende la red. En la figura 7 se muestran algunos de los resultados en la fase de prueba, en donde se puede observar que la red ToniNet confundió una imagen de la clase *coast* por una de la clase *Opencountry*, entre otras que clasificó de forma incorrecta.

- **Resultados con Stanford Dogs:** Esta base de datos es mas complicada que Oliva & Torralba ya que el tamaño de las imágenes es muy variante, contiene 14, 358 imágenes de entrenamiento y 6, 222 de prueba pertenecientes a 120 clases.

  Analizando el comportamiento de las redes en la fase de prueba, en la figura 8, podemos observar que ToniNet tiene el mejor rendimiento con un 55,2 % de éxito alcanzado en 250 épocas y supera en 12 % a AlexNet, en 3,3 % a GoogleNet y en 1,6 % a ResNet 256. En esta prueba, ToniNet se llegó a entrenar con 500 épocas logrando un 56,2 % como máximo en rendimiento en prueba; sin embargo, el pequeño incremento en el rendimiento es poco significativo comparado al tiempo de entrenamiento.

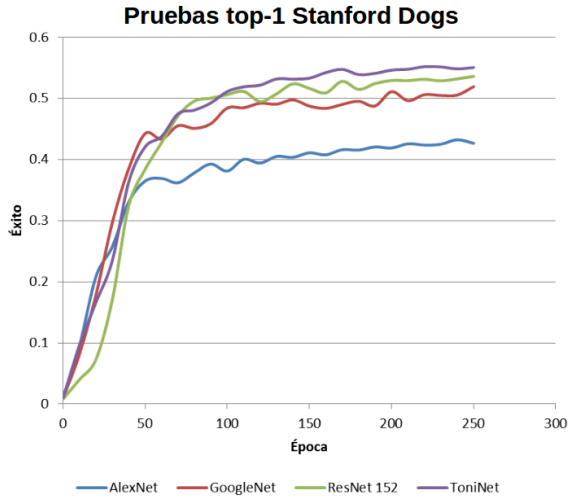

Fig. 8. Resultados de precisión top-1 (Stanford Dogs).

- **Resultados con Caltech 256:** Esta base de datos es más desafiante que Oliva & Torralba y Stanford Dogs, ya que tiene el mayor desequilibrio de imágenes por categoría, contiene 21, 314 imágenes de entrenamiento y 9, 293 de prueba pertenecientes a 256 clases. Analizando el comportamiento de las redes en la fase de prueba, en la figura 9, podemos observar que ResNet 152 obtuvo el mejor rendimiento con un 64,7 % de éxito en 250 épocas, superando a ToniNet en un 2,2 %. Con esta base de datos ToniNet supera en 5,0 % a AlexNet, en 1,9 % a GoogleNet.

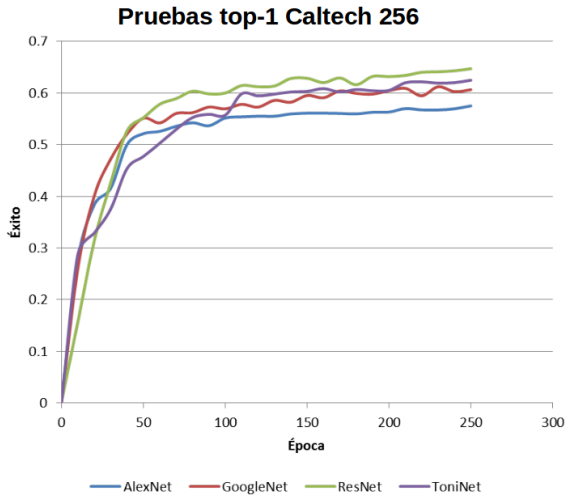

Fig. 9. Resultados de precisión top-1 (Caltech 256).

6. Conclusiones

La red propuesta dio mejores resultados con el conjunto de imágenes de Oliva & Torralba y StanfordDogs comparándola con AlexNet, GoogleNet y ResNet 152. Fue superada por ResNet 152 con el conjunto de imágenes Caltech 256, pero el tiempo de entrenamiento de nuestra red fue menor y con mayor rendimiento que AlexNet y GoogleNet.

La ventaja de la red con respecto a las comparadas; es que tiene diferentes secciones de extracción de características, con esto necesita menos capas, como por ejemplo: ResNet 152 y GoogleNet. La desventaja principal es que necesita más tiempo de entrenamiento que AlexNet y GoogleNet para lograr mejores resultados pero con menor tiempo que ResNet.

Se ponen las bases para la construcción de redes usando diferentes secciones para la extracción de características. Se pueden añadir más secciones dependiendo del hardware con el que se cuenta. Las redes profundas como ResNet con 152 capas extraen las características solo de una sección, en la red propuesta se pueden extraer características de múltiples secciones.

La red se puede mejorar en la etapa de clasificación, principalmente en las capas totalmente conectadas, también ampliando las secciones.

Como trabajo futuro se planea utilizar la red para la detección de objetos y segmentación de imágenes. También se planea experimentar con más secciones; para trabajar con el color, la forma, la textura, entre otras.

Referencias

1. Caltech256: Caltech 256 Dataset. www.vision.caltech.edu/ImageDatasets/Caltech256 (Mayo 2016)
2. Ciresan, D.C., Meier, U., Masci, J., Gambardella, L.M., Schmidhuber, J.: Flexible, high performance convolutional neural networks for image classification. In: Twenty-Second International Joint Conference on Artificial Intelligence (2011)
3. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: international Conference on computer vision & Pattern Recognition (CVPR'05). vol. 1, pp. 886–893. IEEE Computer Society (2005)
4. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision. pp. 1026–1034 (2015)
5. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence 37(9), 1904–1916 (2015)
6. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: Convolutional architecture for fast feature embedding. In: Proceedings of the 22nd ACM international conference on Multimedia. pp. 675–678. ACM (2014)
7. Kaiming He, Xiangyu Zhang, S.R.J.S.: Deep residual learning for image recognition. IEEE Xplore (2015)
8. Khosla, A., Nityananda, Jayadevaprakash, Yao, B., Fei-Fei, L.: Stanford Dogs Dataset. http://vision.stanford.edu/aditya86/ImageNetDogs/ (Septiembre 2017)
9. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
10. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature 521, 436–44 (05 2015)
11. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision 60(2), 91–110 (2004)
12. López, F., Ferreyra, A., Avilés, C., Villegas, J., Zúñiga, A., Rodriguez, E.: Preprocesamiento de bases de datos de imágenes para mejorar el rendimiento de redes neuronales convolucionales. Research in Computing Science 147(7): Robotics and Computer Vision, 35–45 (2018)
13. Mairal, J., Koniusz, P., Harchaoui, Z., Schmid, C.: Convolutional kernel networks. In: Advances in neural information processing systems. pp. 2627–2635 (2014)
14. Oliva, Torralba: . http://cvcl.mit.edu/database.htm (Mayo 2016)
15. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
16. 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. pp. 1–9 (2015)
17. Taylor, L., Nitschke, G.: Improving deep learning using generic data augmentation. CoRR abs/1708.06020 (2017), http://arxiv.org/abs/1708.06020
18. Wan, L., Zeiler, M., Zhang, S., Le Cun, Y., Fergus, R.: Regularization of neural networks using dropconnect. In: International conference on machine learning. pp. 1058–1066 (2013)
19. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European conference on computer vision. pp. 818–833. Springer (2014)