Finding Clusters Within a Class to Improve Classification Accuracy

Yong Jae Lee

The University of Texas at Austin

EE381K Multidimensional Digital Signal Processing

Final Project Report

May 9th, 2008

Abstract

In machine learning, classification is defined as the task of taking an instance of the dataset and assigning it to a particular class. Classifiers are constructed using the training set, such that a novel instance is labeled with the "correct" class. Usually, there is the underlying assumption that instances within a class are similar and instances across classes are dissimilar. For example, given a dataset of images of cars and bicycles, the shapes and appearances of the two classes are different. While this is a valid assumption, in reality, there may also be differences within a particular class, although it may be less pronounced. For example, the car class may be composed of side-view, front-view, and rear-view images of cars. Therefore, variations can be found within each class and homogeneous groups can be formed for each variation. Then, each group can be considered to be a "sub-class" for training a classifier that can focus on the specific aspect of the class.

I. Introduction

Object recognition is a fundamental problem in computer vision that involves the tasks of detecting, categorizing, and identifying objects in images and videos. The goal is to allow a machine to understand and interpret an image or video the way humans do. There are many applications in various fields which can benefit from a successful object recognition system. For example, most of the returned images from search engines are irrelevant to the query term. This problem could be alleviated if the search were performed based on the content of the image rather than the content of the query text. In medical imaging, automatically identifying any irregularities or symptoms would increase early detection of diseases. Vehicle accidents could entirely be eliminated with a vehicle that controls its speed based on road and weather conditions, proximity to other vehicles, etc.

The problem is challenging because of the variability in position, scale, pose, and shape of the object of interest, imaging and lighting conditions, occlusions, and extreme clutter where the object occupies a much smaller portion in the image than the background.

The standard procedure of measuring success of an object recognition system is to test the algorithm on benchmark datasets. A typical dataset contains a handful to hundreds of object categories. The underlying assumption is that images within the same object category are similar and that images from different object categories are dissimilar, with varying degrees depending on the category. While this is a valid assumption, it may be reasonable and even favorable to divide a category further into sub-categories. Each sub-category, or sub-class, can then be used to train the system, thereby focusing on specifics of the class that would otherwise be ignored with a globally trained system using the original class.

An object recognition system that utilizes sub-classes can be divided into four modules: image representation, pairwise image similarity computation, clustering, and classification. The difficulty of the implementation of the system depends on which algorithms are used for each module. Since each sub-class can be treated as a class, the system is a natural extension of a standard global object category learning algorithm. Therefore, the overall implementation is relatively simple.

II. Related Work

The "bag-of-features" approach to visual recognition has been quite successful. The basic idea is to represent an image as a set of local features collected from salient regions. Patches are detected on salient points, e.g., corners, and a visual descriptor vector is evaluated for each patch. Thus, the image can be represented as a distribution of these features. Some examples of salient regions are corner-like regions [1] and "blob-like" regions [2]. The Scale Invariant Feature Transform [3] (SIFT) is another method for detecting and describing salient regions, where its main contribution is the descriptor that describes salient regions based on the magnitude and direction of gradients.

To define a similarity measure between images, the distribution of local features can be compared. There are several options for computing similarity between images, including similarity in appearance and similarity in spatial layout. The Proximity Distribution Kernel [4] (PDK) is a method that measures similarity between images based on the spatial layout of the local appearance features in the images.

Clustering is the partitioning of a data into subsets, such that the instances in each subset have proximity in terms of some defined distance measure. There are several methods for clustering, including k-means [5], agglomerative [6], and spectral clustering [7]. The Normalized Cuts [8] method is a type of spectral clustering algorithm that has been applied to image segmentation and data clustering.

Classification is a tool necessary for labeling novel instances. There are many possible classifiers, such as the Nearest Neighbor Classifier [9], Neural Networks [10], and Support Vector Machines [11] (SVM). SVMs have been shown to produce very good results in the fields of text categorization and image categorization, among many others.

III. Approach and Implementation

Each image in the dataset is represented as a set of local features. This representation allows for recognition systems to be robust to clutter, occlusion, and common transformations of objects such as rotation, translation, and scale. I use the Harris detector to detect corner-like regions [1] and the Maximally Stable Extremal Region [2] (MSER) detector to detect "blob-like" regions.

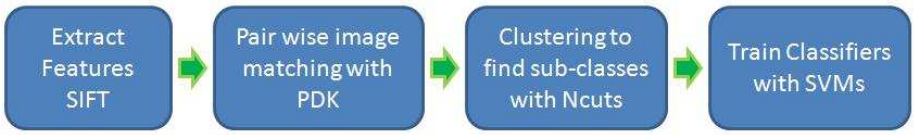

Fig. 1. Block diagram of the system model. First, each image is represented as a set of features using SIFT [3]. Then, pair wise image similarity values are computed using the PDK [4], and sub-classes are formed within each object category with Normalized Cuts [8]. Finally, SVM [11] classifiers are trained on each sub-class to classify novel images.

The Harris detector finds corner-like regions by shifting a window through the image and determining regions that have significant change in contrast in all directions. The MSER detector finds regions in the image that have high or low intensity than all surrounding pixels outside their boundaries. On each of the detected regions, the SIFT [3] descriptor is computed. A set of 8-bin orientation histograms are computed on 4 x 4 arrays (each array composed of 16 pixels) for each detected region. The gradient magnitude of each pixel is weighted by a Gaussian, and added to a bin in the corresponding region histogram. This produces a 128 dimensional feature vector (4 x 4 x 8), that is invariant to illumination conditions and minor viewpoint changes. Experiments in [3] show that high accuracy can be achieved for feature matching in different images, and have been shown to outperform other local descriptors on many types of images. For my experiments, on average, each image was represented by about 1000 features.

The next step is to compute a similarity measure between all images in all classes, in order to train discriminative classifiers that can distinguish one class from another. The PDK [4] is a method that measures similarity between images based on the spatial agreement of their local appearance features. First, the set of features from all images are vector quantized into V codewords which constitute the codebook of the dataset. Each feature in an image is replaced with the nearest codeword in the codebook, and is associated with its x and y image coordinates. Then, for each image, a proximity distribution is measured: for each codeword pair v_i and v_j in the image, a distribution $H_r(i,j)$ of the r-spatially nearest codewords of type j to codewords of type i is stored in a 1-D vector of length r. The collection of these 1-D vectors for every combination of word pairs (all possible i and j pairs) produces the proximity distribution H_r. Finally, the proximity distributions between all images are compared to produce the PDK matrix,

where the PDK between image I^1 and image I^2 is defined as:

$$PDK(I^1, I^2) = \sum_{i,j=1}^{V} \sum_{r=1}^{R} \min(H_r^1(i,j), H_r^2(i,j)) \quad (1)$$

The PDK is invariant to scale, rotation, and translation of the object, since the relative spatial positions between the features in the image are considered, rather than their absolute distances to one another. On many object category datasets, the PDK has been shown to outperform methods that take only appearance into account [4]. For my implementation, I constructed a 200-word codebook (V = 200), and computed the distribution of 64 spatial neighbors (R = 64) per feature.

Once a measure of similarity is computed between all images, this information can be used to find sub-classes within each object category. Graph-theoretic clustering methods can be applied if each image is viewed as a node in a graph and the PDK values are viewed as non-directive weighted edges between each pair of images. The Normalized Cuts [8] spectral clustering algorithm partitions the graph such that the edges between different groups have low weights while edges within a group have high weights. Each object category is clustered this way.

Specifically, the clustering problem is formulated as maximizing the objective function, $\mathbf{w}_n^T A \mathbf{w}_n$, where A is the affinity matrix and $\mathbf{w}_n$ is the vector of weights linking the data points to the n^{th} cluster, subject to $\mathbf{w}_n^T \mathbf{w}_n = 1$. This can be formulated as an eigenvalue problem: $A\mathbf{w}_n = \lambda \mathbf{w}_n$. To find k clusters, the eigenvectors associated with the k largest eigenvalues are computed. The data points are clustered by thresholding the components of the eigenvectors, where the threshold is iteratively refined to produce the best "cut". The problem is adjusted to not only find the minimum cut, but also maximize the inter-cluster difference (hence its name, the "normalized" cut). While, the exact solution is NP-hard, an approximate solution can be found.

Finally, SVMs [11] are used for classification. An SVM classifier finds the optimal hyperplane in feature space that achieves maximum separation between two classes of points. This is done by finding the unique optimal boundary between the the nearest points on either side of the hyperplane (called support vectors). Novel test instances are classified to be positive or negative, depending on which side of the hyperplane they fall on. The objective function for finding the optimal hyperplane assumes that the data is linearly separable. If the data is not linearly separable,

the data can be mapped to a higher dimensional space using the "kernel trick". A kernel function is a function that is equivalent to an inner product in feature space which implicitly maps data to a high-dimensional space. As long as the set of points can be written as an inner product, it can be mapped to a space where it can be separated. This is proven by Mercer's Theorem [12], and is shown that every semi-positive definite symmetric function is a kernel function. The PDK satisfies this condition.

The SVM can be extended to multi-class classification by constructing a binary classifier for each class, where the instances in the class of interest are treated as the positive class, and the instances in all other classes are treated as the negative class. The test instance is labeled with the class associated with the classifier that produces the highest confidence. This is called 1-vs-all SVM classification, and I used this method for my experiments.

IV. Discussion

Other known object recognition systems do not explicitly find groups within a class to train classifiers. Some approaches to capture local information within the data have been proposed. Bottou and Vapnik [13] suggested constructing classifiers using only the training data that is near a given test instance. Zhang et al. [14] constructed a hybrid nearest neighbor and SVM classifier which showed improved results over a global SVM classifier. However, both methods are sensitive to the position of the test instance in feature space and hence training must be performed separately for each test instance. In contrast, the proposed method is able to capture local structure of the data independent to the test instances, and thus training can be done offline.

V. Results

The algorithm is evaluated on an image dataset commonly used to test object recognition algorithms: the PASCAL Visual Object Classes Challenge 2005 dataset [15].

The PASCAL dataset consists of 4 object categories: motorbikes, bicycles, people, and cars. There are 684 training images (214 motorbikes, 114 bicycles, 84 people, 272 cars) and 689 testing images (216 motorbikes, 114 bicycles, 84 people, 275 cars). The train and test sets are disjoint. The objects in the images have significant variation in appearance due to changes in

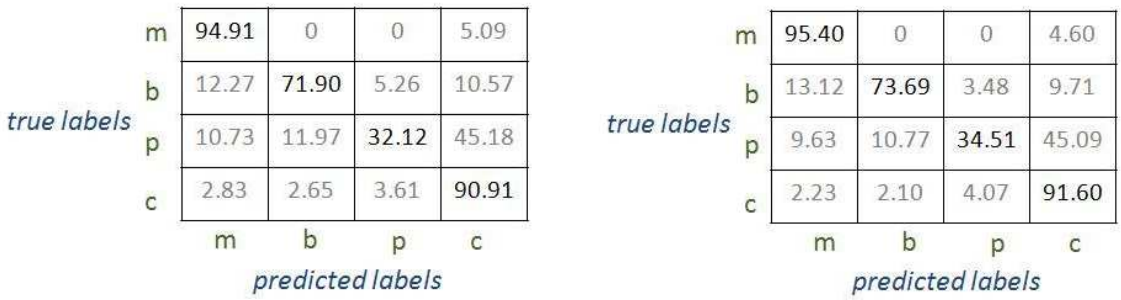

Fig. 2. Confusion matrices showing classification accuracies for the baseline method (a) and the proposed method (b) on the PASCAL dataset [15]. The number of sub-classes for the proposed method was set to 3 per object category. m: motorbikes, b: bicycles, p: people, c: cars.

viewpoint, scale, and rotation. Furthermore, the amount of clutter present in each of the images is large, making the dataset quite difficult. The large variation in the objects also means that homogeneous groups can be formed within each category that capture certain aspects of the class.

For the proposed method, classifiers were trained for each sub-class found from each class. In order to choose the number of sub-classes, I sampled the training set to form a validation set (50 images per class) in which I trained and tested SVM classifiers with possible sub-classes in the range $[2, 5]$. Three sub-classes per class gave the best results. When a test image was classified as belonging to a sub-class, it was labeled with the object category that the sub-class belonged to. I compared the proposed method with the baseline method (where classifiers are trained using all images in a object category and no sub-classes are formed), by computing the mean classification accuracy for each category. Confusion matrices showing these results are shown in Figure 2.

Overall, there is improvement over the baseline method, especially for the bicycle and people classes. These two classes exhibit more variation than the other two classes, and thus the sub-classes capture more meaningful structures within each class. The relatively high overall accuracies for the two methods (around 80%) shows that the combination of feature representation by SIFT, image matching by PDK, and classification with SVMs performed well. The people class produced the lowest accuracy. This is probably due to it having the lowest number of training

TABLE I
OVERALL MEAN ACCURACIES FOR THE THREE CLASSIFICATION METHODS ON THE PASCAL DATASET [15]. THE NUMBER OF SUB-CLASSES FOR THE PROPOSED METHOD WAS SET TO 3 PER OBJECT CATEGORY.

	Accuracy (%)
baseline method	81.86
proposed method	82.87
K-NN (K=1)	53.70

images, and also due to it having the most clutter in its images.

I also compared the proposed method to the K-nearest neighbor [9] (K-NN) classifier, which labels a novel instance by taking the mode of the labels of the K nearest training instances in feature space to the novel instance. It is a commonly used classifier, due to its simplicity and relatively good performance. K is a parameter that needs to be chosen by the user. In order to choose K, the K-NN classifier was trained and tested on a validation set (the same procedure of choosing the number of sub-classes) with values in the range $[1, 10]$. The highest accuracy was achieved when K=1. Overall mean accuracies of the proposed, baseline, and K-NN methods are shown in Table I.

The proposed and baseline methods both outperformed the K-NN method significantly. This shows that on this dataset, the SVM classifier is significantly stronger than the K-NN classifier, presumably due to its better generalization properties. The K-NN classifier only looks at the neighbors within the vicinity of each test instance, and hence can suffer when there are train or test instances that are outliers. It also must "guess" the label of a test instance when the mode label among its K nearest neighbors is not unique. I chose the label randomly among the top labels in such situations.

VI. CONCLUSION

Object recognition is a challenging problem, but in recent years many techniques have achieved some success. There are many components to the problem, including image representation, learning, clustering, and classification. While a dataset representing an object category should be homogeneous in the sense that each image should contain the object of interest, there is

usually enough variation within each object category to form homogeneous sub-categories. These sub-categories can then be treated as sub-classes to train classifiers with the ultimate goal of increasing performance compared to a global classifier which trains on the original class images. I have shown in this project that such a system is able to make improvements over a global classifier, by performing experiments on a standard image dataset.

REFERENCES

[1] K. Mikolajczyk and C. Schmid, "Scale & Affine Invariant Interest Point Detectors," *International Journal of Computer Vision*, vol. 60, no. 1, pp. 63–86, 2004.

[2] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide baseline stereo from maximally stable extremal regions," in *British Machine Vision Conference*, 2002.

[3] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[4] H. Ling and S. Soatto, "Proximity Distribution Kernels for Geometric Context in Category Recognition," *Proc. IEEE International Conference on Computer Vision*, pp. 1–8, 2007.

[5] J. MacQueen, "Some methods for classification and analysis of multivariate observations," *Proceedings Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, no. 281-297, p. 14, 1967.

[6] N. Jardine and R. Sibson, *Mathematical Taxonomy*, 1971.

[7] A. Ng, M. Jordan, and Y. Weiss, "On Spectral Clustering: Analysis and an algorithm," *Advances in Neural Information Processing Systems 14: Proceedings of the 2002 Conference*, 2002.

[8] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[9] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[10] C. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, USA, 1995.

[11] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[12] J. Mercer, "Functions of Positive and Negative Type, and their Connection with the Theory of Integral Equations," *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, vol. 209, pp. 415–446, 1909.

[13] L. Bottou and V. Vapnik, "Local learning algorithms," *Neural Computation*, vol. 4, no. 6, pp. 888–900, 1992.

[14] H. Zhang, A. Berg, M. Maire, and J. Malik, "SVM-KNN: Discriminative Nearest Neighbor Classification for Visual Category Recognition," *Proc. IEEE Computer Vision and Pattern Recognition*, pp. 2126–2136, 2006.

[15] M. Everingham, A. Zisserman, C. K. I. Williams, L. Van Gool, and et al., "The 2005 PASCAL Visual Object Classes Challenge," *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Textual Entailment.*, pp. 117–176, 2006.