Combinando diferentes tipos de información en la clasificación de imágenes

Juan I. Forcen, Miguel Pagola, Edurne Barrenchea, Mikel Galar

Departamento de Automática y Computación
Institute of Smart Cities
Universidad Pública de Navarra

Resumen Existen diferentes tipos de vectores de características que se utilizan en clasificación de imágenes los cuales tienen en cuenta diferente tipo de información de la imagen. En este trabajo proponemos un método para combinar la información de dos tipos diferentes de vectores de características a través de un método de descomposición uno contra uno de problemas multi-clase. Para evitar el problema de los clasificadores no competentes, utilizamos un método en el que la competencia de los clasificadores se calcula de forma relativa a la distancia de la instancia a clasificar con respecto a las clases. En este trabajo proponemos calcular la competencia de los clasificadores utilizando un vector características diferente del usado para el entrenamiento de los clasificadores, teniendo que tengan en cuenta la distribución espacial de la imagen. En los resultados experimentales comprobamos que este método mejora a las estrategias de descomposición normales y al método que calcula la competencia sin tener en cuenta la información espacial.

1. Introducción

La clasificación imágenes no es una tarea fácil debido a factores cómo la variabilidad, la iluminación, la orientación o la existencia de partes ocultas. Existen dos estrategias básicas en la literatura para afrontar este problema. En la primera se utilizan características de bajo nivel como los histogramas de color [1] o el histograma de gradientes orientados (*Histogram of Oriented Gradients*, HOG) [9] para construir un vector de características sobre el cual aplicar un algoritmo de aprendizaje. El vector de características que representa a la imagen se construye concatenando los histogramas de diferentes celdas de la imagen. De esta forma los vectores construidos almacenan información espacial de la imagen.

La segunda estrategia, que además es la más utilizada, es el método conocido como bolsa de palabras (*Bag of Words*, BoW). A partir de características locales (intensidades, HOG, histogramas de color) se definen unos descriptores, también conocidos como palabras visuales [7]. A través de estas palabras visuales se transforman las características locales en otras características llamadas de nivel medio. Para crear el vector de características final que representa a una imagen se agregan todos los vectores de nivel medio de la imagen dando lugar a la perdida de la información espacial. Es decir, en el vector de características

Actas de la XVII Conferencia de la Asociación Española para la Inteligencia Artificial, pp. 907-916

resultante aparecen los descriptores que se encuentran en la imagen, pero no en qué posición. Este hecho a veces es una ventaja ya que la clasificación es robusta al cambio de posición y orientación de los objetos de la imagen. Además se ha probado experimentalmente que estas características son más discriminatorias [7], pero también se pierde una información valiosa para encontrar la clase correcta.

Teniendo en cuenta todo esto se han propuesto varios métodos en los que se intenta utilizar ambas metodologías de tal forma que se tiene un mayor poder discriminante y se considere la información espacial. En [10] se combina la información dos tipos de características en un proceso iterativo de selección de instancias. Shotton et. al. [11] utilizan *boosting* en selección de características para incorporar la textura y la información espacial en el aprendizaje del clasificador. En [12] se propone un método en dos pasos, en el primero se realiza una clasificación en función de la información hiper-espectral y en un segundo paso utilizando la información espacial se refina la clasificación obtenida anteriormente.

Un factor importante a tener en cuenta en la clasificación de imágenes es que se trata de un problema multi-clase. En [5] presentamos un método, ponderación dinámica basada en distancias (DRVW-OVO), en el que se modifica la matriz de voto en una estrategia uno contra uno [4] en función de la competencia del clasificador. Para calcular la competencia de los clasificadores se utiliza la distancia de la instancia nueva a clasificar a cada una de las clases.

En este trabajo vamos a utilizar la técnica DRCW-OVO en la clasificación de imágenes como medio para combinar las características de BoW y aquellas con información espacial. Los clasificadores base los entrenaremos con características de BoW y para definir la competencia de los clasificadores vamos a utilizar la distancia relativa entre las imágenes utilizando características con información espacial. Es decir, primero representaremos las imágenes utilizando características de bajo nivel y encontraremos la distancia a los k vecinos más cercanos de cada clase. Después utilizaremos esta distancia para calcular la competencia de los clasificadores. De esta forma, vamos a incluir la información espacial de la imagen en la clasificación final.

Este trabajo se divide en las siguientes secciones: primero en los preliminares explicamos los conceptos básicos de extracción de características en imágenes y el método DCRW-OVO. En la sección 3 explicamos en detalle nuestra propuesta y en la siguiente sección presentamos los resultados experimentales obtenidos con dos bases de datos de imágenes. Finalizamos con las conclusiones y los trabajos futuros.

2. Preliminares

En esta sección describimos los conceptos básicos que utilizamos en nuestro trabajo y que son necesarios para el resto de la contribución. En primer lugar explicaremos las dos técnicas de extracción de características de imágenes: el histograma de gradientes orientados y el método *Bag of Words*. Después

explicaremos la metodología de descomposición DRCW-OVO para problemas multi-clase.

2.1. Histograma de gradientes orientados

La idea básica del histograma de gradientes orientados [9] es que la apariencia local de un objeto y su forma pueden ser caracterizados por la distribución de los gradientes locales de las intensidades de los píxeles o las direcciones de los bordes. Para obtener este histograma se divide la imagen en diferentes celdas, y para cada celda se crea un histograma de orientaciones que acumula las direcciones de los gradientes de todos los píxeles de esa celda. El histograma suele tener 8 o 9 bins [9]. Para evitar los problemas de diferente iluminación, se divide la imagen en diferentes bloques que contienen varias celdas cada bloque y se normaliza el contraste dentro de cada bloque. La concatenación de esos histogramas para todas la celdas de la imagen forma el vector de características de la imagen. Estos vectores de características se utilizan generalmente para entrenar un clasificador [9]. En la figura 1 vemos una representación del HOG, en cada celda están representadas las direcciones del histograma.

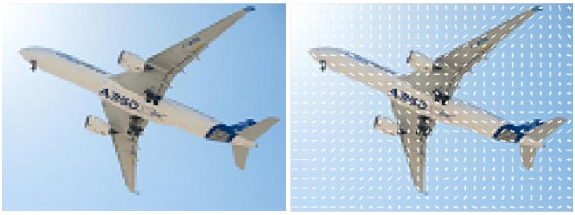

Figura 1. Imagen de un avión y la representación del HOG de cada celda sobreimpresionada.

2.2. *Bag of Words*

Un aspecto crítico en la clasificación de imágenes es encontrar características que sean descriptivas. El método bolsa de palabras, Bag of Words (BoW), trata las características de las imágenes como si fuesen palabras. En la clasificación de textos una bolsa de palabras es un vector con de las ocurrencias de las palabras, es decir, un histograma del vocabulario. En clasificación de imágenes el vector se construye con las ocurrencias de las palabras visuales de un diccionario de descriptores locales en la imagen. Los pasos en que se divide en método BoW son los siguientes:

Descriptores locales Comenzamos extrayendo ventanas de tamaño $w \times w$ de forma aleatoria del conjunto de imágenes de entrenamiento. Cada una de las ventanas tiene $d = 3$ canales (son imágenes RGB), por lo tanto está representado por un vector $\boldsymbol{x}$ de tamaño $N = w \times w \times d$. Se realizan dos pasos de preprocesamiento. En primer lugar se realiza una normalización de cada ventana restando la media y dividiendo por la desviación típica de sus elementos. Esto corresponde a una normalización del contraste local [2]. Después de normalizar cada vector, al dataset $\boldsymbol{X}$, formado por todas las imágenes, se le aplica un proceso de *whitening* [6]. Este proceso consiste en transformar linealmente los vectores $\boldsymbol{X}$ de tal forma que los componentes de los nuevos vectores no estén correlacionados. Una forma de realizar el proceso de *whitening* es a través de la descomposición en autovectores.

Aprendizaje del diccionario Utilizando un algoritmo de aprendizaje no supervisado se descubren un conjunto de descriptores que se llama diccionario. Normalmente se utiliza el algoritmo k-means, donde el número de clusteres encontrados es el número de elementos del diccionario y cada descriptor se conoce como palabra visual. En la figura 2 se muestran varias palabras visuales (descriptores locales) que hemos obtenido para la base de imágenes cifar-10 (que utilizaremos en los experimentos) utilizando los descriptores locales descritos previamente.

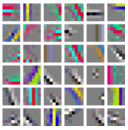

Figura 2. Palabras visuales obtenidos para la base de imágenes cifar-10.

Codificación El paso de codificación consiste en transformar los descriptores locales de una imagen en un nuevo vector descomponiendo dicho descriptor en el diccionario. Puede entenderse como una función de activación del diccionario, activando cada palabra visual de acuerdo con su parecido al descriptor. El vector de características, por lo tanto, tiene tantas componentes como palabras visuales del diccionario que se ha aprendido en el aprendizaje no supervisado. La forma más sencilla de codificar una ventana en un vector es asignando un 1 en la posición de la palabra visual más parecida y ceros en el resto. En [2] propusieron una forma más efectiva de codificación, en la que se asigna un valor según la distancia media a las palabras visuales:

$$\boldsymbol{\alpha}_{i,j} = \max\{0, \mu(x_i) - ||\boldsymbol{x}_i - \boldsymbol{d}_j||_2^2\} \quad (1)$$

donde $\mu(\boldsymbol{x}_i) = \frac{1}{K}\sum_{k=1}^{K} ||\boldsymbol{x}_i - \boldsymbol{d}_k||_2^2$ es la media de las distancias entre el vector de características locales y todas las palabras visuales del diccionario. Con este método cualquier característica cuya distancia es mayor que la media se codifica como 0.

A través de esta función de codificación, dada una ventana de una imagen de tamaño $w \times w$ podemos calcular su representación en características de nivel medio. De esta forma definimos la representación de la imagen completa aplicando esta función de codificación a todas las ventanas de la imagen. Como la codificación es costosa computacionalmente, en lugar de calcular el vector codificado para todas las ventanas centradas en todos los píxeles de la imagen, extraemos las características a lo largo de la imagen avanzando la ventana con un paso mayor a un pixel. Para un paso s se acaba construyendo una representación de tamaño $((n-w)/s+1) \times ((n-w)/s+1)$.

Pooling En la fase de *pooling* o agregación, todos los vectores codificados de toda la imagen (o de zonas de la imagen) se combinan para formar un único vector. Los operadores de agregación comúnmente utilizados son la suma y el máximo. En este caso utilizaremos una forma muy simple de pooling. Dividimos la imagen en 4 cuadrantes y sumamos todos los vectores dentro de cada cuadrante. Es decir los vectores que representan a cada imagen tienen $4K$ dimensiones (siendo K el tamaño del diccionario).

Clasificación El entrenamiento y la clasificación se lleva a cabo utilizando estos vectores de características por un clasificador, típicamente un SVM [3].

2.3. DRCW-OVO

Las estrategias de descomposición permiten afrontar problemas multi-clase mediante clasificadores binarios (ver [4] para una revisión completa). Entre ellas, una de las estrategias más utilizadas es la estrategia Uno-contra-Uno (*One-vs-One*, OVO). OVO divide un problema de m clases en $m(m-1)/2$ subproblemas binarios (todos los posibles pares de clases) que son afrontados por clasificadores base independientes. Una nueva instancia se clasifica obteniendo la salida de cada uno de los clasificadores. Cada clasificador que distingue un par de clases $\{C_i, C_j\}$ devuelve un grado de confianza $r_{ij} \in [0,1]$ en favor de la clase C_i ($r_{ji} = 1 - r_{ij}$). Estas salidas pueden almacenarse en lo que se denomina matriz de votos de la siguiente forma:

$$R = \begin{pmatrix} - & r_{12} & \cdots & r_{1m} \\ r_{21} & - & \cdots & r_{2m} \\ \vdots & & & \vdots \\ r_{m1} & r_{m2} & \cdots & - \end{pmatrix} \tag{2}$$

Para establecer la clase a la que pertenece la instancia puede usarse cualquiera de las agregaciones existentes en la literatura [4] sobre la matriz de votos. La

más simple es la estrategia del voto, donde cada clasificador da un voto para la clase que predice y la clase con más votos es con la que se etiqueta la instancia.

En OVO un clasificador se dice que es no competente cuando este intenta clasificar una instancia cuya clase real no es ninguna de las clases en las que el clasificador ha sido entrenado.

La participación de los clasificadores no competentes en la clasificación de nuevas instancias hace que puedan aparecer valores de salida que lleven al sistema final a una predicción incorrecta. El problema es que la competencia de los clasificadores no la podemos conocer a priori. Para estimar el valor de competencia de los clasificadores en [5] se utiliza la distancia relativa de una nueva instancia a cada una de las clases. La distancia a cada clase se calcula con los k vecinos más cercanos. Por lo tanto, se consideraban mayores pesos a las salidas de las clases que están cerca de la instancia a clasificar, ya que las salidas de los clasificadores correspondientes a las clases que están más cercanas de la instancia serán más competentes que aquellas que se encuentran lejos.

3. DRCW-OVO en clasificación de imágenes

En este trabajo proponemos utilizar el método DRCW-OVO para fusionar la información de dos tipos diferentes de características en la clasificación de imágenes. Es decir, vamos a clasificar la imagen en función de los vectores de características de BoW con un método de descomposición OVO y después calcular la competencia de los clasificadores según lo parecida que sea la imagen a clasificar con las imágenes más cercanas de cada clase. La distancia entre dos imágenes la calculamos utilizando el vector de características de HOG y un vector de pesos en los que damos más importancia al centro de la imagen (suponemos que el objeto está en esa posición):

$$d(I1, I2) = sw_i * |hog(I1(i)) - hog(I2(i))| \tag{3}$$

donde sw_i es el peso de cada celda i de la imagen.

En la figura 3, utilizando imágenes de la base cifar-10 [8], mostramos dos imágenes y las imágenes de cada clase (rana, ciervo, perro, gato, etc.) más parecidas en orden descendente. Podemos observar que entre entre las imágenes más parecidas siempre hay alguna de la clase a la que realmente pertenece la imagen. Por lo tanto, los mayores pesos en DCRW-OVO se corresponden con los clasificadores correctos.

Al procedimiento le llamaremos SDCRW-OVO y consiste en los siguientes pasos:

1. Entrenar los clasificadores base con las características de BoW, a partir de los cuales obtenemos la matriz de votos R de la imagen a ser clasificada.
2. Calcular los k vecinos más cercanos de cada clase para cada imagen a clasificar, utilizando la expresión 3) y calcular la distancia media de los k vecinos de cada clase almacenándolos en un vector $\boldsymbol{d} = (d_1, \ldots, d_m)$.

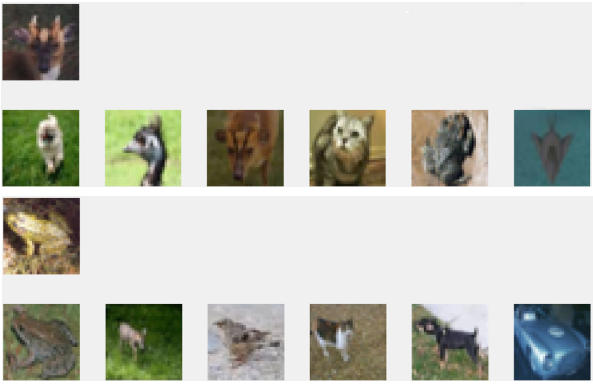

Figura 3. Arriba imagen de un ciervo y abajo imagen de una rana junto con las clases de imágenes más parecidas ordenadas de menor a mayor distancia.

3. Crear una nueva matriz de votos R^w en donde cada salida r_{ij} del clasificador que distingue entre las clases i y j está ponderado de la siguiente forma: $r^w_{ij} = r_{ij}w_{ij}$. Donde w_{ij} es la competencia del clasificador en esa instancia calculada utilizando las distancias d_i de la instancia a los vecinos más cercanos de cada clase i. En [5] se recomienda utilizar la siguiente expresión: $w_{ij} = \frac{d_i^2}{d_i^2+d_j^2}$
4. Usar la estrategia de voto ponderado en la matriz de votos modificada R^w para obtener la clase a la que pertenece la instancia.

4. Resultados experimentales

En esta sección comprobamos el comportamiento del método propuesto en dos bases de datos de imágenes. Cifar-10 es un conjunto de imágenes en color de tamaño 32×32 de 10 clases diferentes que contiene 50.000 imágenes de entrenamiento y 10.000 de test. Cifar-100 es similar a cifar-10 pero tiene 100 clases diferentes [8].

Para la extracción de las características HOG utilizamos los siguientes parametros: tamaño de bloque $= 8 \times 8$, tamaño de celda $= 4 \times 4$, número de bins $= 9$. Por lo tanto el vector de características tendrá 324 componentes. La matriz de pesos, que es una distribución gaussiana, para calcular la distancia entre imágenes se muestra en la Figura 4 (cada cuadricula corresponde con una celda de la imagen).

En la extracción de características BoW utilizamos un tamaño de ventana de 6×6, los procesos de nomalización y *whitening* son los descritos anteriormente.

0.0489	0.0551	0.0597	0.0621	0.0621	0.0597	0.0551	0.0489
0.0551	0.0621	0.0673	0.0701	0.0701	0.0673	0.0621	0.0551
0.0597	0.0673	0.0729	0.0759	0.0759	0.0729	0.0673	0.0597
0.0621	0.0701	0.0759	0.0790	0.0790	0.0759	0.0701	0.0621
0.0621	0.0701	0.0759	0.0790	0.0790	0.0759	0.0701	0.0621
0.0597	0.0673	0.0729	0.0759	0.0759	0.0729	0.0673	0.0597
0.0551	0.0621	0.0673	0.0701	0.0701	0.0673	0.0621	0.0551
0.0489	0.0551	0.0597	0.0621	0.0621	0.0597	0.0551	0.0489

Figura 4. Pesos para calcular la distancia entre imágenes.

Seleccionamos 400.000 ventanas al azar y utilizando el k-means para generar el diccionario de K palabras visuales. Para codificar utilizamos la expresión de la ecuación (1). Una vez que tenemos todos los vectores de entrenamiento se realiza un proceso de normalización.

Utilizamos el algoritmo L2-SVM con el kernel lineal como clasificador base. El L2-SVM utiliza la suma del cuadrado de las variables del termino de regularización en la función objetivo en lugar de la suma. De esta forma se hace al SVM menos susceptible a los outliers y mejora la generalización. El valor del término C de regularización se calcula utilizando validación cruzada en el conjunto de entrenamiento.

En el primer experimento utilizamos cifar-10 y un diccionario de 512 palabras visuales para calcular cual es la mejor expresión de competencia de los clasificadores y cual es el mejor número de vecinos cercanos a la hora calcular la distancia de una imagen a una clase. Probamos estas 5 expresiones diferentes: $w_{(1)} = \frac{dj^2}{dj^2+di^2}$, $w_{(2)} = \frac{dj^2}{di^2}$, $w_{(3)} = \frac{dj^3}{di^3}$, $w_{(4)} = \frac{dj^3}{di^{4.5}}$ y $w_{(5)} = \frac{dj^3}{di^6}$ en el cuadro 1.

	k=1	k=3	k=5	k=10	k=15	k=Todos
$w_{(1)}$	73.01	72.52	72.34	72.05	72.05	70.77
$w_{(2)}$	74.91	74.51	74.09	73.59	73.38	70.65
$w_{(3)}$	75.78	75.53	75.22	74.74	74.41	70.52
$w_{(4)}$	76.67	76.6	76.11	75.67	75.36	70.41
$w_{(5)}$	76.78	**77.25**	76.93	76.49	76.04	70.14

Cuadro 1. Porcentaje de acierto en el conjunto de test cifar-10 con diferentes combinaciones de número de vecinos y expresiones de competencia

El mejor porcentaje de imágenes de test acertadas corresponde a $k = 3$ vecinos más cercanos y la competencia calculada con $w_{(5)} = \frac{dj^3}{di^6}$. Además, podemos observar que los mejores resultados se obtienen siempre con esta expresión. Esto significa que en este problema se debe penalizar mucho a los clasificadores en los que la clase esté "lejos" de la imagen.

Utilizando el valor de k y la expresión de la competencia en el segundo experimento comparamos la eficacia del método propuesto SDRCW-OVO contra el método OVA, OVO y DRCVW-OVO estándar en el que se calculan la com-

petencia basado en la distancia de los vectores de características BoW (en cada caso se ha hecho validación cruzada de mejor k y expresión de distancia). Comparamos lo casos en los que el tamaño del diccionario es 64, 128, 255, 512, 1024 y 2048 (ver cuadro 2).

	64	128	256	512	1024	2048
OVA	62.54	67.36	71.43	74.26	76.51	78.67
OVO	61.67	65.47	68.61	70.83	72.13	74.02
DCRW-OVO	67.01	69.82	72.22	74.05	75.15	76.12
SDCRW-OVO	**71.6**	**74.02**	**75.59**	**77.25**	**78.28**	**79.08**

Cuadro 2. Porcentaje de aciertos para cifar-10

Atendiendo a los resultados obtenemos las siguientes conclusiones:

- En este problema OVO se ve perjudicado por los clasificadores no competentes. En todos los casos OVA es mejor que OVO.
- DCRW-VO corrige el problema de los clasificadores no competentes y mejora los resultados de OVO.
- La inclusión de la información espacial a través de nuestra propuesta es eficaz ya que SDRCW-OVO mejora en todos los casos a DRCW-OVO.
- Cuanto más discriminantes son las características de BoW nuestro método tiene un menor margen de mejora.

Remarcar que el mejor resultado de K-NN (K=10) con las características HOG es el 65.81 % y con las características de BoW es el 60.98 %.

El tercer experimento es similar al anterior utilizando cifar-100 (ver cuadro 3). En este caso los resultados son similares al caso anterior salvo que a partir de un tamaño de diccionario de 512 palabras visuales OVA es el mejor método. Esto es debido a que con 100 clases, en OVO existen muchos clasificadores no competentes y DRCW-OVO y SDRCW-OVO no consiguen calcular la competencia de los clasificadores correctamente. Esto también puede ser debido a que existen muchos menos ejemplos por clase y por lo tanto, la distancia a la clase no está correctamente calculada. Remarcar que el mejor resultado de K-NN (K=10) con las características HOG es el 28.7 % y con las características de BoW es el 32.13 %.

	64	128	256	512	1024	2048
OVA	35.42	41.3	45.8	47.09	50.06	47.67
OVO	27.8	30.11	31.50	32.75	32.83	
DCRW-OVO	39.00	41.18	42.72	42.70	43.32	43.04
SDCRW-OVO	43.68	45.02	45.94	46.70	47.30	

Cuadro 3. Porcentaje de aciertos para cifar-100

5. Conclusiones y trabajos futuros

En este trabajo hemos proponemos calcular la competencia de los clasificadores, en un ensemble OVO, utilizando información diferente a la que se utiliza para entrenar los clasificadores base en clasificación de imágenes. Probamos que en los conjuntos de test cifar-10 y cifar-100 mejoramos los resultados de OVO y de DCRW-OVO.

Como trabajo futuro queremos investigar otro tipo de características relacionadas con el color o texturas para calcular la competencia del clasificador.

Agradecimientos Parte de este trabajo ha sido financiado por el proyecto de investigación TIN2013-40765-P del Ministerio de Economía y Competitividad de España.

Referencias

1. Chapelle, O., Haffner, P., Vapnik, V.: SVMs for Histogram-Based Image Classification. IEEE Transactions on Neural Networks (10), 1055–1064 (1999)
2. Coates, A., Arbor, A., Ng, A.Y.: An Analysis of Single-Layer Networks in Unsupervised Feature Learning. Aistats 2011 pp. 215–223 (2011)
3. Cortes, C., Vapnik, V.: Support-Vector Networks. Machine Learning 20(3), 273–297 (1995)
4. Galar, M., Fernández, A., Barrenechea, E., Bustince, H., Herrera, F.: An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes. Pattern Recognition 44(8), 1761–1776 (2011)
5. Galar, M., Fernández, A., Barrenechea, E., Herrera, F.: DRCW-OVO: Distance-based relative competence weighting combination for One-vs-One strategy in multi-class problems. Pattern Recognition 48(1), 28–42 (2015)
6. Hyvärinen, A., Oja, E.: Independent component analysis: algorithms and applications. Neural networks 13(4-5), 411–430 (2000)
7. Koniusz, P., Yan, F., Mikolajczyk, K.: Comparison of mid-level feature coding approaches and pooling strategies in visual concept detection. Computer Vision and Image Understanding 117(5), 479–492 (2013)
8. Krizhevsky, A.: Learning Multiple Layers of Features from Tiny Images. Science Department, University of Toronto, Tech. pp. 1–60 (2009)
9. N. Dalal, B.T.: Histograms of Oriented Gradients for Human Detection. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2005)
10. Pasolli, E., Melgani, F., Tuia, D., Pacifici, F., Emery, W.: SVM Active Learning Approach for Image Classification Using Spatial Information. IEEE Transactions on Geoscience and Remote Sensing 52(4), 2217–2233 (2014)
11. Shotton, J., Winn, J., Rother, C., Criminisi, A.: TextonBoost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. International Journal of Computer Vision 81(1), 2–23 (2009)
12. Tarabalka, Y., Fauvel, M., Chanussot, J., Benediktsson, J.A.: SVM- and MRF-based method for accurate classification of hyperspectral images. IEEE Geoscience and Remote Sensing Letters 7(4), 736–740 (2010)