RECOGNITION USING RAPID CLASSIFICATION TREE

Keith Haynes[†], *Xiuwen Liu*[‡], *Washington Mio*[*]

[†]Dept. of Electrical Eng. and Computer Sc. United States Military Academy West Point, NY 10996

[‡]Dept. of Computer Science Florida State University Tallahassee, FL 32306

[*]Dept. of Mathematics Florida State University Tallahassee, FL 32306

ABSTRACT

This paper proposes a method to achieve object classification with high throughput and accuracy using a rapid classifcation tree. To achieve this, we decouple the training and test stages. During the training stage, we learn optimal discriminatory features from the training set and then train a classifier with high accuracy. Then we create a classifcation tree, where each node uses a lookup table to store the solutions, resulting high throughput at the test stage. To make the lookup tables feasible for applications, we learn a projection matrix through stochastic optimization. We illustrate the effectiveness of the proposed method using several datasets; our results show the proposed method achieves often several orders of magnitudes of improvement in throughput while maintaining a similar accuracy.

Index Terms— Object recognition, object classification, feature extraction, image analysis, face recognition

1. INTRODUCTION

With advances in sensoring and communication technologies, many sensors are becoming increasing afforable and as a result, and many automated solutions become a competitive alternative in many applications. To realize these solutions, a common problem is pattern recognition. As data are being generated at an ever increasing speed, recognition systems with high throughput become important. In addition, time-sensitive applications require real-time or near real-time processing, where fast and accurate classifiers are critical.

Despite the emphasis on pattern recognition and the considerable progress that has been made, rapid object recognition receives little attention as accuracy has been the focus of most pattern recognition research. Commonly used classifiers include Bayesian inference, neural networks, and k-nearest neighbor (KNN). Although these classifiers can be effective, a direct use often can not meet the throughput requirements of applications. Bayesian inference classifiers make classification decisions based on posterior probabilities, which are derived from likelihood and prior probabilities; in applications, these probabilities are estimated using either parametric or non-parametric techniques. Note that Bayesian inference classifiers require an optimization and thus limit their throughput. Neural networks are also widely used to avoid estimation of probability models. In theory, a neural network can be trained to solve any computable problem. As the problems become more complicated, so does the structure of the network. With the increase in complexity, there is an increase in the computational cost of classification. The KNN is a simple, yet highly effective classifier which benefits from its ability to handle complicated non-linear decision boundaries. In many cases, KNN tends to increase in accuracy as the training set population increases; when the training set is unlimited, the misclassification rate is no more than twice of that of the optimal classifier[1]. Unfortunately, a large training set also results in a loss of computational performance.

This paper proposes a framework for rapid object recognition. The core is a new classifier called rapid classification tree (RCT). In this framework, first a set of discriminant features are learned from training examples and then the classification problem is decomposed into smaller problems, where a lookup table can be implemented by learning a dimension reduction matrix an a KNN classifier (or any other classifier). For applications we have tested, this classifier gives accuracy on par with other classifiers, but because of its computational efficiency, its throughput will far exceed that of the others. The rest of the paper is organized as follows: Section 2 describes the rapid classification tree and experimental results are given in Section 3; Section 4 concludes the paper with a brief discussion.

2. RAPID CLASSIFICATION TREE

Implicitly we adopt an appearance-based approach to pattern recognition. Our goal is to quickly identify objects contained in two-dimensional digital images or other inputs. More formally, given C classes of objects and a set of M labeled images representative of the object classes, which are divided into two disjoint subsets (training and testing), the problem is to identify a set of N discriminatory features and obtain a classifier to quickly assign the correct class label to unseen images. In order to achieve the goal, there are three requirements. First, it is essential that the discriminatory features can be computed rapidly. Secondly, the problem must allow

classification to occur in an iterative fashion. Each step decomposes the classification problem into several smaller and simpler problems until the final classification is reached. Finally, there must be a decoupling of the training and testing classifiers. In other words, in the training phase, the concern of the classifier is accurate classification, not speed and thus, the training classifier may be as complicated as necessary to solve the problem. On the other hand, the classifier in the testing phase needs an efficient mapping of the training classifier's solution. Therefore, the effectiveness of the testing or final classifier depends on the effectiveness of its mapping of the training classifier's input space to its output space. However, this mapping is a difficult undertaking, because of the size of the input space that must be mapped. A direct mapping of this space, in all but the simplest cases, will far exceed the memory capacity of a computer. So the problem now becomes how to decompose this massive space into a form that does not exceed memory capacity while maintaining an accurate representation of the training classifier's solution. Additionally, the decomposition must be constructed in a fashion that allows for rapid access.

2.1. Core Concept

The objective is to build a fast and accurate classifier. One way to accomplish this is to utilize a data structure that allows for quick access such as a lookup table. A lookup table can be created with dimensionality equal to that of the feature vector. The table could be constructed in such a manner that each feature corresponds to an index in the table. The feature vectors of the training set would be converted to a new set of discrete-valued vectors. Each vector element would fall within the range of the indices of the lookup table. Next, all possible combinations of the lookup table values would be classified using the KNN rule and the label stored at that position in the table. In other words, the KNN classification solution is mapped into the lookup table. Note that the choice of classifiers is not limited to a KNN and other classifiers can be used. Given the table, classification of test images would be simply to convert the feature vectors into the index space of the lookup table and returning the value at that location. This method provides a boost in classification throughput orders of magnitude higher than KNN. However, there are two main drawbacks to this method. First, in conversion from real number to discrete values, there may be a loss in accuracy if there is not sufficient resolution provided by the range of index values. The second problem is that the size of the table may easily exceed the memory capacity of the host computer. These problems are overcome by, RCT, which is a means of taking advantage of the computational efficiency of a lookup table without the enormous storage requirements.

Classification trees make decisions by subdividing a complex problem into a series of smaller easier problems. The RCT is a classification tree where each node consists of three main elements; reduction matrix, index conversion factors, and a lookup table. The construction of the tree begins with the feature vectors of the training set, which are used to compute a reduction matrix. This matrix will be used to reduce the training and test sets to a dimensionality that will allow the RCT node lookup table to fit into memory. The reduction technique that should be used is the one most effective for the particular dataset. An effective reduction matrix is essential to the success of the RCT and the ability to choose from a wide variety of reduction techniques is advantageous. Once the reduction matrix is computed, the training set is reduced to the desired dimensionality.

The next task to be accomplished is the assessment of the effectiveness of classification using the reduced training set. This step determines the type and number of children nodes. Various numbers of groupings or clusters of classes are examined from a predetermined maximum number down to two. The clustering with the highest classification accuracy is selected. If more than one clustering achieves the maximum, the one with the largest number of clusters is selected. As with the reduction, the clustering method is not fixed. Various methods can be used. However, in this set of experiments, a max-min clustering method was used. In other words, the maximum distance between all of the clusters was calculated. Then the clusters with the minimum of the maximum distances were combined. This process was continued until the desired numbers of clusters were reached. The best case is when the classification rate is 100% with each cluster consisting of a single class.

Once the best clustering is determined, each cluster becomes a child of the current node. If a cluster contains only one class, it becomes a leaf node and the classification process for that class is complete. On the other hand, if a cluster contains more than one class, it becomes an intermediate node and the process continues on the classes contained in that node. Note, that if a node cannot separate the classes completely with the reduced feature vector, the classification problem for the children nodes will be simplified due to clustering at the parent node.

After clustering is completed, the factors are computed to map the reduced feature vectors of the training set into the index-space of the lookup table for the node. The conversion factors are applied to the reduced training set. When the training set is in the index-space of the lookup table, all possible combinations of indices are classified using the nearest neighbor rule. Once the class for a given combination has been determined, the value of the cluster, to which the class is assigned, is determined. If the class is directed to a cluster that is a leaf node, the negative value of the class is entered into the table; indicating that the classification process is complete. Each intermediate node has a unique positive identifier. If the class is directed to an intermediate node, the identifier for the node is entered into the lookup table. The positive number indicates that the classification process must

continue. This will repeat until all intermediate nodes have been processed.

After tree construction is completed, classification is relatively simple. The tree is loaded into memory. The images are also loaded into memory and their initial feature vector is calculated. Beginning with the root node, the feature vector is reduced to the dimensionality of the lookup table. The reduced vector is mapped into the discrete index-space. The new vector is looked up and either a negative value of a class or the identifier of the next node in the process is returned. This is repeated until a class node is reached.

Most effective features are application dependent. In RCT there are two main requirements for the features. First, they must be computed rapidly. The second criterion is that the dimensionality of the features cannot exceed the capacity of the lookup tables of a tree node. In this paper we use Haar features [5]. Through intergal images [5], Haar features can be computed using only a few computer instructions regardless of the feature size.

As mention earlier, the desired features are the ones which best separates the data in classes. Here we use a procedure for finding the optimal linear representation of images for object recognition proposed in [3]. For a single feature, this is accomplished by computing the candidate feature for all of the training images. Once this is accomplished, a "leave one out" analysis is performed on the training set. Given a set of I images distributed over C classes with a total of k training images per class, the performance measure F is computed as follows. The minimum distance between the image that is left out and a nearest member of the same class is computed. A second minimum distance between the image that is left out and the nearest images not of the same class is computed. These two minimums are used to form the ratio in equation 1.

$$x_{c,i} = \frac{min_{c \neq c`,j} d(I_{c,i}, I_{c`,j})}{min_j d(I_{c,i}, I_{c,j}) + \epsilon} \quad (1)$$

$$h(x_{c,i}) = \frac{1}{1 + exp(-2\beta x_{c,i})} \quad (2)$$

$$F = \sum_{c=1}^{C} \sum_{i=1}^{k} h(x_{c,i}) \quad (3)$$

Note that epsilon is a small value in the denominator to prevent a division by zero error. If a good candidate is selected, the ratio will be large (greater than one); if it is poor, the ratio will be small. Instead of using the ratio directly, an exponential form of the ratio is used in equation 2. This is done to restrict the ratio between zero and one and is necessary to prevent extreme success on one image from dominating the evaluation. Again the goal is to find features that separate all classes, not just a few. The exponential ratio is summed for every image in the training set and is referred to as F. The feature with the highest F is selected. The β in equation 2 is used to control the smoothness of F. The process is then repeated, finding candidate features that work best with the previously-selected features until the desired number of features are selected.

2.2. Data Separation and Clustering

The feature extraction phase results in a set of N features. N is the minimum number of features necessary to accurately classify the training set. Because of memory constraints, d is the maximum dimensionality of a lookup table that can fit into memory and therefore, is also the maximum number of features that can be used at a given node for classification. When $N \leq d$, the N features can be used directly because there is sufficient information to completely classify the data into individual classes. The case where $N > d$ requires dimension reduction. The feature set must be projected into a smaller space of d dimensions. It is not sufficient for the projection to yield a feature set of d dimensions. It must maintain information for class discrimination. In other words, the projection should maximize the separation between data of different classes. The effectiveness of the projection is analyzed in reference to classification accuracy. If all data at the node can be classified accurately, the classification decision is stored as leaf nodes and no further processing occurs down this branch of the tree. If, on the other hand, it cannot be classified accurately, the problem then becomes a clustering one. Now the goal is to find the best grouping of classes or clusters that can be made. Accuracy is now defined in terms of clusters, not classes. The accuracy achieved through class level clustering will always be no worse than that of individual classes and in most cases, will be higher. How effective it is depends on the data and the clustering technique used. Data separation through data projections and class clustering is the key to decomposing a complicated classification problem into a series of simple ones.

3. EXPERIMENTAL RESULTS

The first and by far the most significant result is the substantial increase in classification throughput without a significant loss in accuracy. Comparison between RCT and other methods are listed in Table 1. The throughput (images/sec.) column is adjusted to account for the differences in CPU performance for different experiments. In a very recent paper, Westphal and Wurtz [6] propose a classifier based on mininum entropy with best accuracy result of 99.22%; however, the throughput is only 2.76 images per sec. On the same dataset, RCT achieved a throughput of 280,868 images/sec with 99.50% accuracy. Er et. al. [2] presented a high-speed face recognition system using DCT for reduction and a neural network for classification. On the ORL dataset, they achieved 97.45% accuracy with a throughput of 331.44 images per sec. The RCT's accuracy was 95%, however, its throughput was 76,088 images per sec. Finally the RCT was used for handwritten digit recogni-

tion. Yang et. al. [4] used KPCA and LDA for digit recognition. They utilized the CENPARMI dataset and achieved 88.79% accuracy at 41.38 digits per sec. Using the USPS digit dataset, RCT achieved 88.58% accuracy at 194,547 digits per sec. Clearly these results are significant, because in each case the RCT throughput is orders of magnitudes faster than other reported results.

Table 1. Method Throughput Comparison

Method	Dataset	Accuracy	Throughput
Min. Entropy [6]	COIL-100	99.22%	2.76
RCT	COIL-100	99.50%	280,868
Neural Network [2]	ORL	97.55%	331.44
RCT	ORL	95.50%	76,088
KPCA	CENPARMI	88.79%	41.38
RCT	USPS Digits	88.58%	194,547

There are two fundamental reasons that account for the substantial difference in performance between RCT and the other approaches. First, all of these other methods utilize the global approach to feature extraction which normally results in a matrix multiplication between two matrices of high dimensionality. This is avoided in this paper by extracting a small set of local Haar features for data representation. The second reason for the boost in performance is a conceptual change in the approach to object recognition. As mentioned earlier, the normal approach to object recognition is to train a classifier in the training phase and to use the same classifier in the testing phase. This concept is acceptable in reference to accuracy; however, it is not sufficient for throughput. For complex classification problems, the classifier must be able to effectively handle the high degree of variability. This will require varying degrees of complexities in the classifier. With all else being equal, the more complicated a classifier, the slower its performance. A classifier that performs slowly in training will perform slowly during testing. The RCT avoids this problem by not using the training classifier, rather its solution.

3.1. Comparison of Classifiers

The above comparisons are not true comparison of classifiers; they are comparisons of classification systems. For example complicated features will slow a classification system, even if the classifier is fast. Therefore, to ascertain a more accurate comparison of RCT and other classifiers, we perform experiments measuring classification time only after feature extraction is completed. The results are shown in Table 2. The RCT is compared to a neural network and a KNN classifier on both the ORL and COIL dataset. The training set, test set, and features are identical. Here we use 80% of images for training and 20% for testing; With all else being equal, clearly RCT outperforms the other classifiers by orders of magnitude. Please note **these numbers are different from those shown in Tab. 1** because here features are assumed to be computed already.

Table 2. Classifier Comparison

Classifier	Dataset	Accuracy	Throughput
KNN	ORL	100%	16,735
Neural Network	ORL	92.50%	53,522
RCT	ORL	97.50%	982,110
KNN	COIL	100%	19,204
Neural Network	COIL	89%	53,709
RCT	COIL	100%	3,781,933

4. CONCLUSION

In this paper we have proposed a framework for rapid recognition. The key components are features that can be computed efficiently and a tree structure that allows us to store solutions using a lookup table. Experimental results show the RCT is effective for a variety of recognition problems by providing a significantly higher throughput, while maintaining accuracy. Note that the proposed RCT is not limited to images and can be used for other inputs as well. For example, using the wine dataset, available at the Machine Learning respository as UC Irvine, RCT achieves 97.78% accuracy with a throughput of 3,234,153 inputs per second, while a neural network obtains 91.01% accuracy with 403,377 records per second[1].

5. REFERENCES

[1] Richard O. Duda, Peter E. Hart, David G. Stork, "Pattern Classification", Wiley-interscience Publication, 2001

[2] M. Er, W. Chen, and S. Wu, "High-Speed Face Recognition Based on Discrete Cosine Transform and RBF Neural Networks," *IEEE Transactions On Neural Networks*, vol. 16, no. 3, pp. 679–691, 2005

[3] X. Liu, A. Srivastava, and Kyle Gallivan, "Optimal linear representations of images for object recognition," *IEEE Transactions on Pattern Recognition and Machine Intelligence*, vol. 26, no. 5, pp. 662–666, 2004.

[4] J. Yang, A. F. Frangi, J. Yang, D. Zhang, and Z. Jin, "KPCA Plus LDA: A Complete Kernel Fisher Discriminant Framework for Feature Extraction and Recognition," *IEEE Transactions On Pattern Analysis And Machine Intelligence*, vol. 27, no. 2, pp. 230-244, 2005

[5] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[6] G. Westphal, R. P. Wurtz, "Fast Object And Pose Recognition Through Minimum Entropy Coding," In the *Proceedings of the 17th International Conference on Pattern Recognition*, vol. 3, pp. 53–56, 2004.

[1] Note that different neural networks will give different performance and different throughput.