Image Classification Model Using Visual Bag of Semantic Words

Anonymous CVPR submission

Paper ID Mo Rokibul Islam (40060110)

Abstract

A graph cut based automatic segmentation algorithm is used to identify the segmented parts of an image by using SVM predictive models. Visual Bag of Words (BoW) is a widely used in image classification problem. However, due to its lower accuracy for semantic extraction and huge computational cost arises from large size of vocabulary lead us to find a new way to solve the image classification problem. Visual Bag of Semantic Words (BoSW) is our proposed model that will address those issues properly. The proposed model includes an automatic segmentation algorithm based on graph cuts to extract major semantic regions. A supervised learning approach for 4-class classification problem will be applied to label the segmented regions of target classes. Apart from SVM, other classifiers: Logistic Regression (LR), Decision Tree (DT), Random Forest (RDF), Naive Byes (NB) and Nearest Neighbor (KNN) algorithms also will be used to fit the relationship and be compared their performance.

This project is re-implementation of the articles [9].

1. Introduction

Image classification has become very important and interesting to the research community due its widely use in face detection, computer vision, medical image, self-driven car, and many other areas. Extracting information from image are the key challenge of the image classification problem. The widely used concept of semantic extraction description is the Visual Bag of Words (Bow) [11, 13]. The Bag of Words (BOW) is commonly utilized in Natural Language Process (NLP) application, the main concept is to count the number of each word appears in a document and make a frequency histogram of from it. Same concept is used in visual BoW, we make a frequency histogram of unique features instead of word which help find another similar images. First, we extract local key-point features form the images, second clustering the key-point then name them visual words (features), finally make frequency histogram of each image. We use this histogram to predict similar images.

Collecting features from the image is the key challenge in visual BoW. Various methods are available to collect features such as the spatial pyramid matching method (SPM) [12, 16], distribution of local features method [6], sparse coding methods [10, 18], high-dimensional coding methods [3], and visual salience and visual word similarity topological constraints [1, 2, 14].

However, features are commonly constructed by clustering of local key-points within images causes lower accuracy for semantic extraction. Moreover, number of the features become higher and introduce higher computation cost. Solution might be to reduce the semantic gap in the visual BoW model [4, 17] or reduce size of the features such as dimension reduction methods [5, 15], spectral clustering [7] and semantic spectral clustering methods [8].

This paper proposes a novel model based on a visual bag of semantic words (BoSW) to improve classification accuracy and efficiency [9]. The proposed model has three main features: (1) low-level visual features are extracted from regions within images, rather than local key-points within images; (2) visual word vectors are replaced by visual semantic words; (3) visual semantic word annotations are utilized to represent images. The classification process is based on the semantic word annotation.

The remainder of this report is organized as follows. The methodology of the proposed model, including a semantic auto-segmentation algorithm and visual semantic word annotation algorithm is described in Section 2. In Section 3, data source, model parameter, result, discussion, model performance. Finally, Section 4 discusses our conclusions and summery. Acknowledgment and all the references used in this project, including article and online materials, will be provided end of the report.

2. Methodology

The proposed method includes three main steps, as shown in Fig. 1. First, we design an auto-segmentation

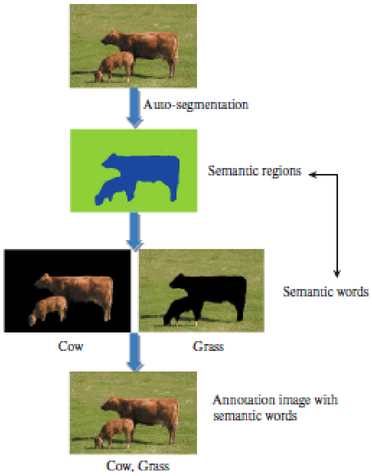

Figure 1. Image representation based on visual BoSW

algorithm based on graph cuts to extract semantic regions, each of which can be described by a semantic word. Second, we annotate the semantic regions with semantic words by using support vector machine. Finally, we utilize the annotated semantic words to classify images.

2.1. Semantic Region Segmentation

The segmentation algorithm utilizes graph cuts to segment semantic regions. We utilize main colors and the $\alpha-$expansion move algorithm to optimize the segmentation process.

2.1.1 Main colors

The colors which comprise larger proportions of an image are called Main colors and few main colors cover most of the pixel in an image. The low proportion colors can be merely considered as $'noise'$ since it does not contribute much towards the main content of the image. So, it necessary to filter out the noise to get main colors. To do that, we have created small color intervals (called as a bin) and put each pixel into its closet bin. In the RGB color space (max value = 255), we have 3 channels or dimension (R, G, B) and we make 3 interval [$[0-84]$, $[85-170]$, $[171-255]$] per channel. Thus, we have total 27 bins. A color histogram can then be obtained by calculating the number of pixels in each bin. Finally, the main colors can be obtained by adjusting a threshold parameter δ. The initial value of the δ is typically determined through trial and error.

2.1.2 Image segmentation based on graph cuts

Image segmentation based on graph cuts utilizes energy minimization to solve the pixel-labeling problem. The inputs are a set of pixels and a set of labels. The goal is to assign a joint label f , which can be obtained by minimizing an energy function $E(f)$.

$$E(f)=\sum\nolimits_{p\in P} D_p\left(f_p\right)+\sum\nolimits_{p,q\in N} V_{p,q}\left(f_p,f_q\right) \qquad (1)$$

where $D_p()$ is called $Data\ Cost$, $V_{pq}()$ is called $Smooth\ Cost$, N is a set of all pairs of neighboring pixels.

The energy function in our algorithm is composed of data cost and smooth cost, similar to Eq. (1).

$$E(f)=DataCost(f)+SmoothCost(f) \qquad (2)$$

After getting main colors, we initialized each pixel to its corresponding main color called pixel labeling. Note that, we have used another type of labeling which is used for supervised learning to label each observation. Just keep in mind these two labeling are not the same. In our graph cut model, the main color features are Label and used as terminal node features $L=(L_1, L_2, ...)$, where as the labeled pixels are the node set $N(n_1, n_2, ...)$.

A. Data Cost :

Data cost calculate the probability each pixel among the main colors. The higher possibility indicates that the node belongs to the relative class. In our approach, data cost can be described as negative log-likelihood of the following[28]: intensity model:

$$DataCost(i)=-\log\left(\frac{P_{xy}}{\sum P_{xy}}\right) \qquad (3)$$

where i is any node, P_{yx} is the probability of one node belonging to a certain class.

B. Smooth Cost :

Smooth cost measures the cost for neighboring nodes p and q. There are many ways to consider neighbor, for example $4-point$ stencil which consider left, right, top and bottom point, $8-point$ stencil consider all the 4 corners also. Here, we have considered first one 4-point stencil so if a node has position $p[i, j]$ then our four neighboring points would be $p[i+1, j], p[i-1, j], p[i, j+1], p[i, j-1]$. Suppose a node p belongs to label L_p, if a neighboring point belong to same label $i.e.$ L_p then we add cost 0 other wise 1, this called Potts model:

$$SooothCost(i)=\begin{cases}0 & \text{if } L_p == L_q\\ 1 & \text{if } L_p \neq L_q\end{cases} \qquad (4)$$

We revise the smooth cost, and use a smoothing parameter ε to adjust the smoothing efficiency. In our model, the values of ε can be adjusted from $\varepsilon \in [1-3]$.

$$SooothCost(i)=\begin{cases}0 & \text{if } L_p == L_q\\ \varepsilon & \text{if } L_p \neq L_q\end{cases} \qquad (5)$$

C. Fast α-expansion Move Algorithm :

The α-expansion move algorithm are used iteratively to optimize the energy function for multi-label segmentation. General trend is to relabel data point only once in each iteration that greatly limits the movement space which is very time consuming as many iterations are required to converge. In this paper, a fast α-expansion move algorithm to optimize the energy function for global minimization is used. First, we cluster the points according with main colors for example: set X_α for the label α, $\alpha \in$ *where color*. Second, if the point set X_p labeled p are closer to the point set X_α, then the label p will change to α.

In order to guarantee that the new labeling is valid, we optimize the α-expansion iteration ensuring the labeling process with maximum reduction energy. The reduced energy R_{xp} is the energy reduced in the α-expansion process. To get the maximum of R_{xp} is equivalent to get the minimum of $E(f)$ in Eq. (2) for each iteration of α-expansion.

$$R_{xp} = \sum_{x_i \in x_p} \left(DataCost\left(x_i, x_p\right) - DataCost\left(x_i, x_\alpha\right)\right) \quad (6)$$

Because the number of main colors is less, the iteration times of α-expansion is not too much. Then the time consuming of segmentation will be reduced substantially.

2.2. Annotation via Visual Semantic Words

Once segmentation is done, we calculate the color moment, mean, and variance and skewness of the three-primary colors as input features where as the positive and negative semantic region are considered as label of the image. We use support vector machine (SVM) with a Gaussian radial basis function kernel. The hole process is, we input a image then we get segmented regions of image. Finally, each segmentation predict a class.

3. Result and Discussion

In this section, we talk about data source, the results from the different models and results will be compared among models using four kind of metric score. For our project, we use variety of technology: python as a base programming language, sci-kit learn as a machine learning library, matplotlib, seaborn for visualization, Scipy as statistic library and for the data-frame Numpy and Pandas are used.

In this project, we have selected to train and evaluate 6 classifiers: *Support Vector Machine (SVM)*, *Logistic Regression (RN)*, *Random Forest (RDF)*, *Decision Tree (DT)*, *K-Nearest Neighbors (KNN)*, and *Gaussian Naive Bayes (NB)*.

3.1. Data Source and Processing

In this project, the row data are mainly images that has been collected from randomly from internet (Google image). There are four classes of data: Cow, Car, $House$ and $Tree$ are used to train models. The number of the data of each class are $[Cow, Car, House, Tree] = [76, 57, 56, 42]$.

After completing segmentation regions, we calculate the moment, mean, and variance and skewness of the three-primary colors of each segmentation regions. Then those statistical measures are labeled according to segmentation regions which are finally used as input data sets to train and test our models. As the values of moment is 0, so it have been discarded from final data set. In the case of libeling, due to limitation of data classes we have only labeled the segmented region which corresponding to main object and rest of regions were not included to train the model. Actually, it becomes a single-label problem instead of multi-label problem. For example, for both Cow and $Tree$ class, we have only labeled first segmented region which means only Cow and $Tree$ to train models and the segmented region of $Grass$ or Sky were not labeled and utilized. Thus, once we test similar images, it will only be classified as Cow class instead of $(Cow, Grass)$ and $Tree$ class instead of $(Tree, Sky)$ class.

3.2. Models Parameters

Final data set have been split in to 70% to train the models and 30% to evaluate model performance. We have search best parameter for individual model and k-fold cross validation process is used to get it. The hyper-parameters of models have been calculated using k-fold cross validation process, where k is selected as 5. Table 1 show the list of hyper-parameter.

The threshold parameter δ is determined through trial and error and finally set to $\delta = 0.5$ which means any color bin that comprise atleast 5% of a image will be considered as main color and less than 5% will be considered as noise and removed. The smoothing parameter is set to $\varepsilon = 3$.

3.2.1 Model Performance

Fig. (2) and (3) describe the performance of the different model for training and test data sets respectively.

The training accuracy of different model, in Fig. (2), we clearly observe *RDF* provides higher accuracy is 100% and followed by *DT* scored over 95%. *SVM* secured modest score which are around 80%. On the other hand, *KNN* and *LN* provided average performance around 65% and *NB* gave poor accuracy which is close to 50% means under-fitting.

However, form our test accuracy, in Fig.(3), provides very

Model	Hyper-parameters
SVM	C = [1,10,100, 1000], γ = [0.01,0.1,1,10], 'kernel' =['linear', 'poly', 'rbf']
LN	'solver': ['lbfgs', 'sag', 'newton-cg']

RDF	max_depth = [2,4,..,10],
	n_estimators=[50,60,..,100]

DT	max_depth = [2,4,..,10]
KNN	n_neighbor = [7, 14, 21]

Table 1. Hyper parameter of individual Classifier model

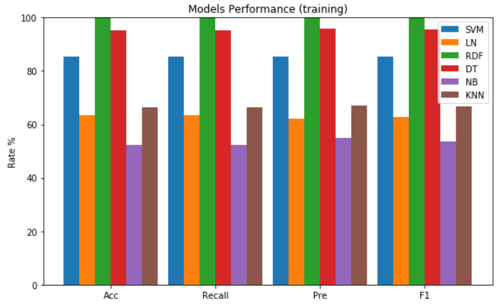

Figure 2. Model Performance with training data set

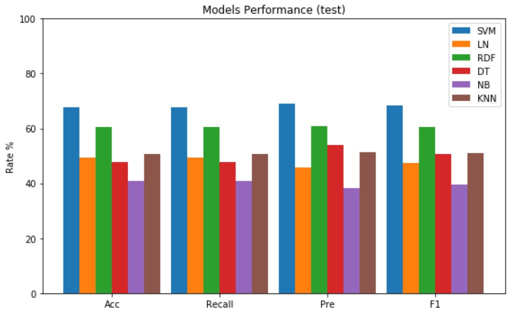

Figure 3. Model Performance with test data set

interesting results. *SVM* scored highest accuracy which is around 70%. On the other hand *RDF* and *DT* prediction score below 60% and 50% respectively, who has very good score in training. Thus this suggest that two models are having over-fitting problem. *RDF* widely used classifier but may cause over-fitting due to small data set or data imbalance. But our data set is very balanced so data size might be issue. *KNN* and *LN*have very similar score around 50%. *NB* has the lowest accuracy which is 40%.

It is clearly shows that our result has similarity with that of this article although we had different data sets and classes.

4. Conclusion and Summary

In this project, we started our journey to correctly predict the contents of an image. To do that we re-implement a graph cut based automatic segmentation algorithm to obtain semantic regions. We then take statistical measures of each region as input of SVM algorithm to label semantic regions with visual semantic words. The visual BoSW was used to represent images, rather than a standard visual BoW. The proposed model use small numbers of color as a main color by filtering low-portion color of the image. Then utilize energy minimization to solve the pixel-labeling problem. A Fast α-expansion move technique has been introduced to minimize the energy equation which is faster and time consuming as it only deals with small set of main colors. We have also compared the model performance with others model. We have found that the proposed model improves the accuracy of classification and reduces computational cost.

The main challenge was implementation. Because no code are available, I had to start from scratch to write code.

5. GitHub Repository

All the codes, data sets and details can be found in the following link: *https://github.com/rokibMcGill/-Image-Classification-Model-Using-Visual-Bag-of-Semantic-Words/tree/main*

6. Acknowledgment

Thanks to Professor for this wonderful course. I enjoyed and learned a lot from this course specially mathematical background behind of different models. The lecture materials and videos was precises and well organized that made understandable to us very easily. TA's also deserve another thanks for answering our question promptly and efficiently.

References

[1] M.M. Cheng, N. J. Mitra, X. Huang, and et al. Global contrast based salient region detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(3):569–582, 2015. 1

[2] X. Guo and X. Cao. Good match exploration using triangle constraint. *Pattern Recogn. Lett.*, 33(7):872–881, 2012. 1

[3] H. Jégou, F. Perronnin, M. Douze, and et al. Aggregating local image descriptors into compact codes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(9):1704–1716, 2012. 1

[4] R. Khan, C. Barat, D. Muselet, and C. Ducottet. Spatial histograms of soft pairwise similar patches to im-

prove the bag-of-visual-words model. *Comput. Vision Image Understand.*, 132:102–112, 2015. 1

[5] S. Lafon and A. B. Lee. Diffusion maps and coarsegraining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(9):1393–1403, 2006. 1

[6] L. Liu, L. Wang, and X. Liu. In defense of soft-assignment coding. In *2011 IEEE Int. Conf. on Computer Vision (ICCV 2011)*, page 2486–2493, Barcelona, Spain, 2011, 2011. 1

[7] Z. Lu and Y. Peng. Exhaustive and efficient constraint propagation: A graph-based learning approach and its applications. *Int. J. Comput. Vision*, 103(3):306–325, 2013. 1

[8] Z. Lu, L. Wang, and J. R. Wen. Image classification by visual bag-of-words refinement and reduction. *Neurocomput*, 173 (Part 2):373–384, 2016. 1

[9] Yali Qia, Guoshan Zhanga, and Yeli Lib. Image classification model using visual bag of semantic words. *Pattern Recognition and Image Analysis*, 29(3):404–414, 2019. 1

[10] J. Shi, Y. Li, J. Zhu, and et al. Joint sparse coding based spatial pyramid matching for classification of color medical image. *Comput. Med. Imaging Graphics*, 41:61–66, 2015. 1

[11] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proceedings Ninth IEEE International Conference on Computer Vision*, page 1470–1477, Nice, France, 2003. 1

[12] K. Vyas, Y. Vora, and R. Vastani. Using bag of visual words and spatial pyramid matching for object classification along with applications for ris. *Procedia Comput. Sci.*, 89:457–464, 2016. 1

[13] C. Wang and K. Huang. How to use bag-of-words model better for image classification. *Image Vision Comput.*, 38:65–74, 2015. 1

[14] R. Wang, K. Ding, J. Yang, and L. Xue. A novel method for image classification based on bag of visual words. *J. Visual Commun. Image Represent*, 40 (Part A):24–33, 2016. 1

[15] S. Yan, D. Xu, B. Zhang, and et al. Graph embedding and extensions: A general framework for dimensionality reduction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(1):40–51, 2007. 1

[16] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *2009 IEEE Conf. on Computer Vision and Pattern Recognition*, page 1794–1801, Miami, FL,USA, 2009. 1

[17] C. Zhang, R. Li, Q. Huang, and Q. Tian. Hierarchical deep semantic representation for visual categorization. *Comput. Vision Image Understand.*, 257:88–96, 2017. 1

[18] C. Zhang, J. Liu, C. Liang, and et al. Image classification by non-negative sparse coding, correlation constrained low-rank and sparse decomposition. *Comput. Vision Image Understand.*, 123:14–22, 2014. 1