Información Tecnológica
Vol. 28(3), 209-224 (2017)
doi: 10.4067/S0718-07642017000300021

Clasificación de Imágenes Urbanas Aéreas: Comparación entre Descriptores de Bajo Nivel y Aprendizaje Profundo

Antonio Arista-Jalife, Gustavo Calderón-Auza, Atoany Fierro-Radilla, Mariko Nakano*
Sección de Estudio de Posgrado e Investigación, ESIME-Culhuacán, Instituto Politécnico Nacional.
Av. Santa Ana No. 1000, Col. San Francisco Culhuacán, Coyoacán, Ciudad de México, C.P. 04420.
(e-mail: arista.antonio@gmail.com, gus_auza@hotmail.com, afierror@hotmail.com, mnakano@ipn.mx *)

* Autor a quien debe dirigirse la correspondencia

Recibido Nov. 11, 2016; Aceptado Ene. 12, 2017; Versión final Feb. 6, 2017, Publicado Jun. 2017

Resumen

En este artículo se presenta una comparación entre diferentes algoritmos de descripción de texturas de bajo nivel acoplados con una máquina de soporte vectorial versus el algoritmo de aprendizaje profundo, en la tarea de reconocimiento y clasificación de imágenes aéreas. Para dicha tarea, una base de datos de 1,200 imágenes es utilizada para realizar los entrenamientos supervisados. El objetivo consiste en clasificar las imágenes en seis categorías comúnmente encontradas en zonas urbanas, de tal manera que pueda ser utilizado en cualquier parte del mundo. Los resultados arrojan que con 150 muestras de cada clase, el algoritmo de aprendizaje profundo es capaz de clasificar imágenes de avenidas, edificios, industrias, zonas naturales, zonas residenciales y cuerpos de agua, con un 87% de exactitud. Los resultados experimentales presentados muestran que las imágenes etiquetadas como edificios e industrias son las más complejas de discernir entre ellas, tanto para descriptores de bajo nivel como para las técnicas de aprendizaje profundo.

Palabras clave: aprendizaje profundo; máquina de soporte vectorial; imágenes aéreas; descriptores de texturas; base de datos

Classification of Urban Aerial Images: A Comparison between Low-Semantic Descriptors and Deep Learning

Abstract

This paper presents a comparison between different low-semantic descriptive algorithms coupled with a support vector machine and the deep learning algorithm, for the task of recognition and classification of aerial images. For this task, a database composed of 1200 images is used to fulfill the supervised trainings. The objective consists on classifying images in six categories that are commonly found on urban areas, in order to be used in any part of the world. The results show that with 150 samples of each class, the deep learning algorithm is capable of classifying images of avenues, buildings, industries, natural areas, residential areas and water bodies with an 87% of accuracy. Experimental results also prove that the labeled images as industry and buildings are the most complex ones to distinguish among these two classes, both for low-level descriptors and deep learning techniques.

Keywords: deep learning; support vector machine; aerial images; texture descriptors; database

INTRODUCCIÓN

Con el crecimiento de zonas urbanas y el desarrollo de vehículos aéreos no tripulados (UAV por sus siglas en inglés), su utilización para obtener información del terreno urbano en imágenes y videos se ha visto incrementado drásticamente en los últimos años. Las imágenes obtenidas pueden ser utilizadas para un amplio rango de aplicaciones, desde indicadores de desarrollo en las ciudades hasta útiles mecanismos que ayuden a la planeación de nuevos servicios como transporte público o despliegue de seguridad (Maza et al., 2011). De igual manera, los UAV pueden ser utilizados en zonas de desastre para determinar puntos que se encuentran damnificados, ya que dichos vehículos podrían sobrevolar áreas de difícil acceso (Waharte y Trigoni, 2010). Sin embargo, debido a que en un solo vuelo pueden ser capturadas una enorme cantidad de imágenes, el clasificar e interpretar cada una de ellas puede ser una tarea compleja, sobre todo en situaciones de desastre donde el tiempo es un factor determinante y valioso (Fierro et al., 2015; Rui et al., 1999).

Un sistema que sea capaz de interpretar y clasificar imágenes de diferentes zonas geográficas conlleva el transformar características semánticamente simples y observables -como son el color, textura y forma- a características conceptualmente complejas –como son árboles, cuerpos de agua o avenidas-. Esta diferencia entre características de bajo nivel y de alto nivel se le conoce como brecha semántica (Wan et al. 2014). Un adecuado sistema de clasificación debe ser capaz de cerrar la brecha semántica lo más posible, por medio de descriptores que extraigan la información de la matriz de píxeles de una imagen y la condensen en conceptos utilizables (Datta et al., 2008). Si un sistema automático es capaz de difuminar o suprimir la brecha semántica, cualquier imagen presentada podrá ser adecuadamente descrita.

En este artículo se presentan diferentes métodos de descripción y clasificación basados en descriptores de texturas, y un método de clasificación relativamente novedoso conocido como *aprendizaje profundo*, el cual cuenta con la ventaja de no requerir algoritmos descriptivos (Nielsen, 2015). En otras palabras, los algoritmos de aprendizaje profundo solamente necesitan imágenes de entrada para realizar una clasificación por aprendizaje supervisado, mientras que otros algoritmos requieren forzosamente sistemas que describan la información insertada con antelación. Para llevar a cabo los entrenamientos supervisados, una nueva base de datos de 1,200 imágenes aéreas es presentada.

Para la tarea de reconocimiento y clasificación de imágenes capturadas vía aérea, se requiere una colección de imágenes previamente categorizada en clases que servirán como base para el entrenamiento, validación y prueba de los sistemas que se propongan. Comúnmente esta categorización se hace manualmente o con supervisión humana y es deseable que la base de datos contenga la misma cantidad de imágenes por cada una de las clases, de tal manera que una clase no reciba más muestras de entrenamiento, validación y prueba que otra, afectando su eficacia para procesar clases menos pobladas. Existen diversas bases de datos que pueden ser utilizadas para realizar la tarea de clasificación de imágenes aéreas, sin embargo, algunas de ellas presentan la particularidad de tener una cantidad desigual de imágenes por cada clase, como es el caso de la base de datos de Banja Luka (Risojević et al, 2011). Otras bases de datos tienen fotografías de poco tamaño, las cuales hacen perder detalles relevantes. Y otras más contienen un gran número de clases que no son relevantes para aplicaciones en zonas geográficas diferentes a donde fueron extraídas, como por ejemplo la base de datos de Merced (Yi y Shawn, 2010), la cual contiene entre otras clases, canchas de tenis, campos de golf y estadios de béisbol.

Debido a ello, la base de datos que fue utilizada para estos experimentos consiste en 1,200 imágenes separadas equitativamente en 6 clases: 1) Avenidas y caminos, 2) Edificios 3) Industrial 4) Naturaleza y agricultura 5) Residencial y 6) Cuerpos de agua. Cada una de estas clases contiene 200 imágenes de resolución de 256 x 256 píxeles en escala de valores RGB. Debido a que muchas de las ciudades del mundo contienen casi todas las clases mencionadas, esta base de datos puede ser fácilmente utilizable para diferentes zonas geográficas. La base de datos combina muestras de la base de datos de Merced y Banja Luka, en Bosnia y Herzegovina, y además inserta imágenes extraídas del software de creación de mapas ArcGIS correspondientes a varias zonas de la Ciudad de México y de Nueva York. Dicha base de datos se encuentra disponible para su uso irrestricto en futuros experimentos y líneas de investigación.

DESCRIPTORES

Un descriptor es un conjunto de valores en los que se pretende que elementos similares posean valores iguales o muy cercanos a iguales cuando son sometidos a un algoritmo de descripción. En el caso de clasificación de imágenes, dos imágenes idénticas deben poseer los mismos descriptores, dos imágenes similares deben poseer dos descriptores con poca distancia numérica entre ellos, y por ende, dos imágenes enteramente diferentes deben poseer una amplia distancia numérica (Manjunath et al., 2002).

Cuando un descriptor logra realizar una distinción adecuada, la tarea de categorizar se vuelve más sencilla y genera mejores resultados, es por ello que la adecuada elección de un descriptor -o conjunto de ellos- es una tarea crucial para la clasificación (Calderón et al. 2016). Acotando el problema al dominio de imágenes, los descriptores pueden englobarse en tres ramas: basados en colores, en texturas, y en formas (Castelli y Bergman, 2002). Debido a la naturaleza del problema de clasificación de imágenes aéreas, los descriptores más adecuados para llevar a cabo la tarea son los basados en texturas, ya que las estructuras naturales y artificiales en las imágenes presentan a simple vista cierta similitud en sus texturas.

En este artículo se utilizarán tres descriptores basados en texturas y/o colores para la clasificación de las 6 clases mostradas en estos experimentos. Algunos descriptores utilizan como base un concepto llamado Motif (Calderón et al., 2016). Un Motif puede definirse como un patrón de la intensidad de una ventana de pixeles en escala de grises, de tamaño 2x2 en una imagen. Para ejemplificar cómo funciona el algoritmo de Motif, suponga que existe una ventana de 2x2 pixeles, cada uno con un valor en el rango de 0 a 255. Si se toma como punto de partida en todos los casos el elemento (1,1) de la ventana, el algoritmo consiste en elegir de los 3 pixeles restantes aquel que posee el valor más cercano al valor contenido en (1,1). Una vez elegido el segundo elemento, se elige el tercero con el mismo proceso –el valor más cercano al segundo- y el cuarto como el elemento restante. En la figura 1 puede observarse un ejemplo de ello.

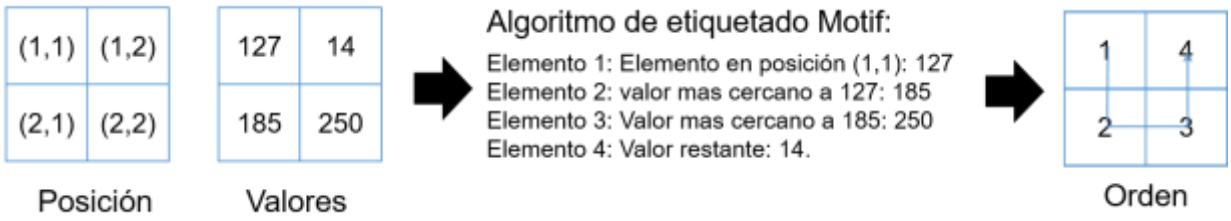

Fig. 1: Algoritmo de etiquetado Motif.

Es posible deducir que a partir del algoritmo de etiquetado Motif descrito, cualquier combinación de pixeles puede reducirse a solamente seis posibles casos, con lo que puede asignarse un número a cada uno de los casos mencionados, como se muestra en la figura 2.

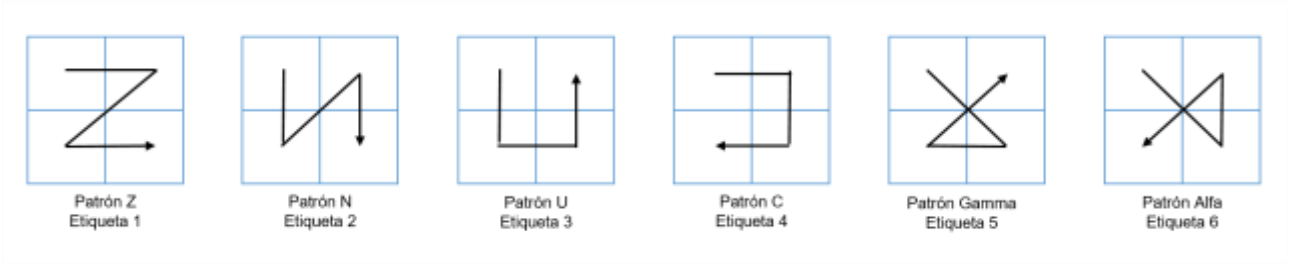

Fig. 2: Los 6 posibles casos de los patrones Motif.

Con este algoritmo, es posible dividir una imagen en escala de grises en ventanas no traslapadas de 2x2 pixeles, de tal manera que cada 2x2 píxeles puede ser etiquetado con una de las 6 etiquetas de la figura 2, reduciendo una imagen de M x N píxeles a una matriz de (M/2) x (N/2) valores. Un ejemplo de ello se puede apreciar en la figura 3. Por medio de la reducción de una imagen en escala de grises a un patrón Motif, es posible obtener un descriptor resistente a la rotación.

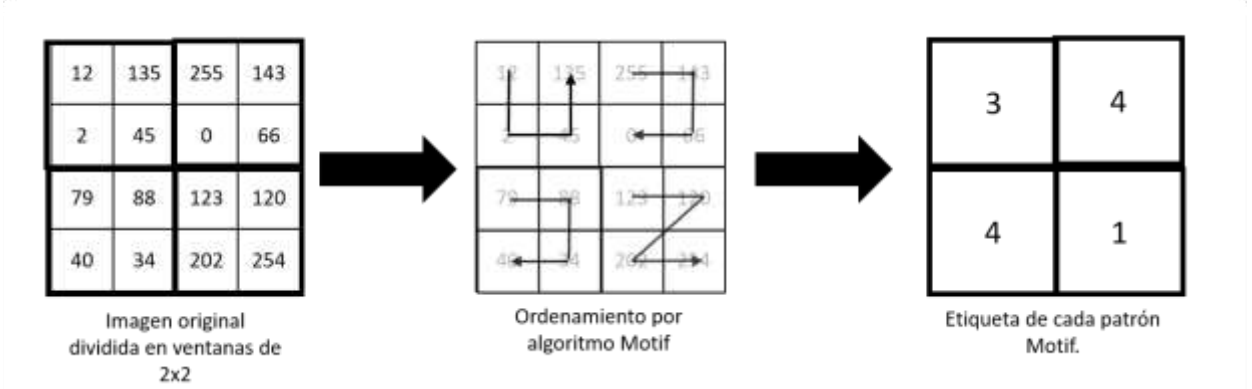

Fig. 3: Ejemplo de un algoritmo para transformar una imagen en escala de grises a un patrón Motif

Matriz de coocurrencia de Motif Direccional (MCMD): La manera de calcular la Matriz de co-ocurrencia de un patrón Motif consiste en buscar cuantas veces aparece una etiqueta a lado de otra. Un ejemplo del cálculo de la matriz de coocurrencia puede apreciarse en la figura 4. Nótese en dicha figura que las etiquetas contiguas 1-1 aparecen 3 veces, las etiquetas contiguas 1-2 aparecen 9 veces, y así consecutivamente hasta llenar la matriz de coocurrencia. La matriz resultante es el descriptor de la imagen original.

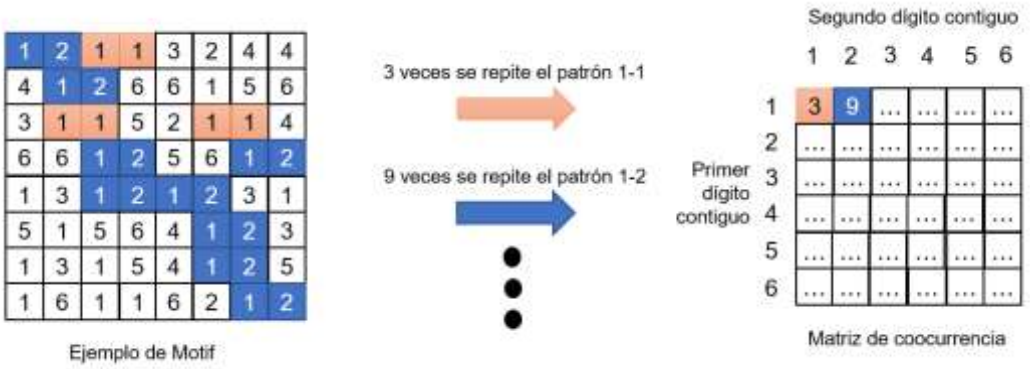

Fig. 4: Ejemplo de algoritmo de Matriz de Coocurrencia de Motif.

Es posible observar que si una imagen se ve afectada por cierta rotación, el patrón de Motifs puede verse alterado y con ello el descriptor a utilizarse. Para aminorar el impacto de la rotación de la imagen, El algoritmo de descripción MCMD considera el elemento contiguo a la derecha (0°), y los elementos que se encuentran a 45°, 90°, y 135° (Calderón et al., 2015). Tal como se muestra en la figura 5.

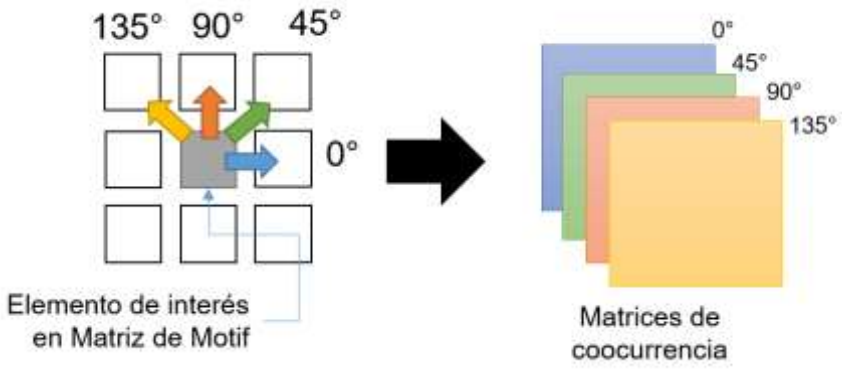

Fig. 5: Elementos a considerar para construir las matrices de coocurrencia.

Patrón local binario uniforme invariante a rotación (LBP-UIR): Los patrones locales binarios fueron propuestos en primera instancia por Ojala et al., 2002, con la finalidad de describir texturas en una imagen. Dicho algoritmo es utilizado para diversas áreas del procesamiento de imágenes y visión computacional, tales como el reconocimiento de humo, detección de rostros, y otros. El patrón local binario consiste en calcular la relación entre un pixel g_c y sus vecinos $g_i \mid i = 0,1,\dots,P$ los cuales se encuentran en un radio de vecindad R. La forma en la que el patrón local binario se calcula es por medio de la siguiente fórmula:

$$LBP_{P,R}(g_i,g_c)= \sum_{i=0}^{P-1} 2^i f(g_i - g_c) \quad (1)$$

Donde la función $f(x)$ es una función por partes con la siguiente estructura:

$$f(x)= \begin{cases} 1, & x\geq 0 \\ 0, & x<0 \end{cases} \quad (2)$$

Del concepto del patrón local binario parte una variante uniforme resistente a la rotación. (LBP-UIR por sus siglas en inglés), el cual genera el mismo valor descriptivo sin importar si la imagen se encuentra sujeta a una rotación. Para crear este histograma, es necesario el asegurar su uniformidad por medio de calcular la cantidad de intercambios de valores de 1 a 0 o viceversa que existen entre los valores vecinos en el vector generado por la función $LBP_{P,R}(g_i, g_c)$ de la ecuación 1. Para que un patrón se considere uniforme, no deben existir más de dos cambios de 1 a 0 o de 0 a 1. Más detalles del cálculo de uniformidad pueden ser consultados en (Ojala et al., 2002).

Cuando un patrón no es uniforme, puede ser etiquetado directamente con cualquier valor previamente asignado, como por ejemplo el valor -1. Si el patrón es uniforme, los bits pueden acarrearse en una rotación circular hasta alcanzar el mínimo valor posible, debido a la uniformidad, el corrimiento mencionado solo puede generar vectores de bits con los valores decimales 0, 1, 3, 7, 15, 31, 63, 127 y 255. Por tanto, cualquier patrón estará descrito con alguno de estos 9 valores decimales sin importar si la imagen está sujeta a rotación, lo que permite generar un descriptor en forma de histograma. Los valores obtenidos del histograma pueden ser utilizados como un descriptor de 10 números enteros, con los cuales una imagen puede ser descrita con invariancia a la rotación por medio de este algoritmo.

Filtros de Gabor: Los filtros de Gabor se utilizan para clasificar y segmentar la textura en las imágenes (Manjunath et al, 2001). Suelen presentar un adecuado desempeño cuando son utilizados como descriptores, sin embargo presenta la desventaja de tener un elevado costo computacional en comparación con otra clase de algoritmos descriptivos. Un filtro de Gabor puede definirse por la siguiente ecuación:

$$g(x,y)= \left(\frac{1}{2\pi\sigma_x\sigma_y}\right)\exp\left[-\frac{1}{2}\left(\frac{(x_0-x')^2}{\sigma_x{}^2}+\frac{(y_0-y')^2}{\sigma_y{}^2}\right)+2\pi j\omega x'\right] \quad (3)$$

$$x'=x\cos\phi+y\sin\phi$$
$$y'=-x\sin\phi+y\cos\phi \quad (4)$$

La variable ω representa la frecuencia espacial del filtro, la variable ϕ representa la orientación deseada del filtro, Las variables x_0 y y_0 definen el centro de la ventana considerada para aplicar el filtro, las variables σ y σ^2 representan la desviación estándar y la varianza respectivamente, y la variable j es el valor imaginario $j = \sqrt{-1}$. Una de las peculiaridades del filtro de Gabor es que la modificación de la orientación, varianza y frecuencia espacial pueden modificar la textura que procura obtenerse. Debido a que es posible que existan demasiadas variaciones y combinaciones de filtros de Gabor, Manjunath y Ma (1996) proponen limitar las configuraciones a solamente 24, las cuales se obtienen por la combinación de las orientaciones $\phi = \{0°, 30°, 60°, 90°, 120°, 150°\}$ y las frecuencias $\omega = \{0, 1, 2, 3\}$ de tal forma que el descriptor se defina en un solo vector de parámetros P del filtro de Gabor

$$P= \{\phi_0\omega_0, \phi_0\omega_1, \phi_0\omega_2 \ldots \phi_1\omega_0, \ldots \phi_5\omega_3\} \quad (5)$$

MÉTODOS DE CLASIFICACIÓN

Aun cuando los descriptores mencionados se enfocan en englobar las características de color, formas y/o texturas de una imagen, se requiere un mecanismo que utilice esas características descriptivas y encuentre un patrón común entre ellas, de tal manera que imágenes similares –que a su vez generan descriptores similares- puedan ser categorizadas de manera automática. Si dicho objetivo logra alcanzarse, nuevas colecciones de imágenes podrán ser etiquetadas de forma adecuada sin intervención humana, lo que permitirá realizar un censado automático del terreno cuando un UAV capture ciertas áreas.

Para efectos de este artículo, se prevé el uso de dos métodos de clasificación: Máquina de soporte vectorial y Aprendizaje profundo. La máquina de soporte vectorial (en inglés *Support Vector Machine* o SVM). Es un algoritmo de clasificación que, dado un vector de características n-dimensional, intenta predecir a que clase pertenece por medio de la generación de un hiperplano de separación (n-1)-dimensional (Cortes, 1995). Para efectos de este trabajo, la máquina de soporte vectorial implementa los descriptores mencionados para realizar una tarea de clasificación multiclase, mientras que la técnica de aprendizaje profundo utiliza cierto porcentaje de las imágenes directamente para aprender y reconocer patrones relevantes en la imagen que se presenta.

Aprendizaje profundo

El aprendizaje profundo (o Deep Learning) es un mecanismo de clasificación relativamente novedoso el cual está basado en redes neuronales convolucionales (Ciresan et al. 2011; Nielsen, 2015). Esta técnica presenta la valiosa característica de que no requiere forzosamente de descriptores para realizar la tarea de clasificación, ya que al presentarse imágenes directamente a los sistemas de aprendizaje profundo, los algoritmos infieren las características relevantes y los implementan como filtros de textura, color o forma. Debido a esta valiosa característica, los algoritmos de aprendizaje profundo han sido utilizados con antelación para tareas de reconocimiento de imágenes (Krizhevsky et al. 2012; Wan et al. 2014; Simonyan 2014).

El aprendizaje profundo consiste en realizar operaciones de convolución en una imagen, y resumir las características más relevantes, acentuando aquellas que tienen mayor preponderancia. Cuando varias capas de procesamiento se conectan de forma lineal, se logra un proceso muy similar al de las redes neuronales de segunda generación (*E.g*. las redes neuronales artificiales de tipo perceptrón multicapa, o las redes MADALINE). En una red neuronal convolucional -también llamada ConvNet- las neuronas no se encuentran implementadas de forma lineal o vectorial, sino en una estructura bidimensional a manera de matriz. Debido a ello, la imagen de entrada también se considera una retícula de neuronas, donde cada neurona corresponde a un pixel de la imagen.

Cada neurona de una capa posterior cuenta con un rango de visión de la capa anterior, llamado campo local receptivo, en donde todas las neuronas contenidas dentro de ese campo contribuyen a la entrada de la neurona oculta, de tal manera que las neuronas de la capa anterior que presenten una activación adecuada, contribuirán a la entrada de la capa posterior, como se observa en la figura 6a. Dichos campos locales receptivos pueden superponerse. La cantidad de pixeles que la ventana del campo local receptivo se mueve se le conoce como paso o *stride* (como se muestra en la figura 6b). El objetivo de dichas neuronas en la capa posterior es detectar una cierta característica en un área de la imagen.

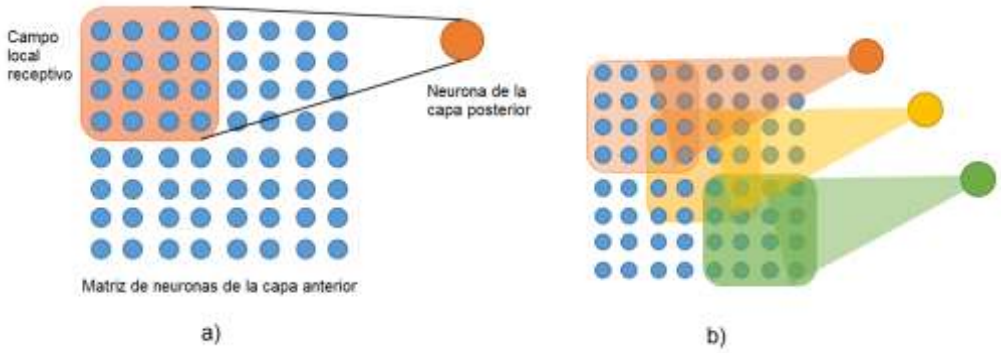

Fig. 6: a) Campo local receptivo; b) Varios campos locales receptivos superpuestos.

Cada una de las neuronas en una capa utilizará los mismos pesos sinápticos y bias. Por ende, para una neurona oculta, en una posición determinada (x, y), la salida s de dicha neurona es:

$$s = \sigma\left(b + \sum_{l=0}^{L}\sum_{m=0}^{M} w_{l,m}\, a_{(x+l,y+m)}\right) \tag{6}$$

Donde b es el bias de la capa de neuronas, $w_{l,m}$ es el peso sináptico contenido en la matriz de pesos sinápticos W, en la posición (l, m). La variable $a_{(x+l,y+m)}$ representa el valor de entrada de la matriz de la capa anterior A, en la posición $(x + l, y + m)$, y la función $\sigma(x)$ Es la función sigmoidea definida por

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{7}$$

Debido a que todas las neuronas utilizan los mismos pesos sinápticos, el objetivo de una matriz de neuronas en una capa posterior es detectar la misma característica en ventanas superpuestas y en diferentes posiciones de la imagen, lo que causa que las características detectables sean invariantes a traslación. Es por ello que las capas ocultas o las capas posteriores se le conocen como mapa de características. Comúnmente, diversos mapas se conectan con una sola imagen de entrada los cuales detectan solamente una característica, como se muestra en la figura 7.

Debido a que la información generada por un mapa de características puede tener una dimensión considerable, y puede incrementarse conforme más mapas se agreguen, es necesario resumir los mapas de características a una matriz que contenga solamente la información más relevante. Este proceso se le conoce como Pooling y consiste en generar una matriz de neuronas que posean un campo local receptivo no traslapado (Boureau et al., 2010). En el caso del algoritmo de Max Pooling, cada neurona extraerá solamente el valor máximo de cada ventana, ignorando el resto. Para el caso del algoritmo de Pooling por promedio, se obtiene el promedio de todas las neuronas en el campo local receptivo. Un ejemplo de ello se puede observar en la figura 8.

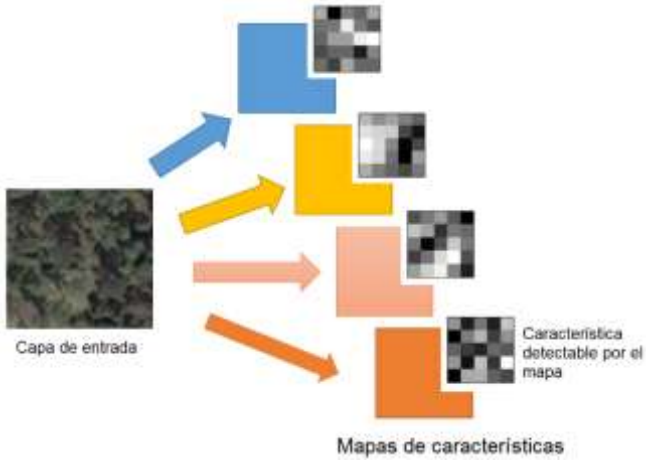

Fig. 7: Capa de entrada conectada a varios mapas de características. Diferentes mapas de características pueden activarse en paralelo.

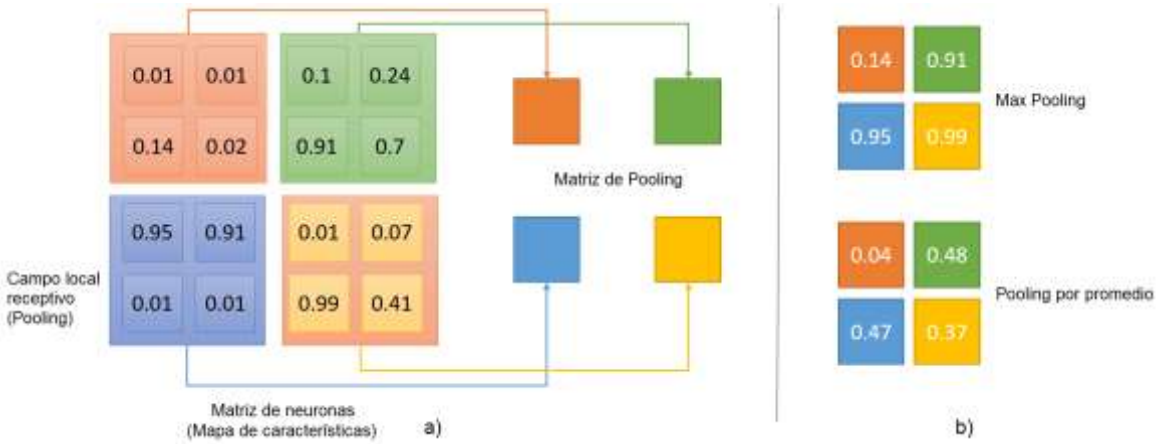

Fig. 8: a) Mecanismo de Pooling; b) resultado si el algoritmo utilizado es Max Pooling o Pooling por promedio.

Finalmente, una red neuronal multicapa puede ser conectada en las últimas capas de la red de aprendizaje profundo, sin embargo, este proceso requiere una transformación de la matriz de salida de la última capa a un vector de valores, de tal manera que dicho vector será presentado a una red neuronal multicapa, la cual llevará a cabo la última tarea de reconocimiento. Dichas redes convolucionales tienen la particularidad de que las capas más cercanas a la imagen buscan características de bajo nivel, como es el caso de formas, texturas y colores. Mientras que las capas más cercanas a la salida buscan características semánticas de alto nivel como son las estructuras industriales, zonas verdes y edificios. Cabe destacar que las redes neuronales convolucionales de aprendizaje profundo basan su eficacia en la cantidad de capas dentro de su topología. Entre más cantidad de capas posea una red convolucional, mejor será su desempeño en el reconocimiento, con la desventaja de un elevado costo computacional.

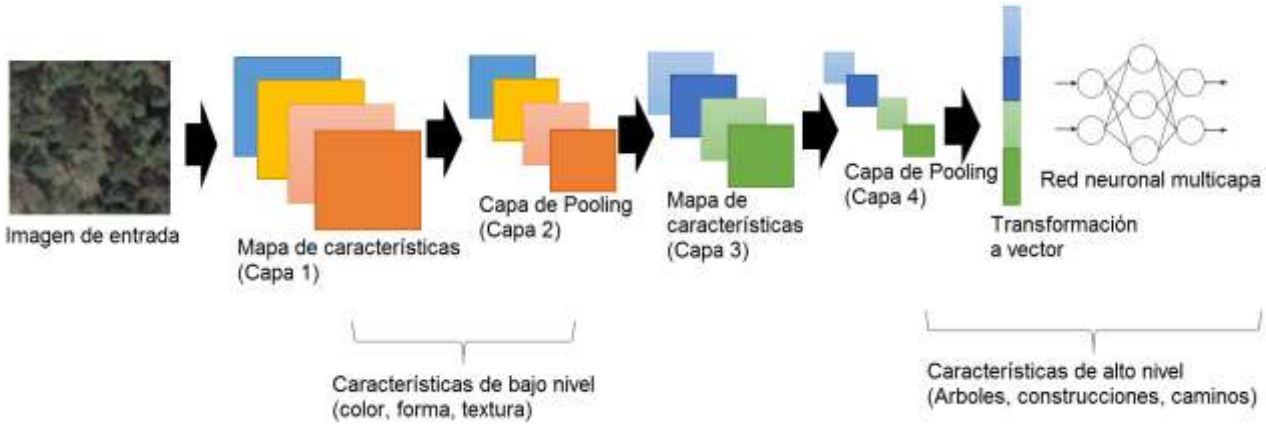

Fig. 9: Ejemplo de una topología de red neuronal convolucional.

Entrenamiento de una red de aprendizaje profundo

El entrenamiento de una red neuronal convolucional se basa en el algoritmo de retropropagación comúnmente utilizado en las redes neuronales multicapa de tipo perceptrón. Cuando una imagen de entrada es entregada al sistema, la última capa de la red neuronal recibe la salida y la compara con la salida esperada. La diferencia entre la salida esperada y la salida obtenida se le conoce como error (o pérdida), mismo que se retropropaga a las capas posteriores. La retropropagación en una red neuronal multicapa es bien conocida y ha sido descrita en numerosos artículos (Werbos, 1974, 1982) sin embargo hay varias consideraciones para las capas de Pooling y los mapas de características. Así como se presenta a la red neuronal multicapa un vector de datos de entrada, la red neuronal multicapa retropropaga un error de la misma dimensión del vector de entrada. Este error debe ser transformado inversamente a una matriz (o conjunto de matrices) para ser procesada, este error transformado de forma matricial puede ser etiquetado con la derivada parcial

$$\text{Error}= \frac{\partial E}{\partial y^l} \tag{8}$$

Donde y^l es la salida de la capa l. Dependiendo de la capa l, es como dicha matriz será tratada. Si la capa es una capa de Pooling, se debe realizar el proceso inverso, esto es, repartir el error en partes iguales para el caso del Pooling por promedio, o repartirlo en la máxima contribución, para el caso de Max Pooling. Un ejemplo de ello puede observarse en la figura 10.

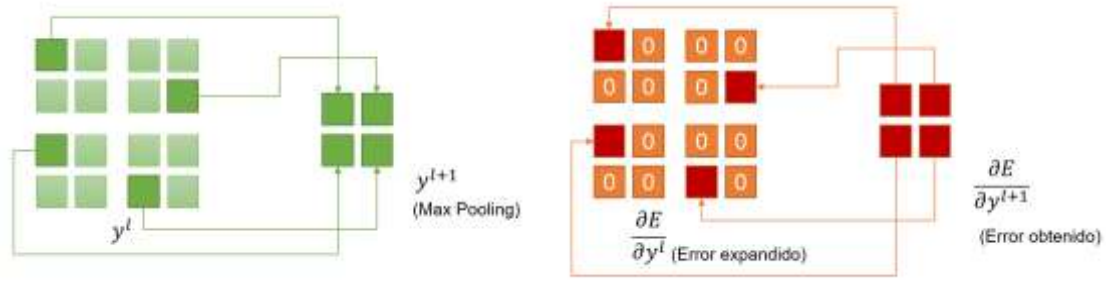

Fig. 10: Retropropagación del error en una capa de Max Pooling.

Para el caso de mapas de características se requieren dos procesos, el ajuste de pesos y la retropropagación del error. Para ajustar los pesos sinápticos, hay que considerar que cada mapa de características posee una matriz W, tal que cada elemento de dicha matriz es un peso sináptico definido con la variable w_k y puede ajustarse con la fórmula:

$$w_k \leftarrow w_k - \gamma \frac{\partial E}{\partial w_k} \tag{9}$$

Donde la variable γ es un factor de transferencia (también llamada tasa de aprendizaje), y su valor típico es de $\gamma = 0.01$. La derivada parcial del error E con respecto al peso sináptico w_k puede definirse por medio de la siguiente ecuación:

$$\frac{\partial E}{\partial w} = \sum_{i=0}^{N-M}\left[\sum_{j=0}^{N-M}\left[\frac{\partial E}{\partial x_{ij}^l} \cdot \frac{\partial x_{ij}^l}{\partial w}\right]\right] \tag{10}$$

Es posible definir que la derivada parcial $\frac{\partial x_{ij}^l}{\partial w}$ es la entrada x de la capa l que se estudia, la cual es la salida y de la capa anterior $l-1$, por ende, podemos realizar la substitución:

$$\frac{\partial x_{ij}^l}{\partial w} = y_{(i+a)(j+b)}^{l-1} \tag{11}$$

Por otro lado, por la regla de Lyapunov, la derivada parcial $\frac{\partial E}{\partial x_{ij}^l}$ puede descomponerse de la siguiente forma:

$$\frac{\partial E}{\partial x_{ij}^l} = \frac{\partial E}{\partial y_{ij}^l} \cdot \frac{\partial y_{ij}^l}{\partial x_{ij}^l} \tag{12}$$

Se puede corroborar que la salida y_{ij}^l está generada por la función sigmoidal $\sigma(x_{ij}^l)$, por lo cual podemos substituirla libremente de la ecuación 12 de la siguiente manera:

$$\frac{\partial E}{\partial x_{ij}^{l}} = \frac{\partial E}{\partial y_{ij}^{l}} \cdot \frac{\partial y_{ij}^{l}}{\partial x_{ij}^{l}} = \frac{\partial E}{\partial y_{ij}^{l}} \cdot \frac{\partial}{\partial x_{ij}^{l}} \sigma(x_{ij}^{l}) = \frac{\partial E}{\partial y_{ij}^{l}} \cdot \sigma'(x_{ij}^{l}) \tag{13}$$

Donde $\sigma'(x_{ij}^{l})$ es la derivada parcial de la función $\sigma(x_{ij}^{l})$ con respecto a la entrada, y la derivada parcial $\frac{\partial E}{\partial y_{ij}^{l}}$ es la matriz de errores que fueron retropropagados. Por lo que podemos hacer la substitución de la ecuación 10 de la siguiente forma:

$$\frac{\partial E}{\partial w} = \sum_{i=0}^{N-M}\left[\sum_{j=0}^{N-M}\left[\frac{\partial E}{\partial x_{ij}^{l}} \cdot \frac{\partial x_{ij}^{l}}{\partial w}\right]\right] = \sum_{i=0}^{N-M}\left[\sum_{j=0}^{N-M}\left[\frac{\partial E}{\partial y_{ij}^{l}} \cdot \sigma'(x_{ij}^{l}) \cdot y_{(i+a)(j+b)}^{l-1}\right]\right] \tag{14}$$

Lo cual puede ser escrito en forma de convolución de la siguiente manera:

$$\sum_{i=0}^{N-M}\left[\sum_{j=0}^{N-M}\left[\frac{\partial E}{\partial y_{ij}^{l}} \cdot \sigma'(x_{ij}^{l}) \cdot y_{(i+a)(j+b)}^{l-1}\right]\right] = \frac{\partial E}{\partial y} * \sigma'^{(x^{l})} * y_{(submatriz)}^{l-1} \tag{15}$$

Por lo cual la fórmula de ajuste de pesos se define como una triple convolución de la siguiente forma:

$$w_k \leftarrow w_k - \gamma \cdot \left[\frac{\partial E}{\partial y} * \sigma'^{(x^{l})} * y_{(submatriz)}^{l-1}\right] \tag{16}$$

Para explicar de forma más clara y esquematizada la fórmula de ajuste de pesos, supóngase que se cuenta con un mapa de características que posee A) una matriz de pesos sinápticos W de dimensiones 3 x 3. B) una matriz de entrada $y^{l-1} = x^{l}$ de dimensiones 6 X 6. C) una matriz de salida $y^{l} = \sigma(x^{l})$, D) una matriz $\sigma'(x^{l})$ de 4 x 4 y E) una matriz de errores retropropagados de la capa anterior $\frac{\partial E}{\partial y}$, de dimensiones 4 x 4. Con este esquema en mente, la ecuación 16 puede interpretarse de una forma esquemática como se muestra en los dos ejemplos de la figura 11, para el caso del ajuste del peso w_1 y w_8

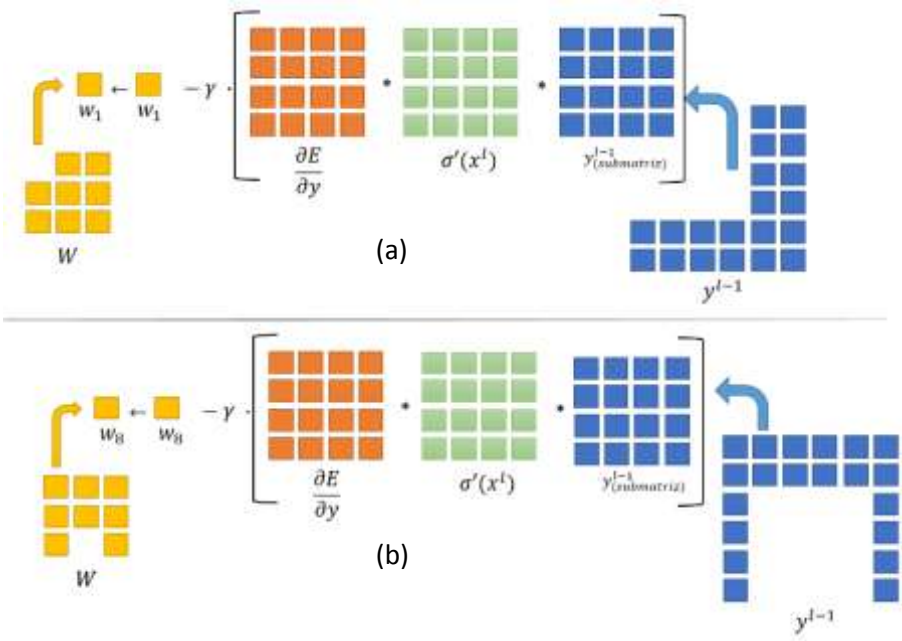

Fig. 11: Ajuste de pesos de un mapa de características. (a) Ejemplo para el peso w_1, (b) Ejemplo para el peso w_8.

Una vez ajustados los pesos sinápticos, es imperante el retropropagar el error, para ello se utiliza la siguiente fórmula que asigna el error a la capa anterior:

$$\frac{\partial E}{\partial y_{ij}^{l-1}} = \sum_{a=0}^{M-1}\left[\sum_{b=0}^{M-1}\left[\frac{\partial E}{\partial y_{(i-a)(j-b)}^{l}} \cdot \sigma'(x_{(i-a)(j-b)}^{l}) \cdot w_{ab}\right]\right] \tag{17}$$

En donde $\frac{\partial E}{\partial y_{ij}^{l-1}}$ representa el error en la capa previa. Existe un algoritmo que resume y simplifica la programación de la retropropagación del error a la capa previa. Consiste en realizar dos rotaciones de la matriz de pesos sinápticos w_{ab} y realizar una multiplicación punto a punto con la matriz de la derivada de la función sigmoidea $\sigma'(x^l)$ y a su vez con la matriz de errores $\frac{\partial E}{\partial y^l}$. Sin embargo, es posible que las dimensiones de la matriz de errores en la capa previa no correspondan con el resultado, por lo que se sugiere utilizarse bordes con ceros tanto en la matriz de errores de la capa actual, como en la matriz de derivada de la función sigmoidea. Esquemáticamente, puede observarse este algoritmo en la figura 12 y con ello, retropropagar el error por todas las capas y ajustar los pesos sinápticos hasta alcanzar el comportamiento deseado.

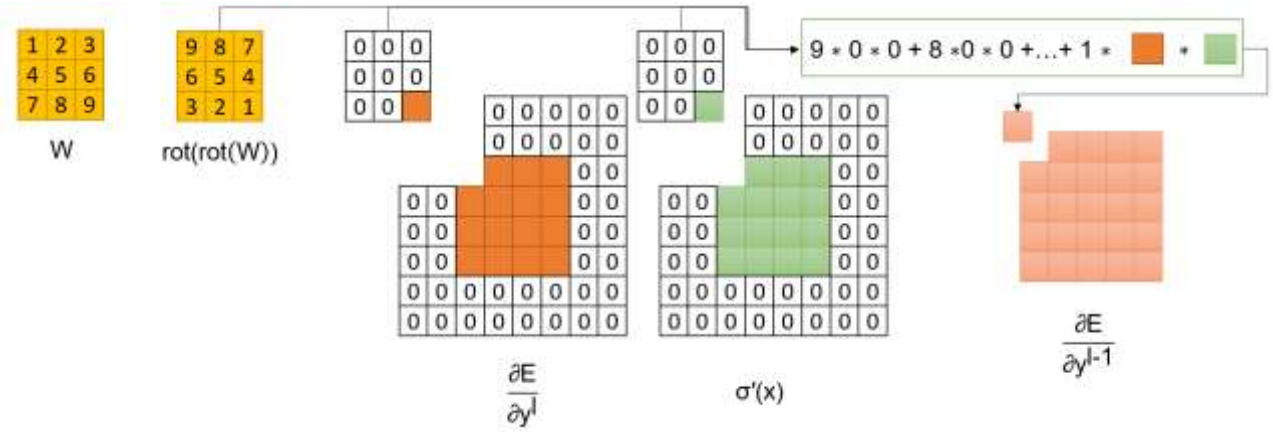

Fig. 12: Ejemplo de la retropropagación del error a la capa anterior.

RESULTADOS EXPERIMENTALES

En esta sección se realiza una comparativa entre los algoritmos descriptivos sometidos a una máquina de soporte vectorial y el algoritmo de aprendizaje profundo. Para el caso de la máquina de soporte vectorial, todas las imágenes de la base de datos fueron sometidas al algoritmo de Matriz de Coocurrencia de Motif Direccional, Patrón Local Binario Uniforme Invariante a Rotación, y los Filtros de Gabor. Los tres algoritmos mencionados generan un vector de valores que describen la imagen, mismos que son sometidos a la máquina de soporte vectorial con dos modos de discriminación: algoritmo de clasificación de uno versus todos –en el cual una clase trata de distinguirse de todas las demás- y uno versus uno –en el cual cada clase es discriminada contra otra, de forma separada-. Para brindar mayor precisión a los experimentos, se utilizó como núcleo la función de base radial (RBF por sus siglas en inglés) (Buhmann, 2003).

Para el caso del algoritmo de aprendizaje profundo, las imágenes de entrada fueron proveídas de forma directa, sin ninguna clase de filtro previo, de tal manera que el aprendizaje profundo infiera los filtros requeridos por medio del algoritmo de retropropagación para la tarea de reconocimiento y clasificación. En ambos casos se utilizó el 75% de las imágenes para entrenamiento y 25% para prueba, ambos conjuntos elegidos aleatoriamente de la base de datos para el proceso. Para medir la eficacia de cada experimento, se presenta la matriz de confusión y las métricas de exactitud, precisión y sensibilidad (Metz, 1978), las cuales son calculadas por medio de las siguientes fórmulas:

$$\text{Exactitud} = \frac{\sum_{i=1}^{M} TP_i}{\text{Total}}$$

$$\text{Precisión} = \frac{\left(\sum_{i=1}^{M} \frac{TP_i}{TP_i + FP_i} \right)}{M} \tag{18}$$

$$\text{Sensibilidad} = \frac{\left(\sum_{i=1}^{M} \frac{TP_i}{TP_i + FN_i} \right)}{M}$$

Donde M representa el número de clases, Total representa el número de muestras presentadas a clasificar, TP representan a los valores adecuadamente marcados como positivos, FP los valores erróneamente

marcados como positivos, y FN los valores marcados erróneamente como negativos. Resultados idóneos de una clasificación perfecta son Exactitud = 1, Precisión = 1, Sensibilidad = 1.

Matriz de Coocurrencia de Motif Direccional (MCMD) y clasificador SVM

Para el caso del descriptor MCMD con el clasificador SVM, los resultados pueden observarse en la Tabla 1 y 2. El cálculo de la exactitud, precisión y sensibilidad al emplearse la función de base radial generó valores de exactitud = 0.77, precisión = 0.78, y sensibilidad = 0.77 para el algoritmo de clasificación de uno versus todos. Para el caso del algoritmo de uno versus uno los valores son de exactitud = 0.77, precisión = 0.77, y sensibilidad = 0.77.

Tabla 1: Matrices de confusión para descriptor MCMD + Clasificador SVM con RBF. Clasificación del SVM de uno versus todos.

		Decisiones obtenidas por SVM + RBF: Clasificación Uno-versus-todos					
Etiquetas		Avenida	Edificios	Industria	Naturaleza	Residencial	Agua
	Avenida	35	7	3	1	4	0
	Edificios	2	41	3	0	3	1
	Industria	5	13	29	1	1	1
	Naturaleza	0	0	0	46	1	3
	Residencial	1	2	3	2	41	1
	Agua	0	0	2	6	1	41

Tabla 2: Matrices de confusión para descriptor MCMD + Clasificador SVM con RBF. Clasificación del SVM de uno versus uno.

		Decisiones obtenidas por SVM + RBF: Clasificación Uno-versus-uno					
Etiquetas		Avenida	Edificios	Industria	Naturaleza	Residencial	Agua
	Avenida	43	2	3	0	2	0
	Edificios	5	32	6	0	5	2
	Industria	3	7	33	1	4	2
	Naturaleza	0	0	0	46	1	3
	Residencial	2	1	8	0	38	1
	Agua	0	2	4	2	2	40

Patrón local binario Uniforme Invariante a Rotación (LBP-UIR) y clasificador SVM

Para el caso del descriptor LBP-UIR se utilizó un descriptor de radio R=1. Para el caso del SVM con función de base radial, los valores de exactitud, precisión y sensibilidad fueron de 0.756, 0.762 y 0.756 respectivamente, lo cual conlleva un error de 24.4% de todas las muestras presentadas para el algoritmo de uno versus todos. El algoritmo de uno versus uno mejoró el reconocimiento al obtener valores de 0.77, 0.78 y 0.77 en exactitud, precisión y sensibilidad. Los valores obtenidos pueden observarse en la matriz de confusión de las tablas 3 y 4.

Tabla 3: Matrices de confusión para descriptor LBP-UIR + Clasificador SVM con RBF, Algoritmo uno versus todos.

		Decisiones obtenidas por SVM + RBF: clasificación uno versus todos					
Etiquetas		Avenida	Edificios	Industria	Naturaleza	Residencial	Agua
	Avenida	35	8	4	0	3	0
	Edificios	9	31	6	0	3	1
	Industria	1	4	33	3	8	1
	Naturaleza	0	0	4	44	1	1
	Residencial	1	3	0	1	45	0
	Agua	0	3	2	3	3	39

Tabla 4: Matrices de confusión para descriptor LBP-UIR + Clasificador SVM con RBF, Algoritmo uno versus uno.

		Decisiones obtenidas por SVM + RBF: clasificación uno versus uno					
Etiquetas		Avenida	Edificios	Industria	Naturaleza	Residencial	Agua
	Avenida	42	2	1	0	5	0
	Edificios	4	36	7	0	2	1
	Industria	2	8	33	1	5	1
	Naturaleza	1	0	1	44	3	1
	Residencial	5	3	1	0	39	2
	Agua	1	0	0	7	3	39

Filtros de Gabor y clasificador SVM

Los vectores obtenidos por los filtros de Gabor fueron sometidos al clasificador SVM, los cuales presentaron métricas de desempeño superiores a los descriptores MCMD y LBP-UIR. Para el caso de la función de base radial en SVM, las métricas de desempeño fueron de exactitud = 0.786, precisión = 0.781, y sensibilidad = 0.786, haciéndolo una aceptable combinación entre un descriptor robusto y un clasificador adecuado con el algoritmo de no contra todos. Para el caso de uno versus uno, las métricas fueron exactitud = 0.783, precisión = 0.797, y sensibilidad = 0.783. Las matrices de confusión para ambos experimentos pueden observarse en la tabla 5 y 6.

Tabla 5: Matrices de confusión para descriptor por filtros de Gabor + Clasificador SVM con RBF. Algoritmo uno versus todos.

		Decisiones obtenidas por SVM + RBF: clasificación uno versus todos					
Etiquetas		Avenida	Edificios	Industria	Naturaleza	Residencial	Agua
	Avenida	37	7	4	0	0	2
	Edificios	8	29	6	0	5	2
	Industria	4	7	30	0	3	6
	Naturaleza	0	0	0	49	1	0
	Residencial	0	2	3	0	45	0
	Agua	0	1	2	0	1	46

Tabla 6: Matrices de confusión para descriptor por filtros de Gabor + Clasificador SVM con RBF. Algoritmo uno versus todos.

		Decisiones obtenidas por SVM + RBF: clasificación uno versus uno					
Etiquetas		Avenida	Edificios	Industria	Naturaleza	Residencial	Agua
	Avenida	33	14	1	0	0	2
	Edificios	3	39	4	0	3	1
	Industria	4	10	31	0	3	2
	Naturaleza	0	0	1	45	2	2
	Residencial	1	4	1	0	43	1
	Agua	0	0	3	0	3	44

Aprendizaje profundo

A diferencia de los tres experimentos anteriores, en el caso de los experimentos de aprendizaje profundo se sometieron las imágenes directamente a la red convolucional, sin pasar por procesos descriptivos con la finalidad de que las redes convolucionales infirieran los filtros y descriptores adecuados para la tarea.

Para el caso del aprendizaje profundo se utilizó la conocida estructura de AlexNet (Krizhevsky et al. 2012), con una tasa de aprendizaje de $\gamma = 0.01$ el cual decaía cada tercio del experimento en valores de $\gamma = 0.001$ y de $\gamma = 0.0001$ de tal manera que la convergencia del experimento se alcanzara con facilidad, por ende, fueron requeridas 30 épocas de entrenamiento. El 75% de las imágenes fue utilizado para entrenamiento. Los resultados fueron obtenidos del 25% restante, el cual se destinó para probar la eficacia del clasificador. Las métricas obtenidas para este clasificador superan incluso a las de la aplicación del filtro de Gabor, con una exactitud de 0.87, una precisión de 0.873 y una especificidad de 0.87. Los resultados del clasificador pueden observarse en la matriz de confusión de la tabla 7.

Tabla 7: Matrices de confusión para el algoritmo de Aprendizaje Profundo.

		Aprendizaje profundo. 30 épocas, estructura AlexNet					
Etiquetas		Avenida	Edificios	Industria	Naturaleza	Residencial	Agua
	Avenida	42	3	2	0	3	0
	Edificios	0	43	6	1	0	0
	Industria	0	10	38	0	2	0
	Naturaleza	0	0	2	47	0	1
	Residencial	2	0	1	0	46	1
	Agua	1	0	1	3	0	45

A manera de comparativa, los resultados de cada experimento se colocan juntos en la tabla 8, en donde el valor de una clasificación perfecta equivale a 1.0 en los tres rubros. Como puede observarse en dicha tabla, el algoritmo de aprendizaje profundo tuvo un mejor desempeño, sin necesidad de requerir algoritmos de descriptores.

Tabla 8: Tabla comparativa entre los diferentes algoritmos de descripción y clasificación.

	MCMD + SVM Uno vs todos	*MCMD + SVM Uno vs uno*	*LBP-UIR + SVM Uno vs todos*	*LBP-UIR + SVM Uno vs uno*	*Gabor + SVM Uno vs todos*	*Gabor + SVM Uno vs uno*	*Apr. Profundo*
Exactitud	0.7767	0.7733	0.7567	0.7767	0.7867	0.7833	0.8700
Precisión	0.7812	0.7754	0.7621	0.7804	0.7820	0.7976	0.8737
Especificidad	0.7767	0.7733	0.7567	0.7767	0.7867	0.7833	0.8700

Todos los experimentos pueden constatar una confusión relativamente elevada entre edificios e industrias, esto es debido a que muchas de las imágenes podrían resultar confusas inclusive para la tarea de clasificación para un ser humano, por lo que es una característica esperada en los clasificadores. Prueba de la complejidad de la tarea mencionada puede observarse en la figura 13.

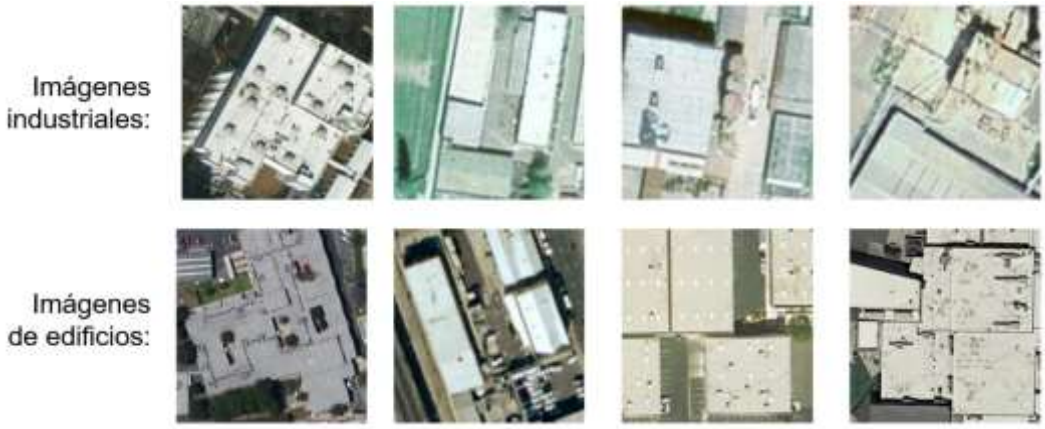

Fig. 13: Muestras muy similares entre imágenes industriales e imágenes de edificios. Nótese la complejidad de discernir entre las clases, aún para una tarea humana.

DISCUSIÓN FINAL

En este artículo se ha mostrado el desempeño de distintos algoritmos descriptivos y clasificadores sobre una base de datos propuesta, compuesta de imágenes aéreas, para definir qué tipo de algoritmo descriptivo y clasificador obtiene mejores resultados de clasificación. Debido a la tabla comparativa presentada en la sección de resultados, se puede concluir que el algoritmo de aprendizaje profundo presenta un adecuado desempeño para la tarea, debido a que su naturaleza intrínseca es generar filtros y descriptores semánticos de alto nivel, a comparación con los descriptores de textura de bajo nivel como son el MCMD, LBP-UIR y Filtros de Gabor. Sin embargo, a pesar de su elevado desempeño (87% de exactitud), aún los sistemas de aprendizaje profundo encuentran difícil discernir entre muestras de clases muy similares, tal y como un ser humano pudiese responder ante la tarea de clasificación.

Si bien es cierto que estos descriptores de textura han sido utilizados con antelación en diferentes trabajos de clasificación (Ahonen et al. 2006; Álvarez et al. 2010; Calderón et al. 2015; Yuan 2011), el algoritmo de aprendizaje profundo es capaz de tener un mejor desempeño, debido a que cada una de las capas se enfoca en encontrar características semánticas particulares al problema. Dichas capas van teniendo más relevancia semántica conforme avanzan en profundidad, hasta brindar características semánticas de alto nivel (Nielsen, 2015). La estructura de aprendizaje profundo de AlexNet (Krizhevsky et al. 2012) es una buena combinación entre profundidad en redes convolucionales y costo computacional, por ende, fue empleada como estructura para los experimentos mencionados.

La base de datos de imágenes aéreas propuesta se creó uniendo las conocidas bases de datos de Merced, Banja Luka, Anotirana No Bazaa y algunas muestras obtenidas desde el sistema ArcGIS, para generar una base de datos de 1200 imágenes divididas en 6 categorías, las cuales pueden ser utilizadas para entrenar sistemas automáticos que reconozcan las categorías incluidas sin importar la zona geográfica donde se apliquen. Debido a esta característica, el sistema entrenado puede ser utilizado irrestrictamente en muchas zonas urbanas, sin importar la localidad en donde se aplique.

Sin embargo, los sistemas de aprendizaje profundo tienen mejor desempeño conforme más cantidad de datos se presenten, ya que aprenden e infieren mejores características conforme más muestras se introduzcan a las redes convolucionales. Por ende, son idóneos en aplicaciones con grandes cantidades de datos o un flujo constante de ellos. Si grandes cantidades de datos no se encuentran presentes -como es el caso de una base de imágenes de pocas muestras- es recomendable utilizar descriptores semánticos de bajo nivel junto con un clasificador, ya que las técnicas de aprendizaje profundo podrían sobre-entrenarse con pocas muestras y no reconocer adecuadamente información nueva. Experimentos de Aprendizaje Profundo en otras áreas han demostrado que una sintonización fina de los pesos sinápticos en las capas convolucionales puede aumentar la tasa de reconocimiento, por lo que un razonable camino para mejorar los resultados en trabajo futuro pueden conllevar el uso de la sintonización fina de un modelo pre-entrenado.

CONCLUSIONES

Con base en los resultados experimentales, las tablas que contienen la matriz de confusión y la discusión final mostrada en la sección anterior, es posible enlistar las siguientes conclusiones:

(i) El algoritmo de aprendizaje profundo presenta un desempeño superior para la tarea de reconocimiento de imágenes aéreas, sobrepasando a los algoritmos de bajo nivel semántico. Debido a su naturaleza, es posible generar filtros específicos que se adapten a la tarea encomendada, lo que mejora el desempeño en el reconocimiento; *(ii)* Debido a que la base de datos utilizada para entrenar emplea categorías comunes a cualquier zona geográfica, el algoritmo de aprendizaje profundo propuesto es capaz de emplearse en cualquier ambiente con la misma tasa de éxito; *(iii)* pesar del elevado desempeño de las técnicas de aprendizaje profundo (87% de reconocimiento), estos algoritmos presentan dificultad en distinguir entre las clases de edificios e industria, problemática que se presenta también en los filtros de bajo nivel. Por ende, la investigación a futuro podría mejorar el desempeño si se enfoca en técnicas de discernimiento entre estas dos clases.

REFERENCIAS

Ahonen, T., A. Hadid y M. Pietikäinen. Face Description with Local Binary Pattern. Application to face recognition, IEEE Trans. on Pattern Analysis and Machine Intelligence, 28 (12), 2037-2041 (2006)

Álvarez, M. J., E. González, F. Bianconi, J. Armesto y A. Fernández. Colour and texture Features for Image Retrieval in Granite induatry, Dyna: 77(161), 121-130 (2010)

Boureau, Y-Lan, Ponce J. y LeCun Y. A theoretical analysis of feature pooling in visual recognition. Proceedings of the 27th international conference on machine learning, ICML-10, (2010)

Buhmann, M. D. Radial basis functions: theory and implementations. Cambridge Monographs on Applied and Computational Mathematics, 12, 147-165 (2003)

Calderón, G., A. Fierro, K. Toscano, M. Nakano y H. Pérez. Extracción de Imágenes de Textura usando MCMs Direccionales, Simposio Iberoamericano Multidisciplinario de Ciencias e Ingenierías, Hidalgo, México, 21-23 Septiembre (2015)

Calderón G., A. Fierro, M. Nakano, H. Pérez. Efecto de la Transformada Motif en Desarrollo de Descriptores de Textura para la Extracción de Imágenes: Información Tecnológica, 27(3), 199-214 (2016)

Castelli, V., y Bergman, L. D. Image databases: search and retrieval of Digital Imagery: Jon Wiley & Sons (2002)

Ciresan, D. C., Meier, U., Gambardella, L. M., y Schmidhuber, J. Convolutional neural network committees for handwritten character classification. International Conference on Document Analysis and Recognition (pp. 1135-1139). IEEE. (2011)

Cortés, C., y Vapnik, V. Support-vector networks. Machine learning, 20(3), 273-297 (1995)

Datta, R., D. Joshi, J. Li y J.Z. Wang, Image Retrieval: Ideas, Influences, and Trends of a New Age, ACM Computing Survey: 40(2), 5:3-5:60 (2008)

Fierro, A.N., G. Calderón, M. Nakano y H. M. Pérez. Motif Correlogram for Texture Image Retrieval, Intelligent Software Methodologies, Tools and Techniques, 496-505, Naples, Italia, 15-17 September (2015)

Krizhevsky, A., Sutskever, I., Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks, Advances in Neural Information Processing Systems 25 NIPS (2012)

Manjunath, B. S. y W.Y. Ma. Texture Features for Browsing and Retrieval of Image Data, IEEE Trans. on Pattern Analysis and Machine Intelligence, 18(8), 837- 842 (1996)

Manjunath, B. S., J-R. Ohm, V. V. Vasudevan y A. Yamada, Color and Texture Descriptors, IEEE Trans. on Circuit and Systems for Video Technology, 11 (6), 703-715 (2001)

Manjunath, B. S., Salembier, P., and Sikora, T. Introduction to MPEG-7: multimedia content description interface (Vol. 1). John Wiley & Sons (2002)

Maza, I., Caballero, F., Capitán, J., Martínez-de-Dios, J. R., y Ollero, A. Experimental results in multi-UAV coordination for disaster management and civil security applications. Journal of intelligent & robotic systems, 61(1-4), 563-585 (2011)

Metz, C. E. Basic principles of ROC analysis. In Seminars in nuclear medicine. WB Saunders, 8(4), 283-298 (1978)

Nielsen, Michael A. Neural Networks and Deep Learning. Determination Press (2015)

Ojala, T., Pietikainen, M. y Maenpaa, T., Multiresolution Gray-scale and Rotation Invariant Texture Classification with Local Binary Patterns, IEEE Trans. on Pattern Analysis and Machine Intelligence, 24(7), 971–987 (2002)

Rui, Y., T.S. Huang y S. Chang. Image Retrieval: Current Techniques, Promising Directions and Open Issues, Journal of Visual Communication and Image Representation, 10 (1), 39-62 (1999)

Risojević V., Momić S, y Babić. Z. Gabor Descriptors for Aerial Image Classification, In A. Dobnikar, U. Lotrič, and B. Šter, editors, Proceedings of 10th International Conference on Adaptive and Natural Computing Algorithms, ICANNGA 2011, Part II, vol. 6594 of Lecture Notes in Computer Science, pp. 51-60 (2011)

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

Waharte, S., Trigoni, N. Supporting search and rescue operations with UAVs, 2010 International Conference in Emerging Security Technologies (pp. 142-147). IEEE. (2010)

Wan, J., Wang, D., Hoi, S. C. H., Wu, P., Zhu, J., Zhang, Y., Li, J. Deep learning for content-based image retrieval: A comprehensive study. In Proceedings of the 22nd ACM, Int. Conference on Multimedia (pp. 157-166). ACM (2014)

Werbos, Paul. Beyond regression: New tools for prediction and analysis in the behavioral sciences (1974)

Werbos, Paul. Applications of advances in nonlinear sensitivity analysis. In System modeling and optimization pp. 762-770, Springer Berlin, Heidelberg (1982)

Yi, Y., Newsam, S. Bag-Of-Visual-Words and Spatial Extensions for Land-Use Classification, ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM GIS, (2010)

Yuan, F., Video-based smoke detection with histogram sequence of LBP and LPBV pyramids, Fire Safety Journal, 46 (3) 132-139 (2011)