PHOW Classification
Laboratorio de Clasificación

Juliana Sánchez Posada
Universidad de los Andes
Visión Artificial
j.sanchez679@uniandes.edu.co

Abstract

En este artículo se utilizó una de las aplicaciones de 'VLFeat' que construye por defecto un clasificador de imágenes por reconocimiento de escenas de la base de datos 'Caltech-101', y se modificó el código de tal forma que con éste se pudiera crear un clasificador de otra base de datos llamada 'Tiny ImageNet'. En el artículo se describe la forma como se construye el clasificador, se describen las características de las dos bases de datos y se presentan los resultados de clasificación al variar, para las dos bases de datos, diferentes parámetros del algoritmo de clasificación. Finalmente se comparan y discuten los resultados.

1. Introducción

La visión artificial, un subcampo de la inteligencia artificial, tiene como objetivo principal programar los computadores de tal forma que éstos logren entender, interpretar y clasificar las imágenes tal y como un humano es capaz de hacerlo [1]. Uno de los objetivos secundarios que hace parte de este objetivo principal corresponde al proceso de clasificación, cuyo fin último es lograr clasificar una imagen dentro de un grupo de categorías dadas.

1.1. Clasificación

El problema de clasificación de imágenes, corresponde a un problema de aprendizaje automático supervisado y discreto. En él, se entrena un clasificador de imágenes con un grupo de imágenes de entrenamiento que se encuentran clasificadas en diferentes categorías. Posteriormente el clasificador entrenado se pone a prueba con imágenes nuevas que pertenecen a alguna de las categorías, pero que no hacían parte de las imágenes de entrenamiento, y se verifica la categoría en la cual el clasificador la clasifica. El proceso de clasificación y prueba de un clasificador se puede ver en las Figuras 1 y 2.

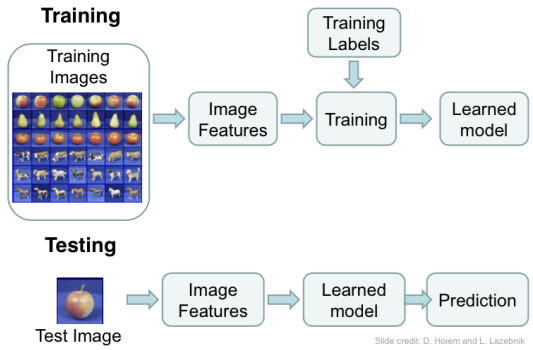


Figure 1: Proceso de entrenamiento y prueba de un clasificador [3]

.

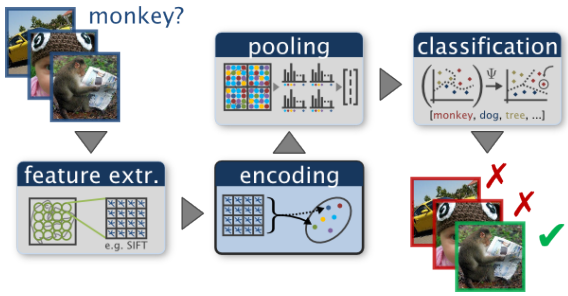


Figure 2: Proceso de clasificación de una imagen. Tomado de: `http://www.robots.ox.ac.uk/~vgg/research/encoding_eval/`

1.1.1 VLFeat

VLFeat es una librería abierta, compuesta por diferentes algoritmos especializados en el entendimiento de imágenes y de sus características locales, además de procesos de extracción y emparejamiento. Algunos de los algoritmos con los que cuenta esta librería son: Vector de Fisher, SIFT, k-

means, k-means jerárquico, entrenamiento de SVM ('support vector machines') de larga escala, entre otros.

El algoritmo de reconocimiento básico (phow_caltech101.m) que hace parte de la librería VLFeat, fue el algoritmo que se utilizó en el desarrollo del presente laboratorio. Éste, utiliza VLFeat para entrenar y probar un clasificador de imágenes en la base de datos Caltech-101. Con las parámetros que se encuentran establecidos por defecto (una única característica por imagen y 15 imágenes de entrenamiento por categoría) el clasificador alcanza una exactitud del 65%. El algoritmo utiliza:

- Características PHOW (descriptores densos multi-escala SIFT)
- k-means para la construcción de un diccionario de palabras visuales
- Un kernel que logra transformar una máquina de vectores de soporte (SVM) Chi-cuadrado en una lineal.

Para mayor información acerca de VLFeat o del algoritmo de reconocimiento básico dirigirse a la siguiente página web: `http://www.vlfeat.org/applications/apps.html`. A continuación se mencionan algunas de los algoritmos internos utilizados por la libraría, en especial por el código de MATLAB phow_caltech101.m. (Para descargar el código de MATLAB, ingresar al siguiente link [5]).

1.1.2 SIFT

SIFT es un algoritmo usado en visión artificial para lograr extraer características de la imagen, también llamadas descriptores relevantes de imágenes que después se pueden utilizar en el reconocimiento de objetos, clasificación de imágenes entre otros. Las características en particular que logra extraer éste algoritmo son características invariantes frente al cambio de escala y frente a las rotaciones de la imagen. Una vez se han identificado las características de las imágenes, estas se clusterizan para obtener un vocabulario visual que en el caso de $bagofwords$ genera un diccionario de palabras [3].

1.1.3 Bag of Words

Los modelos de bolsa de palabras ($bagofwords$) es una técnica popular para la clasificación de imágenes, en la que se ignora información espacial de la imagen y clasifica las imágenes basadas en un histograma de frecuencias de palabras visuales [3].

1.1.4 Creación del clasificador

Tal y como se mencionó anteriormente el método de reconocimiento que se utilizó fueron los descriptores PHOW, los cuales corresponden a una variante de los descriptos densos SIFT. El proceso que se sigue para entrenar el clasificador es el que sigue:

1. Semejante a lo que se realizó en el laboratorio de textones, se deben obtener algunos descriptores PHOW de las imágenes de entrenamiento de las dos bases de datos.
2. El proceso de clusterización (con k-means) de estos descriptores para todas las imágenes de entrenamiento, obtiene como resultado un diccionario de palabras o en inglés $bagofwords$. El código por defecto tenía definido el número de palabras para el diccionario, 600.
3. Posterior a este proceso se genera la representación de las imágenes de entrenamiento en los descriptores PHOW y se identifica a cuál de las 600 palabras del diccionario ya creado, se encuentra más cercana cada imagen. De esta manera a cada una de las imágenes de entrenamiento se le van a asociar determinada cantidad de palabras del diccionario de palabras que la describen de mejor forma.
4. Ya teniendo esta información se entrena una máquina de vectores de soporte Chi-cuadrado (SVM), lo que resulta en la generación del clasificador final, con el número de clases escogidas por el usuario.

El proceso de prueba del clasificador consiste simplemente en encontrar los descriptos PHOW de las imágenes de prueba, asociar los resultados a las palabras del diccionario de palabras y finalmente seleccionar la categoría a la que pertenecen las imágenes de acuerdo a la categoría que haya obtenido palabras del diccionario semejantes. Los dos resultados que arroja el algoritmo son: la matriz de confusiones y una matriz de puntaje.

- La primera es el resultado que se presenta al haber clasificado un número de imágenes de test (definidas por el usuario) con el clasificador y haber comparado éstos resultados con la clasificación de la verdad terreno para esas mismas imágenes. Una clasificación correcta de todas las imágenes de test de una categoría en esa categoría arrojaría un punto rojo fuerte en el punto donde la categoría resultante se cruza con la categoría correspondiente en la verdad terreno. Este punto debe ubicarse en la diagonal de la matriz de confusiones.
- La segunda corresponde a una matriz donde se puede ver el puntaje que le asigna el clasificador a la clasificación recibida tanto por las imágenes de entrenamiento como por las de prueba.

1.2. Bases de Datos

Diferentes bases de datos se han creado a lo largo de los últimos años para poder poner a prueba diferentes algoritmos de clasificación y poder comparar los resultados que de ellos se obtengan. Dos de estas bases de datos de utilizaron para el presente laboratorio, las cuales se describen a continuación.

1.2.1 Caltech-101 [2]

Caltech-101 se encuentra compuesta por imágenes de objetos que pertenecen a 101 categorías diferentes, en las cuales una imagen no puede pertenecer a dos o más categorías. Cada categoría tiene entre 40-800 imágenes. Todas las imágenes cuentan con dimensiones variadas de aproximadamente 300x200 píxeles. Un ejemplo de algunas de las imágenes de 5 de las categorías de esta base de datos se presenta en la Figura 3. Las imágenes de esta base de datos cuentan siempre con los objetos (categorías) en una posición estereo-típica, centrada en la imagen, con más o menos la misma escala y apariencia. En ningún caso los objetos se encuentran ocluidos.

Nota: Para conocer un poco más acerca de esta base de datos, ingresar al siguiente link: `http://www.vision.caltech.edu/Image_Datasets/Caltech101/`

1.2.2 Tiny Imagenet [4]

'ImageNet Tiny' es una base de datos extraída de una base de datos mayor conocida como 'ImageNet'. 'ImageNet Tiny' se encuentra compuesta por imágenes de objetos que pertenecen a 200 categorías diferentes. Nuevamente, al igual que Caltech-101, las imágenes solo pueden pertenecer a una categoría. Cada categoría cuenta con 100 imágenes. Todas las imágenes cuentan con dimensiones de 256x256 píxeles. Un ejemplo de algunas de las imágenes de 6 de las categorías de esta base de datos se presenta en la Figura 4. Las imágenes de esta base de datos, contrario a Caltech-101, cuentan con los objetos (categorías) en posiciones, escalas, puntos de vista e iluminaciones diferentes. El objeto al que corresponde la categoría puede no estar centrado en la imagen y puede no ser el protagonista de la misma. En muchos casos los objetos se encuentran además ocluídos, deformados y con apariencias diferentes. Algunas de estas características en las imágenes se pueden ver en la Figura 4, donde se muestran algunos ejemplos de imágenes de esta base de datos.

Estas características de esta base de datos hace que el problema de clasificación para estas imágenes sea mucho más complejo que para la base de datos de Caltech-101.

Nota: Para conocer un poco más acerca de esta base de datos, ingresar al siguiente link: `http://image-net.org/about-overview`

2. Resultados y discusión de resultados

Inicialmente se entrenó el clasificador para la base de datos de Caltech-101 y posteriormente para Tiny ImageNet.

Para que el código que se descarga con la base de datos de Caltech-101 por defecto (phow_caltech101.m.), pudiera correr en la base de datos Tiny ImageNet fue necesario realizarle algunas modificaciones, las cuales se mencionan a continuación:

- Cambiar el formato de la imagen de '*.jpg' a '*.JPEG'.
- Modificar el directorio ($'conf.calDir'$) en el cual se encuentran las diferentes categorías separadas en carpetas con imágenes.
- Comentar la sección en la que se descargan las imágenes de Caltech-101 ('Download Caltech-101 data')

La Figura 5 muestra uno de los resultados que se obtuvo para la matriz de confusiones al entrenar el clasificador con 15 imágenes de entrenamiento, y probarlo con 15 imágenes de prueba, manteniendo los demás parámetros por defecto constantes. Esto, para las dos bases de datos.

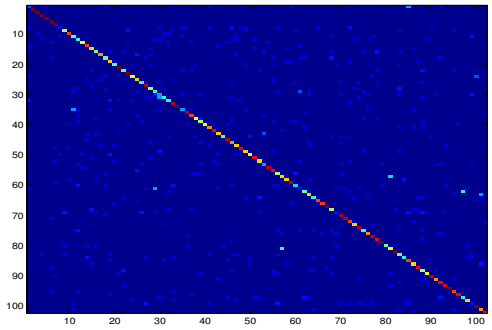
Uno de los valores que permite medir el desempeño de un clasificador al clasificar imágenes de prueba se conoce como la exactitud ($accuracy$). La forma como se obtiene éste valor para cada uno de los clasificadores entrenados y probados en cada una de las bases de datos es realizando el siguiente cálculo con la información obtenida de la matriz de confusión:

$$Exactitud = 100 \times mean\left(\frac{diag(MatrizConfusion)}{No.ImagenesTest}\right) \quad (1)$$

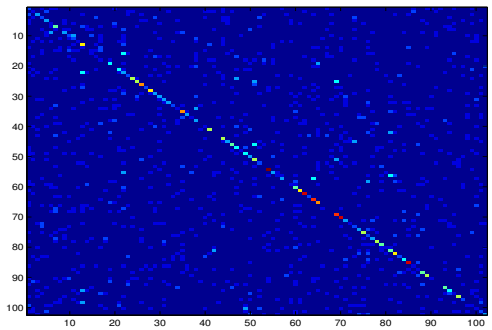
dónde $diag(MatrizConfusion)$ corresponde a obtener los valores asociados a la diagonal de la matriz de confusión y $mean$ a sacar el promedio de ellos.

Ya que uno de los objetivos principales de este laboratorio corresponde a ver el efecto que tienen ciertos parámetros importantes del algoritmo sobre el clasificador resultante, se variaron los parámetros que se presentan en Tabla 1 en combinaciones de los valores ahí reportados, generando todas los posibles clasificadores. Posteriormente se calcularon la exactitud y el tiempo que tomó cada clasificador en ser entrenado y en clasificar 15 imágenes de prueba, y se compararon. El tiempo que le toma al clasificador en entrenarse y en clasificar un grupo de imágenes de prueba da una idea acerca del costo computacional y de los recursos computacionales que se están invirtiendo en generar el clasificador. Por ésto, un mayor tiempo de entrenamiento, implica un mayor costo computacional.

De acuerdo a lo anterior los resultados que se obtienen de exactitud al variar el número de imágenes de entrenamiento,



(a) Caltech-101



(b) ImageNet

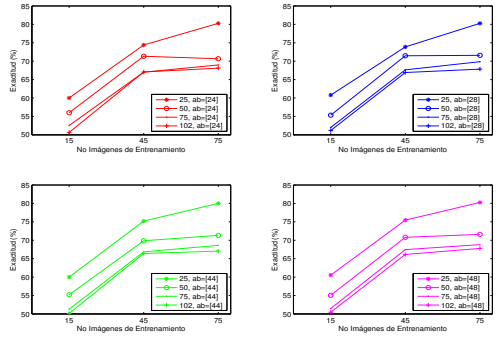
Figure 5: Matriz de confusión para clasificadores con 15 imágenes de entrenamiento, 15 de test, y 102 categorías

número de categorías y número de particiones espaciales se presentan en las figuras 6, 8, 11 y 12 respectivamente.

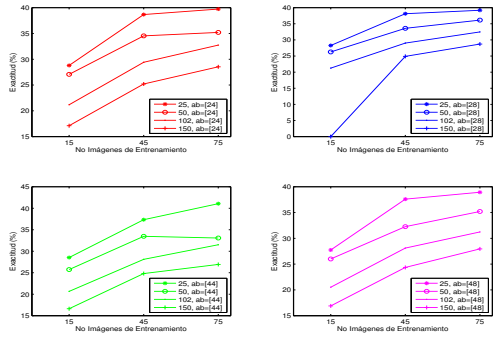
Table 1: Parámetros modificados en el clasificador.

Parámetro	Valores
No.Imágenes de Train	15, 45, 47
No.Categorías - Caltech-101	25, 50, 75, 102
No.Categorías - ImageNet	25, 50, 102, 150
No.Partición espacial [a,b]	[2,4], [2,8], [4,4], [4,8]

Inicialmente al observar cómo afecta la cantidad de imágenes de entrenamiento con la que se entrena el clasificador en la exactitud con la que clasifica las imágenes de prueba (ver Figura 6), mientras que se mantiene el número de categorías constante, es evidente que en todos los casos la exactitud aumenta a medida que el número de imágenes de entrenamiento aumenta. Por ejemplo, el sólo hecho de aumentar el número de imágenes de entrenamiento de 15 a 75 genera un aumento en la exactitud de la base de datos de Caltech-101 entre el 20 y 30%, y para la base de datos de Tiny ImageNet entre el 10-15%. Lo anterior ya que en la construcción del clasificador se cuenta con una mayor cantidad de información de descriptores PHOW con los cuales se entrena el SVM, arrojando un clasificador más robusto. Sin embargo, aumentar el número de imágenes de entrenamiento aumenta el tiempo que le toma al clasificador en entrenarse, pues son más imágenes las que se deben procesar y analizar para construir el diccionario y finalmente entrenar el clasificador. Lo anterior se puede ver en la Figura 7, lo que evidencia un mayor gasto computacional.



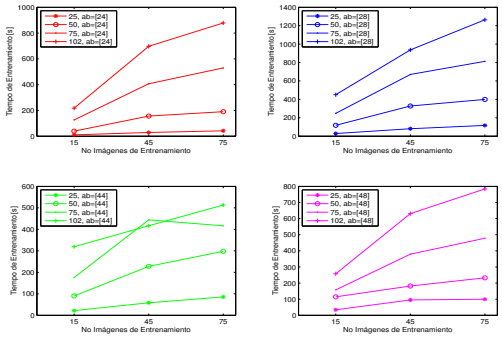
(a) Caltech-101



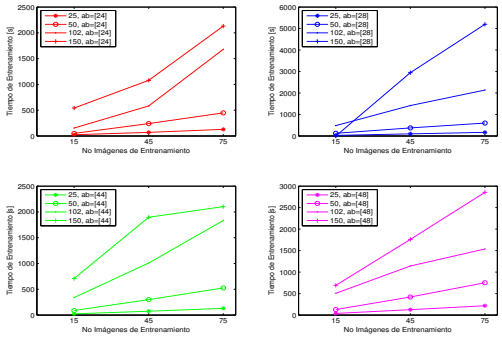
(b) ImageNet

Figure 6: Comparación de la exactitud de cada uno de los clasificadores al variar el número de imágenes de entrenamiento.

Ahora, al observar qué pasa con la exactitud al aumentar el número de categorías dentro de las cuales el clasificador puede entrar a clasificar una imagen de prueba, se puede ver que en todos los casos disminuye. Lo anterior, nuevamente tiene sentido pues si se mantiene el número de imágenes de entrenamiento constante, a medida que se aumenten las categorías, se debe construir entonces, un clasificador más complejo con la misma información que ya se tenía, por lo que es de esperarse que la exactitud con la cual el clasificador va a seleccionar la categoría de una imagen de prueba en particular debe ser menor. El comportamiento que se ob-

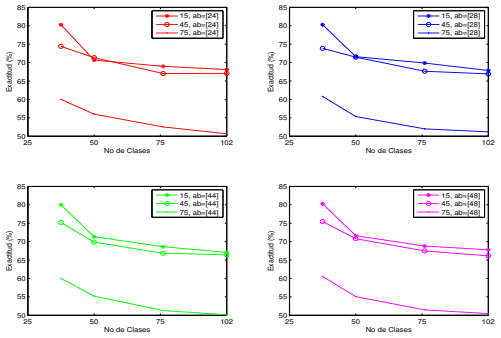


(a) Caltech-101

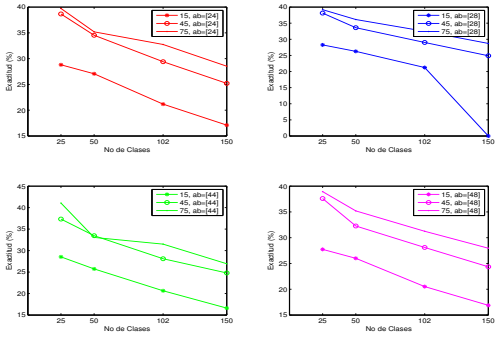


(b) ImageNet

Figure 7: Comparación del tiempo de entrenamiento de cada uno de los clasificadores al variar el número de imágenes de entrenamiento.



(a) Caltech-101



(b) ImageNet

Figure 8: Comparación de la exactitud de cada uno de los clasificadores al variar el número de categorías.

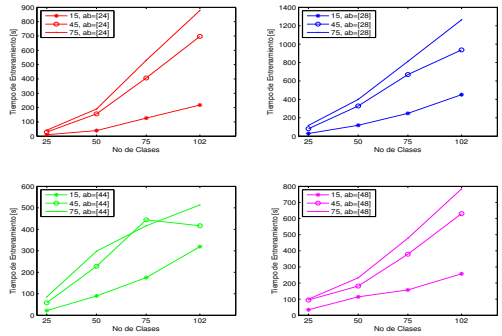
serva en la Figura 8 parece ser exponencial negativa para las base de datos de Caltech-101, y lineal decreciente para la base de datos Tiny ImageNet en el rango de categorías evaluadas para cada base de datos (ver Tabla 1). Por ejemplo, el sólo hecho de aumentar el número de categorías de 25 a 102 para Caltech-101 y de 25 a 150 para Tiny ImageNet genera una reducción en la exactitud de la base de datos entre el 10 y 15%.

Nuevamente, aumentar el número de categorías dentro de las cuales se quiere que el clasificador clasifique, aumenta el tiempo que le toma al clasificador en entrenarse, tal y como se puede ver en la Figura 9. La variación del tiempo de entrenamiento con respecto a las categorías presenta un comportamiento casi lineal en la mayoría de los casos. Sin embargo, el aumentar el número de categorías le es indiferente al tiempo que le toma al clasificador en clasificar un grupo de imágenes de prueba, pues como se puede ver en la Figura 10, este tiempo no presenta ninguna variación preferente en las gráficas.

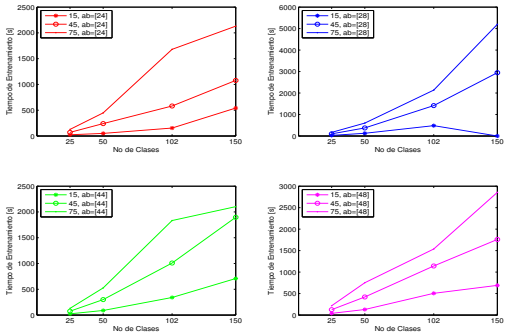
Analizando el efecto que tiene sobre la exactitud variar la partición espacial x $[a, b]$ es posible ver que manteniendo los otros dos parámetros constantes (número de imágenes de entrenamiento -Figura 11- y número de categorías 12), la exactitud no varía, si no que más bien permanece constante en todos los casos para las dos bases de datos.

Si se analiza el efecto que puede tener este parámetro sobre el tiempo que le toma al clasificador en ser entrenado (ver Figura 13) se observó que al mantener el valor a constante y aumentar el valor b incrementa el tiempo de entrenamiento del clasificador. Lo mismo ocurre cuando se mantiene constante b y se aumenta el valor de a. En cuanto al tiempo que le toma al clasificador clasificar imágenes de prueba no hay algún patrón que sigan todas las imágenes tal y como se puede ver Figura 14 pues en algunos casos parece que aumentar el valor de b manteniendo a constante genera un aumento en el tiempo de prueba, sin embargo en otros casos pasa lo contrario. Algo semejante pasa al variar a manteniendo b constante.

Ya habiendo visto todos los resultados para las dos bases de datos es posible realizar una comparación de las dos. Inicialmente resulta evidente que el clasificador entrenado para 'Caltech-101' resulta mucho más exacto que aquel entrenado para 'Tiny ImageNet'. Lo anterior teniendo en cuenta

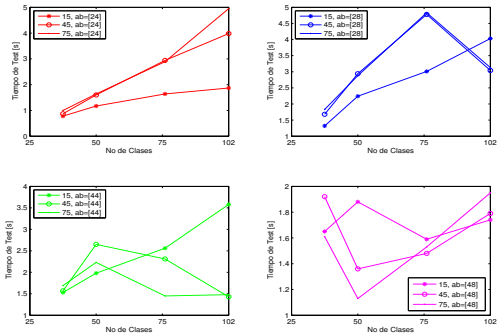


(a) Caltech-101

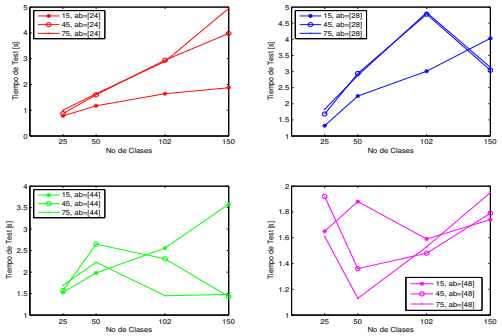


(b) ImageNet

Figure 9: Comparación del tiempo de entrenamiento de cada uno de los clasificadores al variar el número de categorías.



(a) Caltech-101



(b) ImageNet

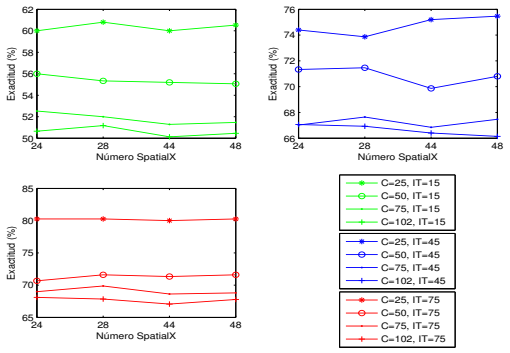
Figure 10: Comparación del tiempo de prueba de cada uno de los clasificadores al variar el número de categorías.

que esta última base de datos es mucho más compleja, pues los objetos de acuerdo a los cuales ésta se encuentra dividida en categorías, no son necesariamente los únicos que aparecen en la imagen o no son los protagonistas de la misma, lo que hace más compleja su clasificación. Algo diferente ocurre con las imágenes de la base de datos de 'Caltech-101' en las cuales los objetos se encuentran en el centro de la imagen, y con una orientación y escala en particular, lo que vuelve más sencillo el proceso. De ahí que en todos los casos, los resultados de exactitud sean siempre mayores para 'Caltech-101' (superiores siempre al 50%) que para 'ImageNet (inferiores siempre al 40%). La mejor exactitud se obtiene para la siguiente combinación de parámetros: 75 imágenes de entrenamiento y 25 categorías, sin importar la partición espacial x, ya que esta se mantiene constante al mantener los otros dos parámetros constantes.

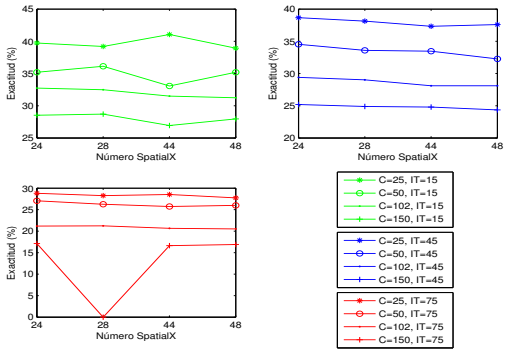
Se observó además que la modificación de ninguno de estos parámetros genera tendencias específicas o diferentes sobre el tiempo que le toma al clasificador clasificar un número fijo de imágenes de prueba. Se cree que este resultado tiene sentido pues depende del tipo de imagen de prueba que ingrese al clasificador y qué tan complicada y diferente sea de las imágenes con las que se entrenó el clasificador, el tiempo que este va a tomar en decidir a cuál categoría asignarlo. Evidentemente entre mayor número de categorías haya en el clasificador, más específica es cada categoría y la clasificación para una imagen de prueba diferente a las de clasificación más compleja sería.

3. Conclusiones

Fue posible entrenar un clasificador de imágenes con descriptores PHOW con dos bases de datos diferentes: Caltech-101 y Tiny ImageNet. Se modificaron varios parámetros dentro del clasificador y se observó como éstos afectaban la exactitud de los mismos. Se observó que aumentar el número de imágenes de entrenamiento por categoría con las que se entrena el clasificador aumenta la exactitud con la que este le asigna una categoría a una imagen de prueba que ingrese a ser clasificada, así como también aumenta el tiempo de entrenamiento y por ende el costo computacional. Aumentar el número de categorías manteniendo los demás parámetros constantes, reduce la exactitud del clasificador, pues se debe construir un clasificador más

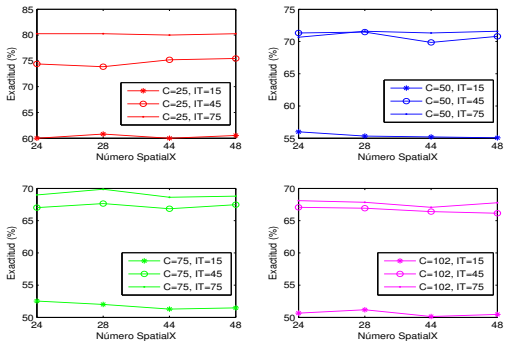


(a) Caltech-101

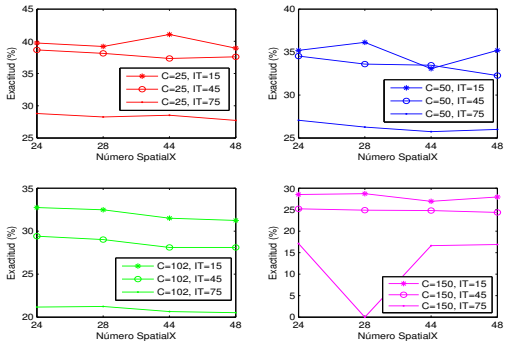


(b) ImageNet

Figure 11: Comparación de la exactitud de cada uno de los clasificadores al variar el número de partición espacial x.



(a) Caltech-101

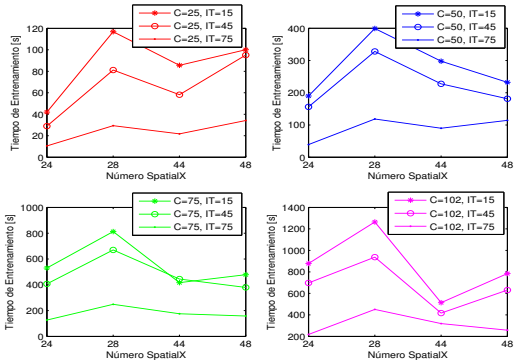


(b) ImageNet

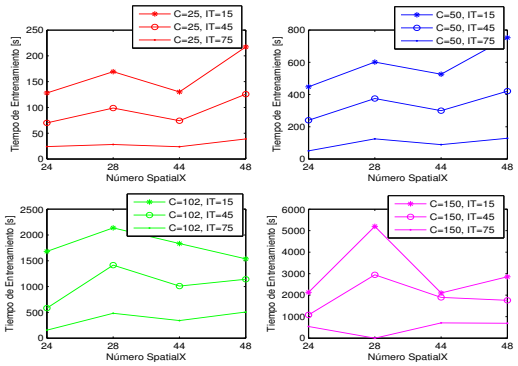
Figure 12: Comparación de la exactitud de cada uno de los clasificadores al variar el número de partición espacial x.

complejo con la misma información, lo que evidentemente reduce el la precisión del clasificador en el momento en que tenga que asignar una categoría a una nueva imagen. Aumentar este parámetro, aumenta el tiempo de entrenamiento del clasificador. Finalmente, al variar los parámetros de la partición espacial x, mientras que los otros dos parámetros se mantienen constantes, no modifica la exactitud del clasificador, sin embargo al aumentar alguno de sus dos valores, manteniendo el otro constante, genera un aumento en el tiempo de entrenamiento.

Teniendo en cuenta todos los resultados obtenidos, se podría decir que dependiendo del problema de clasificación al que uno se vaya a enfrentar, existirán ciertos valores para éstos parámetros ideales para obtener la mejor exactitud sin dejar a un lado la especificidad del clasificador. La selección del número de categorías debe estar basado principalmente en el problema a resolver, en la variedad de imágenes que se tengan disponibles para entrenar el clasificador y en qué tan diferentes pueden llegar a ser las imágenes de prueba con respecto a las de entrenamiento. A su vez se debe tener en cuenta así como en las dos bases de datos estudiadas, las características particulares de las imágenes a clasificar, su entorno, si se encuentran o no centradas y con una orientación y escala en particular o si por el contrario pueden presentar escalas diferentes y orientaciones variadas que requiere de un clasificador un poco más complejo y robusto para que se logre una buena clasificación. A partir de esto se sugiere que una vez estudiado el problema y las imágenes de entrenamiento, el usuario establezca un número de categorías mínimas en las cuales se espera poder clasificar las imágenes de entrenamiento, tal que se alcance el grado de especificidad deseado, siempre teniendo en cuenta que reducir el número de categorías para mejorar la exactitud del clasificador no siempre resulta siendo una buena solución al problema, pues reduce su especificidad. En cuanto a la selección del número de imágenes de entrenamiento, es importante mencionar que a medida que éste se aumenta, aumenta de manera lineal y significativa la exactitud del clasificador, por lo que a pesar de que este tome más tiempo en entrenarse debido al aumento en este parámetro, se podría decir que resulta bueno contar con un alto número de imágenes de entrenamiento por categoría que aseguren un buen desempeño y exactitud del clasificador, umbrales establecidos por el usuario y por



(a) Caltech-101



(b) ImageNet

Figure 13: Comparación del tiempo de entrenamiento de cada uno de los clasificadores al variar el número de partición espacial x.

el tipo de problema a resolver.

Finalmente se observó que todo aquello que genere un aumento en la exactitud, genera un aumento en el tiempo computacional y por ende un mayor gasto computacional.

References

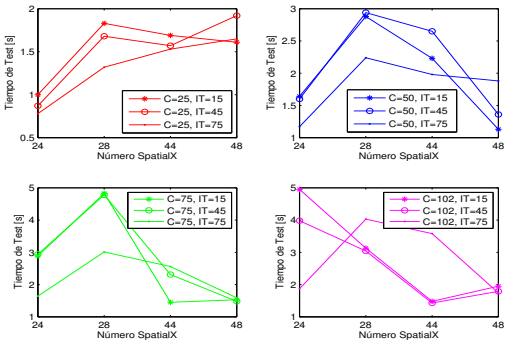
[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. 33(5):898–916.

[2] Caltech, 2006. Marzo 30, 2015.

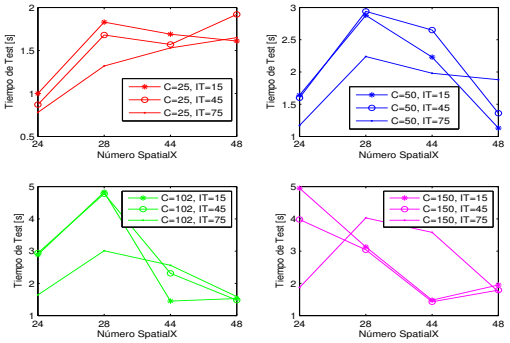
[3] Pablo Arbelaez. Presentaciones clase vision artificial.

[4] P. Stanford Vision Lab, Stanford University, 2014. Marzo 30, 2015.

[5] VLFeat.org, 2007. Marzo 30, 2015.



(a) Caltech-101



(b) ImageNet

Figure 14: Comparación del tiempo de prueba de cada uno de los clasificadores al variar el número de partición espacial x.

Figure 3: Ejemplos de imágenes de la base de datos Caltech-101.Se muestran 3 imágenes de 5 categorías diferentes: Hormiga, Mariposa, Helicóptero, Joshua Tree y Fresa.

Figure 4: Ejemplos de imágenes de la base de datos Tiny ImageNet. Se muestran 3 imágenes de 6 categorías diferentes: Banano, Bikini, Maleta, Camaleón africano, Container y Ganzo.