CLASIFICACIÓN CON PHOW

Juan Pablo Moreno Ortiz
Maestría en Ingeniería Biomédica
Universidad de los Andes

INTRODUCCIÓN

Una de las formas de clasificación de imágenes se da a partir de la extracción de características descritas como pirámides de histogramas de las palabras visuales (PHOW), estas palabras visuales son vectores cuantificados con SIFT, o en el caso del método utilizado en la librería vl_feat dense SIFT, que es un método rápido para el cálculo; estas palabras visuales capturan la apariencia visual local dentro de la imagen de acuerdo a su distribución.

En el problema de clasificación de imágenes se han desarrollado variedad de programas, en este laboratorio se utiliza la librería de código abierto desarrollada por VLFeat.org, dentro de la cual se encuentra el software para ejecutar y editar en Matlab, en principio este se ejecuta sobre la base de datos de Caltech 101, que contiene 102 categorías para la clasificación, el objeto que define la categoría de la imagen se encuentra ocupando la mayor parte de esta, por lo general el objeto se encuentra completo dentro de la imágen y cuenta con imágenes muy similares entre sí para cada categoría, donde los objetos aparecen con la misma orientación. Por otra parte se tiene la base de datos de Imagenet que cuenta con 200 categorías con imágenes más realistas y con una complejidad mayor, en esta base de datos los objetos por lo general no cuentan con la misma orientación, no se ven completos en todas las imágenes y cuentan con diversas posiciones y tamaños dentro de la imagen, la complejidad es tan alta que en algunas imágenes no es posible a simple vista diferenciar entre categorías por humanos inexpertos.

MATERIALES Y METODOS

VLFeat

Es una organización que desarrolla software libre para el tratamiento de imágenes, que desarrolló una librería de software para Matlab, dentro de la que se encuentra el script "phow_caltech101", utiliza PHOW para clasificar las imágenes, si bien este programa en un principio está desarrollado para funcionar con la base de datos de caltech101 este puede ser editado.

Como primera medida es necesario ejecutar el archivo vl_setup.m que se encuentra en la carpeta "toolbox" de la librería, después de esto se puede ejecutar el script lo que iniciará la descarga de la base de datos caltech101 desde internet, el programa viene configurado para tomar 15 imágenes de entregamiento, 15 de prueba, identifica 102 clases, utiliza 600 palabras visuales que son el resultado de la cuantificación de una característica por medio de un algoritmo de clustering, en este caso se utiliza la función kdtree, esta función aleatoria habilita el algoritmo rápido de vecinos cercanos para medianas y grandes escalas. El método PHOW utiliza Support Vector Machine (SVM) para la separación de los datos de entrenamiento, por esto es necesario configurar el lambda del SVM que depende del valor declarado en conf.svm.C, esta variable por defecto viene en un valor de 10. La cuantificación de las palabras visuales se hace tomando secciones de la imagen, de esta forma no se pierde

la información espacial del análisis, estas secciones o parches vienen establecidos por las variables numSpatial, tanto para el eje x como para y.

El algoritmo inicia con la configuración inicial de variables, en este punto se modifica la dirección del directorio donde se encuentra almacenada la base de datos, que en este caso se direcciona a la base de datos de imagenet; también permite limitar el problema de clasificación habilitando la opción tiny, que toma un menor número de clases, de palabras visuales, disminuye el espacio de análisis de cada ventana y utiliza el prefijo Tiny para los archivos que se generan, en la ejecución del código para la base de datos de prueba se mantuvo en false la opción tiny, y se modificó la cantidad de imágenes de entrenamiento, el número de clases, el número de palabras visuales y el número de particiones espaciales xy.

Luego se encuentra la sección de código que permite la descarga de las imágenes, para la aplicación específica con imagenet fue necesario eliminar esta sección para que no se generará el error por haber cambiado la opción autodonwnload data a false. La siguiente sección dentro del código se titula Setup Data, esta configura finalmente la dirección donde se encuentra la base de datos que se va a utilizar, en este punto del código es necesario hacer una modificación puesto que viene para archivos con extensión jpg, mientras que la base de datos a utilizar contiene las imágenes en formato JPEG, si esto no se modifica genera error; esta sección también define las etiquetas de las clases a partir del nombre de las carpetas contenedoras de las imágenes.

Una vez se termina la configuración del proceso se pasa al entrenamiento del vocabulario, en este se utiliza la función "vl_phow" que extrae las características PHOW de la imagen, esto aplica a su vez la función "vl_sift" a diferentes resoluciones, en la misma sección se toma el resultado de vlphow y se cuantifican los descriptores utilizando kmeans para formar las palabras visuales, este resultado se almacena en el archivo vocab. Luego se computan los histogramas espaciales, que son guardados en el archivo hists, este procedimiento permite observar su ejecución al mostrar el porcentaje de procesamiento sobre cada una de las imágenes. Después el algoritmo genera el mapa de características que se hallan aplicando la función "vl_homkermap", configurado por defecto para calcular el kernel de intersección de chi2 con un grado de homogeneidad del kernel dado por gamma que viene configurado en 0.5.

El Support vector machine es entrenado aplicando la función "vl_svmtrain" para esta sección viene por defecto la opción de solucionador lineal "Stochastic Dual Coordinate Ascent" (SDCA) que permite optimizar la ejecución del SVM sobre los datos de entrenamiento. A esta función entra el mapa de características de las imágenes seleccionadas como de entrenamiento, las etiquetas son el segundo parámetro y son extraídas de los nombres de las carpetas que contienen las imágenes de prueba, la función de entrenamiento genera dos modelos, uno de peso (w) que contiene la puntuación de cada punto evaluado, otro de ajuste (B), ambos son almacenados en el archivo model para ser utilizados en la sección siguiente del código.

Finalmente se utilizan las imágenes de prueba para evaluar el funcionamiento del SVM entrenado, se obtiene la puntuación de cada imagen nueva para definir la clase a la que pertenece, esto a partir de los modelos W y B generados en el entrenamiento del SVM. De

acuerdo con la puntuación máxima para alguna de las clases que presente la imagen, esta recibe esa clasificación. A partir de este resultado se genera la matriz de confusión y de puntuaciones.

RESULTADOS

Los resultados del clasificador se guardan en archivos que llevan un prefijo que puede ser cambiado dentro del script, dentro de los archivos con extensión .mat que genera, se encuentran los histogramas, el vocabulario de palabras visuales generadas y los resultados de la clasificación. A continuación se observa la matriz de confusión tanto para el programa sin editar con las 102 categorías de caltech 101, como para el programa utilizando Imagenet que cuenta con 200 categorías, por razones prácticas del procesamiento se aplicó el algoritmo a tan solo a 50 categorías o clases y se utilizaron 50 imágenes de entrenamiento.

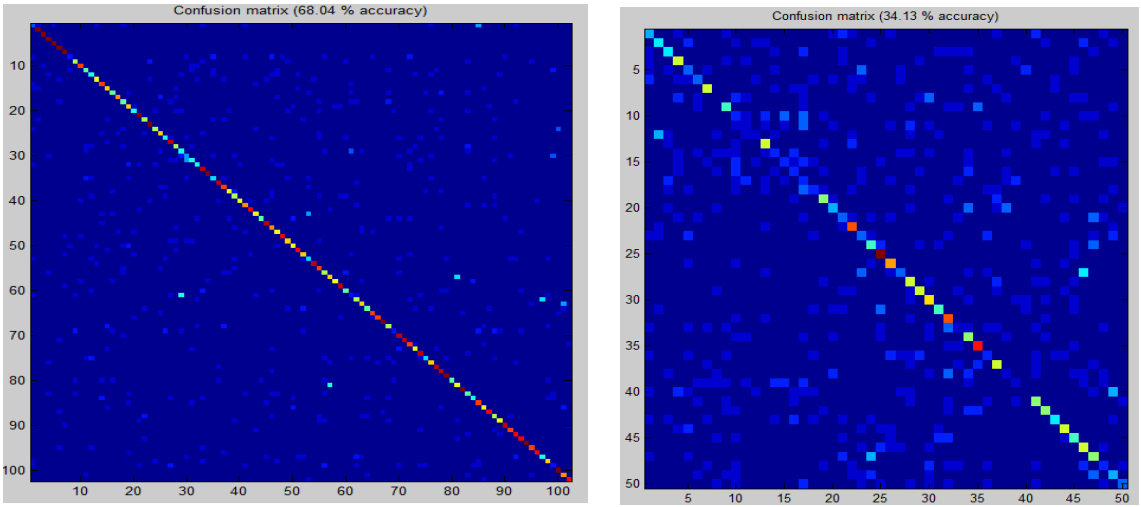
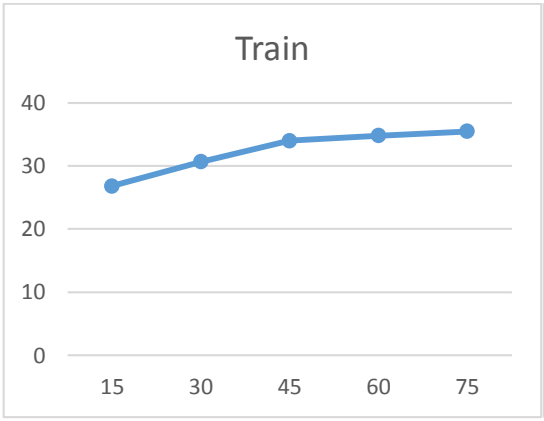
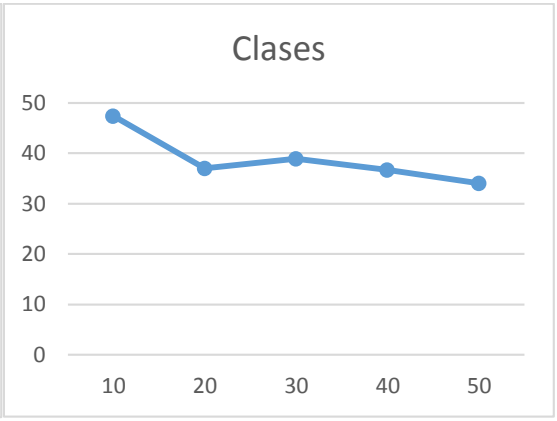


Ilustración 1. Matriz de confusión, Caltech 101, Image Net

En el resultado se evidencia la dificultad de las imágenes en la base de datos, pues al utilizar caltech 101 alcanza un 68.04% de exactitud, con solo 15 imágenes de entrenamiento para todas las categorías, mientras que utilizando 50 imágenes de entrenamiento en imagenet solo alcanza un 34.13% de exactitud, esto teniendo en cuenta 50 clases; la configuración del algoritmo es igual en ambos casos, exceptuando el número de clases y el número de imágenes de entrenamiento, por esto se puede pensar que la diferencia entre los resultados se debe a una mayor complejidad de las imágenes de la base de datos imagenet, comparada con caltech 101.

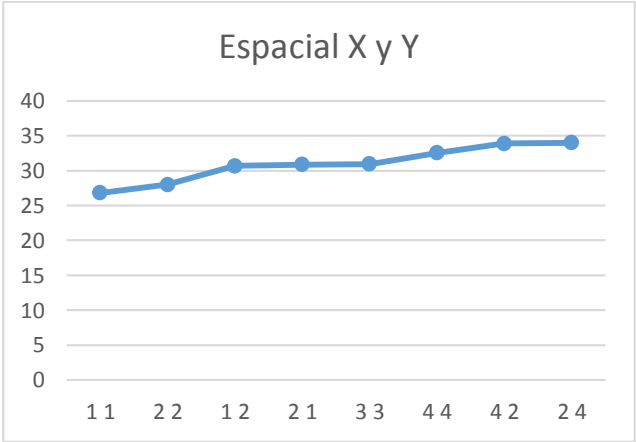
Para evaluar el efecto del número de imágenes de entrenamiento fue variada esta configuración manteniendo el número de clases constante en 50 y el tamaño de los parches en [2 4]. De la misma forma se evaluó el efecto de la modificación en el número de clases manteniendo 45 imágenes de entrenamiento con el mismo tamaño de los parches que la evaluación anterior. Finalmente se evaluó el tamaño de los parches que evalúan la imagen modificándolos igual tanto en x como en y, manteniendo 50 clases y 45 imágenes de entrenamiento. A continuación se observan las gráficas de la variación en la exactitud para los diferentes casos.





Se observa una mejoría en la exactitud en la clasificación del algoritmo a medida que aumenta la cantidad de imágenes de entrenamiento, esta mejoría es significativa hasta las 45 imágenes, después de esto aunque aumenta el requerimiento de procesamiento no sucede igual con la exactitud, siendo así que de 15 a 45 imágenes de entrenamiento aumenta en 7,2% la exactitud, mientras que de 45 a 75 aumenta solo 1,47%, por esta razón se utilizan 45 imágenes de entrenamiento para las demás pruebas.

Con respecto al número de clases es evidente la desmejora en el desempeño del clasificador a medida que aumentan las clases o categorías de imágenes, en este caso se seleccionaron 50 categorías para las demás pruebas, principalmente por el requerimiento de procesamiento que aumenta a medida que aumentan las categorías. Por lo cual el computador personal no soportaba un número alto de categorías alcanzando en algunos momentos el 100% de la capacidad del procesador y del disco duro, alcanzando tiempos de ejecución de más de una hora en la clasificación.



El espacio del parche definido tanto para x como para y, influye sobre la medida alcanzando la mejor exactitud para la configuración con la que cuenta por defecto, esta es la misma para el eje x como para el eje y [2 4], cabe resaltar que en todos los casos fue superior la combinación de términos diferentes que cuando se utilizaba el mismo valor en los dos espacios.

Este método presenta dos limitaciones importantes, por un lado se encuentra un alto requerimiento de recursos tanto en tiempo, capacidad de procesamiento, memoria y disco duro. La otra limitación es la exactitud en la clasificación de las imágenes alcanzando con los parámetros de prueba. La forma en que se podría mejorar es utilizando métodos de medición que cuenten con una mayor eficiencia. Para mejorar el desempeño también se puede probar el script en equipos con mayor capacidad de procesamiento, es necesario tener en cuenta la disminución en la capacidad de clasificación entre las dos bases de datos, esto evidencia que si bien las imágenes de imagenet son más realistas que las de la otra base de datos puede ser que no aporten la información suficiente para clasificarlas.