UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Leonid Dashko

Road Detection and Recognition from Monocular Images Using Neural Networks

Master's Thesis (30 ECTS)

Supervisor(s): Amnir Hadachi, PhD

Tartu 2018

Road Detection and Recognition from Monocular Images Using Neural Networks

Abstract

Road recognition is one of the important aspects in Autonomous Navigation Systems. These systems help to navigate the autonomous vehicle and robot on the ground. Further, road detection is useful in related sub-tasks such as finding valid road path where the robot/vehicle can go, for supportive driverless vehicles, preventing the collision with the obstacle, object detection on the road, and others.

The goal of this thesis is to examine existing road detection and recognition techniques and propose an alternative solution for road classification and detection task.

Our contribution consists of several parts. Firstly, we released the road images dataset with approximately 5,300 unlabeled road images. Secondly, we summarized the information about the existing road images datasets. Thirdly, we proposed the convolutional LeNet-5-based neural network for the road image classification for various environments. Finally, our FCN-8-based model for pixel-wise image recognition has been presented.

Keywords: deep convolutional neural networks, road detection, pixel-wise classification, scene segmentation.

CERCS: P170 Computer science, numerical analysis, systems, control

Monokulaarsetelt piltidelt tee tuvastamine ja eristamine kasutades tehisnärvivõrke

Lühikokkuvõte

Teede eristamine on oluline osa iseseisvatest navigatsioonisüsteemidest, mis aitavad robotitel ja autonoomsetel sõidukitel maapinnal liikuda. See on kasutusel erinevates seotud alamülesannetes, näiteks võimalike valiidsete liikumisteede leidmisel, takistusega kokkupõrke vältimisel ja teel asuvate objektide avastamisel.

Selle töö eesmärk on uurida eksisteerivaid teede tuvastamise ja eristamise võtteid ning pakkuda välja alternatiivne lahendus selle teostamiseks.

Töö jaoks loodi 5300-pildine andmestik ilma lisainfota teepiltidest. Lisaks tehti kokkuvõte juba eksisteerivatest teepiltide andmestikest. Töös pakume erinevates keskkondades asuvate teede piltide klassifitseerimiseks välja LeNet-5'l põhineva tehisnärvivõrgu. Samuti esitleme FCN-8'l põhinevat mudelit pikslipõhiseks pildituvastuseks.

Võtmesõnad: neurovõrgud, tee tuvastamine, pikslipõhine klassifitseerimine, stseeni segmenteerimine

CERCS: P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine

Acknowledgement

First of all, I would like to thank my supervisor, Amnir Hadachi, for his supportive guidance, knowledge, patience, inspiration, and ideas which have resulted in this written thesis.

Also, I am grateful to Artjom Lind for his creative ideas and a piece of advice.

I would like to thank High Performance Computing Center (University of Tartu) who provided the access to GPU Rocket Cluster for running thesis-related tasks.

Finally, I want to thank my parents and friends for their support.

Abbreviation and Acronyms

ITS Intelligent Transportation Systems

AVN Autonomous Navigation System

GPU Graphics processing unit

CPU Central processing unit

CNN Convolutional neural network (also known as ConvNet)

OpenCV Open Computer Vision library

RoI Region of Interest

Color spaces:

- RGB Red Green Blue
- HSL or HSI Hue Saturation Lightness (Intensity)
- HSV or HSB Hue Saturation Value (Brightness)

Table of Contents

1 Introduction

1.1 General view

Over the last decade, the great interest has been rising to the field of ITS and its sub-field, Autonomous Navigation systems (AVNs). To be more specific, the AVN involves the navigation of autonomous vehicles and ground robots. One of the important tasks in the AVN systems is the road detection which is the topic of this master's thesis.

Generally, the road detection is extremely important due to several reasons: finding the valid path where the vehicle/robot can go; driverless driving; limit the region of interest for other tasks (such as preventing the collision with obstacle, and object detection on the road).

So far, many researchers have tried to solve the road detection problem. However, their algorithms can handle only favorable conditions and may not work under varying environments. That is why this problem is still open and needs to be solved.

The main problem of the state-of-the-art techniques is that they cannot handle all the cases of road types or handle only specific environment. For example, the algorithm behaves well only on the images with highway roads (structured roads) and produces unacceptable prediction for the images with roads in rural areas. Also, some of the existing algorithms have not been trained to consider curvy, snowy, rainy roads, and different daytime conditions which change the colors in the scene completely.

1.2 Objectives

The primary objective of this thesis is to develop a system for the road detection from single image.

At the same time, there are other tasks related to the primary objective such as analyzing existing papers, state-of-the-art benchmarks and their solutions, search for relevant and available road image datasets, finding dominant features, training and validation of the own solution for road recognition problem.

1.3 Contribution

Several contributions have been done while working on this thesis:

- New dataset of 5,300 images of unstructured/structured road images has been collected and categorized by type of road by the author
- Analyzed and summarized information about existing datasets of road images
- Analyzed various color spaces and their outputs on the images under different environments
- Tested and applied different types of models for the road detection/recognition
- Built a general classifier for road image classification which predicts whether the image has road or not
- Built a pixel-wise classification CNN model for the road recognition

1.4 Road Map

Chapter 2 provides an overview of some of the previous work on the topic of road detection and available road images datasets.

Chapter 3 describes the color analysis experiments; software and tools which have been used for constructing the neural networks; the structure of the models (classification and recognition of the road in images). Furthemore, this chapter contains the information about the data used for training and validation steps.

Chapter 4 illustrates the obtained results and summarizes them.

Chapter 5 concludes this thesis and discusses future related work.

2 State-of-the-art

2.1 Introduction

This chapter briefly summarizes the information about latest published articles related to the field of image classification, road detection and scene segmentation, and popular techniques. Besides, the advantages and disadvantages are mentioned of the state-of-the-art approaches.

Additionally, at the end of this chapter, the available road images datasets are described.

2.2 Related work

One of the earliest projects for controlling an autonomous vehicle on real outdoor roads was the ALVINN project (Pomerleau 1992). In this project, a 3-layered feed-forward neural network was used to detect the road from a single grayscale image and the output of the network were the commands for steering the vehicle. On the one hand, the system performed satisfactorily in the environment similar to training data. On another hand, the system was limited in generalizing to new road environments.

To tackle the limitation of generalizing the road in ALVINN project, the same authors proposed a new system called MANIAC (Jochem et al., 1993). MANIAC (Multiple ALVIIN Networks in Autonomous Control) has a modular architecture, where the several networks were trained to handle different types of roads where the final output was a combined result from all the modules. Although the system performed better than ALVINN, it still was not accurate in the sense of handling the entire range of possible road-types that in its turn requires a larger number of individually trained modules. Also, when a new module was added to the system, the entire system had to be retrained. However, one of the drawbacks of this approach is that it only works on straight or slightly curved roads.

In 2012, Krizhevsky, Sutskeyer, and Hinton published an article [1] that is considered as one of the breakthroughs in computer vision during the last decade about their winning model in the ImageNet ILSVRC-2012 contest. In this article, they describe their large deep convolutional neural network that helped to classify the 1.2 million high-resolution images into the 1000 different classes. Their approach performed extremely well and they achieved a winning top-1 and top-5 error rates of 17.0% and 37.5% accordingly using purely supervised learning. Also, they report that their network's performance degrades if a single convolutional layer is removed. They highlight that removing any of the middle layers results in a loss of about 2% for the top-1 performance of the network. That is why the depth is really important for achieving such results. During the training phase, the authors added "dropout" layers to reduce overfitting. Furthermore, they emphasize that using rectified linear units, called "ReLU", as nonlinear activations were beneficial for their classification task. Their neural network structure consists of five convolutional layers, some of which are

followed by max-pooling layers, and three fully-connected layers with a final 1000-way soft-max layer. Overall, the network has 60 million parameters and 650,000 neurons.

During the last decade, the number of additional novel techniques has been introduced to tackle the road detection problem. We would like to group them into several categories.

Color and edge detection-based techniques.

In 2010, a non-parametric model has been introduced in [2] for road recognition of different types of road. The kernel density estimation combined with the block-classifying method has been used to extract the lane boundaries. The former method studies the characteristics of data distribution only from the data samples itself and does not make any assumptions or use any prior knowledge about the data distribution. The latter one uses B-spline curve model for fitting the boundaries of the road lanes. Authors report that the proposed method is robust against the interference from shadows, light changes, and other factors. However, we may observe that the algorithm was not tested on images with objects (vehicle or pedestrians). Also, it cannot handle the T-junction due to the use of edge detection method which can search for one road at a time. Additionally, we tried to find the image dataset, which contains dusty, snowy, shadows, and other test environments) that the authors used in their work, however, it is no longer available online.

Another work [3] was published in 2013, tried to enhance the techniques against the effect of shadow based on normalized differences index technique and morphological operations. As a result, the proposed algorithm is capable of detecting and eliminating the shadow. Interestingly that all operations with shadow removal happen in HSV color space, after finishing them, they convert back the image to RGB, and pass it forward to support vector machine (SVM) for pixel-wise classification.

In 2014, the color-based approach has been described in [4]. The approach is based on statistical analysis of RGB pixel values and plane extraction through V-disparity map. As a result, the authors receive a confidence map for every pixel. The proposed methodology has been tested on KITTI dataset, where the accuracy was approximately 92-94% (depending on the type of road).

Regardless our work is about road detection, we have also had a look into [5] about the haze or fog removal (dehazing technique) from the image which can be used in road recognition task as well. The authors write that the dehazing techniques proposed till now are computationally complex and time-consuming (the processing time is approximately 20 seconds or even more per one frame) and thus not suited for real-time applications. The technique proposed by researchers of that article concentrates on the fast restoration of scene color and contrast based on color analysis of the scene objects. As the image detail suffers due to haze, the road edge of an outdoor scene degrades significantly. Also, the proposed model has been extended to road edge detection and tracing along with on-road obstacle detection. The proposed method for dehazing requires about 1-2 seconds per frame which is about 15-20-times faster than existing techniques. The proposed technique includes the analysis of different signals such as peak-signal-to-noise ratio (PSNR), contrast-to-noise

ratio (CNR - is a measure for assessing the quality of the restored image in case the image is affected by haze), signal-to-noise ratio (SNR). It is important to note that processing time varies depending on the image content.

Besides, we familiarized with [6] that describes sub-task of road detection problem – vehicle detection. The authors of [6] focus on monocular vision-based vehicle detection under challenging lighting conditions, and they tried to solve it through the sliding window of a fixed size and utilizing adaptive global Haar (AGHaar) technique for feature classification and vehicle detection in both, daylight and night conditions.

Horizon line detection.

This sub-task helps to simplify the task for the neural network because once we detected the horizon line, we can remove all pixels above this line. Usually, the horizon line algorithms are based on detecting a sharp change between colors in the upper part of the image. This technique has been used in the following papers: [7], [8].

Vanishing Point Detection (VPD).

In 2010, a new VP algorithm, called locally adaptive soft-voting (LASV), has been introduced for road detection task in [9]. This road detection method integrates texture orientation and color information of the road. The texture orientation estimation relies on Gabor filter where they consider 36 different orientations. To find the road area, a vanishing-point and the group of dominant edges based on an orientation consistency ratio (OCR) feature. Further, two dominant edges are selected from this group as the road borders by combining color cue. Furthermore, they have additional constraints such as the assumption that the vanishing angle between the two road borders should be larger than 20 degrees; the smallest size of the road line is set to be one-third of the height of the image (it helps to avoid possible false detection caused by short edges). LASV has several advantages such as almost real-time processing (17 frames of size 240x170 per second), confidence level for every pixel, the algorithm relies on the combination of techniques - texture and color features. Also, the algorithm also takes into account paved and moderately curved roads. The key disadvantages are the failure of road borders detection due to extreme illumination conditions (e.g., intensity saturation or strong edge of shadow caused by trees); fail cases when the vehicle is going up or down the mountain, or making a turn.

In 2013, Bui, Saitoh, and Nobuyama have published [10] where they proposed a new road detection method based on texture orientations estimation and vanishing point (VP) detection. Firstly, the method estimates a VP through a texture-based soft voting algorithm. Further, two road borders are detected by using a histogram where the texture orientations and color information are analyzed. For histogram generation, the values of angular difference (AD) and color difference (CD) are calculated for each voter. The road area is defined as a region between the two detected road borders and below the estimated vanishing point. The method was tested on 1000 road images which depict different roads with various variations in color, texture, lighting condition. Even the proposed method shows a high accuracy on authors' dataset, the performance of the algorithm was not

compared to the state-of-the-art methods. One of the drawbacks is that the algorithm requires tuning the parameters.

Similar work [11] has been published in 2017, where the researches introduced the multi-task network VPGNet for lane and road marking detection which is guided by a vanishing point under different weather and day/night-time conditions. VPGNet performs four tasks: grid regression, object detection, multi-label classification, and vanishing point prediction. VPGNet was trained on 20,000 labeled images (structured roads only) with 17 classes under four different scenarios: no rain, rain, heavy rain, and night. The results show that the VPGNet achieves the accuracy approximately 80% (it varies depending on the task) and robustness under various conditions in real-time (20 fps). During the training, authors flipped from left to right the original images in order to extend the dataset in two times. Also, it helped to simulate a lift-sided environment as the initial dataset contained only images from right-sided environment.

A big contribution to the field of road detection has been made by J. Alvarez, A. Lopez, and co-authors which resulted in a series of articles.

In [12] in 2011, they introduced a novel approach based on combining illuminant-invariant feature space with a likelihood-based classification. The main benefits of proposed approach are such as the algorithm does not depend on road shape or temporal restrictions; benchmarks show the good results on detecting both, shadowed and lighten, road areas. As for shortcomings, there are such as the technique mainly relies on analyzing the color and may not perform well for detecting roads in rural areas (not asphalted roads) where the road may contain a mix of colors; some of the areas which are over or under-saturated may not be detected; technique could not detect the road area when the lane markings were too big. Also, sometimes the objects with color that is similar to road color may be misclassified.

Later, in 2012, they have published another work [13] where they moved from parametric model to ensemble of CNN models, and they mainly focus on semantic segmentation. The algorithm first extracts learned features at multiple scales and multiple resolutions and then, fuses them at pixel level using a weighted linear combination. Features and weights are learned offline directly from training data. Their approach shows the accuracy 93.5% for road detection task on CamVid dataset, hence this model outperformed other fusion methods with fixed rules at that time. Besides the road detection, they also tried to classify the trees, sky, cars, buildings, sidewalks, and other objects, but the accuracy for these types of objects was lower oftentimes.

Based on prior knowledge, they published the additional paper [14] in 2014. This paper analyzed the performance of several approaches for detecting the road, namely contextual cues, including horizon lines, vanishing points, lane markings, 3-D scene layout, road geometry using geographical information combined with a navigation system. Moreover, they built and tested different ensembles of models that eventually showed the overall performance growth (for example, ensemble that includes the analysis of color, horizon line, vanishing point, and road shape). Furthermore, a novel approach for road

recognition has been introduced where the information about the objects from geographic information systems (GIS) has been used together with projecting the road map onto the driver's view. They also tested an ensemble of models in video sequences together with Kalman filter that keeps track of the previously detected road. They report that sometimes the errors may be propagated from previous frames (for example, the shadow from other vehicles or pedestrians). Also, the experiments have only been conducted on the structured roads, hence the output of the proposed techniques for unstructured roads is unknown.

Morphological operations.

In 2017, the authors from [15] proposed a set of methods based on the mathematical morphological operations for road detection problem such as erosion and dilation operations (Figure 1). To be more specific, these morphological operations are applied for denoising after the grascale image has been segmented by 2-D Otsu adaptive threshold method into road/non-road regions. Further, the LOG-operator is applied to detect the edges. Finally, Hough transform is performed to detect and mark the road boundaries. As a result, the algorithm works properly only on the roads with clear boundaries but does not work well in the case of shadows and obstacles.

Figure 1. Example of morphological operations. Left: original image. Middle: dilated image. Right: eroded image.[1]

Patch-based and image segmentation techniques.

In 2012, the researchers from [7] proposed a visual road detection system based on feeding patches from the image of fixed size to multiple artificial neural networks that can identify the road based on color and texture. The system extracts dominant features from different color spaces (RGB, HSV, YUV) through analyzing such values as average, entropy, energy, and variance. Also, the model is able to estimate the classification and the confidence factor for each sub-image (patch). The ANNs used in the proposed system consist of a multilayer perceptron (MLP). The author reports that the proposed patch-based system performs better for longer paths with sudden lighting condition changes.

In 2013, Hung, Huo, Yu, and Sun from [16] proposed an analogous approach by operating under image patches instead of the whole image at once. Interestingly, the researchers chose to use YcbCr color space inasmuch as"Y" color component, which represents a luminance, contains the most information among all color channels. Further, they calculate a standard deviation of each patch, find the maximum difference between them, and use it as a threshold later. Then, remaining patches are analyzed on brightness for

[1] OpenCV documentation on morphological operations. Accessed: 20/5/2018. https://docs.opencv.org

detecting road area. Finally, they apply a set of post-processing procedures such as erosion, dilation, resizing. As a result, the authors report a high accuracy, approximately 90%. However, all tests have been conducted on a single dataset with structured roads provided by the authors, which is not publicly available.

In 2015, Brust, Sickert, Simon, Rodner, and Denzler released their open-source CN24 framework for semantic segmentation based on processing patches and published related work [17] where they describe experiments on the KITTI and LabelMeFacade datasets. The presented results show that the authors were able to achieve state-of-the-art results. Also, the benchmarks show that sometimes the results were better than the techniques which use stereo data. We also noticed that author reports a low processing speed where the segmentation of single image takes approximately 30 seconds.

Additionally, another patch-based network has been described in the paper [18] in 2016 which is similar to [7], [19]. Furthermore, the authors of this article claim that their network can be used for real-time processing. Also, after testing the network on different size of patches (contextual window), they report that their network works better for larger contextual windows (50x50 and 66x66 pixels) and shows the accuracy 93.9% on KITTI dataset that stays in line with other state-of-the-art methods while maintaining real-time processing. One of the limitations of proposed approach is the inability to correctly classify different types of road surfaces or regions under extreme lighting conditions.

In a similar way, another work [19] has presented a patch-based network in 2017. In this paper, the authors have used a pre-trained SegNet convolutional network (encoder-decoder architecture). The model has been trained on 3,433 images of different types of the road under varying illumination. The choice of SegNet was made due to its state of art performance and the availability of the pre-trained model. Besides using SegNet for modeling road texture, researchers also used another technique named “color lines model” for learning different illumination conditions in the road image, based on conditional random field (CRF) framework. The combined model was validated on three datasets: KITTI, CamVid, and the dataset of the roads in India collected by authors. The results on KITTI shows the combined model proposed by authors outperforms the SegNet model, whereas the benchmark on CamVid dataset is basically the same as the results of SegNet predictions.

Combined approaches.

In 2015, Xiqun Lu presented a work [8] where he tried to detect road through applying road masks to image, removing everything above the horizon line, modeling the road surface with a multivariate Gaussian model which was tested on the Sowerby and CamVid datasets. The results show that the proposed approach outperforms [9], [12] on the given datasets. However, the algorithm failed on highly curved roads, T-junctions.

Y. Li, W. Ding, X. Zhang, and Zhaojie Ju presented an alternative algorithm for detection of various types of road including the scenes with objects such as vehicles and pedestrians in [20] in 2016. The proposed algorithm firstly segment the image based on the

dark channel together with K-means clustering method. Then, the road region is extracted by detecting the vanishing point and the soft voting rules proposed by the authors. On the final phase, a set of post-processing methods are applied. The key advantage in proposed method is the processing speed for road detection which is approximately 40-times faster than well-known approaches. However, the proposed algorithm did not work well in some scenes, where the detected road region was out of road boundaries due to similar color of surrounding regions in the dark channel image. The proposed approach also failed due to such factors as the reflection of light, refraction of rainwater, and shadows; thus, some road regions were missing or over-detected. Besides, the method did not work properly on the images that contain many unstructured roads due to the wrong detection of vanishing point and shadow.

Further, in 2016, the researchers from [21] introduced a novel technique for map-supervised learning which does not require a human effort for image labeling. For automatic annotation generation of drivable road area, the authors use localization sensors, GPS and inertial measurement unit (IMU) on the vehicle, and publicly available OpenStreetMap data. Firstly, the authors use OpenStreetMap data and vehicle position for reconstructing the 3D scene around the vehicle. Secondly, they project the reconstructed 3-D scene onto the image taken from camera. Thirdly, they try to reduce the label errors depending on the pixel appearence. In order to achieve robustness from shadows, the technique only uses the H (hue) and S (saturation) channels from HSI color space, and Cb (blue-difference component) and Cr (red-difference component) channels from YCbCr space. Once, the labeling is done, they trained a Fully Convolutional Neural Network (FCN) for road detection using the generated annotations.

The results from [21] show that the proposed technique with and without refinement reached the accuracy 86% and 80% accordingly on KITTI dataset on automatically labeled images. Meanwhile, the proposed fully supervised method where the training includes annotated images by the human reaches the accuracy 91%. The similar methods such as GRES3D+SELAS, RES3D+VELO [22] which evaluates information from other sensors, reach the accuracy of approximately 82%. However, current work does not present results on data with different environments such as extreme weather conditions (rainy or snowy roads).

In 2017, A. Narayan, E. Tuci, F. Labrosse, M. Alkilabi has published a paper [23] where they described their road detection experiments through using different color models (RGB, HSV, YUV, YCbCr, LAB, CbCra) and two convolutional neural networks (light CNN and modified AlexNet). The benchmark results have shown that the deep-convolutional neural network (AlexNet) was able to perform equally in comparison with the adaptive statistical color-based method, and sometimes even better in the off-line road detection tests. Besides, the light CNN (LCNN) can achieve similar accuracy compared to AlexNet network. Finally, the authors used a trained convolutional model to navigate a Pioneer 3-AT robot on different paths. The robot successfully reached the end of the road in 23 out the 25 trials.

2.3 Road image datasets

It is remarkable that the KITTI dataset is widely used in various papers related to road detection due to its quality. The dataset includes two types of data: high-quality images and sensor data captured by LiDAR. Additionally, the authors of KITTI dataset provide an evaluation tool to validate the correctness of the predictions. Furthermore, the authors published the raw video sequences which can be used for further analysis.

The authors of KITTI dataset also published a series of papers [24], [25], [14] related to road detection. Other researches, such as [26], used the dataset for analyzing and building the model based on the stereo data. Additionally, [18], [17] used KITTI dataset for constructing a CNN and FCN, whereas [4] used the dataset for the color-based road detection.

Another well-structured dataset is iRoads which has been classified into 7 classes: daylight; night; rainy day; rainy night; snowy; sun stroke; tunnel. However, this dataset is missing labels which limits the researches to observing only qualitative results.

Furthermore, one of the largest datasets that are currently available is NEXET which contains 50,000 images for training and 5,000 images for testing. Unfortunately, this dataset contains only the labels for vehicle objects in the scene, hence this dataset can be used only for qualitative analysis in our work.

Dataset	Total number of road images	Structured roads	Other types of roads	Different weather conditions	Different daytime conditions	Labels for road	Video	Used in article
ROMA [27]	116	+	-	-	-	-	+	
KITTI [28] (raw video [29])	612	+	-	-		+	+	
iRoads [30]	4,656	+	-	+	+	-	-	[6]
SUN [31]	426	+	+			-	-	
CamVid [32]	701	+	-	-	+	+	+	[8]
LabelMeFacade [33]	945	+	-	-	-	+	+	[17]
NEXET [34]	55,000	+	+	+	+	-		
Mapillary [35]	25,000	+	+	+	+	+		
Dataset provided by Amnir Hadachi	1,780	+	+	+	-	-	-	
ImageNet - collected by the author [36]	494	+	+	+	+	-	-	
Dataset collected from Google, Yahoo, Flickr images by the author [36]	5,390	+	+	+	+	-	-	

2.4 Conclusion

In the first part of this chapter, related works toward road detection have been summarized which base on analyzing colors, processing patches, horizon line detection, vanishing point detection, edge detection, neural network for classification/scene segmentation tasks, and the ensemble of models.

In the second part, we described the main set of available datasets used for road recognition and scene segmentation tasks.

3 Road detection and classification algorithm

3.1 Introduction

We have started from analyzing color spaces of road and non-road images (appendixes 1, 2, 3). As can be seen from images, in some color spaces the information about the target features can be lost, such as grayscale, HLS. It is also noticeable that L- channel (lightness), from HLS color space, represents the shows boundaries of the road more clearly than other color spaces and color-channels oftentimes. However, when we look at L-channel, we lose the information about colors and image becomes similar to grayscale image. In our research, we have decided to use only RGB color space.

In computer vision, people widely use the CNNs (Convolutional Neural Networks) for image classification, object detection, image segmentation tasks. This type of neural network will also be used in our experiment. Example of the structure of CNN for classification task can be seen in Figure 2.

In our experiments, we use Keras 2 (library for deep learning) and Python 3.6.3.

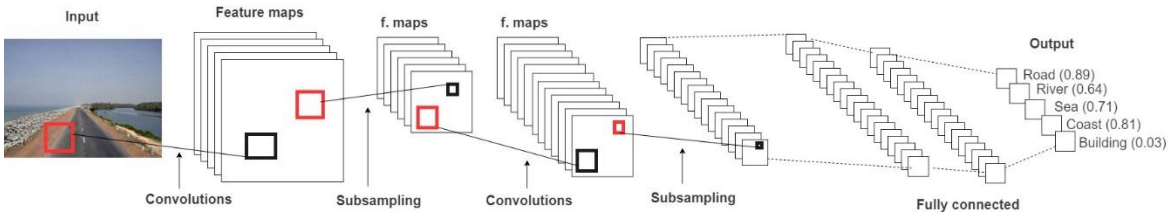


Figure 2. Example of the structure of CNN (Convolutional Neural Network) for classification task.

As an evidence, the paper [1] is considered as one of the breakthroughs in computer vision during the last decade. In this article, the researchers used a large deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet ILSVRC-2012 contests into the 1000 different classes. Their approach performed extremely well and they achieved a winning top-1 and top-5 error rates of 17.0% and 37.5% accordingly using purely supervised learning.

The number of trainable parameters is dependant on several factors such as the structure of the network, the size of input image, and the number of target classes which have to be recognized.

In our experiments, we split the input data into training (80%) and testing (20%) data sets and set the Adam optimization algorithm (optimizer) for our networks. Also, we always normalize the input images to the range [0, 1] instead of [0, 255].

To prevent overfitting and reaching better performance, we apply "data augmentation" techniques. By applying data augmentation to our input data, we generate additional samples as input to our network, thus the network learns features better from various similar images than from a fixed set of images. We enable performing the rotation of input images on +=30

degrees, horizontal flip, zooming (+-20%), applying shearing transformations. Also, there are other operations available for data augmentation such as adding noise, changing lighting conditions, perspective transform.

3.2 General overview of structure of convolutional neural network

Besides Figure 2 shows a typical CNN, the represented network is also called fully-connected neural network (FCN) due to the fact the network has at least two fully-connected (dense) layers at the end which represent the layer structure of multilayer perceptron (MLP) artificial neural network.

CNN architecture for classification tasks can be summarized as following:

- Receives a vector of images as input layer and pass it further to hidden layers (forward-pass).
- Hidden layers perform a set of mathematical operations for extracting low-level features through applying convolutional filters, activation functions, max-pooling operation, resizing the layer, transforming layer to fully-connected layer. The neurons inside specific hidden layers such as (de)convolutional) receive the weights which affect the output of the neuron (Figure 3).
- In the end of hidden layers of CNN for classification problem, there is one additional activation layer which returns probabilities for predicted classes for every pixel. It can be either softmax or sigmoid activation function.
- Finally, the network compares the predicted result with actual class of every pixel through certain validation function and according to positively/negatively predicted result, apply an optimizer function to correct the weights in the network through one backpropagation iteration. Usually, the validation function is MSE (1).

$$\text{Mean square error, } MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \quad (1)$$

The example of cell of artificial neural network is shown in Figure 3 that computes a dot product of their weights with the input.

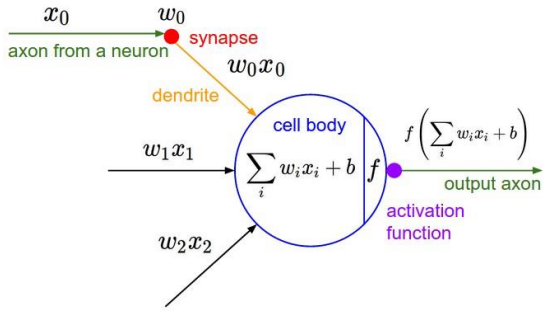


Figure 3. The structure of artificial neuron in the NN[2].

[2] Course CS231n Convolutional Neural Networks for Visual Recognition in Stanford Univeristy. Accessed 5/5/2018. http://cs231n.github.io/convolutional-networks/

In general, the learning of the network works through adjusting the weights which are the parameters of the network layers and hence, affects the output of the network. In the begging, the weights are initialized randomly and after the network computed the output with these weights, it can calculate the loss function which shows the error between actual and predicted values. Further, we propagate the error back in order to correct the initial weights. One iteration of the feed-forward and back-propagation operation is called an epoch. Depending on the optimizer which calculates the derivative of loss function, the network is able to minimize the error faster or slower. The weights are updated based on a delta rule formula:

$$W_{new} = W_{old} - LR \times DR \quad (2)$$

where W represents the weights, LR - learning rate, DR - partial derivatives of loss function.

While constructing the convolutional neural network, different types of layers can be used such as (de)convolutional, (un)pooling, fully-connected (dense), data normalization, activation layer.

Convolutional filters. The convolutional filter consists of applying matrix filter (kernel) to the receptive field (the region of target pixels). It includes two mathematical operations: dot multiplication between kernel and receptive field's values; sum the received values. The output of result of convolutional layer is also called a feature map. Example of performing convolution on input data with 3 dimensions (RGB) is shown in Figure 4.

Interestingly, the convoluted data from the first convolutional layers in the networks may detect low-level features, for example, lines, edges, and curves. Further, when the network becomes deeper through max-pooling operations and has more convolutional layers, the deeper convolutional layers can be more complex as they are built up to the already discovered filters. Usually, the CNN has more than one convolutional filter that ends up in a series of convolutional layers. For every convolution filter, we must specify the filter size (this is how many pixels we want to supply to filter). The example of output of convolutional layer with low-level features is shown in Figure 4 (right). This source[3] contains an interactive visualization of image kernels.

[3] Visualization of image kernels. Accessed 6/5/2018. lihttp://setosa.io/ev/image-kernels/

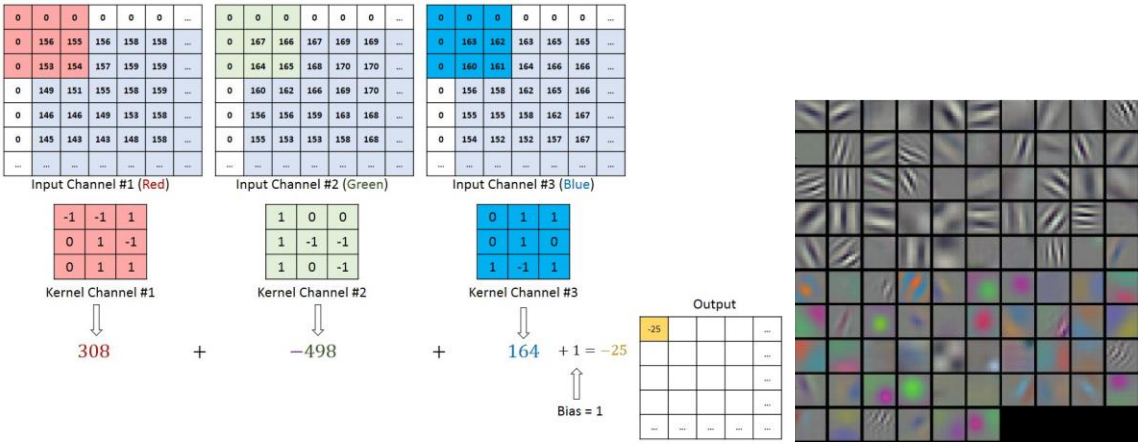


Figure 4. Left: example of multi-channel input convolution 4. Right: Example of output of convolutional filters.

It is important to note that kernels in convolutional layers are the weights which are trainable parameters and they are calculated during the learning of the network through backpropagation.

There are several optional parameters which we can supply to (de)convolutional filter (animated visualization of convolutional operations - [37]) such as stride and padding.

Stride (Figure 5) is the number of pixels by which the filter shifts every time during convolution (by default the shift is one unit at a time).

Padding (Figure 5). In CNNs we move square filters around the image, but we cannot go all the way to the edges of images if there is no padding, since part of the filter would be outside the image, that is why we can apply additional padding which will extend the margins of input tensor and fills those margins with zero-values.

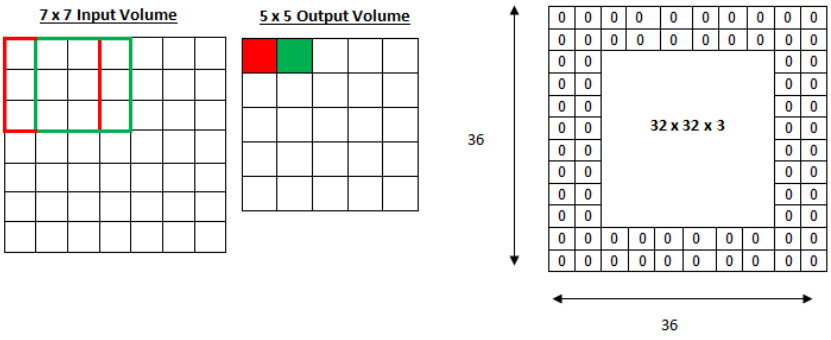


Figure 5. Example of stride of size 1 which is default (on the left). Zero-padding of size 2 (on the right).

Activation function/layer (Figure 6). It is common practice to apply activation function after convolutional layers in order to transform the output values in desired form

[4] Undrestanding Convolutional Layers in Convolutional Neural Networks. Accessed 6/5/2018. machinelearninguru.com/computer_vision/basics/convolution/convolution_layer.html

or modify the values through specific (activation) function. In brief, the layer with activation function basically maps the resulting values depending on the function. In case of ReLU activation function, the input values will be passed through function (3) that basically transforms negative values to zeros.

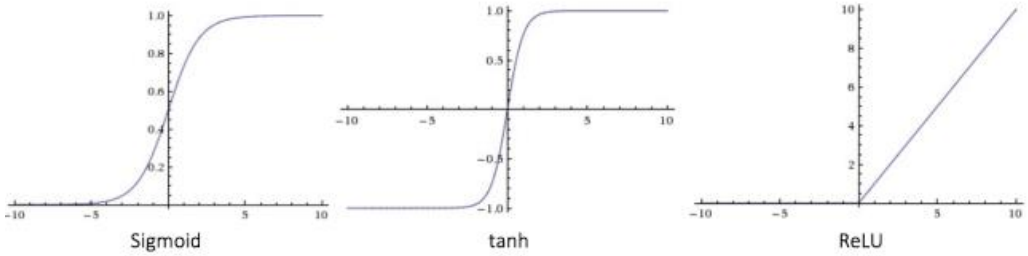


Figure 6. Main activation functions [38].

$$\text{ReLU activation function} \quad f(x) = \max(0, x) \tag{3}$$

Max-pooling (down-sampling) layers (Figure 8). After activation function, pooling layer can be applied to the output of any layer (usually, it is applied after the activation function). In brief, it takes a window (usually, it has size 2x2), and takes the maximum value in this window. This serves two main purposes. The first is that the number of parameters or weights is reduced by 75%, thus lessening the computation cost. The second is that it will control overfitting.

Deconvolution layer also called "transposed convolutional layer" or "fractionally strided convolutions" (Figure 7). This layer is widely used for the pixel-wise classification task where the feature maps have to be restored to original input image size. Its aim is to upsamples the image to get the same resolution as the input image. In case of doing resizing operation as we can do with any image, we will loose the important details which will produce incorrect result. But deconvolutional layer is similar to "resize" operation, where we can train the parameters for preserving the lose of information [39].

In general, the deconvolution is an opposite operation to convolution. That is why, the input of convolution becomes the output of deconvolution, and the output of convolution becomes the input for deconvolution.

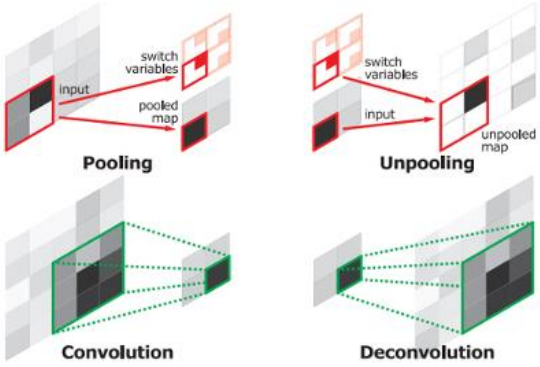


Figure 7. Illustration of deconvolution and unpooling operations [39].

Unpooling layer (Figure 7 and Figure 8). This operation performs the reverse operation of pooling and reconstructs the original size of activations. In other words, when the network processes each max-pooling layer, they remember the locations of values chosen by each max-pooling layer. Later, the network uses these pooled maps during the un-pooling operation to retrieve back the value previously chosen by max-pooling layer at initial location in the tensor.

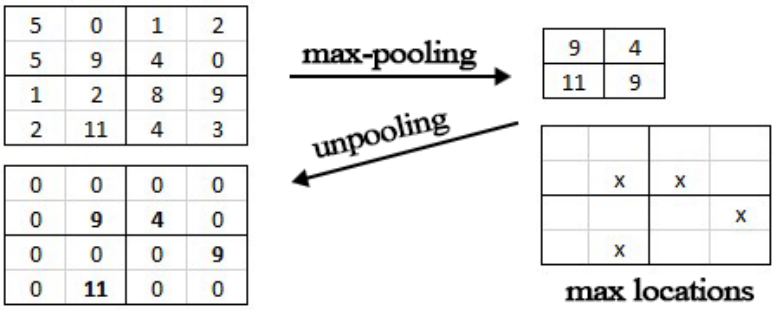


Figure 8. Example of unpooling operation after performing max-pooling operation.

Batch-normalization layer. This layer normalizes the output of a previous layer. It subtracts the mean and divides by the standard deviation. It helps to speed up the learning of neural network.

Dropout layer. To prevent overfitting, the researchers add dropout layers in their networks which deactivate a certain amount of neurons for specific layer during the training.

3.3 General image classification

To classify whether the input image contains the road or not, we used layers from LeNet-5 CNN that has three consecutive convolutional layers for sub-sampling with max-pooling layers, and two fully-connected layers in the end.

LeNet 5 is 7-level convolutional network that has been created for classification the digits from 32x32 images by Y. LeCun in 1998 (Figure 9). Furthermore, for processing the images with higher resolutions, the network must be larger and deeper with more convolutional layers.

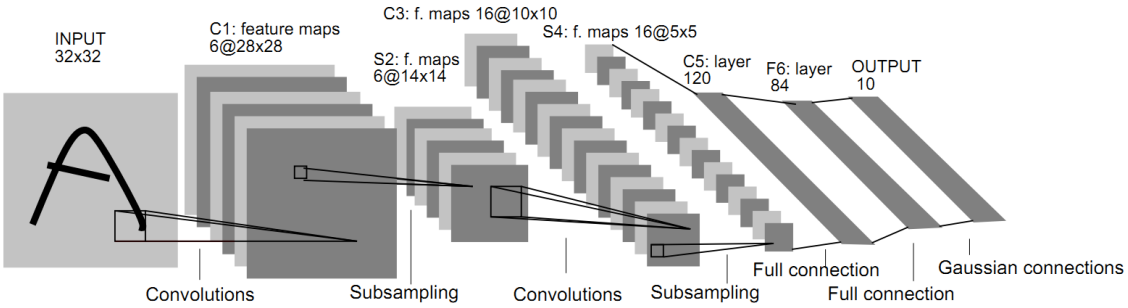


Figure 9. Architecture of LeNet-5 CNN for digit recognition [40].

The input parameters for our network (Figure 10) are the following:

- Size of input RGB images: 50x50.
- Number of epochs: 25.
- Batch size: 25 (the number of images that will be processed within 1 iteration).
- Initial learning rate (learning step): 0.001.
- Convolutional filter size is (5x5), padding is always the same.
- Total number of trainable parameters with current NN structure: 3,628,072.
- Classes: 2 (road; non-road image)

It is important to note that the first fully-connected (dense) layer #6 with 500 nodes contains the biggest number of trainable parameters (3,600,500) in comparison to other layers. This big number appears due to the preceding flatten layer #5 that has size 7,200 nodes which flattens the max-pooling layer #4 (12x12x50).

Additionally, we apply the “softmax” activation function as the last layer in the network in order to receive a confidence map for every class, and the label with the highest confidence has been chosen as the predicted label. As an alternative, we could use “sigmoid” activation function which can be used only for two-class forecast.

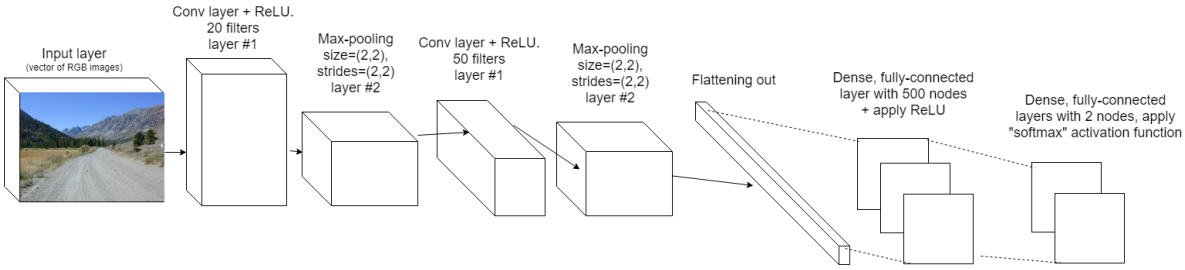


Figure 10. Structure of our CNN for road image classification which is based on LeNet network.

3.4 Pixel-wise road recognition

First of all, the pixel-wise road image recognition means that we want to predict the class for every pixel. This task may look similar to the general classification of image, however, the network output for image recognition task should be the matrix of the size of initial image size where each pixel associated with specific label. In computer vision, this procedure is known as "semantic segmentation" task. Since we have only two classes (road, non-road pixel), in our case the network output is a binary matrix where 0 is non-road pixel and 1 is road pixel.

For pixel-wise classification, we use two models: a convolutional autoencoder network and modified FCN-8 model [41].

Autoencoder network. Autoencoder is the type of neural networks with feed-forward propagation where the output of the network is restored input data. Auto-encoders are designed in such a way that they cannot accurately copy the input to the output due to the constraints that the next layer should have a smaller size than the current layer or they are penalized for activations. That is why, the input data is restored with errors due to encoding losses, but in order to minimize losses, the network has to learn to choose the most dominant features. The general schema of autoencoder network is shown in Figure 11.

Autoencoders are composed of two components: encoder g and decoder f .The encoder transforms the signal (input data) into its compressed representation: $h = g(x)$, whereas the decoder restores the signal from compressed representation: $x = f(h)$. By changing f and g, autoencoder tries to fit $x = f(g(x))$ by minimizing the error function:

$$L(x, f(g(x)) \tag{4}$$

By itself, the ability of autoencoders to compress the data is rarely used, since they usually work worse than human-written algorithms for specific types of data such as sound records or images. Having trained the autoencoder on the digits, it cannot be used to encode other types of data (for example, human faces). That is why it is important that the data must belong to the similar group of data on which the network has been trained.

However, the autoencoders can be used for unsupervised learning for the classification task with applying backward error propagation. Additionally, they are used for dimensionality reduction or for learning the most useful features of the input data.

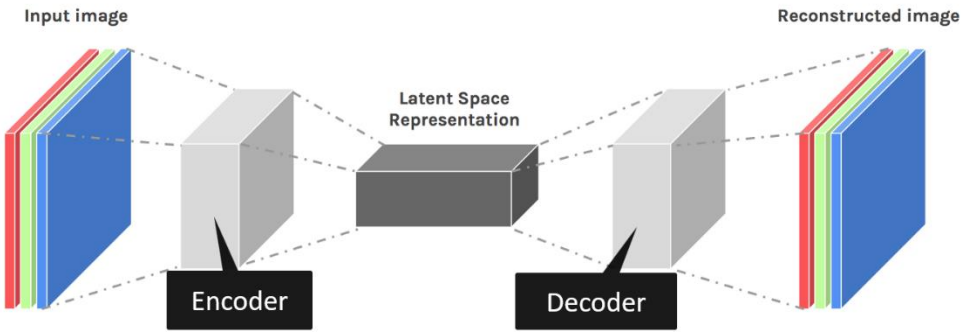


Figure 11. Generalized structure of autoencoder network.

In our experiment, we use SegNet architecutre proposed in [42] for image segmentation. As can be seen from the SegNet autoencoder architecture (Figure 12), there are no fully-connected layers and upsampling layers have been used in decoder layers. The final decoder output feature maps are fed to a soft-max classifier for pixel-wise classification. In the paper [42], the authors tested the SegNet architecture on 37 indoor scene classes, however, in our research we apply this network only to two classes (road, non-road).

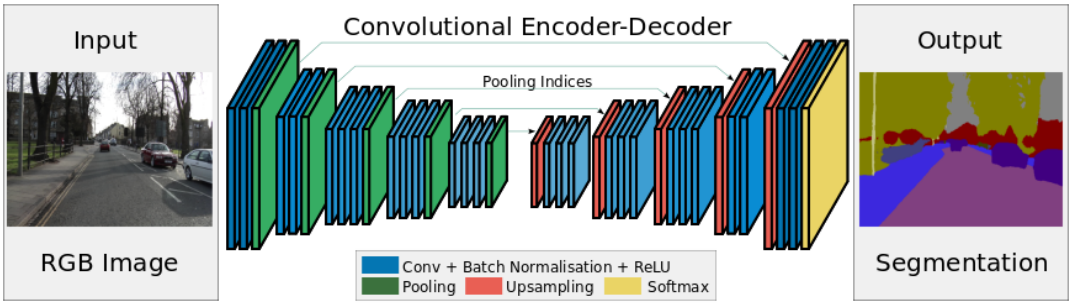


Figure 12. The architecture of SegNet autoencoder network [42].

FCN-8 network. FCN-8 is convolution neural network built on top of VGG-16 network introduced in [41]. The key difference between FCN-8 and VGG16 is that the first netowork does not contain fully-connected layer that speeds up the learning process and include skip-connections. Skip-connection allows to skip some layer in the network and feeds the output of specific layer to layers on deeper levels by skipping certain layers. The general intuition for using skip-layers is that some information was captured in the initial layers and needs to be used during the reconstruction (up-sampling), otherwise, this information would have been lost or distorted.

As can be seen from the structure of FCN-8 on Figure 13, there have been added three skip-connections to the pool3, pool4. Since these pooling layers are further used in the network, they operate on low-level features and able to capture finer details.

Additionally, the researchers from [41] merge skip-connection in FCN-8 in the middle and at the end of the network (there are two such layers in FCN-8). In FCN-8 the merge operation is “sum”, however, there are different set of operations available on the input weights such as average, max, multiplication, and others.

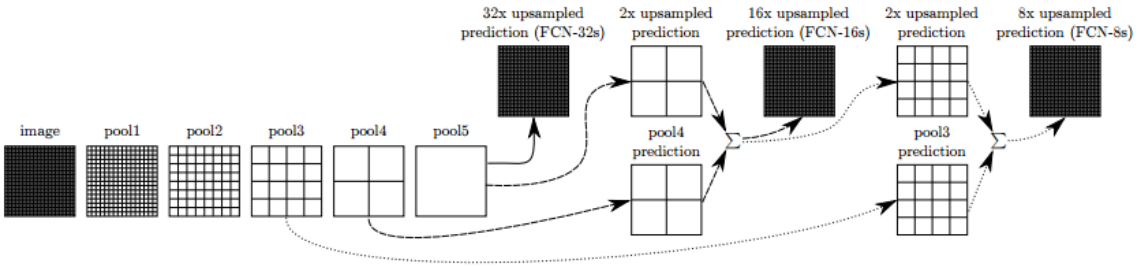


Figure 13. Network architecture of FCN-8 with skip-connections [41].

Our network has a similar structure to FCN-8 and contains 28 layers including 13 convolutional, 5 max-pooling, 3 deconvolution, 3 skip-connection, 3 cropping layers and additionally one merge, reshape, permute, „soft-max“ activation layer. The key difference betweet our network and FCN-8 is that we have only one merge layer without intermediate merge. Our network is shown in Appendix 4.

3.5 Conclusion

At the beginning of this chapter, we described the main types of layers of CNN such as convolution, max-pooling, activation and other layers.

Further, we illustrated the original LeNet-5 network for digit recognition and introduced our LeNet-5-based network for general image classification.

Finally, we described two our networks for pixel-wise road classification which are based on encoder-decoder and FCN-8 with skip-connections architectures accordingly.

4 Results and discussion

4.1 Introduction

This chapter discusses the results of several experiments conducted on the CNN models from chapter 3. These tests are related to both tasks, general image classification and pixel-wise image classification.

In order to be assured that our models are resistant to different environments, we test them on different datasets. Afterward, we provide the quantitative and qualitative results of our experiments.

4.2 Image classification

We have decided to use different objects and environments which surrounded by nature which and downloaded 1,000 non-road images from Google.

We trained and validated our classification model (Figure 10) on different datasets, our results are listed below.

Run #1. Training model on "KITTI" dataset [28] which contains 612 images of only structured roads.

- 596 training images (roads: 484 (81.21%), non-roads: 112 (19%).)
- 150 testing images (roads: 128 (85.33%), non-roads: 22 (15%)).

The results of 150 classified images are shown in the Table 1 and has high accuracy (98%). As can be seen, only 3 images were misclassified which means that the input data has a low entropy and the model may be overfitted during the training.

Table 1. Confusion matrix for model results on "KITTI" testing data set.

Actual / Predicted	Road	Non-road
Road	126	2
Non-road	1	21

Run #2. Training on "ImageNet" dataset images collected from ImageNet image repository [43] by the author using scrapper script. After cleaning up and classifying 1200 images by folders into 3 categories (structured roads - 319, unstructured roads 175, vehicle/object in scene 387) manually, we decided to ignore images with vehicle/object in the scene to simplify the task for our network. That is why we use only 494 road images from this dataset. Our data sets are split into the following sets:

- 502 training images (roads: 393 (78.29%), non-roads: 109 (22%)).
- 126 testing images (roads: 101 (80.16%), non-roads: 25 (20%)).

The accuracy of classifying 126 images reached 89% (Table 2) where only 13 images were misclassified. Some results of classification are shown in Figure 14.

Another test was conducted on this dataset where we tested trained model on non-roads images only which were not included in test/train data sets. The results of 850 classified images show the accuracy 89% where 86 images were wrongly classified, and other 764 non-road images were correctly classified.

Table 2. Confusion matrix for model results on "ImageNet" testing data set

Actual / Predicted	Road	Non-road
Road	93	8
Non-road	5	20

Figure 14. Results of classification on ImageNet dataset. Description of every row: True Positive non-roads; False Negative non-roads; True Positive roads; False Negative roads.

Run #3. Testing the previously trained model from Run #2 on 569 snowy road images from "iRoads" dataset [30]. The accuracy reached 78.38%, where 123 road images were misclassified. The image samples are shown in Figure 15.

We also decided to re-train our model on this dataset but increase the image size to 100x100 pixels. As a result, our accuracy increased to 80.14%.

Figure 15. Image samples from “iRoads” dataset.

Run #4. As we mentioned earlier, in the sections “road datasets” from chapter 2, we created a new dataset where approximately 5,300 road images were collected from Google, Yahoo,

Flickr image search engine [36]. The image samples of every class of road from our dataset are depicted in Figure 16.

Before we trained our classification model on this dataset, we split our data into the following sets:

- 5,427 images in training set (roads: 4,311 (79.44%), non-roads: 1116 (21%)).
- 1,357 images in testing set (roads: 1,075 (79.22%), non-roads: 282 (21%)).

To sum up, we decided to validate 1,357 test images within every class (the results are shown in Table 3) that gave the overall accuracy 94.18%, where 1,278 images were classified correctly and only 79 were misclassified.

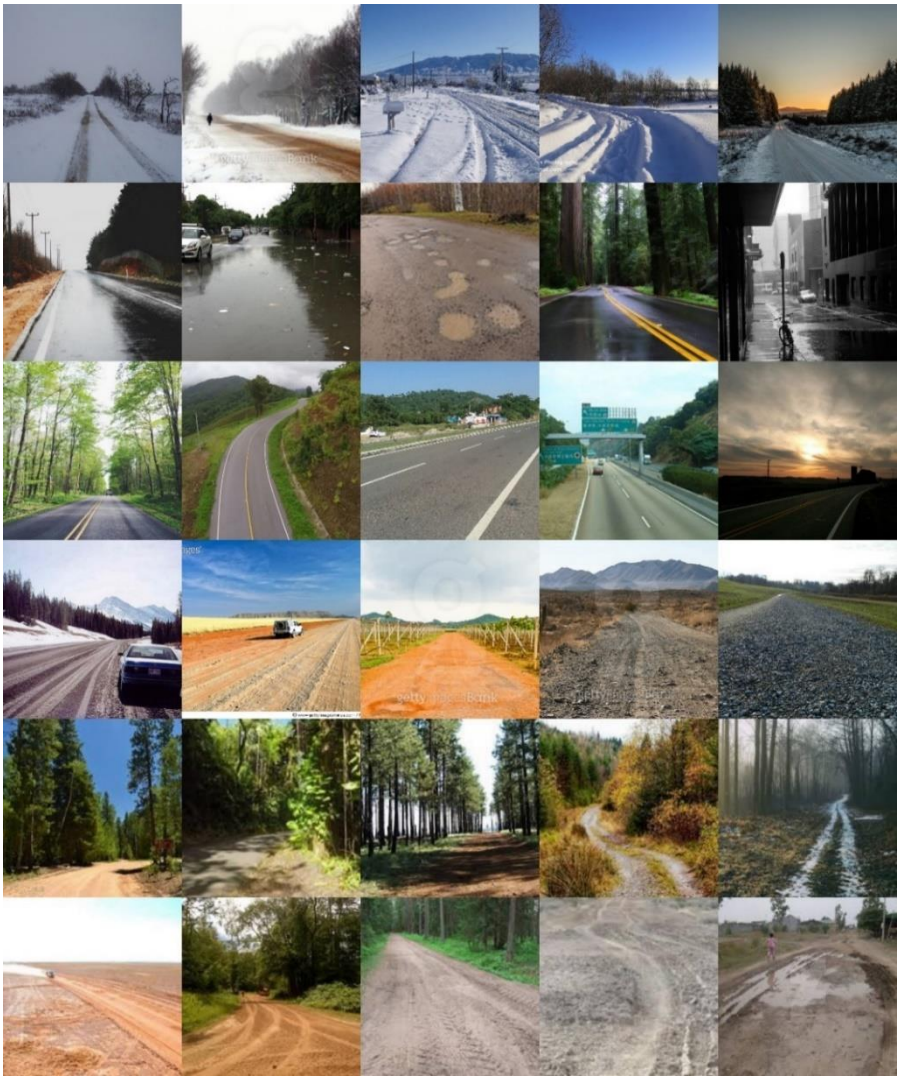


Figure 16. Example of images from our dataset line by line: snowy, rainy, highway, gravel, forest, dirt roads accordingly.

Table 3. Confusion matrix for test dataset on our dataset among 7 classes.

Actual / Predicted	Road	Non-road	Accuracy
Snowy	132	18	88.00%
Rainy	118	2	98.33%
Gravel	149	8	94.90%
Forest	174	1	99.43%
Dirt	256	7	97.43%
Highway	204	6	86.88%
Non-road	37	245	86.88%

4.3 Pixel-wise image classification

Initially, we tried to run SegNet model on CamVid and KITTI datasets separately. We cropped by the center and resized the images to 480x160 pixels. The number of training epochs was set to 25.

For calculating the accuracy, we use the following fomula:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

where TP – True-Positive, TN – True Negative, FP – False Positive, FN – False Negative rates.

In the predicted images we use the following colors:

- Green: road, True Positive
- Purple: road True Negative
- Red: road False Negative
- White (or transparent): road False Positive

The results tested on SegNet network were not satisfactory. There was a case when the network focused only on one specific image, and all other predictions were the same. That is why we added dropout layers which prevent overfitting through deactivating random number of neurons specified by user. It helped only partially, since the network started producing different predictions, however, the accuracy was still low, about 10%. The failure results are shown leheküljel 32.

Also, we noticed that the network could not reconstruct training images as well. Unfortunately, we could not figure out what was exactly the problem.

Another experiment was conducted on our proposed model depicted in Appendix 4, based on FCN-8 architecture. Also, we resize the input images to 224x224 pixels, set the number of epochs to 50. The number of trainable parameters is 9 million and a half which is 3 times larger than FCN network used for image classification on 50x50 pixel images.

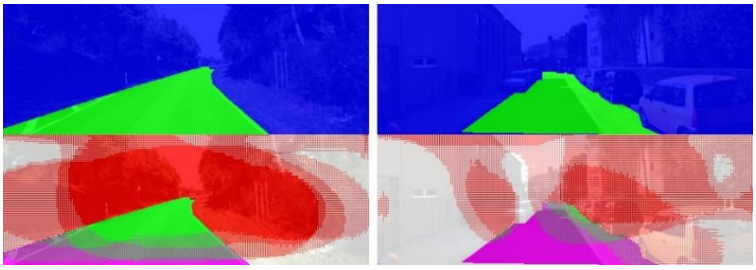


Figure 17. SegNet model results predicted on test data from KITTI dataset (ground-truth – first line, predicted images – second line).

Figure 18. SegNet model results predicted on test data from CamVid dataset (ground-truth – first line, predicted images – second line).

While training the model on the Rocket cluster provided by the University of Tartu, the training process failed several times due to memory exceeding, cluster task timeout problems. Later, we set memory allocation for our task to 10 GB and the task timeout was set to 17 hours. Besides, we added intermediate save of the model after every epoch, so if the network fails, in the end, we still have results and ability to resume the training from the latest saved model.

Our FCN-8 based network trained on CamVid dataset has been tested on 100 images from the same dataset which was not involved in training/testing sets. The results are shown in Figure 19 and the average accuracy close to 89%.

Furthermore, we have validated proposed model on the test set of 289 RGB images from KITTI dataset with 3 skip-connection layers trained on CamVid dataset. Note that the initial panoramic images from KITTI dataset were cropped by the center and resized to 224x224 pixels. The average accuracy is 87%. The classified road images from KITTI dataset are shown in Figure 20.

Also, we noticed that the network oftentimes fails on curvy roads and on the vehicle detection in the scene for both, KITTI and CamVid, datasets.

The processing time for single prediction is approximately 900 ms. while running the script on the machine with CPU 2.2 GHz with 8 GB of RAM.

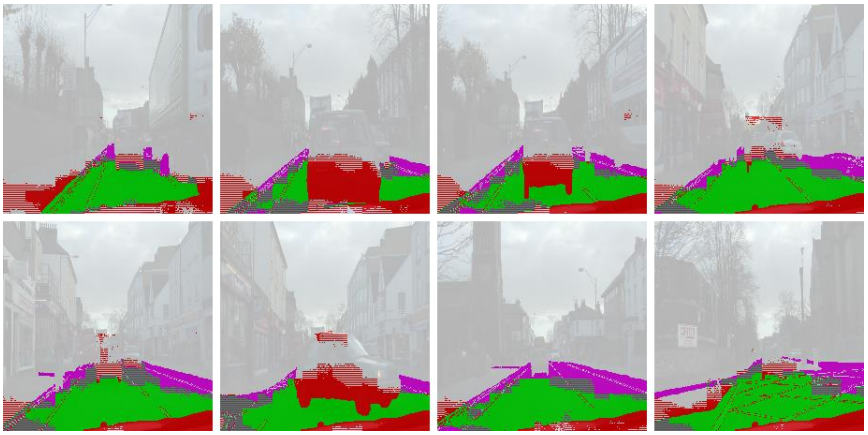


Figure 19. Results from our FCN-8 based convolutional network trained and tested on CamVid data set.

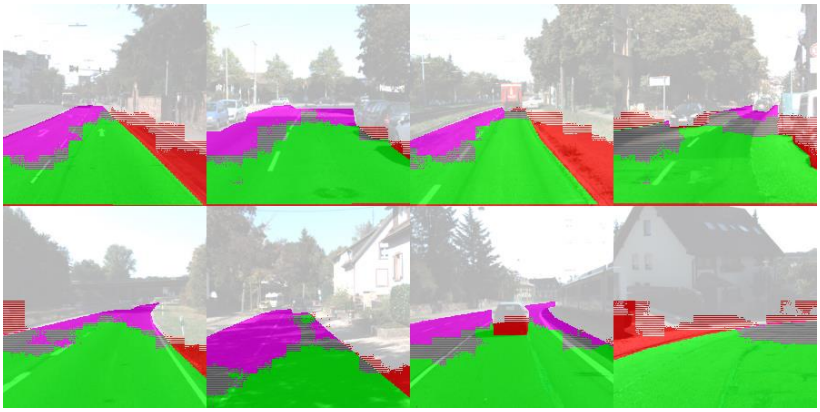


Figure 20. Results from our FCN-8 based convolutional network trained on CamVid dataset tested on KITTI data set.

Finally, we trained proposed FCN-8 based model on both, KITTI and CamVid, datasets and did a qualitative analysis on 200 random images from our created dataset [36]. Despite we did not apply any post-processing methods, we were still able to receive satisfactory results on different types of roads (except rural roads) which are shown in Figure 21. We also would like to share the wrongly classified road regions shown on Figure 22.

Figure 21. Satisfactory detected roads by proposed FCN-8 based model.

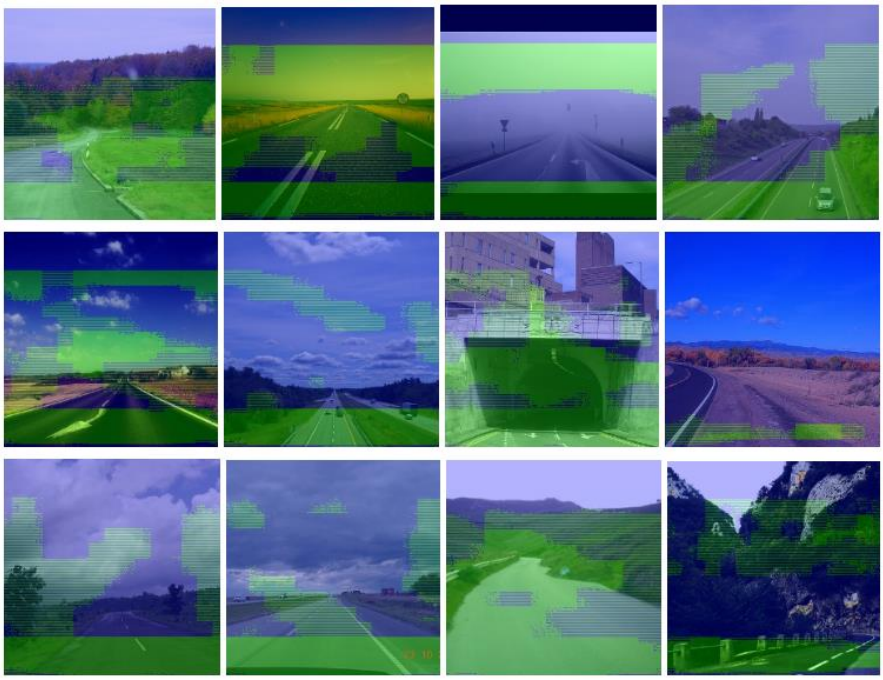


Figure 22. Wrongly detected roads by proposed FCN-8 based model.

4.4 Conclusion

In this chapter, we demonstrated the obtained outputs based on the models from chapter 3.

Firstly, we show the results of general image classification of CNN trained on 50x50 pixel images on different datasets such as KITTI, ImageNet, iRoads, and our own dataset with ~5,300 images. We noticed that due to the low heterogeneity of the data in the KITTI dataset, the network was overfitted that resulted in the high accuracy 98%. One of the solutions for this problem can be adding "dropout" layers to the network which deactivate a certain number of neurons. By contrast, the accuracy of other datasets reaches ~80-94% depending on the chosen dataset.

Secondly, we tested SegNet (encoder-decoder) model on CamVid and KITTI datasets. The classified pixels showed an extremely low accuracy ~50%, and when we ran the model on training data to validate the network is able to recognize trainable data, it still produced a low accuracy. Unfortunately, we could not figure out why the result was so imprecise.

Thirdly, the modified FCN-8-based model has been tested on several datasets. When the model was only trained on CamVid dataset, the accuracy was close to 89%. Meanwhile, the already trained model was tested on KITTI dataset that showed the accuracy 87%. It is important to note, we used 224x224-pixel images as training/testing data in order.

Finally, we report the qualitative results for our pixel-wise classification model trained on CamVid and KITTI dataset together and tested on 200 road images taken from different angles from our own dataset. From these results, we can see that model can detect the road, however, sometimes the network pays too much attention to colors and this may produce misclassification.

5 Conclusion

5.1 Conclusion

In this thesis, road classification and road recognition problems have been described.

In the first chapter, we discussed the necessity of road recognition techniques and its applications, general restrictions of current approaches, and the objectives of this thesis.

In the second chapter, already existing methodologies have been analyzed and summarized. Moreover, at the end of this chapter, we have summarized the information about available datasets and presented our own.

In the following chapter, we summarized the main types of layers used in CNNs and introduced our networks for road image classification and road recognition tasks. The results of experiments and their analysis have been described in the further section.

Also, we have to admit that the networks evolve through the time that results in more sophisticated and deeper architectures.

5.2 Future perspectives

From the results section (chapter 4), we can observe that some of road regions were either not detected or misclassified. To tackle this issue, we could apply such morphological operations as dilation and erosion during post-processing phase.

Furthermore, our trainable datasets did not contain unstructured roads due to lack of time and unavailability of such dataset. There are at least three options to solve this issue: 1) use a paid Mapillary dataset 5. 2) Annotate our created dataset [36]. 3) Use google maps or OpeenStreetMap for semi-supervised road image annotations.

Also, we assume that the use of other color spaces such as HSI, HSV, YUV, or YCbCr may help to reduce misclassification inasmuch the network will be able to ignore the sharp color difference between objects and learn convolutional filter more wisely.

Another limitation is the learning of 2D textures from images while the video sequences or another source of data (such as stereo data from LiDAR) can provide additional temporal information, and it can help to keep track of the previously detected road.

[5] Mapillary - map data from street-level imagery. Accessed: 10/5/2018. https://www.mapillary.com

References

[1] I. S. G. E. H. Alex Krizhevsky, „ImageNet Classification with Deep Convolutional Neural Networks,“ *In Advances in neural information processing systems (NIPS)*, p. 1097–1105, 2012.

[2] C. X. X. W. a. Z. Y. Z. Tian, „Non-parametic model for robust road recognition,“ *IEEE 10th INTERNATIONAL CONFERENCE ON SIGNAL PROCESSING PROCEEDINGS, Beijing*, pp. 869-872, 2010.

[3] T. Z. A. O. J. K. Z. Olusanya Y. Agunbiade, „Enhancement performance of road recognition system of autonomous robots in shadow scenario,“ *Signal & Image Processing : An International Journal (SIPIJ)*, kd. 4, nr 6, 2013.

[4] V. F. S. A. R. B. Wang, „Color-based Road Detection and its Evaluation on the KITTI Road Benchmark,“ *IEEE Intelligent Vehicles Symposium (IV)*, 2014.

[5] S. K. S. B. T. S. S. a. S. S. C. K. Roy, „Dehazing technique for natural scene image based on color analysis and restoration with road edge detection,“ *2017 1st International Conference on Electronics, Materials Engineering and Nano-Technology (IEMENTech), Kolkata*, pp. 1-6, 2017.

[6] M. T. Mahdi Rezaei, „Vehicle Detection Based on Multi-feature Clues and Dempster-Shafer Fusion Theory,“ *Image and Video Technology, Springer*, kd. 8333, pp. 60-72, 2014.

[7] V. G. F. S. O. a. D. F. W. P. Y. Shinzato, „Fast Visual Road Recognition and Horizon Detection Using Multiple Artificial Neural Networks,“ *2012 IEEE Intelligent Vehicles Symposium*, pp. 1090-1095, 2012.

[8] X. Lu, „Self-supervised road detection from a single image,“ *2015 IEEE International Conference on Image Processing (ICIP), Quebec City, QC*, pp. 2989-2993, 2015.

[9] J. Y. A. a. J. P. H. Kong, „General Road Detection From a Single Image,“ *IEEE Transactions on Image Processing*, kd. 19, nr 8, pp. 2211-2220, 2010.

[10] T. S. a. E. N. T. H. Bui, „Road area detection based on texture orientations estimation and vanishing point detection,“ *The SICE Annual Conference 2013, Nagoya, Japan*, pp. 1138-1143, 2013.

[11] J. K. J. S. Y. S. S. O. B. N. K. T.-H. L. H. S. H. S.-H. H. I. S. K. Seokju Lee, „VPGNet: Vanishing Point Guided Network for Lane and Road Marking Detection and Recognition,“ *IEEE International Conference on Computer Vision (ICCV 2017)*, 2017.

[12] J. M. A. A. M. Lopez, „Road detection based on illuminant invariance,“ *IEEE Transactions on Intelligent Transportation Systems*, kd. 12, nr 1, pp. 184-193, 2011.

[13] Y. L. T. G. A. M. L. J. M. Alvarez, „Semantic Road Segmentation via Multi-scale Ensembles of Learned Features,“ *Computer Vision – ECCV 2012. Workshops and Demonstrations,* pp. 586-595, 2012.

[14] A. M. L. T. G. F. L. Jose M.Alvarez, „Combining Priors, Appearance and Context for Road Detection,“ *IEEE Trans. Intelligent Transportation Systems (ITS),* pp. 1168 - 1178, 2014.

[15] Z. J. a. F. Z. X. Ming, „Research on unstructured road detection algorithm based on improved morphological operations,“ *4th International Conference on Smart and Sustainable City (ICSSC 2017), Shanghai,* pp. 1-5, 2017.

[16] C. L. H. Y. H. Y. a. T. Y. S. C. W. Hung, „Road area detection based on image segmentation and contour feature,“ *2013 International Conference on System Science and Engineering (ICSSE), Budapest,* pp. 147-151, 2013.

[17] S. S. M. S. E. R. J. D. Clemens-Alexander Brust, „Convolutional Patch Networks with Spatial Prior for Road Detection and Urban,“ *VISAPP, March,* 2015.

[18] V. F. D. F. W. C. C. T. Mendes, „Exploiting fully convolutional neural networks for fast road detection,“ *2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm,* pp. 3174-3179, 2016.

[19] S. P. C. A. a. S. B. S. Yadav, „Deep CNN with color lines model for unmarked road segmentation,“ *2017 IEEE International Conference on Image Processing (ICIP), Beijing,* pp. 585-589, 2017.

[20] Y. L. a. W. D. a. X. Z. a. Z. Ju, „Road detection algorithm for Autonomous Navigation Systems based on dark channel prior and vanishing point in complex road scenes,“ *Robotics and Autonomous Systems,* kd. 85, nr 11, pp. 1-11, 2016.

[21] M. K. K. L. E. N.-S. a. M. H. A. Laddha, „Map-supervised road detection,“ *2016 IEEE Intelligent Vehicles Symposium (IV), Gothenburg,* pp. 118-123, 2016.

[22] D. F. W. a. C. S. P. Y. Shinzato, „Road terrain detection: Avoiding common obstacle detection assumptions using sensor fusion,“ *2014 IEEE Intelligent Vehicles Symposium Proceedings, Dearborn, MI,* pp. 687-692, 2014.

[23] E. T. F. L. M. A. A. Narayan, „Road detection using convolutional neural networks,“ *Proceedings of the 14th European Conference on Artificial Life, ECAL 2017,* pp. 314-321, 2017.

[24] T. G. A. M. L. Jose M. Alvarez, „Online Road Detection by One-Class Color Classification: Dataset and Experiments,“ *arXiv:1412.3506v2 [cs.CV],* 2014.

[25] Y. L. T. G. A. M. L. Jose M. Alvarez, „Road Scene Segmentation from a Single Image,“ *Proc. European Conf. on Computer Vision (ECCV), Florence,* 2012.

[26] D. G. a. D. F. W. P. Y. Shinzato, „Road estimation with sparse 3D points from stereo data,“ *17th International IEEE Conference on Intelligent Transportation Systems (ITSC), Qingdao,* pp. 1688-1693, 2014.

[27] T. a. T. J.-P. a. N. P. a. C. P. Veit, „Evaluation of Road Marking Feature Extraction,“ *Proceedings of 11th IEEE Conference on Intelligent Transportation Systems (ITSC'08),* pp. 174-181, 12-15 October 2008.

[28] T. G. A. M. L. Jose M. Alvarez, „KITTI dataset,“ [Võrgumaterjal]. Available: https://rsu.data61.csiro.au/people/jalvarez/research_bbdd.php. [Kasutatud 22 04 2018].

[29] „KITTI raw data,“ [Võrgumaterjal]. Available: http://www.cvlibs.net/datasets/kitti/raw_data.php. [Kasutatud 22 04 2018].

[30] M. T. Mahdi Rezaei, „iRoads dataset,“ [Võrgumaterjal]. Available: https://www.researchgate.net/publication/260219141_iROADS_Datasetzip. [Kasutatud 22 04 2018].

[31] „SUN2012 dataset,“ [Võrgumaterjal]. Available: https://groups.csail.mit.edu/vision/SUN/. [Kasutatud 22 04 2018].

[32] „CamVid dataset,“ [Võrgumaterjal]. Available: http://mi.eng.cam.ac.uk/research/projects/VideoRec/. [Kasutatud 22 04 2018].

[33] W. F. Filip Korc, „LabelMeFacade dataset,“ [Võrgumaterjal]. Available: http://www.inf-cv.uni-jena.de/index.php?id=166&site=dbv&lang=en. [Kasutatud 22 04 2018].

[34] „NEXET dataset,“ [Võrgumaterjal]. Available: https://www.getnexar.com/challenge-2/. [Kasutatud 26 4 2018].

[35] „Mapillary dataset,“ [Võrgumaterjal]. Available: https://blog.mapillary.com/product/2017/05/03/mapillary-vistas-dataset.html. [Kasutatud 26 4 2018].

[36] „GitHub: Road image datasets collected by the author,“ [Võrgumaterjal]. Available: https://github.com/kagan94/road_image_datasets. [Kasutatud 25 4 2018].

[37] „GitHub: Animated visualization of convolutional operations,“ [Võrgumaterjal]. Available: https://github.com/vdumoulin/conv_arithmetic. [Kasutatud 27 4 2018].

[38] „A Quick Introduction to Neural Networks,“ [Võrgumaterjal]. Available: https://ujjwalkarn.me/2016/08/09/quick-intro-neural-networks/. [Kasutatud 28 4 2018].

[39] S. H. B. H. H. Noh, „Learning deconvolution network for semantic segmentation,“ *2015 IEEE International Conference,* pp. 1520-1528, 2015.

[40] L. B. Y. B. a. P. H. Y. LeCun, „Gradient-based learning applied to document recognition,“ *Proceedings of the IEEE,* 1998.

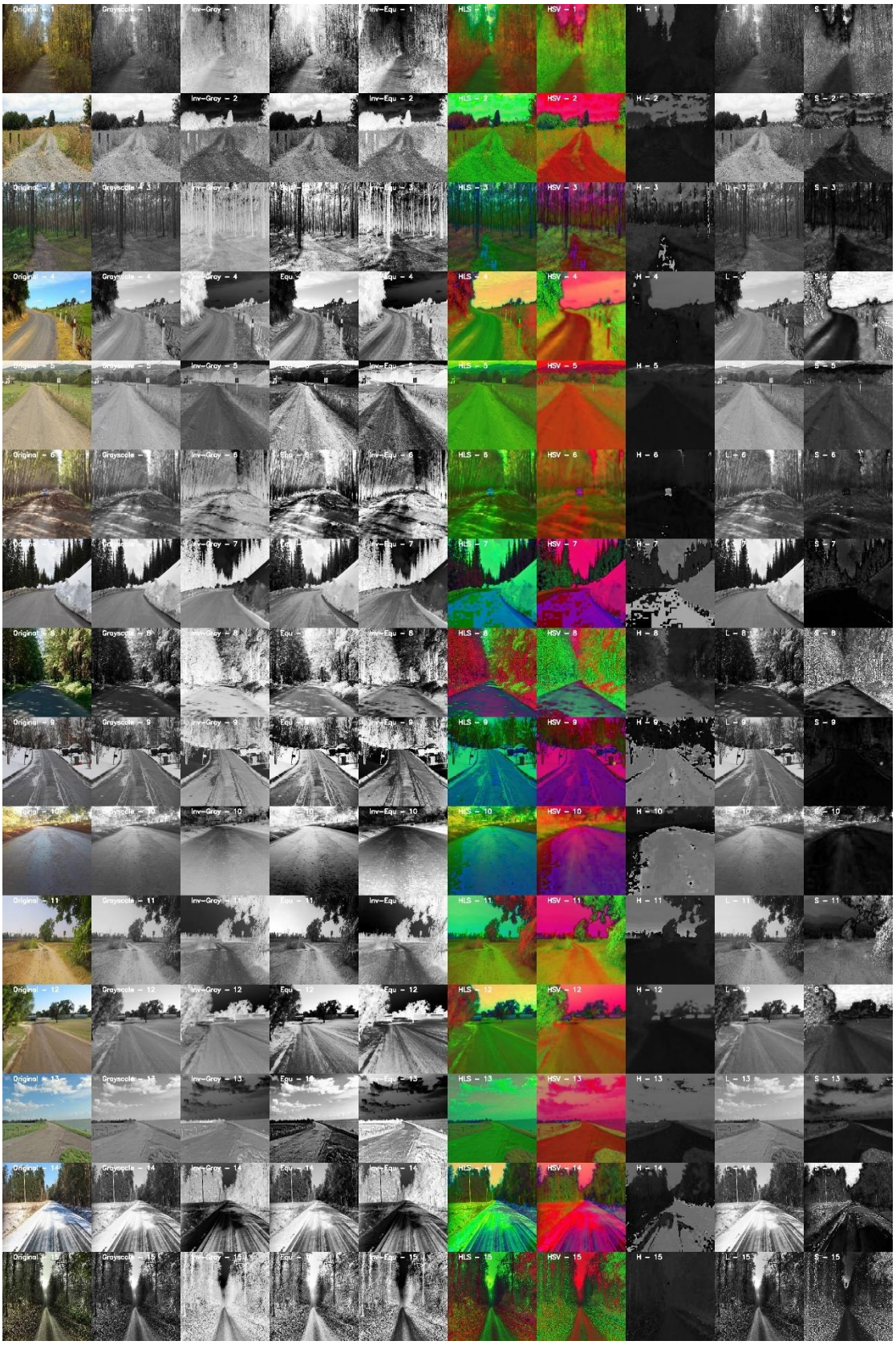
[41] E. S. a. T. D. J. Long, „Fully convolutional networks for semantic segmentation,“ *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 3431-3440, 2015.

[42] A. K. a. R. C. V. Badrinarayanan, „SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation,“ *IEEE Transactions on Pattern Analysis and Machine Intelligence,* kd. 39, nr 12, pp. 2481 - 2495, 2017.

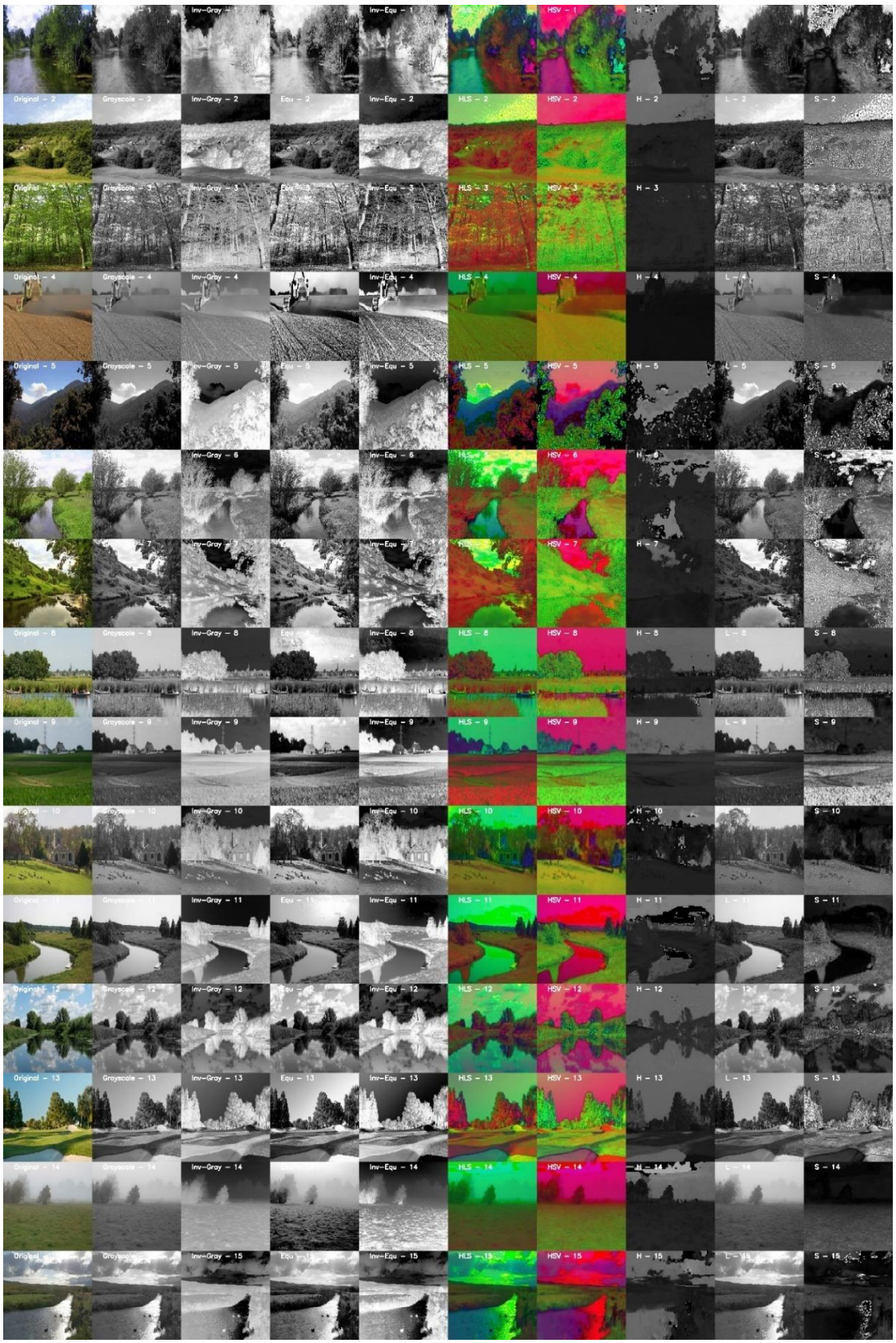
[43] „ImageNet image catalog,“ [Võrgumaterjal]. Available: http://www.image-net.org/synset. [Kasutatud 29 4 2018].

Appendices

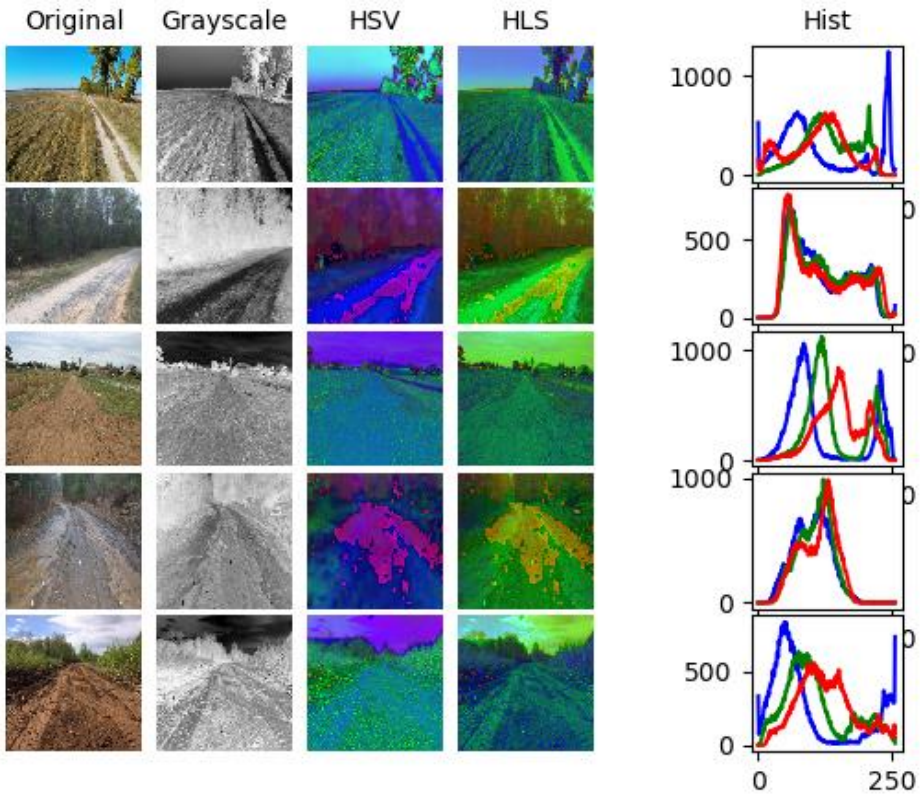
Appendix 1. Color spaces of different road images: original (RGB); grayscale, inverse-grayscale; apply Histogram Equalization (HE); inverse-HE; HLS; HSV; H, L, S channels.



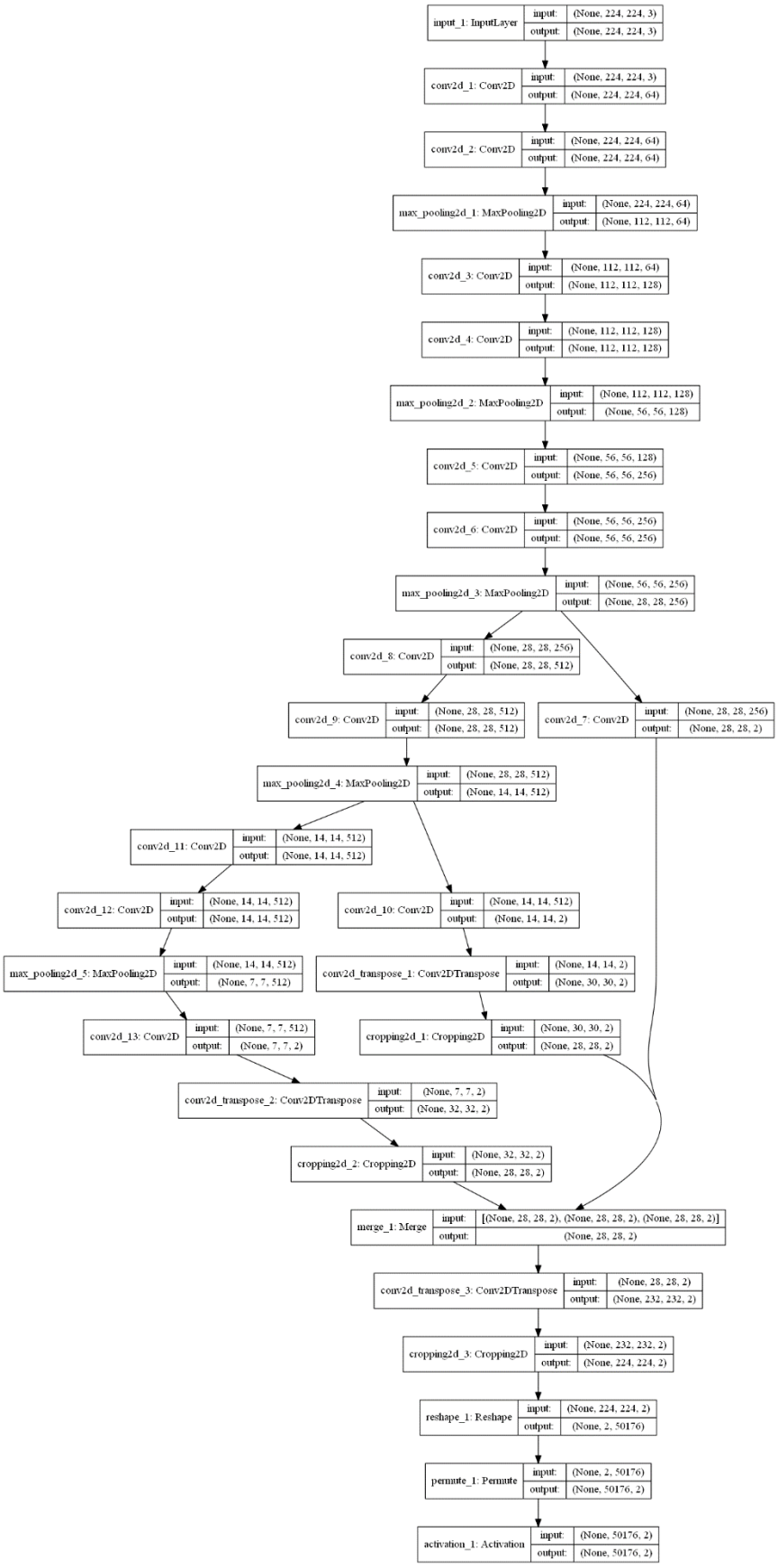
Appendix 2. Color spaces of different non-road images: original (RGB); grayscale, inverse-grayscale; apply Histogram Equalization (HE); inverse-HE; HLS; HSV; H, L, S channels.



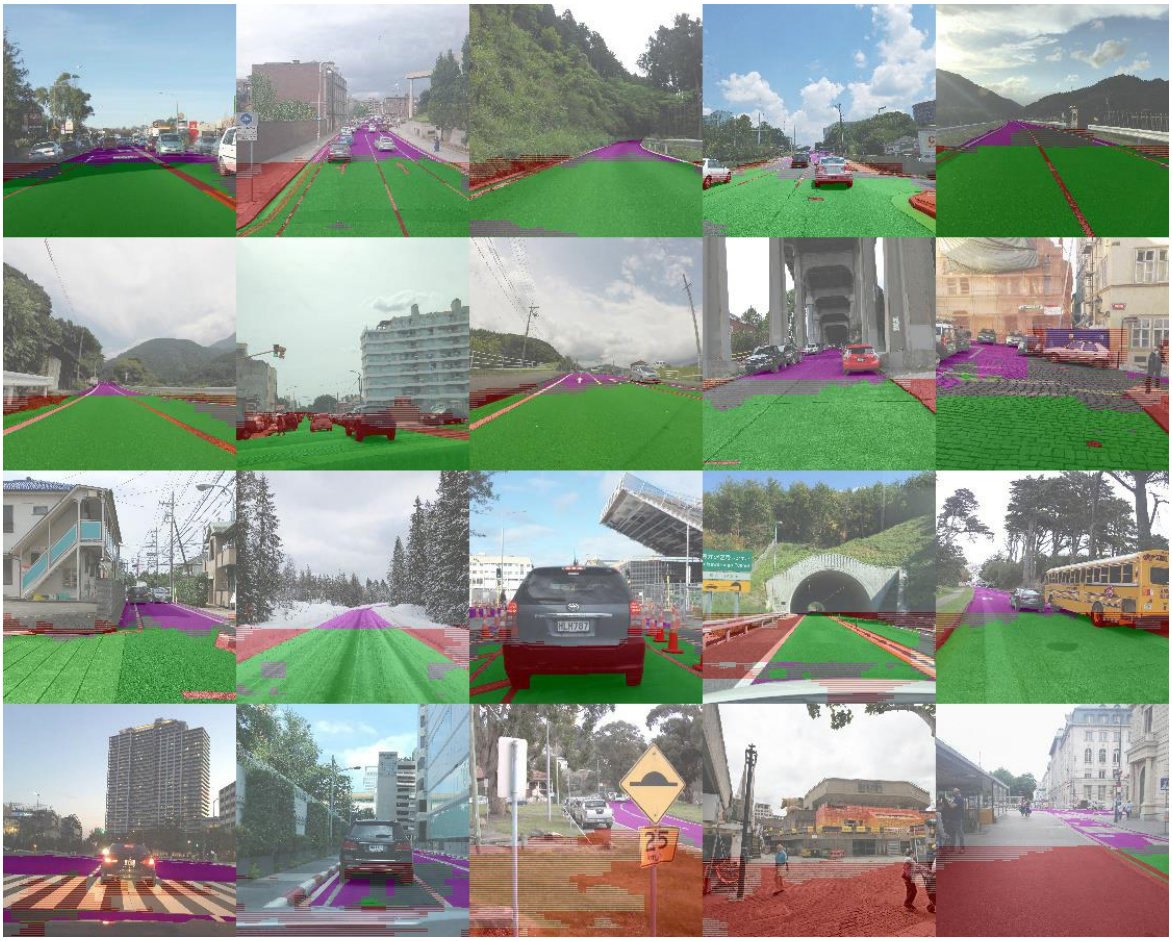
Appendix 3. Original road images(RGB), grayscale, HSV, HLS color spaces accordingly, and histogram of original RGB image.



Appendix 4. Architecture of proposed network for pixel-wise image classification (similar to FCN-8 architecture).



Appendix 5. Results on running our FCN-8-based network on Mapillary dataset [6]. Row 1-3 – satisfactory predictions. Row 4 – failure predictions.



Additional notes:

In addition to the main results, we decided to conduct an additional experiment of our FCN-8-based model on Mapillary dataset of 4,000 images with mainly structured roads. Also, we validated the trained model on 2,000 new images from the same dataset. The minimum, average, and maximum accuracy were 49%, 87%, 99% accordingly. At the same time, the average TP, TN, FP, FN rates were 68%, 90%, 10%, and 31% respectively.

Besides, we noticed that due to avoidance of using pedestrian walk labels in training data, the network still detects it as a road due to color and shape similarity with real roads. Also, we did not include line marking that also affected the predicted images.

[6] G. Neuhold, T. Ollmann, S. R. Bulò and P. Kontschieder, "The Mapillary Vistas Dataset for Semantic Understanding of Street Scenes," IEEE International Conference on Computer Vision, 2017, pp. 5000-5009.

License

Non-exclusive licence to reproduce thesis and make thesis public

I, **Leonid Dashko**,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:

    1.1. reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and

    1.2. make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

of my thesis

Road Detection and Recognition from Monocular Images Using Neural Networks,

supervised by Amnir Hadachi

2. I am aware of the fact that the author retains these rights.

3. I certify that granting the non-exclusive license does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, 21.05.2018