Traffic Sign Detection in Colour Images

Marvin Klaus[1], Daniela Schacherer[2]
[1]*Heidelberg University, M.Sc. Applied Computer Science*
[2]*Heidelberg University, M.Sc. Applied Computer Science*

Problem Definition

Our project compares two state-of-the art approaches towards the detection of traffic signs in color images. The first approach uses a linear classifier, more precisely a support vector machine (SVM), based on Histogram of oriented gradients (HOG) features. The second approach employs a region based convolutional neural network (R-CNN). Our training set was made available by the University of Bochum in the context of a traffic sign detection competition in 2013 [**Houben-IJCNN-2013**]. It comprises 900 images (1360 × 800 pixels) containing 1206 traffic signs. In addition, the image sections containing only the traffic signs and a CSV file containing ground truth information (location of the traffic signs within the images) are provided.

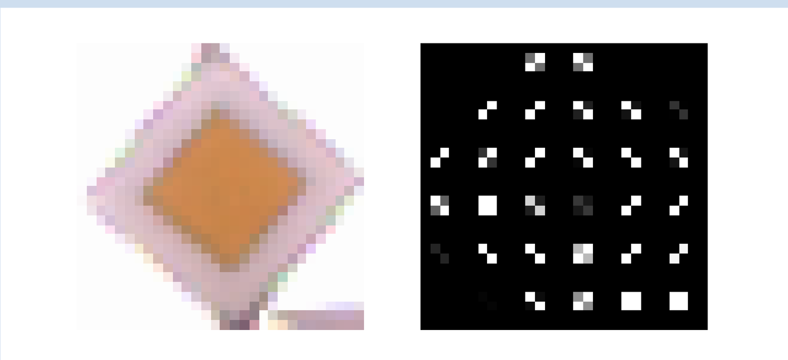
Approach 1: SVM based on HOG features

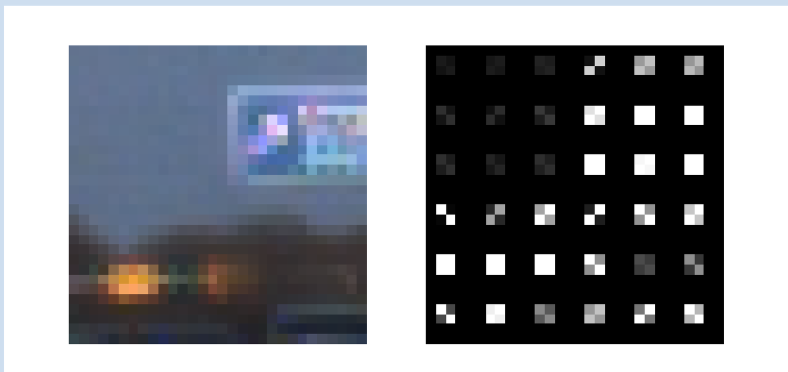
Data Preprocessing

- Preparation of positive image patches (30 × 30 pixels) showing traffic signs.
- Preparation of negative image patches (30 × 30 pixels) showing something that is not a sign.

Feature extraction

- Extraction of HOG features from the positive and negative image patches using a HOG implementation from scikit [**scikithog**].





Training

- Fit the SVM to the prepared training data.
- Hard negative mining: apply the trained SVM to some of the training images - obtained with a sliding window of 30 × 30 pixels size and a step size of 10 pixels at different scales. Each falsely detected patch is taken and explicitly added as a negative example to the training set.
- Fit the SVM using the enlarged training data.

Detection of traffic signs in an image

- Slide a window of 30 × 30 pixels size with a predefined step size (here: 5 pixels) across the image at different scales and obtain a prediction for every image section from the SVM.

Results SVM

Performance Measure: Each predicted bounding box P is compared against every ground-truth G using the Jaccard similarity (intersection over union):

$$S(P,G) = \frac{|P \cap G|}{|P \cup G|}$$

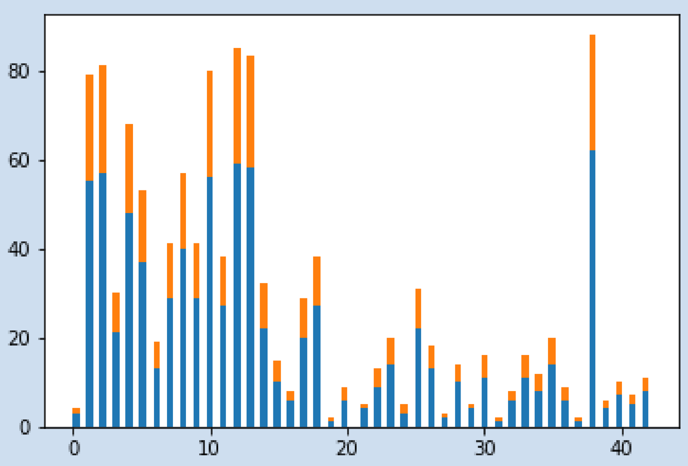
When $S \geq 0.6$, the ground truth sign is considered as detected. Finally we record the fraction of detected ground truth signs in all images.

Result: The amount of correctly found traffic signs on 30 images from the train respectively test set obtained with and without HNM is shown in the following table.

	train acc	test acc
w/o HNM	93.05, 95.0%	89.17, 99.17%
w HNM	85.56, 86.39%	79.17, 91.67 %

Approach 2: R-CNN

Data Preprocessing

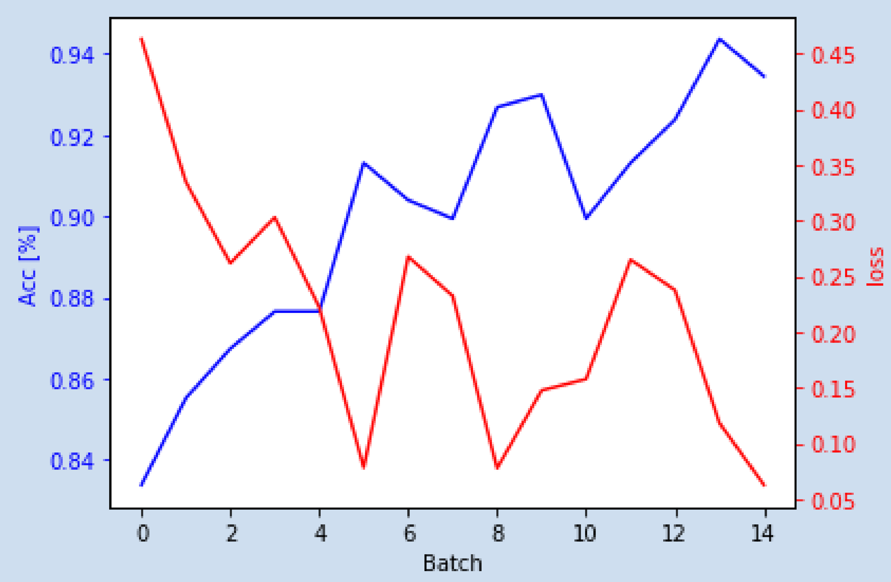


- Create negative image patches by taking samples from traffic scenes without signs
- Create train and test dataloader with following partition

Network

- Loss function: Cross Entropy Loss
- Optimizer: Adam

Results R-CNN



- used two convolution layers with ReLU activation function followed by fully connected layers.
- Selective Search to find interesting regions
  - parameter: scale=250 and sigma=0.9
- Run network with every region to get signs
- Use $S > 0.5$ since the regions are selected with the selective search algorithm
- Since the selective search algorithm often couldn't find the correct regions and the network had issues to work with the found regions, the accuracy is just 32% for traffic scene images

Comparison

Comparison

The network of course is much faster when it comes to classify if something is a traffic sign or not. The downside is that the selective search algorithm takes much time to define interesting regions. Since we have to use a sliding window in the HOG approach it takes some time to find the regions of traffic signs.

From accuracy perspective the HOG approach gives convincing results for finding traffic signs. The selective search algorithm couldn't always find all traffic signs in the images and the network had some issues to work with the regions found by the selective search algorithm which explains the surprisingly bad performance.