Wrocław University of Science and Technology
Faculty of electronics

Road detection and recognition algorithm

Author: Paweł Boruch
Class: Intermediate Project
Supervisor: Ph.D. Witold Paluszyński
Date: February 13, 2021

Abstract

The aim of this project is to design and develop road detection and recognition algorithm. Algorithm has to apply visual mapping of the detected road based on and with road landmarks that can be important to differentiate between if some region should be detected as road or not. Program should be able to take as input image or video of car driving on the road (road image with camera at the front of the car) and also should be able to work with Euro Truck Simulator 2 video (realistic simulator of truck driving) without separate classifiers or systems made especially for that simulator. Output of the system is input video with colored overlay specifying which part was detected as road with landmarks.

Program works as specified (detects and outputs overlay of specified data), however to increase robustness and decrease number of classifiers used some of the landmarks that might seem important were rejected. Also approach was chosen to increase robustness to train 2 classifiers with different methods which should classify all road and landmarks individually and with comparison of those 2 generalization of area chosen should be better (both classifiers operate on different datasets). However different approach then CNN was hard to apply and system algorithm is working with 0 classifier which specifies if image is image with road on it, and classifier 1 with classification. Attempts on building second classifier are mentioned with generally assumed structure in the rest of raport.

Project was successful. All the major goals were fulfilled.

1. Introduction

With the steady improvement and development of self-driving cars and technology over the recent years, there is growing need for fast and reliable road detection and recognition algorithms, which might support onboard sensors used for self-mapping of those cars in the dynamic environment.

The goal of the project is to with the use of open source resources and code develop robust algorithm for road detection and recognition based on important road landmarks and their visual mapping on provided source picture. For robustness improvement final classifier should be classifier matrix which consists of two or more classifiers. Using multiple classifiers and taking to a count their predictions (if they for example overlap on each pixel of classified data) might improve correctness of results, if each classifiers have good performance but providing bad classifier to classifier set might disturb results, so making new unchecked classifier was omitted and only classifiers based on scientific papers and well known sources were used.

Many algorithms and resources shorten the task of road detection to simply detecting lanes on the road and space in between. This approach might be good in a lot of modern, good quality, road data applications, however its' robustness is very limited to only one landmark, and proves very not reliable if lines are distorted, or there is no one.

First approach to specify those landmarks that should be detected were: lines on the road and direction of the moving cars. First approach were proven as right landmark for some cases, however in the process of data collection (to ensure robustness with different non-standard examples) the second proved not correct, and not improving robustness.

According to worldstandards.eu 85 countries have policy of driving on the left side of the road (such as India, Great Britain and Japan). Population wise those countries however could be under or overrepresented in datasets, and robustness can't be ensured for example in cases of changing lines on empty road.

Output classifier of the project should be able to correctly classify video streams / files of simulator (such as Euro Track Simulator 2) and general video files of front camera on the car during normal driving.

2. Summary

After analysis of problem and literature (resources section) general graph of the system was designed, showed on figure 1.

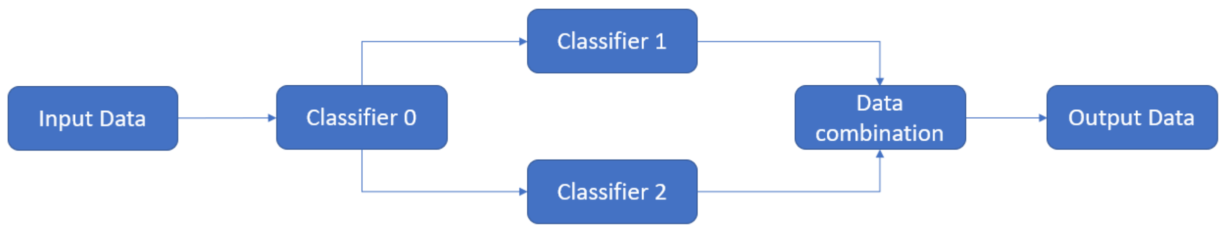

Figure 1. General graph of the system

Input data frame for example picture is first handled by classifier 0, which is general classifier which solves the first question that algorithm should answer: Is there road in the picture? With Boolean classification, true or false. Firstly this classifier should resize the picture for frame size that could be used in different classifiers. Frame size used: 352x288.

If the result of classifier 0 is true, resized data frame is used in classifier 1 and 2. Those classifiers provide classified overlay for original frame.

Data combination section apply weight the provided overlay by each individual classifier and apply coloring for each of them.

Output data consists of sum of weighted images of overlay and Input data. (opencv function addWeighted(frame_in, 0.5, frame_out,0.5,1.0 was used, where frame_in is input frame and frame_out is overlay (result of model)[4])

3. Classifiers' specification and theory

3.1 Classifier 0

As entry classifier that differentiates between if on the input frame there exists road or not convolutional neural network was chosen (tensorflow package was used).

CNN (Convolutional Neural Network) is a class of deep, feed forward, artificial neural networks most commonly applied to analyze visual imagery. Dependent on the problem to solved CNN always contains input layer of the size of input data (in our example of the size of input image) and at the end fully connected and output layer. Those end layers are flattened results of CNN classification. Between input and fully connected layer are hidden convolution layers with specific size, max pooling algorithms in between and rectified linear activation function. Based on problem type and data number and size of those convolutional layers might differ.

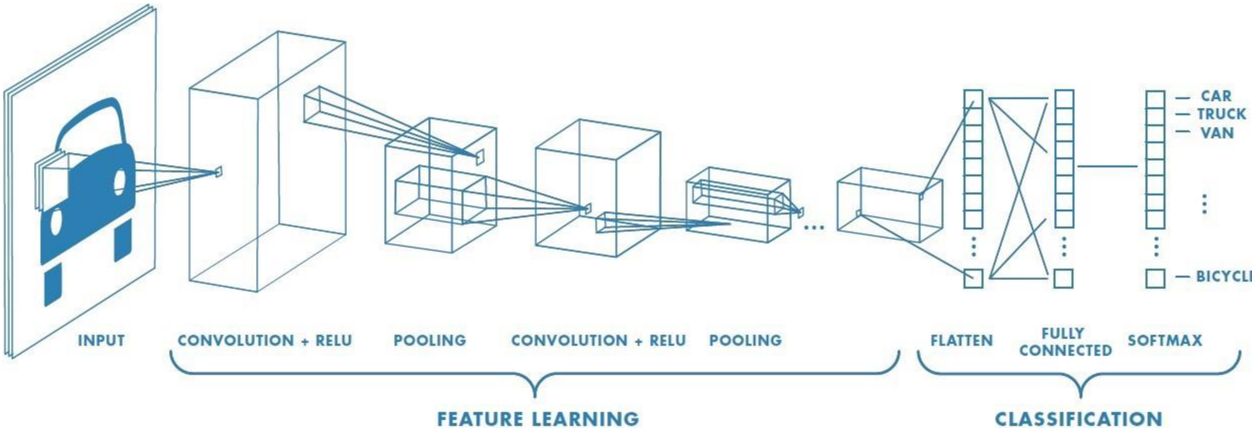

Figure 2. General graph of CNN [5]

For this specific problem feature learning like on Figure 2 was added 3 times with first convolutional layer of size of 32 and second of size 64 with rectified linear activation functions and max pool 2d in between.

Datasets used:

For training and validation process 900 road pictures (300 from each countries: Czech, India, Japan) (from image dataset for road damage detection available at webpage https://data.mendeley.com/datasets/5ty2wb6gvg/1) and for non-road pictures Stanford Background Dataset (available at https://www.kaggle.com/balraj98/stanford-background-dataset) was used. Pictures that containing road were removed from second dataset, however to ensure robustness border cases were not removed (for example sidewalks, car pictures on the road when road was not taking most of the picture frame).

Partial results and their dependance on dataset:

After multiple attempts and 10 epochs of learning with learning rate of 0.001 results below were accomplished with accuracy of 0,961 and loss of 0,108 and for validation set: loss of 0,17606 and validation accuracy of 0,934.

However to check if model isn't overfitting from training, predictions on 1000 files from each country resulted in approximately 95,3% correctness after that all files of first dataset (training directory 10535 pictures) were used to check how much of these files would be classified correctly as road. Correct assignment of all files were on level of 94,2%.

After that learning rate was changed to 1e-4 with 20 epochs loss reduced to 0,06957, accuracy increased to 0,9917 and on validation set values also decreased but the changes were minor with validation loss of 0,16103 and validation accuracy 0,94.

With 10535 files as test correct assignment was on the level of 94.65%.

As the input verification this is too low value to be satisfied with. Dataset was modified with excluding border cases, and increasing number of epochs to 50.

loss	accuracy	Validation loss	Validation accuracy
0,00122	1	0,17736	0,9414

With 10535 files as test correct assignment was on the level of 95.9%.

Further improvement would require further dataset rearrangements. Improvement of 29% from first mentioned try is satisfactory and 4% can be in the error margin of the system based on large scale data.

Classification example:

Figure 3. Results of classification by classifier 0

As we can see images which strictly didn't correspond for example traffic light image were not classified as road and others for example image of pedestrian walkway also wasn't detected as road (which is correct) image.

3.2 Classifier 1

Classifier 1 uses Road Surface Semantic Segmentation for detecting road irregularities and surface as landmarks. This classifier model was created based on resources [1], [7] with the use of RTK dataset (available at https://lapix.ufsc.br/pesquisas/projeto-veiculo-autonomo/datasets/?lang=en). Based on code and data provided in [6] testing dataset was increased by hand classified data and parameters modification.

For Semantic segmentation of the road surface CNN was used (like in classifier 0, however here was used old version of fastai 1 package to achieve task with different architecture) with architecture U-NET. U-NET architecture was designed to perform semantic segmentation tasks in medical images, however can be applied to other semantic segmentation tasks. ResNet (resnet34) based encoder and decoder are used. Resnet34 is a 34 layer convolutional neural network that can be utilized as a state-of-the-art image classification model. This is a model that has been pre-trained on the ImageNet dataset--a dataset that has 100,000+ images across 200 different classes. However, it is different from traditional neural networks in the sense that it takes residuals from each layer and uses them in the subsequent connected layers (similar to residual neural networks used for text prediction)(according to [8]).

Semantic Segmentation is a process of classification each pixel of image to a particular label without detection of uniqueness of each object. For example in this work Road surface segmentation is used and as part of this project these labels can be classified as road landmarks for road detection. For example speed bumps, asphalt, unpaved road, road lines, storm drain, pothole etc.

With the usual approach without by hand data specification, usual approach is to specify ROI (region of interest) and teach classifiers based on that image, that approach is correct, if we have standardized camera position and generally same incline, however if those things vary, more data is needed for classification (ROI should vary or we should just take whole image).

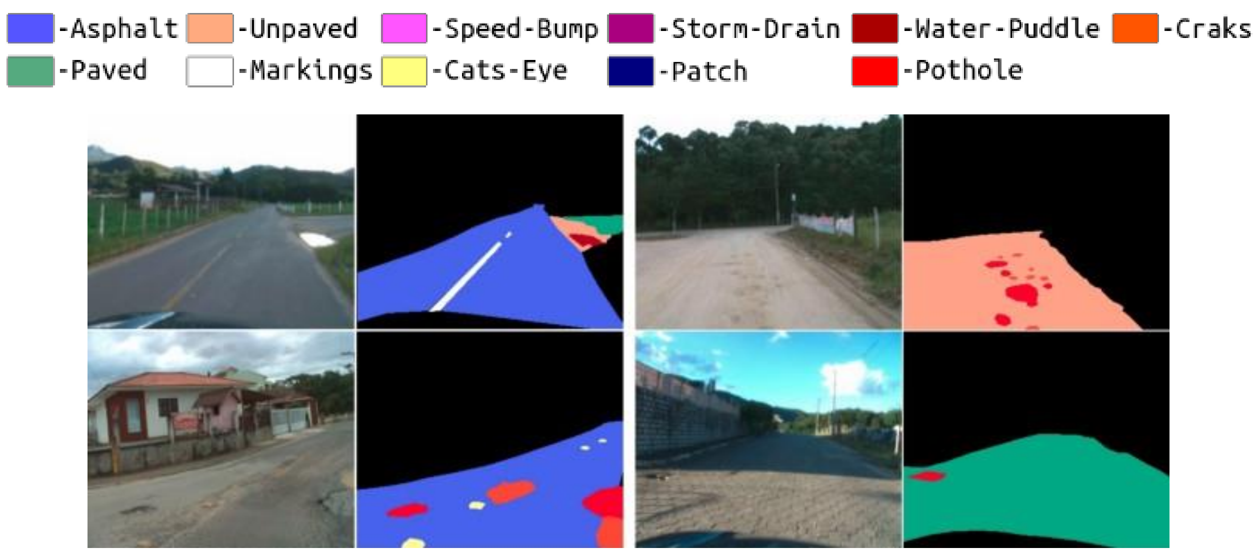

Figure 4. Pairs of images specifying input image and classified that are used for validation [7]

With sample data from different dataset this is program result with 0 and 1 classifier:

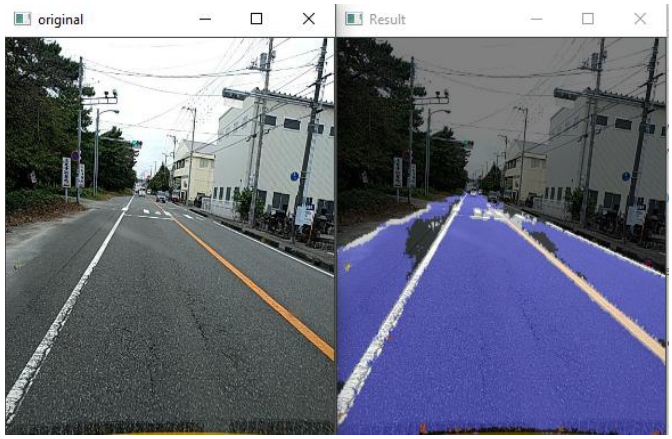

Figure 5.Program result windows for road file as input

On the left is visible data input and on the right classification with an weighted overlay. With some spots missing most of the road space was classified well.

On training and validation set: training loss is at 0,058 level, and on validation set is at 0,2 level. General accuracy of the model was around 96,5%.

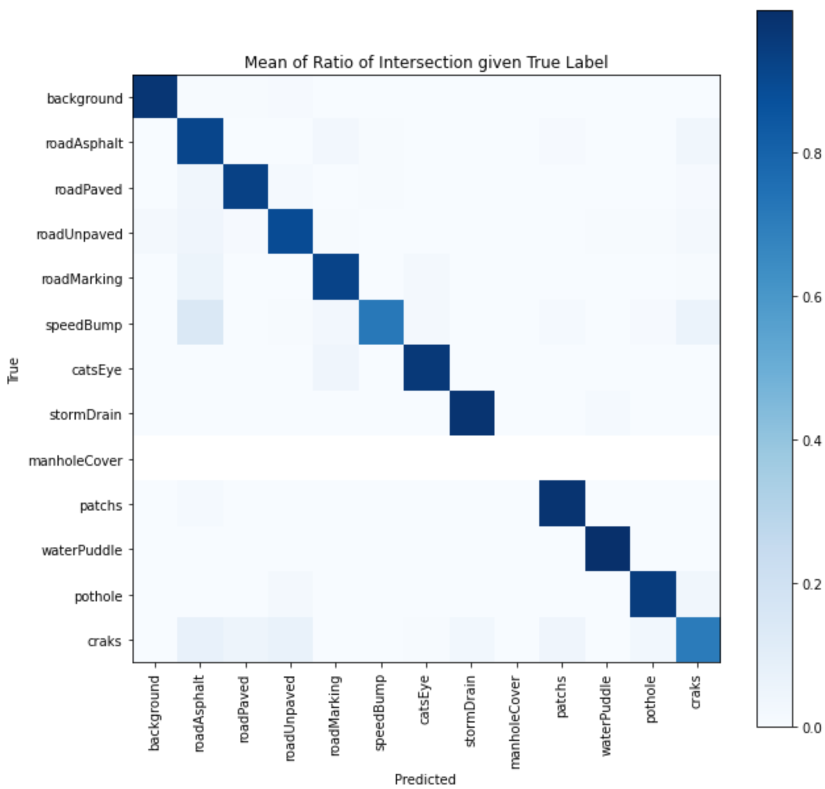

Figure 6.Confusion matrix of the model

In case of performance this trained classifier model takes a lot of time to execute approximately 3s, with the use of this classifier main part of the task will be performed, but not in real time.

3.3 Classifier 2

For second classifier multiple attempts were made to make classifier without or detecting algorithm without the use of CNN due to high time of code execution.

Attempts were made to substitute semantic segmentation which is related to pixel structure with sum of K-means Clustering and seeded region growing algorithms for algorithm speed increase, however finding corresponding parameters for both algorithms with many detection target was proven difficult and unapplicable.

Seeded region growing algorithm: First we select a seed (position of initial pixel in image) and h value which is +- degree of color values. Then from that pixel we search neighboring pixels that have the same color parameters of original pixel with + and – range of h value and then select them to ROI and execute selection for every new pixel chosen to the point when no pixels will be in that range.

Problem with this algorithm was to specify the seeds and ranges to be robust. Example shown below:

Figure 7. Classification based on seeded region growth and Kmeans clustering

As it is visible classification is possible with similar lighting condition however with different lighting algorithm shows a lot of problems.

Next attempt based on article [2], which detects a road by boosting using feature combination. One of the main problems is lack of proper datasets and problems with using region growing technique mentioned in [3] (hardly available resource).

With generally specified seeded region growing algorithm, training dataset and labeling was generated from image dataset for road damage detection used in classifier 0 (due to hand assignment only 200 samples were chosen as dataset). Points and parameters of seeded region growing algorithm were chosen arbitrarily for each file of training dataset to feel the most of the road image.

Parameters chosen for labeling: Coordinates, color, size, h value. In [2] h value is not specified which is due to usage of different algorithm, but luminance is specified, which in case of whole picture is hard to measure and in case of field is contained in color specification. H value is maximum threshold possible for variance between near pixels, if it is too high different colors will be in final region.

With many attempts of feature generation I was not able to choose distinct feature sets to run algorithm, and even chosen datasets were not ideal due to problem mentioned in first attempt (ROI of seeded region growing algorithm).

Final attempts of creating second classifier with the use of edge detection and algorithm mentioned in [9].

For edge detection 2 directional Sobel derivatives were calculated and even with specifying of right ROI, algorithm could not provide enough robustness to be included in final classifier set due to lack of robustness of such approach (in which algorithm could not provide clear zones).

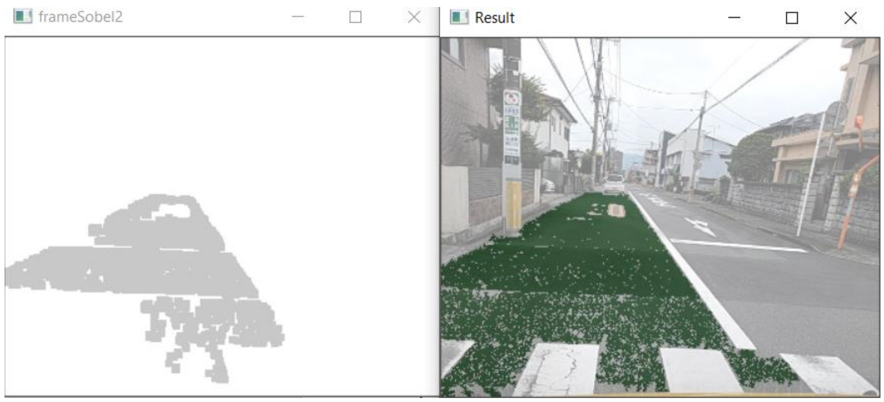

Figure 8. Comparison of results edge detection algorithm with previous

Overall results were comparable or worse in first classifier. Even with speed increase of algorithm no one of classifiers attempted were chosen as second classifier due to their performance.

4. Results

Resulting system consists of system mentioned in summary without Classifier 2. Resulting program took as input 2 values: first would be mode and second file name. Modes available: file (program takes input image in correct format (png for example) and outputs weighted image with overlay) (result example Figure 5), video, takes video and frame after frame computes and outputs overlayed images, screen (takes specified with Trackbars zone of screen and outputs result based on data-stream of screen section).

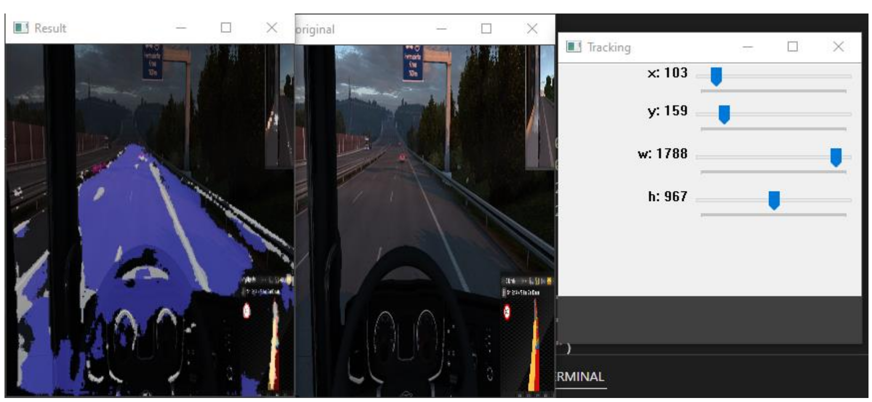

Figure 9. screen mode with toolbars of Euro Truck simulator 2 with results

As we can see road detection with small errors works on Euro Truck Simulator 2 as it was specified.

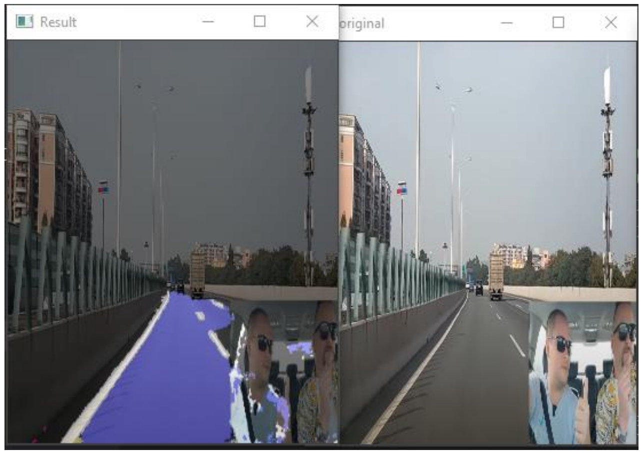

Figure 10. Video mode of you tube video with road image [10]

Video also is well classified and as image example we can take Figure 5. Resulting classifier set could perform data stream classification with average FPS of 0,2.

5. Conclusion

CNN algorithms provide very good way of classification images however with larger number of classification parameter and long training model could have long execution.

Classifiers without the use of AI or NN with strict use of mathematical algorithms were proven to be much faster (5-10 FPS), however robustness of such algorithms can be questionable.

Joint algorithm of 2 classifiers and pre processing and final data processing takes on average 5s to execute so real time operation of such algorithm, even if precision is quite good, is not possible.

6. References

[1] – "Road Surface Classification with Images Captured From Low-cost Camera - Road Traversing Knowledge (RTK) Dataset" by Thiago Rateke, Karla A.Justen and Aldo Von Wangenheim

[2] - https://ieeexplore.ieee.org/abstract/document/4290141/references#references

[3] - Ming-Yang Chern and Shi-Chong Cheng, "Finding Road Boundaries from the Unstructured Rural Road Scene", *16th IPPR Conference on Computer Vision Graphics and Image Processing(CVGIP*, 2003.

[4] - https://docs.opencv.org/3.4/d6/d00/tutorial_py_root.html

[5] - https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-sthe-eli5-way-3bd2b1164a53

[6] - https://towardsdatascience.com/road-surface-semantic-segmentation-4d65b045245

[7] - https://towardsdatascience.com/road-surface-semantic-segmentation-4d65b045245

[8] - https://models.roboflow.com/classification/resnet34

[9] - https://ieeexplore.ieee.org/abstract/document/1438028

[10] - https://www.youtube.com/watch?v=9LA3QQZbPz0