Dirty Money: Feature selection using AdaBoost

J. van Turnhout (0312649) jturnhou@science.uva.nl,
N. Raiman (0336696) nraiman@science.uva.nl,
S. L. Pintea (6109969) s.l.pintea@student.uva.nl

January 23, 2010

Abstract

In the month January of 2010 a project on the classification of the fitness of money was proposed. During this month we have tested and implemented various techniques to handle the problem of money classification. The results from our experiment show promising development comparing to the current state of research.

In the below sections we have described the approaches that we have tried and the results obtained for each one of them.

1 Introduction

The goal of this project is to determine a reliable method that would be able to distinguish between dirty bills and clean bills. In order to achieve this goal we had to think of what are the representative features of dirt that can be found on money bills and what is the best method that can be used to model this.

Throughout this project we have tried a number of different approaches in order to gain good results and in the same time we have tried to get a better understanding of what methods are fit for describing the features of the dirty money and clean money.

The main techniques used in this project are: *Eigen-faces* in combination with *PCA* and *AdaBoost*, *Haar-like features* and *Adaboost*, *Convolution with predefined kernels* and *Adaboost*, *edge-detection* and *intensity* distributions.

In section 2, related work on *Haar-like features*, *convolution with kernels*, *PCA*, *SVM*, *AdaBoost*, *edge detection* and *intensity* are discussed. The implementation of the *AdaBoost* algorithm applied on the different techniques are explained in 3. We discuss our experiments and their results in section 4. Finally, we conclude in section 5 and propose some topics for future research.

2 Background

Finding the main features of dirty money and clean money was a true challenge in this project. In this section are indicated in more detail the ups and downs of the techniques used and the theoretical knowledge that represents the basis of these methods.

2.1 PCA and Adaboost

The *AdaBoost* algorithm is used in combination with different techniques throughout this project, such as: *PCA*, *Haar-like features* or *edge* and *intensity* distributions over different regions.

Adaptive Boosting (also known as AdaBoost) is a machine learning technique which can be used in conjunction with various other learning algorithms. The idea is to have a (convex) set of weak classifiers (classifiers that perform at least better than random) and then minimize the total error over the training-set by finding the best classifier at each stage of the algorithm.

The theoretical basis of this algorithm is that given a set of models (or features), *M*, the algorithm will determine the subset of

T models that are the best for distinguishing between the two classes (fit and unfit bills). Thus, *AdaBoost* will learn the most representative features of the two classes. The algorithm for determining the best models is shown in section 3: *Algorithm 1*.

Another very important characteristic of this algorithm is that it also specifies a method in which the models that were chosen as being the best, can be combined in order to give a strong classifier. The corresponding algorithm can be seen in section 3: *Algorithm 2*.

PCA !!!

2.2 Intensity and Edge distributions

Intensity & Edge!!!

2.3 Haar-like features and Convolution over kernels

The first idea that we have tried was to implement the *Viola & Jones* approach for object detection. The final cascade used in this paper was not implemented because the main idea of this cascade was not suitable for the purpose of this project. We have used the strong classifier computed in *AdaBoost* as the final output.

The first step of the algorithm is defining the

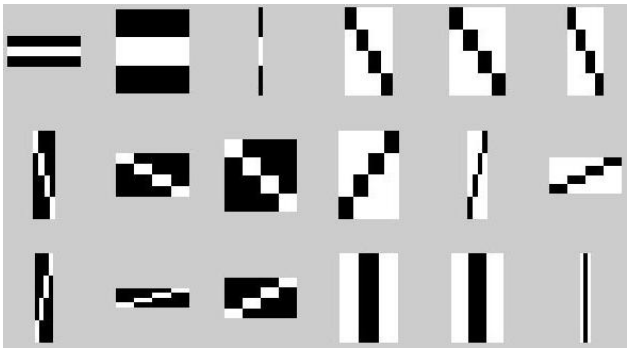

Figure 1: Front and Rear images

patterns, that are mainly matrixes of different dimensions containing low and high values for intensity (-1 and 1) – Figure1 indicates a subset of the patterns used. The *Haar-features* algorithm was designed to loop over all predefined patterns on each position in the image and convolve the specific region of the image with the patterns using the formula:

$$Value = \sum_x \sum_y Image(y : y+h, x : x+w).*Pattern,$$

where: h – the heigh of the pattern
w – the width of the pattern

The resulted values for each pattern and location in the image would, then, be used in *AdaBoost* to train an *SVM* classifier. Taking into account the fact that the set of features generated by the algorithm for each pattern and each image, was extremely large, and the training took too much time, we have decided to use instead just random locations at which to convolve the patterns with a region of the image that would have the same size. Figure 2 indicates what *AdaBoost* would choose as being the most representative 5 features for the front side and rear side of the bills.

The results obtained using *Haar features* were

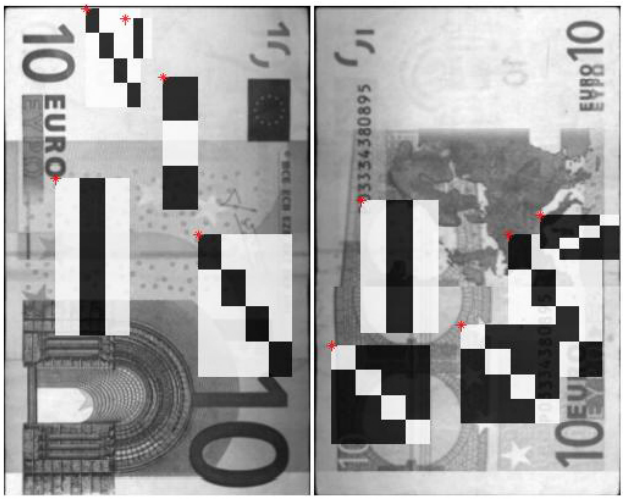

Figure 2: Front and Rear images

not as good as we would have expected. An explanation for this may be the fact that the patterns defined were not entirely able to model the dirt present on the bills.

The second approach that we have tried was to define a set of patterns and to segment each image into smaller regions that would be, then, convolved with the patterns. Thus, the resulted set of models would represent the input features used in *AdaBoost*. For establishing the set of the best T features we have tried using both *SVM* and *Gaussian distribution*, and the results retrieved by the first one seemed slightly better than the ones obtained while using *Gaussinas*.

3 Implementation

Algorithm 1 AdaBoost learning features

function AdaBoostLearn(T, M, S)
T = nr. of hypothesis
M = Models
S = training-set, $\{(x_1, y_1), ...(x_n, y_n)\}$ with $x_i \epsilon X$ and $y_i \epsilon \{-1, 1\}$
$D_{1(i)} \leftarrow \frac{1}{n}$, with $i = 1, ..., n$
for $t = 1$ to T **do**
    $error_t \leftarrow 0$
    for $m \epsilon M$ **do**
        $h_j(x_i) \leftarrow predict(x_i)$ *%svm or gaussian distribution*
        $error_j \leftarrow \sum_{i=1}^{n} D_t(i)[y_i \neq h_j(x_i)]$
        if $error_j < error_t$ **then**
            $error_t \leftarrow error_j$
            $h_t \leftarrow h_j$
    $\alpha_t \leftarrow 0.5 \cdot \log \frac{1-error_t}{error_t}$
    for $i = 1$ to n **do**
        $D_{t+1}(i) \leftarrow \frac{D_t(i) \exp -\alpha_t \cdot y_i \cdot h_t(x_i)}{Z_t}$
return α, h

Algorithm 2 AdaBoost Prediction

function AdaBoostPredict(α, h, I)
α = weights
h = weak classifiers
I = image
p = prediction
for $t = 1$ to $length(\alpha)$ **do**
    $p \leftarrow p + \alpha_t h_t(I)$
return $sign(p)$

4 Results

5 Conclusion

References

[1] P. Viola & M. Jones:
Rapid Object Detection using a Boosted Cascade of Simple Features (CVPR 2001)

[2] AdaBoost:
http://en.wikipedia.org/wiki/AdaBoost