Effective Classifiers for Detecting Objects

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

Abstract

Several state-of-the-art machine learning classifiers are compared for the purposes of object detection in complex images, using global image features derived from the Ohta color space and Local Binary Patterns. Image complexity in this sense refers to the degree to which the target objects are occluded and/or non-dominant (i.e. not in the foreground) in the image, and also the degree to which the images are cluttered with non-target objects. The results indicate that a voting ensemble of Support Vector Machines, Random Forests, and Boosted Decision Trees provide the best performance with AUC values of up to 0.92 and Equal Error Rate accuracies of up to 85.7% in stratified 10-fold cross validation experiments on the GRAZ02 complex image dataset.

1 Introduction

Object detection is the problem of building a classifier that can detect a particular class of object, such as a car or a person, in an image.

Two general approaches to the learning of object detectors appear in the literature: the "strongly supervised" approach, and the less common but considerably more difficult "weakly supervised" method. In the strongly supervised case, the training images are neatly segmented into object and background so that only parts of the target object are used for training. For example, face detectors are trained only on small images of faces that have been removed from larger overall images. Parts of images not containing a face can be used as the negative class.

On the other hand, depending on the circumstances, it may be infeasible to preprocess the training images in this way. Therefore strongly supervised object detection becomes unworkable. One reason may be the cost: it could simply require too much time to manually segment each training image, especially if there are hundreds of such images. An example of such an application would be a personalized photo collection where the user chooses the classes and training must occur immediately from a few example images in order to classify an entire collection.

Each training image can therefore be labeled as either positive (contains the object relevant to the class) or negative (does not contain the object), with no further information provided. This is the weakly supervised object detection that is the focus of this paper.

In addition to weak supervision, there is also considerable interest in building object detectors from training data that is complex. What is meant by "complexity" is that the training images do not contain the object of interest in the immediate foreground of the image, but only "somewhere" in the image. Many image databases such as Caltech-101 [1] consist of images with the objects of interest in a dominant foreground position, occupying most of the image.

Figure 1. Examples of (a) the planes and cars categories in the Caltech101 database [1], and (b) the bikes, cars and people categories from the GRAZ02 database [6].

Alternatively, the GRAZ02 database [6] contains more variability with respect to scale and clutter. Objects of interest are often occluded, and they are often not dominant in the image. Thus this dataset is a more complex dataset to learn classifiers from, but of more interest because it better reflects the real world complexity likely to occur in practical applications.

Figure 1 depicts some images from the Caltech and GRAZ02 databases to illustrate this difference in complexity. Because the GRAZ02 database is the more complex set of images, it is the focus of the experiments reported here.

Finally, there is also interest in the amount of training data required to learn effective object detectors. Acquiring image data can be expensive, and feature extraction and training times given large datasets can require considerable computational resources in terms of training time and memory. To this end, all the experiments reported here were repeated with varying amounts of training data, specifically 10%, 50% and 90% of the images in the GRAZ02 database, and the remainder as test data. After giving the results of these experiments, a comparison with previous works reported in the literature will be made.

2 Background

In this section, the features used to represent the images, and the machine learning classifiers that were compared and evaluated, are described.

2.1 Ohta Color Space

For image classification applications, the transformation of images into the Ohta color space [4] is advantageous as opposed to working with them directly in the more common Red-Green-Blue (RGB) color space. Several papers in the literature such as [8] and [2] have compared Ohta color space histograms with histograms derived from other color spaces, and concluded that the Ohta color space is the most effective.

Briefly, the Ohta color space is a linear transform of the RGB color space in such a way that the intensity component of each color is separated from the color components. The two remaining color components are orthogonal.

The definition of the Ohta color space is:

$$I_1 = \frac{R+G+B}{3}$$
$$I_2 = R-B$$
$$I_3 = (2G-R-B)/2$$

where R, G and B refer to the red, green and blue components respectively, and I_1, I_2 and I_3 are components of the Ohta color space. Clearly, the I_1 component corresponds to the intensity value of the color, while the color information has actually been moved exclusively to the I_2 and I_3 components.

2.2 Local Binary Patterns

Pattern and structure information are represented in these experiments by histograms of Local Binary Patterns (LBPs), as described in [5]. Although LBPs are most commonly used for recognizing textures, they are also useful for capturing the structure of images such as the number of pixels falling on edges, corners, and points.

A LBP is a description of the intensity variation around the neighborhood of a particular point in the grey-scale (intensity) version of an image. LBPs can be used to represent troughs (dark points), peaks (bright points), edges, corners, and everything in between. They are also invariant to rotation. Figure 2 depicts the points that must be sampled around a particular point (x,y) in order to calculate the LBP at (x,y).

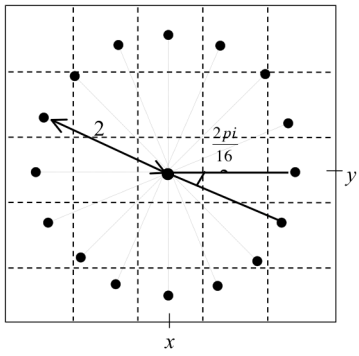

Figure 2. Points sampled to calculate the LBP around a point (x,y).

In our implementation, each sample point lies at a distance of 2 pixels from (x, y) with an equal angular spacing of 2pi/16 radians. Different values for the number of sample points and the radius are possible, but in the experiments described in this paper we considered only the depicted values.

Next, the intensity of each sample point around (x,y) is measured with those not falling in the centre of a pixel being sampled using bilinear interpolation. From these samples S[0], S[1], …, S[15], a bit string B of length 16 can be calculated where:

$$B[i] = \begin{cases} 1 & S[i] \geq I[x,y] \\ 0 & otherwise \end{cases}$$

for i=0..15, where I[x,y] is the intensity at point (x,y).

Once the bit string has been calculated, it is then bitwise circularly rotated so that it has the maximum possible number of most significant bits: this effectively achieves rotation invariance.

Finally, the bit string is assigned to a LBP category, and this assignment rule is straightforward: if the bit string has no more than two 0→1 or 1→0 transitions, then it is assigned to the LBP category specified by the number of 1s in the string, and there are 17 of these (as the bit string can have between 0 and 16 on bits inclusive). Otherwise, the bit string is assigned to a catch-all "non-LBP" category. The net result of this is that most points falling on edges, corners, peaks and troughs tend to fall into one of the LBP categories, while other points with more complex and noisy neighborhoods are discarded in the non-LBP

category. A histogram of LBP frequencies can then be constructed for a particular image.

For a more complete introduction to LBPs with comprehensive examples, the reader is referred to [5].

2.3 Data Mining Classifiers

A number of state-of-the-art machine learning classifiers were evaluated during this research for the purposes of constructing an object detector via supervised learning. Most data mining textbooks (e.g. [10]) contain more details for the interested reader.

The classifiers utilized were Support Vector Machines (SVMs), Random Forests, and Boosted Decision Trees, along with three meta-classifier combinations of these individual classifiers known as Voting, Stacking and Grading.

Briefly, SVMs are classifiers that construct a maximum margin hyperplane between positive and negative examples, which is then used to classify unseen examples. Random Forests are collections of decision trees, in which each individual decision tree is learned in a standard way but with the exception that only a small random subset of the attributes (typically 10%) is available for learning. The votes of the individual trees in the forest are then averaged to classify new examples. Boosted Decision Trees refers to a method in which each decision tree is learned on the entire dataset, but after the first tree is learned, the weights of the instances in the data are adjusted so that incorrectly classified examples are given a higher weight. The decision tree algorithm is then run again, but this time with a focus on the "harder" examples. This process is repeated many times. The resulting collection of decision trees classifies new examples in a similar manner to the Random Forest, by averaging the individual predictions, except that the individual trees are weighted.

Three meta-classifiers were also used in these experiments: Voting, Stacking, and Grading. Voting takes a number of individual base classifiers (in this case, some combination of Random Forest, SVM or Boosted Decision Tree classifier) and simply averages the individual prediction. Stacking, on the other hand, takes the base classifiers and constructs a new dataset in which each instance consists of the predictions of the individual base classifiers. It then attempts to learn a classifier not from the original features, but from the predictions of the base classifiers. Grading, finally, is an approach in which a meta-classifier attempts to learn which of the base classifiers will perform best on a particular example. New examples are not classified by combining predictions in this case, but instead by selecting the single "best" base classifier to make the classification.

3 Classifier Evaluation

In this section, the experimental setup is described.

3.1 Dataset

The GRAZ02 dataset [6], a collection of 640 x 480 24-bit color images, was selected as the most challenging dataset for these experiments. As Figure 1 illustrates, the GRAZ02 dataset is interesting because of occlusions, differences in scale, and variations in the viewpoint of the target objects.

There are three binary classification problems in the GRAZ02 dataset: bikes vs. background, people vs. background, and cars vs. background. Table 1 lists each class along with the number of images in that class, and the total number of images overall.

Bikes	365
Cars	420
People	311
Background	380
TOTAL	**1476**

Table 1. Number of images in each class in the GRAZ02 dataset.

According to [7] the average ratio of object size to image size counted in number of pixels is 0.22 for bikes, 0.17 for people, and 0.09 for cars.

3.2 Feature Extraction

Each image was scaled to 320 x 240 to speed up processing, and converted to the Ohta color space. The values of I_1, I_2 and I_3 were each scaled to range from 0 to 255 inclusive. The following global statistics were then calculated for each image and for each Ohta color plane I_p (p=1..3):

- The mean, median, mode, minimum, and maximum values of I_p.
- The standard deviation, skewness, and kurtosis of I_p.
- A normalized histogram consisting of 16 bins, in which bin 1 contains the frequency of pixels with I_p values of 0..15 inclusive, bin 2 contains the frequency of pixels with I_p values of 16..31 inclusive, and so on.

This produced a total of 24 numeric features for each color plane, or a total of 72 features related to color overall.

In addition to these features, 17 further features related to LBPs, namely the normalized frequencies of each of the 17 different LBPs in the image, were added to the feature set giving a total of 89 features per image.

3.3 Classifier Implementation & Parameters

The state-of-the-art classifier implementations used are those found in the WEKA machine learning workbench [10], version 3.4.3. The first three classifiers (namely, SVMs, Random Forests and

Boosted Decision Trees) were trained individually, while the Voting, Stacking, and Grading classifiers combined all three of those individual classifiers into a single ensemble classifier.

In all cases, the default parameters were used with the following exceptions: the WEKA implementations of SVMs, namely Sequential Minimal Optimization (SMO), had a complexity parameter of 2.0 and radial basis function kernel with a gamma parameter of 0.5. These parameters were chosen from informal trial-and-error experiments on the Bikes vs. Backgrounds dataset prior to running the experiment, and appeared to be the best parameter choices. The number of decisions trees in the Random Forests classifier was set to 200. For the Boosted Decision Trees classifier, the boosting algorithm utilized was AdaBoostM1 with unpruned J48 decision trees as the base classifier, and there were 200 iterations. The stacking classifier used M5P, a regression algorithm, as the meta-classifier, and the grading classifier used an SVM as the meta classifier with the same parameters as specified above.

3.4 Experiments

There are a total of three binary classification datasets (Bikes vs. Backgrounds, Cars vs. Backgrounds, and People vs. Backgrounds) in the GRAZ02 dataset, and each of the six classifiers were trained with varying amounts (10%, 50% and 90% respectively) of randomly selected training data. This gives 36 different combinations of dataset, amount of training data, and classifier. All images not selected for the training split were put into the test split, so the entire GRAZ02 dataset was always used in every experiment.

For the 10% training data experiments, 10% of the images were selected randomly with the remainder used for testing. This was repeated 20 times. For the 50% training data experiments, stratified 5 x 2 fold cross validation was used. Each cross validation selected 50% of the dataset for training and tested the classifiers on the remaining 50%; the test and training sets were then exchanged and the classifiers retrained and retested. This process was repeated 5 times. Finally, for the 90% training data situation, stratified 1 x 10 fold cross validation was performed, with the dataset divided into ten randomly selected, equally sized subsets, with each subset being used in turn for testing after the classifiers were trained on the remaining nine subsets.

4 Results

The primary measure used to record classifier performance in these object detection experiments is a statistic known as the Area Under the ROC Curve (AUC).

Briefly, given a test instance, most classifiers produce a probability estimate between 0 and 1 that the instance belongs to the positive class. However, the classifier ultimately has to make a binary decision about the test instance: it is either positive or negative. In most cases, the default strategy is to classify a test instance as positive if the probability of it being positive is greater than or equal to 0.5. However, this default strategy can often lead to suboptimal performance. What happens, for example, if alternative thresholds such as 0.4 or 0.8 give better performances? The AUC is a measure of classifier performance that is independent of the threshold: it summarizes not the accuracy, but how the true positive and false positive rate change as the threshold gradually increases from 0.0 to 1.0. An ideal, perfect, classifier has an AUC value 1.0 while a random classifier has an AUC of 0.5.

Statistical significance comparisons between the AUCs of the classifiers on each dataset will be discussed, and the accuracy of the best classifier at the Equal Error Rate (EER) on all three datasets will be reported.

4.1 Mean AUC Performance

Tables 2, 3, and 4 give the mean AUC values across all runs to 2 decimal places for each of the classifier and training data amount combinations, for the bikes, cars and people datasets respectively.

	SVM	RF	Boost	Vote	Stack	Grade
10%	0.82	**0.86**	0.81	0.85	0.85	0.77
50%	0.90	0.90	0.89	**0.91**	**0.91**	0.83
90%	0.91	0.91	0.90	**0.92**	**0.92**	0.84

Table 2. Mean AUC performance of six classifiers on the Bikes vs. Backgrounds dataset, by amount of training data.

	SVM	RF	Boost	Vote	Stack	Grade
10%	0.73	**0.79**	0.75	0.77	0.78	0.71
50%	0.80	**0.85**	0.82	**0.85**	0.84	0.77
90%	0.82	0.85	0.83	**0.86**	0.85	0.78

Table 3. Mean AUC performance of six classifiers on the Cars vs. Backgrounds dataset, by amount of training data.

	SVM	RF	Boost	Vote	Stack	Grade
10%	0.80	**0.84**	0.77	**0.84**	0.83	0.75
50%	0.86	**0.88**	0.84	**0.88**	**0.88**	0.80
90%	0.89	**0.90**	0.86	**0.90**	**0.90**	0.83

Table 4. Mean AUC performance of six classifiers on the People vs. Backgrounds dataset, by amount of training data.

It can be seen that the highest AUC achieved is 0.92 on the bikes dataset. Classifiers trained on the people dataset reach an AUC of 0.90, while the cars dataset appears to be the most difficult, with classifiers trained on it achieving a maximum AUC of only 0.86.

With respect to the amount of training data, there is a large difference in AUCs between 10% and 50% training data, but much less of a difference between

50% and 90% training data. For 10% training data, the Random Forests classifier is consistently the best (e.g. compare 0.86 AUC using Random Forests on the bikes dataset to 0.82 using SVMs). This suggests that if a smaller number of images is available (in the order of about 30-40), then Random Forests is likely to be the best classifier. For larger amounts of training data, the Random Forests classifier is sometimes slightly worse, sometimes equal to the ensemble classifiers Voting and Stacking. The performance of SVMs improves dramatically with the amount of training data.

4.2 Statistical Significance Comparisons

The AUC values for the best classifiers such as Random Forests, Voting and Stacking, appear quite close, and therefore to perform a finer comparison, the performance of each pair of classifiers on each dataset and amount of training data combination was tested for a statistically significance difference. The test was at 5% level using the corrected paired T-tester available in WEKA 3.4.3.

	SVMs	RFs	Boost	Stack	Vote	Grade
SVMs		-/L/-	-/-/W	L/L/-	L/L/L	W/-/W
RFs	-/W/-		-/-/W	-/-/-	-/-/-	W/W/W
Boosting	-/-/L	-/-/L		L/L/L	L/-/L	W/W/W
Stacking	W/W/-	-/-/-	W/W/W		-/-/-	W/W/W
Voting	W/W/W	-/-/-	W/-/W	-/-/-		W/W/W
Grading	L/-/L	L/L/L	L/L/L	L/L/L	L/L/L	

Table 5. Statistical significance comparison at 5% significance of each classifier against every other classifier with 50% training data.

	SVMs	RFs	Boost	Stack	Vote	Grade
SVMs		-/-/-	-/-/-	-/-/-	L/L/L	W/W/W
RFs	-/-/-		-/-/W	-/-/-	-/-/-	W/W/W
Boosting	-/-/-	-/-/L		-/L/L	-/L/L	W/W/W
Stacking	-/-/-	-/-/-	-/W/W		-/-/-	W/W/W
Voting	W/W/W	-/-/-	-/-/W	-/-/-		W/W/W
Grading	L/L/L	L/L/L	L/L/L	L/L/L	L/L/L	

Table 6. Statistical significance comparison at 5% significance of each classifier against every other classifier with 90% training data.

Interestingly, no significant difference was found between the AUC values of classifiers trained using only 10% of the training data. When the reason for this was investigated, it was found that the standard deviation of the AUC values was 0.06 – quite a high variation. On the other hand, for the classifiers trained using 50% and 90% of the training data, the AUC standard deviation is between 0.01 and 0.02, which is a much more acceptable value enabling statistical significance comparisons.

Tables 5 and 6 present the results of the statistical significance comparisons for 50% and 90% training data amounts respectively. The tables must be read as follows: if *X* is a classifier labeling the row, and *Y* a classifier labeling the column, then the table entry indicates which datasets *X* wins over, loses to, or draws with *Y* on. The order of the datasets is: bikes, cars, people.

	Classifier	Wins	Losses	Draws
50% TD	**SVMs**	3	6	6
	RFs	5	0	10
	Boosting	3	7	5
	Stacking	**8**	**0**	**7**
	Voting	**8**	**0**	**7**
	Grading	0	1	14
90% TD	**SVMs**	3	3	9
	RFs	4	0	11
	Boosting	3	5	7
	Stacking	**5**	**0**	**10**
	Voting	**7**	**0**	**8**
	Grading	0	0	15

Table 7. Summary of statistical significance comparisons at 5% significance between all pairs of classifiers on the 50% and 90% training data runs.

For example, the entry for Stacking and SVMs in Table 5 is "W/W/-" which indicates that Stacking significantly outperforms SVMs on the bikes and cars datasets, but there is no difference on the people dataset (with 50% training data). Similarly, the entry "-/L/L" in Table 6 for Boosting (on the row) and Voting (on the column) indicates that Boosting has a significantly lower AUC than Voting on the cars and people datasets, but there is no difference on the bikes dataset.

Given the analysis in Tables 5 and 6, it is possible to count the number of times that each classifier has a statistically significantly better AUC than the others, and this summary is presented in Table 7. This table clearly shows when the amount of training data is at the 50% level, voting and stacking are best performing classifiers. However, with an increase in training data to 90%, Voting becomes the single best classifier.

4.3 ROC Curves and Equal Error Rates for Voting

Figure 3 depicts the ROC curves for the Voting classifier after running a 1 x 10 fold cross validation experiment on the bikes, cars and people datasets respectively. The AUC is defined as the area under this curve, and as can be expected from Tables 2-4, the performance of Voting on the bikes dataset produces the best ROC curve, and the cars dataset is the worst.

From the ROC curves is it possible to calculate the accuracy of the Voting classifier at the Equal Error Rate (EER), which is the accuracy achieved when the threshold is set such that the false positive rate equals the false negative rate (rather than a default value such as 0.5). Like AUC, the EER accuracy is a threshold-independent means of reporting classifier performance. Table 8 lists the mean EER accuracies

for Voting on each of the binary classification tasks in the GRAZ02 dataset.

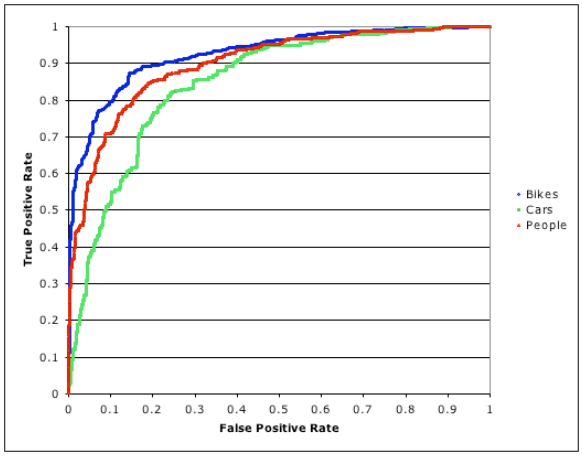

Figure 3. ROC curve for Voting after a 1 x 10 fold cross validation experiment on the Bikes, Cars and People datasets.

	EER Accuracy
Bikes	**85.75%**
Cars	**78.15%**
People	**82.65%**

Table 8. Accuracy of the Voting classifier at the EER for Bikes, Cars and People.

5 Comparison To Previous Work

It is instructive at this point to compare the current results with that of previous researchers who have worked on the same dataset.

Opelt & Pinz [7] proposed a method for weakly supervised image classification in which an object localization step is performed prior to training. The purpose of this step is to find the parts of the training images in which the object of interest (a bike, car, or person depending on the problem) actually appears, so that the rest of the positive image containing only background can be safely ignored. In combination with their boosting approach for classification, they achieved ROC equal error rates (in the best cases) of 76.4% for bikes, 81% for people, and 70.2% for cars.

More recently, a patch-based approach was proposed in [9]. In this approach, "interesting" patches in an image are located and then various different features are extracted from each interest point, such as grey values, multi-scale autoconvolution transforms, and Haar integral-based invariants, and these interest point features are used for classification. The best EER accuracy results achieved when 100 interest points were selected was 72.7%, 68.8% and 79.5% for bikes, cars, and people respectively.

Finally, [3] proposes a method called "saliency maps", which is a novel visual attention technique. An SVM is used to both construct the saliency maps and classify the images at the same time. The authors report EER accuracies of 79% for bikes and 71.7% for cars in the weakly supervised learning case.

6 Conclusion

The results of this research show that the Voting ensemble of an SVM classifier, a Random Forests classifier, and a Boosted Decision Trees classifier provides the best performance in terms of AUC on the GRAZ02 dataset. When compared to previous related work, these results are promising and can be used as a baseline for future research.

7 References

[1] Fergus R., Perona P. and Zisserman. A. 2003. Object Class recognition by unsupervised scale-invariant learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition vol. 2*, pp. 264-271.

[2] Jiang S., Huang Q., Ye Q., and Gao W. 2006. An effective method to detect and categorize digitized traditional Chinese paintings. *Pattern Recognition Letters* 27 pp. 734-746.

[3] Moosmann F., Larlus D., and Jurie F. 2006. Learning saliency maps for object categorization. In *ECCV International Workshop on The Representation and Use of Prior Knowledge in Vision*. Springer, 2006.

[4] Ohta Y., Kanade T., and Sakai T. 1980. Color Information for Region Segmentation. *Computer Graphics and Image Processing* 13, pp. 222-241.

[5] Ojala T., Pietikäinen M., and Mäenpää T. 2000. Grey Scale and Rotation Invariant Texture Classification with Local Binary Patterns. *LNCS 1842*, pp. 404-420.

[6] Opelt A., Fussenegger M., Pinz A. and Auer P. 2006. Generic object recognition with boosting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(3) pp. 416-431

[7] Opelt A. and Pinz A. 2005. Object localization with boosting and weak supervision for generic object recognition. In Kalvianen H. et al. (Eds.) *SCIA 2005*, LNCS 3450, pp. 862-871.

[8] Pietikäinen M., Mäenpää T and Viertola J. 2002. Color Texture Classification with Colour Histograms and Local Binary Patterns. In *Proc. 2nd International Workshop on Texture Analysis and Synthesis*, pp. 109-112.

[9] Teynor A., Rahtu E., Setia L., and Burkhardt H. 2006. Properties of patch-based approaches for the recognition of visual object classes. In Franke K. et al., *DAGM 2006*, LCNS 4174, pp. 284-293.

[10] Witten I. and Frank E. 2005. *Data Mining: Practical Machine Learning Tools and Techniques* (2nd Edition), Morgan Kauffman.