Etiquetado Automático de Imágenes Usando Múltiples Segmentaciones Basándose en Modelos Probabilistas

Gerardo Arellano Cervantes
Instituto Nacional de Astrofísica Óptica Y Electrónica
Cordinacción de Ciencias Computacionales
Modelos Gráficos probabilistas
garellano@ccc.inaoep.mx

Abstract. La anotación automática de imágenes consiste en etiquetar imágenes o regiones de manera automática. En una clasificación supervisada, se requiere un conjunto de imágenes previamente segmentadas manualmente y etiquetadas para entrenar un clasificador, el etiquetado de imágenes de forma manual se realiza por medio de segmentaciones de regiones de la imagen y asignarle su clase correspondiente, La segmentación de imágenes ha demostrado ser una de las mejores formas para identificar objetos y detectar regiones en una imagen [1][3]. Ninguno de los muchos segmentadores que existen hoy en día son capaces de dividir una imagen en todos los objetos correctamente. Se propone utilizar múltiples segmentaciones de la imagen[11], para posteriormente verificar y decidir cuales fueron las mejores segmentaciones y hacer una selección de estas segmentaciones con el objetivo de fusionar estas segmentaciones y realizar un mejor reconocimiento de los objetos y posteriormente el etiquetado.

Key words: Segmentación, Clasificador bayesiano, Etiquetado de regiones.

1 Introducción

La cantidad de base de datos de imágenes digitales ha crecido de manera sorprendente en los últimos años, esta situación demanda eficiencia en los métodos de búsqueda para la extracción de imágenes. La anotación automática de imágenes consiste en etiquetar imágenes o regiones de manera automática, uno de los métodos para realizar el etiquetado consiste en comenzar a segmentar una imagen en diferentes regiones, reconocer la segmentación y asignarle su etiqueta correspondiente a la clases definidas. En el enfoque de clasificación automática se necesita un conjunto de imágenes previamente etiquetadas manualmente, (Fig. 1 se muestra un ejemplo), que son usadas para entrenar un clasificador, y entonces una vez teniendo listo nuestro corpus de entrenamiento el clasificador es usado para etiquetar el resto de las imágenes, en algunos casos las etiquetas son asignadas a

una región especifica de la imagen o a la imagen completa. En el enfoque de la clasificación supervisada también depende de la calidad y cantidad del conjunto de entrenamiento el cual fué etiquetado de manera manual, ya que si se tiene un buen corpus de entrenamiento la clasificacion puede mejorar sustancialmente. Hoy en día no existe un segmentador capaz de dividir una imagen en múltiples regiones de manera completamente correcta, es decir que no existe el segmentador perfecto, si no que hay segmentadores son buenos para algunas clases, algunos otros son buenos para otro tipo de clases de objetos aunque pudieran ser lentos, y otros no son tan buenos para segmentar pero si son rápidos en procesamiento, el objetivo es poder segmentar una imagen con diferentes segmentadores [2][4], pudiendo hacer una combinación de diferentes segmentadores con diferentes características y con ayuda de modelos probabilísticos realizar una selección de cuáles fueron las mejores segmentaciones y hacer clasificación mejor de los objetos para las regiones segmentadas. El enfoque de la múltiple segmentación está apelando por su simplicidad y modularidad. Por otro lado la clasificación supervisada se realizó con un clasificador bayesiano simple, es decir que de los atributos que se obtuvieron de cada región segmentada se manejaron de manera independiente dada la clase,

Fig. 1. Segmentación Manual de Imágenes para entrenar un clasificador.

2 Trabajo relacionado

Existen algunos enfoques que han abordado esta forma de reconocimiento de imágenes en forma de multiples segmentaciones en los que han combinado segmentadores tales como Normalized Cuts[12], Felzenszwalb and Huttenlocher(FH) [13], Mean-shift [14] entre otros, para generar una sopa de segmetadores aunque algunos artículos reportan mejoras en cuanto el reconocimiento, la idea de la fusión entre sus segmentadores no queda

completamente clara como se realizó. Otros artículos han propuesto otros nuevo enfoque para el aprendizaje incorporando apariencia, información del contexto, incluyendo la forma de la imagen y haciendo algún tipo de boosting para objetos multiclase. Algunos otros artículos han querido atacar el mismo problema con un aprendizaje semi-supervisado, en el cual no se cuenta con una gran cantidad de imágenes o no se tiene las herramientas necesarias para procesar todas la imágenes porque pueda ser demasiado grande,o incluyendo que el etiquetado manual puede llegar a ser tedioso y pesado, aquí no desea para crear un buen corpus de entrenamiento para hacer la clasificación con mejores resultados, si no que con el corpus que tiene, trate de ir aprendiendo con las nuevas instancias que se vayan agregando, y posteriormente ir mejorando el corpus y por ende la clasificación.

3 Metodología y Desarrollo

El desarrollo de este artículo se siguió de la siguiente manera, (i) Se generó el conjunto de entrenamiento con un etiquetado de regiones de manera manual, (ii) Se extraen la características de cada región segmentada del etiquetado manual y se define su clase a la que pertenece esa región, una vez que se ha generado el conjunto de entrenamiento, se pasa a la etapa de segmentación de la imagen a ser procesada, (iii) Se segmenta la imagen con diferentes segmentaciones, (iv) se extraen las características de las regiones generadas con la diferentes segmentaciones, (v) Se selecciona la mejor segmentación de las regiones segmentadas

3.1 Conjunto de Entrenamiento

En este artículo, se utilizó el conjunto de datos de Microsoft Research Cambridge(MSRC) [5] de 9 clases, como observación este conjunto de datos esta densamente dividido en regiones y también está segmentado con casi todos los píxeles de la imagen, además de que contiene un gran número de categorías de clases, y es también comúnmente utilizado para evaluar algoritmos de segmentación para el reconocimiento de múltiples clases, en este artículo se redefinió el conjunto de datos, es decir solo se utilizarón 6 clases quedando un total de 394 regiones segmentadas de forma manual para el entrenamiento que se definieron edificio, césped, árbol, vaca, cielo y rostro como clases.

3.2 Segmentación de Imagen y Características Visuales

Antes de aplicar el método de clasificación supervisada para el etiquetado de imágenes, se realizaran las siguientes operaciones (i) Segmentación de regiones y (ii) Extracción de características. En este artículo se propone como etapa inicial realizar una segmentación sencilla de la imagen se realizarón múltiples segmentaciones en forma de rejillas(grids)[6], (Fig. 2) muestra un ejemplo). Se divide la imagen con 3 diferentes grids con diferente longitud, de 16, 20 y 32 píxeles cada segmentación, por lo que se obtuvo diferentes regiones, es decir sí tenemos una imagen de tamaño 320x210 se obtienen 260, 160 y 60 regiones para tamaño 16,20 y 32 píxeles respectivamente. Una vez que se obtienen estas regiones de rejillas con diferentes longitudes, cada región o segmento se representa por un conjunto características visuales y se construye el conjunto de prueba. Se consideró un conjunto de características para forma y color.

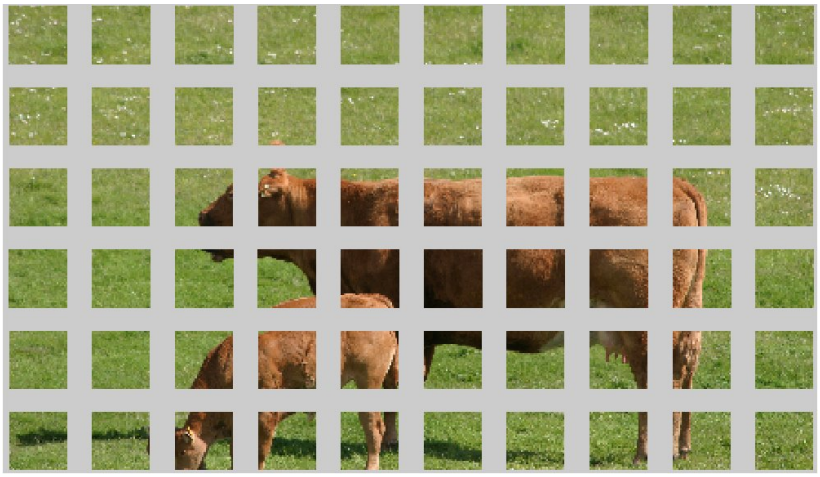

Fig. 2. Imagen Segmentada con grids de 32 píxeles de longitud.

3.3 Forma

Las características de forma define datos relevantes de una imagen en cuanto tamaño, longitud, posición, etc. de una imagen, se definen los atributos para forma de la siguiente manera:

- Área: Área normalizada de un segmento.
- Ancho: Anchura normalizada de un segmento.

- Alto: Altura normalizada de un segmento.
- Media x, y: Promedio de la posición de los píxeles x y y de un segmento
- Desviación Estándar x, y: Desviación estándar de la posición x y y de un segmento.
- Limite Área: Radio entre el límite y el área de un segmento.
- Convexidad: Convexidad de un segmento.

3.4 Color

Las características de color define datos relevantes de una imagen en cuanto color y luminosidad, se maneja el color con el modelo básico que es RGB, y tambien se consideró el espacio de color LAB, que se utiliza para medir la luminosidad de una imagen [7], se definen los atributos para color de la siguiente manera:

- Promedio RGB: Promedio de los valores RGB de un segmento.
- Desviación Estándar RGB: Desviación estándar RGB de un segmento.
- Asimetría RGB: Asimetría de los valores RGB de un segmento.
- Promedio Lab: Promedio de los valores Lab de un segmento.
- Desviación Estándar Lab: Desviación estándar Lab de un segmento.
- Asimetría Lab: Asimetría de los valores Lab de un segmento.

De cada uno de los atributos anteriores se obtuvieron 3 diferentes atributos cada uno, es decir que para promedio RGB, se obtuvieron el promedio para R, para G y para B. Lo mismo se realizó para los demas atributos incluyendo el espacio de color LAB.

3.5 Clasificador Base

Como clasificador base se usó el clasificador Naive Bayes, el cual es un simple método que ha mostrado buen desempeño en muchos dominios, es también muy eficiente para entrenar y para clasificar. Un clasificador bayesiano obtiene la probabilidad posterior de cada clase C_i, usando la regla de Bayes. El clasificador Naive Bayes (NBC) [9][10] hace la simple asunción que los atributos, A, son condicionalmente independientes entre ellos dada la clase (ver Fig. 3), así que la probabilidad puede ser obtenida por el producto de las probabilidades individuales condicionales de cada atributo dada la clase. La probabilidad posterior $P(C_i|A_1, ..., A_n)$ es dada por:

$$P(C_i|A_1, ..., A_n) = P(C_i)P(A_1|C)...P(A_n|C)/P(A) \quad (1)$$

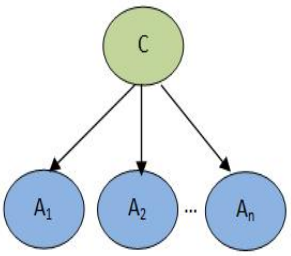

Fig. 3. Red Bayesiana

4 Experimentos y resultados

Para la clasificación de las clases primero se construyó el conjunto de entrenamiento, de 250 imágenes, se obtuvieron 394 regiones con 6 clases diferentes, edificio, césped, árbol, vaca, cielo y rostro. De cada una de las 394 regiones obtenidas de forma manual se obtuvo sus características mencionadas anteriormente. Todas estas características se extraen de una herramienta[6]. Una vez que tenemos el conjunto de entrenamiento podemos empezar a extraer una imagen y empezar la múltiple segmentación, primero se utilizó una segmentación en rejillas con longitud de 32 píxeles, el cual nos generaba 60 regiones de cada imagen, estas 60 regiones fueron etiquetadas y posteriormente se extraían sus características. Al principio se probo con 3 clases y una sola segmentación para ver resultados del clasificador, y no eran tan malos, logró reconocer más del 70% de las regiones, después se implemento las otras dos segmentaciones con grids de tamaño 20 y 16 pixeles para ver los resultados, en un principio se pensó que si la longitud de los segmentos se hacía más pequeño podría hacer una mejor clasificación de las clases ya que las características se pudieron pensar serian más refinadas, una vez segmentando con longitudes menores, no se notó una gran diferencia a la hora de la clasificación, mejoraba de un 70% a casi un 72% de la longitud de 32 píxeles a 16 píxeles, posteriormente después de agregar las otras 3 clases, se pudo notar que la clasificación bajo a un 54%(Ver Tabla 1), con segmentación de 32 píxeles de longitud, 27 características y 6 clases, después de esta etapa se empezó a pensar en la correcta selección de los atributos, se pudo notar que había algunos atributos que tenían el mismo valor para todas las clases, y se llegó a la conclusión de que esto era obvio, ya que como los segmentos son en forma de rejilla cuadrada, los valores tales como el área, ancho y alto serían de 1 una vez normalizados. Esta selección de rejilla también afecto a la media de x,y, desviación estándar x,y, límite del área, y convexidad. Este prob-

lema también se extendio a las segmentaciones de tamaño 20 y 16, en algunos atributos siempre nos generaba los mismos valores, aunque entre diferentes tamaño si habia una pequeña diferencia. Por ejemplo para una segmentación de tamaño de 32 píxeles nos generaba estos valores para los atributos de forma de las diferentes segmentaciones en todas las clases.

- Área: 1
- Ancho: 1
- Alto: 1
- Media x,y: 0.51562, 0.51562
- Desviación Estándar x,y: 1,1
- Límite del Área: 0.12109
- Convexidad: 0

Correctly Classified Instances: 148 54.0156%
Incorrectly Classified Instances: 126 45.9854%

Table 1. Matríz de confusión con todos los atributos que definen los segmentos

a	b	c	d	e	f	classifies ad
31	0	0	0	0	0	a=building
0	112	7	0	0	0	b=grass
12	20	2	0	0	0	c=tree
2	25	16	0	0	0	d=cow
12	10	2	0	3	0	e=sky
8	3	9	0	0	0	f=face

Sí el tamaño de la rejilla lo disminuíamos, entonces cambiaban los valores de media x y y, limite del área, etc. pero lo hacía para todas las clases, pasaba prácticamente lo mismo y el porcentaje de clasificación no mejoraba sustancialmente. Se decidió probar omitiendo los valores de los atributos que no hacían diferencia en la clasificación y se tomaron solo 18 atributos con 6 clases, y se logró hacer una mejora del 63%, aunque no se deseaba en un principio omitir estos atributos, ya que la clasificación de la imagen solo iba a depender del color(Tabla 2)

Correctly Classified Instances: 112 63.6364%
Incorrectly Classified Instances: 64 36.3636%

Se puede observar las gráficas de ambas pruebas, tomando en cuenta todos los valores de los atributos y omitiendo aquellos que definen la forma de los segmentos, se muestran el la (Fig. 4 y en la Fig. 5 respectivamente).

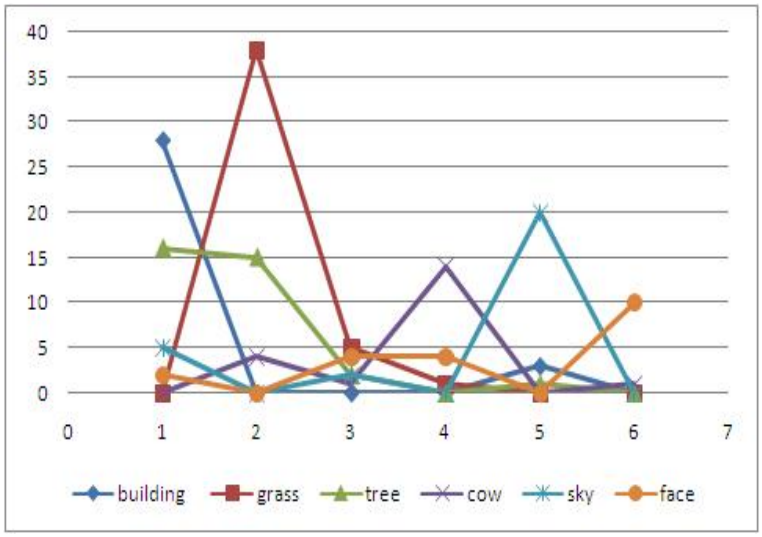

Fig. 4. Gráfica de Matríz Confusión tomando solo 18 atributos.

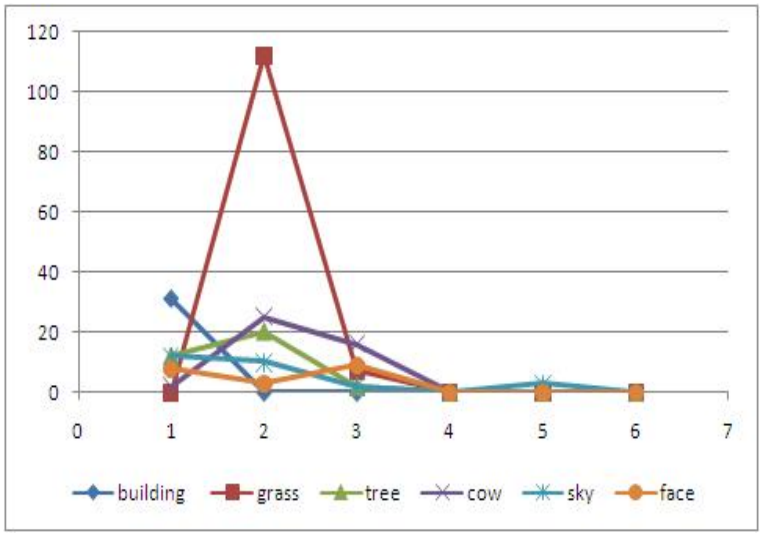

Fig. 5. Gráfica de Matríz Confusión tomando los 27 atributos.

Table 2. Matríz de confusión sin tomar atributos que definen la forma de los segmentos

a	b	c	d	e	f	classifies ad
28	0	0	0	3	0	a=building
0	38	5	1	0	0	b=grass
16	15	2	0	1	0	c=tree
0	4	1	14	0	1	d=cow
5	0	2	0	20	0	e=sky
2	0	4	4	0	10	f=face

Obteniendo estos resultados y haciendo más pruebas se llegó a la conclusión de que el conjunto de entrenamiento estaba teniendo combinación de atributos para las formas de los objetos, es decir que tenia objetos con formas rectangulares tales como edificio, césped, cielo, y por esa razón se comportaba mejor a la hora de clasificar, y en los objetos como la vaca sus atributos de forma no son siempre cuadrados por esa razón, se pudo notar que no detectaba mucho esta clase, porque a la hora de construir el conjunto de entrenamiento las características de forma variaban muy significativamente.

5 Conclusiones y trabajo futuro

Una vez que se observaron los resultados de la clasificación, se llegó a la conclusión de que no era necesario aplicar las demás segmentaciones de rejillas con diferente longitud ya que los atributos de la imagen que definen la forma no ayudarían mucho a la clasificación, se pudo notar que existen algunas clases que son fácilmente detectadas por algunos modelos de segmentación y algunas otras clases no son tan faciles con esos modelos, se propone usar el manejo de textura de las imágenes como atributos con este mismo modelo de segmentación(grids) y ver resultados de su clasificación, y también se podría empezar con el uso de algún algoritmo de segmentación más refinado, como Ncuts, Viola-Jones, Superpixel, etc. y hacer diferentes pruebas con los atributos con los que se cuentan y después implementarlo con otro tipos de atributos para imágenes como textura pero ya con estos algoritmos de segmentación más robustos.

References

1. S. Agarwal, A. Awan, and D. Roth. Learning to detect objects in images via a sparse, part-based representation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 26:14751490, 2004.

2. D. Hoiem, A. Efros, and M. Hebert. Geometric context from a single image. In Proc. Int. Conf. Comp. Vision, 2005.
3. A. Opelt, M. Fussenegger, A. Pinz, and P. Auer. Weak hypotheses and boosting for generic object detection and recognition. In Proceedings of the 8th European Conference on Computer Vision, volume II, pages 7184, 2004.
4. Bryan C. Russell, Alexei A. Efros, Josef Sivic, William T. Freeman, and Andrew Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In Proc. CVPR, 2006.
5. Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In ECCV, 2006.
6. Peter Carbonetto, Nando de Freitas, and Kobus Barnard, A Statistical Model for General Contextual Object Recognition, Computer Vision - ECCV 2004, Volume 3021/2004
7. Espacio de Color LAB http://en.wikipedia.org/wiki/Lab_color_space.html
8. Eduardo F. Morales y Jess Gonzlez, Aprendizaje Computacional, Aprendizaje Bayesiano http://ccc.inaoep.mx/ emorales/Cursos/NvoAprend/node63.html
9. Enrique Sucar, Modelos Gráficos Probabilistas, Metódos Básicos y Clasificadores http://ccc.inaoep.mx/ esucar/Clases-mgp/mgp.html
10. J Francisco Martnez Trinidad, J. Ariel Carrasco Ochoa, Reconocimiemto de Patrones, Clasificación Supervisada, http://ccc.inaoep.mx/ ariel/Bayes.pdf
11. Tomasz Malisiewicz and Alexei A. Efros, Improving Spatial Support for Objects via Multiple Segmentations, BMVC 2007
12. J. Shi and J. Malik. Normalized cuts and image segmentation. IEEE Trans. PAMI, 22(8):888905, August 2000.
13. Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. Textonboost: Joint ap- pearance, shape and context modeling for multi-class object recognition and segmentation. In ECCV, 2006.
14. D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. IEEE Trans. Patt. Anal. Mach. Intell., 24(5):603619, 2002.