Clasificación de objetos por extracción de descriptores tipo PHOW

Juan Sebastián Cuéllar
Universidad de los Andes, Bogotá, Colombia
js.cuellar169@uniandes.edu.co

1. Introduction

En este artículo se describe y se evalúa un método de clasificación a partir de los descriptores de las pirámides de histogramas de palabras (PHOW) para solucionar uno de los problemas fundamentales de la visión computacional: reconocimiento y clasificación de objetos en una imagen. Para evaluar el desempeño de los métodos se hará uso de una parte del set de imágenes del ImageNet Large Scale Visual Recognition Challenge (ILSVRC), del cual se utilizarán las imágenes de entrenamiento y sobre las cuales se dividirán los grupos de test y entrenamiento. Se utilizará el código creado por Vedaldi [1] como herramienta para extraer los descriptores PHOW y para clasificar las imágenes de test. Posteriormente se definirá su rendimiento con matrices de confusión.

El subset de imágenes del ImageNet Large Scale Visual Recognition Challenge (ILSVRC) está constituido por 20000 imágenes a color, en formato JPEG y con una resolución de 256 x 256 pixeles. Está dividido en 200 categorías semánticas cada una compuesta de 100 imágenes.

2. Descripción del método de reconocimiento

El código de Vedaldi [1] recibe en principio la dirección de la carpeta donde se encuentran las categorías, el número de categorías a evaluar, el número de imágenes de entrenamiento y el número de imágenes de test. Cabe resaltar que el método recibe otros parámetros de ajuste que no son de vital importancia para este estudio. A partir del número n de categorías de entrada, reorganiza las carpetas en una celda (primero mayúsculas y luego minúsculas en orden alfabético) y toma las primeras n posiciones para clasificación. Dentro de las carpetas tomadas se extrae el número indicado de imágenes de test y de entrenamiento aleatoriamente. A partir de las imágenes elegidas se calculan los descriptores usando (PHOW), que incorpora la información espacial dentro una bolsa de palabras (BoW) usando pirámides espaciales (SP) [5]. La BoW es un modelo que maneja los descriptores individuales como palabras y las retiene en un histograma [2]. El método PHOW divide la imagen en diferentes parches y subparches en donde se calculan los histogramas de las orientaciones de los gradientes para todos los pixeles incluidos en dicho subparche. La cantidad de subparches del algoritmo se puede ajustar modificando ciertos parámetros que permiten ingresar el número de particiones espaciales [5]. Con el espacio de representación definido se entrena un support vector machine (SVM) con kernel chi-cuadrado usando la base de entrenamiento y finalmente se emplea la base de test para probar el clasificador. Por último se calcula la matriz de confusión y se halla el promedio de su diagonal para definir el desempeño. Para los propósitos de este estudio el método se utiliza para cuatro números de categorías diferente (10, 50, 100 y 200) usando un solo número de particiones espaciales en el eje x y cinco cantidades diferentes de imágenes de entrenamiento (3, 6, 9, 12, 15). Para ver el efecto del número de particiones espaciales en el eje x se utiliza el método en tres cantidades diferentes ([1 2], [2 4], [4 8]) con un número fijo de categorías (100) y cinco cantidades diferentes de imágenes de entrenamiento (3, 6, 9, 12, 15). Para mostrar una base de comparación con el set de imágenes utilizado se tomará el desempeño del algoritmo para la base de datos caltech-101 [3].

3. Resultados

En la fig 1 se evidencia la tendencia de reducción del desempeño del método con la inclusión de más categorías y una mejoría con el aumento de imágenes de entrenamiento. La tendencia de las curvas permite considerar un comportamiento exponencial entre las exactitudes y el número de imágenes de entrenamiento. Así mismo se muestra que la cantidad de aciertos máxima se alcanza con 10 categorías y 12 imágenes de entrenamiento (60%) y la mínima con 200 categorías y tres imágenes de entrenamiento (0.7%). Adicionalmente, se puede observar el sobresalto de rendimiento alcanzado en el set de imágenes caltech-101 para 15 imágenes de entrenamiento y su reposicionamiento alcanzando el mejor resultado (69%). Por otro lado, con 10 categorías se observa un pico de rendimiento que decae rápidamente con el aumento de imágenes de entrenamiento. En la fig 2 se observan pequeñas variaciones entre las cantidades de particiones espaciales lo que indica

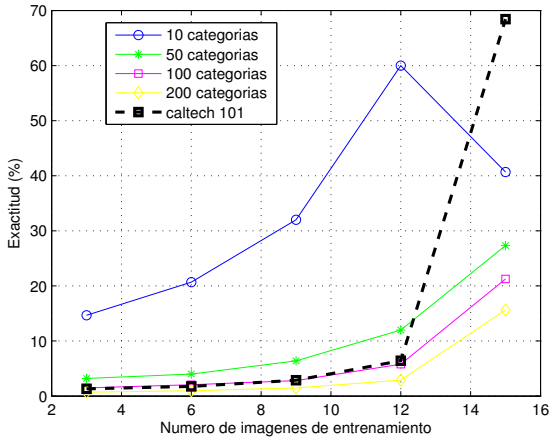

Figure 1. Curvas de exactitud vs número de imágenes de entrenamiento por categoría para un número constante de particiones espaciales.

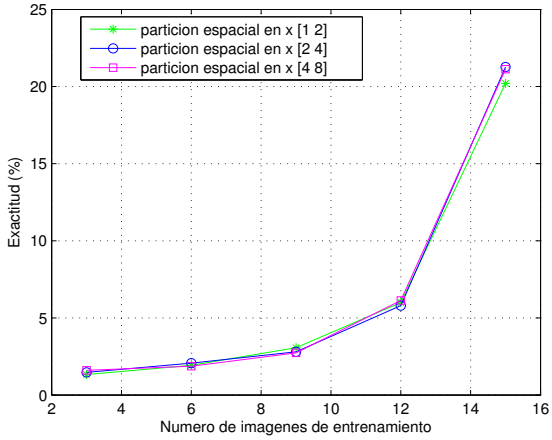

Figure 2. Curvas de exactitud vs número de imágenes de entrenamiento por categoría para un número constante de categorías.

un desempeño poco dependiente de dicho parámetro.

4. Discusión

La evaluación del algoritmo se limitó a 15 imágenes de entrenamiento y 15 imágenes de test por cada categoría evaluada. Aunque este grupo reducido funciona como muestra representativa de todo el set, no permite utilizar todo el potencial de la base de datos para extraer mejores descriptores y emplear su posterior uso práctico en la clasificación de imágenes fuera del set. Esto sumado a la amplia variedad de formas, deformaciones y tamaños de las imágenes para cada categoría en comparación a caltech-101 podría explicar la diferencia en cuanto al desempeño del algoritmo entre las dos bases de datos. Adicionalmente, es pertinente utilizar una mayor cantidad de imágenes de test para encontrar una mayor cantidad de respuestas de clasificación y calcular una matriz de confusión mejor descrita en términos probabilísticos.

Incluir dentro de la clasificación un número alto de categorías implica obtener espacios de representación muy variados y bien definidos en cada categoría para establecer un buen clasificador. Utilizar una base de entrenamiento de 15 imágenes no resulta suficiente para describir todo el set, por ende, clasificar más de 50 categorías resulta en un desempeño poco satisfactorio y además muy dispendioso computacionalmente porque se necesita emplear un clasificador para cada categoría. Aunque los mejores resultados se obtienen con 10 categorías, el objetivo final es clasificar correctamente las 200 categorías, por lo que 10 categorías resultan insuficientes.

Aunque el rendimiento de los algoritmos no es muy prometedor, los resultados muestran una tendencia incremental del desempeño con el aumento del tamaño del set de entrenamiento en tres diferentes números de categoría (50, 100, 200). Para estas cantidades es posible obtener mejores rendimientos utilizando más imágenes de entrenamiento. Sin embargo, la evaluación para 10 categorías muestra un pico de rendimiento con 12 imágenes de entrenamiento que indica una sobrespecialización del clasificador a partir de este umbral, esto muestra que un aumento desmedido del tamaño del set de entrenamiento también es perjudicial para el desempeño del clasificador. Adicionalmente, incluir más imágenes de entrenamiento supone un gasto de memoria computacional muy apreciable puesto que la extracción de los descriptores utiliza grandes recursos computacionales. Por consiguiente, se debe definir un tamaño apropiado de base de entrenamiento teniendo en cuenta las compensaciones que se presentan.

Para este caso particular el número de particiones espaciales no afecta el desempeño del algoritmo significativamente. Esto se puede deber a que una partición más fina no implica un cambio significativo en los descriptores del subset de entrenamiento empleado, además, particiones finas implican más cálculos para la extracción de los descriptores lo que vuelve el método más costoso computacionalmente.

Los SVM consumen mucha memoria computacional y la elección del kernel no siempre permite los mejores resultados [4]. En efecto la mayoría de los recursos computacionales del algoritmo se emplean en el entrenamiento del SVM lo que lo hace poco práctico para sets de entrenamiento grandes. Para mejorar el desempeño del método se pueden utilizar otros kernels más elaborados dentro del SVM que permitan una mejor clasificación de los descriptores a costa de velocidad de procesamiento, por otro lado, para incrementar la velocidad de procesamiento se puede cambiar todo el SVM por otras alternativas como random forests o nearest neighbor, a riesgo de presentar menores

desempeños.

References

[1] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. 2008.

[2] J. Wu and J. M. Rehg. CENTRIST: A visual descriptor for scene categorization. *IEEE Transactions on Pattern Analysis and Machine Learning*, 33(8):1489–1501, 2011.

[3] 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. *IEEE. CVPR 2004, Workshop on Generative-Model Based Vision*, 2004.

[4] Laura Auria and Rouslan A. Moro. Support Vector Machines (SVM) as a Technique for Solvency Analysis. *DIW Berlin German Institute for Economic Research*, Aug. 2008.

[5] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.