DEGREE PROJECT IN COMPUTER SCIENCE AND ENGINEERING,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2017

Terrain Classification to find Drivable Surfaces using Deep Neural Networks

AGNEEV GUIN

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF COMPUTER SCIENCE AND COMMUNICATION

KTH Computer Science
and Communication

Terrain Classification to find Drivable Surfaces using Deep Neural Networks

Semantic segmentation for unstructured roads combined with the use of Gabor filters to determine drivable regions trained on a small dataset

AGNEEV GUIN

Master's Thesis at Scania CV AB
Supervisor: Christian Smith
Examiner: Patric Jensfelt
Industrial Supervisors: Mikael Salmén, Marco Trincavelli

January 11, 2018

Abstract

Autonomous vehicles face various challenges under difficult terrain conditions such as marginally rural or backcountry roads, due to the lack of lane information, road signs or traffic signals. In this thesis, we investigate a novel approach of using Deep Neural Networks (DNNs) to classify off-road surfaces into the types of terrains with the aim of supporting autonomous navigation in unstructured environments. For example, off-road surfaces can be classified as asphalt, gravel, grass, mud, snow, etc.

Images from the camera mounted on a mining truck were used to perform semantic segmentation and to classify road surface types. Camera images were segmented manually for training into sets of 16 and 9 classes, for all relevant classes and the drivable classes respectively. A small but diverse dataset of 100 images was augmented and compiled along with nearby frames from the video clips to expand this dataset. Neural networks were used to test the performance for the classification under these off-road conditions. Pre-trained AlexNet was compared to the networks without pre-training. Gabor filters, known to distinguish textured surfaces, was further used to improve the results of the neural network.

The experiments show that pre-trained networks perform well with small datasets and many classes. A combination of Gabor filters with pre-trained networks can establish a dependable navigation path under difficult terrain conditions. While the results seem positive for images similar to the training image scenes, the networks fail to perform well in other situations. Though the tests imply that larger datasets are required for dependable results, this is a step closer to making the autonomous vehicles drivable under off-road conditions.

Referat

Terrängklassificering för att hitta körbara ytor med hjälp av Djupa Neurala Nätverk

Autonoma fordon står inför olika utmaningar under svåra terrängförhållanden som landsbygds- eller skogsvägar på grund av bristen av körfältinformation, vägskyltar och trafikljus. I denna avhandling undersöker vi ett nytt tillvägagångssätt att använda Djupa Neurala Nätverk (DNN) för att klassificera terrängytor utifrån deras körbarhet i syfte att stödja autonom navigering i ostrukturerade miljöer. Till exempel kan terrängytor klassificeras som asfalt, grus, gräs, lera, snö etc.

Bilder från kameran monterad på en gruvbil användes för att utföra semantisk segmentering och klassificera vägytor. Bilderna delades manuellt upp i träningsset på 16 samt 9 klasser för alla relevanta klasser respektive körbara klasser. Ett litet men mångsidigt dataset med 100 bilder förstärktes med närliggande bilder från videoklippen för att expandera detta dataset. Neurala nätverk användes för att testa prestandan hos klassificeringen under dessa terrängförhållanden. Det förtränade nätverket AlexNet jämfördes med nätverken utan träning. Gaborfilter, kända för att särskilja texturerade ytor, användes vidare för att förbättra resultaten av det neurala nätverket.

Experimenten visar att förtränade nätverk presterar bra med små dataset och många klasser. En kombination av Gaborfilter med förtränade nätverk kan skapa en pålitlig navigationsväg under svåra terrängförhållanden. Även om resultaten verkar positiva för bilder som liknar träningsbildscenen presterar nätverken inte bra i andra situationer. Även om testen tyder på att stora dataset krävs för tillförlitliga resultat, är detta ett steg närmare att göra de autonoma bilarna körbara i svåra terrängförhållanden.

Acknowledgement

I am grateful to my supervisor, Christian Smith, whose expertise, guidance and support kept me going through this long journey of my thesis.

I am deeply indebted to the entire team of REP at Scania CV AB, especially my industrial supervisors, Mikael Salmén and Marco Trincavelli, my manager, Per Sahlholm, and my colleague, Nazre Batool. All of you provided me the right help whenever it was needed. Your support during all the technical glitches kept me going and supported me with the best facilities among the limited human resources.

I thank my parents for everything they did for me. Thank you for always placing me in front of your own needs, for the sacrifices endured to make my life easier and meaningful. Your love, care, and encouragement have been the reason for all the good qualities that I have today.

A hug and kiss for my girlfriend, Surovi, for understanding me in these last two years of studies. Thanks to you I jumped from a good student to a great student. You did more than just helping me in this jump by carrying me whole in the tough times. Thank you for all the patience that I often required.

A big loving hug to my friend, Aditi. Your friendship is one of the most valuable things I am proud to have. Thank you for all the good moments that we shared, it's still hard for me to believe the luck I had to meet you. In you, I have found an extra motivation to excel in my studies and also the importance of slowing down and enjoying life.

To Pranav and Akshay, a warm hug to you guys. Your young spirit and fresh personalities managed to bring so many laughs to otherwise boring moments. Thanks for intoxicating me with that teenager joy that I was starting to lose.

To every friend in India who helped me in some way or other, your smiles, kindness and friendship deserve much more than these mere words.

Finally to all of the amazing friends I had the chance to meet during my stay in Sweden. A big skål for making this year the best of my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Autonomous navigation in unstructured environments requires a detailed interpretation of the road surface to avoid obstacles and to optimize the driving experience. Understanding the type of surface requires information about the colour, texture, and depth, to be able to categorize it as drivable or non-drivable.

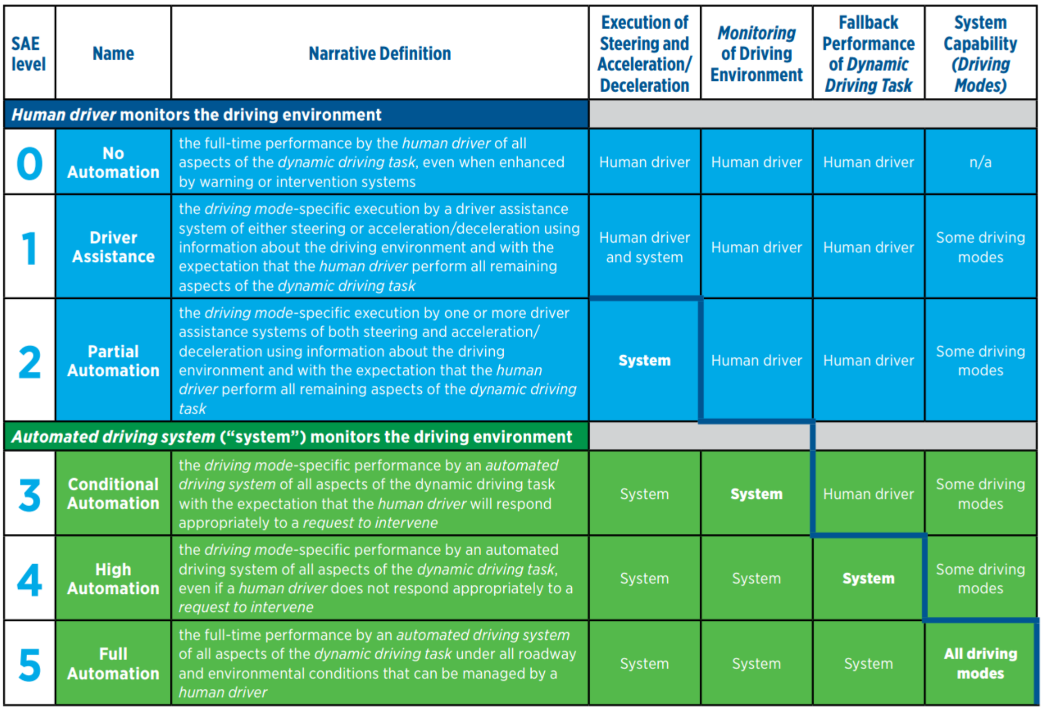
In this thesis, we use the images from the front-facing camera mounted on a mining truck to perform semantic segmentation for classifying the type of road surface. We use a neural network approach to find the optimum network trained only on a small training database of 100 images. We aim to segment the images into different drivable surfaces under off-road conditions, such as asphalt, gravel, grass, mud, etc. Evaluation of these segmented images may be used to prioritize the autonomous vehicles to drive over the smoother and low-risk terrains.

1.1 Background

The Society of Automotive Engineers (SAE) has published the Surface Vehicle Information Report [1], which states the 6 levels of automation for on-road motor vehicle automated driving systems from no automation to full automation (J3016). The levels are defined based on the amount of interference with the human to share the information with the vehicle controller.

Unmanned vehicles are designed to perform their assignments without the need for human intervention. These vary from case to case such as transportation, exploration, mining, rescue operations and military applications [2]. Humans perceive the environment by merging the environmental features with the contextual information. In the autonomous field, imitating the same requires various sensors to be placed in and around the vehicle. The data is then processed and high-level decisions are made to replicate human behavior.

iQMatic, a collaborative project between KTH University, Linköping University, Scania CV AB, SAAB, Autoliv and Combitech, aims to develop autonomous trucks to accomplish tasks like debris collection and transportation at mining areas. The environmental regulations and the dangerous work conditions can be resolved by



SAE level	Name	Narrative Definition	Execution of Steering and Acceleration/ Deceleration	*Monitoring* of Driving Environment	Fallback Performance of *Dynamic Driving Task*	System Capability *(Driving Modes)*
***Human driver* monitors the driving environment**						
0	No Automation	the full-time performance by the *human driver* of all aspects of the *dynamic driving task*, even when enhanced by warning or intervention systems	Human driver	Human driver	Human driver	n/a
1	Driver Assistance	the *driving mode*-specific execution by a driver assistance system of either steering or acceleration/deceleration using information about the driving environment and with the expectation that the *human driver* perform all remaining aspects of the *dynamic driving task*	Human driver and system	Human driver	Human driver	Some driving modes
2	Partial Automation	the *driving mode*-specific execution by one or more driver assistance systems of both steering and acceleration/ deceleration using information about the driving environment and with the expectation that the *human driver* perform all remaining aspects of the *dynamic driving task*	**System**	Human driver	Human driver	Some driving modes
***Automated driving system* ("system") monitors the driving environment**						
3	Conditional Automation	the *driving mode*-specific performance by an *automated driving system* of all aspects of the dynamic driving task with the expectation that the *human driver* will respond appropriately to a *request to intervene*	System	**System**	Human driver	Some driving modes
4	High Automation	the *driving mode*-specific performance by an automated driving system of all aspects of the *dynamic driving task*, even if a *human driver* does not respond appropriately to a *request to intervene*	System	System	**System**	Some driving modes
5	Full Automation	the full-time performance by an *automated driving system* of all aspects of the *dynamic driving task* under all roadway and environmental conditions that can be managed by a *human driver*	System	System	System	**All driving modes**

Figure 1.1: SAE's Levels of Automation. (Copyright © 2014 SAE International. Summary table distributed from SAE International and J3016. [1])

turning to autonomous technologies with the focus on safety as well as fuel efficiency. The project aims to develop fully autonomous vehicles by limiting human interventions to only supervision and troubleshooting, thereby reducing human accidents at risky work environments [3, 4].

"Knowledge of terrain properties could allow a system to adapt its control and plan strategies to enhance its performance by either maximizing wheel traction or minimizing power consumption" [5]. Classification of terrains into traversable or non-traversable can be determined by evaluating two conditions, namely, terrain classification and terrain characterization. Terrain classification categorizes the type of terrains like gravel, sand, dirt or mud, whereas terrain characterization describes the effect on the driving experience. For example, driving on muddy roads after rain is difficult compared to the same conditions on an asphalt road. Thus, for a vehicle to work autonomously, it needs to map itself in the environment, determine the drivable surfaces, avoid obstacles and plan a path along the most convenient route. Thus, the terrain roughness, slope, discontinuity, and hardness need to be considered to assess the quality of drive [6, 7]. The terrain identication can be done by either retrospective technique or prospective technique. Retrospective technique, also known as the contact based way, uses the past data from the traversed terrain to identify the terrain whereas prospective technique, also known as a range-based way, estimates the terrain from the near future. [8, 9].

There has been an extensive research for paved road detection [10, 11, 12]. However, under off-road conditions like marginally rural or back-country roads, the same algorithms cannot be applied due to lack of lane information, road signs or signals. The autonomous vehicles can comprehend the road ahead based on the distinguishing factors of the visible drivable path from its immediate surroundings. Foedisch and Rasmussen *et al.* [13, 14] provides grounds that there has not been a reasonable development with conditions like fuzzy road borders, low-intensity contrast, non-planar surfaces and different road building materials like mud, clay, sand, gravel, and asphalt along the same road.

1.2 Motivation

The surface type classification for unstructured environments plays a key role in maneuvering autonomous vehicles safely and efficiently over the drivable regions. The current sensor technologies are able to determine the drivable regions in a structured environment like highway or city road, but it is still a challenge to classify off-road terrain surfaces.

Today, lidar data is used to identify free space while camera data is used to determine road details like lane markings and road signs. Both types of data are combined with other sensors to comprehend the possible obstacles or other vehicles. However, it is still a challenge to drive vehicles at unstructured environments like mining areas as it requires knowledge of the drivable region. Perception sensors are not developed enough to evaluate the colour and texture information to categorize the drivable road surfaces. Thus, a breakthrough technology is required for the evaluation of the road surface type.

1.3 Aim and Scope of the Project

In this thesis, we investigate a new approach to classify surfaces based on the terrain type. We train a Deep Neural Network (DNN) based on human understanding of drivable road conditions.

We evaluate if a small dataset of 100 images from a training track is enough to train the system to determine and classify the visual data into various surface type classes. Considering the advancements in the field of texture classification and neural networks, we also evaluate how texture information and pre-trained networks can improve the performance of DNNs.

The drivable regions at mining areas are mostly fictive tracks that appear as a result of other vehicles traversing over them. These tracks are mostly semantic in nature and the human mind tends to conceptualize a pathway based on these assumptions. However, a number of factors like rain or storm can affect these tracks. Thus, a certain scope is defined for the thesis work by ignoring these factors and favoring the limitation of time and budget.

Factors included within the scope of the experiment:

- Open sky conditions with day-time lighting conditions are considered.
- The classification variables primarily focus on the texture, colour and different off-road terrains.
- A certain camera type and viewing position will be used.
- Near-flat road terrains will be preferred with either straight or curved roads.
- Visual data without the depth information will be considered. Thus, small bumps and holes on the road would be considered as drivable.
- The area of interest is limited to a range of up to 50 meters from the vehicle on a flat terrain.

Factors out of the scope of the experiment:

- Cloudy and night time driving is avoided due to the mixture of colour and illumination information.
- Motion blurred images would be eliminated from the training data as it might lead to false positives during the real-time.
- Cross-roads and road diversions will be avoided.
- Compressible terrain like tall grasses or low bushes would be considered as non-traversable.
- The classification information only will not be enough to drive the vehicle autonomously.

Chapter 2

Literature Review

In this Section, we present the current technologies which are used to classify the road surface types. We mention about the theory required to perform the experiments and also the basics of neural networks with the methods to analyze their efficiency.

2.1 State of the Art Technology

Visual terrain classification is performed based on the features attained from pixels from a certain region of the image, like colour, position, luminance, and texture. The environment's geometry provides information for the prospective terrain which is mostly attained from sensors like Mono/Stereo cameras, Laser scanners, Radars or Ultrasound sensors.

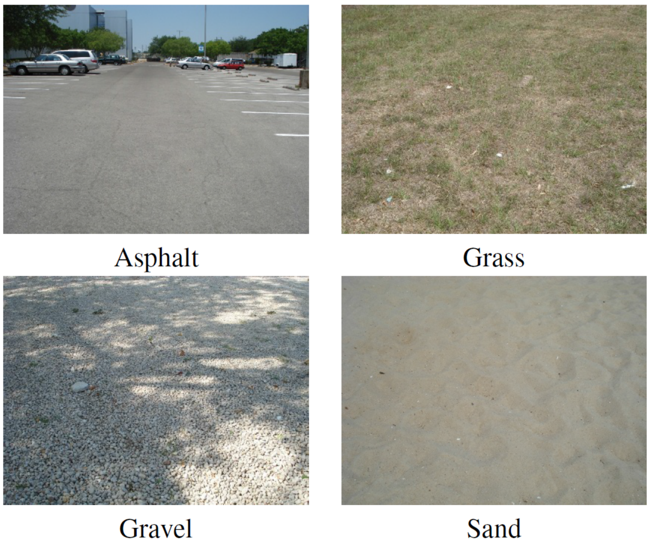


Figure 2.1: Different outdoor terrain environments. (Copyright © 2009 IEEE. [15])

Different vision algorithms have been used to determine the distinguishable features, such as elevation maps (height variation in a cell) [16, 17] and point-wise classifications (based on local point cloud geometry) [18]. Image classifiers of varying complexity are then applied on these features to obtain different levels of successes. The commonly used classification methods are the Nearest Neighbour (NN), Artificial Neural Network (ANN), Support Vector Machine (SVM), Random forests, Multilayer Perceptron (MLP), Extreme Learning Machine (ELM), J48 Decision Tree, Naive Bayes, k-Nearest Neighbor and Fuzzy rule-based system [5, 8, 9].

Recently, Deep Neural Networks (DNNs) have taken over the image processing challenges with the introduction of AlexNet in 2012 at the ImageNet ILSVRC Competition [19]. Bittel *et al.* [20] incorporated the use of neural networks for the road classification task. A framework called Street Segmentation Toolkit (SST) was developed as a python package to classify the roads from its surroundings and perform regression task to obtain the best network scores. However, the application of neural networks has only been to classify the road from its surroundings. Their use in road type classification is still in its early phases. This has been further emphasized in the Section 2.2.5. In this thesis, we focus on the use of the neural networks to determine the type of road surface.

2.2 Related Works in Visual Road Classification

Various classification techniques have been used in the past depending on the type of camera and the segmentation techniques. In this section, we review the works that have been used in the past for the vision based road classification techniques.

The driver's perspective view usually covers a trapezoidal area where the road is centered in the lower part of the image with the top edge defining the road boundary and the horizon. This simple concept was demonstrated by Foedisch *et al.* in 2004 [13]. This is depicted in the Figure 2.2. To distinguish road patches from the off-road zones, Rasmussen [14] used height, smoothness, colour and texture in 2002. Information about height and smoothness is provided by the laser data whereas colour and texture are observed from the image results based on patches of the road.

2.2.1 Stereo Vision

The majority of the work addressing the perception problem for road classification utilizes stereo vision. It uses a minimum of two cameras displaced horizontally, similar to human binocular vision. It is used to obtain the depth information, calculated by the inverse distance from the disparity map. This is further merged along with the data from the laser sensors [21, 22, 23, 24]. Soquet *et al.* [25] performed road extraction by colour segmentation for intelligent vehicles by the use of stereo vision. Algorithms for localization and mapping of vehicles using stereo vision have been implemented on a forest terrain by Agrawal *et al.* [26]. Additional

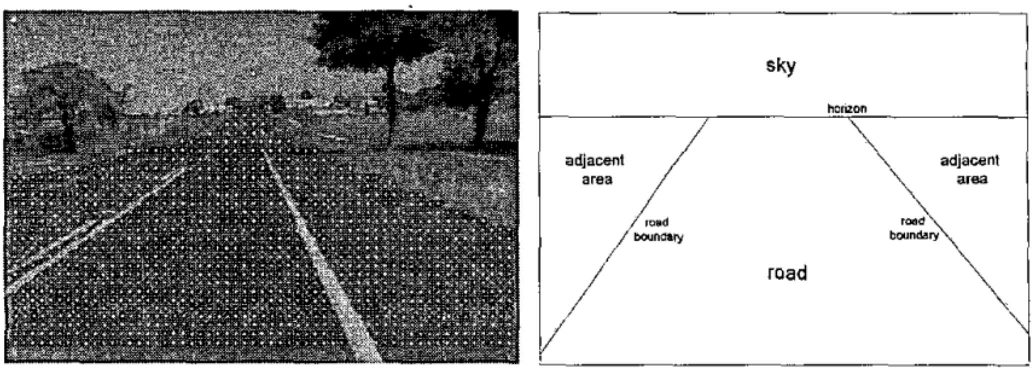


Figure 2.2: (Left) Classification result where road is shown with white squares and non-road sections are shown with black dots. (Right) Typical road structure perspective. (Image from [13])

possibilities like volumetric density mapping, geometry based terrain classification and multi-spectral terrain classification are explored by Kelly *et al.* [27].

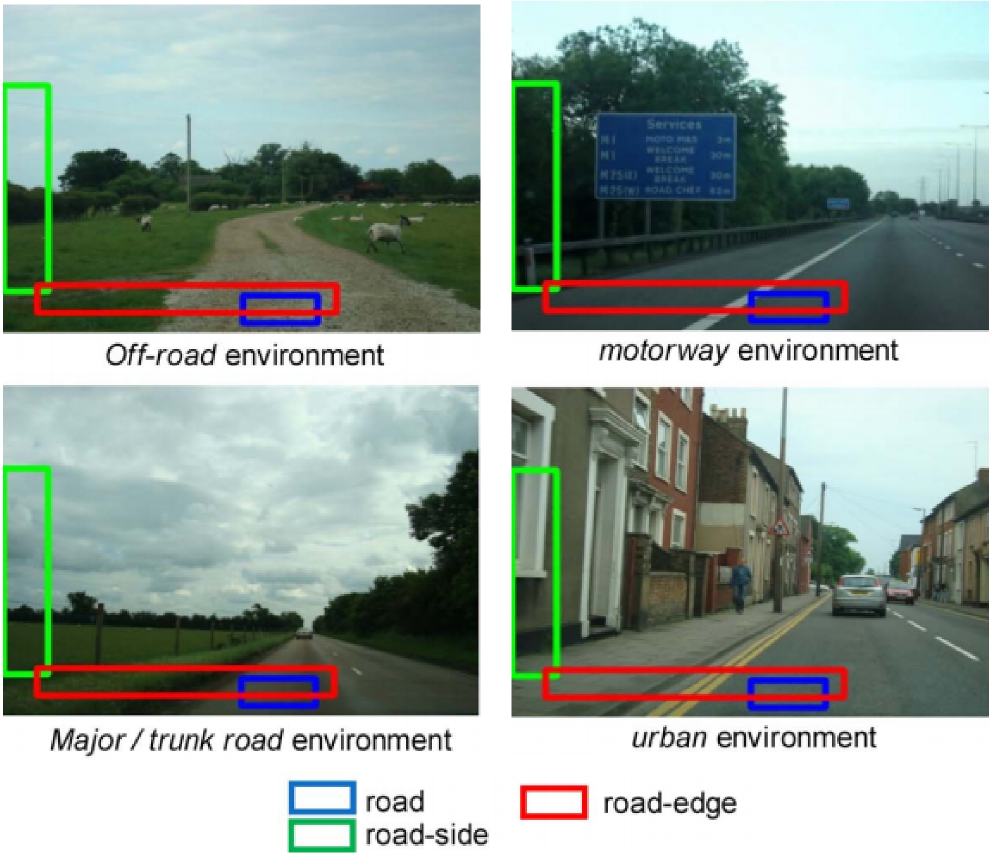
2.2.2 Color Segmentation

A significant development has been observed in the field of colour segmentation for road detection [28, 29, 30, 31, 32]. As per Tang *et al.* [28], the forward facing camera detecting the road edge should separate the road from the road sides. This is depicted in the Figure 2.3. A choice of colour space is used to represent with a combination of RGB, HSV and YCrCb colour variations. The resultant of the colour feature over all colour channels combine to form a 91-D colour descriptor for each image frame. Rasmussen [14] generates colour histograms with the probable colours of brown and gray representing the road, while colours like green and blue depicting the background, and thus characterizing the off-road features. The study by Angelova *et al.* [33] provides a comparison of the colour histogram with an average colour determined by normalizing R and G values. Though the colour histogram provides better and faster representation than the average colour for terrain classification, it is recommended and further acknowledged to fuse it with other texture classifiers [34, 35].

2.2.3 Texture Analysis Methods

Gabor Filters

Gabor filter is one of the well-known texture feature extractors which is represented as a 2-D convolution kernel with a specified spatial frequency and orientation [36, 37]. A local Fourier transform, composed of a sinusoidal harmonic oscillator modulated by a Gaussian window, is convoluted to obtain the spatial information about its neighboring pixels. The filter is further explained in Section 3.3.9.



Off-road environment

motorway environment

Major / trunk road environment

urban environment

Figure 2.3: Road environment observed by driver's perspective camera view. (Image from [28])

GLCM Matrices

The Gray level co-occurrence matrix (GLCM) picks up the statistical parameters about the spatial relationship between pixels and their light intensities [38]. It is used to obtain the local correlations between the image pixels on a predetermined scale. Texture descriptors like entropy, energy, contrast, correlation and local homogeneity are compared for the different terrain classes [15, 28]. An improvement in this domain is observed by the use of Gray level Aura matrix (GLAM) by Haliche *et al.* [39]. It relates the global interaction of the pixels by relating the covariance matrices and Markov random fields and computing with the symmetric and asymmetric neighborhood system.

Wavelet Transform

Wavelets are mathematical functions that split the data into different frequency components as a series of images with different scales. The transform decomposes the data into the elemental functions of wavelets and scaling. Wavelets behave as high-pass filters whereas the scalings are similar to low-pass filters. An inverse transform is performed to obtain the horizontal, vertical and diagonal details. Different factors like the type of features, window size, and type of wavelets affect the decomposition by wavelet transform process. Sung *et al.* [34] uses a discrete wavelet transform to extract information about colour and texture from the images. Additionally, spatial coordinate information was added to improve the terrain classification.

2.2.4 Classification Methods

Different classification methods can be used to evaluate the raw feature set and the selected feature set. A comparative study of the classifiers by Yun *et al.* [40] gives a relation between the random forest classifiers, support vector machine and a feature selection algorithm based on boosting. The boosting classifier resolves a feature selection process by boosting single features and attempting to improve the accuracy of a weak learning algorithm. The work uses three-fourth of the images for training and the remaining for evaluation.

A comparative study by Khan *et al.* [41] for visual terrain classification has introduced five further texture classifiers, namely, the Local Ternary Patterns descriptor (LTP), the Local Adaptive Ternary Patterns descriptor (LATP), the Speeded Up Robust Features descriptor (SURF), the Daisy descriptor and the Contrast Context Histograms descriptor (CCH). The first two descriptors involve a threshold parameter for a 3x3 pixel pattern on the image. The adaptive nature makes it less sensitive to noise and illumination changes. The SURF descriptor, however, expects a key interest point location and is usually concentrated around sharp gradients which are mostly absent in homogeneous terrain patches.

A neural network based classifier for terrain classification has been used by Ojeda *et al.* [6], which uses a feed-forward network for a five output result. This approach requires a robust training set based on the extracted set of feature vectors and is applied over a test set of the real-time data.

According to Halatci *et al.* [5], high-level classifiers may be designed by fusing the existing results to ensure better accuracy. Bayesian classifier fusion picks up the classifier, outputs in the same class space for all the sensing modes and computes the distributions based on the Bayes' Rule. A meta classifier fusion utilizes data association based on a patch obtained by the pixel value averaging. This high-level classifier is expected to use a different training set than the ones used for training low-level classifiers.

Recently, a novel learning algorithm for a single hidden layer feed-forward network (SLFN), namely, extreme learning machine (ELM), has been proposed by

Huang *et al.* [42]. Huang *et al.* [43] also proposed that ELM can be applied for regression and classification problems. ELM is found to be simple to tune network parameters and fast to learn training samples. This is because, instead of tuning, it randomly chooses the input weights and the hidden layer neurons. Junior *et al.* [44] have successfully applied ELM in the texture classification and the results present higher success rates as compared to all different texture classifiers. Although it is important to note that the method uses more descriptors than the other methods, it reflects similar result for the individual feature vectors.

2.2.5 Feature based Neural Networks

Texture based neural networks were first used by Tivive *et al.* in 2006 [45]. The paper tested the architecture with images from the Brodatz texture database. According to the paper, the proposed network achieved similar or better classification performance compared to some of the most popular texture classifications like Gabor filters, wavelets, and co-occurrence matrix methods. A comparison of the different supervised learning algorithms by Caruna [46] showed that the feed-forward neural nets perform better than the standard learning methods like SVMs, boosting, random forests. Thus, the neural networks have been widely used over the last decade. However, their use in road detection to differentiate the road from non-road has been performed by several works [20, 47, 48]. It is only in the year 2017, that Bystrov *et al.* [49] used neural networks to classify the road surface type. They used the features from radar and sonar back-scattered signals to use it for the supervised classification using neural networks. However, no related work for image-based neural networks was found to classify the wide range of surfaces for off-road driving during our study.

2.2.6 Semantic Segmentation

Semantic segmentation is a sub section of a classification task where the algorithm is designed to cluster parts of the image belonging to the same object class by training on a central decision [50]. The classification is done at the pixel level to categorize the subjects to a set of classes. It answers two questions, namely *what* and *where* [51]. The global information gives what the context indicates and the local information gives where it is seen in relation to its surroundings.

A supervised segmentation algorithm uses a fixed set of classes, which may be binary classes like foreground and background or may possess more classes depending on the training and need. While the patches, boundary segments, vertices, holes and line segments act as the descriptive elements; parameters and pointers in each element give the topological relations between each of them [52].

Chapter 3

Theory

The extraction of sensible data from the environment to attain certain levels of perception is dependent on the capacities of the sensors used. Human brain understands its surroundings by its capability to see, feel and intuit based on its past experiences, however, a machine requires apt sensors and estimation techniques to perform the same. An image sensor conveys the visual information by converting the variable attenuation of light waves into current signals. Machine vision is used to evaluate the visuals and recognize the content in the image for a fruitful comprehension. Machine vision involves two components, namely obtaining features, and pattern classification (based on the features) [53].

For a machine to understand the image, it needs additional levels of processing to be able to figure the useful section of the image. This involves the following steps:

- **De-noising**: Blurry, distorted and noisy images need to be removed or reduced to carry any further processes.

- **Segmentation**: Finding out the appropriate regions and differentiating them from their background or partitioning them into connected regions.

- **Feature extraction**: Processing the image by performing transformations, allowing the determination of similarities and distinguishable representations.

- **Consistency**: Labeling of a single object, a pair of objects or an entire scene to provide detailed information about the 3D environment from the 2D data.

- **Classification and matching**: Recognizing objects from the image.

To perform the different image processing algorithms, it is important to provide the real world geometry. Camera calibration plays an important role in this. Similarly, to be able to evaluate the results, certain evaluation metrics are necessary. These are explained in Sections 3.1 and 3.2. We will further learn about the neural networks in Section 3.3.

3.1 Camera Calibration

Intrinsic camera calibration provides the relation between the image pixels and real world dimensions. These are estimated by the following:

1. Focal length of lens along X axis
2. Focal length of lens along Y axis
3. Lens displacement along X axis
4. Lens displacement along Y axis

Extrinsic calibration aligns the camera with the real world coordinates. The following parameters give the extrinsic calibration.

1. The position of the camera in the real world (x, y, and z)
2. The orientation of the camera in the real world (around x, y and z axis)

Distortion parameters give the estimate of the deformation from the real world. The distortion parameters are determined by the following:

1. Radial distortion: Represented by 3 values
2. Tangential distortion: Represented by 2 values

OpenCV provides an algorithm to obtain the checkerboard corners from a checkerboard pattern in an image [54, 55]. The out of the box functionality picks the feature points to generate the camera matrix and the distortion matrix. The output 3x3 matrix is in the form:

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

where, the coefficients f_x and f_y represent the focal lengths, indicating the flaws and errors due to the digital camera sensor or the camera lens. The coefficients c_x and c_y give the principal point offset representing the offset from the image center. The calibrate camera function of the OpenCV generates an output vector of distortion coefficients $(k_1, k_2, p_1, p_2[, k_3[, k_4, k_5, k_6]])$ of 4, 5, or 8 elements.

3.2 Performance Metrics

Training a supervised model requires a set of labeled data. Some of this data should relate to the data which needs to be predicted. A subset of this data is left aside to evaluate the performance of the model. Evaluating the response of the machine learning algorithm requires knowledge about certain terminologies. Further on, the binary metric will be extended to multiclass or multilabel problems by treating the data as a collection of binary problems, one for each class. It is important that none of the metrics are considered in an isolated way as there is not the best way to evaluate any system, but rather, different metrics provide different insights into how a classification model performs.

Confusion Matrix

A comparison chart generated by the relation between the predicted and the actual results gives a confusion matrix, as given in the Table 3.1. A binary comparison is provided between the predicted and the actual values here.

Table 3.1: Confusion Matrix

	Actual = positive	Actual = negative
Prediction = positive	True Positive	False Positive
Prediction = negative	False Negative	True Negative

True positives (TP) and true negatives (TN) are the states which were predicted correctly for the corresponding classes and their negatives respectively. A false positive (FP) is a state of false alarm, where the result of the test is positive even when it should have received a negative result. FP is a condition of type I error when the null hypothesis is incorrectly rejected. On the other hand, a false negative (FN) is when a negative test result is incorrectly predicted, i.e. it should have given a positive result. FN is a condition of type II error when the null hypothesis is not rejected when it should have been.

Accuracy

Accuracy is the measure of correctly predicted observations. It is given by the ratio of the correct predictions to the total number of predictions. However, it may be noted that the accuracy itself is not the best evaluation tool due to a state called accuracy paradox. This is when TP < FP and the event conditions are negated. In this case, accuracy always increases.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{3.1}$$

Precision

Precision gives the ratio of correct positive observations. Thus, it indicates the fraction of correct results for all results that were predicted positive.

$$Precision = \frac{TP}{TP + FP} \tag{3.2}$$

Recall

Recall is the ratio of correctly predicted positive events. It is also known as sensitivity or true positive rate. Recall indicates the amount of results picked out of all the actual positive samples.

$$Recall = \frac{TP}{TP + FN} \tag{3.3}$$

F-Score

The F-Score, or the F1-Score, is the weighted average of Precision and Recall. It is given by the harmonic mean of both the values and scaling it to 1. Therefore, this score takes both false positives and false negatives into account and gives a better understanding of accuracy, precision or recall separately.

$$F - Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{3.4}$$

Intersection over Union

The Intersection over Union (IoU) is an evaluation metric to measure the accuracy of prediction. This is calculated by finding an overlap between the ground truth data and the predicted data and is given by the union of the two regions. The contours enclosing the actual object and the likelihood of the object corresponding to the same contour indicates the outcome of the prediction [56]. The ratio of the area of overlap and the area of union gives the IoU. An illustration of the IoU can be seen from the Figure 3.1. Averaging the same over all the classes gives the mean Intersection over Union (mean IoU). Observing the relevance, the mean IoU Score is calculated by averaging the ratio of true positives and the sum of true positives, false positives and false negatives. Thus,

$$Mean\ IoU = \frac{TP}{TP + FP + FN} \tag{3.5}$$

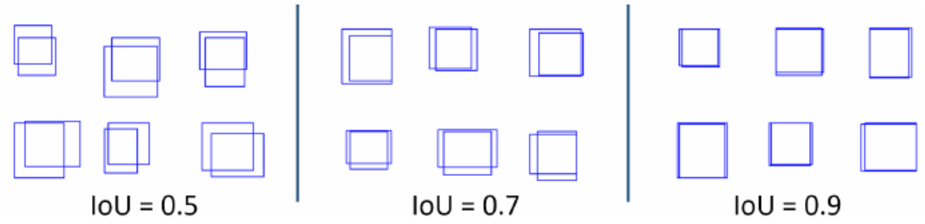


Figure 3.1: Illustration of IoU for IoU = 0.5 (poor), 0.7 (good) and 0.9 (excellent). (Image from [56])

3.3 Neural Networks

Neural networks in machine learning are popularly known as Artificial Neural Networks (ANNs). These are said to be “a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs.” [57]. It is a programming paradigm based on the interconnection of neurons in the human brain which enables a computer to learn from observational data. In the ANNs, multiple layers comprising of interconnected nodes formulate a learning rule based on a set of example data. The nodes represent weights which are updated over multiple cycles during the supervised training process.

A typical neural network can contain millions of artificial neurons in the form of input, output, and hidden layers. They are designed to receive various forms of information, learn from them, recognize the significant features and activate the neurons when the corresponding features are determined. Most neural networks are fully connected, meaning that all the units in each layer between the input and output layers are connected to each other to the other layers. Thus, when an input data is fed, the patterns of information trigger the layers of the network and updates the weights, which are either positive (when one unit excites another) or negative (when one unit suppresses another). Higher weights define more significance for the particular feature [58].

3.3.1 Labeled Datasets

As mentioned earlier, the neural network requires a set of labeled data for activation. The performance of neural networks depends on the amount of training data. It is generally recommended that a network should be trained on millions of non-identical images to be able to perform well for an unknown image [59].

The labeled datasets are distributed in two or three categories for training and evaluation purposes, namely, training, validation and test datasets. The distributions are described as below. The datasets are divided in different ratios like 50:25:25, 60:20:20 or 80:10:10 based on the distribution of the labels. The validation set is sometimes ignored for a 50:50, 60:40 or 80:20 ratios to evaluate the test data directly.

Training Set

Mostly taken about 60% of the original dataset, the training set is used to generate the prediction algorithm. The algorithm tunes and tweaks the weights according to this data. The generated model is tested with different algorithms to compare the performances during the cross-validation phase.

Validation Set

The validation set, sometimes known as the cross-validation set, is about 20% of the original data set. The algorithms are tested with each other to be able to select the best performing algorithm. A simultaneous training and validation can help in improving the results with every training loop.

Test Set

The test data should be taken as high as possible to comprehend the performance of the prediction algorithm on an unseen data on which it has not been trained on. Better performance on the untrained data will reflect whether or not the network will perform well at real-world unknown scenarios.

3.3.2 Available Datasets

A number of open source services have emerged providing labeled data in the form of images and videos. CamVid (Cambridge-driving Labeled Video Database) [60] is a collection of hand annotated images for a 10-minute video sequence with frames captured at 30 Hz. There is a total of 701 semantically annotated frames available at 1 Hz. The paper also gives a comparison of all the previous datasets which gives pixel-wise masks like the VOC 2007 [61].

A recent paper by Zhou *et al.* [62] gives the comparison of the existing datasets with semantic segmentation. It also enlists a number of datasets like COCO, PASCAL, Cityscape, and SUN, which classify the objects of interest. While COCO focuses on the scene annotation, the others focus on object categories. Among these datasets, SUN enlists a few of the surface type classes like mud, sand, gravel, and grass. However, there are only a handful of these images and the classes contribute to only a small part of the image. Thus, none of the available datasets are apt sources for categorizing the road surface type classes.

3.3.3 Vanishing Gradient Problem

To optimize a machine learning problem, the weights need to updated proportional to the gradient of the error functions with respect to the current weight. The optimization process thus involves performing multiple iterations to determine the maxima or the minima.

The feed-forward networks observe an enduring problem as the training is dependent on the initial stages with randomly set activations. At early stages, the random assignment of the weights causes high errors and results in an exponential decrease in the gradients causing a very slow learning outcome. This challenge is popularly known as the Vanishing Gradient Problem.

The optimization function, given by the sum of differential functions, is called as the Stochastic Gradient Descent. This is further explained in Section 3.3.5. However, a substantial development to reduce the Vanishing Gradient Problem is observed with the development of the Recurrent Neural Networks which is explained in the Section 3.3.10.

3.3.4 Layers

As mentioned earlier, neural networks are composed of different layers performing different functions. We discuss the functionalities of some of the most widely used layers below.

Convolutional Layer

The convolutional layer is generally the first layer to a neural network where the image passes through a filter of weights or parameters. The filter is also known as a neuron or a kernel and is of the same depth as the image. This ensures that there is no dimensional hindrance. The part of the image where the filter acts is called its receptive field. The result of the convolution for a fixed region of the image gives the multiplicative of the height, width, and depth. The output of the convolution gives an activation map or a feature map. The filter strides over the image by shifting at all possible placements of the filter and thus a 5x5 filter reduces the dimension of the activation layer from the original image size by 4 units for a stride of 1 pixel. To maintain the dimensions of the output image, a zero padding of size 2 may be applied to the input image to raise the height and width dimensions by 4 pixels. Thus, if the same filter is passed over, we get the output as the same dimension as the input.

Filters with different element values are used to determine the features and are called as feature identifiers. Thus, features like edges, curves, corners are obtained in the early stages of the network and more complex features are obtained as we go deeper in the network.

ReLU (Rectified Linear Units) Layer

The ReLU layer is a type of a neuron layer which obtains element-wise operations to introduce non-linearity to the system and to generate output blobs of the same size. This is generally added after the convolutional layers and introduces non-linearity by nonlinear functions. Recent ReLU layer outputs x if $x > 0$ and $negative\_slope * x$ if $x <= 0$. If the negative slope is not defined, it uses the function $f(x) = max(0, x)$ which eliminates the negative activations to 0.

Pooling Layer

The pooling layers are mostly positioned after the ReLU layers to downsample the result. They are sometimes known as a downsampling layers. The different types of pooling include max-pooling, average pooling and L^2-Norm pooling. This layer not only reduces the amount of parameter weights by 75% resulting in a drastic improvement in computational cost but also eliminates the possibility of categorizing the exact position of the feature in the image. This is done by providing a relative location with respect to other features.

Dropout Layer

To ensure that overfitting does not occur, a random subset of the activations is set to zero in the forward pass. Thus, the network trains itself to give the correct estimate with certain activations dropped out.

Local Response Normalization (LRN) Layer

This type of layer performs normalization over the local input regions. The values are defined by the input values being divided by a factor of $(1+(\alpha/n)\sum_i x_i^2)^\beta$, where, n is the size of the local region or the side length of the square region to sum over, α is the scaling parameter, and β is the exponent used to place the sum in the center of the region. Thus, it performs the computation to exhibit 'Lateral Inhibition', a term used in neurobiology to refer to the potential of an excited neuron to reduce the activity of its neighboring neurons. The normalization allows the spreading of the action potentials to neighboring neurons.

Softmax with Loss Layer

The softmax with loss layer is a combination of a softmax layer followed by a multinomial logistic loss layer. Regression analysis is generally observed to obtain the relation of the independent variables with the dependent variables. The multinomial loss function for a multi-class scenario is given by

$$J(\theta) = -[\sum_{i=1}^{m}\sum_{k=1}^{K} 1\{y^{(i)} = k\} log\ P(y^{(i)}) = k|x^{(i)};\theta)]$$

where K is the number of the classes and $1\{\cdot\}$ is an indicator function, so that 1{a true statement} = 1, and 1{a false statement} = 0.

The softmax regression classifier is suited best when the classes are mutually exclusive. Also, the softmax regression's parameters are overparameterized, i.e., all the data needs to be fitted into either of the classes. Thus, it is always recommended to have a null class or a class contributing to none of the other classes.

Inner Product Layer

The inner product layer is mostly called the fully connected layer. It takes the input image and generates an output in the form of a single vector with height and width set to 1 unit. In case the number of filters is not provided, it observes as many numbers of classes as the input. Thus, this layer is generally at the end of the network and outputs an N-Dimensional vector with N representing the number of classes. The fully connected layer is thus used to take the high-level features and correlate them into the particular classes.

Accuracy and Top-K Layer

The accuracy layer obtains the accuracy by comparing it with a target value. This layer is not included in the loss function and thus, has no backward step. It is also known as the Top-K layer as it compares the true label with the top K scoring classes.

3.3.5 Stochastic Gradient Descent (SGD)

Machine learning optimization algorithms are used to formulate an optimization objective or a cost function. Stochastic Gradient Descent (SGD) is one of the most widely used algorithms and works by randomizing the order of the training data and obtaining their gradient descents. The algorithm picks each training data at random, calculates their gradients and iterates over the entire set to obtain the global minima. The randomization of the set ensures that the gradients are not settled for a particular pattern and avoid getting stuck at local minimas. While having a fixed learning rate is common, the learning rate should be reduced over time to ensure the convergence of SGD [63]. The process snippet from [64] is given in the Algorithm 1.

Algorithm 1 Stochastic gradient descent (SGD) update at training iteration k

```
Require: Learning rate $\epsilon_k$
Require: Initial parameter $\theta$
  while stopping criterion not met do
    Sample a minibatch of $m$ examples from the training set $\{x^{(1)}, ..., x^{(m)}\}$ with corresponding targets $y^{(i)}$.
    Compute gradient estimate: $\tilde{g} \leftarrow +\frac{1}{m}\nabla_\theta \Sigma_i L(f(x^{(i)}\ \theta), y^{(i)})$
    Apply update: $\theta \leftarrow \theta - \epsilon\tilde{g}$
  end while
```

3.3.6 Back-Propagation

Back-propagation is the method to adjust the weights as it learns with the training process. The process is divided into 4 steps. The details of the processes are given below.

Forward Pass

During the forward pass, the training images and their corresponding labels are passed through the filters. The weights are randomly initialized for the first forward pass.

Loss Function

The loss function determines the gap between the actual and the predicted results. The loss function may be defined in different ways, for example, the mean squared error or L^2-Norm. Since the network is initialized with random weights, the loss is extremely high in the first rounds of training.

Backward Pass

Once the loss function is determined, a backward pass is performed to determine the gradients which result in the various losses. The gradient, represented by the derivative, is calculated for every layer with respect to the weights.

Weight Update

The final step is to adjust the weights with an opposite direction to the gradient. A learning rate determines the rate at which the weights are updated. A higher learning rate would result in faster learning but might result in losing out on the optimal weights.

3.3.7 FCN AlexNet

Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton developed a “large, deep convolutional neural network” to be competed in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012 [19]. The FCN AlexNet has been trained on the ImageNet Database which contains about 1.2 million high resolution training images to classify objects into 1000 categories. The database also provides about 50,000 validation images, and 150,000 testing images.

AlexNet contains 5 convolutional layers, max-pooling layers and 3 fully-connected layers. The model was trained using batch stochastic gradient descent. Data augmentation techniques were implemented for image translations and were eventually trained using ReLU and dropout layers to incorporate non-linear functions and to combat the problem of overfitting to the training data. The record breaking performance in the competition was a result of training on two GTX 580 GPUs for about five to six days. The AlexNet topology may be found in Chapter 3 in Figure 4.5.

3.3.8 Transfer Learning

The transfer of the layers from a base network to a target network is called transfer learning. The first n layers are copied and the remaining layers are trained for the target task. Yosinski *et al.* [65] figured that the transfer of features from distant tasks is better than setting random weights. The general performance seems to improve with transfer learning for a new task compared to fine tuning the new network. Thus, the system is expected to perform better for a network trained on millions on images and then transferring the weights for a new task which might have Fewer training samples.

3.3.9 Gabor Based CNN

The Gabor filter is a sinusoidal wave modulated by a Gaussian envelope. It behaves as a rotation sensitive local frequency detector. Gabor filters are effective for texture detection as they give the spatial position and the spectral frequency simultaneously [37, 66]. A 2D Gabor filter is composed of a real part and an imaginary part and is represented over the image domain (x, y) as:

$$G(x,y;\lambda,\theta,\phi,\sigma,\gamma) = exp(-\frac{x'^2+\gamma^2 y'^2}{2\sigma^2})exp(i(2\pi\frac{x'}{\lambda}+\phi)) \tag{3.6}$$

where, $x' = x\cos\theta + y\sin\theta$, $y' = -x\sin\theta + y\cos\theta$, σ is the bandwidth of the Gaussian envelope, θ is the filter orientation, λ is the wavelength, ϕ is the phase shift, and γ is the spatial aspect ratio, or the amount the kernel is "stretched".

A Gabor filter bank is composed of kernels with different orientations between 0^o and 180^o. The kernels consist of only the half circle as the same orientations would be achieved for the remaining circle. Convolution of an image with the kernels from the filter bank provides micro-level texture information for the images in the Gabor space.

Yao *et al.* [67] suggested an improvement to the Convolutional Neural Networks by pre-training on images extracted off the Gabor features at three directions, viz. 0^o, 45^o and 90^o. The images are combined to a 3 channel image, and the generated Gabor feature map is used to train a CNN. The model parameters obtained are overlapped with the original images to obtain the results. The tests conducted on the AlexNet with varying learning rates and different combination of convolutional layers, show an improvement of 1.26% in accuracy with the use of Gabor features.

3.3.10 Recurrent Neural Networks

The Recurrent Neural Networks (RNNs) takes into account all the previous data. The process involves looping of a layer for every time step of the input sequence. Thus, a feedback signal is sent by the neurons to the neighboring neurons, the preceding ones or to itself. As suggested by Williams *et al.* [68], the weights are updated at the end of certain time intervals. This loses the real-time significance due to the time interval but it adds a temporal value to the data. The process uses an architecture called Long Short Term Memory (LSTM). It is a type of Recurrent Neural Network designed to contain the information over long periods and is composed of the input and output gates along with memory cells to control the information flow over time. A recent work by Valipour *et al.* [69] demonstrates the implementation of RNNs using temporal data from the previous frames for video segmentation. Future works on RNNs propose the use of multiple recurrent layers for better semantic segmentation results.

Chapter 4

Terrain Classification Method

The challenge with machine learning is to retrieve the true data based on the available information. However, the availability of true data for the desired terrain types is limited considering the absence of human supervised data. To overcome the challenge to having a small dataset, we introduce Gabor filters and pre-trained networks which supplements additional features to determine the effects on the network. As discussed in Section 3.2, there are several methods to evaluate the system. The performance of a pixel based segmentation can be best depicted using a confusion matrix, where high F-Scores represent that both the precision and recall values are good. While Intersection over Union (IoU) scores for each class give the performance of the particular class, the mean IoU scores represent how the model performed for all classes. We design our experiments to minimize the misclassifications between drivable and non-drivable classes.

In this chapter, we present the methods used to extract the different classes. In Section 4.1, we describe the use of the Gabor filters to extract the texture features. To compare the behavior of the networks with different augmented datasets, we explain the selection of neural network models with their evaluation methods in Sections 4.3 and 4.4.

4.1 Gabor Feature Extraction

The images are passed through the Gabor kernel to obtain the texture response of the images. A standard set of parameters was chosen to evaluate the effect of the filter on the neural network. A filter size of 31x31 pixels was used to convolute the image frames of 1280x1080 pixels. The kernel size is chosen such that the small differences in the textures are captured effectively. The parameters chosen are given in the Table 4.1 and the sample kernels for the orientations between 0^o and 180^o are given in Figure 4.2. Figure 4.1 demonstrates the effect of the Gabor filter on the images.

Table 4.1: Gabor filter parameters.

k : Kernel Size	31x31 pixels
σ : Bandwidth	4.0 pixels
θ : Orientation	0^o to 180^o (In 16 steps)
λ : Wavelength	10.0 pixels/cycle
ψ : Phase offset	0^o
γ : Aspect ratio	0.5

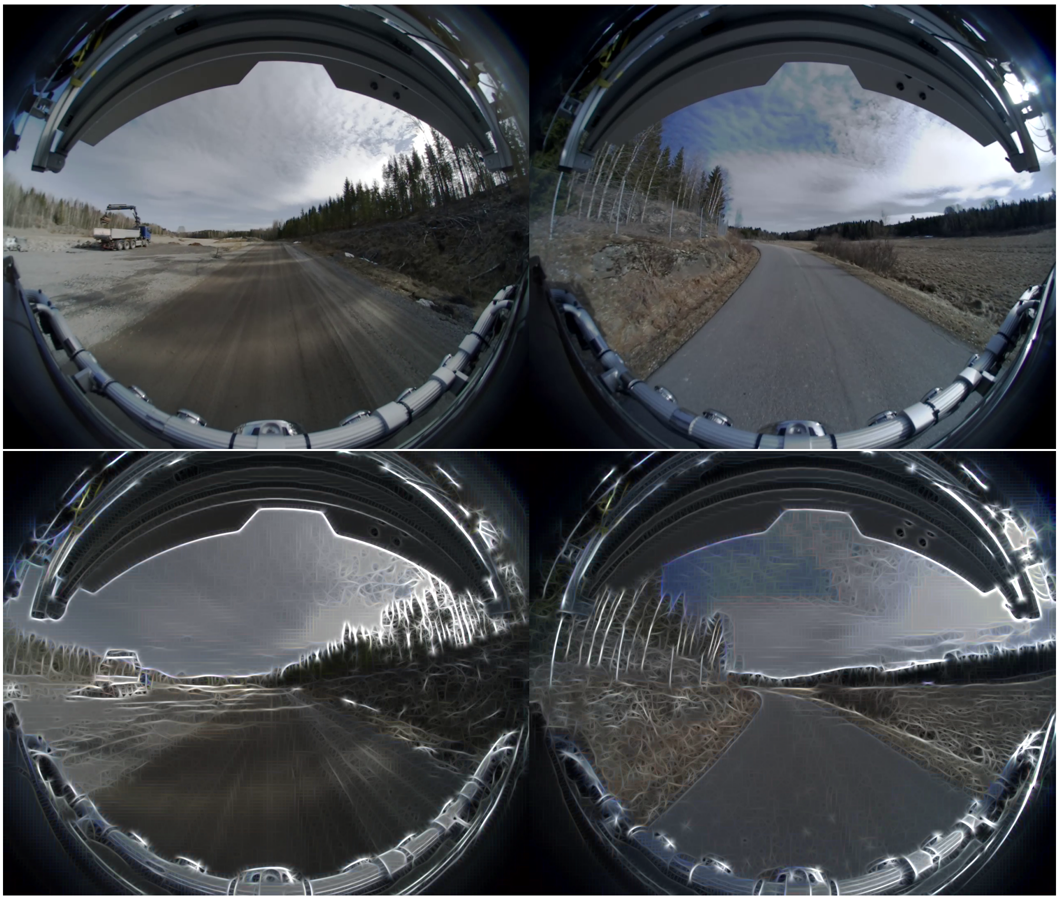


Figure 4.1: Effect of Gabor filter. (Top) Original Images, (Bottom) Gabor response for the Images.

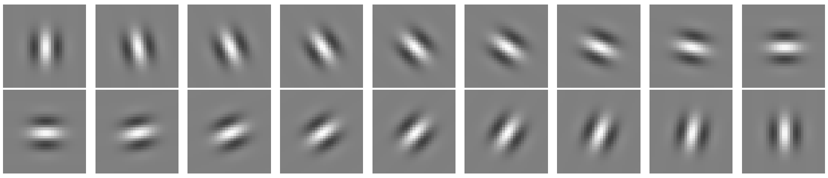


Figure 4.2: Gabor kernels with above specifications for orientations between 0^o to 180^o in 16 steps.

4.2 Inference Evaluation

The Caffe framework [70] requires a certain number of parameters to be able to obtain an inference for any input. We use the python library and to be able to deduce effectively, we must provide the following:

1. Model: The prototxt file describing the neural network based on its available layers.
2. Caffemodel: The weights learned during the training process.
3. Colours: A colour chart in the form of a look-up table. A single pixel row with the order of the pixels representing the class numbers.
4. Data: The image(s) to test.

In the case where the model is trained on processed data, the test data needs to be pre-processed before passing it through the model. A set of transforms is performed followed by a forward pass. The forward pass takes in data in the form of an N-Dimensional array and outputs a cell array. The predictions are generated in the form of scores for each class. The highest score represents the maximum possibility among the given classes. These classes are then chosen and matched with the colour chart to develop a segregated image.

4.3 Neural Network Models

The segmentation task can be performed by any of the popular networks like the AlexNet or GoogleNet [20]. We stick to AlexNet for our experiments and explore the possibilities to improve its result.

With the Gabor response picking out the texture features from the image, we try to incorporate the effective Gabor result into the neural network. To evaluate the Gabor features, we perform 3 rounds of experiments for the deep learning process. We compare the effect of Gabor response with pre-trained networks by applying the same to the default images. The models have been generated for each of the rounds as illustrated in the Figure 4.3. They have been explained below:

1. In Model 1, the deep learning inference is performed on the pre-trained AlexNet neural network (Figure 4.5) with the default images and their corresponding labels.

2. In Model 2, the deep learning inference is performed with the Gabor response images on the same pre-trained AlexNet neural network.

3. In Model 3, the deep learning inference is performed again with the Gabor response images but the model has been trained on the neural network model from Model 1. Thus, the network has already been trained on the given data and has been improvised by the provision of the texture details from the Gabor filter.

In the Model 2, all the data is pre-processed to obtain the Gabor responses over the original images. On the other hand, Model 3 uses the Gabor images and then applies them on the network inferred from the Model 1. To obtain the inference, the test images are also pre-processed with the Gabor filter and run on the models obtained. This has been illustrated in the Figure 4.4 The provision of the Gabor features to the network highlights the texture features for the images.

4.4 Performance Evaluation

A huge training data is optimum for training the models and similarly, a huge and randomized selection of the test data helps in justifying that the network performs well under diverse situations. However, a large number of pixels from the test images fills the confusion matrix with huge values, making it hard to analyze. Thus, we use the precision and recall values to indicate if the classes are segmented properly. All the performance metrics like the accuracy, precision, recall, F-scores, and mean IoU values, range from 0 to 1, where 0 represents poor performance and 1 represents the best performance.

As discussed in Section 3.2, accuracy does not provide the best evaluation criterion, and the precision and recall values are best judged with the F-Scores. Thus, for our experiments, we rely on F-Scores and IoU values. The individual IoU values for the classes represent how a model performs for the particular class and thus we evaluate if the most relevant drivable classes perform well.

For the simplicity of the reader, we have segregated the scores into 3 categories, viz. 0.0 to 0.4, 0.4 to 0.8, and 0.8 to 1, represented by red, yellow and green depicting poor, average and good performances. The categorization is only for highlighting the relative performance of classes for different models.

We use the test data on the different models to compare the results with the ground truths. We also test our models on images from the internet to observe the responses, however, there was no ground truth provided for these cases to obtain their performances.

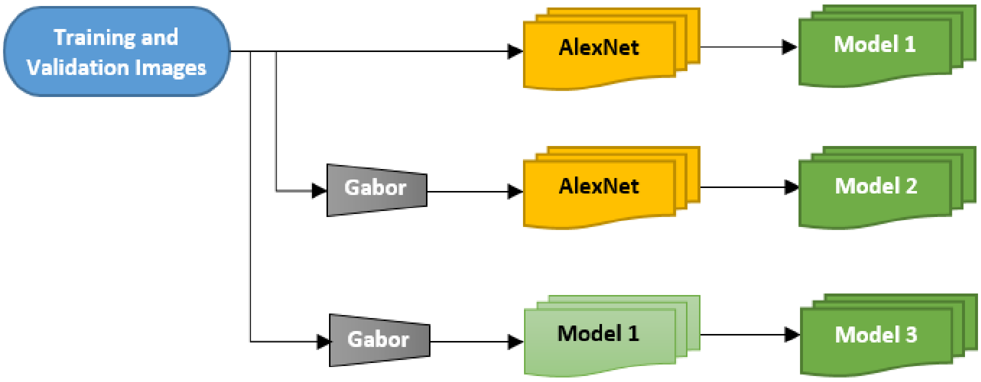


Figure 4.3: Flowchart to demonstrate the generation of the 3 models.

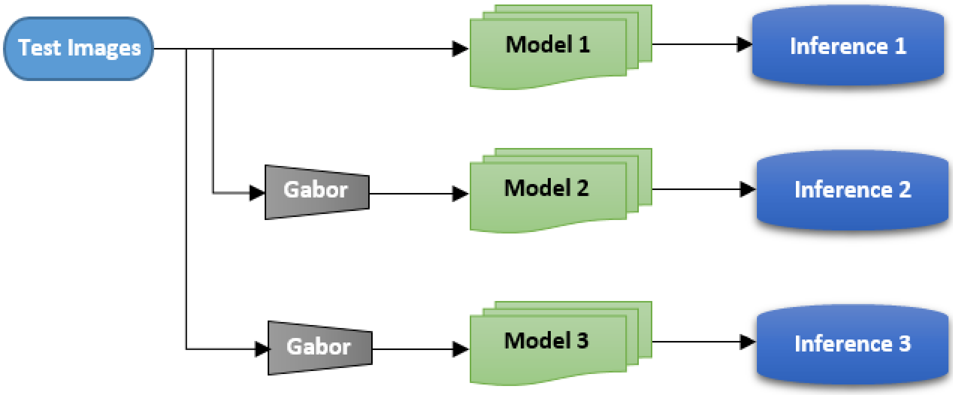


Figure 4.4: Flowchart to demonstrate the inference for the 3 models.

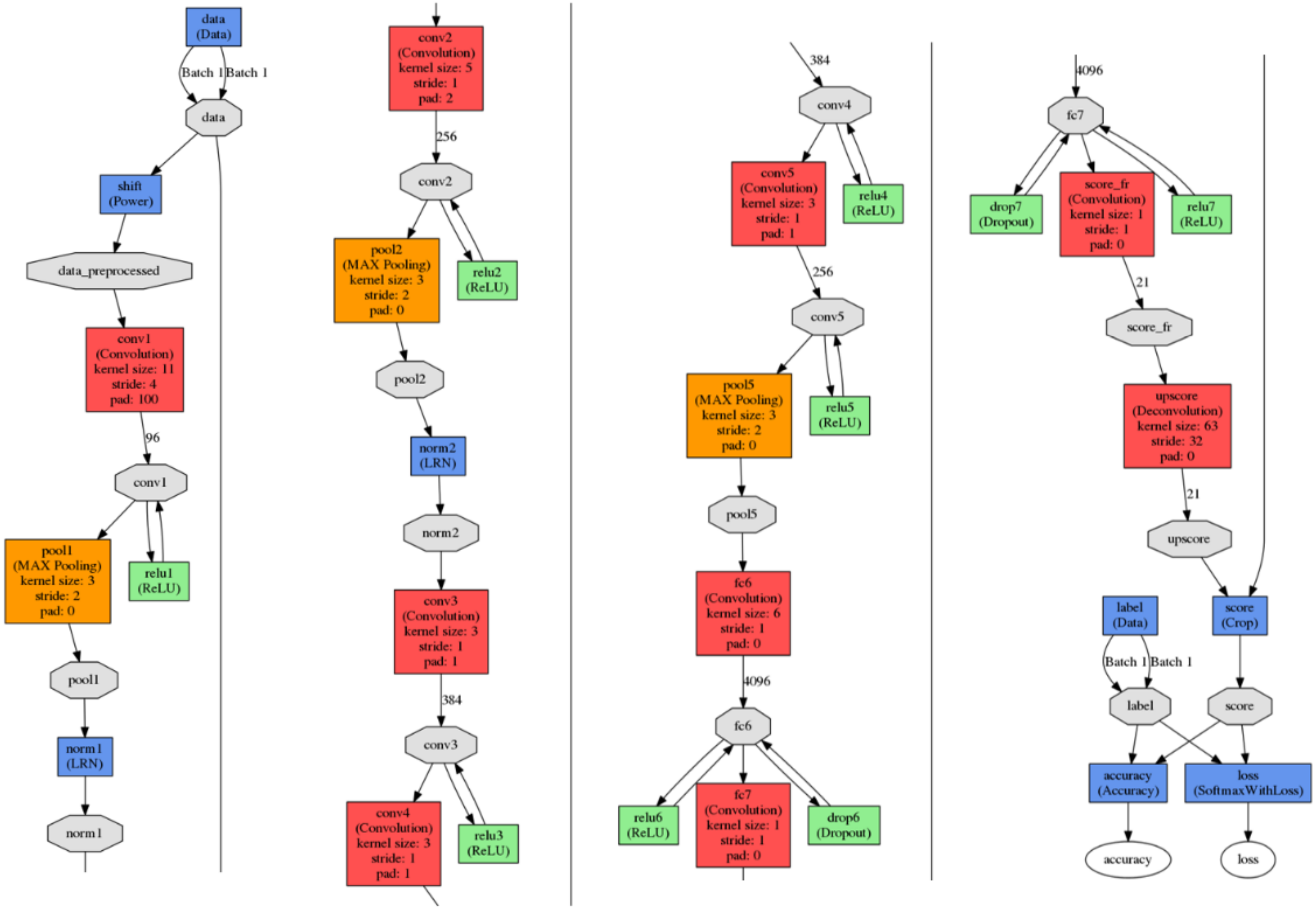


Figure 4.5: AlexNet Topology (The four columns are connected one below the other).

Chapter 5

Datasets

In the field of machine learning, the choice of having a supervised or unsupervised learning mainly depends on the available data, where having a huge amount of known data is required for a good training set. Deep learning algorithms use models defined by a large number of parameters which are obtained from the available datasets. For a supervised training model, the training data needs to be annotated. Due to the limitation in the time and scope of the thesis, we relied on a smaller dataset and analyzed the effects of the network. The Appendix A gives the details on the hardware and the software used in the process.

In this chapter, the data collection and its augmentation process have been focused upon. Sections 5.1 and 5.2 describe the dataset and rectification process performed. The choice of classes defines the effective resultant of the classification process. Section 5.3 gives a detailed account on this choice by the distribution of the classes. The Sections 5.4 and 5.5 describe the data augmentation and the splitting for the training and test phases used in the experiments.

5.1 Image Dataset

The data was collected by using the Drive PX2 system with the Omnivision camera fixed on the windscreen of the Scania truck. The videos were recorded at the training center of Scania, over a span of 3 days for an average of 1 hour each day during the daytime. The placement of the camera is shown in Figure 5.1

The videos were obtained at 30 fps and ensured to observe a high variance in the weather and road conditions. A set of 100 images were picked from about 50,000 frames from the videos taken from the Scania training center. The images were picked considering different road and weather conditions to engage maximum information to normal driving conditions. The selection of images became an important assumption considering that it relates to a typical off-road terrain. A typical off-road terrain, here, refers to the roads without lane markings, with muddy regions, water ponds, etc. However, there was an absence of other vehicles, pedestrians or similar obstacles during the entire recording. While this is not a real-life scenario,

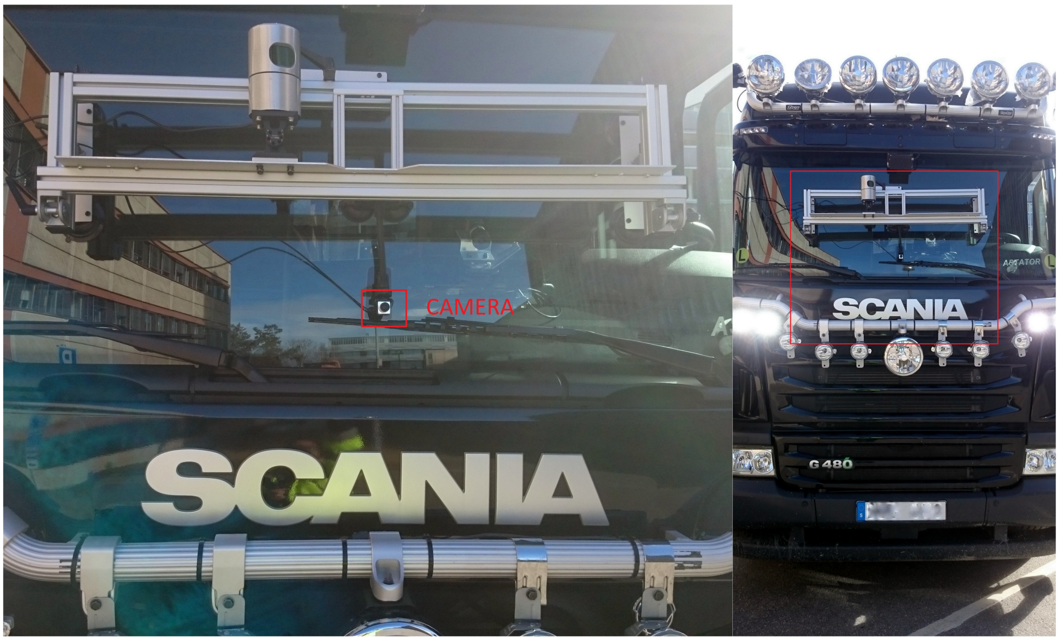


Figure 5.1: Camera placement on the truck.

it resulted in a focus primarily on the ground surface classification and avoiding the additional task to eliminate the obstacles.

The images were manually annotated using the LabelMe software [71]. Each of the polygon sections was carefully marked based on the human-understanding of the categories of classes. The XML files generated from the LabelMe were used to obtain PNG mask images of the same size as the images. The black pixels in the mask represents the background, not categorized as any of the known classes while the other RGB colour pixels indicates the corresponding classes. Different augmentations were performed on these 100 images to obtain multiple datasets. This is described in Section 5.4.

5.2 Image Undistortion

The 190° FOV fish eye lens generates curved images like the ones given in the top images of Figure 5.2. To calibrate the intrinsic parameters, we use a 9x7 checkerboard pattern with each square size of 0.041m as given in the Figure 5.3. We use about 20 images with the checkerboard pattern to calibrate the camera. Using the OpenCV functions, the checkerboard corners are used to determine the camera matrix K and the distortion matrix D. They are given as follows:

$$K = \begin{bmatrix} 532.19748493 & 0 & 608.45088493 \\ 0 & 531.23989078 & 552.83389049 \\ 0 & 0 & 1 \end{bmatrix}$$

$$D = \begin{bmatrix} -0.27918479 & 0.06520399 & 0.00043639 & 0.0031953 & -0.00600766 \end{bmatrix}$$

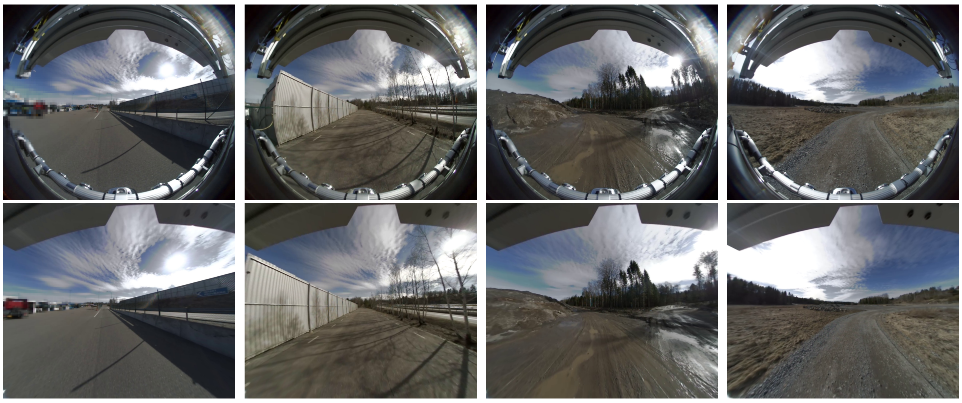


Figure 5.2: Effect of Undistortion. (Top) Original Images, (Bottom) Calibrated Images.

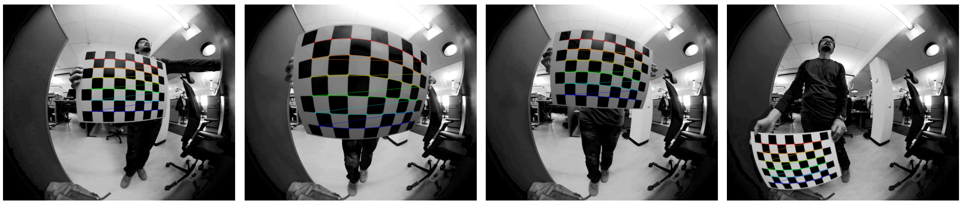


Figure 5.3: Calibration With Checkerboard Pattern.

5.3 Class Distribution

Determination of the type of road requires the training images to specify the relevant road surface types. Selections of 16 and 9 classes were chosen based on the observation of most frequently driven road surfaces. A 3 class distribution was also chosen to categorize the drivable regions only. An illustration of the 16, 9 and 3 class distributions is given in the Figure 5.4. It is also important to note that these classification categories are different from the benchmark datasets like the Cityscapes or PASCAL-Context Dataset which are directed towards detecting objects rather than terrain types.

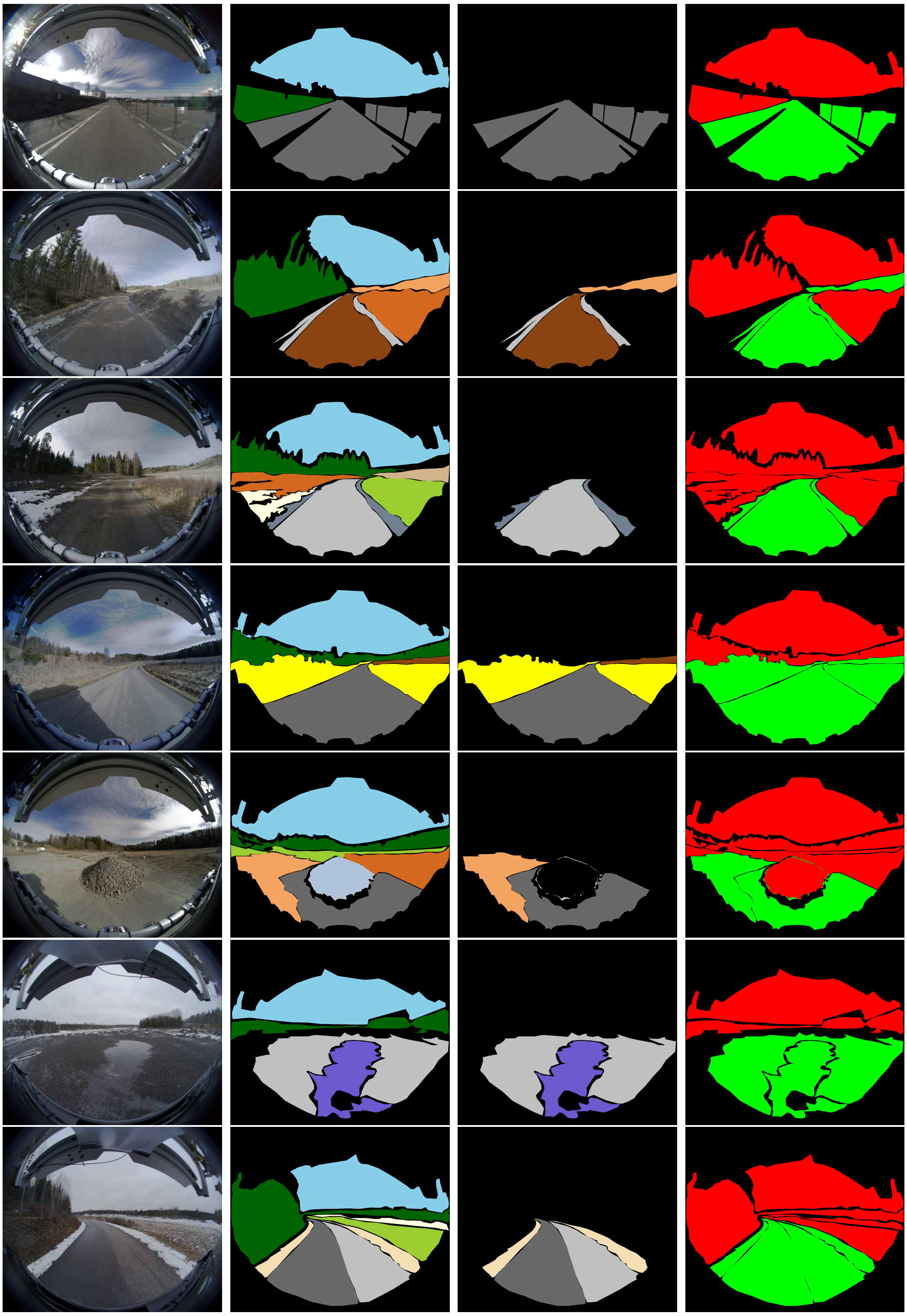


Figure 5.4: Manual annotation masks for each class distribution. (Columns from Left to Right) Original images, 16 class masks, 9 class masks and 3 class masks.

5.3.1 16 Class Description

The 16 class distribution takes into consideration the entire image and tries to categorize all the subjects irrespective of the relevance to the vehicle. It compiles of 8 types of drivable classes, 7 non-drivable classes, and the background. Apart from the drivable classes, the 7 non-drivable classes are gravel (referring to gravel heaps), mud, sand, sky, vegetation, grass, and snow. These are entirely based on the human understanding of non-drivable regions of the road and mostly relate to the regions away from the normal pathway. This type of classification is useful in understanding the environment and estimating the damage for prediction errors, for example, a non-drivable grass may be of less damage than the vegetation class, corresponding to trees or heavy bushes. Similarly, the data from the class sky may be utilized to understand the lighting conditions. There is a lot of information due to the number of classes and it is hard to have a consistent class determination due to the rapid changes of the predictions. A detailed list of the classes can be seen in the Figure 5.5.

5.3.2 9 Class Description

The 9 class distribution picks only the 8 drivable classes and marks all other classes as background. The classes include asphalt, high-density gravel, low-density gravel, mud, sand, water ponds, grass, snow, and background. This would let the vehicle prioritize the motion based on the road type. The dependency of the road surface for the vehicle can be easily understood by the manner in which a human driver drives on different terrains. For example, asphalt surfaces can allow the vehicle to run at higher speeds while a snow surface needs to be driven cautiously.

5.3.3 3 Class Description

The 3 class distribution segregates the 16 classes to define them under drivable, non-drivable and background classes. This is how most of the benchmark datasets are defined which categorize different objects and enlist road as a single class similar to other classes such as car, trees or buildings. Thus, in this case, all sorts of obstacles and non-drivable regions are categorized as non-drivable and any non-annotated regions are marked as the background.

5.4 Data Augmentation

Augmenting the training data not just helps in increasing the dataset but also helps the training network to not overfit the available data. Neural networks are said to be data-hungry as more the data, higher the likelihood of better accuracy towards missing data [72]. There are different methods of data augmentation like flipping, stretching, panning, zooming, gray scale, colour histograms, elastic transformations, etc. However, it is important to ensure that the data generated does not muddle up the logic of the real data, for example, an image representing the sky and road should not be flipped vertically as the sky always remains over the horizon, at the top of the image (unless the camera is rotated).

An approximate method to augment the data is by picking the nearby frames from a high frame rate video and annotating the nearby frames similar to the one with the known annotation. This is not an accurate method as the frames need not be exactly similar and might lose out on essential data. However, this near-scene annotation may help generate additional training information as it bridges the semantic gap between the images.

5.4.1 682 Images Dataset (Set 1)

The videos were captured at 30 fps. This implies that 3 image frames are captured in a span of 0.1 seconds. Using the neighboring frames from the videos, an approximation is made that the image frames captured within a few hundred milliseconds will not observe a major difference from the given annotation mask images. With this approximation, an average of 2 frames was picked from the images before and after the selected annotated image. Further on, the images were flipped horizontally giving the scope of horizontal inversions. Providing a mix of these images with the corresponding ground truths, a dataset of 682 images was obtained.

5.4.2 1364 Images Dataset (Set 2)

The undistortion of the images was considered in this case and was combined to the existing dataset. This type of dataset is relevant when the camera type is unknown and undistortion of the data is an additional processing cost. The availability of the given images and the undistorted images maintains the semantic nature of the classes for their relative positions in the image.

5.4.3 6820 Images Dataset (Set 3)

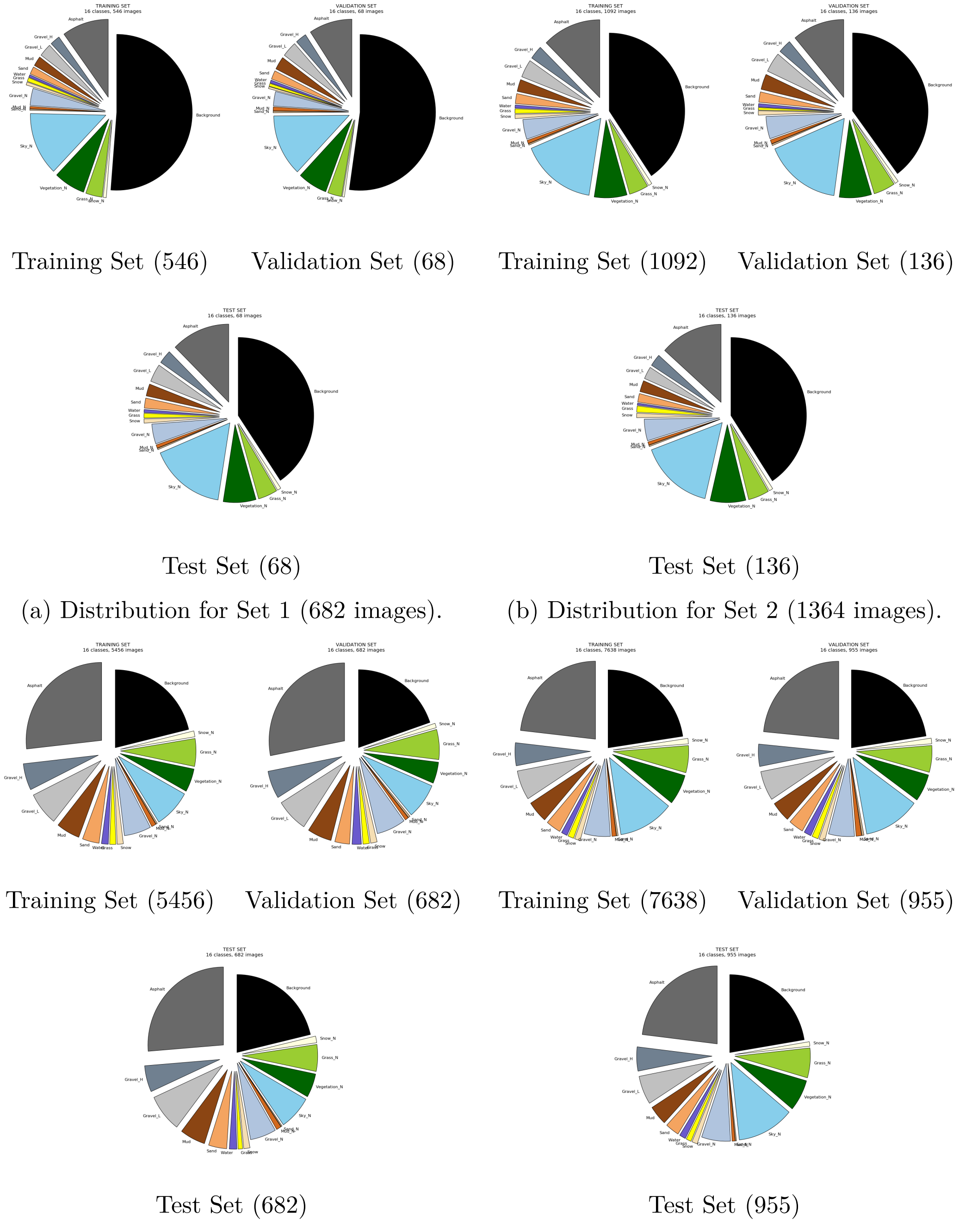
In this case, square images of 256x256 pixel size were taken. We focused on the lower half of the image with the understanding that the relevant drivable classes are seen closer to the vehicle. The dataset composed of cropped sections of the lower half of the image and reduced scaled sections of the whole image. Thus, a majority of the road portions are perceived from the lower half while simultaneously containing portions of the whole image.

5.4.4 9548 Images Dataset (Set 4)

The dataset is similar to the previous one, however, in this case, cropped sections from the whole of the image are used irrespective of the position in the image. This is done to ensure a uniform spread of the 16 classes and provide a training which is independent of the environment. The main reason is to understand the effect of using small sized images for training and providing more images with the available classes.

5.5 Data Split

Each of the datasets is split in an 80:10:10 ratio where 80% of the data is used for training, 10% for validation, and 10% for testing. The images are selected at random and are assumed to contain all classes in all the categories. However, this is not the case in all the situations, which might result in abnormalities in the smaller datasets. It is eminent that 68 images as test data for the Set 1 do not suffice to all conditions. An estimate of the 16 classes to each of the datasets has been given in the Figure 5.5. Further categorization to 9 and 3 classes may be understood by combining the corresponding classes. The legend displays all the 16 classes where the first 8 classes are the drivable classes, next 7 representing the non-drivable classes and the last being the background. The non-drivable classes have been represented by '_N'.



Training Set (546) Validation Set (68) Training Set (1092) Validation Set (136)

Test Set (68) Test Set (136)

(a) Distribution for Set 1 (682 images). (b) Distribution for Set 2 (1364 images).

Training Set (5456) Validation Set (682) Training Set (7638) Validation Set (955)

Test Set (682) Test Set (955)

(c) Distribution for Set 3 (6820 images). (d) Distribution for Set 4 (9548 images).

⬛	Asphalt	🟧	Sand	⬜	Gravel_N	🟩	Vegetation_N
⬛	Gravel_H	🟪	Water	🟧	Mud_N	🟩	Grass_N
⬜	Gravel_L	🟨	Grass	🟫	Sand_N	⬜	Snow_N
🟫	Mud	⬜	Snow	🟦	Sky_N	⬛	Background
Drivable Classes				Non - Drivable Classes			

Figure 5.5: Class distribution for all the Sets.

Table 5.1: Distribution of classes (All values in percentage).

CLASSES	Set 1 Training (546)	Set 1 Validation (68)	Set 1 Test (68)	Set 2 Training (1092)	Set 2 Validation (136)	Set 2 Test (136)	Set 3 Training (5456)	Set 3 Validation (682)	Set 3 Test (682)	Set 4 Training (7638)	Set 4 Validation (955)	Set 4 Test (955)
Asphalt	10	9.3	9	12.6	11.2	13.6	26.8	28	26.3	23.2	23.5	23.1
Gravel_H	2.1	2.4	2.7	2.8	2.8	2.5	5.5	5.8	5.5	4.8	4.6	5
Gravel_L	2.7	3.2	2.9	3.7	4.1	2.6	6.9	6.9	7.8	6.1	6.1	5.9
Mud	2.1	2.7	1.6	2.5	3.2	2.3	5.1	5.2	5.5	4.5	4.2	4.1
Sand	1.6	1.8	1.4	2.1	2.1	1.8	3.6	3.2	3.9	3.2	3.2	3
Water	0.5	0.6	0.3	0.6	0.7	0.4	1.5	2	1.6	1.3	1.6	1.3
Grass	0.8	0.6	0.7	0.9	0.5	1.3	1.3	1.4	1	1.3	1.3	1.1
Snow	0.6	0.4	0.7	0.9	0.9	0.9	1.3	1.4	1.3	1.2	1.2	1.2
Gravel_N	3.5	3	3.8	4.1	4.6	4.5	5.8	6.2	5.6	5.6	5.6	6.2
Mud_N	0.5	0.8	0.9	0.7	0.8	0.7	1	0.8	1.1	0.9	1.2	0.7
Sand_N	0.3	0.3	0.4	0.3	0.3	0.3	0.3	0.4	0.2	0.3	0.4	0.2
Sky	13.1	12.8	12.7	16.2	16.4	15.6	8	7.4	7.1	11.9	12.1	12.2
Vegetation	6.7	6.4	6.6	7	6.9	7.6	4.9	4.4	5	6.3	5.7	6.5
Grass_N	3.5	3	3.8	4.1	4.6	4.5	5.8	6.2	5.6	5.6	5.6	6.2
Snow_N	0.7	0.4	0.6	0.8	0.8	0.8	1.2	0.9	1.4	1.1	1.2	0.9
Background	51.2	52.3	51.9	40.6	39.9	40.8	21.2	19.8	21.3	22.6	22.6	22.4

Chapter 6

Results and Discussion

In this section, we study the effects of the different classes and their responses to the neural networks. Section 6.1 describes the effect of Gabor filters being applied on the images and Section 6.2 gives a detailed evaluation of the results of the deep neural networks.

6.1 Effect of Gabor Filters

Gabor filters are said to give promising results for distinguishing textured surfaces indoors, like segregating the foreground from the background. We can see from the Figure 6.1 for indoor environments that the objects like furniture, lamps, clothes and window panes are highlighted. Primarily, the road classification requires the specification of the main types of drivable surfaces like asphalt, gravel, mud or grass. Gabor filter acting as a texture classifier is expected to categorize these different types of surfaces. Image frames from the videos were randomly picked to figure out their response towards the filter and the resultant images were able to classify the asphalt road with distinguishable textures, similar to the one obtained indoors.

We notice from Figure 6.2 that the filter generates good results to distinguish smooth and rough surfaces like the gravel road from mud or mud from the snow. It can be seen that the asphalt road is being classified well with all the lane borders and markings distinguished as highlighted features. The gravel lane exhibits a rough surface whereas the drivable mud between them is smooth. Similarly, the snow is very smooth and the rough mud is segregated. Also, we see a very striking feature for the non-drivable gravel heap, which gets segregated from its neighboring flat gravel due to its coarse nature.

However, we also observe mixed results for other terrains. For example, while observing the characteristics of the Gabor filter for the gravel and grass regions, there is hardly any distinguishing partition. It can be seen from the Figure 6.3 that the features being picked for the drivable surfaces are barely distinguishable. The smooth mud merges with grass and water, and also, the shadows observed on the mud are hardly recognizable. The submissive feature observed visually depicts an uncertainty to use texture features independently.

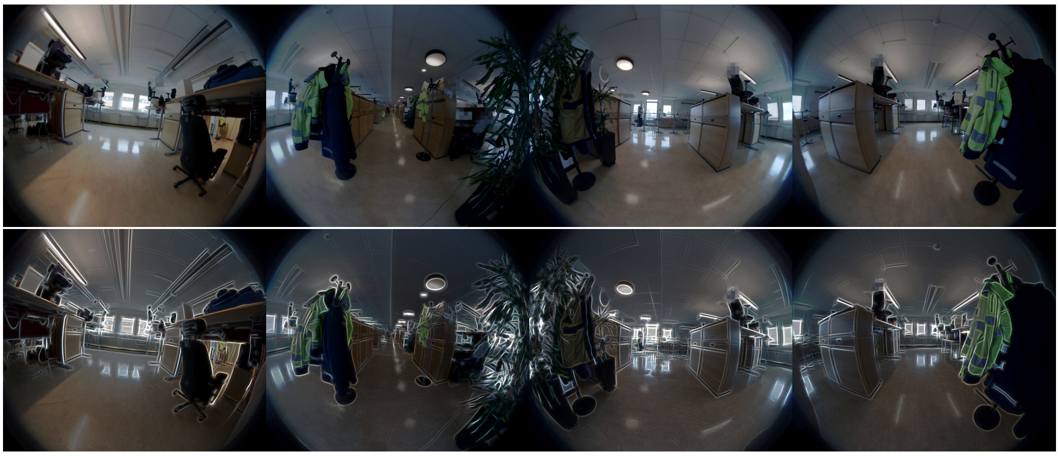


Figure 6.1: Gabor response for indoor images. (Top) Original and (Bottom) Gabor responses.

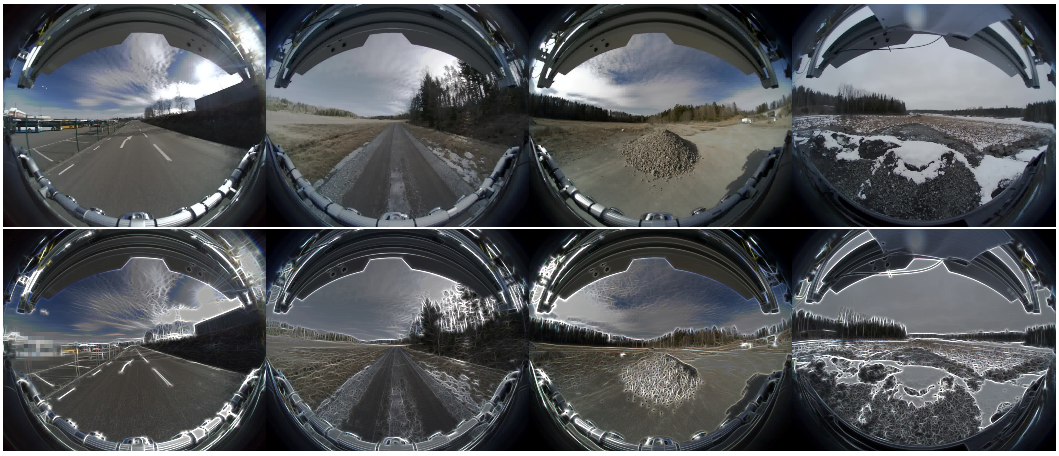


Figure 6.2: Conditions when Gabor filter performed well. (Top) Original and (Bottom) Gabor responses. (L to R) Asphalt road classification, Gravel and mud classification, Gravel pile classification, Mud and snow classification.

6.2 Deep Learning Inference

In this Section, we evaluate the results from the deep learning method. We hereby compare the effects of undistortion of images, the number of classes, the number of images in the dataset and Gabor filters. We also try to evaluate the effect of these factors with the application of different models by testing their effects on the

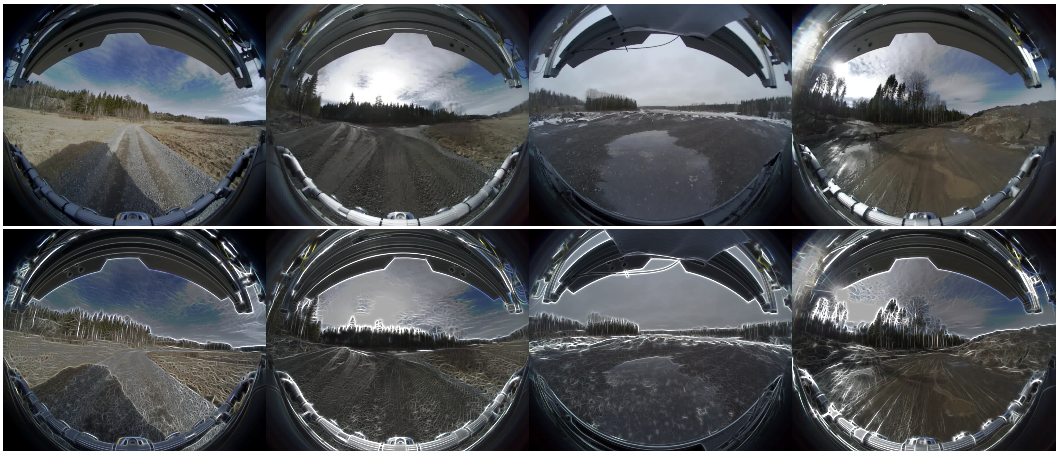


Figure 6.3: Conditions when Gabor filter performed poor. (Top) Original and (Bottom) Gabor responses. (L to R) Grass and gravel classification, Mud and grass classification, Mud and water classification, Shadows on muddy road.

augmented test images. The major requirement is to minimize the misclassifications for drivable classes.

The training times for Sets 1, 2, 3 and 4 (as defined in Section 5.4) took about 40, 90, 70 and 90 minutes respectively for each of the models. The duration of training depended on the size and number of images in the Sets. However, we note that the inference took about 400ms for all the sets.

To analyze the results of this machine learning algorithm, we generate confusion matrices by obtaining the number of pixels of each class from the ground truth and comparing them with the same from the predicted results. We also compute the performance metrics for each of the classes.

We thus evaluate the precision and recall tables from the confusion matrices. Due to the ample amount of data, we only presented the precision tables for Model 1 in Appendix D. The values in the tables are given as the percentage of relevant pixels and have been rounded off to 2 decimal places.

We present the mean IoU scores and the individual IoU scores for all the class distributions in Figure 6.4 and 6.5.

6.2.1 Annotation Guide

From now on, we order the classes based on the priority of drivability, where asphalt gets the highest priority being considered as very safe to drive and the non-drivable snow given the least priority. The priority is based on human understanding of these classes and does not necessarily need to be the same for all driving conditions.

Model 1 represents the model trained with original images, Model 2 represents the model trained on Gabor response of the images, and Model 3 represents the model trained on the Gabor responses from Model 1. The names of the models are shortened and marked as M1, M2, and M3 for representation purposes. A detailed explanation can be found in Section 4.3.

We generated 4 sets of data for evaluation purpose comprising of different number of images. We label the sets with 682, 1364, 6820 and 9548 images as Set 1, Set 2, Set 3 and Set 4 respectively. The details about the sets can be found in the Sections 5.4 and 5.5.

To focus on the important values in tables 6.1 to 6.4 and Appendix D, a colour coding has been maintained. Green colour corresponds to the diagonal elements representing the true positives. Yellow colour corresponds to the misclassifications with inaccuracy ranging between 10% and 20%. Yellow also highlights the last row where the background misclassification becomes a point of concern. Red colour corresponds to the misclassifications requiring high attention, pointing to the inaccuracies of over 20%. Certain levels of blue colour are also used to denote the mismatch when the same class is interchanged with the non-drivable class. High-intensity blue denotes higher levels of mismatch.

6.2.2 Effect of Camera Calibration

Intrinsic calibration results in undistortion of the images. This provides a standardized platform for all cameras. However, it is a challenge to ascertain the same pixel densities for all sections of the image, like the center compared to the edges. The fish eye lens provides maximum pixel to pixel information at the center and minimum at the corners. This results in lower texture information towards the corners. For this experiment, the images were annotated before the undistortion to avoid losing the pixel information.

It must also be noted that undistortion of the annotation masks resulted in sampling artifacts due to geometric image transformation. The interpolation of the annotation masks introduced pixels with different RGB values between the edges of the classes. These pixels values were a blend of the 2 RGB values. To avoid any undesirable classes these pixel values were further processed to be interpreted as the background.

6.2.3 Effect of Data Split and Uneven Class Distribution

The 80:10:10 distribution of the dataset for training, validation and test phase can be seen to have a similar distribution in the split data. On most cases, due to the small dataset certain images with smaller classes are missed from the test or validation sets. However, to ensure that we do not lose any class, the dataset has been engineered to ensure augmentation of images. The smaller classes like water, snow and non-drivable mud were multiplied and more frames closer to these images were added to the dataset. Thus, a notable amount of augmentation can help in curtailing the problem of lack of classes in the test data.

A homogeneous distribution of classes in the dataset ensures that all classes are given equal preference during training and testing phases. However from Figure

5.5, we note that there is a huge variance in the class distribution. It can be clearly seen from Figure 5.5a that the background class dominates for Set 1 with over 50% pixels. This background percentage reduces for Set 2 in Figure 5.5b with the other negligible classes tending towards values crossing 1%. A drastic change is noticed for Set 3 and Set 4 in Figure 5.5c and 5.5d, where the background class has reduced to less than 25% and the other classes contribute to a better uniform distribution. It should also be noted that the background class in the 9 class distribution accounts for the background pixels as well as the non-drivable class pixels. Thus, there is always a higher number of background pixels for the 9 class distribution.

Considering the tests were conducted with different sets for all the class distributions, we try to evaluate the variation in the test results. From the tables in Appendix D, we do not observe a significant difference in the precision values. However, we can note the differences between the IoU scores for each class for the 682 (Set 1), 1364 (Set 2), 6820 (Set 3) and 9548 (Set 4) from the Figure 6.5.

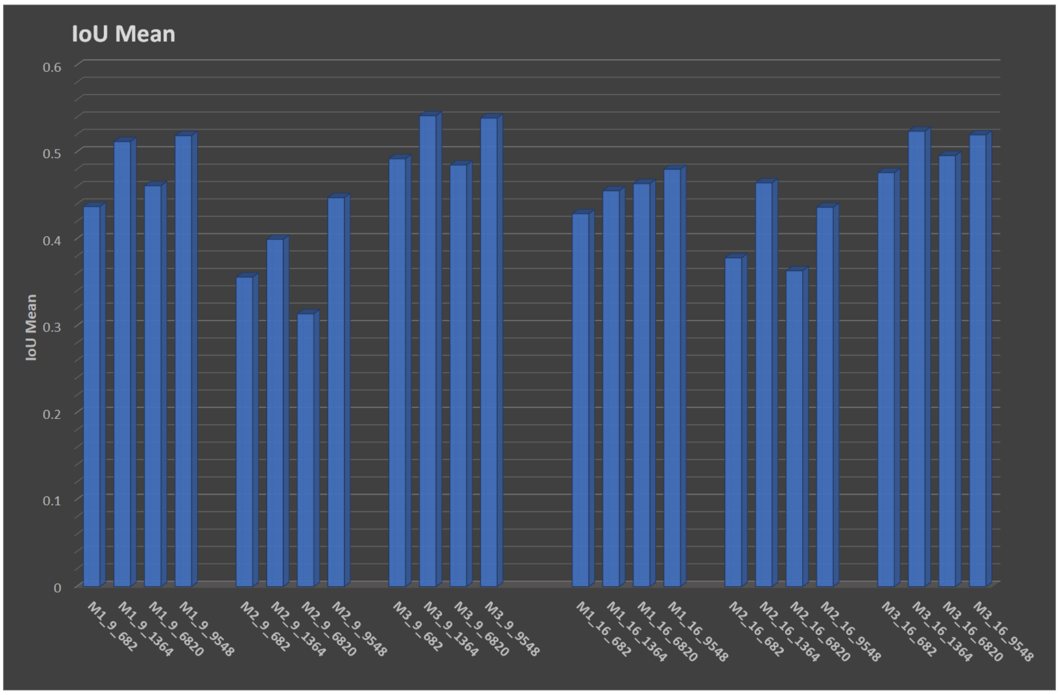


Figure 6.4: IoU mean scores for all Sets. M1, M2 and M3 correspond to the Models as described in Section 6.2.1.

6.2.4 Effect of 16 and 9 Class Distribution

For both 16 class and 9 class distributions, the last row of the confusion matrix is most dangerous where the background is being classified as drivable. Among the drivable classes, most of the misclassifications are observed for the least priority classes, namely, grass and snow. This mostly seems to occur because parts of the grass and snow were annotated based on the individual's understanding of drivable or non-drivable, and thus, sometimes have been considered as background in the ground truth. On the other hand, the last column represents the drivable regions classified as background. This scenario exhibits the fact that these regions will not be driven upon even they were drivable. The low values represent that a lot of data loss occurred during predictions. The IoU scores for 16 and 9 class distributions are given in Figure 6.5.

16 Classes

For Set 1 given in the Table 6.1, certain drivable regions reflect precision values less than 50% while most of the non-drivable classes have been classified well. It can be noted that a lot of mismatches occur between drivable gravel and mud. The blue colour cells denoting the mismatch between the same drivable and non-drivable classes reaches up to 20.93% for non-drivable grass classified as the drivable grass. Another prominent mismatch for the vegetation which is categorized as drivable grass with a 5.14% precision. This condition poses a high threat to the reliability of class distribution. However, it can also be seen that the non-drivable mud is classified with 99.16% precision which is quite high for a class but this might have been caused due to the low number of images for this particular class.

For Set 2 given in the Table 6.2, the precision of the non-drivable mud has seen a significant drop from 99.16% to 57.3%. The reason behind this shift is probably a result of the undistortion process. Non-drivable mud lying mostly at the edges of the images has a change in pixel densities with undistortion. It can also be noted that there has not been any significant change in the last row where the background is being classified as drivable regions.

For Set 3 given in the Table 6.3, it can be seen that the misclassifications between same classes of drivable and non-drivable regions have significantly reduced. Set 3 incorporates the lower cropped images and thus, images containing the reference to the nearby classes are prioritized over the classes from the whole image.

For Set 4 given in the Table 6.4, it can be noted that the misclassifications due to the same classes have increased compared to Set 3. The simple justification for this is the consideration of the whole image portions as compared to the focus on lower half in Set 3. The inclusion of irrelevant classes like the sky reduces its performance and thus, it requires a more detailed evaluation.

Considering the 4 Sets, we can provide an effectiveness based on the classifications as Set 1 performing the weakest, Set 2 and Set 4 being partially better while Set 3 performing the best. We can provide another relation with respect to the last row of misclassifications where the background is being detected as drivable. In this

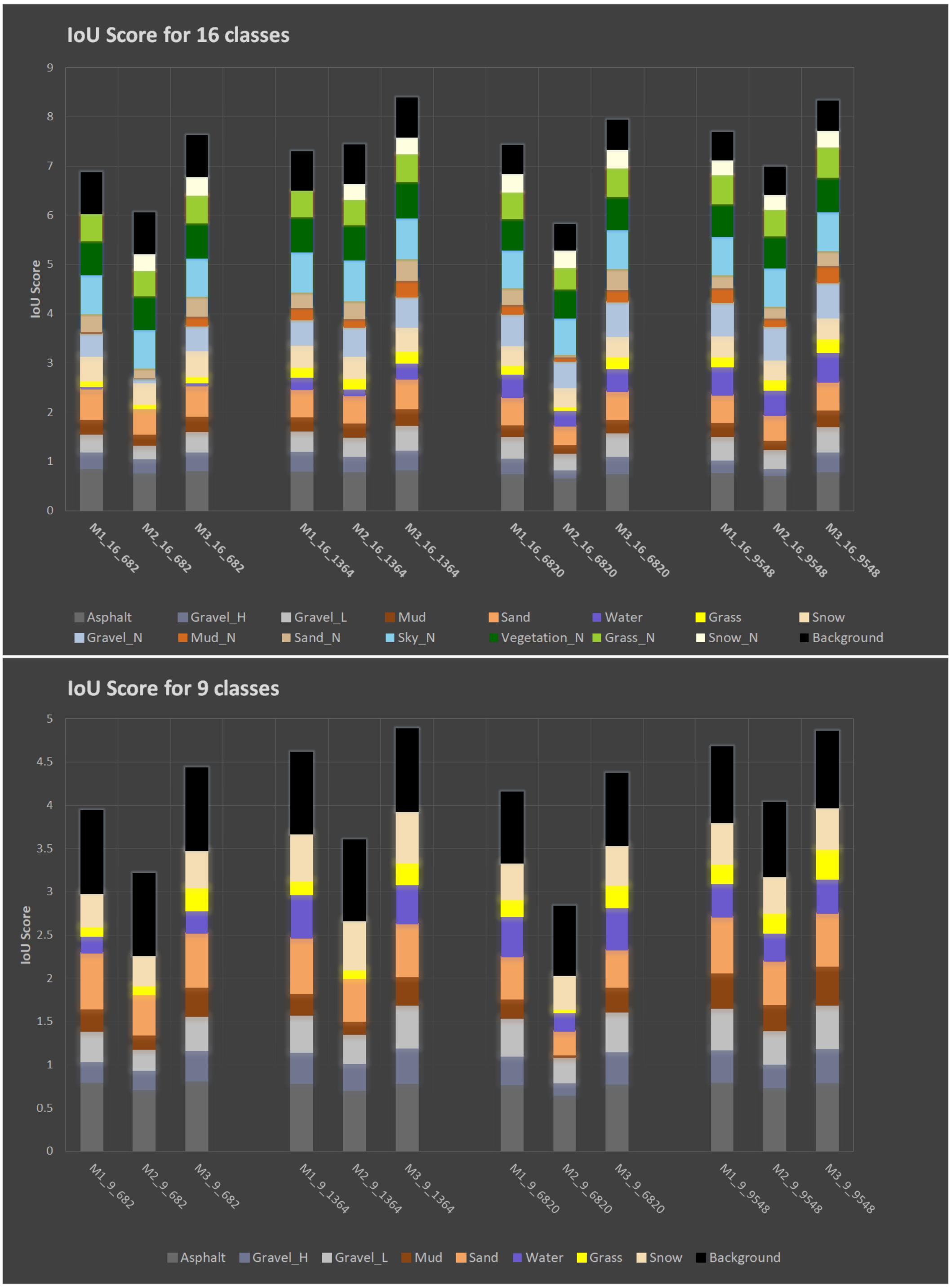


Figure 6.5: IoU scores for individual classes for 16 classes (Top) and 9 classes (Bottom). M1, M2 and M3 correspond to the Models as described in Section 6.2.1. Under perfect conditions, each colour should correspond to 1 unit in y-axis.

case, Set 4 performs the worst, while Set 1, Set 3 and Set 2 can be rated moderately better.

9 Classes

The precision tables for the 9 class is given in Appendix D with the Sets 1 to 4 given in tables D.1 to D.4 respectively.

We note from Set 1 that a major misclassification occurs where asphalt is predicted as water pond marked at 50.48%. For this class distribution, the last row of the background is extremely dangerous compared to the 16 classes as the misclassification could either be irrelevant like the background, or be disastrous like the non-drivable class, which could be an obstacle or a potential accident. Here, we tend to notice a pattern specifically in the first column where the precision decreases for asphalt based on the priority order.

The last row still remains a major concern for Set 2 and requires high precaution. However, we notice lesser misclassifications than Set 1, implying some robustness due to the image undistortion.

We note in Set 3 that the misclassifications between the classes like gravel, mud, sand, and water seem to have increased. This is most likely because Set 3 focuses on the lower part of the image giving priority to possible drivable regions of the image, and the cropped regions of the image are not wide enough to estimate the whole class.

For Set 4, the misclassifications reduce drastically. This set picks out the cropped regions from the whole image and does not focus on the ground regions like Set 3. Thus, the results depict the absolute relation, providing a possibility of positioning the estimate in the entire image.

6.2.5 Effect of 3 Class Distribution

The 3 class distribution consists only of the drivable, non-drivable and the background classes. Any part of the image that is not annotated belongs to the background and it is hard to specify what kind of surface or class it might represent.

The precision tables for the 3 class distribution are given in Appendix D with the Sets 1 to 4 given in tables D.5 to D.8.

The true positives are seen over 89% for drivable class and attain over 81% for both the non-drivable class and background. It should also be highlighted that a drivable class being sensed as non-drivable or background may not possess a huge problem as a vehicle will avoid such classes.

However, in the case of such specific class distribution, it is highly important that all other misclassifications are near zero. A background region classified as drivable

Table 6.1: Precision table for M1, 16 classes, 682 images.

CLASSES	Asphalt	Gravel_H	Gravel_L	Mud	Sand	Water	Grass	Snow	Gravel_N	Mud_N	Sand_N	Sky_N	Vegetation_N	Grass_N	Snow_N	Background
Asphalt	91.94	0.29	1.51	4.25	11.31	0.86	5.31	1.31	2.02	0	0	0.01	0	0.34	0	0.55
Gravel_*H*	2.09	50.08	12.75	20.34	3.1	3.13	0.85	0.11	0	0	0	0	0.08	1.67	0	0.4
Gravel_*L*	0.59	22.31	61.81	21.48	2.82	38.05	3.12	2.69	9.09	0	0	0	0.04	1.56	0	0.38
Mud	0.52	11.23	3.77	39.52	0.09	0	0.15	0.02	0	0.06	0	0	0.79	2.34	0	0.2
Sand	0.46	0	0.17	0.23	66.78	0	0.13	0	0	0	12.09	0.01	0.03	0.84	0	0.1
Water	0.25	0.88	2.6	2.88	1	54.96	0.72	0	0	0	0	0	0.02	0.44	0	0.06
Grass	0.12	0.03	0.09	0.67	0.17	0	38.73	0.93	0	0.78	0	0.02	1.93	6.75	0	0.18
Snow	0.03	0.3	0.37	0.01	0	0	0.22	59.83	0	0	0	0	0.01	0.31	0	0.28
Gravel_*N*	0	0.34	1.43	0.12	0.79	0	0	0	83.86	0	0	0	0.03	0.35	0	0.04
Mud_*N*	0.01	2.99	1.68	3.05	0.44	0	1.39	0.09	1.95	99.16	5.39	0.01	2.82	7.54	0	0.16
Sand_*N*	0	0	0	0	3.05	0	0.99	0	0	0	62.67	0	0.08	1.71	0	0.08
Sky_N	0	0	0	0	0	0	0	0	0	0	0	84.55	0.12	0	0	2.04
Vegetation	0	0	0	0.46	0	0	5.14	1.12	0	0	1.4	0.06	74.99	1.91	0	1.11
Grass_*N*	0.18	0.84	1.8	0.35	1.5	0	20.93	0.24	0	0	5.29	0	2.98	63.14	0	0.58
Snow_*N*	0.01	0.52	0.14	0	0	0	0.04	12.8	0	0	0	0.16	0.09	0.1	0	0.9
Background	3.79	10.16	11.87	6.64	8.93	3	22.28	20.85	3.07	0	13.16	15.18	16.01	11.02	0	92.93

Table 6.2: Precision table for M1, 16 classes, 1364 images.

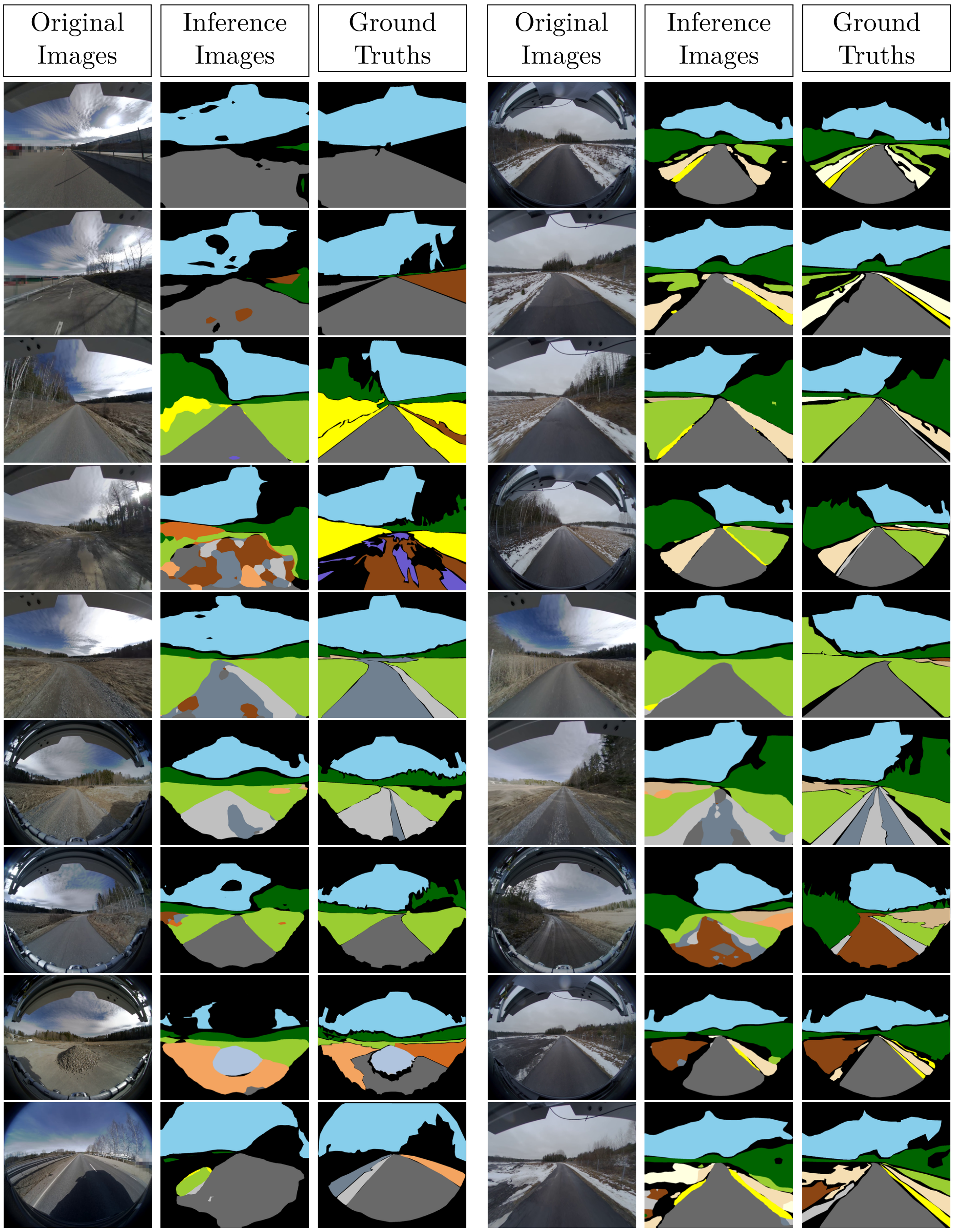
CLASSES	Asphalt	Gravel_H	Gravel_L	Mud	Sand	Water	Grass	Snow	Gravel_N	Mud_N	Sand_N	Sky_N	Vegetation_N	Grass_N	Snow_N	Background
Asphalt	89.54	3.44	11.98	7.91	15.48	10.45	8.13	0.46	0	0.06	0.1	0.03	0.23	2.34	0	0.57
Gravel_*H*	1.04	50.46	10.5	11.21	3.31	1.65	0.31	0.05	0	1.79	0.16	0	0.04	1.31	0	0.35
Gravel_*L*	1.04	15.31	57.09	6.54	1.52	4.06	3.17	0.86	6.46	0.4	0.25	0	0.02	0.87	0	0.28
Mud	1.56	21.16	5.56	56.43	2	1.77	0.07	0.1	2.48	8.83	0	0	0.83	1.57	0	0.46
Sand	1.84	0.39	0.59	0.42	62.76	0.03	0.17	0	0.29	0.4	9.74	0	0	0.61	0	0.1
Water	0.35	0.93	1.19	3.55	2.61	71.53	0.06	0.54	0	1.21	0.63	0	0.03	0.09	0	0.12
Grass	0.13	0.08	0.04	0.88	0.61	0	50.55	0.52	0	8.05	2.61	0	1.11	11.25	0	0.29
Snow	0.03	0.08	0.22	0	0.14	0.73	0.12	48.9	0	0	0.2	0.05	0.03	0.09	0	0.23
Gravel_*N*	0	0.06	0.69	0.27	0.34	0	0.34	0	80.29	2.98	0	0	0.02	0.22	0	0.01
Mud_*N*	0.02	1.29	0.99	2.04	0.41	0.36	0.02	0.01	0.22	57.3	2.45	0	1.16	3.19	0	0.08
Sand_*N*	0.01	0	0	0	2.03	0	0.17	0	0	0.31	38.69	0	0.11	0.68	0	0.05
Sky_N	0	0	0	0	0	0	0	0	0	0.02	0	85.84	0.09	0	0	2.42
Vegetation	0	0.02	0	0.22	0	0.48	3.38	0.17	0	3.71	1.47	0.03	79.19	2.34	0	1.66
Grass_*N*	0.17	0.68	1.56	0.44	0.74	0	9.4	0.83	1.38	6.39	10.54	0	2.89	59.95	0	0.6
Snow_*N*	0.02	0.09	0.03	0	0	0	1.01	25.96	0	0	10.57	0.09	0.06	0.63	0	0.72
Background	4.26	6.01	9.57	10.07	8.06	8.94	23.08	21.61	8.87	8.53	22.58	13.95	14.2	14.85	0	92.05

Table 6.3: Precision table for M1, 16 classes, 6820 images.

CLASSES	Asphalt	Gravel_H	Gravel_L	Mud	Sand	Water	Grass	Snow	Gravel_N	Mud_N	Sand_N	Sky_N	Vegetation_N	Grass_N	Snow_N	Background
Asphalt	81.28	4.46	3.66	11.52	11.91	5.87	9.41	1.23	0.19	1.06	0	0	0.18	0.77	0.02	2.21
Gravel_*H*	2.9	47.61	10.96	14.9	5.66	2.3	2.15	0.07	0	2.65	0	0	0.19	1.78	0.96	1.37
Gravel_*L*	3.64	12.78	68.31	13.82	3.99	5.17	8.35	3.21	14.69	10.68	0.26	0	0.51	2.25	1.05	2.73
Mud	5.89	21.24	4.39	45.93	0.53	1.9	3.87	0.07	3.74	12.1	0.05	0	1.04	1.87	0.27	1.06
Sand	2.23	0.43	0.48	0.43	66.05	0	0.42	0	3.56	0.16	20.14	0.01	0.08	0.87	0.06	0.17
Water	0.42	2.52	1.73	2.42	0.88	72.68	0	0.36	0.08	0.97	0	0	0.02	0.17	1.84	0.5
Grass	0.09	0.01	0.12	0.42	0.14	0	40.53	1.02	0	6.81	0.9	0	1.19	6.5	0.37	0.56
Snow	0.02	0.14	0.25	0.08	0	2.18	0.12	52.24	0	0.15	0	0	0.07	0.4	15.42	0.74
Gravel_*N*	0	0	0.06	0	0.35	0	0	0	66.3	0.04	0	0	0.03	0.18	0	0.04
Mud_*N*	0.12	0.98	1.92	2.12	0.49	0.12	0.63	0.02	2.15	47.48	1.38	0	0.71	5.19	0.05	0.22
Sand_*N*	0	0	0	0	0.85	0	0.03	0	0	0	43.88	0.01	0.06	0.47	0	0.07
Sky_N	0	0	0	0	0	0	0	0	0	0	0.1	82.43	0.64	0	0	3.12
Vegetation	0.06	0.2	0	0.13	0	0	0.97	0.75	0	1.15	1.39	0.81	72.38	3.47	0.07	2.4
Grass_*N*	0.15	0.92	0.59	0.91	2.2	0.04	9.64	0.34	0.44	3.23	15.9	0.01	3.58	60.89	1.01	1.03
Snow_*N*	0.01	0.13	0.13	0.03	0	0.06	0.36	15.27	0	0	0.61	0.01	0.22	0.45	50.05	1.42
Background	3.18	8.57	7.39	7.29	6.95	9.68	23.53	25.42	8.87	13.53	15.38	16.72	19.1	14.73	28.83	82.36

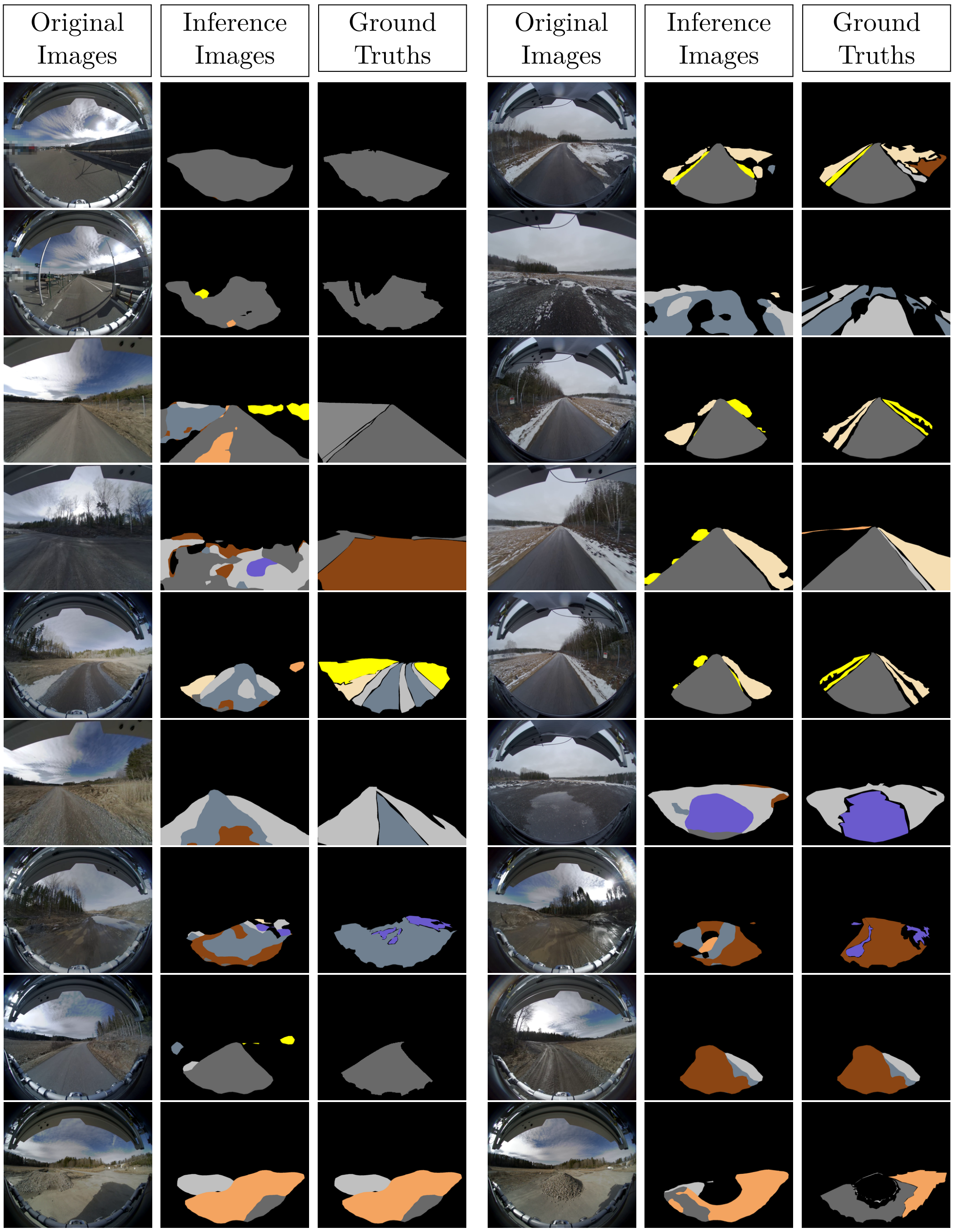
Table 6.4: Precision table for M1, 16 classes, 9548 images.

CLASSES	Asphalt	Gravel_H	Gravel_L	Mud	Sand	Water	Grass	Snow	Gravel_N	Mud_N	Sand_N	Sky_N	Vegetation_N	Grass_N	Snow_N	Background
Asphalt	81.27	0.23	0.98	5.01	14.36	1.41	2.22	1.36	0.38	0	0	0	0.13	0.49	0.04	1.58
Gravel_*H*	5.74	57.34	17.87	10.88	4.59	1.4	0.13	0.11	0	0.19	0.08	0	0.18	0.78	1.75	0.59
Gravel_*L*	2.63	12.08	60.06	7.97	4.17	2.3	3.67	1.7	5.44	2.18	0.28	0	0.16	1.89	1.19	0.99
Mud	4.38	13.7	8.1	64.38	2.57	0.31	0.15	0.48	2.08	12.66	0.12	0	2	1.69	0.25	0.55
Sand	1.23	0.53	0.13	0.02	61.48	0	1.15	0	1.84	0.29	24.95	0.01	0.08	1.25	0	0.12
Water	0.5	1.7	1.61	1.34	1.56	82.08	0	0.18	0	1.46	0	0	0.02	0.11	0.44	0.07
Grass	0.11	0.08	0.15	0.1	0.35	0	42.29	0.96	0	15.74	3.27	0	0.84	6.31	0.84	0.52
Snow	0.03	0	0.17	0.09	0	0.46	0.17	52.55	0	0.44	0	0	0.03	0.24	18.43	0.63
Gravel_*N*	0	0	0.37	0	0.7	0	0	0	76.77	0	0	0	0.01	0.04	0	0.04
Mud_*N*	0.02	0.49	0.34	2.39	0.22	0.45	0.12	0.05	3.62	52.31	1.6	0	0.8	2.29	0.09	0.13
Sand_*N*	0	0.01	0	0	0.53	0	0.24	0	0	0	36.79	0	0.08	1.06	0.16	0.06
Sky_N	0	0	0	0	0	0	0	0	0	0.03	0.27	86.95	0.79	0	0.01	7.88
Vegetation	0.01	0.24	0.01	0.2	0	0	1.31	0.54	0.02	1.52	4.42	0.28	72.77	2.31	0.55	2.58
Grass_*N*	0.17	1.84	0.92	0.66	1.79	0	25.4	0.44	3.02	6.59	8.87	0.01	3.8	66.4	1.81	1.18
Snow_*N*	0.01	0.17	0.12	0.1	0	0	0.28	14.97	0	0.02	0.56	0.01	0.17	0.35	39.82	0.74
Background	3.9	11.57	9.16	6.82	7.69	11.57	22.87	26.64	6.83	6.57	18.8	12.73	18.16	14.78	34.62	82.31



■	Asphalt	■	Sand	■	Gravel_N	■	Vegetation_N
■	Gravel_H	■	Water	■	Mud_N	■	Grass_N
■	Gravel_L	■	Grass	■	Sand_N	■	Snow_N
■	Mud	■	Snow	■	Sky_N	■	Background
Drivable Classes				Non - Drivable Classes			

Figure 6.6: Inferences for 16 Class Distribution.



▨	Asphalt	🟧	Sand	▨	Gravel_N	🟧	Vegetation_N
▨	Gravel_H	🟪	Water	🟧	Mud_N	🟩	Grass_N
▨	Gravel_L	🟨	Grass	▨	Sand_N	□	Snow_N
🟫	Mud	▨	Snow	🟦	Sky_N	⬛	Background
Drivable Classes				Non - Drivable Classes			

Figure 6.7: Inferences for 9 Class Distribution.

is hard to predict. But on the other hand, it would be a disastrous situation if a non-drivable road is marked as drivable. It might end up with collision or damage to the vehicle depending on the surface or object. A lack of this prime information can be vulnerable to severe consequences. While the misclassifications of non-drivable road detected as drivable is less than 2.65%, the background being detected as drivable is as high as 8.05%. Both these conditions are desired to be zero, as it needs a zero tolerance to misclassification among these classes.

On a broader note, taking into consideration these misclassifications and the true positives, the result gets better with more images. Thus, Set 1 performs worse compared to Set 2 and both of these perform worse than Set 3 and 4. There has not been a significant difference between Set 3 and 4 but the gradual productivity with increasing images promises a better result for more images.

6.2.6 Effect of Gabor

The Gabor filter picks up the texture features. We trained the Model 2 only on the images retrieved from the Gabor response. From the Tables E.3 and E.4, we note that the misclassifications which occurred with the original images have further worsened exhibiting poorer performance for these images.

While it is interesting to note that the non-drivable gravel, i.e. the gravel pile, was seen to have high precision, the class failed to score high on the F-score. This behavior may be seen as a result of failure to recall coarse nature of the gravel pile and the contribution of the class to be less than 1% as per Figure 5.5a.

We can also note that the addition of undistorted images in Set 2 does not provide enough improvement or deterioration in the result. We also observe that a majority of the grass is seen to be classified as non-drivable mud. The true positive for non-drivable mud was over 99% for original images. The similar texture features of the mud and the grass can be challenging to obtain, and thus the overall performance does not seem to be effective in this case.

Overall, we note that the texture features fail to perform well by themselves as they overlap the colour features with that of the texture. The filter might perform well on the classifiers but deep learning algorithm fails to incorporate these features.

6.2.7 Effect of Pre-trained Networks

Due to the small size of the dataset, it becomes very important to have the weights defined based on a pre-trained network. The random allocation of weights for a non-trained network resulted in a settling of the back propagation to about 63 - 65% accuracy. This occurs as the system updates the weights based on the first propagation and these tend to overfit the training data, resulting in a saturated value with the first iteration of the training. On the provision of the pre-trained weights, the model has been trained on some other data. For example, the Alexnet

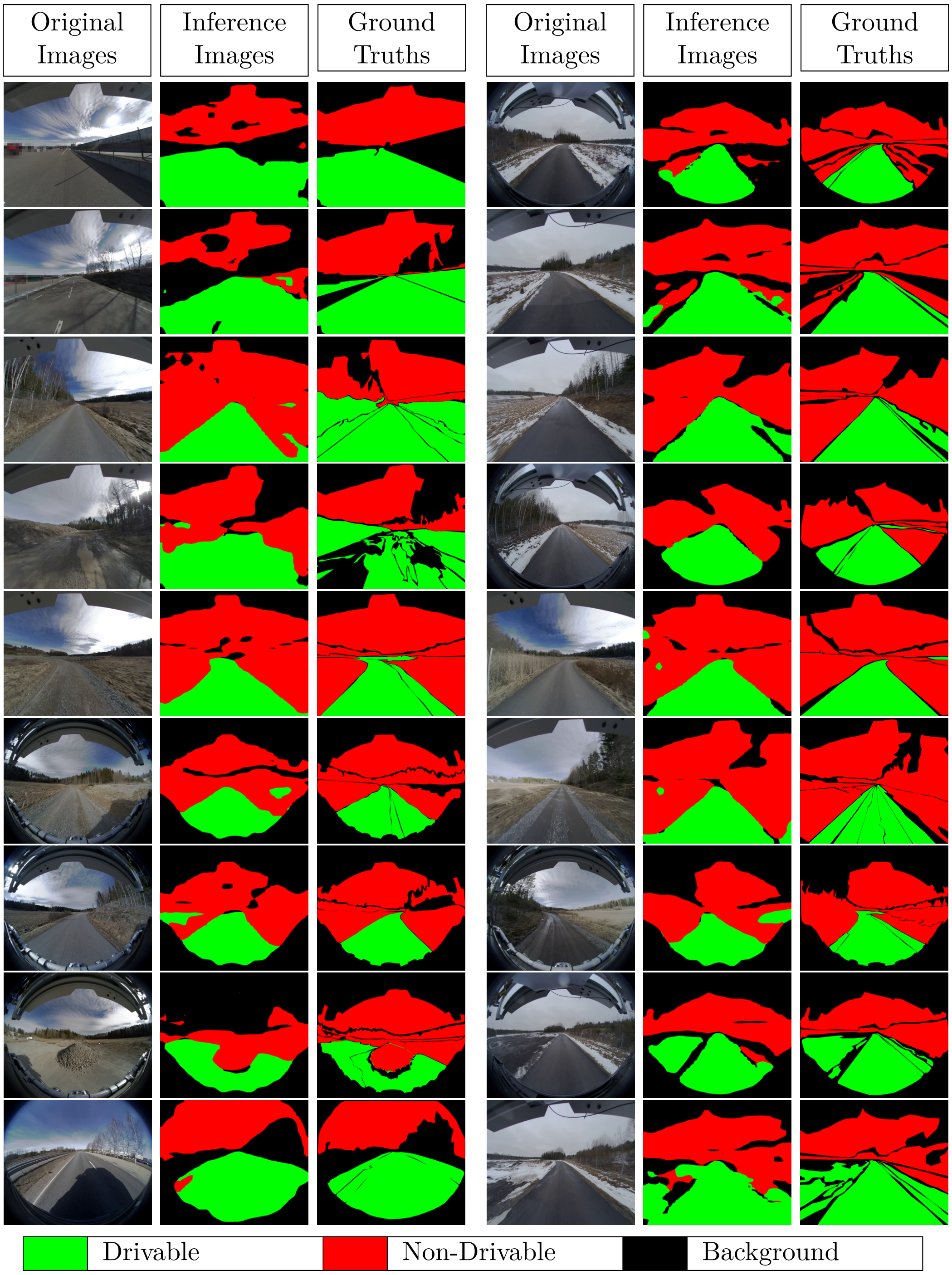


Figure 6.8: Inferences for 3 Class Distribution.

caffemodel uses the weights from the PASCAL VOC Dataset and utilizes the initial layers to be overlapped with the data from the dataset, at the final layer. The addition of the pre-trained weights can be easily noted with the precision jump to over 90%. The features like the edges, corners, colour, and texture are already acquired from a larger dataset and the class specific information is learned at the final step.

In the Figure 6.4, a comparison between the models was presented based on the mean IoU. Models 1 and 2 are trained on AlexNet whereas Model 3 has been trained over Model 1. It can be seen here that Model 2 performs worse than Model 1 indicating that the Gabor images lose some prime information by picking only the texture information. However, there is a significant improvement of Model 3 with respect to Model 1. This can be considered as one of the prime benefits of a system trained on various parameters.

6.2.8 Effect of Individual Classes

The F-Scores and the IoU scores for the individual classes for all of the models are tabulated in Appendix E.

Observing the results from the F-Scores, we determine some interesting facts, such as the drivable grass has performed poorly for all the models. We already noted the poor response of Model 2, but here we can indicate the classes which correspond to such results. Thus, we note that drivable mud and drivable water are a constant source of error for the models of 9 as well as 16 class. This is an effect of the lower amounts of pixels in the images for these classes. We also see a significantly poor response of the high and low-density gravel for 9 classes. However, the drivable grass performed the worst among all classes and for all the models. This is again a result of the human understanding of drivable regions of grass.

On the other hand, we can see that asphalt and sky classes performed very well with vegetation class nearing the same. Not only we had a wide distribution of these classes among all, but the distinguishing colour of these classes is hard to blend. Even though background seems to reflect a very good response for the 9 class, it does not provide enough information about what the region corresponds to.

6.2.9 Demo Video

A demonstration of the 9 and 16 class inference for Model 1 can be seen at:
goo.gl/RARWFi.

The video demonstrates four different road conditions, namely, highway, asphalt, gravel, and muddy roads with the responses for 9 and 16 classes. The video is a compilation of frame by frame inference of the video frames being run at 5 fps for 500 frames. The video gives us some compelling results as given in the Table 6.5.

While the roads are being categorized well, the determination of the water ponds on a muddy road is a huge challenge with the absence of enough data on an absolute water pond. The colour of water ponds is dependent on various factors like the

Table 6.5: Notable results from the Demo video.

Duration	Details
00:08 - 00:13	• The highway and asphalt roads are classified properly with the patches of gravel being distinguished clearly. • The gravel and muddy roads judge the road well for 9 classes but increase in the number of classes is a clutter for 16 classes. • It can also be seen that the muddy road contains a lot of non-drivable classes on the road implying high processing demand at such places.
00:17 - 00:23	• The road type remains almost similar for this duration for the highway, asphalt and gravel roads, and there are not many jumps between classes implying it is recommended to use these models on such roads. • The muddy road contains water ponds and there are negligible surfaces which are suitable to drive as seen in the 16 class distribution.
00:53 - 00:58	• The gravel road tends to fail partially under these conditions with certain drivable regions categorized as background. The background sections can either be drivable or non-drivable and thus may pose as risky situations. • This region sees an improvement in the muddy regions implying the presence of training images available for this neighborhood.
01:18 - 01:23	• Highway road detects some drivable mud and partial water ponds in the place of shadows. This occurs as there was no training data for asphalt roads containing shadows. • Simultaneously, gravel road has been trained for shadows and does not pose a problem for the same situation. • It can be noticed that the asphalt, gravel, and muddy roads perform well in this duration with a good detection of the water ponds due to the availability of training data for these conditions.

material underneath, the refraction due to the density of the liquid and reflection of light due to the sky. There is no deterministic approach to find these water ponds but the texture along with the depth data is expected to provide a smooth surface for water bodies compared to an uneven muddy surface.

However, the roads are being reciprocated well even with some incorrect class inference. Under conditions like asphalt or gravel roads, a border between drivable and non-drivable regions is segregated well. This implies that the vehicle may still be driven autonomously by categorizing the type of road. However, it needs to be noted that the jumps between classes are puzzling and do not happen under most scenarios. On the other hand, the muddy regions and water bodies are considerably hard to drive and unless a considerable amount of training data is provided, it is good to hand over to a manual drive mode.

6.2.10 Test on Google Images

We also evaluate the models on several dash cam pictures obtained from Google. The results of the Model 3 and Set 4 tests are shown in Figure 6.9. We see few good classification results as well as numerous failures. We have ordered the images with top ones depicting good results and the lower ones with more absurd responses.

In the first two pictures, the gravel and asphalt roads are both classified as asphalt, but we can easily notice the segregation between drivable regions and non-drivable regions. In the next two images, the mud and the gravel roads seem to respond with a number of classes. The amount of data is hard to be analyzed under these situations but we can see that the presence of water content and mud is being distinguished. It may appear that the exact boundaries are unclear but driving under such conditions will always be troublesome. It is also important to note that this image with the muddy region is similar to the ones provided during the training phase. Thus, we are able to establish a connection for the different classes in this result. On the other hand, the gravel road seems to respond well with nearer regions but it is also surprising to observe parts of the sky in between the road. The effect of illumination due to the environment results in such clusters.

As we notice for the other images, there is hardly any accuracy of predictions as the classifier fails terribly. The sky is being categorized well but the off-road terrains with the drivable regions appear mostly as sand. We can see from images in Row 4 and 5 that the muddy road in these images is inconsistent to the muddy regions in our training set. Thus, the same region being classified as sand, non-drivable grass and gravel is confusing. Similarly, Row 6 and 7 have roads easily distinguishable by the human eye but the lack of training images produces unexpected results. Thus, it can easily be seen that the model fails to produce reliable results in most cases. Establishing a proper boundary between the drivable and non-drivable region is a must to determine better drivable conditions.

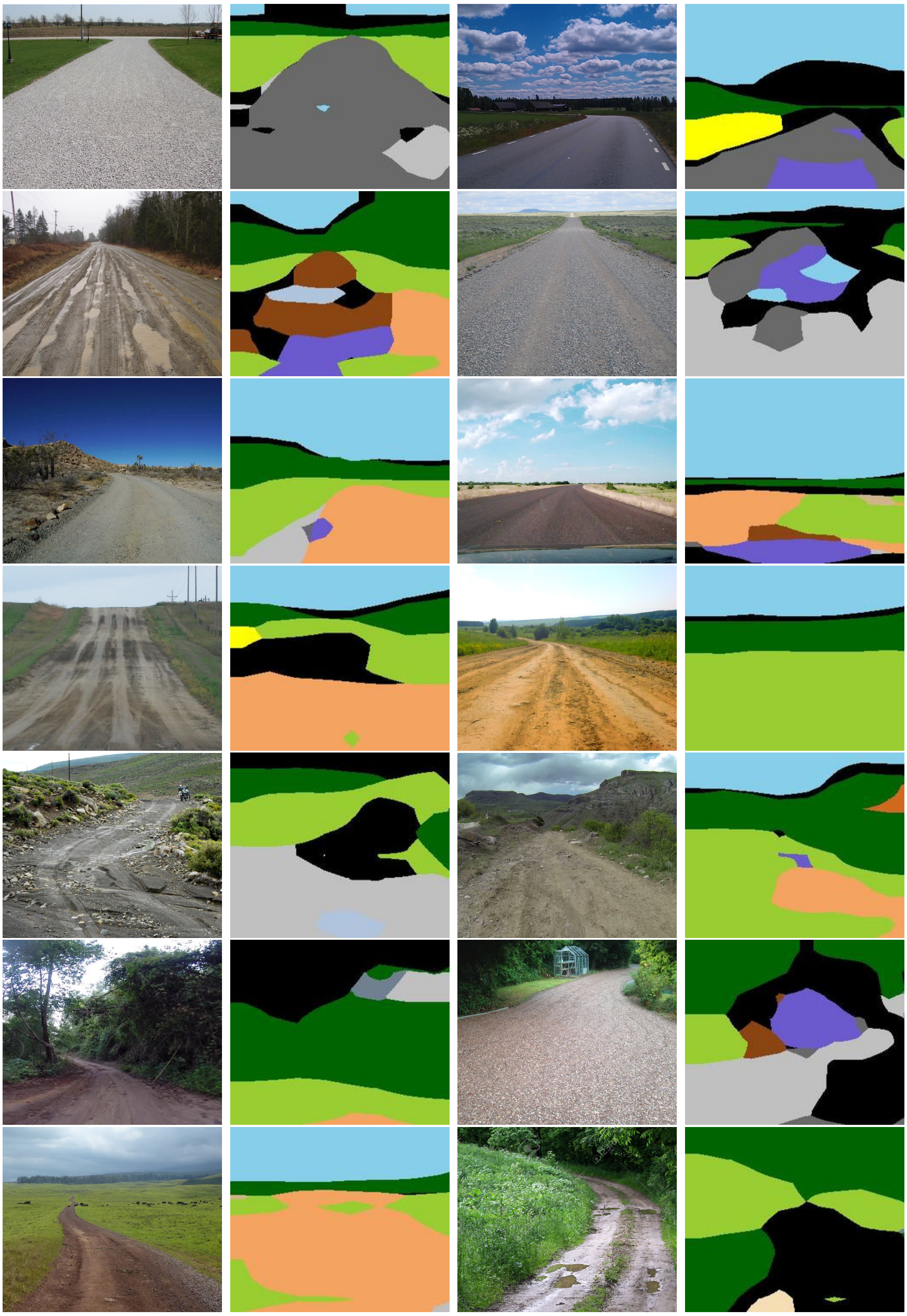


Figure 6.9: Deep learning Inference for Google images for 16 Class Distribution. (Columns 1 and 3) Original images, (Columns 2 and 4) Corresponding inferences.

6.3 Discussion

While a number of conclusions may be made based on the comparison of the above results, a few of the prominent inferences are mentioned here.

Gabor features work quite well for different patterns. The discrete use of the Gabor filter is best for picking out the texture information and thus, the same can be used to improve the results of the neural network. The detection of the texture features vary gradually for each of the classes and these kind of patterns are easy to determine with the use of Gabor filters.

We note that adding both the distorted and the undistorted images resulted in a diverse dataset which proved to be successful for all the class distributions. This can be seen as an effective alternative to having camera specific results as the rectified images can be standardized for data from any camera. The addition of the rectified images in the training dataset also seems to assist by avoiding the loss of pixel to pixel information during the post-rectification process.

The splitting of the datasets for training, validation and test phases are important for neural network analysis. The results depict that they are not affected by the class distribution and all classes perform equally well. This clearly implies that under these evaluation conditions, the irregularity in the class distribution does not affect the performance of deep learning.

Overall, it can be seen that the 16 classes do not generate a lot of misclassification results even with the presence of so many classes. Checking out the results from the precision tables, we notice that the last row which represents the background pixels is often inferred as drivable regions, which may not be acceptable under most conditions. On the other hand, the results are promising for the non-drivable classes as they are comparatively less confused with the drivable classes. It can be noted that the true classification rates for 3 classes are the highest compared to the 16 and 9 class results. The predominant reason for this being the less number of classes and less chances of errors. The figures state that the misclassifications mostly occur with the confusion between the same classes of drivable and non-drivable surfaces. This kind of misunderstanding occur for humans as well and there is no fixed rule to justify classes like grass or snow as non-drivable. We thus place them on the lower priority level to allow some understanding of the drivability.

Simultaneously, considering the 9 class distribution, we notice worse results than the 16 class distribution, especially when we see the misclassifications of the background regions. Clearly, this class distribution misses out on important information about the surroundings which are been marked as background during training. The uncertainty to categorize the background class from the drivable classes produces an unreliable result. However, seeing the overall comparison in Figure 6.4, we see only a marginal difference between the two class distributions.

We also note that the addition of the undistorted images does not improve the effect of the Gabor filters. Thus, it also seems to be justified that even when the pixel density reduces at the edges, the Gabor effect retains the corresponding information.

Another development seen by the use of pre-trained networks depict that the models perform better with the use of transfer learning. It can be noted that the neural network performs better by first training the images on the desired images and then retraining them over the relevant features. Effectively, the weights have already been generated to form the model and these weights further improve by propagating with the desired features.

We see a poor response for all the classes in terms of the IoU scores. While this seems to be a challenge, it is quite understandable that the IoU scores which compare the predictions to the manual annotations may not give accurate results. The manual annotations being hand-drawn are not expected to be perfect and rely on the human eye. While certain classes like asphalt, sky, vegetation, non-drivable grass and non-drivable snow seem to perform well, it is a pixel to pixel comparison and needs to be justified on human understanding compared to numerical values.

The effect of a properly augmented class distribution ensures that the classes are not missed during training or validation phases. A similar test dataset would ensure that the accuracies obtained from the system produce a dependable result.

Chapter 7

Summary

The absence of technologies to determine the type of road remains a challenge for the autonomous vehicles to drive at off-road terrains. This thesis provides a study to overcome this problem using deep learning technique. It also provides a comparison of a choice of different models and classes to improve the results of this classification problem.

7.1 Conclusion

In this thesis, we have proposed a method to determine the road surface types for unstructured environments using visual information. We classify the terrain types into drivable and non-drivable classes using only a small dataset of 100 labeled images. The experimental results demonstrate that pre-trained networks assist the system to learn the significant high-level features. Annotated masks with varying environmental conditions provide a reliable training set to be able to generate inferences with high accuracy. Further on, we have also been able to justify that the combination of Gabor filters with the trained networks improves the results.

We can also state from the results that the small dataset is unreliable for all real-life scenarios, indicating the extensive need for a larger dataset. From the discussion in Section 6.3, we may note that the choice of classes is important as it decides the admissible amounts of misclassification. The lesser the number of classes, the more is the threat to the background class, reflecting unreliable results for drivability at such regions. While the autonomous vehicles are yet to establish a high confidence meter for off-road driving, the results are promising that neural networks with a combination of other feature determiners, can establish a dependable navigation path under such conditions.

7.2 Future Work

While the current technologies are able to determine the road regions in urban environments, we believe that this work has provided sound results to apply deep learning algorithms to determine the type of the road surface.

The amount of data is one of the main limitations of this thesis. The hand annotation of the images consume a lot of manual labour and effort. It is also important to note that the difference in the distribution of classes causes certain small classes to be neglected. An unbiased and large training data should be maintained with all types of road surfaces. Another aspect while annotating the images is the difference in the human understanding of surface type when provided with an abstract image. However, this challenge of data collection may be compensated. On most occasions, a road surface type does not change often, thus videos may be gathered from such road scenarios and multiple frames with similar image clusters may be annotated in the same manner. However, manual annotation still remains a big challenge in the deep learning field and various online forums, like Amazon Mechanical Turk [73], are providing a platform for the general public to help in these processes.

In the analysis section, a set of fixed parameters was used for the Gabor filters in this work, so it might also be advantageous to use different parameter values based on the environment. The sample surface from the test terrain may help determine the best parameters. Moreover, the tests were conducted only with the AlexNet architecture, so it is essential to compare with the other architectures to establish the preferred choice of network. Once the surfaces are classified, algorithms can be formulated to prioritize driving on more confident terrains like gravel or sand, and avoid muddy or water regions. Further modification and comparisons need to be performed to evaluate the optimum performances which were bounded by the scope of this thesis.

The work also provides a possibility to use Recurrent Neural Networks (RNNs) as mentioned in the Section 3.3.10, which can be used to analyze the previous frames and suggest an inference based on the prior evaluated data. Since the chances of a road type changing drivable to non-drivable are generally uncommon, the past data can be overlapped with the trained data to exhibit better inferences. Also, it is always recommended to have a fast evaluation process to reduce the latency between real-time image capture and the segmented information. Deep neural networks are the state of the art systems in semantic segmentation, and modifying the networks to obtain its full potential may not only reduce their latency times but also improve the results significantly.

Bibliography

[1] S. O.-R. A. V. S. Committee *et al.*, "Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems," 2014.

[2] A. Finn and S. Scheding, *Developments and challenges for autonomous unmanned vehicles.* Springer, 2012.

[3] R. Oliveira, *Planning and Motion Control in Autonomous Heavy-Duty Vehicles.* PhD thesis, Master's thesis, KTH Royal Institute of Technology, 2014.

[4] P. F. Lima, M. Trincavelli, J. Mårtensson, and B. Wahlberg, "Clothoid-based model predictive control for autonomous driving," in *Control Conference (ECC), 2015 European*, pp. 2983–2990, IEEE, 2015.

[5] I. Halatci, C. A. Brooks, and K. Iagnemma, "Terrain classification and classifier fusion for planetary exploration rovers," in *Aerospace Conference, 2007 IEEE*, pp. 1–11, IEEE, 2007.

[6] L. Ojeda, J. Borenstein, G. Witus, and R. Karlsen, "Terrain characterization and classification with a mobile robot," *Journal of Field Robotics*, vol. 23, no. 2, pp. 103–122, 2006.

[7] A. Howard and H. Seraji, "Vision-based terrain characterization and traversability assessment," *Journal of Field Robotics*, vol. 18, no. 10, pp. 577–587, 2001.

[8] Y. N. Khan, P. Komma, and A. Zell, "High resolution visual terrain classification for outdoor robots," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pp. 1014–1021, IEEE, 2011.

[9] Y. Zou, W. Chen, L. Xie, and X. Wu, "Comparison of different approaches to visual terrain classification for outdoor mobile robots," *Pattern Recognition Letters*, vol. 38, pp. 54–62, 2014.

[10] C. J. Taylor, J. Malik, and J. Weber, "A real-time approach to stereopsis and lane-finding," in *Intelligent Vehicles Symposium, 1996., Proceedings of the 1996 IEEE*, pp. 207–212, IEEE, 1996.

[11] M. Bertozzi, A. Broggi, and A. Fascioli, "Vision-based intelligent vehicles: State of the art and perspectives," *Robotics and Autonomous systems*, vol. 32, no. 1, pp. 1–16, 2000.

[12] J. Goldbeck, B. Hürtgen, S. Ernst, and L. Kelch, "Lane following combining vision and dgps," *Image and Vision Computing*, vol. 18, no. 5, pp. 425–433, 2000.

[13] M. Foedisch and A. Takeuchi, "Adaptive real-time road detection using neural networks," in *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*, pp. 167–172, IEEE, 2004.

[14] C. Rasmussen, "Combining laser range, color, and texture cues for autonomous road following," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 4, pp. 4320–4325, IEEE, 2002.

[15] L. Lu, C. Ordonez, E. G. Collins, and E. M. DuPont, "Terrain surface classification for autonomous ground vehicles using a 2d laser stripe-based structured light sensor," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 2174–2181, IEEE, 2009.

[16] D. Langer, J. Rosenblatt, and M. Hebert, "A behavior-based system for off-road navigation," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 6, pp. 776–783, 1994.

[17] R. Simmons, E. Krotkov, L. Chrisman, F. Cozman, R. Goodwin, M. Hebert, L. Katragadda, S. Koenig, G. Krishnaswamy, Y. Shinoda, *et al.*, "Experience with rover navigation for lunar-like terrains," in *Intelligent Robots and Systems 95.'Human Robot Interaction and Cooperative Robots', Proceedings. 1995 IEEE/RSJ International Conference on*, vol. 1, pp. 441–446, IEEE, 1995.

[18] J.-F. Lalonde, N. Vandapel, D. F. Huber, and M. Hebert, "Natural terrain classification using three-dimensional ladar data for ground robot mobility," *Journal of field robotics*, vol. 23, no. 10, pp. 839–861, 2006.

[19] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[20] S. Bittel, V. Kaiser, M. Teichmann, and M. Thoma, "Pixel-wise segmentation of street with neural networks," *arXiv preprint arXiv:1511.00513*, 2015.

[21] I. Halatci, C. A. Brooks, and K. Iagnemma, "A study of visual and tactile terrain classification and classifier fusion for planetary exploration rovers," *Robotica*, vol. 26, no. 06, pp. 767–779, 2008.

[22] B. Wang and V. Frémont, "Fast road detection from color images," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pp. 1209–1214, IEEE, 2013.

[23] P. Vernaza, B. Taskar, and D. D. Lee, "Online, self-supervised terrain classification via discriminatively trained submodular markov random fields," in *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pp. 2750–2757, IEEE, 2008.

[24] M. Häselich, M. Arends, N. Wojke, F. Neuhaus, and D. Paulus, "Probabilistic terrain classification in unstructured environments," *Robotics and Autonomous Systems*, vol. 61, no. 10, pp. 1051–1059, 2013.

[25] N. Soquet, D. Aubert, and N. Hautiere, "Road segmentation supervised by an extended v-disparity algorithm for autonomous navigation," in *Intelligent Vehicles Symposium, 2007 IEEE*, pp. 160–165, IEEE, 2007.

[26] M. Agrawal, K. Konolige, and R. C. Bolles, "Localization and mapping for autonomous navigation in outdoor terrains: A stereo vision approach," in *Applications of Computer Vision, 2007. WACV'07. IEEE Workshop on*, pp. 7–7, IEEE, 2007.

[27] A. Kelly, A. Stentz, O. Amidi, M. Bode, D. Bradley, A. Diaz-Calderon, M. Happold, H. Herman, R. Mandelbaum, T. Pilarski, *et al.*, "Toward reliable off road autonomous vehicles operating in challenging environments," *The International Journal of Robotics Research*, vol. 25, no. 5-6, pp. 449–483, 2006.

[28] I. Tang and T. P. Breckon, "Automatic road environment classification," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 476–484, 2011.

[29] Y. Deng and B. Manjunath, "Unsupervised segmentation of color-texture regions in images and video," *IEEE transactions on pattern analysis and machine intelligence*, vol. 23, no. 8, pp. 800–810, 2001.

[30] J. Liu and Y.-H. Yang, "Multiresolution color image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 7, pp. 689–700, 1994.

[31] C.-L. Huang, T.-Y. Cheng, and C.-C. Chen, "Color images' segmentation using scale space filter and markov random field," *Pattern Recognition*, vol. 25, no. 10, pp. 1217–1229, 1992.

[32] M. Mirmehdi and M. Petrou, "Segmentation of color textures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 2, pp. 142–159, 2000.

[33] A. Angelova, L. Matthies, D. Helmick, and P. Perona, "Fast terrain classification using variable-length representation for autonomous navigation," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pp. 1–8, IEEE, 2007.

[34] G.-Y. Sung, D.-M. Kwak, and J. Lyou, "Neural network based terrain classification using wavelet features," *Journal of Intelligent & Robotic Systems*, vol. 59, no. 3, pp. 269–281, 2010.

[35] Z. Kato and T.-C. Pong, "A markov random field image segmentation model for color textured images," *Image and Vision Computing*, vol. 24, no. 10, pp. 1103–1114, 2006.

[36] M. Idrissa and M. Acheroy, "Texture classification using gabor filters," *Pattern Recognition Letters*, vol. 23, no. 9, pp. 1095–1102, 2002.

[37] I. Fogel and D. Sagi, "Gabor filters as texture discriminator," *Biological cybernetics*, vol. 61, no. 2, pp. 103–113, 1989.

[38] S. Graovac and A. Goma, "Detection of road image borders based on texture classification," *International Journal of Advanced Robotic Systems*, vol. 9, no. 6, p. 242, 2012.

[39] Z. Haliche and K. Hammouche, "The gray level aura matrices for textured image segmentation," *Analog Integrated Circuits and Signal Processing*, vol. 69, no. 1, pp. 29–38, 2011.

[40] S. Yun, Z. Guo-Ying, and Y. Yong, "A road detection algorithm by boosting using feature combination," in *Intelligent Vehicles Symposium, 2007 IEEE*, pp. 364–368, IEEE, 2007.

[41] Y. N. Khan, P. Komma, K. Bohlmann, and A. Zell, "Grid-based visual terrain classification for outdoor robots using local features," in *Computational Intelligence in Vehicles and Transportation Systems (CIVTS), 2011 IEEE Symposium on*, pp. 16–22, IEEE, 2011.

[42] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: a new learning scheme of feedforward neural networks," in *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, vol. 2, pp. 985–990, IEEE, 2004.

[43] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 513–529, 2012.

[44] J. J. d. M. S. Junior and A. R. Backes, "Elm based signature for texture classification," *Pattern Recognition*, vol. 51, pp. 395–401, 2016.

[45] F. H. C. Tivive and A. Bouzerdoum, “Texture classification using convolutional neural networks,” in *TENCON 2006. 2006 IEEE Region 10 Conference*, pp. 1–4, IEEE, 2006.

[46] R. Caruana and A. Niculescu-Mizil, “An empirical comparison of supervised learning algorithms,” in *Proceedings of the 23rd international conference on Machine learning*, pp. 161–168, ACM, 2006.

[47] P. Y. Shinzato, V. Grassi, F. S. Osorio, and D. F. Wolf, “Fast visual road recognition and horizon detection using multiple artificial neural networks,” in *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pp. 1090–1095, IEEE, 2012.

[48] C. C. T. Mendes, V. Frémont, and D. F. Wolf, “Exploiting fully convolutional neural networks for fast road detection,” in *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pp. 3174–3179, IEEE, 2016.

[49] A. Bystrov, E. Hoare, T.-Y. Tran, N. Clarke, M. Gashinova, and M. Cherniakov, “Automotive system for remote surface classification,” *Sensors*, vol. 17, no. 4, p. 745, 2017.

[50] M. Thoma, “A survey of semantic segmentation,” *arXiv preprint arXiv:1602.06541*, 2016.

[51] E. Shelhamer, J. Long, and T. Darrell, “Fully convolutional networks for semantic segmentation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 4, pp. 640–651, 2017.

[52] Y. Ohta, T. Kanade, and T. Sakai, “An analysis system for scenes containing objects with substructures,” in *Proceedings of the Fourth International Joint Conference on Pattern Recognitions*, pp. 752–754, January 1978.

[53] W. E. Snyder and H. Qi, *Machine vision.* Cambridge University Press, 2010.

[54] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.

[55] J.-Y. Bouguet, “Matlab calibration tool,” 2015.

[56] C. L. Zitnick and P. Dollár, “Edge boxes: Locating object proposals from edges,” in *European Conference on Computer Vision*, pp. 391–405, Springer, 2014.

[57] M. Caudill, “Neural networks primer, part i,” *AI expert*, vol. 2, no. 12, pp. 46–52, 1987.

[58] C. Woodford, “Neural networks.” `http://www.explainthatstuff.com/introduction-to-neural-networks.html`, 2017. Online; accessed 20 July 2017.

[59] A. Vedaldi and K. Lenc, "Matconvnet: Convolutional neural networks for matlab," in *Proceedings of the 23rd ACM international conference on Multimedia*, pp. 689–692, ACM, 2015.

[60] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognition Letters*, vol. 30, no. 2, pp. 88–97, 2009.

[61] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[62] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ade20k dataset," *arXiv preprint arXiv:1608.05442*, 2016.

[63] L. Bottou, "Stochastic gradient learning in neural networks," *Proceedings of Neuro-Nımes*, vol. 91, no. 8, 1991.

[64] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning.* MIT Press, 2016. `http://www.deeplearningbook.org`.

[65] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," in *Advances in neural information processing systems*, pp. 3320–3328, 2014.

[66] J. G. Daugman, "High confidence visual recognition of persons by a test of statistical independence," *IEEE transactions on pattern analysis and machine intelligence*, vol. 15, no. 11, pp. 1148–1161, 1993.

[67] H. Yao, L. Chuyi, H. Dan, and Y. Weiyu, "Gabor feature based convolutional neural network for object recognition in natural scene," in *Information Science and Control Engineering (ICISCE), 2016 3rd International Conference on*, pp. 386–390, IEEE, 2016.

[68] R. J. Williams and D. Zipser, "Gradient-based learning algorithms for recurrent networks and their computational complexity," *Backpropagation: Theory, architectures, and applications*, vol. 1, pp. 433–486, 1995.

[69] S. Valipour, M. Siam, M. Jagersand, and N. Ray, "Recurrent fully convolutional networks for video segmentation," in *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*, pp. 29–36, IEEE, 2017.

[70] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[71] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, “Labelme: a database and web-based tool for image annotation,” *International journal of computer vision*, vol. 77, no. 1, pp. 157–173, 2008.

[72] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, “Learning and transferring mid-level image representations using convolutional neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1717–1724, 2014.

[73] “Amazon Mechanical Turk.” `https://www.mturk.com/`, 2005. Online; accessed 25 December 2017.

[74] T. M. Crimmins, “Management guidelines for off-highway vehicle recreation,” *National Off-Highway Vehicle Conservation Council*, 2006.

[75] J.-F. Bonnefon, A. Shariff, and I. Rahwan, “The social dilemma of autonomous vehicles,” *Science*, vol. 352, no. 6293, pp. 1573–1576, 2016.

[76] J. J. Thomson, “The trolley problem,” *The Yale Law Journal*, vol. 94, no. 6, pp. 1395–1415, 1985.

[77] Q. Bui, “Map: The Most Common* Job In Every State.” `http://www.npr.org/sections/money/2015/02/05/382664837/map-the-most-common-job-in-every-state`, 2015. Online; accessed 12 August 2017.

[78] N. Kitroeff, “Robots could replace 1.7 million American truckers in the next decade.” `http://www.latimes.com/projects/la-fi-automated-trucks-labor-20160924/`, 2016. Online; accessed 12 August 2017.

[79] B. Bashiri and D. D. Mann, “Automation and the situation awareness of drivers in agricultural semi-autonomous vehicles,” *Biosystems engineering*, vol. 124, pp. 8–15, 2014.

Appendix A

Hardware and Software used

The Appendix A provides a detailed account of the hardware and software setup used to perform the experiments. Sections A.1 and A.2 describe them in detail.

A.1 Experimental Hardware

A.1.1 Drive PX2 Platform

The Drive PX2 is the second generation of NVIDIA's computers aimed at providing an end-to-end deep learning platform for self-driving cars. The technology developed for autonomous vehicles and driver assistance functionality uses deep neural networks to process data from multiple cameras and sensors and is designed to enable Enhanced Autopilot and full self-driving functionality. It relies on 2 next-gen Tegra SOCs and two discrete Pascal GPUs. It features 12 ARM64-based CPU cores and four chips that pack Pascal GPUs, rounding up to 8 TFLOPs of performance.

NVIDIA's open mapping platform is built on the NVIDIA DriveWorks software toolkit and was developed with a focus to accelerate development of autonomous driving functionalities. The system incorporates deep learning algorithms on the camera inputs to detect lanes, signs and other landmarks, and can be used to both create maps and determine changes in the environment. It provides a development platform for all computationally intensive algorithms such as object detection, localization, and path planning.

A.1.2 Omnivision Camera

The Omnivision camera OV10640 is a fish eye lens RGB type camera consisting of AR0231 OnSemi CMOS Image sensor. The ultra high resolution lens has a 190° horizontal field of view and a 100° vertical field of view (FOV). The images are captured at 30 fps at 1280x1080 resolution in RAW format. The camera model targets to meet the automotive qualification. A metal housing with AEC-Q100 Grade 2 qualification allows operations between -40^{o}C to +105^{o}C to use it outdoors during extreme weather conditions. The FAKRA Z type connectors provide the connection to Drive PX2. The cameras are known as GMSL cameras, an acronym for Gigabit Multimedia Serial Link, and are designed to be compatible with the NVIDIA platform.

A.1.3 System Specifications

GPU enabled systems provides accelerated computing and are most suited to perform the neural network's training and tests. The GPU systems offloads the computing intensive applications, fastening the process and reducing the load from the CPU. Here we use different systems for training and testing. While it is optimum to perform the training on fixed computers with high-end GPU, testing requires real time processing in the mobile systems to perform the classification within the autonomous vehicles.

We used two NVIDIA Tesla P100 GPUs with 16 GB RAM for training and transferred the training load to batches to perform simultaneous processing. For testing, we used an Intel Core i7-7820HK CPU @ 2.90 GHz with 32 GB RAM and NVIDIA GeForce GTX 1070 GPU with 8 GB RAM.

A.2 Software Libraries and Technologies

A.2.1 DriveWorks

NVIDIA DriveWorks is a Software Development Kit (SDK) that is developed for automakers and research institutions with a prime focus on detection, localization, planning and visualization algorithms. It consists of sample applications, tools and library modules to accelerate the development on the NVIDIA Drive PX platforms.

A.2.2 LabelMe

The web based online annotation tool provides researchers the ability to label images and share the annotations with the world. It uses a Javascript drawing tool with a possibility to upload images or label the existing images with the objects they desire. The user annotates the objects by clicking on the object boundary and continuing the clicks along the boundary until the starting point is clicked again. The selected object can then be labeled on a pop-up dialog box. The resulting images are stored in XML file format. There is a possibility to rename the annotations or combine them based on the sets of classes as required. The sets of annotated masks or XML files may then be downloaded for personal use [71].

A.2.3 DIGITS

The NVIDIA's Deep Learning GPU Training System (DIGITS) is a platform provided by NVIDIA with a Graphical User Interface (GUI) to administer all types of deep learning tasks like object detection, classification, and recognition. The interactive GUI can be maintained locally or over a server and it supports managing data, designing and training neural networks. The platform performs all the necessary GPU needs and simplifies the programming and debugging processes.

The segmentation algorithm requires a set of trained images. The dataset can be fed to the system by providing the list of feature images and their label images. It allows a possibility to choose the amount of validation images. A colour map specification is also recommended to indicate the class labels for corresponding label colours.

Using the segmentation dataset, models may be generated by using either Caffe or Torch framework. There are certain parameters which can be set for this purpose like the number of training epochs, the intervals between the snapshots and validations, batch size and base learning rate. The solver algorithm may also be provided here. There is a possibility to select the GPU depending on the availability. Finally and most importantly, a training network needs to be defined. A custom network model on Caffe framework requires a prototxt file defining the layers of the network. This can be overlapped with a pre-trained model for a deeper training system. The pre-trained Caffe models are the ones which have already been trained on a certain dataset beforehand and is overlapped with the custom layers that are provided.

The training time depends on various factors like the number of images in the dataset, the epochs, and the learning rate. The system provides a GUI to indicate the training results with the accuracy and loss functions based on the network. The system is considered reliable if it has a high value on the accuracy and a low loss function. Retraining over and over increases the confidence of the network and this can be seen with the models in Appendix C.

DIGITS also provides the facility to test individual images or a set of images with their local paths submitted in a text file. The interface allows the user to visualize the inferences by selecting their choices for their score values or viewing the segmented images.

Appendix B

Ethical Responsibility

The off-road terrain is a wonderful juncture for recreation purposes for hikers, bicyclists, equestrians and off-highway vehicles (OHVs). Driving responsibly on such terrains is a matter of concern for the wildlife, vandalism as well as for self-safety reasons. There is a high dependency of established trails on such areas and relying on these trails is an aid for future travelers. Apart from the safety regulations for the use of public or state lands for OHVs, there needs to be a constant monitoring of seasonal closures for rainy seasons and wildlife breeding areas [74].

According to Bonnefon *et al.* [75], autonomous vehicles cannot avoid all accidents and the challenge to act ethically at such situations will lie in the hands of three groups, viz. the customers owning these vehicles, the manufacturers programming them and the government regulating the manufacturers and the consumers. The platform by MIT (`moralmachine.mit.edu`) which reciprocates the trolley problem [76] to simulate real-life ethical situations, is a brilliant approach to formalize the logic for developing the autonomous vehicles. It allows the users to select their views on whose life to spare under critical scenarios and thus, gathers human perspectives for intense situations. The continuous development in this field will eventually help develop machines with human-like intelligence to help bloom the driverless community.

We are already aware that the autonomous systems are taking away the jobs of humans. A study by National Public Radio (NPT) [77] states that truck driving is the most common job for 28 states in the United States of America (USA). According to NPT, the job has been immune to automation. Jerry Kaplan, a Stanford lecturer, states for an article in the Los Angeles Times [78], "If you can get rid of the drivers, those people are out of jobs, but the cost of moving all those goods goes down significantly." The article questions the stake of the jobs for the 1.7 million truckers in the USA. While there are too many delicate maneuvers, turns, and unforeseen circumstances for trucks to be handed over to a robot, the possibility of the trucks running 24/7 can improve the efficiency of the trucks and indulge the drivers into jobs which require more creative minds.

While small cars are already being driven autonomously on city roads and high-ways, heavy duty vehicles require more precision and developments. The available technologies to detect obstacles and roads are not sufficient for the off roads. A study conducted for agricultural semi-autonomous vehicles states that a highly automated vehicle would reduce the operator's situation awareness compared to the one with partially automated support [79]. The tasks with high-precision require the full attention of the operators stating that the tasks cannot be compromised for. Such scenarios question the authenticity of autonomous vehicles. Thus, for unknown or unusual circumstances, it is still debatable whether to have a human take over the control under extreme or terminal conditions.

Appendix C

Training Models

The Appendix C gives the training models for all the tests. The orange lines represent the accuracy (calculated by cross-validation with respect to the validation data), the blue lines represent the training losses (observed with the training data) and the green lines represent the validation losses (observed with the corresponding validation data). All the tests have been conducted for 30 epochs for the different datasets. The learning rates are varying every 10 epochs, i.e. the learning rate is 0.01 for first 10 epochs, 0.001 between 11 and 20 epochs and 0.001 for 21 to 30 epochs. This allows the network to overcome getting stuck at local minimas.

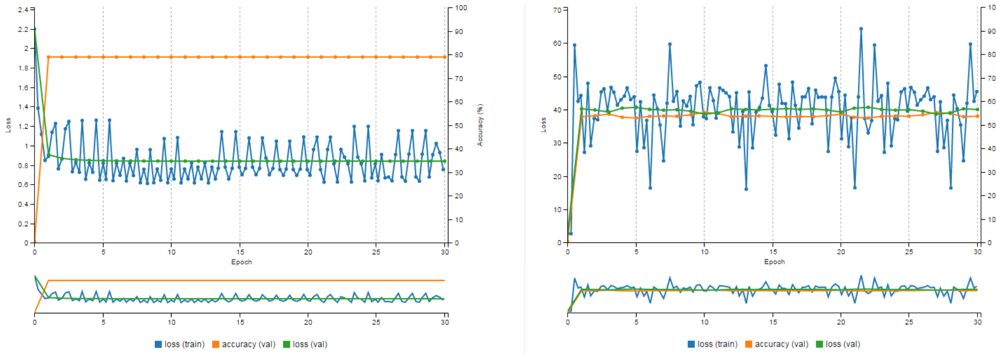


Figure C.1: Preliminary Training Models using Original Images for Set 1 (682 images) with No Pre-trained Models for 9 classes (Left) and 16 classes (Right).

In Figure C.1, the Set 1 with 682 images is used to train the 9 and 16 class models without any pre-trained model. For both the cases, the accuracy becomes constant after the first epoch. This implies that the weights have settled and indicates overfitting to the training data. The losses can be seen fluctuating due to the randomization in the order of training data. The learning rate also modifies at epoch 10 and 20 but clearly, there is no improvement on the accuracy.

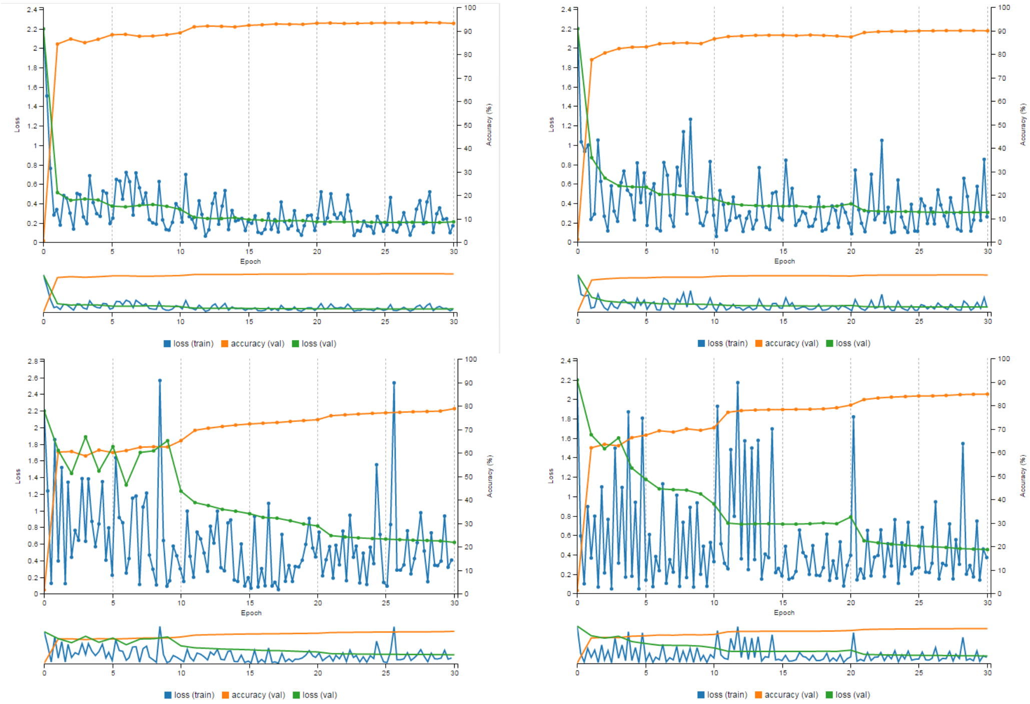


Figure C.2: Preliminary Training Models using Original Images with Pre-trained AlexNet Model for 9 classes. (Clockwise from Top) Set 1 (682 images), Set 2 (1364 images), Set 4 (9548 images) and Set 3 (6820 images).

In Figure C.2, the model is pre-trained on AlexNet network and an immediate improvement is noticed with respect to the non pre-trained model. Henceforth, the models use the pre-trained network for all the tests. The Sets 1, 2, 3 and 4 with 682, 1364, 6820 and 9548 images are used to generate the model for the 9 drivable classes.

We can notice that the Set 1 performs the best but it is also good to note that the drop of learning rate at epoch 10 and 20 results in the sudden improvement in the accuracies. Set 1 containing the distorted images contain lower variance in the losses compared to the other datasets containing the more diverse augmented images.

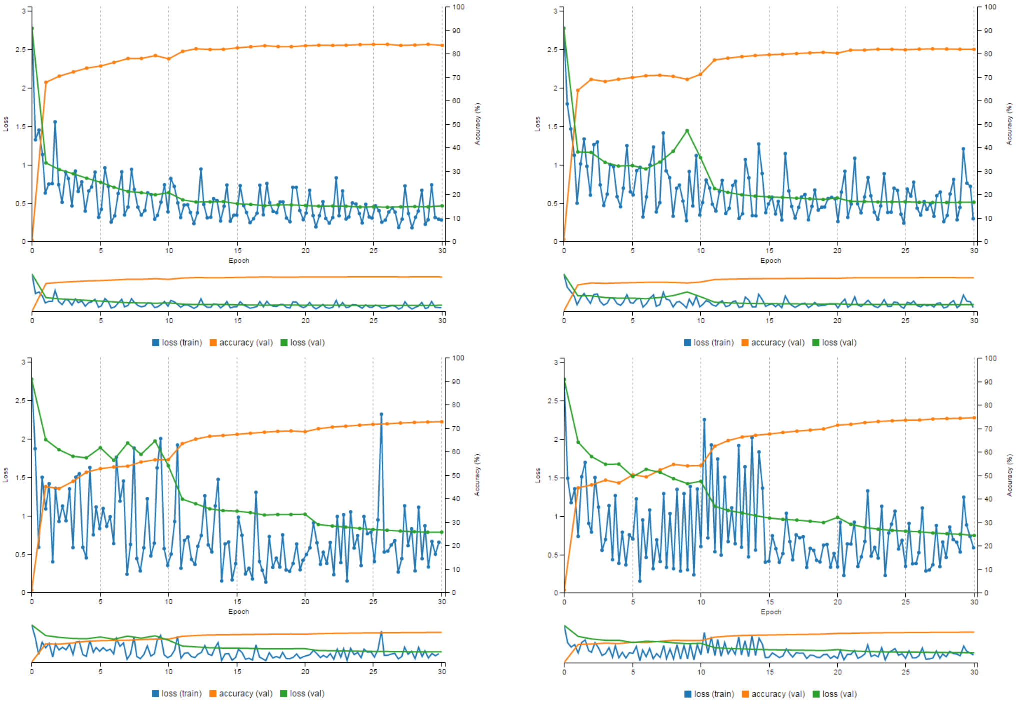


Figure C.3: Preliminary Training Models using Original Images with Pre-trained AlexNet Model for 16 classes. (Clockwise from Top) Set 1 (682 images), Set 2 (1364 images), Set 4 (9548 images) and Set 3 (6820 images).

In Figure C.3, the model is pre-trained on AlexNet network again but was provided with 16 classes. A subsequent drop is noticed in the accuracies for all the sets compared to the 9 class models.

The 16 class is different from the 9 drivable classes as it categorizes the background regions into the non-drivable classes. This implies that the additional classes led to a decrease in the validity of the model. While the training losses are fluctuating, the validation losses seem to generate certain peaks which result in improving the accuracy.

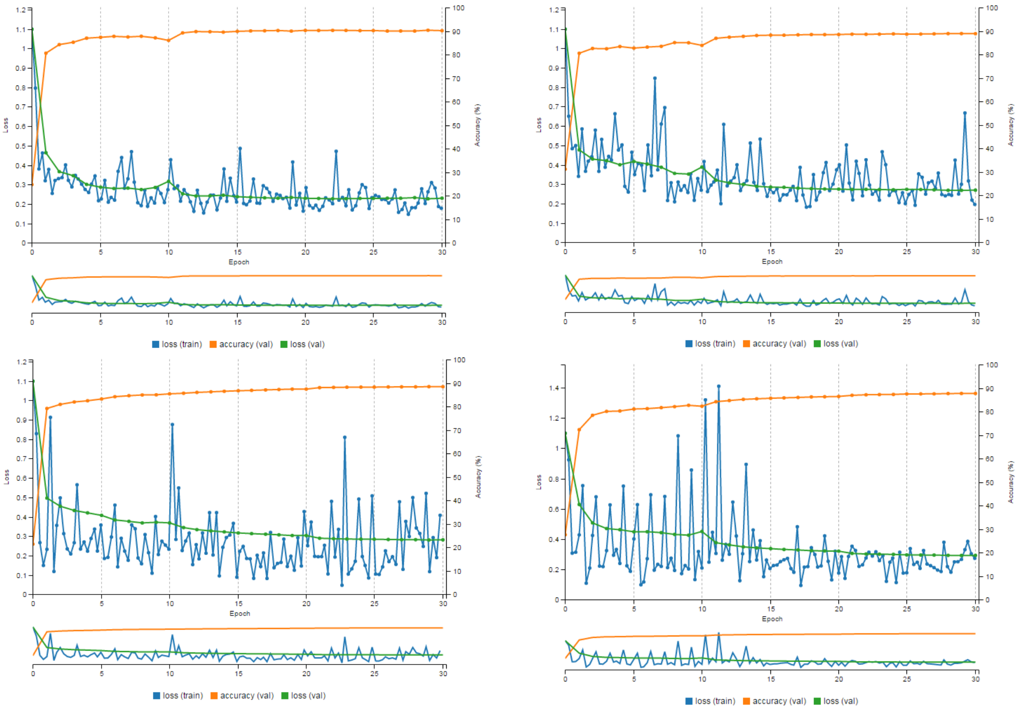


Figure C.4: Preliminary Training Models using Original Images with Pre-trained AlexNet Model for 3 classes. (Clockwise from Top) Set 1 (682 images), Set 2 (1364 images), Set 4 (9548 images) and Set 3 (6820 images).

In Figure C.4, the model is pre-trained on AlexNet network for 3 classes. The classes namely, drivable, non-drivable and background. This type of categorization is mostly confusing to the system as the human understanding of a drivable region may be just conceptual, for example, drivable snow is indifferent from non-drivable snow, but it is dependent whether the snow is next to the road or in between the woods.

It is interesting to note that the accuracy for Set 1 and 2 for the 3 class model is better than the 9 class and poorer than the 16 class models. However, this model seems to be performing the best for Set 3 and 4 implying that the cropped images and lesser classes improvised the pixel classification for the smaller images.

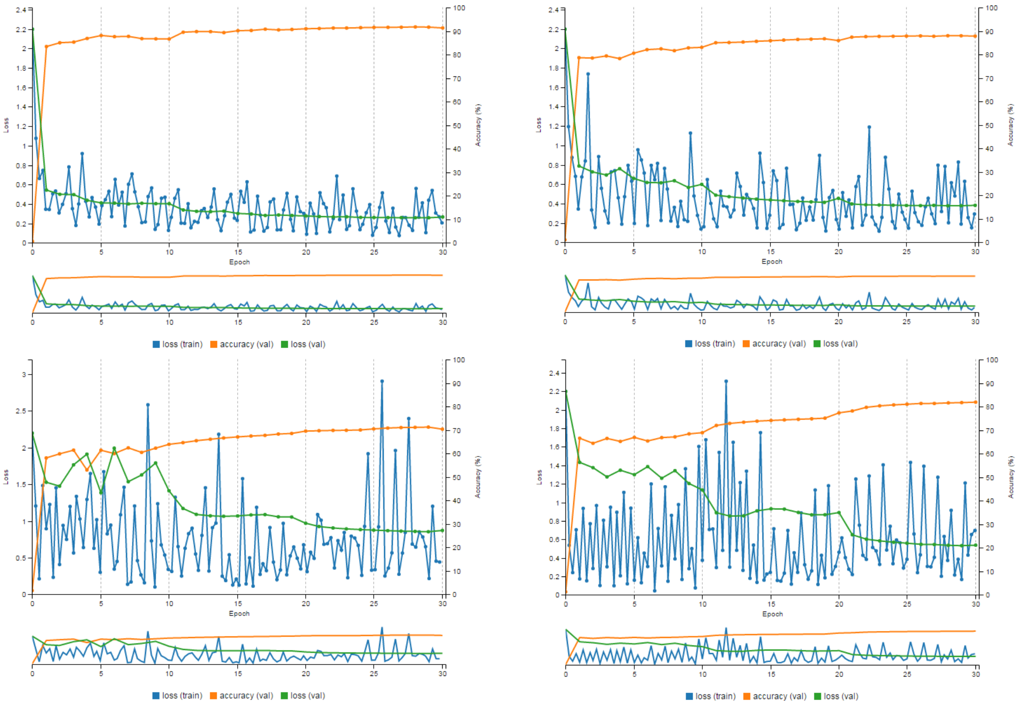


Figure C.5: Training Models using Gabor Images with Pre-trained AlexNet Model for 9 classes. (Clockwise from Top) Set 1 (682 images), Set 2 (1364 images), Set 4 (9548 images) and Set 3 (6820 images).

In Figure C.5, the 9 class model is trained on the Gabor responses of the original images and pre-trained on the AlexNet network. A minute drop in the accuracies can be seen in all the 4 Sets compared to the models trained on the original images. This reflects that the supply of the texture features suppressed the effect of the other features.

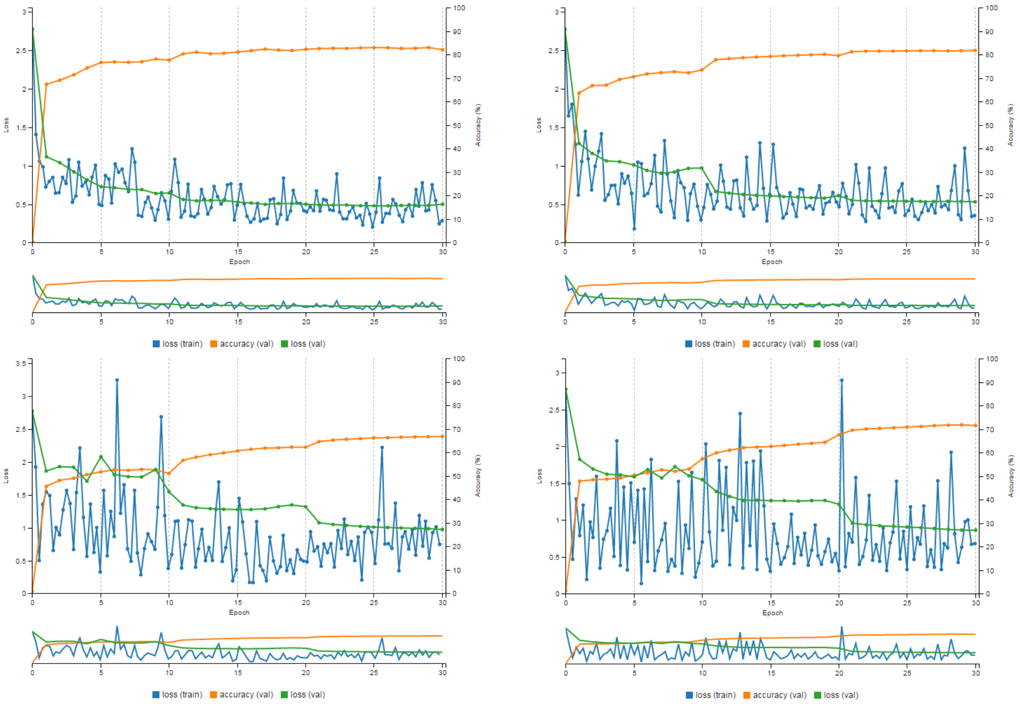


Figure C.6: Training Models using Gabor Images with Pre-trained AlexNet Model for 16 classes. (Clockwise from Top) Set 1 (682 images), Set 2 (1364 images), Set 4 (9548 images) and Set 3 (6820 images).

In Figure C.6, the 16 class model is trained on the Gabor responses of the original images and pre-trained on the AlexNet network. This replicates a similar phenomenon as seen in Figure C.5 regarding the drop of accuracies with respect to the original images. The contribution of the texture features fails to differentiate among the relevant classes.

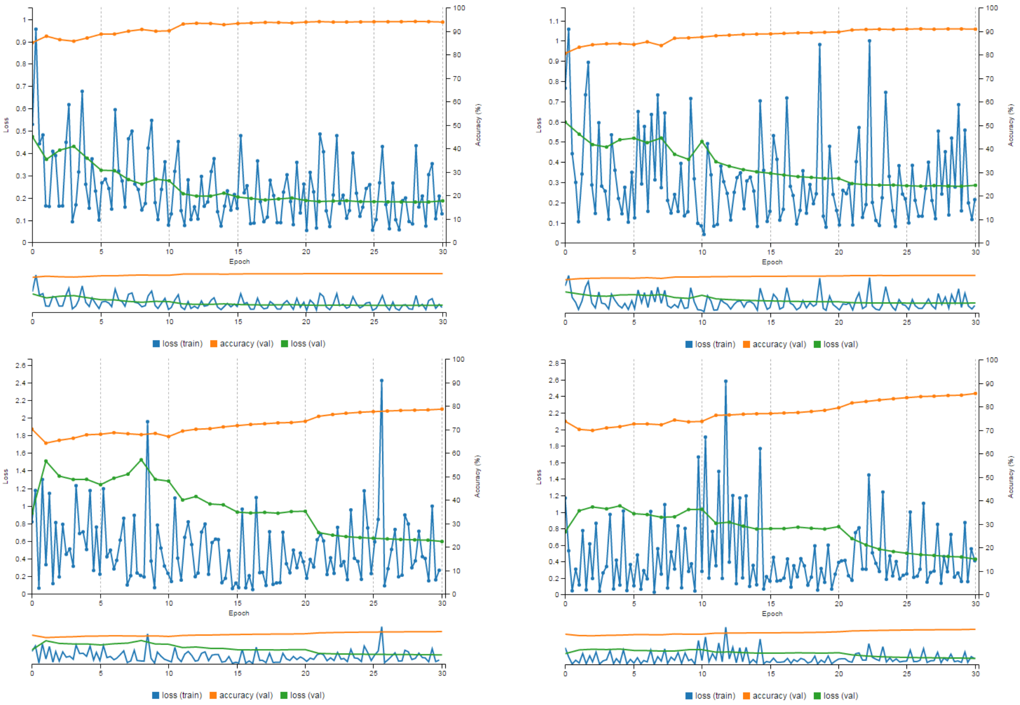


Figure C.7: Training Models using Gabor Images with Pre-trained Preliminary Training Model for 9 classes. (Clockwise from Top) Set 1 (682 images), Set 2 (1364 images), Set 4 (9548 images) and Set 3 (6820 images).

In Figure C.7, the model is pre-trained over the preliminary models, demonstrated in Figure C.2. A minute improvement is seen in the accuracies for all the sets compared to the original models.

The 9 class models generated used the models which were trained earlier on original images and has now been fed with the texture information. While this enhancement is only about 1-2% per model, there is a huge scope of improvement.

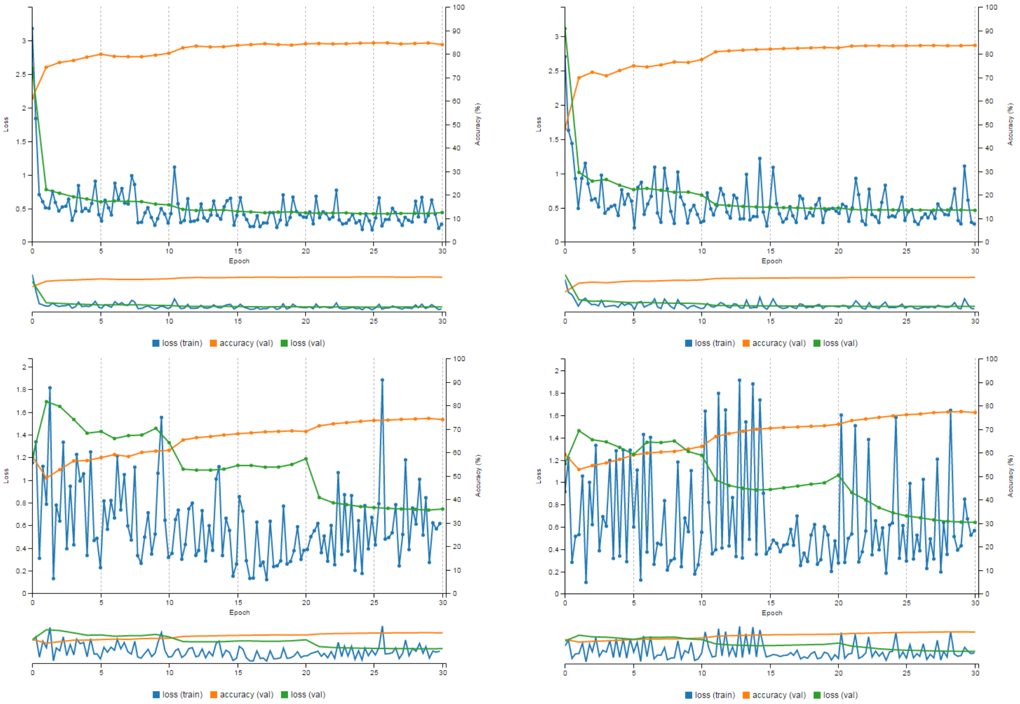


Figure C.8: Training Models using Gabor Images with Pre-trained Preliminary Training Model for 16 classes. (Clockwise from Top) Set 1 (682 images), Set 2 (1364 images), Set 4 (9548 images) and Set 3 (6820 images).

In Figure C.8, the model is pre-trained over the preliminary models, demonstrated in the Figure C.3. It is apparent that this model has improved the accuracy by about 2 - 3% compared to the original images. It is also important to note that there is a significant drop in the validation losses for both Set 3 and 4 for epochs 10 and 20. These sets have a higher variation because of the cropped regions and thus indicates the improvement by picking the global minimas.

Overall, the addition of the features using the Gabor filter on top of the model provided marginal improvements in the results indicating that the features overlapped to existing models are beneficial for the inferences.

Appendix D

Precision Matrices

Table D.1: Precision table for M1, 9 classes, 682 images.

CLASSES	Asphalt	Gravel_*H*	Gravel_*L*	Mud	Sand	Water	Grass	Snow	Background
Asphalt	88.04	0.72	4.29	9.5	6.32	50.48	3.07	1.56	0.54
Gravel_*H*	4.42	59.66	15.3	24.39	4.92	0	0	0	0.41
Gravel_*L*	2.18	12.62	48.76	14.4	3.17	4.25	0.04	0.21	0.45
Mud	1.14	19.89	13.23	43.82	0.14	0.36	1.6	0	0.74
Sand	0.49	0.21	0.06	0	70.02	0	0	0	0.14
Water	0.14	1.53	4.04	2.89	1.29	33.12	0.16	0	0.05
Grass	0.17	0	1.17	0.04	1.37	1.14	41.36	4.21	0.78
Snow	0.02	0.37	1.11	0.27	0	0.07	0	51.92	0.26
Background	3.4	5	12.04	4.68	12.78	10.58	53.78	42.1	96.63

Table D.2: Precision table for M1, 9 classes, 1364 images.

CLASSES	Asphalt	Gravel_H	Gravel_L	Mud	Sand	Water	Grass	Snow	Background
Asphalt	86.46	5.28	4.16	11.62	7.76	5.79	4.16	0.55	0.6
Gravel_H	0.65	44.75	5.44	9.37	0.59	1.79	0.21	0.03	0.47
Gravel_L	1.52	24.15	64.22	5.8	3.25	2.21	2.07	0.44	0.57
Mud	5.51	17.25	12.08	62.67	0.89	8.16	0.05	0.14	0.69
Sand	2.87	0.46	0.59	0.02	77.96	0.55	0.33	0	0.37
Water	0.17	0.34	2.29	1.88	0.16	74.89	0	0.27	0.07
Grass	0.14	0.16	0.04	1.42	2.57	0	35.26	0.09	0.73
Snow	0.01	0.1	0.26	0	0	0.01	0.06	71.14	0.48
Background	2.68	7.5	10.92	7.22	6.83	6.6	57.85	27.34	96.01

Table D.3: Precision table for M1, 9 classes, 6820 images.

CLASSES	Asphalt	Gravel_H	Gravel_L	Mud	Sand	Water	Grass	Snow	Background
Asphalt	84.6	3.66	5.66	17.65	15.09	6.35	3.77	0.76	1.4
Gravel_H	3.03	50.81	11.68	14.9	5.92	1.67	0	0.01	0.77
Gravel_L	2.73	15.71	66.06	10.73	3.94	16.62	7.23	1.42	2.47
Mud	4.26	18.34	7.39	41.76	1.03	1.1	0.68	0.15	1.76
Sand	1.28	0.2	0.06	0.31	56.05	0	0.05	0	0.58
Water	0.64	2.64	1.66	3.48	2.31	65.38	0	0.09	0.29
Grass	0.11	0.01	0.31	0.84	0.2	0	48.48	0.68	1.82
Snow	0.05	0.13	0.06	0.01	0	0.47	0	54.1	1.11
Background	3.3	8.5	7.12	10.32	15.47	8.4	39.78	42.79	89.82

Table D.4: Precision table for M1, 9 classes, 9548 images.

CLASSES	Asphalt	Gravel_*H*	Gravel_*L*	Mud	Sand	Water	Grass	Snow	Background
Asphalt	88.04	4.93	1.8	10	11.44	22.36	3.6	1.04	0.93
Gravel_*H*	4.64	54.04	11.04	9.66	3.24	1.06	0	0.01	0.93
Gravel_*L*	2.21	13.74	72.21	9.58	3.07	8.19	5.5	1.01	1.05
Mud	1.53	16.65	2.88	58.1	2.59	1.07	0.27	0.26	1.03
Sand	0.36	0.2	0.11	0.01	68.01	0	0	0	0.31
Water	0.13	1.55	1.47	3.81	0.61	54.59	0	0.09	0.15
Grass	0.13	0.23	0.15	0.34	0.36	0.02	60.11	0.69	2.11
Snow	0.02	0.05	0.15	0.03	0	3.98	0.02	57.31	0.6
Background	2.94	8.61	10.18	8.48	10.69	8.74	30.49	39.58	92.89

Table D.5: Precision table for M1, 3 classes, 682 images.

CLASSES	Drivable	Non Drivable	Background
Drivable	89.29	3.79	1.28
Non Drivable	2.65	82.48	5.21
Background	8.05	13.73	93.51

Table D.6: Precision table for M1, 3 classes, 1364 images.

CLASSES	Drivable	Non Drivable	Background
Drivable	90.18	3.9	1.76
Non Drivable	2.51	83.27	6.97
Background	7.3	12.83	91.26

Table D.7: Precision table for M1, 3 classes, 6820 images.

CLASSES	Drivable	Non Drivable	Background
Drivable	92.33	5.1	5.74
Non Drivable	1.59	81.7	8.99
Background	6.08	13.2	85.27

Table D.8: Precision table for M1, 3 classes, 9548 images.

CLASSES	Drivable	Non Drivable	Background
Drivable	91.91	4.63	3.91
Non Drivable	1.48	82.68	12.93
Background	6.6	12.69	83.17

Appendix E

Performance Matrices

Table E.1: F-Scores for 16 classes.

CLASSES	M1_16_682	M1_16_1364	M1_16_6820	M1_16_9548	M2_16_682	M2_16_1364	M2_16_6820	M2_16_9548	M3_16_682	M3_16_1364	M3_16_6820	M3_16_9548
Asphalt	0.9226	0.8973	0.8641	0.8766	0.8708	0.8843	0.8028	0.8406	0.9066	0.9118	0.8664	0.8891
Gravel_H	0.5098	0.5666	0.4813	0.4112	0.4429	0.4783	0.2851	0.2386	0.5423	0.5763	0.5192	0.5711
Gravel_L	0.5297	0.593	0.6122	0.6426	0.4467	0.564	0.501	0.5663	0.5893	0.6676	0.643	0.6778
Mud	0.4634	0.4492	0.3709	0.446	0.3631	0.4513	0.3008	0.3091	0.4688	0.4982	0.4289	0.508
Sand	0.7578	0.6918	0.7086	0.6981	0.6786	0.6973	0.5357	0.6528	0.7488	0.7454	0.7168	0.7133
Water	0.1038	0.4303	0.6555	0.7491	0.0048	0.2567	0.4875	0.6957	0.1513	0.5085	0.6428	0.7616
Grass	0.2023	0.3196	0.2886	0.3363	0.1536	0.3602	0.1491	0.3567	0.2254	0.3768	0.3759	0.4218
Snow	0.6683	0.6251	0.5784	0.5875	0.6141	0.6211	0.5559	0.5659	0.6921	0.6596	0.597	0.6068
Gravel_N	0.6294	0.6666	0.7665	0.7999	0.1335	0.7328	0.6908	0.8029	0.6555	0.7485	0.8153	0.8205
Mud_N	0.058	0.4112	0.3378	0.4626	0.0249	0.3061	0.1713	0.2913	0.3251	0.5041	0.3925	0.5267
Sand_N	0.5444	0.4698	0.5035	0.4251	0.3383	0.5208	0.098	0.3828	0.5703	0.6101	0.5967	0.4597
Sky_N	0.8806	0.8982	0.8695	0.875	0.8745	0.9076	0.8446	0.8746	0.878	0.9082	0.8861	0.8824
Vegetation_N	0.8162	0.8397	0.7786	0.8006	0.814	0.8372	0.7443	0.7915	0.8319	0.8509	0.8035	0.8236
Grass_N	0.7082	0.697	0.708	0.7336	0.6859	0.6784	0.6074	0.6996	0.7264	0.7169	0.7309	0.7597
Snow_N	0	0	0.5407	0.4616	0.5054	0.4901	0.5194	0.4606	0.543	0.5181	0.5581	0.5113
Background	0.9127	0.8784	0.7317	0.727	0.9098	0.8835	0.6944	0.7257	0.915	0.8866	0.7496	0.7483

COLOR INDEX 0.0 < F-Score < 0.4 0.4 < F-Score < 0.8 0.8 < F-Score < 1.0

Table E.2: F-Scores for 9 classes.

CLASSES	M1_9_682	M1_9_1364	M1_9_6820	M1_9_9548	M2_9_682	M2_9_1364	M2_9_6820	M2_9_9548	M3_9_682	M3_9_1364	M3_9_6820	M3_9_9548
Asphalt	0.8896	0.8846	0.875	0.8917	0.8377	0.8318	0.7895	0.8516	0.9012	0.882	0.8772	0.8885
Gravel_H	0.3889	0.5272	0.4938	0.5424	0.357	0.4734	0.2542	0.4246	0.5153	0.5822	0.543	0.5659
Gravel_L	0.5199	0.603	0.6079	0.6534	0.3976	0.5038	0.452	0.559	0.5656	0.6648	0.6331	0.6697
Mud	0.4092	0.3989	0.3664	0.5782	0.2771	0.2582	0.0604	0.4618	0.5099	0.49	0.4447	0.6177
Sand	0.7813	0.7727	0.646	0.777	0.6398	0.6616	0.4149	0.6628	0.7615	0.7538	0.5954	0.7513
Water	0.3285	0.6748	0.647	0.5676	0	0	0.3719	0.502	0.4229	0.6302	0.663	0.5782
Grass	0.1942	0.2797	0.3265	0.3695	0.1839	0.1816	0.0691	0.3687	0.4188	0.4059	0.4052	0.5074
Snow	0.5642	0.7021	0.5904	0.646	0.5176	0.7239	0.5661	0.5926	0.5992	0.7457	0.6315	0.6533
Background	0.9729	0.9669	0.8989	0.9291	0.9681	0.9628	0.8793	0.9201	0.9757	0.9703	0.9062	0.9338

COLOR INDEX 0.0 < F-Score < 0.4 0.4 < F-Score < 0.8 0.8 < F-Score < 1.0

Table E.3: IoU Scores for 16 classes.

CLASSES	M1_16_682	M1_16_1364	M1_16_6820	M1_16_9548	M2_16_682	M2_16_1364	M2_16_6820	M2_16_9548	M3_16_682	M3_16_1364	M3_16_6820	M3_16_9548
Asphalt	0.8563	0.8137	0.7608	0.7803	0.7712	0.7926	0.6706	0.725	0.8292	0.8379	0.7643	0.8003
Gravel_H	0.3421	0.3952	0.3169	0.2588	0.2844	0.3143	0.1662	0.1355	0.372	0.4048	0.3507	0.3996
Gravel_L	0.3603	0.4214	0.4411	0.4734	0.2876	0.3927	0.3343	0.395	0.4178	0.501	0.4739	0.5126
Mud	0.3016	0.2897	0.2277	0.287	0.2218	0.2914	0.1771	0.1828	0.3062	0.3318	0.273	0.3405
Sand	0.61	0.5289	0.5487	0.5362	0.5136	0.5353	0.3658	0.4846	0.5985	0.5941	0.5586	0.5544
Water	0.0547	0.2742	0.4875	0.5989	0.0024	0.1473	0.3223	0.5334	0.0818	0.3409	0.4736	0.615
Grass	0.1125	0.1902	0.1686	0.2021	0.0832	0.2197	0.0806	0.2171	0.127	0.2321	0.2315	0.2673
Snow	0.5018	0.4547	0.4069	0.4159	0.4431	0.4505	0.3849	0.3946	0.5291	0.4921	0.4256	0.4355
Gravel_N	0.4592	0.5	0.6214	0.6665	0.0715	0.5783	0.5276	0.6707	0.4875	0.5981	0.6882	0.6956
Mud_N	0.0299	0.2588	0.2032	0.3009	0.0126	0.1807	0.0936	0.1705	0.1941	0.3369	0.2441	0.3575
Sand_N	0.374	0.307	0.3364	0.2699	0.2036	0.3521	0.0515	0.2367	0.3989	0.439	0.4252	0.2985
Sky_N	0.7866	0.8152	0.7691	0.7778	0.777	0.8308	0.731	0.7771	0.7826	0.8319	0.7955	0.7896
Vegetation_N	0.6895	0.7237	0.6374	0.6675	0.6863	0.7199	0.5927	0.655	0.7121	0.7405	0.6715	0.7001
Grass_N	0.5482	0.5349	0.548	0.5793	0.5219	0.5133	0.4361	0.538	0.5704	0.5587	0.5759	0.6125
Snow_N	0	0	0.3706	0.3	0.3381	0.3246	0.3508	0.2992	0.3727	0.3497	0.387	0.3435
Background	0.8395	0.7832	0.577	0.5711	0.8346	0.7914	0.5319	0.5695	0.8433	0.7962	0.5995	0.5978

COLOR INDEX $0.0 <$ IoU Score < 0.4 $0.4 <$ IoU Score < 0.8 $0.8 <$ IoU Score < 1.0

Table E.4: IoU Scores for 9 classes.

CLASSES	M1_9_682	M1_9_1364	M1_9_6820	M1_9_9548	M2_9_682	M2_9_1364	M2_9_6820	M2_9_9548	M3_9_682	M3_9_1364	M3_9_6820	M3_9_9548
Asphalt	0.8012	0.7931	0.7777	0.8045	0.7207	0.712	0.6523	0.7416	0.8202	0.7889	0.7813	0.7993
Gravel_H	0.2414	0.358	0.3279	0.3721	0.2173	0.3101	0.1456	0.2695	0.3471	0.4106	0.3727	0.3946
Gravel_L	0.3512	0.4316	0.4367	0.4852	0.2481	0.3367	0.292	0.3879	0.3943	0.4979	0.4631	0.5034
Mud	0.2573	0.2492	0.2243	0.4067	0.1608	0.1482	0.0311	0.3002	0.3422	0.3245	0.2859	0.4469
Sand	0.6411	0.6296	0.4771	0.6353	0.4704	0.4943	0.2617	0.4957	0.6149	0.6048	0.4239	0.6017
Water	0.1965	0.5092	0.4782	0.3963	0	0	0.2284	0.3351	0.2681	0.46	0.4958	0.4066
Grass	0.1075	0.1626	0.1951	0.2266	0.1013	0.0998	0.0358	0.226	0.2649	0.2546	0.2541	0.3399
Snow	0.3929	0.5409	0.4188	0.4771	0.3491	0.5673	0.3948	0.4211	0.4278	0.5945	0.4614	0.4851
Background	0.9471	0.9359	0.8164	0.8676	0.9382	0.9283	0.7846	0.852	0.9525	0.9423	0.8285	0.8758

COLOR INDEX 0.0 < IoU Score < 0.4 0.4 < IoU Score < 0.8 0.8 < IoU Score < 1.0