Face Classification by a Random Forest

A.Z. Kouzani, S. Nahavandi, K. Khoshmanesh
School of Engineering and IT
Deakin University
Geelong, Victoria 3217, Australia

***Abstract*— This paper presents a random forest-based face image classification method. The random forest is an ensemble learning method that grows many classification trees. Each tree gives a classification. The forest selects the classification that has the most votes. Three experiments are performed. The random forest-based method together with several existing approaches are trained and evaluated. The experimental results are presented and discussed.**

I. INTRODUCTION

There has been an increasing interest in face recognition due to the importance of non-intrusive security and surveillance applications. So far there has been a number of great surveys of face recognition research, e.g. [1],[2],[3]. In addition, the face recognition webpage [4] contains an excellent collection of materials describing important existing face recognition techniques. Some popular existing techniques include: geometrical features, shape templates, intensity templates, eigenfaces, local feature analysis, independent components analysis, neural networks, elastic graph matching, trace transform, support vector machines, fractals, three dimensional, and several more. In addition, each of these methods may have numerous variations and improvements. The study of the existing face recognition systems and their performances indicates that most existing face recognition methods have one thing in common - they are limited in their performances.

The main motive for this work has been the desire to further reduce the error of facial image classification. Ensemble learning [5] which combines the decisions of multiple classifiers to from an integrated output has recently emerged as an effective classification method. The variety of the members of an ensemble is known to be an important factor in specifying its performance. A random forest [6] is an ensemble learning method that grows many classification trees. To classify an object from an input vector, the input vector is put down each of the trees in the forest. Each tree gives an classification. The forest selects the classification that has the most votes. This paper presents an classification method that employs the random forest method to classify face images. It is shown that the face classification by a random forest achieves lower classification errors than those of some popular classifiers, including the support vector machine.

The paper is organised as follows. Section II reviews the random forest method. Section III presents the experimental results. Section IV discusses the performance of the developed method as well as some existing counterparts. Finally, concluding remarks are given in Section V.

II. RANDOM FOREST

Ensemble learning [5] refers to the algorithms that produce collections or ensembles of classifiers which learn to classify by training individual learners and fusing their predictions. Growing an ensemble of trees and getting them vote for the most popular class has provided a good enhancement in the accuracy of classification. Often, random vectors are built that control the growth of each tree in the ensemble.

The ensemble learning methods can be divided into two main groups: bagging and boosting. In bagging, models are fit in parallel where successive trees do not depend on previous trees. Each tree is independently built using bootstrap sample of the dataset. A majority vote determines prediction. In boosting, models are fit sequentially where successive trees assign additional weight to those observations poorly predicted by previous model. A weighted vote specifies prediction.

A random forest [7] adds an additional degree of randomness to bagging. Although each tree is constructed using a different bootstrap sample of the dataset, the method by which the classification trees ate built is improved. A random forest predictor is an ensemble of individual classification tree predictors. For each observation, each individual tree votes for one class and the forest predicts the class that has the plurality of votes. The user has to specify the number of randomly selected variables (m_{try}) to be searched through for the best split at each node.

Whilst a node is split using the best split among all variables in standard trees, in a random forest the node is split using the best among a subset of predictors randomly chosen at that node. The largest tree possible is grown and is not pruned. The root node of each tree in the forest contains a bootstrap sample from the original data as the training set. The observations that are not in the training set, are referred to as "out-of-bag" observations.

Since an individual tree is unpruned, the terminal nodes can contain only a small number of observations. The training data are run down each tree. If observations i and j both end up in the same terminal node, the similarity between i and j is increased by one. At the end of the forest construction, the similarities are symmetrised and divided by the number of trees. The similarity between an observation and itself is set to one. The similarities between objects form a matrix which is symmetric, and each entry lies in the unit interval $[0, 1]$. Breiman defines the random forest as [7]:

> A random forest is a classifier consisting of a collection of tree-structured classifiers $\{h(\mathbf{x}, \Theta_k), k =$

1-4244-1272-2/07/$25.00

$1, \ldots\}$ where $\{\Theta_k\}$ are independent identically distributed random vectors and each tree casts a unit vote for the most popular class at input **x.**

A summary of the random forest algorithm for classification is given below [8]:

- Draw n_{tree} bootstrap samples from the original data.
- For each of the bootstrap samples, grow an unpruned classification tree, with the following modification: at each node, rather than choosing the best split among all predictors, randomly sample m_{try} of the predictors and choose the best split from among those variables. Bagging can be thought of as the special case of the random forest obtained when $m_{\text{try}} = p$, the number of predictors.
- Predict new data by aggregating the predictions of the n_{tree} trees, i.e., majority votes for classification, average for regression.

The generalisation error of a forest of tree classifiers depends on the strength of the individual trees in the forest and the correlation between them. Using a random selection of features to split each node yields error rates that compare to AdaBoost [9]. An estimate of the error rate can be obtained, based on the training data, by the following [8]:

- At each bootstrap iteration, predict the data that is not in the bootstrap sample, called "out-of-bag" data, using the tree which is grown with the bootstrap sample.
- Aggregate the out-of-bag predictions. On the average, each data point would be out-of-bag around 36% of the times, so aggregate these predictions. Calculate the error rate, and call it the "out-of-bag" estimate of error rate.

The random forest performs well compared to several other popular classifiers, including discriminant analysis, support vector machine, and neural networks. In addition, it is user-friendly as it has only two parameters: (i) the number of variables in the random subset at each node, and (ii) the number of trees in the forest. The random forest is not usually very sensitive to the values of these parameters.

III. Experimental Results

The random forest algorithm is employed to form the proposed face classification method. This section presents the evaluation results of the developed method. The obtained results are compared against those of the support vector machine [10], bagging support vector machine [11], decision tree [12], and AdaBoost decision tree [12] approaches. Using these classifiers, a number of experiments were performed. With regard to the random forest classifier, we explored: (i) different number of trees to grow, and (ii) different number of variables that are randomly sampled as candidates at each split. Concerning the support vector machine classifier, we used the support vector machine with the polynomial kernel. About the bagging support vector machine, we used ten iterations of bagging and polynomial kernel. Finally, with regard to the AdaBoost decision tree we used ten iterations of AdaBoost. Confusion matrices were first calculated for each test. Then classification errors for each class were worked out. Finally, the overall classification error for each test was found. In our experiments with the random forest, we employed Ting Wang's interface [13] to the random forest algorithm that is developed by Leo Breiman and Adele Cutler [14]. Also, in our experiments with the support vector machine, the bagging support vector machine, and the decision tree, the AdaBoost decision tree, we utilised Rong Yan's MatlabArsenal [15] that encapsulates a number of popular classification algorithms.

There are a number of face databases that could be used to test the performance of the method. The AT&T Face Database [16] (formerly The ORL Database of Faces) (see Fig. 1) is a popular face database that contains ten different images for each of the 40 distinct subjects. For some subjects, the images were taken at different times, varying the lighting, facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses). The face images were resized to 56×46.

Fig. 1. Images from the AT&T Face Database.

A. Experiment 1: 50/50

The face images were used to train and test the systems. The pixel intensities were directly used as features for classification. Therefore, the number of samples and features were 400 and 2576, respectively. Two datasets were created: training and test. 50% of the images of each class were randomly selected to form the training dataset, and the other 50% of the images were used to form the test dataset. With regard to the random forest-based method, the experiments were performed in two steps. First, the two parameters of the random forest were varied coarsely from 5 to 2576 with an increment of 275 for no-of-trees-grown, and also no-of-variables-at-each-split. Fig. 2 (top) shows a graph representation of the obtained classification errors.

Second, using the results achieved in Step 1, we varied the two parameters finely from 805 to 850 with an increment of 5 for no-of-trees-grown, and from 1 to 20 with an increment of 2 for no-of-variables-at-each-split. Fig. 2 (bottom) illustrates a graph representation of the computed classification errors.

In addition, several support vector machine-based classifiers with the polynomial kernel of different parameters and also a decision tree based classifier were developed. The support vector machine-based classifier's kernel parameter was changed from 0.05 to 0.95. Moreover, two ensemble classifiers were trained and tested: bagging support vector machine and AdaBoost decision tree.

The first approach used the support vector machine ensemble with bagging (bootstrap aggregating) [17]. In bagging, each individual support vector machine is trained independently

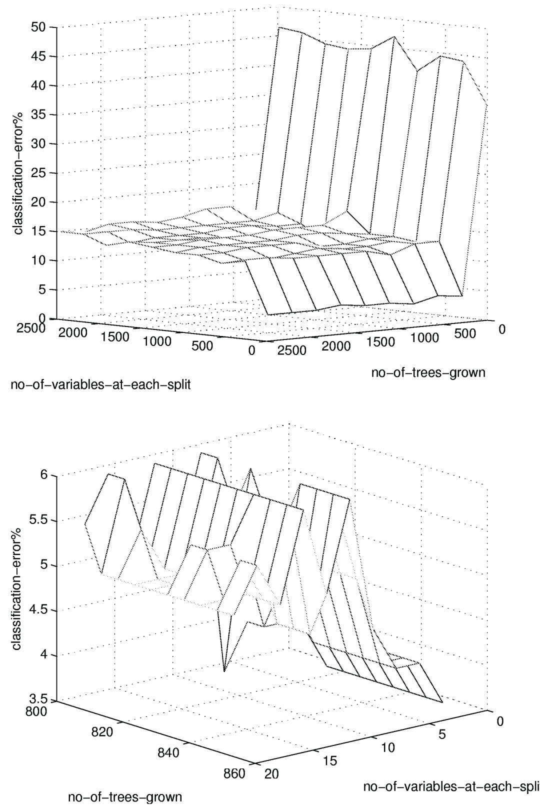

Fig. 2. Classification errors for the random forest-based classifiers created in the first step (top), and the second step (bottom).

using the training samples randomly chosen through a bootstrap technique. Then the trained individual support vector machines are aggregated to make a collective decision. We did not use the support vector machine as component classifiers for AdaBoost because AdaBoost does not perform well with strong component classifiers such as the support vector machine. The classification results are given in Table I.

As can be seen in the table, the bagging support vector machine performed well and produced the classification error as low as 4.0%. However, the lowest classification error (3.0%) was achieved by the random forest-based method employing 805 trees and 7 variables sampled at each split.

B. Experiment 2: 20/80

Two datasets were created: training and test. 20% of the images of each class were randomly selected to form the training dataset, and the other 80% of the images were used to form the test dataset.

With regard to the random forest-based method, the experiments were performed in two steps. First, the two parameters of the random forest were varied coarsely from 5 to 2576 with an increment of 275 for no-of-trees-grown, and also no-of-variables-at-each-split. Second, using the results achieved in Step 1, we varied the two parameters finely from 1905 to 1950 with an increment of 5 for no-of-trees-grown, and from 1 to 20 with an increment of 2 for no-of-variables-at-each-split. In addition, several support vector machine-based classifiers with the polynomial kernel of different parameters and also a decision tree based classifier were developed. The support vector machine-based classifier's kernel parameter was changed from 0.05 to 0.95. Moreover, two ensemble classifiers were trained and tested: bagging support vector machine and AdaBoost decision tree. The classification results are given in Table II.

TABLE I

CLASSIFICATION ERRORS FOR 50/50.

Classifier	Parameters	Error %
support vector machine	polynomial kernel, 0.05	98.5
	polynomial kernel, 0.25	36.5
	polynomial kernel, 0.50	6.5
	polynomial kernel, 0.75	4.5
	polynomial kernel, 0.90	4.4
	polynomial kernel, 0.95	4.5
bagging support vector machine	polynomial kernel, 0.90 10 iterations for bagging	4.0
decision tree	none	47.5
AdaBoost decision tree	10 iterations for AdaBoost	24.5
random forest	5 trees, 5 variables	37.0
	805 trees, 1 variables	4.0
	805 trees, 7 variables	3.5
	850 trees, 20 variables	5.0
	2480 trees, 2480 variables	15.0

TABLE II

CLASSIFICATION ERRORS FOR 20/80.

Classifier	Parameters	Error %
support vector machine	polynomial kernel, 0.05	99.0
	polynomial kernel, 0.25	91.5
	polynomial kernel, 0.50	44.0
	polynomial kernel, 0.75	42.8
	polynomial kernel, 0.92	40.9
	polynomial kernel, 0.95	40.9
bagging support vector machine	polynomial kernel, 0.92 10 iterations for bagging	65.3
decision tree	none	70.0
AdaBoost decision tree	10 iterations for AdaBoost	62.8
random forest	5 trees, 5 variables	37.0
	1905 trees, 1 variables	13.7
	1925 trees, 5 variables	12.1
	1950 trees, 20 variables	14.0
	2480 trees, 2480 variables	17.7

As can be seen in the table, only the random-forest-based method performed reasonably. The lowest classification error (12.1%) was achieved by the random forest-based method employing 1925 trees and 5 variables sampled at each split.

C. Experiment 3: 80/20

Two datasets were created: training and test. 80% of the images of each class were randomly selected to form the training dataset, and the other 20% of the images were used to form the test dataset.

With regard to the random forest-based method, the experiments were performed in two steps. First, the two parameters of the random forest were varied coarsely from 5 to 2576 with an increment of 275 for no-of-trees-grown, and also no-of-variables-at-each-split. Second, using the results achieved in Step 1, we varied the two parameters finely from 250 to 295 with an increment of 5 for no-of-trees-grown, and from 1 to 20 with an increment of 2 for no-of-variables-at-each-split. In addition, several support vector machine-based classifiers with the polynomial kernel of different parameters and also a decision tree based classifier were developed. The support vector machine-based classifier's kernel parameter was changed from 0.05 to 0.95. Moreover, two ensemble classifiers were trained and tested: bagging support vector machine and AdaBoost decision tree. The classification results are given in Table III.

TABLE III

CLASSIFICATION ERRORS FOR 80/20.

Classifier	Parameters	Error %
support vector machine	polynomial kernel, 0.05	95.0
	polynomial kernel, 0.25	90.0
	polynomial kernel, 0.46	3.7
	polynomial kernel, 0.50	3.7
	polynomial kernel, 0.75	5.0
	polynomial kernel, 0.95	5.0
bagging support vector machine	polynomial kernel, 0.46 10 iterations for bagging	2.5
decision tree	none	47.5
AdaBoost decision tree	10 iterations for AdaBoost	27.5
random forest	5 trees, 5 variables	23.7
	250 trees, 1 variables	2.5
	255 trees, 9 variables	0.0
	295 trees, 20 variables	3.7
	2480 trees, 2480 variables	6.2

As can be seen in the table, the random forest-based method has performed very well. The lowest classification error (0.0%) was achieved by the random forest-based method employing 255 trees and 9 variables sampled at each split.

IV. DISCUSSIONS

This study has been motivated by emergence of ensemble-based classification approaches, and also the importance of robust automated face recognition. The results demonstrate that the proposed random forest-based method performs better than the support vector machine as well as the bagging support vector machine, and the AdaBoost decision tree classifiers in all experiments. The lowest classification error (0.0%) was produced by the random forest-based method with 255 no-of-trees-grown and 9 no-of-variables-at-each-split for the 400 face images.

While the two ensemble learning approaches, the bagging support vector machine and the AdaBoost desicion tree improved the performance of their non-ensemble versions, the support vector machine and the decision tree, in Experiments 1 and 3, they produced higher classification errors in Experiment 2 where only 20% of face images used for training. The two ensemble learning approaches were not able to beat the random forest-based method.

The random forest-based method, that is an ensemble learning method which grows many classification trees, has shown to be an accurate classifier as it has performed very well for the face classification problem considered in this work. The method has produced the lowest classification error amongst the methods tested in the three experiments.

V. CONCLUSION

A random forest-based face classification method was presented in this paper. Three experimenters were carried out. Different training/test image ratios were utilised in each experiment: 50/50, 20/80, and 80/20. 400 face images of the AT&T Face Database were used. The random forest-based method together with the support vector machine, bagging support vector machine, decision tree, and AdaBoost decision tree approaches were trained and tested using the same training and test images. The lowest classification error (0.0%) was produced by the random forest-based method with 255 no-of-trees-grown and 9 no-of-variables-at-each-split for the 400 face images. The random forest-based method proved to be an accurate classifier for facial recognition applications.

ACKNOWLEDGMENT

The support of the Australian Research Council (ARC) under a Discovery Grant is gratefully acknowledged.

REFERENCES

[1] W. Zhao, R. Chellappa, A. Rosenfeld, and P. J. Phillips, "Face recognition: A literature survey," *ACM Computing Surveys*, vol. 35, no. 4, pp. 399–458, December 2003.

[2] A. S. Tolba, A. H. El-Baz, and A. A. El-Harby, "Face recognition: A literature review," *International Journal of Signal Processing*, vol. 2, no. 1, pp. 88–103, 2005.

[3] K. W. Bowyer, K. Chang, and P. Flynn, "A survey of approaches to three-dimensional face recognition," in *Proceedings of the 17th International Conference on Pattern Recognition*, vol. 1, 2004, pp. 358–361.

[4] Face Recognition Homepage. [Online]. Available: http://www.face-rec.org/

[5] J. Lu, K. Plataniotis, A. Venetsanopoulos, and S. Li, "Ensemble-based discriminant learning with boosting for face recognition," *IEEE Trans. on Neural Networks*, vol. 17, no. 1, pp. 166 – 178, Jan 2006.

[6] Wikipedia, "Random Forests." [Online]. Available: http://en.wikipedia.org/wiki/Random_forests

[7] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5 – 32, 2001.

[8] A. Liaw and M. Wiener, "Classification and regression by randomForest," *R News*, vol. 2, no. 3, p. 1820, 2002.

[9] Y. Freund and R. Schapire, "A short introduction to boosting," *Journ. of Jap. Soc. for Artificial Intelligence*, vol. 14(5), pp. 771–780, 1999.

[10] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer-Verlag, 1999.

[11] H. Kim, S. Pang, H. Je, D. Kim, and S. Ban, *Pattern Recognition with Support Vector Machines: First International Workshop, SVM 2002 Proceedings*. Springer Berlin/Heidelberg, 2002, ch. Support Vector Machine Ensemble with Bagging.

[12] J. Basak, "Online adaptive decision trees: Pattern classification and function approximation," *Neural Computing*, vol. 18, no. 9, pp. 2062 – 2101, Sep 2006.

[13] T. Wang, "Random Forests." [Online]. Available: Available:http://lib.stat.cmu.edu/matlab/RandomForest.zip

[14] L. Breiman and A. Cutler, "Random forests." [Online]. Available: www.stat.berkeley.edu/users/breiman/RandomForests/cc_home.htm

[15] R. Yan, "MatlabArsenal." [Online]. Available: http://finalfantasyxi.\\inf.cs.cmu.edu/MATLABArsenal/MATLABArsenal.htm

[16] "AT&T Face Database." [Online]. Available: Available:http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

[17] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24(2), 1996.