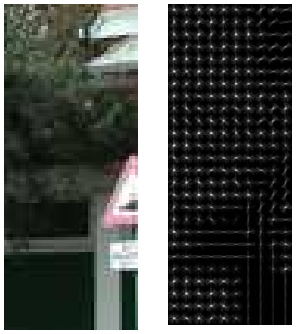
# 1 Persons recognition

## 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 ($1$ - person, $0$ - not), and for every image $9360$ features in a form of a $26 \times 10 \times 36$ cube. There are $1237$ images with a person (at least labeled $1$).

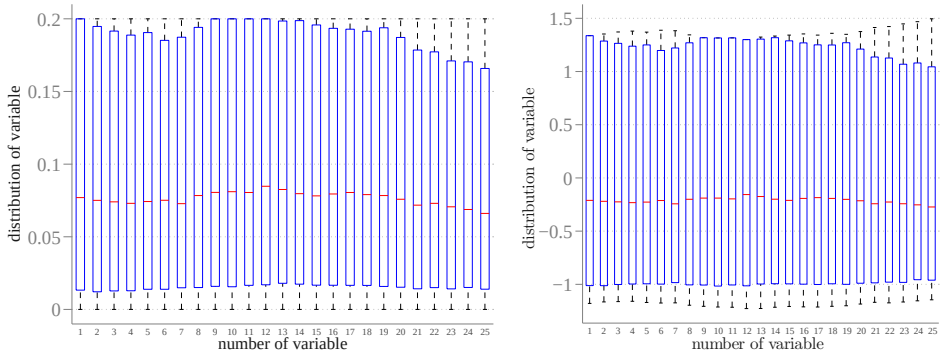


(a) picture (b) features

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

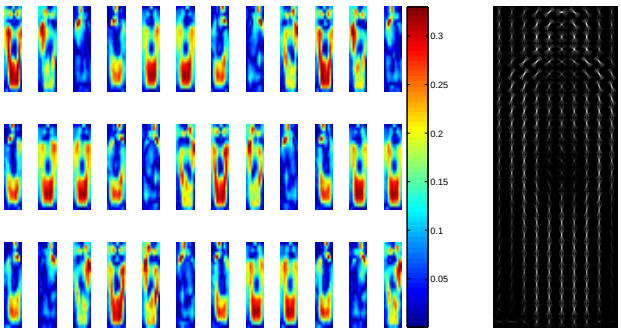
Features first two dimensions $26 \times 10$ of features for particular image correspond to the position of image form where those features1 were extracted and in the third dimension corresponds to the direction of edges in that part of image. White color and big number indicates that there is the edge, 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 will cause overfitting (there is more features than pictures). We calculated correlation of them with output and we obtained picture of




(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. There is a lot of not correlated data (deep blue).

abstract person, see 3(b). Also on figure 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 $0.37$

Other way of shrinking amount of features is singular value decomposition (i.e. taking only some vectors corresponding to the biggest singular values). That is probably better idea, because correlation captures only linear dependence and don't see structure in the data. We did $SVD$ for $1000$ singular values, and then used $8545 \times 1000$ as our new feature matrix for testing different methods.

## 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 doing cross-validation and then taking predictions both for train and test into one big vector and passing it as predictions to function `evaluateMultipleMethods`, which was given to us together with data. That was giving us estimate how big is estimate of particular method.

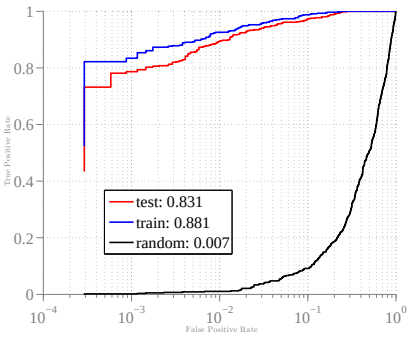
### 2.1 Penalized logistic regression

We started with the the simplest (penalized) logistic regression. Not surprisingly, this method has quite big bias. What's interesting, when we were using $1000$ features (from $SVD$) it also had big variance. Setting bigger penalty ($\lambda$) caused even bigger bias and very poor performance. Decreasing number of features helped decreasing variance. Penalized logistic regression was working the best with only about $120$ features and $\lambda = 0.1$, see 4(a).

### 2.2 SVM

### 2.3 Neural Network

### 2.4 Random Forest




(a) ROC curves for penalized logistic regression (test and train data) compared with random assignment.

Figure 4: ROC curves...