Random Convolution Ensembles

Michael Mayo

Dept. of Computer Science
University of Waikato
Private Bag 3105, Hamilton, New Zealand
mmayo@cs.waikato.ac.nz

Abstract. A novel method for creating diverse ensembles of image classifiers is proposed. The idea is that, for each base image classifier in the ensemble, a random image transformation is generated and applied to all of the images in the labeled training set. The base classifiers are then learned using features extracted from these randomly transformed versions of the training data, and the result is a highly diverse ensemble of image classifiers. This approach is evaluated on a benchmark pedestrian detection dataset and shown to be effective.

Keywords: Image Classification; Random Convolution; Pedestrian Detection

1 Introduction

Methods for the automatic classification of multi-dimensional data objects are one of the central themes in pattern recognition research. Although the most common class of such object is the two-dimensional array, or image, methods should ideally scale to data objects in any number of dimensions.

In contrast to this, machine learning deals with techniques for classifying one-dimensional objects, referred to variously as "instances", "records", or "feature vectors". The most recent machine learning techniques to date, such as support vector machines [1], random forests [2], and instance-based methods (see, e.g. [3]), have proven to be extremely effective feature vector classifiers. The main difficulty that arises is usually deciding which of the techniques (along with its associated parameters) to actually use: currently, this is an empirical problem.

When it comes to designing image classifiers, there is a second significant degree of freedom: how to map the high-dimensional objects in the dataset onto one-dimensional feature vectors, in order to use machine learning for classification. Typically a direct one-to-one mapping of pixels to features is not the best option. Numerous solutions have therefore been proposed in the past, from classical colour histograms (of which there are many variants, e.g. the colour coherence vector [4]), to spatial pyramids [5], to locally receptive fields [6,7]. However, like the problem of classifier selection, there are no hard and fast rules when it comes to a particular problem.

For the remainder of this paper, the term "image classifier", therefore, will be used to denote a system comprising these two main components: a feature extraction function for transforming a multi-dimensional object into a one-dimensional feature vector, along with a machine learning classifier for making a prediction about the image's class given the feature vector.

The main contribution of this paper, then, is to propose a new ensemble method called Random Convolution Ensembles (RCEs) for enhancing the performance of a base image classifier. This method works with any machine learning classifier and feature mapping function. The basic idea is that each base image classifier is trained on a randomly (but consistently) transformed copy of the entire training image set. Each base classifier consequently "sees" a different set of feature vectors extracted from the same training images. When all of the base classifiers are trained, the result is a diverse set of image classifiers whose performance as an ensemble outperforms the performance of any one of the base classifiers individually.

The basic details and motivation underlying RCEs are given in the next section. Section 3 discusses the benchmark pedestrian detection image dataset [7] used to evaluate this technique, and Section 4 describes an experiment on the benchmark dataset, showing that RCEs are effective image classifiers. In Section 5, we add an element of selection to the generation of random image transformations, and describe a second experiment showing that performance improves as a result. Section 6 concludes the paper and discusses the way forward.

2 Random Convolution Ensembles

Ensemble methods are extremely popular in machine learning. The rationale behind them is to learn not a single classifier, but a group of them, where each member of the group is designed to solve the same classification problem. It is important to ensure somehow that each individual classifier is different from the others in the group, because the more similar their structure (and therefore their predictions), the less effective overall the ensemble will be.

Methods for ensuring ensemble diversity include training each classifier with only a random subset of the feature vectors, a method known widely as bagging [8]; weighting the feature vectors in the training data differently for each classifier (e.g. boosting [9]); or even training completely different types of classifier and then combining their predictions in some way (e.g. voting [10] and stacking [11]).

All of these methods work with one-dimensional data. While standard ensemble methods can be used with images (for example, images can be bagged), such approaches do not take advantage of the multi-dimensional nature of the data in its original form (because, for example, bagging images is the same as bagging the feature vectors derived from the images).

The RCE approach proposed here overcomes this problem by generating diversity *before* the feature extraction step, by producing multiple randomly transformed copies of the training images. The basic idea is to take the complete set of training images and make n copies of them. N random image transformations are then generated, and all the images in the ith copy of the training set (where $1 \leq i \leq n$) are transformed by the

corresponding *i*th transformation. The result is *n* copies of the original training set, each transformed in a random but consistent way. We then learn *n* base machine learning classifiers, one on each of the feature vector sets extracted from the transformed copies of the training images. When a test image is to be classified, all of the *n* base image classifiers make a prediction, and the results are combined by a meta-classifier to produce a single, final prediction.

Figure 1 depicts the architecture of an RCE, from the perspective of a test image *I* about to be classified. In the figure, the random image transformation is convolution [13] by a randomly generated kernel, and these kernels are specified by op_1, ops_2, … The *Conv(..)* function denotes the application of one of them to *I* to produce a new, distorted image. We also assume a uniform feature extraction function *Features(.)* for transforming images into feature vectors. The diversity in the ensemble derives from the fact that each of the base machine learning classifiers ($Class_1..Class_n$) in the figure is trained on a different but complete set of feature vectors. In the figure, there are *n* feature vectors $F_1…F_n$ derived from the test image *I*. A meta-classification step at the end combines all of the predictions for each feature vector.

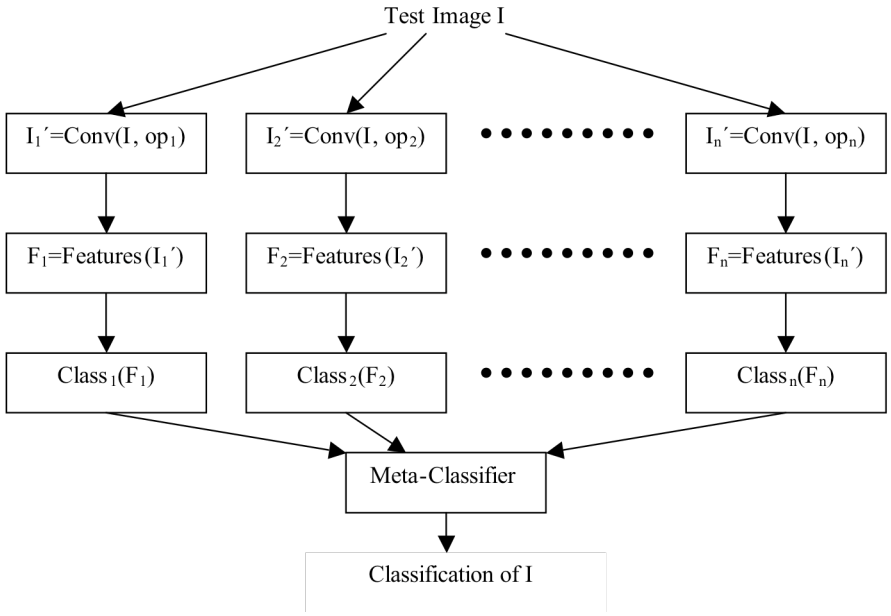

Fig. 1. Architecture of a Random Convolution Ensemble when presented with test image *I*.

3 Benchmark Dataset

This new approach to image classification was evaluated against a benchmark image dataset developed by Munder & Gavrilla [7]. The dataset was proposed in order to compare different solutions to the problem of pedestrian detection in images captured from urban, outdoor environments. The classification problem is binary, in that

images either depict a pedestrian (the positive class) or they do not (the negative class).

The basic version of the benchmark data consists of three training sets and two test sets. Each of these sets comprises 800 positive pedestrian images and 5000 negative, non-pedestrian images. To equalize the classes, the positive images were copied, mirrored and shifted by a few pixels in a random direction to produce five new, slightly different, positive examples for each of the original positive examples. This resulted in 4800 positive images in total. Figure 2 gives examples of some of the images in the datasets. Note that the negative examples were deliberately chosen to be challenging, with many vertical lines similar to those in the positive class (as opposed to an easier negative class with many uniform textures, which would be straightforward to distinguish). The size of each image is a uniform 18x36 grey scale pixels.

To perform an experiment using this dataset, a classifier should trained on the union of two of the three training datasets, and tested on one of the test datasets. Thus, there are six possible different train/test experiments. The results over all six runs should be averaged in order to obtain an overall more reliable and final estimate of any classifier's performance.

Fig. 2. Examples of positive (top row) and negative (bottom row) images in the pedestrian detection dataset proposed by [7].

4 Random vs. Standard Convolution Operators in the Ensemble: A Comparison

An RCE classifier as described in Section 2 was implemented in the Java programming language, within the WEKA (version 3.5.5) machine learning framework [12].

The random image transformation implemented was convolution [13] using a randomly generated 3x3 convolution operator or kernel. For the random operators used in this paper, we set each element in the 3x3 kernel to a random number sampled uniformly in the range -2.5…2.5.

The features used in this experiment (extracted by the *Features(.)* function in Fig. 1) were determined by the following process. Firstly, the image was divided into square blocks of size *s***s* pixels. The blocks were allowed to overlap by 50% in both horizontal and vertical directions, thus ensuring that any important features would not be lost due to the boundary between two blocks. The block size parameter *s* was set to one of the values from the set {18, 9, 6}.

For each block, the sum, mean, variance, skewness, and kurtosis of the pixel values were calculated. These statistics basically describe the intensity histogram for the block. A feature vector was then constructed for each image by concatenating the statistics for all of the blocks in the image. This results, for block size s=18, in 25 features per image; for s=9 it results in 105 features; and for s=6, there are 275 features.

In the experiments performed in this section and the next, the base classifier was set to bagged random forests. Random forests [2] are an ensemble classifier in which each individual classifier is a decision tree learned from a random subset of the features in the dataset. Individual decision trees in the forest average their predictions to give a final prediction for the entire ensemble.

Bagging random forests considerably speeds up the training process, whilst (in our initial tests) only slightly impairing performance compared to a single random forest classifier trained on the entire dataset. In this experiment, each bag contained 5% of the training data, randomly selected. There were 30 bags, meaning that feature vectors could be selected for more than one bag. Each random forest classifier consisted of ten decision trees.

The meta-classifier used to combine all of the individual image classifier's predictions was voting [10], which is straightforward averaging.

The question that the first experiment set out to answer was: does this new method work at all? In other words, does randomly convolving the training data in the way described actually produce sufficiently variable sets of feature vectors for the purpose of creating a diverse ensemble of image classifiers? Or would simply convolving the images with standard operators such as edge detectors do just as well? Indeed, does this new approach provide any gain at all over the simplest possible approach, that of extracting the features directly from the original image and learning only a single base image classifier, without any image convolution at all? (This latter approach is actually the most common taken in the literature.)

$$\mathrm{ID} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad \mathrm{PK} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

$$\mathrm{G_x} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \qquad \mathrm{G_y} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

Fig. 3. Common standard convolution operators: the null filter (ID), which does nothing; a Laplacian peak point detector (PK), which detects bright points; and the Sobel operators for edge detection (G_x and G_y).

Table 1. Convolution operator sets used in Experiment 1.

OPS_1	{ID}
OPS_2	{ID, G_x, G_y, PK}
OPS_3	{R_1, R_2, R_3, R_4}
OPS_4	{ID, G_x, G_y, PK, R_1, R_2, R_3, R_4}
OPS_5	{R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8}

To set the experiment up, four different convolution operators as depicted in Figure 3 were grouped, along with some randomly generated operators, into five sets as shown in Table 1. Each set in Table 1 effectively defines an RCE of size n, where n is the size of the set. OPS_1 is clearly the simplest, corresponding to an ensemble with a single base image classifier (i.e. n=1 in Figure 1) and no image convolution. OPS_2 corresponds to the set of standard image convolution kernels shown in Figure 3, and OPS_3 is four randomly generated convolution kernels. OPS_4 is the set of size n=8 obtained by taking the union of OPS_2 and OPS_3, and this set is a mixture of both random and standard convolution operators. OPS_5, the last of them, consists of eight randomly generated convolution operators. Note that only OPS_3 and OPS_5 define "true" RCEs as per the definition given in the Section 2; the other ensembles are the baselines for comparison.

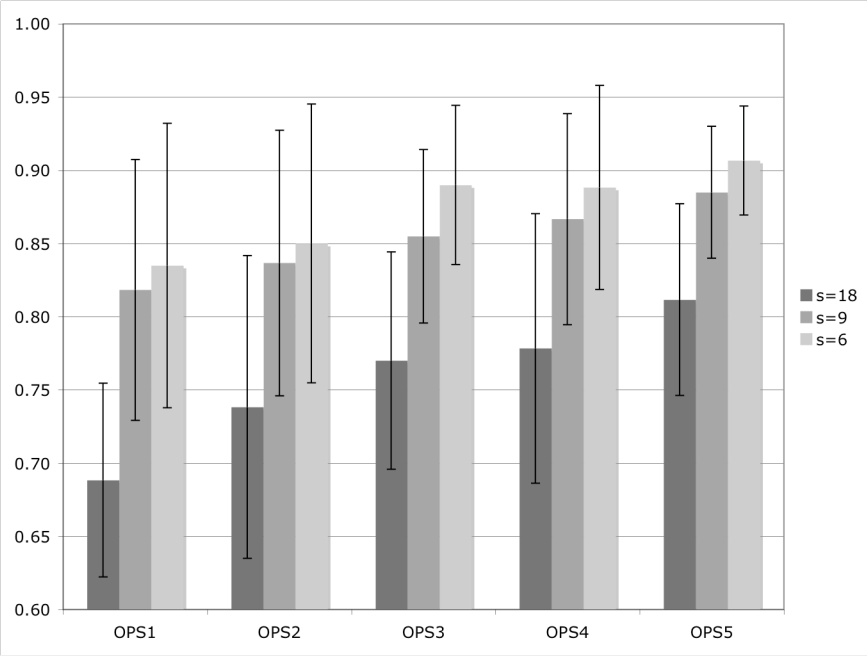

Fig. 4. Results of Experiment 1. The set of convolution operators used to construct the ensemble is specified on the x-axis, and average AUC after six independent train/test experiments is given on the y-axis. The error bars on the columns depict the standard deviation. Results are shown for features extracted using blocks of size *s*x*s* where $s \in \{18, 9, 6\}$.

For each experiment, the Area Under the ROC Curve (AUC) was calculated [14]. When using AUC, the worst possible classifier (i.e. a random classifier) should have an AUC of 0.5, while the best possible classifier (i.e. a 100% perfect classifier) should have an AUC of 1.0. Six train/test runs were performed for each combination of convolution operator set OPS_i and block size s, and Figure 4 depicts the average results.

First of all, the results clearly show that performance depends on the block size s. In every case, RCEs with a smaller block size for feature extraction have a higher final AUC. Interestingly, the ensemble with a single base image classifier without convolution (as specified by OPS_1) is the worst overall performer: AUC values for this classifier range from 0.69 to 0.84, and in every case, larger ensembles beat it.

Most importantly, Figure 4 also shows that the "true" RCEs always outperform ensembles defined using standard convolution operators, or mixtures of random and standard operators. For example, using OPS_3 as the set of convolution operators results in considerably better performance than using the set of standard operators, OPS_2 (the difference is 0.89 AUC compared to 0.85 when the block size s=6). And when the ensemble size is n=8, the completely random set OPS_5 consistently outperforms the set OPS_4, a mixture of random and standard operators (compare 0.91 AUC to 0.89 when s=6).

A second way important way in which Figure 4 shows that RCEs are superior to the other methods is the variance. In Figure 4, each average AUC column is depicted with an error bar showing the standard deviation of the AUC over the six train/test runs. It turns out that whenever only random operators are used, the standard deviation is much smaller. For example, the OPS_3 ensemble has a standard deviation of 0.06 compared to 0.10 for OPS_2, and for the larger ensembles, the standard deviation is 0.04 for OPS_5 compared to 0.07 for OPS_4. This implies that OPS_3 and OPS_5 define ensembles that are not only more accurate than the others, but they are also less sensitive to the variations in the quality of the training and testing data.

5 Random Convolution Ensembles with Selection

The previous experiment established that RCEs outperform (i) a single base image classifier that extracts features directly from the original images, and (ii) similar ensembles, differing only in that they use standard convolution operators, or a mixture of standard and random operators, as opposed to purely random operators.

In the next experiment, we wanted to determine if performance could be further improved by not only randomly generating image transformations, but also selecting them. Previously, if an RCE was of size n, then n random operators were generated. No consideration was given to the fact that one or more of the operators could potentially be useless, therefore impairing the entire ensemble. For example, a randomly generated convolution operator in which all the entries were nearly zero would effectively erase the images, making the learning task impossible for that particular member of the ensemble.

To add selection to the generation process, therefore, we performed the following steps. For every base image classifier that was required, two base image classifiers were considered, each one with a different randomly generated convolution operator. Both of them were then evaluated in a stratified two-fold cross-validation experiment on the training data. The classifier with the least number of classification errors was then retained in the final ensemble, while the other classifier was discarded. In other words, if the ensemble size was n, then $2n$ base image classifiers were generated and trained, but only half of them were retained in the final ensemble.

We considered RCEs of size n=4 and n=8 in order to compare the results of this second experiment to the first. Let OPS_3* and OPS_5* be the final set of convolution operators arrived at using generation with selection. The performance of ensembles defined by these sets on the pedestrian dataset is depicted in Figure 5, alongside the performance of the corresponding ensembles created without selection from the previous experiment.

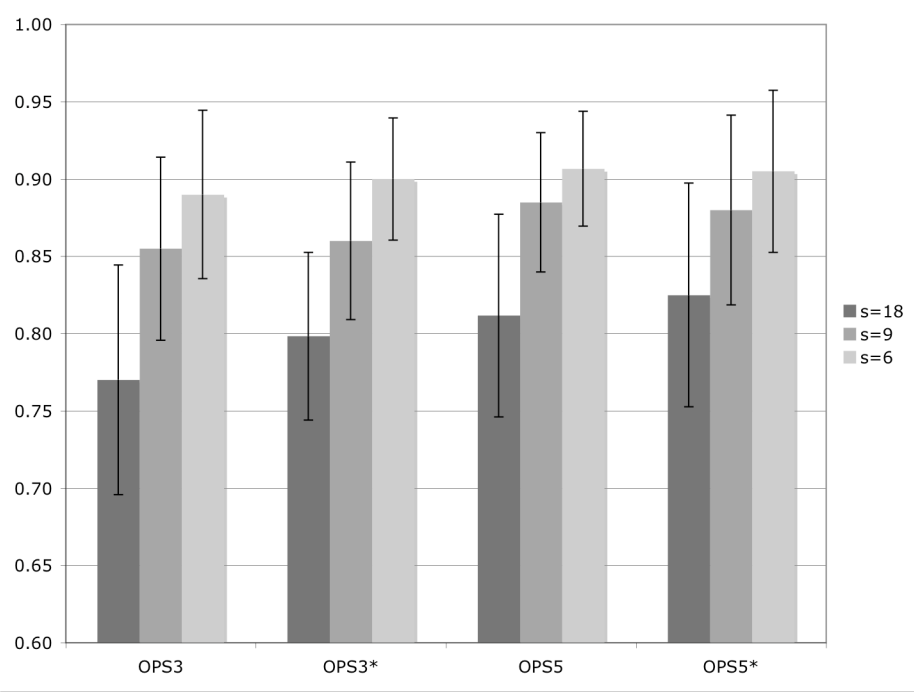

Fig. 5. Results of Experiment 2. The set of convolution operators used to construct the ensemble is specified on the x-axis, and average AUC after six independent train/test experiments is given on the y-axis. The error bars on the columns depict the standard deviation. Results are shown for features extracted using blocks of size sxs where $s \in \{18, 9, 6\}$.

The results show that adding selection to the generation process does sometimes improve the final performance. For block size s=18, selection always leads to an improvement in average AUC, while for the smaller block sizes, selection only leads to slight improvements in AUC for the n=4 ensemble (i.e., *OPS3** performs better than *OPS3*).

Although the gain due to selection is only slight in this experiment, it is suggestive that more sophisticated selection methods would produce much greater gains.

6 Concluding Remarks

This main contribution of this paper is a new method for enhancing the performance of a base image classifier. RCEs were compared to a more traditional image classifier in which features are extracted directly from the original image without using convolution and classified, and also to the strategy of extracting features from versions of the image convolved in standard ways (e.g. edge-detected versions of the training images). Furthermore, adding selection to the random operator generation process sometimes improves performance even more.

We did not compare the results obtained here directly to those of Munder & Gavrilla [7], primarily because their focus was on searching for the best features and classifier for the sole purpose of pedestrian detection. In contrast, we used different, less computationally expensive features and classifier, so as to evaluate this new approach. However, the best result (0.91 AUC) is comparable to Munder & Gavrilla's result for same version of the dataset, which they report as a 90% detection rate for a 10% false positive rate. It would be interesting to implement the same features and classifier as Munder & Gavrilla to determine if RCEs can further enhance performance.

Future work in this area will look at more intelligent methods of random convolution operator generation and selection. For example, a simple hill-climbing algorithm could be used to iteratively improve the quality of a single convolution operator after its initial random generation. If this hill-climbing-based classifier is then boosted, the result will be an RCE that uses both selection, and weighted voting rather than unweighted voting at the meta-classification stage. A more extreme idea in the same vein is to simultaneously evolve, from a random starting point, the n convolution operators using a genetic algorithm.

To conclude, the results presented in this paper are a proof-of-concept that the idea of randomly transforming training images in order to construct a diverse ensemble of image classifiers works. Future work will continue to build on and evaluate this promising new approach. We are interested in applying this technique not only to pedestrian detection, but also to other standard datasets in the literature, in areas such as object detection and natural scene classification.

References

1. Keerthi S., Shevade S., Bhattacharyya C. & Murthy K. 1999. Improvements to Platt's SMO algorithm for SVM classifier design. *Neural Computation* 13(3): 637-649.
2. Breiman L. 2001. Random Forests. *Machine Learning* 45(1): 5-32.
3. Atkeson C., Moore A. & Schaal S. 1996. Locally Weighted Learning. *AI Review* 11: 11-73.
4. Pass G., Zabih R. & Miller J. 1997. Comparing images using color coherence vectors. In Aigrain P. et al. (Eds.) *Proceedings of the 4th ACM international conference on Multimedia*, pp. 65-73.
5. Lazebnik S., Schmid C. & Ponce J. 2006. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *Proceedings of the 2006 IEEE*

Computer Society Conference on Computer Vision and Pattern Recognition, pp. 2169 – 2178.
6. Fukushima K., Miyake S. & Ito T. 1983. Neocognitron: A neural network model for the mechanism of visual pattern recognition. *IEEE Trans. on Systems, Man, and Cybernetics* 13: 826-834.
7. Munder S & Gavrilla D. 2006. An experimental study on pedestrian classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(11): 1863-1868.
8. Breiman L. 1996. Bagging Predictors. *Machine Learning* 24(2): 123-140.
9. Freund Y. & Schapire R. 1996. Experiments with a new boosting algorithm. In *Proc. of the 13th International Conference on Machine Learning*, pp. 148-156.
10. Kittler J., Hatef M., Robert P., Duin W. & Matas J. 1998. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20(3): 226-239.
11. Wolpert D. 1992. Stacked generalization. *Neural networks* 5: 241-259.
12. http://www.cs.waikato.ac.nz/~ml/
13. Seul M., O'Gorman L & Sammon M. 2000. *Practical Algorithms for Image Analysis*. Cambridge University Press.
14. http://en.wikipedia.org/wiki/Receiver_operating_characteristic