Classifying the head-shoulder region and orientation in pedestrians

November 8, 2007

Sanne Korzec, skorzec@science.uva.nl

Informatics Institute, University of Amsterdam

Abstract

This research is an attempt to create an algorithm to classify over images to recognize a human in any orientation towards the camera, to handle occlusion to some degree and to classify the orientation of a human relative to the camera. We want to answer the following questions: Can we implement such a strict system that performs well? If not, where are the weak points? The train set contained around ten thousand similar images of pedestrians facing a direction. The test set were pictures of another domain. We used Haar-like features to train the detectors and a sliding window evaluator to test the detectors. Only the Right single classifier outperforms the general one for lower percentage overlap. For higher percentage overlap, the results become less clear. We conclude that the main contribution to the poor results can be attributed to the difference in the DM train set and the Cassandra test set.

1 Introduction

Surveillance cameras are widely used in buildings and public areas for security. As the use of cameras comes ever more common, it still is tedious work for the operators to keep focused twenty four hours per day to detect suspicious activities like human aggression. The amount of time the videos show any unusual behavior is relatively low in relation to the amount of video data. To aid operators, automated methods are interesting replacements to register these activities for them and to aid them in which items are of more interest than others. The university of Amsterdam (UvA) and the university of Groningen (RUG) are working on Cassandra [1]; an audio- and video-based system for detecting aggression. For the system to be able to detect aggression, it needs a correct representation and classification of humans in a frame of video material. Though there are some methods that work quite well in detecting humans, there are still several open issues that need to be solved. First, many algorithms can only recognize humans in one orientation towards the camera (e.g. face detection). Second, how should we handle occlusion? It is often the case that not the entire body of a human is visible. Third, in the ideal situation one would not only like to know where a human stands in a 2d representation of the world, but also what the 3d positions of his or her several body parts are. Some attempts are being made [2] to address this issue. As these problems tend to be very difficult to solve, we would like to give the 3d approach research some tools to work with. Our research is an attempt to create such a tool where we want to:

1. Classify over images to recognize a human in any orientation towards the camera.

2. Handle occlusion to some degree: We are looking exclusively for the head-shoulder region of humans.

3. Classify the orientation of a human relative to the camera.

In this paper we investigate the possibilities to implement such a system in the Cassandra domain by creating a classifier that single handedly holds these 3 properties (strict method). But as this is difficult to model, we also want to study classifiers that hold one or more of these properties. We will define a classifier general if it does not classify the orientation of a human in addition to occlusion and position. We will try to answer the following questions: Can we implement such a strict system that performs well? If not, where are the weak points? In section 2, we describe our tools and experiments. In section 3, we discuss the results. Finally in section 4, we give our opinion on how to continue with these findings.

2 Methods and Materials

In order to recognize humans in any orientation, we need to train from a data set of images containing such examples. As it is almost impossible to collect each unique orientation, we have to make some assumptions in order to accomplish the goals set in the introduction. In case of orientation we also need to make a discrete mapping. We only focus on four orientations: Facing towards the camera (front), facing away from the camera (back) and facing either way sideways (left, right). We will assume that all other orientations-position combinations are to some degree enveloped by these four.

2.1 Data sets

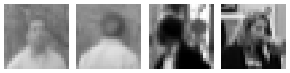

Figure 1: Daimler Chrysler data set containing: front, back, left and right

2.1.1 Daimler Chrysler Train Sets

We used five train sets of images provided by Daimler Chrysler (Figure 1). The entire data set contains around ten thousand similar images of one pedestrian per image. Four train sets contain images of the head shoulder areas of pedestrians in one of the following orientations: Left, Right, Front or Back. These train sets mainly have images of pedestrians with a non-ambiguous orientation, but also have some images of ambiguous orientations (e.g. A person who is facing to the front right: Figure 1). The last train set is the union of the four.

2.1.2 Daimler Chrysler Test Sets

The DM test sets are subsets of the same type of images from the DM train sets. Table 1 gives the number of positive (pedestrians) and negative (non-pedestrians) images of the DM train- and test sets.

	Positives(Train)	Negatives(Train)	Positives(Test)	Negatives(Test)
Front	3087	3087	100	400
Back	3000	3000	324	400
Left	1500	1500	138	400
Right	2300	2300	145	400
Combined	9887	9887	-	-

Table 1: Number of images from the Daimler Chrysler train- and test sets

2.1.3 Cassandra Test Sets

The Cassandra test sets are a collection of full images of pedestrians at a train station. These images differ substantially from the DM data sets in terms of resolution, image clarity, pedestrians and their clothes, color, etc. The test set contains ten different movies, each 100 frames thus 1000 images total.

2.1.4 Labeling Ground Truth

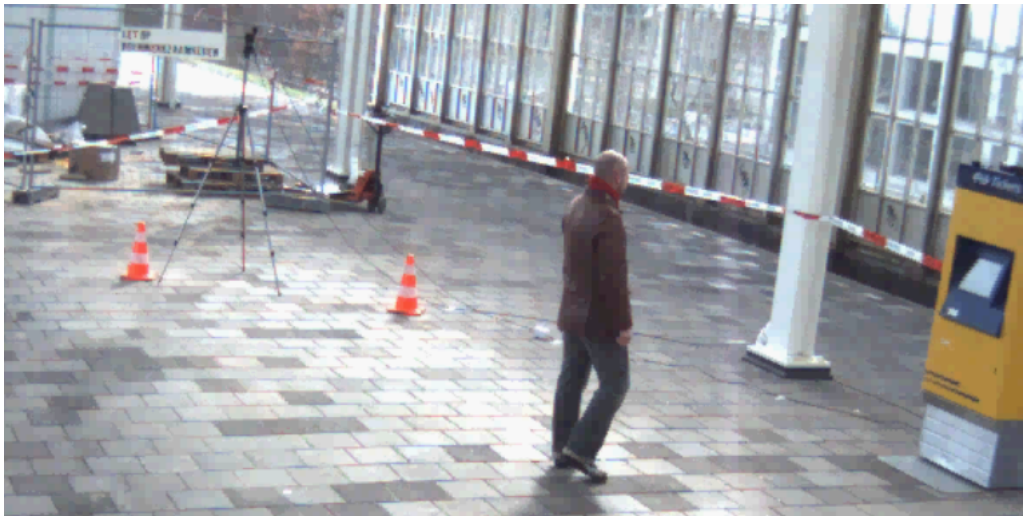

Figure 2: Marked section: Red and White Ribbon in the Cassandra test set

In the Cassandra data set we only labeled pedestrians who are inside the marked area (Figure 2). We only labeled occluded pedestrians if at least the entire head was visible, or if both shoulders were visible. We labeled pedestrians in four orientations: Front, Back, Right or Left. In regard to the orientation we have labeled pedestrians according to their shoulder orientation. This is an important assumption as people can orient both their head and their shoulders in multiple directions. There were some ambiguities as people sometimes orient themselves in an angle of about 45 degrees between frontal or sideways orientation. For those cases a choice was made by the labeler; each instance was only labeled once. 2 gives the number of actual pedestrians present in the Cassandra test set.

	pedestrians present
Front	2342
Back	547
Left	1473
Right	828
Combined	5190

Table 2: Pedestrians present in the Cassandra test set

2.2 Classifiers

We have created several classifiers using Haar-like features in the image with the well known method of cascading the classifiers [3]. When too few layers are used the cascade will create too many false positives. On the other hand if too many layers are used, the cascade will over fit. Therefore we have trained multiple layers and added them to the cascade one by one. Classifiers are labeled with a number and a string, which corresponds to the number of layers the classifier contains and the orientation it tries to classify on respectively. We will focus on creating classifiers that hold all three properties in one cascade (strict method) and then compare these to a cascade that does not include classifying over orientation in addition to occlusion and position (general method).

2.2.1 Choosing The Correct Number Of Layers

As we will use ROC graphs to represent the ratio between correct positives and false negatives, we can choose several numbers of layers to be used for the task at hand. Depending on what algorithms these classifiers will be combined with we need to choose an optimum. For some tasks this may mean, lower recall and less false positives (higher number of layers). For others, higher recall and more false positives (lower number of layers). For the current task, we have no intentions to combine the classifiers with anything yet, so we will approximate the Bayes optimal decision boundary by choosing the most upper left coordinate as most suitable classifier.

2.3 Evaluation

In order to evaluate our classifiers we have created two tests: The performance evaluator and the sliding window evaluator. The performance evaluator takes as input a 24x24 image with a 16 pixel border on all sides, making it a 56x56 image. The performance evaluator then not only classifies whether an image belongs to our data set, but also if it was rejected, at which layer this happened. The sliding window evaluator takes n x m sized images and slides a 24x24 window over the entire image and returns a hit if the classifier thinks it is a pedestrian. The performance evaluator can give a more precise overview of what is happening

Algorithm 1 Pseudo code for area calculation

```
class_area = (pt2.x - pt1.x) * (pt2.y - pt1.y)
real_area = (tp2.x - tp1.x) * (tp2.y - tp1.y)

xmin = min(tp2.x, pt2.x)
xmax = max(tp1.x, pt1.x)
overlapx = xmin - xmax

ymin = min(tp2.y, pt2.y)
ymax = max(tp1.y, pt1.y)
overlapy = ymin - ymax

if  (  overlapx < 0 OR overlapy < 0 )
      overlap_area = 0
else
      overlap_area = overlapx * overlapy

percentage = 100*area / class_area
```

inside a cascade. We will use it on the DM test set. We will use the sliding window evaluator on the Cassandra test set.

2.3.1 Area calculation

In order to decide what a correct positive is, we experiment with area overlap. We will assign a value $percentage : P \rightarrow [0, 100]$ indicating the overlap between the labeled ground truth and the detection of the classifier. To do this we need the upper left and the bottom right coordinates. TP (true point) will be our ground truth and PT (point) our detection. We can calculate the overlap as shown in Algorithm 1.

We studied the classifiers with the following overlap: 0, 10, 25, 50 75 and 90 percent. Obviously the classifiers with a lower overlap percentage will return nicer ROC graphs, but this does not mean that they perform better given the task at hand.

2.4 Tools and Statistical Analysis

Microsoft Visual C/C++ (2003/2005) was used to write and execute a C/C++ program to analyze the performance of the detectors in the train and test sets. Intel® OpenCV [4] was used to add computer vision libraries to the main program. Matlab 6.1/7.0 was used to write scripts, among others to compute areas under the curve and receiver operating characteristic analysis. Areas under the curve were compared according to the methods described by Altman [5]. PowerAssist 0.71 was used to label the images.

3 Results

Figure 3 shows the layers of each classifier. A coordinate in the graph represents a certain threshold value. Layers can perform better or worse if the threshold is altered. Figure 3 can be transformed to the full ROC graph by taking its convex hull (See the appendix: Figure 6). This shows the maximum score allowable on the DM test set. show the ROC curves of the strict classifiers on the DM test set. For both graphs we compute false positives < 1 when there where no false positives present in the test set.

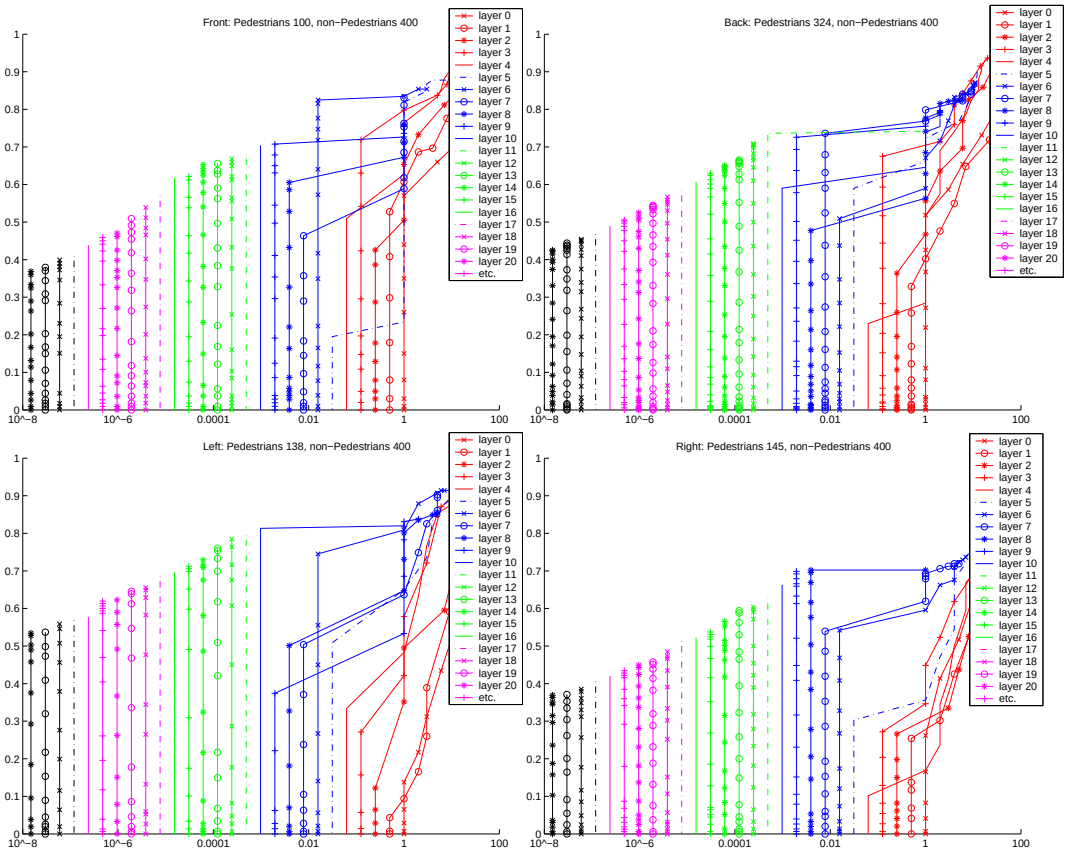

Figure 3: DM test set results. X-axis: false positives. Y-axis: Recall.

Clearly, we expect the classifiers to perform much better on the data set they were trained on. It can be mathematically be proofed that this is always the case according to the no-free-lunch theorem [6]. Figure 4 shows the ROC graphs of the general sliding window evaluation on the Cassandra test sets. The points in the graphs represent a classifier with n layers. The leftmost dot is a classifier with 16 layers and the rightmost dot is a classifier with 8 layers. The results from Figure 4 can be explained as it is understandable that using more images in the train set will improve the classifier. The combined classifier simply has seen all orientations, where the other ones only know their own orientation.

Figure 5 shows the ROC graphs of the strict sliding window evaluation on the Cassandra test sets. We would like to be able to say something about whether using a classifier as the general method will give significantly better results then classifiers using the strict method. The results show us that for the lower percentage overlaps, all classifiers except the Right classifier, perform worse than the general classifier. For the higher percent overlaps, some other classifiers also perform better than the general case.

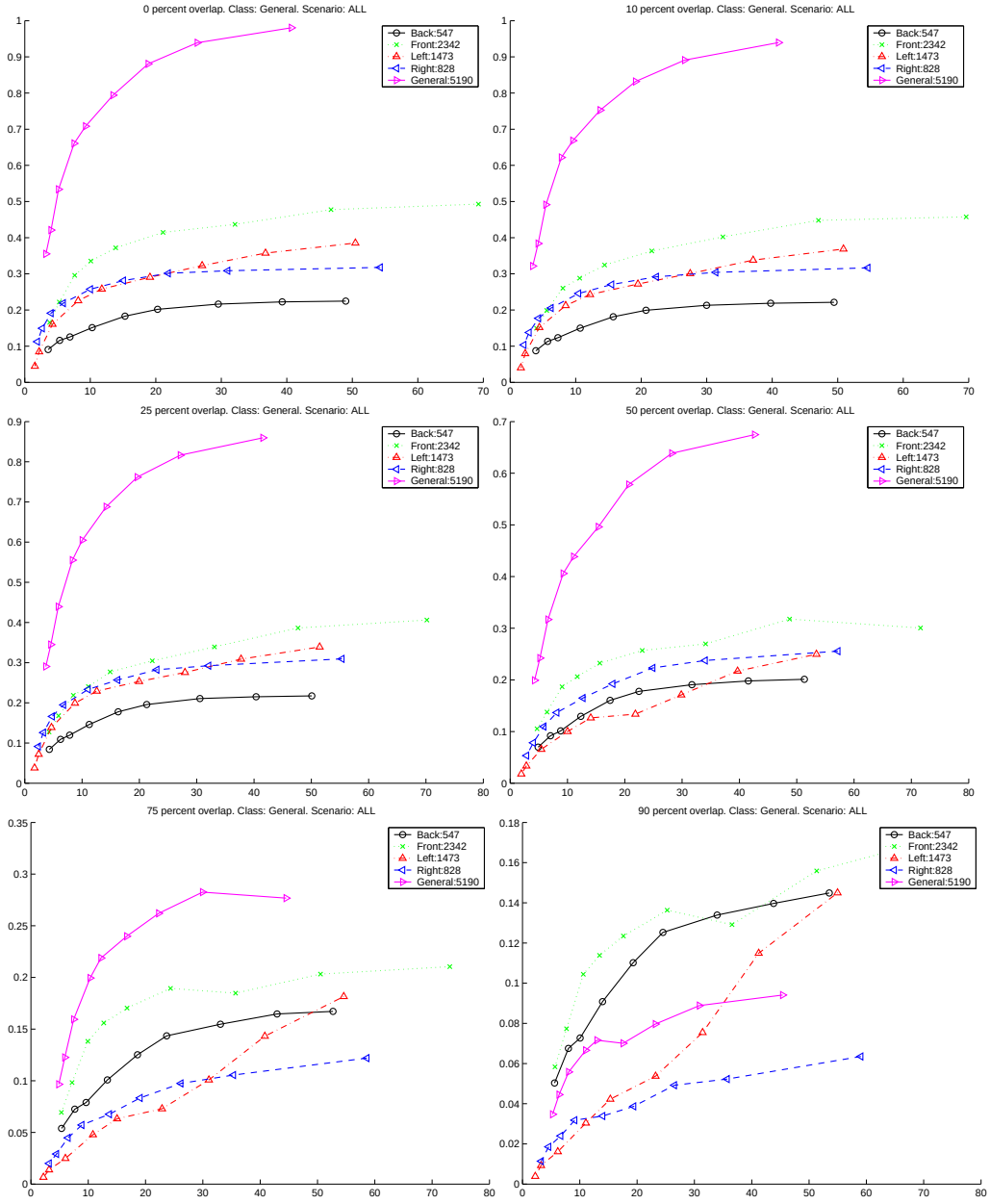

Figure 4: General sliding window evaluation: Cassandra data set. X-axis: the average number of false positives per image. Y-axis: Recall, the classifiers classify over all pedestrians, no matter their orientation.

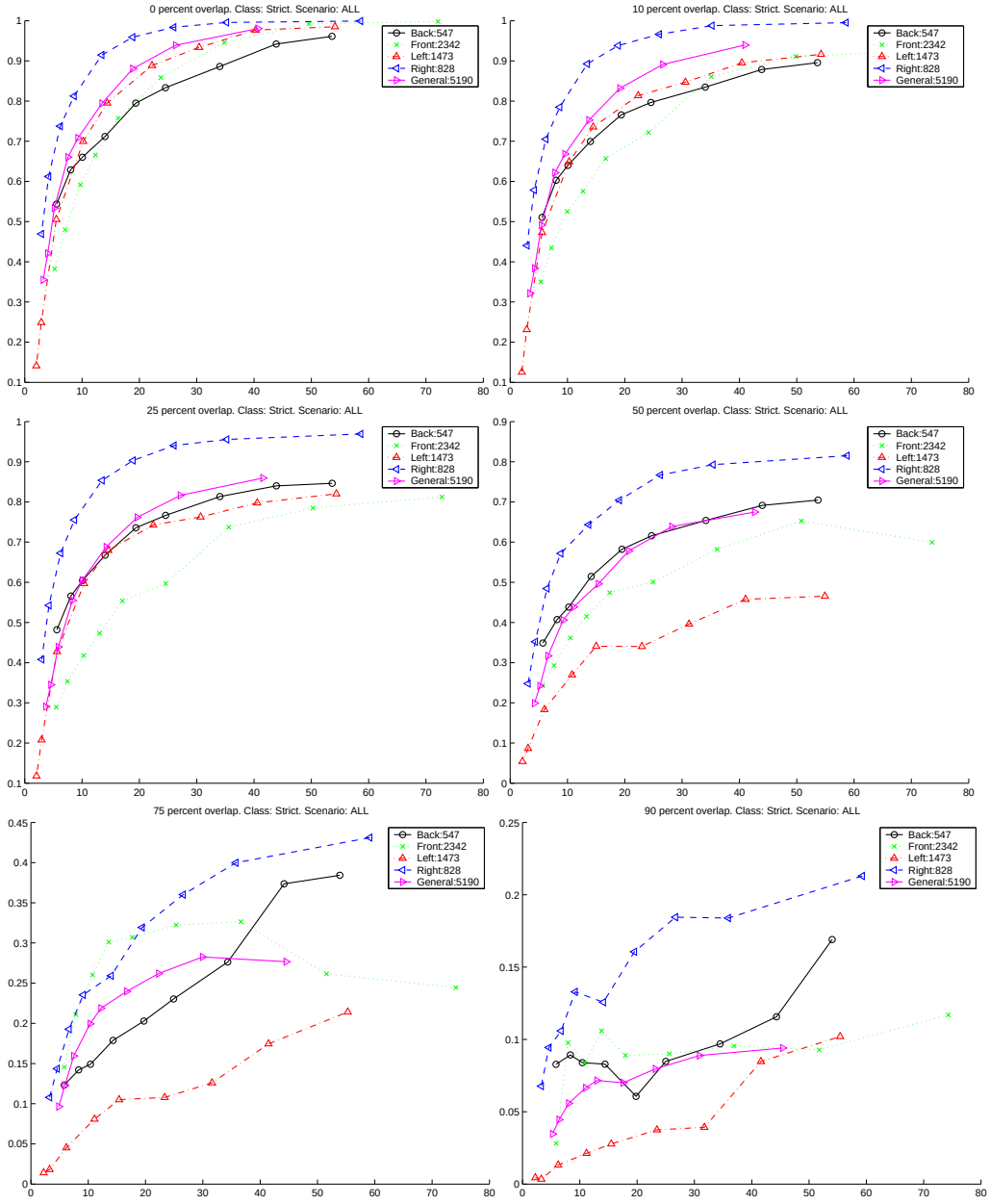

Figure 5: Strict sliding window evaluation: Cassandra data set. X-axis: the average number of false positives per image. Y-axis: Recall, the classifiers only classify over their own orientation.

4 Discussion

We have shown that for lower overlap the general classifier outperforms all other classifiers except the Right classifier. For higher overlap this distinction becomes more vague. When looking at the 50 percent overlap, the general case is exactly in between the four classifiers, but one interesting observation stands out. It seems that the classifiers that outperform the General classifier haven't nearly as many occurrences (roughly $\frac{1}{5}$th and $\frac{1}{10}$th for the Right and Back classifiers respectively). Therefore these results might be overestimated.

Another drawback is the non uniform distribution of the training set, some classifiers have seen more training images than others. This might be an explanation why some classifiers outperform others. Finally, as we train on a different data set and test on another, we might change our domain too extensively to properly perform. We can clearly see a difference between the DM test set and the Cassandra test set. Testing on the train set always gives better results, so it might be interesting to train on the target domain.

4.1 Conclusion

When we want to create a classifier that can: classify over images to recognize a human in any orientation towards the camera, handle occlusion to some degree and classify the orientation of a human relative to the camera; we conclude that it is difficult to model these properties in one classifier. The biggest loss of performance can be attributed to the difference in the DM train set and the Cassandra test set.

5 References

Websites where last accessed 2-Nov-2007.

[1] D. Gavrila. http://www.science.uva.nl/research/isla/themes/perception/cassandra
[2] M. Hofmann. http://staff.science.uva.nl/~mhofmann
[3] P. Viola, M.J. Jones. Robust real-time face detection.
International Journal of Computer Vision, 57(2):137-154, 2004.
[4] Intel Corporation. Open Source Computer Vision Library, 2006.
http://www.intel.com/technology/computing/opencv/index.htm
[5] Altman DG, Machin D, Bryant TN, Gardner MJ.
Statistics with confidence. BMJ Books 2nd edition 2001.
[6] No Free Lunch Theorem. http://www.no-free-lunch.org

6 Appendix

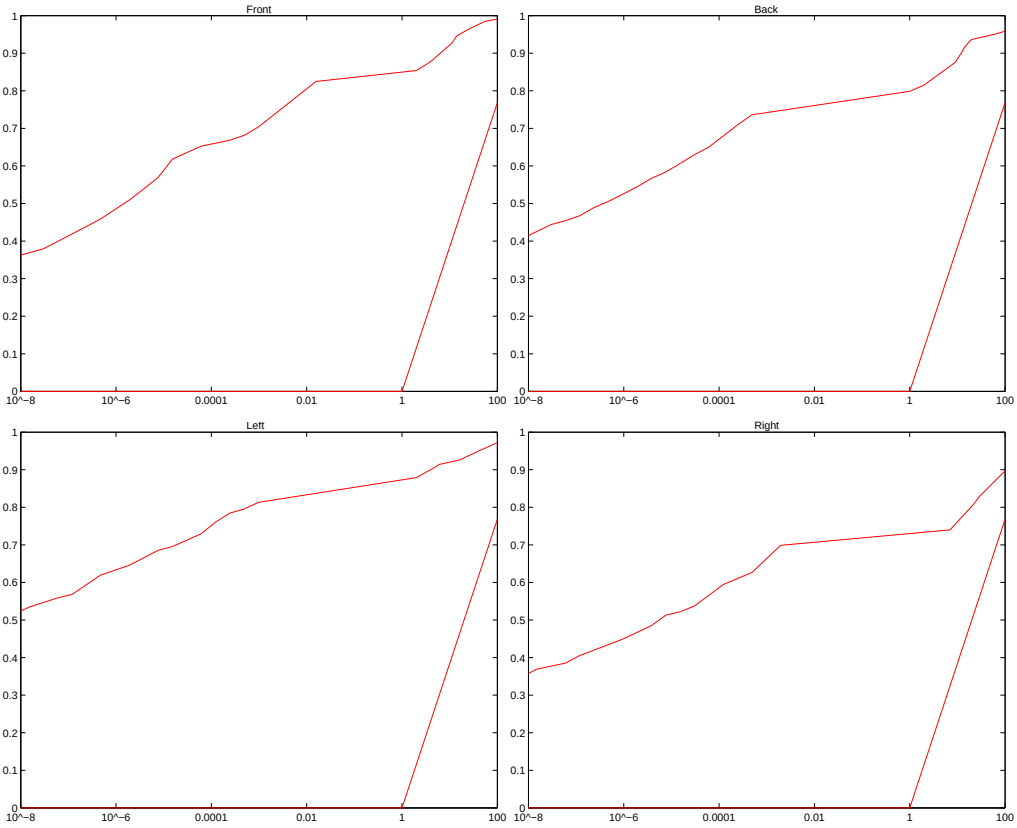

Figure 6: DM testset ROC graphs. X-axis: false positives. Y-axis: Recall.