Traffic sign detection

Mihkel Lepson
Mathematical Statistics, 1^{st} year MSc.
University of Tartu
Tartu, Estonia
mihkel.lepson@ut.ee

Markus Haug
Data Science, 2^{nd} year MSc.
University of Tartu
Tartu, Estonia
markus.haug@ut.ee

Hendrik Šuvalov
Data Science, 2^{nd} year MSc.
University of Tartu
Tartu, Estonia
hendrik.suvalov@ut.ee

I. Introduction

The vehicles' movement in traffic is controlled by road signs. The road signs determine the allowed speed, priority of the vehicles and possible maneuvers. For autonomous vehicles the GPS could be used to determine the vehicles' location and from that the speed and possible maneuvers, but there will always be situations where GPS fails [1] or there could be rearrangements in traffic due to temporary road works. In this case the camera can be used to detect and recognize the road signs [2].

The goal of this project was to create an application that could take an image taken from a car in traffic and detect and recognize traffic signs in it. It first finds all the regions of interest (ROI) using color masks and contour detection, then passes through a binary classification SVM model that detects whether a ROI is a sign or not and is eventually classified using a convolutional neural network (CNN) model. For creating CNN models we used Keras API from Tensorflow platform. Due to the high volume of data we developed our application on HPC Rocket cluster provided by University of Tartu.

II. Overview of the data

The final dataset used in this project consisted of traffic images and signs from four different countries. The four countries were following: Sweden [1], Russia [3], Germany [4] and Belgium [5]. We ended up using these four countries datasets, because three of them are part of the European Union and hence have very similiar road signs and the Russian road signs are very similar to the EU ones. Also the Russian dataset was only one with labelled traffic lights. There were more datasets available like the US - or the Chinese traffic signs, however these countries traffic signs are not similar to the ones used here in the EU.

In total the four datasets we used gave us 53 360 labelled road signs in 78 classes. The major road signs that can be see in every day use were all included in the final dataset and only minor road signs were not present. The two biggest datasets were the Belgian road signs dataset and the German road signs dataset in that order. In the former there were 62 different classes and in the latter 44. In both datasets there were also recurring traffic sign images, however the recurring images were with different quality and brightness.

The Russian and the Belgian dataset both had speedlimit signs in them, however, they were labelled just as speedlimits and were without given speedlimit. To have more speedlimit data, the mentioned countries speedlimit signs were manually and automatically labelled.

III. Regions of interest

The process of finding the regions of interest begins by first capturing the masks of three primary colors used for signs: blue, red and yellow. Then, a morphological filter of dilation is applied to best capture the characteristics of the outlines of potential signs. Then, for each separate mask, the program finds the contours of the image and saves the areas where relevant shapes are found. The shapes that get saved get filtered by their size, how many edges they have and the ratio of their height and weight in order to filter out potential junk. Finally, for each of the areas saved, an SVM classification model is used as a final filter to best ensure that the region is indeed a sign.

Selecting the parameters for the filters in the process of finding the regions of interest was rather difficult. For example, whenever certain color mask ranges worked for one picture, then under different conditions for other pictures they might not have performed as well. The final parameters for each filter were selected with the intent of working best universally under all conditions.

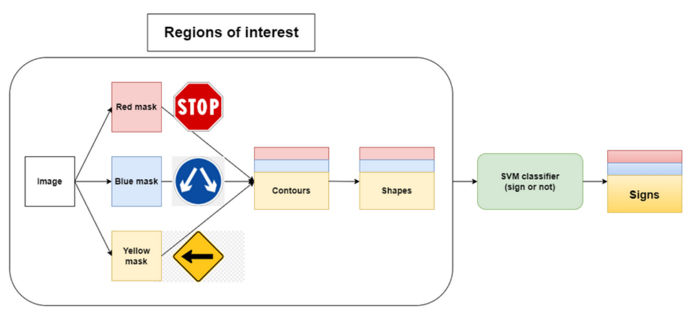

Figure 1. Workflow for finding regions of interest

A. SVM binary sign classification

In order to train the SVM model, we used 600 images of labelled signs from the dataset for the positive data for the model. We used the same pictures the positive samples were taken from and took random coordinates of the same size from the rest of the picture. This was to ensure that the samples of non-signs were still pictures that were relevant to traffic. The accuracy of this model on a testset was around 90%, however, its performance as the last step of the filtering was relatively poor in the actual workflow, but with a relatively high confidence, it did help remove some of the junk.

IV. Traffic sign classification

As traffic sign classification is an image classification problem we used convolutional neural networks (CNN). For further enhancing the CNN performance we preproccessed the images of traffic signs. We resized all the images to 50 pixels in width and height. Some of the images were extremely dark or light, therefore we used Contrast Limited Adaptive Histogram Equalization (CLAHE) algorithm to improve the contrast of images as suggested by Manju and others [6].

Before developing the final CNN classificator we had to relabel some of speed limit signs in the dataset because their labels were either incorrect or too vague. As there were about 1300 images like that we trained a simple CNN model (consisting of 2 convolutional 2D layers and 2 dense layers) for classifying 17 different speed limit signs and later manually corrected the labels if needed. This simple CNN model already yielded an accuracy of 77.4%.

After gathering all the traffic sign data in our disposal we started to further develop the CNN model. The final classification task was to differentiate between 77 traffic signs and a class of noise labeled as 'Other'. The addition of the noise class was meant as a precautionary measure, meaning to lessen the error in cases when the pipeline detected wrong objects. With respect to the aforementioned simple CNN model, we added multiple convolutional and dense layers. For avoiding overfitting we added dropout layers. The final model's architecture can be seen in Tabel 1. The model performed suprisingly well, achieving an accuracy of 98.88% on the training dataset and 98.58% on the validation dataset (split $80/20$). All of the classes were classified with high specificity and recall as the model reached ROC-AUC value of 99.99% on combined data.

We also conducted confidence analysis for our final model. We checked the output of the probability values on which our model made the predictions. In the case of correct predictions, our model was most confident when classifying the signs labelled as 'Go straight or turn right', 'Give way', 'Give way to oncoming' being more than 99.999% confident. The model was least confident (86.22%) when predicting the '15 sign'. In the case of incorrect predictions, the model was eager to classify the sign as 'Pass left side', 'Other' or 'Priority road' being more than 99.9% confident. The model was least confident when predicting 'No overtaking end' (46.46%) wrongly.

It has to be noted that among the wrong predictions there were many cases where the model had very slim chances of classifying correctly. This was due to the fact that some traffic signs in our dataset were either incorrectly labelled or unrecognizable to the human eye. As far as wrongly labelled signs are concerned, it was not feasible to check every image in the dataset. Despite these difficulties, 90% of the classes obtained a f-score higher than 0.99 which is a fairly good result.

CNN architecture		
Layer	Output Shape	Param #
Conv2D	(None, 50, 50, 8)	608
Activation	(None, 50, 50, 8)	0
BatchNormalization	(None, 50, 50, 8)	32
MaxPooling2D	(None, 25, 25, 8)	0
Conv2D	(None, 25, 25, 16)	1168
Activation	(None, 25, 25, 16)	0
BatchNormalization	(None, 25, 25, 16)	64
Conv2D	(None, 25, 25, 16)	2320
Activation	(None, 25, 25, 16)	0
BatchNormalization	(None, 25, 25, 16)	64
MaxPooling2D	(None, 12, 12, 16)	0
Conv2D	(None, 12, 12, 32)	4640
Activation	(None, 12, 12, 32)	0
BatchNormalization	(None, 12, 12, 32)	128
Conv2D	(None, 12, 12, 32)	9248
Activation	(None, 12, 12, 32)	0
BatchNormalization	(None, 12, 12, 32)	128
MaxPooling2D	(None, 6, 6, 32)	0
Flatten	(None, 1152)	0
Dense	(None, 128)	147584
Activation	(None, 128)	0
BatchNormalization	(None, 128)	512
Dropout(0.5)	(None, 128)	0
Flatten	(None, 128)	0
Dense	(None, 128)	16512
Activation	(None, 128)	0
BatchNormalization	(None, 128)	512
Dropout(0.5)	(None, 128)	0
Dense	(None, 78)	10062
Activation	(None, 78)	0

Table I
Final model's architecture

V. Results

The overall model performs well on the similar data it was trained on, which is during daylight time as can be seen on the Figure 2. In the more difficult conditions like a night time city the model does not work that well. From Figure 3 we can see that the model detects yellowish street lights as road signs. Also it was seen that the model is sensitive if there is frost on the vehicle's windshield or images are bit blurry. It is something to keep in mind when one wants to use the road sign detection and recognition on autonomous vehicles that the cameras' vision can be occluded.

Figure 2. Result of the final model

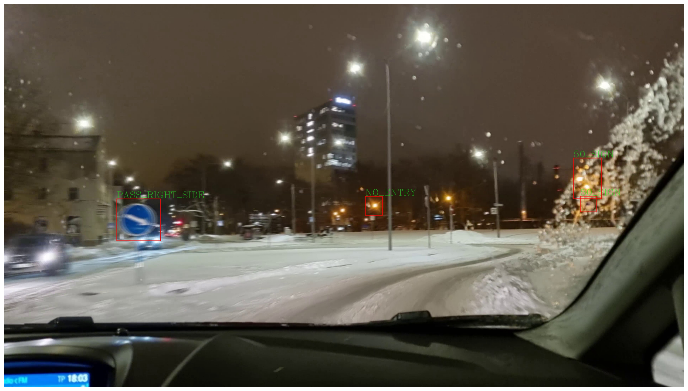

Figure 3. Result of the final model 2

VI. CONCLUSION AND FUTURE WORK

One of the biggest hardships in finding the regions of interest was the fact that the pictures were taken in very different conditions - the cameras themselves differed, meaning that different pictures were produced, but also the weather conditions and the overall environment affected the detection as well. This made finding parameters that work best for all conditions quite difficult, if not impossible. Perhaps in future work additional filtering of picture types could be taken into account to increase the accuracy for finding the regions of interest, for example pictures could be separated into day/night and different color masks could be used depending on the conditions. Then, a different model could also be used for the recognition based on the conditions to further improve the accuracy. When it comes to traffic sign classification, the accuracy of the model could surely be enhanced. These kind of models should be made for each different traffic regulation areas as they have different traffic signs. Furthermore, in this project only 77 different signs were classified, in reality there are much more traffic signs. Additionally, in the future projects even more noisy data could be used for traffic sign recognition to take into account that the road signs can be partially occluded, whether by other objects or by water or snow on the camera's lens.

VII. CODE

The repository for the code can be found here - https://github.com/MihkelLepson/traffic_sign

REFERENCES

[1] Larsson F., Felsberg M., 2011. Using Fourier Descriptors and Spatial Models for Traffic Sign Recognition. In: Heyden A., Kahl F. (eds) Image Analysis. SCIA 2011. Lecture Notes in Computer Science, vol 6688. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-21227-7_23

[2] Häne C., Heng L., Lee G. H., Fraundorfer F., Furgale P., Sattler T., Pollefeys M., 2017. 3D Visual Perception for Self-Driving Cars using a Multi-Camera System: Calibration, Mapping, Localization, and Obstacle Detection https://paperswithcode.com/paper/3d-visual-perception-for-self-driving-cars

[3] Maranhão A., 2020. Road Sign Detection https://www.kaggle.com/andrewmvd/road-sign-detection

[4] Stallkamp J., Schlipsing M., Salmen J., Igel C., 2011. The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In Proceedings of the IEEE International Joint Conference on Neural Networks, pages 1453–1460. 2011

[5] Timofte R., Zimmermann K., van Gool L., 2011, Multi-view traffic sign detection, recognition, and 3D localisation, Journal of Machine Vision and Applications (MVA 2011), DOI 10.1007/s00138-011-0391-3, December 2011, Springer-Verlag

[6] R.A. Manju and G. Koshy and Philomina Simon, Improved Method for Enhancing Dark Images based on CLAHE and Morphological Reconstruction, Procedia Computer Science, volume 165, pages 391-398,2019, DOI https://doi.org/10.1016/j.procs.2020.01.033