Ruprecht-Karls-University Heidelberg

Faculty of Mathematics and Computer Science

Project: Object Recognition and Image Understanding

Traffic Sign Detection in Colour Images

TWO APPROACHES IN COMPARISON

Author:
Daniela Schacherer, 3165890,
M.Sc. Angewandte Informatik
Marvin Klaus, 3486809, M.Sc.
Angewandte Informatik
Lecturer:
Prof. Björn Ommer
Date
24.07.2018

Contents

1 Introduction **1**

2 Dataset **1**

3 First approach: Support Vector Machine (Daniela) **2**

3.1 Data preprocessing . . . 2

3.2 Feature extraction using HOG . . . 3

3.2.1 Theory HOG . . . 3

3.2.2 Application . . . 3

3.3 Classification using SVM . . . 4

3.3.1 Theory SVM . . . 4

3.3.2 Training . . . 4

3.3.3 Detection . . . 5

3.3.4 Performance Measure . . . 5

4 Second Approach: RNN (Marvin) **6**

5 Results **7**

5.1 First Approach (Daniela) . . . 7

5.2 Second Approach (Marvin) . . . 9

6 Discussion **10**

7 Appendix **10**

7.1 Appendix A . . . 10

7.2 Appendix B: Requirements . . . 11

8 References **12**

1 Introduction

Autonomous driving is nowadays one of the most growing research fields. In this context visual detection and recognition of road signs becomes particularly important. The project presented in this report aims at the exploration of two state-of-the-art approaches towards the detection of traffic signs in color images. In contrast to traffic sign recognition, detection aims merely at locating the traffic signs in the images, however, not at classifying them.

2 Dataset

The dataset employed for this task was made publicly available by the University of Bochum as part of a traffic sign detection competition in 2013 [4]. The data set comprises 900 images (1360 × 800 pixels) in PPM format containing 1206 traffic signs. Additionally, 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. Each image contains zero to six of those traffic signs included in the competition like, for example, speed limit or stop signs. For a complete list of the competition relevant categories see Appendix A. The sizes of the traffic signs in the images vary from 16 to 128 pixels w.r.t the longer edge [3].
The images were collected such that different street scenes as urban, rural, or highway as well as different lightning and weather conditions are equally represented in the dataset. Figure 1 shows some examples contained in the data set. [3].

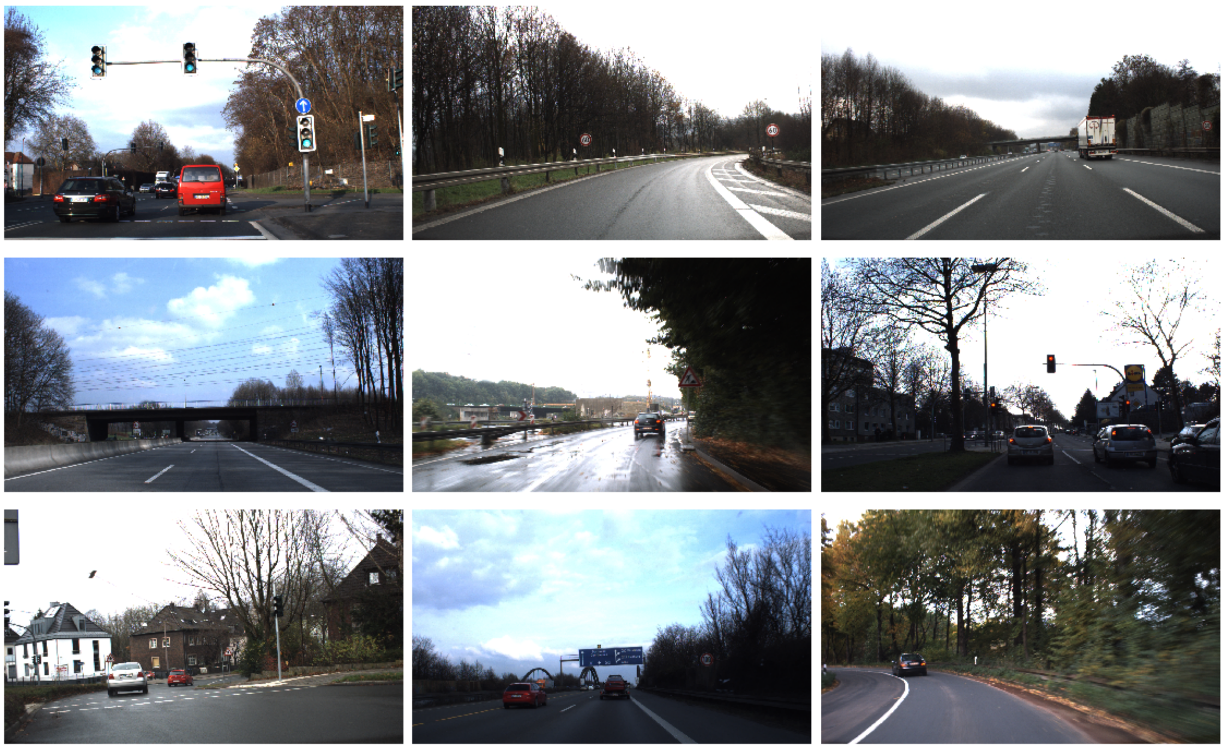


Figure 1: Example images from the dataset.

3 First approach: Support Vector Machine (Daniela)

In the first approach we use a linear support vector machine (SVM) based on Histogram of oriented gradient (HOG) features to detect traffic signs in colour images. This combination is often used for object detection since HOG descriptors were introduced by Dalal and Triggs in 2005 [?]. First, the SVM is trained to distinguish image patches showing traffic signs from image patches showing something else. For detection of road signs in a given image, a window is slided across the image at different scales employing the SVM for every image section.

3.1 Data preprocessing

The training set was randomly divided into 600 images for training and 300 images for testing.

Positive training samples
All traffic signs were extracted from the training set images using the ground truth locations provided. As the ground truth bounding boxes are not necessarily quadratic, they were converted to a quadratic shape by taking the larger side length and enlarging the smaller one equally in both directions. The extracted patches were then resized to a size of 30×30 pixels.

Negative training samples
The easiest way to generate a ”starter” set of negative samples was to take some images from the training set which do not show any sign. 30×30 pixel sized patches were extracted from these images using a stepping window. A stepping window was preferred over a sliding one in order to avoid the negative samples being statistically dependent on each other. For a start, we sampled 5000 negative image patches.

Some positive patches showing a traffic sign, as well as some of the negative patches, showing something else, are depicted in Figure 2.

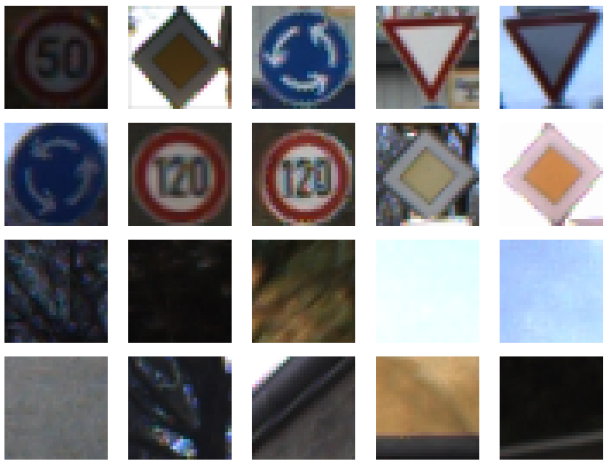


Figure 2: The first and second row show positive patches containing traffic signs, the third and fourth row show negative patches without a traffic sign contained. The size of all patches is 30 × 30 pixels.

3.2 Feature extraction using HOG

3.2.1 Theory HOG

The **H**istogram of **O**riented **G**radients (HOG) is a feature descriptor widely used in the context of object detection. This method basically summarizes gradient orientations in localized sections of an image by performing the following five steps [**?**]:

1. Global image normalization using gamma compression: thereby one can compensate for variations in illumination or local shadowing effects.
2. Computation of first order image gradients.
3. Computation of gradient histograms: in so-called cells of predefined size (e.g. 5×5 pixels) gradient orientations over all pixels in the cell are accumulated in a histogram.
4. Normalization across blocks: as gradient strength must be locally normalized cells are grouped together into so-called blocks, on which normalization is performed. Possible methods for block normalization are, for instance, the L1-norm or the L2-norm.
5. Collection of HOG descriptors from all cells into a combined feature vector.

3.2.2 Application

We used the HOG implementation from scikit-image [2]. As proposed by Houben *et al.* who performed a similar approach, we considered 8 gradient orientations, used cells of size 5×5 pixels and a block size of 2×2 cells [3]. As block normalization method we chose the L2 normalization. The final training set used to train the SVM is then constructed from the HOG features of the positive and negative patches.

3.3 Classification using SVM

3.3.1 Theory SVM

A support vector machine is a well established supervised classification and regression method. Given a set of labeled training examples from two classes - positive and negative - a SVM training algorithm creates a model that can assign new examples to one category or the other. Formally, a SVM constructs a hyperplane by maximizing the margin between positive and negative data points (see Figure 3).

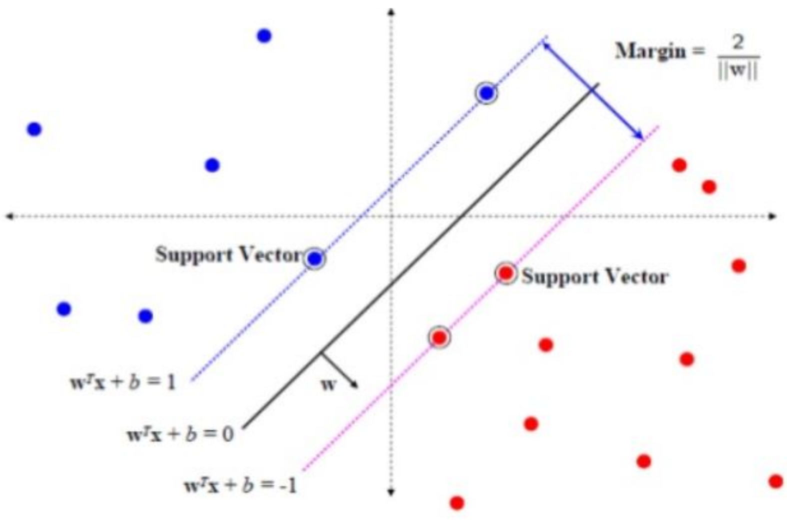


Figure 3: A Support Vector Machine tries to maximize the margin between positive and negative data points. Source: `https://www.slideshare.net/kambliruta/event-classification-prediction-using-support-vector-machine`

3.3.2 Training

We used the SVM implementation provided by sklearn [1] with the default values except for the kernel which we chose to be linear. With the intention to improve performance of the SVM as well as to obtain a sufficiently large training set, we additionally included a step of Hard Negative Mining (HNM). Therefore, the SVM is first trained with the previously constructed training set. Then, for every image and possible scale of the images dedicated for training, that contain no traffic sign, we apply the sliding window technique. At each window we compute the HOG descriptors and employ the pre-trained SVM. Each falsely detected patch is taken and explicitly added as a negative example to the training set. Finally, the SVM is retrained using the enlarged training set. In the results section we compare the outcome with and without Hard Negative Mining based on the achieved training / test accuracies.
We expect a significant improvement when training with Hard Negative Mining as we obtain

more variety in the negative training samples especially by using different scales of the images. Here we used $\sqrt(2)^i$ with $i = 0, ..., -5$ as scale factor to obtain an image pyramid with 6 layers including the original image. By the use of an image pyramid for Hard Negative Mining we can capture larger structures which might get confused easily with a traffic sign and explicitly learn the classifier to recognize them as negative examples.

3.3.3 Detection

For detection of traffic signs in a given image we slide a window of size 30×30 pixels across the image at different scales - here we also used an image pyramid with scale factor $\sqrt(2)^i$ where $i = 0, ..., -5$ - and obtain a prediction for every image section from the SVM. If the thereby received regions possibly showing a traffic sign are obtained from another than the original scale of the image these are back computed to the original image's size. As a result we finally obtain a list of predicted bounding boxes relating to the original input image.

3.3.4 Performance Measure

For a quantitative measure of the detection performance each predicted bounding box P is compared against every ground truth bounding box G by means of the Jaccard similarity, which is defined as the intersection over union of the two bounding boxes:

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

If $S(P, G) \geq 0.5$ for any predicted bounding box P, the ground truth sign G is considered as detected. In the end, we record the fraction of detected ground truth signs in all images.
In addition, a measure for the amount signs detected erroneously - i.e. the false positive rate - is needed. Therefore, we assess the fraction of predicted bounding boxes that have zero overlap with any of the ground truth bounding boxes.

4 Second Approach: RNN (Marvin)

5 Results

5.1 First Approach (Daniela)

The average detection performance on 30 randomly chosen images from the training respectively test set is reported in Table 1. Contrary to our expectations the accuracy seems to decrease when training the SVM with HNM. However, when having a look at the bounding box predictions of the SVM trained with respectively without HNM, which are presented in Figure 4, one can see that Hard Negative Mining significantly lowers the false positive rate. On the left image both ground truth bounding boxes are detected, however there is a large amount of false positives. In the right image the second ground truth bounding box is not found, but the false positive rate is much lower. Unfortunately the amount of false positives is not taken into account by the performance measure we used. Thus we nevertheless assess Hard Negative Mining as an improvement and keep applying it when training our classifier.

	train acc		**test acc**	
	performance	false positives	performance	false positives
w/o HNM	2.1%	2.1%	2.1%	2.1%
HNM	11.6%	11.6%	11.6%	11.6%

Table 1: Performance on the train respectively test set obtained with and without HNM. For training and detection a sliding window of 30×30 pixels was used. The window's step size was 10 pixels for training and 5 pixels for detection.

	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 %

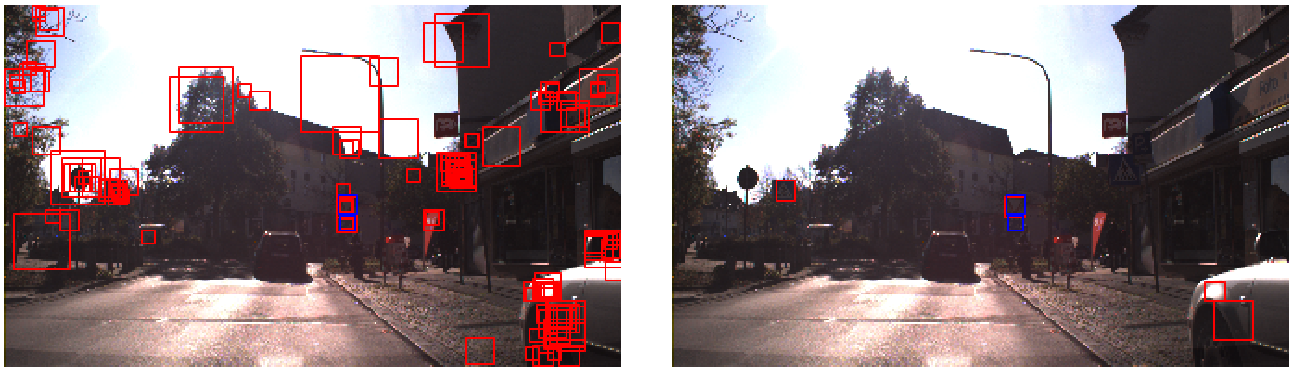
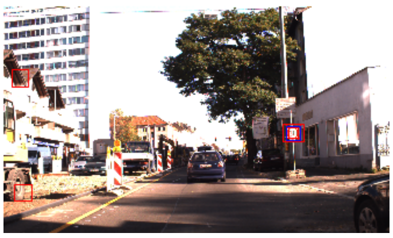
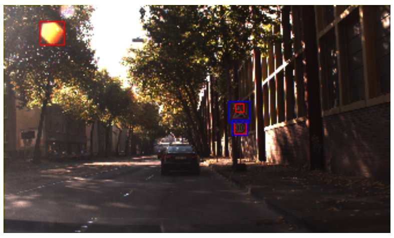


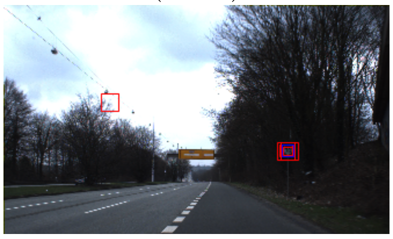
Figure 4: Bounding box prediction for the same image once using a SVM trained without HNM (left) and once trained with HNM (right). Ground truth bounding boxes are indicated in blue and predicted bounding boxes in red.

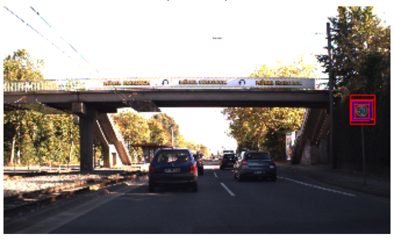
In Figure 5 some training images are shown in which the ground truth bounding boxes are indicated in blue while the predicted bounding boxes are drawn in red. The respective detection performances are noted below. Almost all traffic signs - if present - are detected and there are only few false positive predictions. Especially interesting is the image in the upper middle: while the construction site sign is not detected possibly due to the poor lighting conditions, the hexagonal light spot in the upper left corner is recognized as a traffic sign instead.


(100%)


(50%)

(100%)


(100%)


(100%)

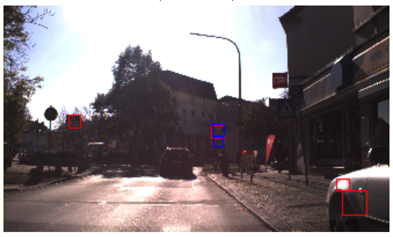

(50%)

Figure 5: Images from the training set with ground truth bounding boxes in blue and predicted bounding boxes in red. Prediction accuracy in percentage according to the prediction measure explained in section 3.3.4 is given below each image.

Figure 6 shows images from the test set along with ground truth and predicted bounding boxes. One can see that almost all traffic signs are detected as it was the case for the training images except for the rightmost sign in the upper middle image and the lower right image. If there is no traffic sign at all like in case of the bottom middle image, the algorithm does also recognize that correctly. However, the upper right and lower left image show that there are still sometimes false positives, which is not punished by our performance measure. Possibly the number of false positives can be reduced if another step of Hard Negative Mining is included.

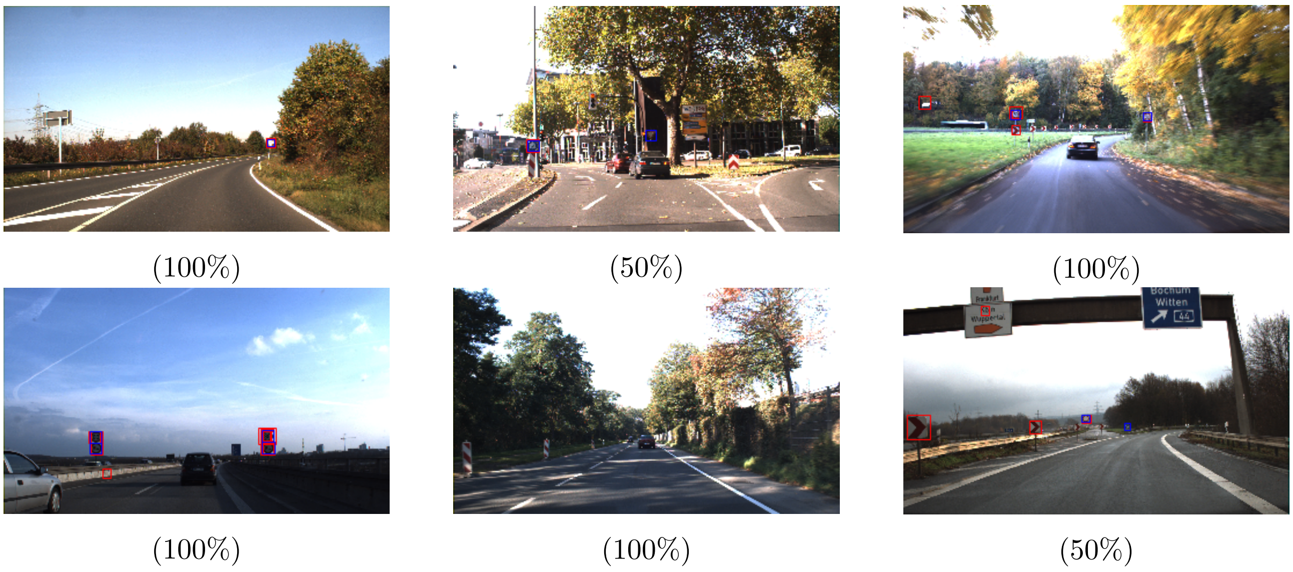


Figure 6: Images from the test set with ground truth bounding boxes in blue and predicted bounding boxes in red. Prediction accuracy in percentage according to the prediction measure explained in section 3.3.4 is given below each image.

Finally, we wanted to observe how the sliding window's step size during detection affects the performance. Figure 5.1 plots the resulting performance against the detection step size. Already with very few data points it becomes clear that the detection step size is very critical to the success of traffic sign detection.

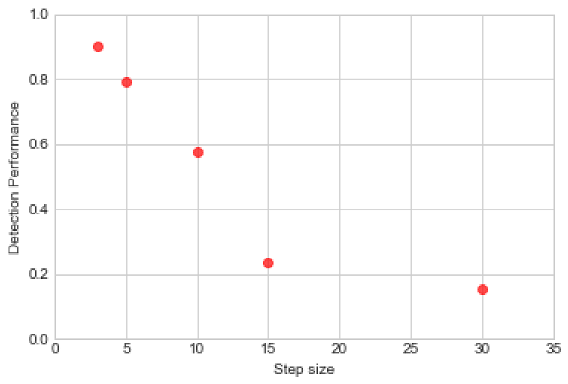


Figure 7: Detection performance on 30 randomly chosen images from the training set versus the step size of the sliding window used for detection.

5.2 Second Approach (Marvin)

6 Discussion

To sum up the first approach, one can say that HOG features learned by a SVM are very well suited to detect traffic signs in color images. By the use of Hard Negative Mining the false positive rate can be lowered significantly, however, one looses some accuracy at the same time. Generally, the approach is very sensitive to the setting of the hyperparameters especially towards the step size of the sliding window for detection. The smaller the step size the better the detection performance - at the price to a much higher computational effort. It could thus be an improvement to use region of interest extraction instead of a sliding window, like for instance selective search, which was basically done in the second approach by the R-CNN.

- je nachdem welches measure genommen werden falsch positive halt ignoriert...außerdem könnte man IOU¿=0.5 auch schon nehmen.

7 Appendix

7.1 Appendix A

0 = speed limit 20 (prohibitory)
1 = speed limit 30 (prohibitory)
2 = speed limit 50 (prohibitory)
3 = speed limit 60 (prohibitory)
4 = speed limit 70 (prohibitory)
5 = speed limit 80 (prohibitory)
6 = restriction ends 80 (other)
7 = speed limit 100 (prohibitory)
8 = speed limit 120 (prohibitory)
9 = no overtaking (prohibitory)
10 = no overtaking (trucks) (prohibitory)
11 = priority at next intersection (danger)
12 = priority road (other)
13 = give way (other)
14 = stop (other)
15 = no traffic both ways (prohibitory)
16 = no trucks (prohibitory)
17 = no entry (other)
18 = danger (danger)
19 = bend left (danger)
20 = bend right (danger)
21 = bend (danger)
22 = uneven road (danger)

23 = slippery road (danger)
24 = road narrows (danger)
25 = construction (danger)
26 = traffic signal (danger)
27 = pedestrian crossing (danger)
28 = school crossing (danger)
29 = cycles crossing (danger)
30 = snow (danger)
31 = animals (danger)
32 = restriction ends (other)
33 = go right (mandatory)
34 = go left (mandatory)
35 = go straight (mandatory)
36 = go right or straight (mandatory)
37 = go left or straight (mandatory)
38 = keep right (mandatory)
39 = keep left (mandatory)
40 = roundabout (mandatory)
41 = restriction ends (overtaking) (other)
42 = restriction ends (overtaking (trucks)) (other)

7.2 Appendix B: Requirements

As programming language python 3.6 was used. The following modules are needed to execute the code:

- os
- numpy
- math
- matplotlib
- imageio
- random
- seaborn
- sklearn
- skimage
- scipy

- pandas
- selectivesearch
- torch

8 References

[1] C-Support Vector Classification. `http://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html`. Accessed: 2018-06-15.

[2] HOG. `http://scikit-image.org/docs/dev/api/skimage.feature.html`. Accessed: 2018-06-15.

[3] Houben, S., Stallkamp, J., Salmen, J., Schlipsing, M., and Igel, C. Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In *International Joint Conference on Neural Networks (submitted)* (2013).

[4] Stallkamp, J., Salmen, J., Schlipsing, M., and Igel, C. The German Traffic Sign Detection Benchmark. `http://benchmark.ini.rub.de/?section=gtsdb&subsection=dataset`, 2013.