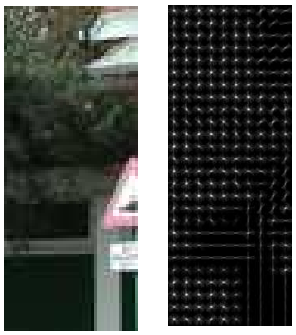
1 Persons recognition

1.1 Problem and dataset descriptions

During the project, we were first given a set images and labels indicating if there is a person. We also were given features extracted from the images and we are supposed to analyze and learn our algorithms only on those extracted features (not on images). Then, a week before the deadline, we were given a test set of features for which we should give our predictions. (Following discussions refer to the train set of features).

Our set contains 8545 images and labels, and for every image 9360 features in a form of a $26 \times 10 \times 36$ cube.

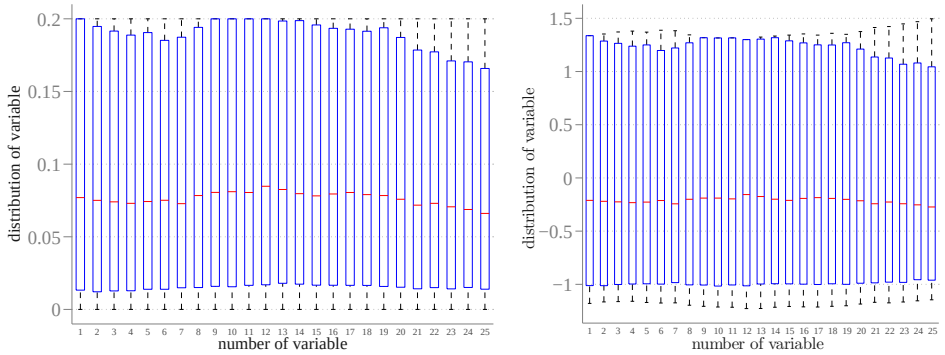


(a) picture (b) features

Figure 1: 1(a) picture No 200 and 1(b) features extracted form it, using Piotr's toolbox (http://vision.ucsd.edu/ pdollar/toolbox/doc/index.html)

1.2 Data analysis and preparation

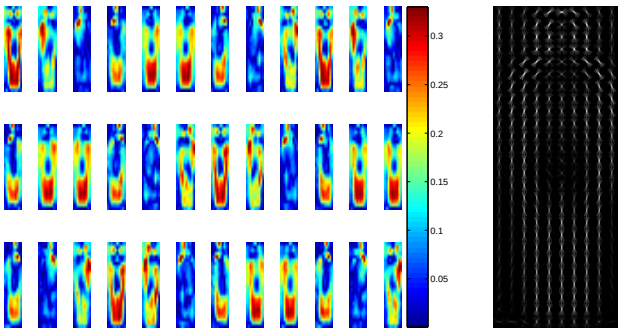
Features first two dimensions 26×10 of features for particular image correspond to the position of image form where those features were extracted and in the third dimension corresponds to the direction of edges in that part of image. White color and big number indicates that there is the edge, see 1(a) and 1(b). Values of features are between 0 and 0.2 and normalization doesn't change them much.



(a) original data (b) after normalization

Figure 2: Boxplots of part of the data before 2(a) and after 2(b) normalization. Rest of the data looks similar.

Since there are 9360 numbers for one picture using all of them may cause overfitting. We calculated correlation of them with output and we obtained picture of abstract person, see 3(b). Also on figure



(a) Absolute value of correlation for different directions of gradient

(b) positive correlation

Figure 3: Absolute value 3(a) and only positive part 3(b) of correlation, plotted as a feature. It may be good to not use the deep blue data.

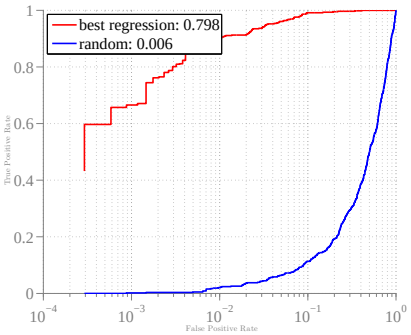
3(a) we can see that there are regions and gradient more and less important for this task and they have shape of a person. Values of correlation were between -0.37 and 0.37

2 Applied methods

We applied many methods for this task, for most of them we used existing libraries. To check which method is better we were were plotting doing cross-validation and then taking predictions into one big vector and passing it as predictions to function `evaluateMultipleMethods`, which was given to us together with data.

2.1 Penalized logistic regression

We started with the the simplest (penalized) logistic regression. We tied different penalties (λ) and different number of features. We were setting parameter $threshold$ to some value between 0 and 0.37, and taking only the part of features which are more correlated to output than $threshold$ (in absolute value). The best True Position Rate we got for this model was 0.231 for $threshold = 0.1$ and $\lambda = 0.8$. Taking more data wasn't changing TPR and changing ROC curve slightly.



(a) ROC curves for linear model and for random assignment.

Figure 4: ROC curves for different models

2.2 SVM

2.3 Neural Network

2.4 Random Forest