Clasificación con POW

Visión por Computador, Universidad de los Andes, Bogotá, Colombia

April 7, 2016

María Alejandra Bravo Sarmiento
ma.bravo641@uniandes.edu.co

Lina María Mejía López
lm.mejia11@uniandes.edu.co

Abstract

Para la clasificación de imágenes es necesario construir representaciones que nos permitan extraer las características más relevantes de las mismas. En este informe se presenta una aplicación del método de representación por medio de Scale-invariant feature transform (SIFT) tomando el gradiente en 8 direcciones en cada pixel. Se utiliza el clasificador de máquinas de soporte vectorial (SVM) utilizando aproximaciones del kernel chi-cuadrado con un gamma igual a 0.5. En este informe se presentan modificaciones en los parámetros de un código de clasificación utilizado en Caltech101 para que funcione en Imagenet. El mejor método obtuvo un resultado de ACA = 0.797 con 3 categorías en Imagenet comparado con ACA = 1 con 5 categorías obtenido en la clasificación de Caltech. Esto refleja una diferencia real entre las bases de datos.

1. Introducción

El problema de clasificación se viene estudiando desde los inicios de la visión computacional. A través de los años, han surgido diversos clasificadores, y la manera de estudiar el problema ha cambiado radicalmente. No obstante, el avance de esta área no se debe solo a los nuevos modelos de clasificación, sino también al surgimiento de nuevas y mejores maneras de representar una imagen (los vectores de descripción). A través de este laboratorio, nos fue posible familiarizarnos con una representación muy útil en clasificación: *Scale-invariant feature transform* (SIFT) publicada en 1999 por David Lowe. Utilizando el código propuesto ganador del challenge de Caltech101 (PHOW) es posible extrapolarlo a una base de datos mayor, Imagenet, considerando algunas modificaciones. Este codigo utiliza 'bag of words' de SIFT, para lo cual calcula un diccioario de 300 palabras, y el vector de reprecentacion de cada imagen es la concatenacion de hitogramas de SIFTs en cada panel. Se exploró como cambiaba la exactitud promedio del codigo al cambiar el numero de categorías, el numero de imagenes de entrenamiento, y las particiones espaciales (los paneles). Con esto fue posible entender como algunos parámetros pueden mejorar la discriminación del modelo.

2. Descripción de la base de datos

Se utilizaron dos bases de datos para este laboratorio. La primera Caltech101 [2] que consta de entre 40 a 800 imágenes por categoría, con un promedio de 50 imágenes por categoría, con 101 categorías. Fue recolectada en el 2003 por el grupo de visión artificial del California Institute of Technology. Las imágenes son de tamaño 300x200 pixeles en las que en su mayoría el objeto a clasificar se encuentra centrado en la imagen y no hay mucho ruido de fondo. Por otro lado la base de datos de Imagenet [1] es una base de datos organizada de acuerdo con la jerarquía de WordNet en la que un concepto puede ser descrito por múltiples palabras. Imagenet tiene 100000 diferentes categorías y en cada una tiene 1000 imágenes. Esta base de datos se hizo pública con un challenge en el que el objetivo era clasificar cada imagen en una de las categorías de las hojas del árbol de jerarquías. Sin embargo dado que al interior de las ramas del árbol de jerarquías no era tan fácil distinguir entre clases cercanas relajaron el método de evaluación dando 5 oportunidades de acierto. Esta base de datos es mucho más realista y complicada que Caltech101 por lo que requiere un mayor procesamiento y mejores clasificadores.

3. Descripción de los Métodos

Como se mencionó anteriormente, el algoritmo de PHOW utiliza 'bag of word'. A diferencia de la práctica pasada, las palabras no consisten vectores de 1x32 (respuesta a 32 filtros de cada pixel), sino vectores de 1x128 (SIFT de cada cuadro en la grilla). Dicho diccionario es calculado con k-means a partir de 10e4 vectores (sacados de 30 imágenes), y contiene 300 palabras.
Después de obtener el diccionario, se calcula el dense-SIFT de todas las imágenes. Finalmente se divide la imagen según la partición espacial especificada, y se calcula el histograma en cada panel, los cuales con concatenados. Después de tener los histogramas de todas las imágenes, los

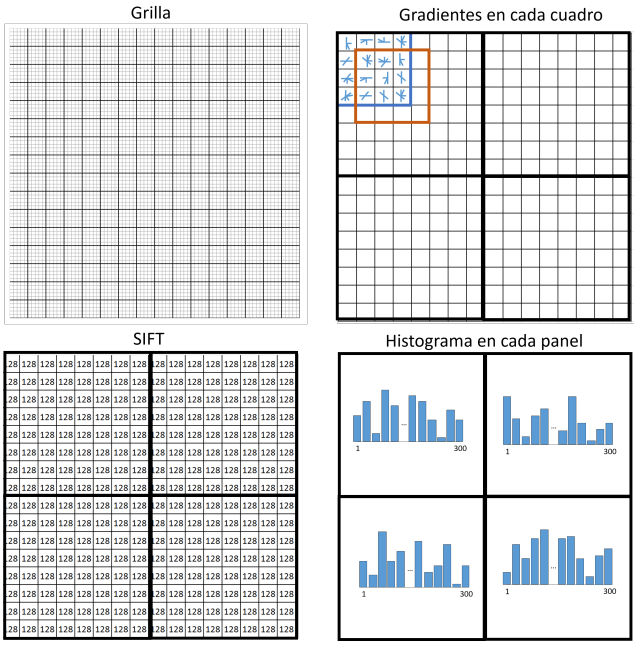


Figure 1. Pasos para calcular descriptor de la imagen

histogramas de las imágenes de entrenamiento se utilizan para entrenar los clasificadores (SVM), y los histogramas de las imágenes de evaluación son evaluados en el modelo.

3.1. Descriptor

Anteriormente se describió el método general, ahora se explicará en qué consisten los dense-SIFT y como se calculan los histogramas. Para empezar, el SIFT no se calcula en cada pixel de una imagen; en cambio, se define una grilla (en este caso, los cuadros de la grilla eran de 5x5 pixeles). Después, se calcula el gradiente de cada cuadro en 8 direcciones. Por cada ventana de 4x4 cuadros, se obtiene un vector de representación SIFT, el cual contiene 128 datos (8 direcciones x 16 cuadros).

Si se tiene un diccionario, a cada vector SIFT se le puede asignar su 'palabra' correspondiente, y posteriormente, se puede realizar un histograma. Como nuestro diccionario tiene 300 palabras, cada histograma es de 1x300. No obstante, el algoritmo no calcula un histograma por imagen, sino por panel. El tamaño y número de paneles se determina por la partición espacial.

3.2. Clasificador

El clasificador que utilizan en el código es una máquina de soporte vectorial (SVM) con la que se busca separar los datos de cada categoría por un hiperplano que separe los datos que pertenecen a esa categoría contra el resto, esto lo hace para todas las categorías. Esto se puede representar de manera matemática de la siguiente forma:

$$f(\mathbf{x}) =< \mathbf{w}, \mathbf{v} > +b = \sum_{i=1}^{d} x_i w_i + b. \tag{1}$$

donde $\mathbf{x}$ es el vector de representación, es decir el histograma, y $\mathbf{w}$ el vector de pesos que se aprende con los datos de entrenamiento. La función f es la funcion de clasificación. Se busca entonces resolver el problema convexo de optimizacion:

$$\begin{gathered} \min \frac{1}{2}||\mathbf{w}||^2 \\ \text{sujeto a } y_i(< \mathbf{x}_i, \mathbf{w} > + b) \geq 1, \\ \forall i = 1, ..., N. \end{gathered} \tag{2}$$

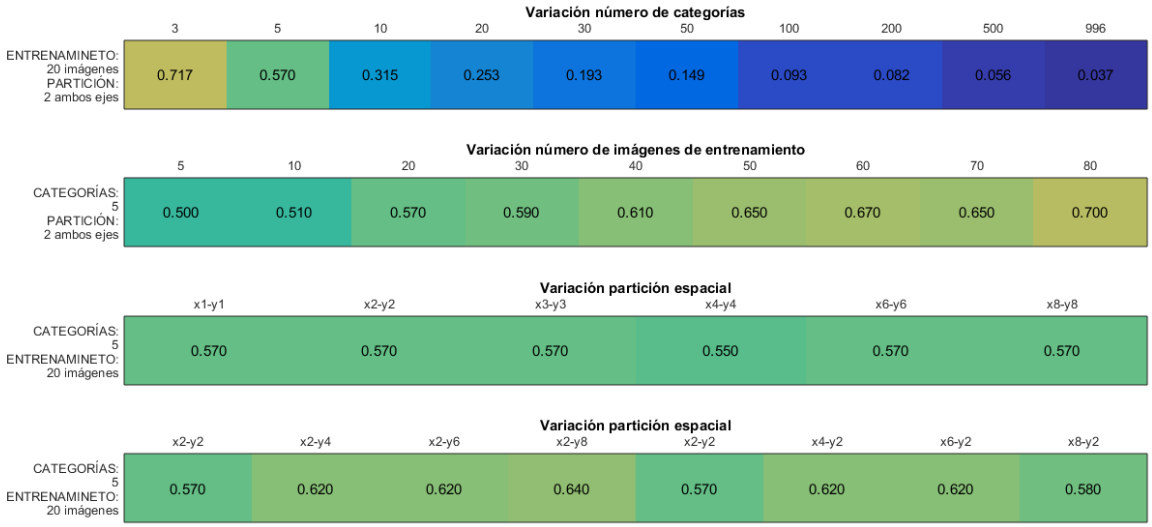
donde y_i son las clases, binarias, de las imagenes y b el sesgo. En el caso ideal, cuando los datos son linealmente separables, la ecuación 3 funciona, pero esto sucede muy pocas veces. Para solucionar esto se realizan entonces dos modificaciones, se agregan unas variables de holgura ξ_i y se realiza una trasformación h del espacio. El problema termina siendo equivalente a utilizar un kernel y agregar las nuevas variables:

$$\begin{aligned} f(\mathbf{x}) &= \sum_{i=1}^{d} \alpha_i y_i < h(\mathbf{x}), h(\mathbf{x}_i) > +\beta_0 \\ &= \sum_{i=1}^{d} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) > +\beta_0. \end{aligned} \tag{3}$$

$$\begin{gathered} \min \left(\frac{1}{2}||\mathbf{w}||^2 + C \sum_{i=1}^{N} \xi_i \right) \\ \text{sujeto a } y_i(< \mathbf{v}_i, \mathbf{w} > + b) \geq 1 + \xi_i, \\ \forall i = 1, ..., N. \end{gathered} \tag{4}$$

El clasificador que utilizan en el código es una SVM con un kernel homogéneo chi-cuadrado (basado en la distacia chi-cuadrado). Para utilizar este kernel utilizan una función *vl_homkermap* que realiza una aproximación del mismo. Esta función tiene en cuenta un parámetro γ que hace referencia a una normalización de los datos, para este caso utilizan un $\gamma = 0.5$ lo que tiende a reducir el efecto de altos picos en los histogramas [3].

Table 1. ACA de Experimentos de Imagenet



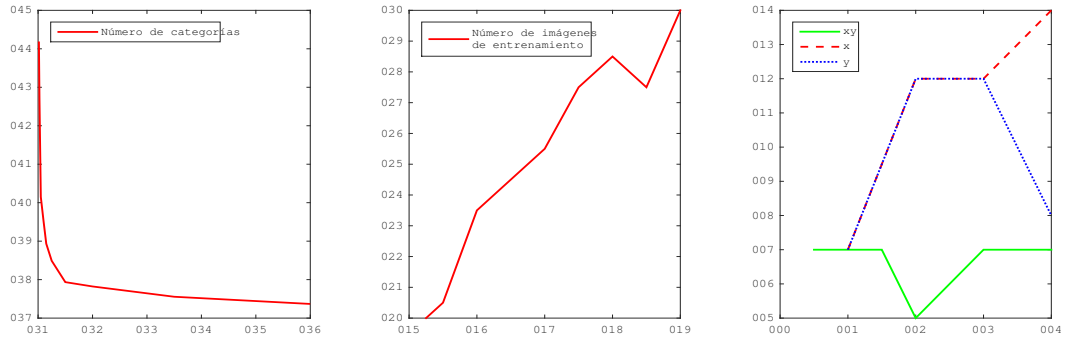


Figure 2. Graficas de los ACA de los experimentos en Imagenet

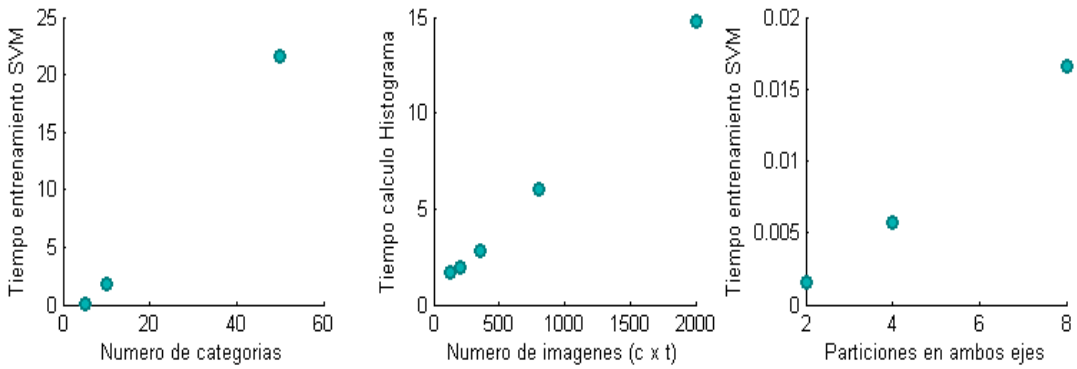


Figure 3. Variacion del tiempo de pocesamiento según el numero de imagenes y particiones

3.3. Modificaciones

Los descriptores y clasificadores utilizados en PHOW pueden ser utilizados para cualquier imagen, por lo cual, no es necesario modificarlos para poder utilizar el algoritmo en une nueva base de datos. En cambio, el código se tuvo que modificar para permitir la lectura de las imágenes. La diferencia más clara, es la dirección de la carpeta donde se leen las imágenes. Pero más allá de eso, la mayor diferencia se debe a que en Chaltec101 las imágenes de entrenamiento y evaluación no está separado en carpetas diferentes. Por ejemplo, si se querían 20 imágenes de entrenamiento y 10 de evaluación, el código original seleccionaba 30 imágenes de cada categoría, y después utilizaba el módulo 15 para identificar los índices de las primeras 20 imágenes de cada categoría. Es decir, el nombre de todas las imágenes se guardaban en una misma matriz, y tienen un vector con todos los índices de las imágenes de entrenamiento, y otro vector, con los índices de evaluación. Para adaptar el código a Imagenet, fue necesario modificar esta parte del código para que todas las imágenes de entrenamiento provinieran de la carpeta de entrenamiento, y lo mismo con las de evaluación. Finalmente, se agregó el ACA y las categorias a las varibales guardadas para poder tener acceso facil y rapido a esta.

4. Experimentos

En esta práctica, no se modificaron los parámetros del diccionario. En cambio, se quiso evaluar el efecto del (1) el número de categorías, (2) el número de imágenes de entrenamiento, (3) la partición espacial de las imágenes. Con este objetivo, se entrenaron y validaron varios modelos en la base de datos de entrenamiento. Cabe resaltar que todos los experimentos se corrieron con 20 imágenes de validación.Los resultados optenidos se presentan en la tabla 1 y la figura 2.

Para empezar, se varió el número de categorías. Para esto, se fijó el número de imágenes de entrenamiento a 20, y la partición a 2 en 'x' y 2 en 'y'. Después fijamos las categorías a 5 y se modificaron el número de entrenamiento hasta 80. Este fue el número máximo posible porque imagenet tiene 100 imágenes de entrenamiento por categoría, y se estaban utilizando 20 para validación. Finalmente, variamos la partición espacial de las imágenes. Primero se mantuvo igual el número de particiones en 'x' y 'y', y después se realizaron particiones desiguales.

Se encontró que los mejores parámetros fueron 80 imágenes de entrenamiento, y una partición de 2 en 'x', y 8 en 'y'. Es importante notar que el número de categorías no es un parámetro del modelo; en cambio, es un parámetro del problema: si la parte del problema que se está trabajando es más o mejor sencilla.

Finalmente, corrimos unos pocos experimentos en caltech101 variando cada uno de los parámetros anteriores, numero de categorías, numero de imágenes de entrenamiento y divisiones de la imagen. Esto se realizó para comparar estas variaciones con las de Imagenet.

Table 2. Tiempos diferentes dependiendo del modelo

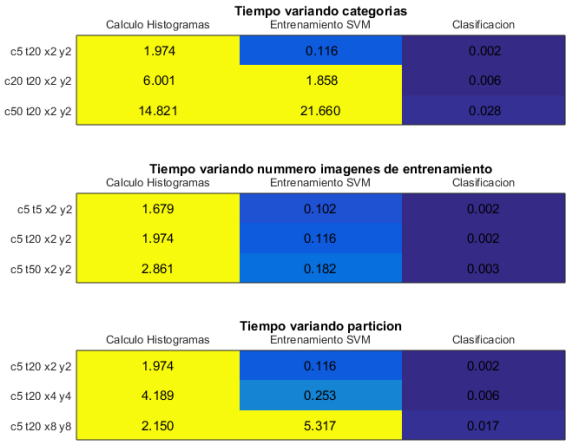
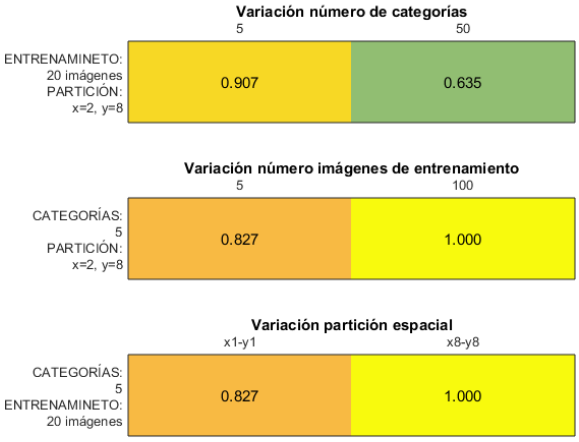


Table 3. ACA de Experimentos de Caltech



5. Resultados

Los resultados obtenido para la base de datos de test se muestran en la tabla 4 y los resultados de la matriz de confusión de la figura 5. Se calcularon los resultados para diferentes números de categorías para ver cómo se proyectaba el modelo de clasificación hacia bases de datos mayores. Se utilizaron entonces 3, 10 y 100 categorías, esto debido a que el tiempo computacional para mayor número de categorías

era demasiado. Estos resultado se obtuvieron utilizando 100 imágenes de entrenamiento, 100 de evaluacion, y una partición de $x = 2$ y $y = 8$. El mejor resultado se obtuvo en el menor número de categorías elegido y a medida que fue aumentando el ACA decrecía.

Table 4. Resultados en la base de datos de Test

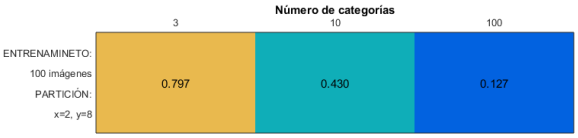
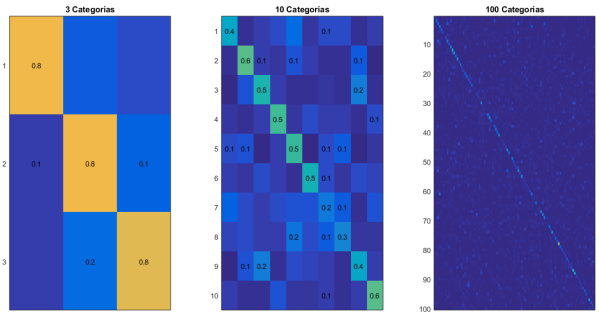


Table 5. Matrices de Confusión en la base de datos de Test



6. Discusión

Basados en los resultados de los experimentos podemos discutir varios efectos de la variación de los parámetros, en los numerales siguientes.

6.1. Efecto del número de categorías

Los resultados de los experimentos variando el número de categorías, fila 1 de la tabla 1, se observa que a medida que a medida que el número de categorías aumenta disminuye el desempeño del clasificador, es decir que a menor número de categorías mejor clasifica el modelo. Esto es de esperarse puesto que, entre más categorías, hay más posibilidades de equivocarse en una clasificación. Al cambiar este parámetro, no se está mejorando o empeorando el modelo, simplemente se está planteando un problema más o menos sencillo. Observamos que el aumento de categorías aumenta la dificultad del problema rápidamente. De hecho tiene un comportamiento inverso como se puede observar en la gráfica 1 de la figura 2, siguiendo la forma 1/(numeroCategorias). Se puede concluir que nuestro modelo no tiene buenos resultados a gran escala y no es tan fácil extrapolar el modelo. Por otro lado al aumentar el número de categorías, por ende el número de imágenes, el tiempo de demora aumentó en gran medida, para la fase de clasificación de los datos de validación con 3 categorías el procesamiento completo duró 0.0949 segundos y con 100 categorías 1.1733. Con esto se puede observar que realmente el costo computacional que esto implica es mucho mayor. El mejor resultado se obtuvo cuando se tenían menos categorías a clasificar.

6.2. Efecto del tamaño de entrenamiento

Después, fijamos las categorías en 5 y se modificaron el número de entrenamiento hasta 80. Este fue el número máximo posible porque Imagenet tiene 100 imágenes de entrenamiento por categoría, y se estaban utilizando 20 para validación. Los efectos en los resultados del modelo variando el tamaño de los datos de entrenamiento presentó un comportamiento creciente casi lineal, este se puede observar en la gráfica 2 de la figura 2 . Así mismo los datos de la tabla 1 lo demuestran con una pendiente aproximada 0.0026. Este comportamiento indica que a mayor número de imágenes de entrenamiento mejor es el clasificador. Esto se debe a que un mayor número de datos de entrenamiento permiten aumentar crear un modelo que se ajuste más a los datos. Sin embargo se sabe que en algunos casos es peligroso aumentar el número de imágenes de entrenamiento indefinidamente, porque esto puede causar sobrenetrenamiento del modelo. No fue posible hacer pruebas con las 100 imágenes de entrenamiento ya que para la validación era necesario usar imágenes también. Nuestra hipótesis de que con todas las imágenes de entrenamiento los resultados son mejores por lo que se decidió utilizar 100 imágenes de entrenamiento para la evaluación final en el conjunto de test y se la hipótesis. El mejor resultado se obtuvo utilizando 80 imágenes de entrenamiento.

6.3. Efecto del número de particiones espaciales

Para el efecto del número de particiones espaciales, para realizar el descriptor, a partir de las últimas dos filas de la tabla 1 y de la gráfica 3 de la figura 2 podemos analizar el comportamiento de estas variables y efecto sobre la exactitud promedio de clasificación. Cuando se varían las dos dimensiones al tiempo en la misma proporción el ACA no cambia, más bien se mantiene constante o tiende a disminuir. Al aumentar el número de particiones en el eje x se puede observar que aumenta en un principio de 2 a 4 y luego disminuye cuando cambia de 6 a 8. En cambio al aumentar y se puede observar que a mayor número de particiones mejor es la clasificación. Esto se puede dar debido a que las imagen pueden tener una variación mayor en el eje vertical, lo que puede ser un efecto producido porque vivimos en un mundo con gravedad, y la mayoría de imágenes tienen un alto contraste entre la parte de arriba de la imagen y la parte de abajo. El mejor resultado se obtuvo con una partición de 2 en ‘x’, y 8 en ‘y’.

6.4. Tiempos de procesamiento

Al variar el número de categorías, las imágenes de entrenamiento, y las particiones, no solo afecta la precisión del algoritmo, también cambian los tiempos de procesamiento. El algoritmo se puede dividir en varios procesos: (1) obtener el diccionario, (2) calcular los histogramas, (3) entrenar los SVM, (4) clasificar las imágenes. El tiempo del cálculo del diccionario debería ser constante pues ninguno de los parámetros variados afecta esta parte del proceso (siempre se calcula el mismo número de palabras a partir del mismo número de vectores).
En la tabla 2 se puede observar que, de manera general, el proceso que tiempo computacional es el cálculo de los SIFT. Este aumenta cuando se incrementa el número de categorías y el número de imágenes de test. Lo anterior se debe a que el número de imágenes totales (y por lo tanto, el número de imágenes a las que hay que calcularle el SIFT), está dado por (numEntrenamiento+numTest)*(numCategorias). En la figura 3, la segunda gráfica muestra que el tiempo del cálculo de los histogramas aumenta linealmente con el número de imágenes.
Adicionalmente, el tiempo del entrenamiento de los SVM aumenta cuando cambia el número de categorías y cuando aumentan las particiones. En el primer caso, más categorías significa que se deben entrenar un mayor número de SVM (grafica 1 de la figura 3). En el segundo caso, mayor partición significa vectores de representación más grande, lo cual implica buscar un hiperplano en un espacio de mayor dimensión (grafica 3 de la figura 3) .

6.5. Comparación entre los resultados de Caltech101 e Imagenet

Corrimos unos pocos experimentos el caltech101. Puesto que esta base de datos no tiene entrenamiento y validación separados, no hay diferencia entre los datos que se pueden obtener en 'experimentos' y 'validación'. Por esta razón, a diferencia de lo realizado con imagenet, con esta base de datos no se realizó una evaluación final con los parámetros escogidos.
Mirando los resultados, se observa la misma tendencia: el aumento en el número de clases disminuye el ACA, y este mismo aumenta cuando se agregan imágenes de entrenamiento. Adicionalmente, una mayor partición ayuda a clasificar con mayor exactitud, y, a diferencia de imagenet, si se obtuvo un aumento significativo cuando el número de particiones en 'x' se mantuvo igual que el número de particiones en 'y'.
Finalmente, la diferencia más notoria entre los resultados es la magnitud los ACA obtenidos. Por ejemplo, con 5 categorías, 20 imágenes de test, y 2 particiones en ambos ejes, se obtuvo un 0.907 en caltech101, y 0.570 en imagenet. Esta diferencia se debe a la dificultad de las imágenes. Caltech101 en su mayoria son imágenes de catálogo: con fondo blanco, objeto centrado grande y son oclusiones. Por otro lado, imagenet tiene imágenes cotidianas, lo cual implica un grado de dificultad mucho mayor. Con imagenet, el clasificador debe tener en cuenta diferentes escalas, posiciones y oclusiones para poder clasificar correctamente una imagen.

7. Limitaciones

El método de PHOW es ideal para una base de datos como caltech101, donde se observa que resuelve el problema. No obstante, en una base de datos más compleja, ya no es suficiente. Además de la disminución en exactitud, este algoritmo no es ideal para una base de datos como Imagenet debido a su tiempo de computación. Calcular los histogramas de cada imagen es un procesamiento bastante pesado, y cuando se tienen 996 categorías, el número de imágenes totales es demasiado grande: durante este laboratorio, no se corrió el método con todas las categorías con las 100 imágenes de test dada la complejidad computacional y el tiempo que esto requería.
Otra limitación es el clasificador. Los SVM son clasificadores binarios, y aunque se pueden entrenar muchos SVM (una clase contra el resto) para poder clasificar en múltiples categorías, esto no es ideal. Entre más categorías hayan, mas desbalanceados serán cada SVM individual, lo cual no es ideal. Sería conveniente intentar otros clasificadores que están diseñados para múltiples categorías, como los árboles.

8. Mejoras

A lo largo de este laboratorio, solo se cambiaron dos parámetros: el número de imágenes de entrenamiento y las particiones. Para mejorar el modelo, es posible experimentar con otros parámetros significativos.
Por un lado, están los parámetros del diccionario de SIFT. Es posible variar el tamaño de la grilla ('Steps'), o la escala ('Size'), el número de vectores utilizados para calcular el diccionario, y el número de palabras. Adicionalmente, el algoritmo utilizado convierte la imagen a escala de grises antes de calcular SIFT, pero este puede ser computado para imágenes a color (RGB, HSV, o La*b*).
Por otro lado, se pueden lograr mejoras en el modelo optimizando los parámetros del clasificador, el C, el γ, el orden de la aproximación del kernel y la distancia utilizada.

References

[1] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[2] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, 106(1):59–70, 2007.

[3] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(3):480–492, 2012.