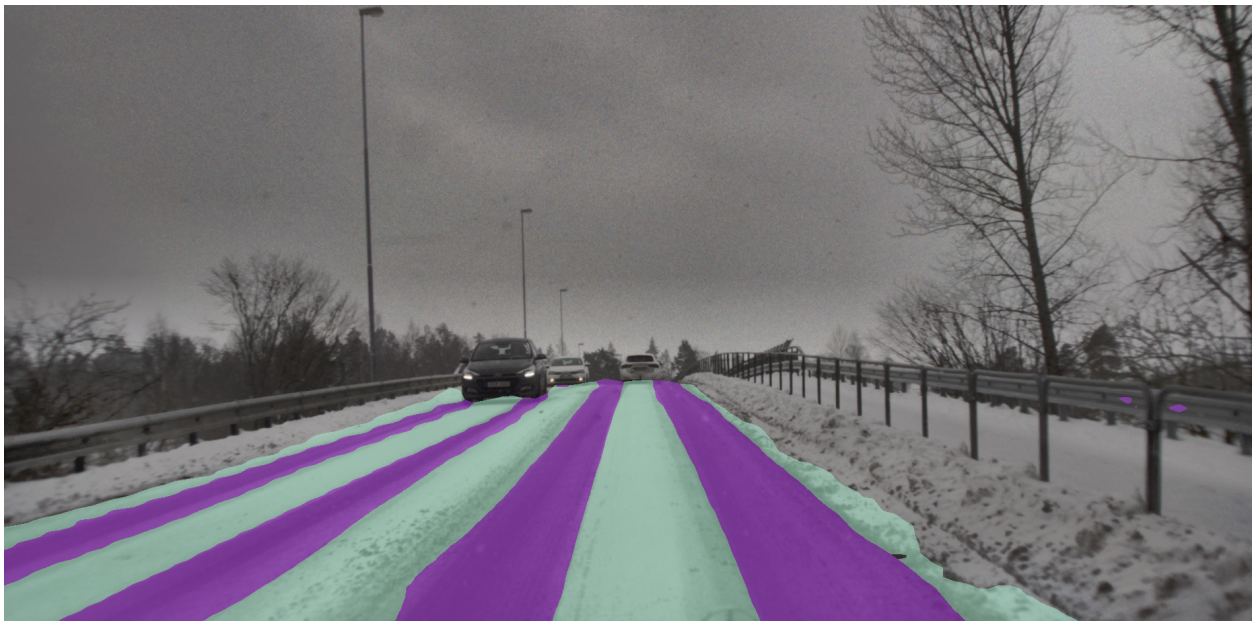
CHALMERS
UNIVERSITY OF TECHNOLOGY



Road Surface Condition Detection

EENX16-23-29

Bachelor's thesis in Electrical Engineering

Jonatan Andersson, Liam Hellring, Dennis Hjertén, Gustav Mattson, Ludvig Möller

DEPARTMENT OF ELECTRICAL ENGINEERING

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2023
www.chalmers.se

BACHELOR'S THESIS 2023

Road Surface Condition Detection

Jonatan Andersson
Liam Hellring
Dennis Hjertén
Gustav Mattson
Ludvig Möller

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Chalmers University of Technology
Gothenburg, Sweden 2023

Road Surface Condition Detection

Supervisor: Hasith Karunasekera, Electrical Engineering
Examiner: Jonas Sjöberg, Electrical Engineering

Bachelor's Thesis 2023
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg
Sweden
Telephone +46 31 772 1000

Cover: Image of a semantically segmented snowy road with tracks, result from a convolution neural network model. More can be read in section 2.3 and in the result section. Original image from the DENSE Seeing Through Fog dataset [1].

Typeset in LaTeX
Gothenburg, Sweden 2023

Abstract

The Road Surface Condition (RSC) significantly impacts a vehicle's manoeuvrability, braking capability and overall performance. Whether the road is dry, wet or covered in snow, it impacts how the vehicle responds to the driver's - or an autonomous system's input. If the vehicle could determine the RSC autonomously, it could help both drivers and autonomous vehicles with decision making. By notifying the driver or autonomous systems about the RSC ahead, it will improve safe operation and efficient vehicle functionalities.

This thesis focused on developing a Convolution Neural Network (CNN) model capable of performing high resolution classification of RSC through semantic segmentation. The input data for this model consists of images captured from a vehicle.

In order to train the CNN model a dataset is constructed by extending a dataset done by a previous project at Chalmers University [2]. The dataset doubled in size by manually labeling images from the DENSE Seeing Through Fog dataset [1] and the Mapillary Vistas dataset [3] using the online tool CVAT [4]. The final dataset consists of 1369 images with semantic labels.

Three segmentation architectures, PSPnet [5], OCRnet [6] and FCN [7], were compared using two loss functions. PSPnet was further fine-tuned to achieve the best performance of 80.19% mIoU (mean Intersection over Union) when segmenting the different RSC classes wet, dry and snow.

Acknowledgements

We would like to express our sincere gratitude to our supervisor of this project, Hasith Karunasekera, post-doctoral researcher in the Mechatronics group at Chalmers University of Technology. For his valuable guidance and help throughout the whole project.

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis:

RSC	Road surface condition
CNN	Convolutional Neural Network
ANN	Artificial Neural Network
SGD	Stochastic Gradient Descent
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
IoU	Intersection over Union
mIoU	mean Intersection over Union
OCRnet	Object-Contextual Representations network
PSPnet	Pyramid Scheme Parsing network
FCN	Fully Convolutional Network
Lr	Learning rate

Contents

List of Acronyms vi

List of Figures viii

List of Tables 1

1 Introduction 2
- 1.1 Background . . . 2
- 1.2 Contribution . . . 3
- 1.3 Limitations / Demarcations . . . 4

2 Theoretical Background 5
- 2.1 Artificial Neural Networks . . . 5
  - 2.1.1 Activation Functions . . . 5
    - 2.1.1.1 Rectified Linear Unit . . . 6
  - 2.1.2 Loss Functions . . . 6
    - 2.1.2.1 Cross-Entropy loss . . . 7
    - 2.1.2.2 Weighted Cross-Entropy loss . . . 7
  - 2.1.3 Optimizer . . . 7
    - 2.1.3.1 Gradient Descent . . . 8
    - 2.1.3.2 Stochastic Gradient Descent (SGD) . . . 8
    - 2.1.3.3 SGD with Momentum . . . 8
  - 2.1.4 Regularization . . . 8
  - 2.1.5 K-fold Cross-validation . . . 8
- 2.2 Convolutional Neural Networks . . . 9
  - 2.2.1 CNN model architectures . . . 10
  - 2.2.2 Transfer Learning . . . 10
  - 2.2.3 Semantic Segmentation . . . 10
- 2.3 Dataset . . . 11
  - 2.3.1 Data Augmentation . . . 11
  - 2.3.2 Class Imbalance . . . 11
  - 2.3.3 Splitting the data . . . 12
- 2.4 Performance Metrics . . . 12
  - 2.4.1 Accuracy . . . 13
  - 2.4.2 Mean Intersection over Union . . . 13

3 Methodology for RSC Segmentation 15

3.1 Creating the Dataset . . . 15
3.1.1 Semantic Labeling . . . 15
3.1.1.1 Tag annotation . . . 16
3.1.2 Labeling Policy . . . 17
3.2 Models used in RSC segmentation . . . 17
3.2.1 Evaluating the models . . . 18
3.3 Optimization of PSPnet . . . 18

4 Results and Discussion **19**
4.1 Dataset . . . 19
4.1.1 Reflection about Dataset . . . 19
4.1.2 Balance . . . 21
4.1.3 Labeling . . . 21
4.2 Performance of Models . . . 21
4.2.1 Models trained with cross-entropy loss and weighted cross-entropy loss . . . 21
4.2.2 Optimizing PSPnet: Different Learning Rates . . . 26
4.2.3 Assessing Performance through Cross-validation for PSPnet . . . 28
4.2.4 Optimizing PSPnet: Extended Training . . . 29
4.3 Qualitative Analysis . . . 30
4.4 Ethical aspects . . . 31

5 Conclusion and Future work **32**
5.1 Conclusion . . . 32
5.2 Future Work . . . 32
5.2.1 Improvement of the dataset . . . 32
5.2.2 RSC Segmentation with Roadtype classification . . . 33
5.2.3 Further fine-tuning of the model . . . 33

Bibliography **33**

List of Figures

1.1 Winter road, from the DENSE dataset [1]. 3

2.1 Visualization of a node. 6
2.2 The ReLU activation function. 6
2.3 Example of an CNN-model structure from [8]. 9
2.4 Semantic segmentation example, from Cityscapes [9] used with permission. 11
2.5 Visualization of TP, FP, TN and FN. 12
2.6 IoU visualized. 13

3.1 Ground truth labels showing the wet, dry, packed snow, tracks snow and loose snow classes. Images taken from DENSE dataset [1]. 16

4.1 mIoU (%) of the models using cross entropy loss. 23
4.2 mIoU (%) of the models, with weighted cross entropy loss. 23
4.3 Confusion matrices for all three models using cross-entropy and weighted cross-entropy loss. 25
4.4 mIoU (%) of PSPnet, with learning rates (Lr) of 0.001, 0.01 and 0.1. . 26
4.5 mIoU (%) of PSPnet, with learning rates (Lr) of 0.005, 0.01 and 0.02. 27
4.6 mIoU (%) of PSPnet with cross validation. 28
4.7 mIoU (%) of PSPnet with cross validation. 29
4.8 Good predictions (left) and the ground truth (right), images taken from the Dense dataset [1]. 30
4.9 Bad predictions (left) and the ground truth (right), images taken from the Dense dataset [1]. 30

List of Tables

3.1 RSC Classes, Labels and corresponding RGB values. 16
3.2 mIoU for the selected models from MMsegmentation benchmark[10], using 40,000 iterations and evaluated on the Cityscapes[11] dataset. . 18

4.1 Presentation of the training dataset. 19
4.2 Presentation of the validation set before. 19
4.3 Presentation of dataset for training, consisting of 969 images. 20
4.4 Presentation of dataset for evaluation, consisting of 200 images. . . . 20
4.5 Presentation of dataset for testing, consisting of 200 images. 20
4.6 RSC Classes, Labels and corresponding RGB values for the merged dataset. 21
4.7 resulting mIoU (%) of the three models for 40,000 iterations, with cross-entropy loss and weighted cross-entropy loss. 22
4.8 Final IoU (%) for each class using cross-entropy loss and weighted cross-entropy loss for the 3 models. 23
4.9 mIoU (%) for PSPnet with learning rates (Lr) of 0.001, 0.01 and 0.1. 26
4.10 mIoU (%) for PSPnet, with learning rates of 0.005, 0.01 and 0.02. . . 27
4.11 Training set for cross validation consisting of 969 images. 28
4.12 Validation set for cross validation containing of 200 images. 28
4.13 Final mIoU (%) of the three evaluations. 29

1
Introduction

Vehicle crashes happens everyday and one of the major factors behind these is the road surface condition (RSC). The RSC of a road is the result of the material the road is made of, wear and tear on the road and the weather conditions. Statistics from [12], show that about 20% out of all car crashes in the US are related to weather conditions (10 year average 2007-2016). How weather conditions such as rain and snow affects the RSC is tested in [13] where it is shown how the friction between a vehicle and the road is changing depending on the weather condition. The friction coefficients decreases for weather conditions such as snow and rain compared to a dry road and roads with different friction changes the way a vehicle behaves when driving on it by for example increasing the breaking distance when driving on a road with a lower friction.

1.1 Background

If it is possible to determine the RSC ahead of a vehicle, that information could be used to either help the driver by warning about slippery road conditions (snow, wet road etc.) or it could be used by active safety functions. For example it could slow down the vehicle if the road ahead is prone to a lower friction because of its RSC to counter the increased braking distance.

With images taken by a camera mounted on a car, deep learning can be used to create a model that is able to predict the RSC of an image. This has previously been done through classification using a convolutional neural network (CNN), training it with images assigned with a label of its RSC on the drivable road as a whole [14]. A flaw with this method however is its incapability of knowing where the different RSC:s are located in the image. There are many cases where getting more information than the most prevalent RSC can be valuable. On a snowy road there could be tracks as displayed in Figure 1.1 and on a mostly dry road there could be wet patches.

Figure 1.1: Winter road, from the DENSE dataset [1].

Being able to find these details can prove helpful as the different RSC:s have different friction coefficients. Dry asphalt roads have coefficients between 0.7 to 0.8, wet asphalt roads have coefficients between 0.4 to 0.5 and snowy roads have coefficients between 0.2 to 0.3 [15]. Because of the big difference between friction coefficients amongst the RSC:s, it is good to be able to distinguish them from each other. There are ambiguities within these three previously mentioned RSC:s, for example how deep the snow is or if a wet road contains standing water which impacts the friction as well. Some of these ambiguities within the snow class can be more precisely classed as loose snow, packed snow and tracks snow. However, most roads could be described using one of the three RSC:s: dry, snow and wet. If it is possible to know which one of these RSC:s are present in the road, then a good understanding of the friction could be established.

1.2 Contribution

The main contribution of this project is the development of a deep learning model for detecting the RSC using semantic segmentation for images taken with a front facing camera mounted on a car. Using semantic segmentation, every pixel in the images get a label which results in a highly detailed outcome. To develop the model, Three different semantic segmentation architectures are compared and one of them is chosen as the best performing.

In order to train a model, a dataset fit for the specific task is created. The dataset consists of 1369 images segmented with the RSC classes dry, wet and snow.

1.3 Limitations / Demarcations

The model is be able to identify the road surface condition of drivable road in an image and where it is located. Therefore the following limitations.

- This study focuses exclusively on five RSC classes: Dry, wet, packed snow, loose snow and tracks snow.
- The study focuses on images taken during the daytime and well lit images during night time.
- Images used are limited to asphalt and cobblestone roads.

2
Theoretical Background

In the theoretical background chapter, the theory behind how a neural network works and how they can be used to classify the RSC using semantic segmentation is explained.

2.1 Artificial Neural Networks

An Artificial Neural Network (ANN) is a machine learning model that can be used for classification tasks. It is constituted of an input layer, a number of hidden layers and an output layer. The first layer, the input layers, is where the data enters the network. The input layer then passes the data to the first hidden layer, there has to be at least one hidden layer but there can also be many hidden layers.
Each layer consists of a number of neurons or nodes. The nodes are interconnected with nodes in the previous layer and the next layer. Each connection holds a weight, which is a scalar that multiplies the value sent through the connection.
The node then passes an output forward created by summing its inputs together with a bias constant and then passing the sum through an activation function, see Figure 2.1. The last layer is an output layer that is designed to produce a result tailored for the task of the network [16].
Neural networks generally make use of supervised training. Meaning the network is provided with both inputs and the expected result of the output. After the network produces a result from the input, it then compares it to the desired output. After comparison the weights of the connections within the network are adjusted and through repeated training the weights are refined [17].

2.1.1 Activation Functions

An activation function is used to determine the output from the nodes in a neural network and is needed as it introduces non-linearity to the ANN. This means that the function is responsible for evaluating if the input to the node should activate it or not [18]. From the input, a value z will be determined from he weights and bias of the node. This value then goes through the activation function which produces the output. This process is visualized in Figure 2.1.

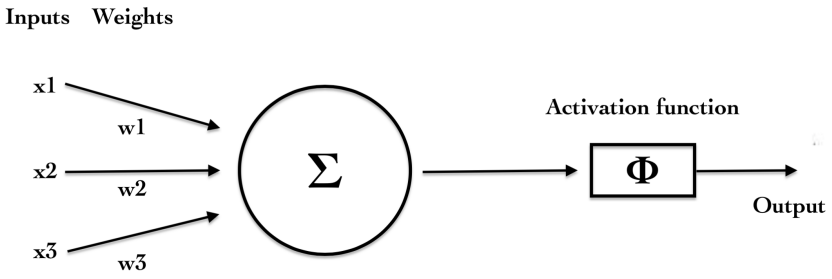


Figure 2.1: Visualization of a node.

2.1.1.1 Rectified Linear Unit

A commonly used activation function is Rectified Linear Unit or ReLU for short, which is a nonlinear activation function. The ReLU activation function looks like the following:

$$f(z) = max(0, z) \tag{2.1}$$

and is visualized in Figure 2.2.

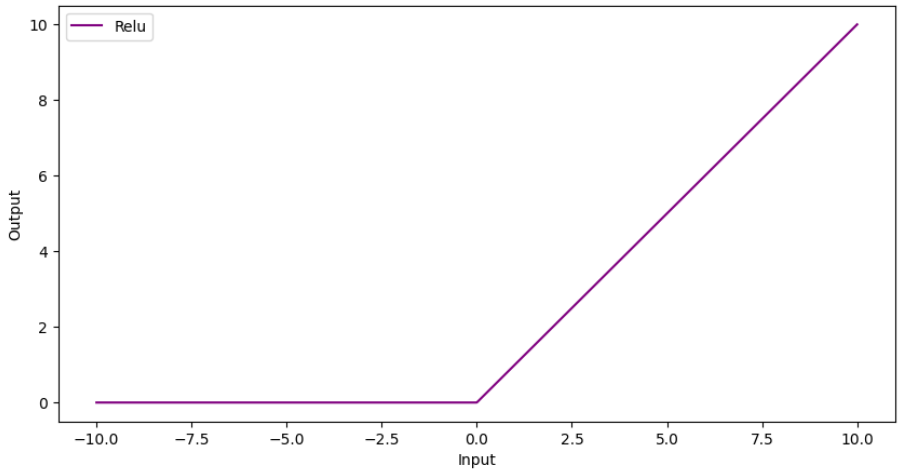


Figure 2.2: The ReLU activation function.

The ReLU function will as long as z is greater than 0, return z and if not it will return 0. [19]

2.1.2 Loss Functions

A loss function in machine learning is used to determine the error or loss between the predicted output of the model and the true output.[8] The loss function is a

scalar value that the model aims to minimize. If the prediction from a model is far from the truth, the loss is greater. There are many different loss functions, all fitting to different types of applications and datasets.

2.1.2.1 Cross-Entropy loss

The cross-entropy loss function is commonly used for CNN models and is mathematicaly expressed as;

$$H(p, y) = -\sum_i y_i \log(p_i) \quad where \quad i \in [1, \sum output\ classes] \tag{2.2}$$

In equation 2.2, p represents the predicted probability of of the ground truth and y being the actual probability of the ground truth. In the case of predicting what number a picture contains, from pictures containing one number of 1-4. The ground truth would look like [0,1,0,0] while the predicted output might look like [0.2, 0.75, 0.025, 0.025]. This would then result in a loss of $-1 * log(0,75) = 0,125$. If the model was to perform better and get a higher probability of the correct label then this loss would be decreased.

2.1.2.2 Weighted Cross-Entropy loss

To work against the problem of a class imbalance in the dataset, a modification of cross-entropy loss called weighted cross-entropy loss can be used [20]. This version is displayed in equation 2.3.

$$H_w(p, y) = -\sum_i w_i y_i \log(p_i) \quad where \quad i \in [1, \sum output\ classes] \tag{2.3}$$

Weighted cross-entropy makes use of precalculated weights (w_i) from the different classes which makes the loss increased for underrepresented classes so that the network trains more heavily on these. To get a deeper understanding of class imbalance, this is further explained in 2.4.3. The classes weights are calculated by looking at the spread of data and using equation 2.4.

$$w_i = \frac{1}{N_i} \cdot \frac{N_{tot}}{c} \tag{2.4}$$

Where w_i is the weight of class i, N_i is the amount of datapoints from class i, N_{tot} is the total amount of datapoints and c is the amount of classes. With these weights, if one class is represented more than another then its weight will be affected with same factor. This means that if a class is represented five times more than another, it will also have a weights that is five times smaller than that class.

2.1.3 Optimizer

An optimizer is used to minimize the loss function [21]. The goal of the optimizer is to find the global minimum of the loss function. However in the case where the function is not convex it will try to reach the lowest value within its neighborhood.

2.1.3.1 Gradient Descent

Gradient descent is a common optimizer. After each iteration during training, the weights are so that the loss function moves in the opposite direction of the gradient. The loss function moves down the slope towards a minimum. Gradient descent use the sum of all gradients for the whole data set. In other words the weights are adjusted after going through the entire set of data. This ensures that the trajectory is in the direction of the local minimum and if the function is convex that is the global minimum.

2.1.3.2 Stochastic Gradient Descent (SGD)

SGD is a variation of gradient descent where the weights are updated after each batch of images. Instead of updating after going through the whole data set. A disadvantage of regular gradient descent is that it can require a lot of memory [22], especially on large data sets. Because SGD does not have to go through the whole data set before adjusting the weights it requires less memory and can therefore be used with larger data sets.

2.1.3.3 SGD with Momentum

In order to increase stability, momentum can be used together with SGD. Momentum means that the previous gradient is also used when adjusting the weights.

2.1.4 Regularization

Overfitting occurs when machine learning models fit the training data too closely and therefore losing performance on classifying new and unseen data [23]. To prevent overfitting, different types of regularization can be used. These techniques can include changing the input data in some ways, called data augmentation, or modifying the networks neurons, such as adding dropout. By adding greater variety in the training, the model becomes more robust and can generalize better to unseen data.

2.1.5 K-fold Cross-validation

K-fold cross-validation is a method for testing the robustness of model performances. The k stands for how many equal parts the dataset should be split into. Before splitting the dataset it needs to be shuffled so that the classes are as spread out as possible. One of these parts will be the validation set and the rest will be used as the training set. Then after one of the parts has been in the validation set, another one is chosen and the previous validation set joins the training set. This way, every part will be used as validation set once and will the results will show how robust the model is [24].

2.2 Convolutional Neural Networks

Covolutional neural networks (CNN) is a common and well known type of neural network. It is inspired by living creatures and their visual perception mechanism [25]. CNN is most commonly used for analysing images.

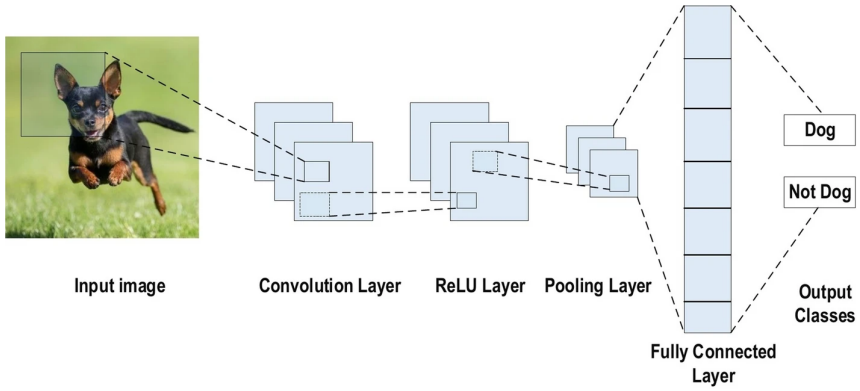


Figure 2.3: Example of an CNN-model structure from [8].

Because CNN is used for analysing images the architecture of CNN-models is specifically made to suit that type of data. An exampel of a CNN aritherture can be seen in the Figure 2.3. CNNs are made up of three layers comprised of convolutional layers, pooling layers and fully-connected layers [26].
Convolutional layers are a vital part of CNN and is the main building blocks and the layers parmeters are mainly focused on kernals that are learnable [26].
Kernal is a grid of descret values and the values is called kernal weights [8]. In the beginning of the training process each weight will be a random number and as the training progress these weights will be adjusted every time the model trains and the kernals learn to extract key features.

The convolutional layers get an image as inputs that is represented as an matrix with a number of N dimensions which then is convolved and generates a feature maps [8].

Pooling layers are used for downsampling the future maps from the convolutional layers by reducing the dimensions [27]. Which means that the parameters in that activation will be reduced.

Fully-connected layers will be the last layers in a CNN-model. Each neuron in this layer are connected to every neuron in the last layer [27]. The last layers of the convolution or pooling are often flatted, transformed to a 1D array of numbers. They are also connected to one or more fully connected layers and when the future maps are extracted the final output will be a percentage of the probability of each class. The Fully-connected layer utilize as the classifier in a CNN.

2.2.1 CNN model architectures

CNN models can be designed in many different ways and have different layouts [8]. This is called model architectures and they play a crucial role in improving the CNN models performances in different types of applications. The first CNN architecture was made in 1989 and a lot has happened since then. Reorganization of the processing-unit, optimizing parameters and network depth are some examples. To get better performance of CNN models, innovations in depth and spital explotations has been a big factor [28]. Depth based CNNs architectures focus on that the network can approximate the target function with non linear mappings. Depth in CNN networks have played a crucial role in supervised learning's success and it is shown that deeper networks can represent classes more efficiently then shallower networks in theoretical studies. Deeper networks also works better and are more efficient for complex tasks.
CNNs have large amount of parameters and hyper-parameters for example weights, processing units(neurons), filter size, learning rate and stride [28]. Convolutional operations considers the input pixels locality which means you different correlations levels can be examined by chancing filer sizes. And research shows that spital filters improve performance.

2.2.2 Transfer Learning

Transfer learning is the method of using an already trained model on a similar task. This way, the training time for the new task is significantly decreased. In the case of image recognition, the earlier layers of the network are often learning to spot lesser complex properties like lines and shapes. It is then the later layers which contribute to the more task specific properties [29]. This means that a network which has been trained for a similar task, has knowledge which it can apply to the task at hand. By using transfer learning, the amount of data which is needed to get a working model is also decreased due to the pretrained model not having to start from scratch [30].

2.2.3 Semantic Segmentation

Semantic segmentation is a method for labeling and classifying images. The main idea behind it that because images are made up of pixels, each pixel can be labeled as what it is a part of in the greater image to be able to point out exactly where in an image its objects are [31]. As visible in Figure 2.4, every pixel of the image has been assigned to one of the predetermined classes.

Figure 2.4: Semantic segmentation example, from Cityscapes [9] used with permission.

What differs semantic segmentation from methods like instance segmentation and object detection is how it does not separate different instances of the same class. This means that even though a class object may appear several times, the different instances will all just get assigned the class's RGB value and not be classified as different objects. This is due to semantic segmentation only looking at an image as its pixels and not looking for objects.

2.3 Dataset

When training a CNN-model a data set is needed with a large amount of data. How much data is needed depends on different factors and it can not be specified to a specific number due to many variables. One factor is how different the classes that the model is going to separate are. For example if the pictures are just black and white, then the data set would not need that many samples compared to separating 10 different quite similar classes.

2.3.1 Data Augmentation

A common data augmentation technique is to incorporate the use of cropping, resizing, rotation and flipping in the data preproccessing [23]. This is particularly useful when dealing with small data sets, as the data set size can virtually be doubled with just the use of mirroring. By increasing the data set size, models can be trained on a greater number of images and improve it's generalization.

2.3.2 Class Imbalance

Class imbalance is the case of one or several classes being over-represented in the dataset. This meaning that the data is skewed. A skewed dataset can show reduced performance from the underrepresented classes as a model has less data to train on. Imbalanced datasets also increases the risk of low separability in the classes affecting the performance due to the risk of two similar classes not being represented enough

and therefore making it hard for a model to recognize the difference.

According to [32], a study was done which indicated that a balanced class distribution often resulted in better results but it was hard to determine which imbalance degree the class distribution would degrade the performance because of the other factors such as low sample size and separability.

When the datasets imbalance problem is fixed and the class distribution is balanced, the sample size has a significant role when evaluating how well a model performs [32]. This is quite obvious because if the data set is larger more information about the different classes is known which helps to reduce errors in the rarer classes. According to the Text the imbalance problem may not be a problem any more if the data set is large enough and the learning time is acceptable.

2.3.3 Splitting the data

When training a model, the dataset is usually split into three different parts. One training, one test and one validation set. The training set is what the model trains on and should therefore be made up of a great majority of the total data set. The validation set is used to validate the training to see how the model performs during training and it is from these results that the hyper-parameters can be tuned and the model can be optimized to get better performance. The test set is the data which the model is lastly tested on. The data from the test set has never been used by the model and is therefore a good set to get an unbiased evaluation of the models performance [33]. When splitting the data, it is important that each of the sets contains similar spread of data amongst the classes so that a class is neither under nor over represented in one of the sets.

2.4 Performance Metrics

Performance metrics are used to evaluate the results of a classification model. Evaluating the results is an important part to able to get a quantitative analysis of the result and be able to compare your result. There are many different performance metrics, for example Accuracy, Recall, F1 score and more. All suitable for different tasks. Measuring your results with a performance metric is also a way to be able to clearly see improvements and/or deteriorations. A result of a classification is often described with 4 terms; True positive, False positive, True negative and false negative or TP, FP, TN and FN for short. What these symbolize is visualized in Figure 2.5.

		Ground Truth	
		Positive	Negative
Predicted Truth	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Figure 2.5: Visualization of TP, FP, TN and FN.

With the help of these it is possible to see where the results are flawed and many performance metrics can be explained by using these.

2.4.1 Accuracy

Accuracy is a commonly used performance metrics. Accuracy calculates the percentage of the correct predicted classes of the number of total evaluated samples [8]. The formula for calculating accuracy is shown in Equation 2.5.

$$Accuracy = \frac{TP+TN}{(TP+TN+FP+FN)}. \tag{2.5}$$

2.4.2 Mean Intersection over Union

The mean intersection over union or mIoU for short is a performance metric commonly used for measuring the performance of semantic image segmentation [34]. The mIoU is calculated by taking the mean of the IoU from the different classes. The IoU is a value from 0-1 (0-100%) which is the result of dividing the area of overlap with the area of union for the predicted class. The area of overlap consists of all the pixels which the model predicts correct for a predicted class and the area of union is all the pixels which the model predicted to be correct and the pixels which are correct. This correlates to the following formula:

$$IoU = \frac{TP}{(TP+FP+FN)}. \tag{2.6}$$

and is visualized in Figure 2.6.

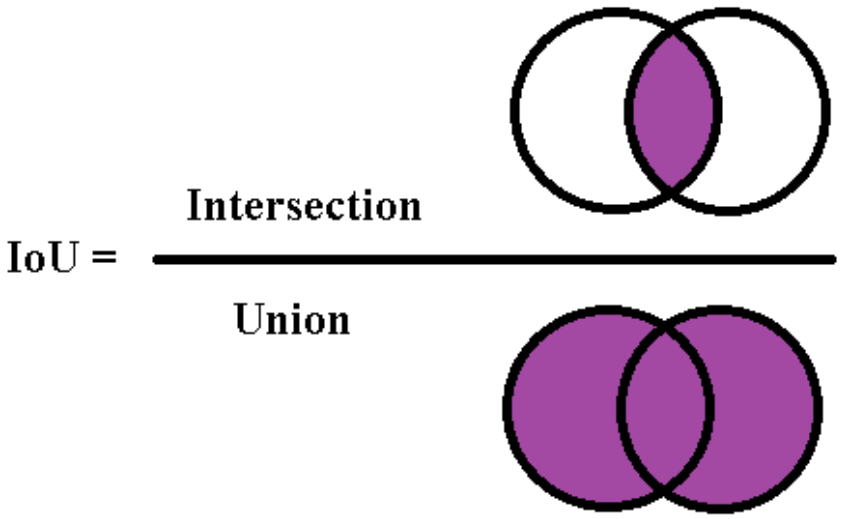


Figure 2.6: IoU visualized.

Why this way of measuring is suitable for semantic segmentation can be explained with the following example: Due to the goal of this project being to find the RSC and the RSC only being represented in the drivable area of the road, a big part of the images are made up of the background class. If the result was to be measured with the accuracy metric and the model was to predict that everything was background in an image that was 60% background, then it would still yield an accuracy score of 60% even though none of the other classes are represented in the result. With mIoU, the final score takes all of the classes into consideration and thus give a more accurate metric for how good the model is performing [35].

3

Methodology for RSC Segmentation

In this chapter the data-set creation process, model selection process and model optimization methodology are explained

3.1 Creating the Dataset

The dataset consists of images taken with a front facing camera that contains the following classes: dry, tracks snow, loose snow, packed snow, wet and background. To develop the dataset, images from the DENSE dataset is used [1] which contains images of road surfaces from across northern europe . On top of the DENSE dataset, images from the Mapillary Vistas Dataset [3] are also incorporated. The Mapillary Vistas Dataset contains images on road surfaces covering 6 continents which allows for choosing a variety of different climates and road surfaces. By making a dataset with images merged from the DENSE and Mapillary Vistas dataset it is possible to improve the balance and variety of the final dataset which provides a more broad selection of images to train the model with. A total of 682 images from a dataset of already labeled images from a previous project [2] are also incorporated in this project. These images are taken from the same datasets as mentioned before (DENSE, Mapillary Vistas) and labeled the same way this project intends. The merged dataset is made up of 1372 images. This results in about 250-300 images per RSC class. Equal parts are chosen to make the dataset balanced and by so, lowering the risk of overfitting to some classes which may appear more than others.

3.1.1 Semantic Labeling

The dataset is labeled using the annotation tool CVAT (Computer Vision Annotation Tool) [4]. In CVAT the visible classes for each image are be manually outlined using the polygon tool. Each pixel of the image have a corresponding class and the RGB value of that class. The classes and their corresponding RGB value are displayed in Table 3.1.

Table 3.1: RSC Classes, Labels and corresponding RGB values.

RSC	Label	RGB value	Color
Background	0	[0, 0, 0]	
Dry	1	[250, 50, 83]	
Loose snow	2	[170, 240, 209]	
Packed snow	3	[52, 209, 183]	
Tracks snow	4	[250, 250, 55]	
Wet	5	[184, 61, 245]	

All classes except Background are in the "drivable area" which shows an area where the vehicle is allowed to be driven. This includes the egolane, sidelines and connecting roads. This drivable area is segmented and labeled with the following classes: dry, loose snow, packed snow, tracks snow and wet (displayed in Table 3.1). When exporting from CVAT, all pixels that are not labeled, automatically have an RGB value of (0,0,0). This part of the images are classified as the background class. Figure 3.1 displays original images overlaid with segmentation masks representing all of the classes. Everything which is not colored belongs to the background class.

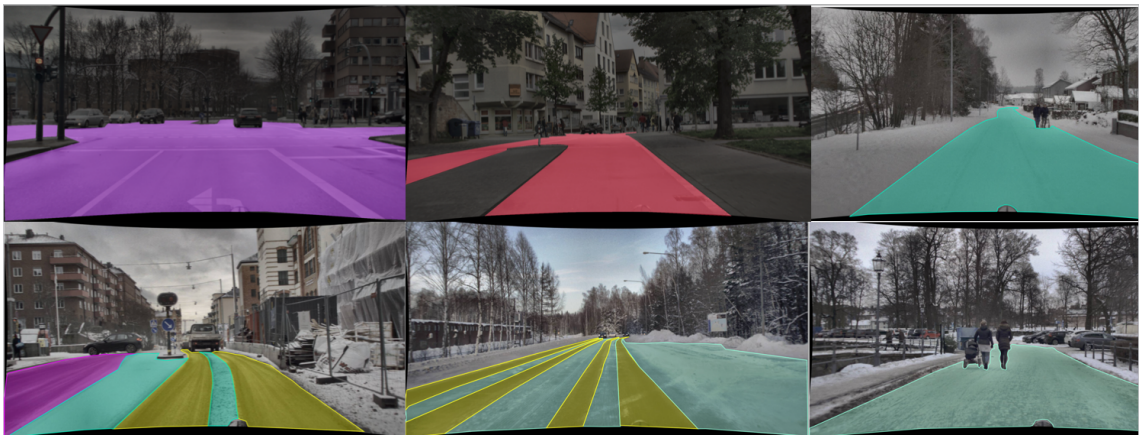


Figure 3.1: Ground truth labels showing the wet, dry, packed snow, tracks snow and loose snow classes. Images taken from DENSE dataset [1].

3.1.1.1 Tag annotation

The images which are manually labeled in this project are also tag annotated with the road type. Tag annotation is another annotation tool available in CVAT and allows you to tag each image with different classes. The classes which this projects images have been tagged with are asphalt, cobblestone and undefined. Undefined are for images where the road type is not visible.

3.1.2 Labeling Policy

A labeling policy is made to help assign labels to images so that there would not be any inconsistencies.

- If there is any uncertainties about a road being wet or dry, label it as wet.
- Only if the image contains clearly visible tracks in the snow where the road is visible should they be labeled as tracks snow.
- The label loose snow is for snow where the road underneath is partly visible.
- If the amount of snow is very minimal, so that it is barely visible then label it as wet.
- Images where the RSC is difficult to determine due to poor quality and/or visibility, are discarded.
- Only the part of the road which is legally drivable is annotated with the RSC classes. Parts such as pavement is a part of the background class.
- Only clearly visible parts of the road are annotated, this means that road which is visible through a fence or similar is not annotated.
- Shadows underneath cars where the RSC is not visible and is therefore labeled as background.
- If a picture is too dark with a small amount of brightness, discard it.
- When tag annotating the road material, everything that is not asphalt or cobblestone is tagged as undefined.
- If the road material is not clearly visible due to snow or other RSCs blocking vision, the image is tagged with undefined.

3.2 Models used in RSC segmentation

There are already existing models trained for semantic segmentation on road and street images. These models focus on identifying several things, for example traffic signs, cars, people and road surface. Even though this project is only interested in the road surface condition, these models are still useful. Using transfer learning, the starting point of the network allows for better immediate performance. Starting out with already trained weights is more beneficial than starting with randomly initialed weights. The already trained weights makes it so that less data is needed for training as the network is already trained on different data. Even though the data which the network is trained on is not meant for the same purpose as this project, many things which the network is taught are still applicable to our purpose. The networks used for this project are for example trained on finding the drivable area which is where the RSC classes are found and being able to find the drivable area is beneficial for being able to separate the background from the RSC classes. More on how this works is explained in 2.2.2.

The github repository MMsegmentation [10] is used for this project. The repository contains several implementations of segmentation models with different architectures. This allowed us to use the same repository to compare different models and architectures. Furthermore, the repository is well documented regarding installation, training and evaluation procedures.

From MMsegmentation, three models are selected. Based on the benchmark figures, see Table 3.2, the Pyramid Scheme Parsing network (PSPnet)[5], Object-Contextual Representations network (OCRnet)[6] and Fully Convolutional Network (FCN)[7] are selected for evaluation. The PSPnet and OCRnet models are built on different backbones while the PSPnet and FCN are built on the same. A backbone refers to an already existing feature extracting network, that has been trained and tested with good results on many other tasks [36]. PSPnet and FCN are built on Residual Neural Network (ResNet) and OCRnet is built on High-Resolution Network (HRnet).

Table 3.2: mIoU for the selected models from MMsegmentation benchmark[10], using 40,000 iterations and evaluated on the Cityscapes[11] dataset.

	PSPnet	OCRnet	FCN
mIoU	77.85	80.58	72.25

3.2.1 Evaluating the models

The three models are trained for 40,000 iterations, with evaluation on the validation set every 4000 iterations. The models are first trained with cross entropy loss, a learning rate of 0.01 and SGD with momentum as optimzer and the momentum is 0.9. To compensate for a possible imbalance in the dataset the three models are also trained with weighted cross entropy loss before selecting one to move forward with. mIoU on evaluation set is analyzed to decide whether the model has converged or not.

3.3 Optimization of PSPnet

After evaluation, the model and setup with the best mIoU is selected for further fine-tuning where learning rate is adjusted and evaluated. In the first set of experiments, the learning rate (Lr) is multiplied by 10 in the first test, and in the second test divided by 10. This results in a set of experiments where the model is trained using a Lr of 0.1, the original 0.01 and 0.001. The best performing Lr is selected and a similar set of experiments are conducted, but instead with the learning rate both multiplied by 2 and divided by 2.

To ensure that the model is robust and generalizes outside of just the original training and validation split, another training and validation split is created and then utilized for training and evaluation. This is a simplified version of k-fold cross validation. Instead of splitting the dataset into a training and a validation set several times, two different splits are utilized. This is done to evaluate the robustness and generalization of the model.

In order to analyze the convergence of the model's parameters, the model is trained for 80,000 iterations using the best performing learning rate. Model weights are evaluated and saved every 4000 iterations, that way the best weights can be selected utilizing the early stopping strategy.

4
Results and Discussion

In this chapter the result of the project are presented and discussed. This includes the dataset and the performance of the different models with the dataset.

4.1 Dataset

The final dataset consisting of 1369 images are display split up into a train set of 1169 images and a validation set of 200 images in Table 4.1 and 4.2. The tables also shows how many images each of the classes appear in and how many the total number of pixels in the dataset the classes make up.

Table 4.1: Presentation of the training dataset.

	Pixels		Images	
Class	Amount (millions)	Percentage	Amount	Percentage
Background	2900	75.9%	1169	100%
Dry	475	12.4%	377	32.2%
Loose snow	75.6	0.2%	321	27.4%
Packed snow	113	0.3%	282	24.1%
Tracks snow	34.7	0.1%	237	20.2%
Wet	221	5.8%	446	38.1%

Table 4.2: Presentation of the validation set before.

	Pixels		Images	
Class	Amount (millions)	Percentage	Amount	Percentage
Background	660	74.9%	200	100%
Dry	132	15.05%	56	28%
Loose snow	8.21	0.92%	29	14.5%
Packed snow	20.8	2.36%	42	21%
Tracks snow	4.13	0.47%	32	16%
Wet	55.7	6.3%	97	48.5%

4.1.1 Reflection about Dataset

The dataset is labeled with the 5 RSC classes as seen in Table 4.1 and 4.2. However the three snow classes are severely underrepresented in terms of pixels compared

to the other classes. Therefore, the decision to merge some of the classes is made. In order to tackle the imbalance in the distribution of the data. The loose snow and packed snow classes are merged into one class called snow and the tracks snow class are merged into the wet class. Merging wet and tracks snow make sense as the tracks snow class is only differentiated itself from wet by its location and sometimes minimal amounts of snow residues in the tracks. This new merged dataset consists of 1369 images, split into train, validation and test sets is presented in Table 4.3, 4.4 and 4.5. The new datasets label ids and RGB values are also displayed in Table 4.6. With the new dataset, some information is lost as it consists of fewer classes but the possibility of better performance is something which is valued more because if the model has difficulties finding the less represented classes, then they do not add much value.

Table 4.3: Presentation of dataset for training, consisting of 969 images.

	Pixels		Images	
Class	Amount (millions)	Percentage	Amount	Percentage
Background	2103.3	75.28%	969	100%
Dry	328.6	11.76%	305	31.5%
Snow	137.7	4.929%	433	44.7%
Wet	224.2	8.026%	562	58%

Table 4.4: Presentation of dataset for evaluation, consisting of 200 images.

	Pixels		Images	
Class	Amount (millions)	Percentage	Amount	Percentage
Background	704.4	76.0%	200	100%
Dry	135.8	14.6%	61	30.5%
Snow	41.5	4.48%	97	48.5%
Wet	45.7	4.93%	100	50%

Table 4.5: Presentation of dataset for testing, consisting of 200 images.

	Pixels		Images	
Class	Amount (millions)	Percentage	Amount	Percentage
Background	747.0	76.79%	200	100%
Dry	145.9	15.0%	67	33.5%
Snow	35.9	3.68%	75	37.5%
Wet	43.9	4.52%	118	59%

Table 4.6: RSC Classes, Labels and corresponding RGB values for the merged dataset.

RSC	Label	RGB value	Color
Background	0	[0, 0, 0]	
Dry	1	[250, 50, 83]	
Snow	2	[170, 240, 209]	
Wet	3	[184, 61, 245]	

4.1.2 Balance

The final merged dataset is an improvement to the non merged from a balance perspective. However the imbalance problem still remains. The background class being represented far more than the others is something which is inevitable but the dry, snow and wet classes would benefit from a more even spread as the class balance would improve.

4.1.3 Labeling

The dataset is labeled using CVAT [4] and by following the labeling policy mentioned in 3.1.2. It is merged with the dataset from [2] that was made before this project started. It is hard to guarantee that the images are labeled correctly according to the labeling policy every time. Also when merging two datasets its hard to guarantee that the road surface condition have been labeled similarly. For those reasons, some inconsistencies can confuse the model when training. This could lead to decreased performance by the models. It is especially hard to label images when there is an uncertainty whether it is wet or dry for darker images that are taken later in the the day/night.

4.2 Performance of Models

Three different models are used: PSPnet, OCRnet and FCN. They are trained for 40,000 iterations.

4.2.1 Models trained with cross-entropy loss and weighted cross-entropy loss

The models are trained with a learning rate of 0.01 and SGD with momentum (momentum=0.9) as optimizer.

Table 4.7 shows how FCN has the best mIoU after 40,000 iterations with cross-entropy loss. When looking at the final IoU for the different classes, all of the models show similar performance, most similar is PSPnet and FCN as OCR performs better for the Snow class but worse for the dry and wet class. The graph in Figure 4.1 also shows how the three models performances are not very stable.

From the benchmarks in the MMsegmentation repository [10] displayed in Table 3.2. FCN has the worst performance on the Cityscapes dataset [11]. Therefore all three models are also evaluated when trained with weighted cross-entropy loss and the same parameters as the initial training.

Training done with weighted cross-entropy loss showed slightly worse mIoU for all the models as well as decreased IoU for all the classes. Because of the remaining imbalance in the dataset, the hope was that the weighted cross-entropy loss would result in a more evenly spread IoU for the three classes but this is not the case. However, the graph in Figure 4.2 shows how the performance of PSPnet and FCN being more stable which points towards weighted cross-entropy loss making the models more regularized and robust. OCRnet on the other hand does show the highest peak but because of the instability of the results, this model might not be as fitting for the task as PSPnet and FCN.

When trained using weighted cross-entropy loss, PSPnet performs the best. The graph in Figure 4.5 also suggests that PSPnet still has a growing curve and that its potential performance has not yet been reached. Therefore, the PSPnet model with weighted cross-entropy loss is chosen for further optimization.

Table 4.7: resulting mIoU (%) of the three models for 40,000 iterations, with cross-entropy loss and weighted cross-entropy loss.

	Cross-entropy			Weighted cross-entropy		
Iterations	PSPnet	OCRnet	FCN	PSPnet	OCRnet	FCN
4000	70.60	70.82	63.39	64.80	70.92	59.88
8000	65.76	67.75	73.33	64.38	69.90	63.97
12000	69.04	69.32	75.50	67.30	58.54	68.64
16000	73.98	70.42	70.16	64.90	71.49	69.94
20000	76.51	74.40	72.32	72.98	73.99	70.60
24000	75.85	74.01	77.09	75.73	79.02	66.45
28000	71.72	77.45	74.90	74.96	75.50	73.11
32000	74.73	71.77	77.17	76.21	73.56	75.54
36000	76.61	73.47	74.09	78.11	74.53	76.20
40000	78.25	77.16	78.71	78.08	75.52	77.08

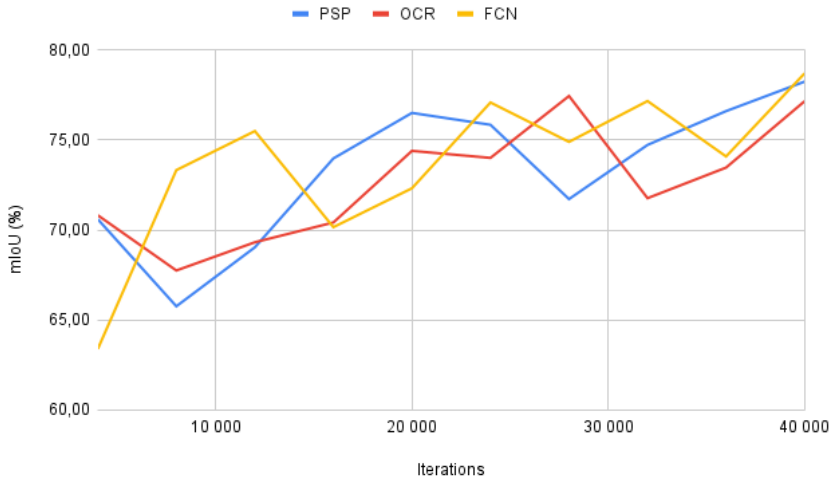


Figure 4.1: mIoU (%) of the models using cross entropy loss.

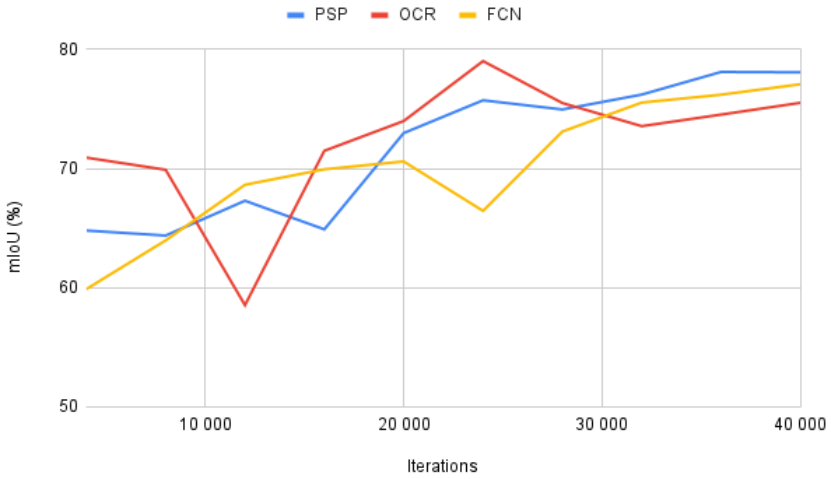


Figure 4.2: mIoU (%) of the models, with weighted cross entropy loss.

Table 4.8: Final IoU (%) for each class using cross-entropy loss and weighted cross-entropy loss for the 3 models.

	Cross-entropy			Weighted cross-entropy		
Class	PSPnet	OCRnet	FCN	PSPnet	OCRnet	FCN
Dry	81.12	76.69	80.92	81.29	73.67	79.95
Snow	69.24	74.97	69.82	66.46	74.38	67.30
Wet	67.34	61.58	68.52	69.70	58.87	66.59

Confusion matrices of the models provide normalized accuracy values for the four classes. They show the extent of misclassification for the predicted labels on the test set. It can be observed that the biggest three groups of misclassification are *Dry* when actual wet and *Background, Wet* when actual Snow. When actual Snow, OCR-net trained with cross-entropy loss doesn't demonstrate as high misclassification for predicted *Background.*

Confusion matrices of the models trained using weighted cross entropy loss doesn't show the same misclassification with *Background* when actual Snow as before. Both the Resnet models FCN and PSPnet show misclassification with *Wet* when actual Snow and with *Dry* when actual Wet. FCN also expresses predicted *Dry* when actual Snow. This is not easily argued because the classes *Dry* and *Snow* rarely occur in the same image or in the same drivable area. *Wet* and *Snow* commonly occur in the same driveable area as the snow melts or creates tracks.

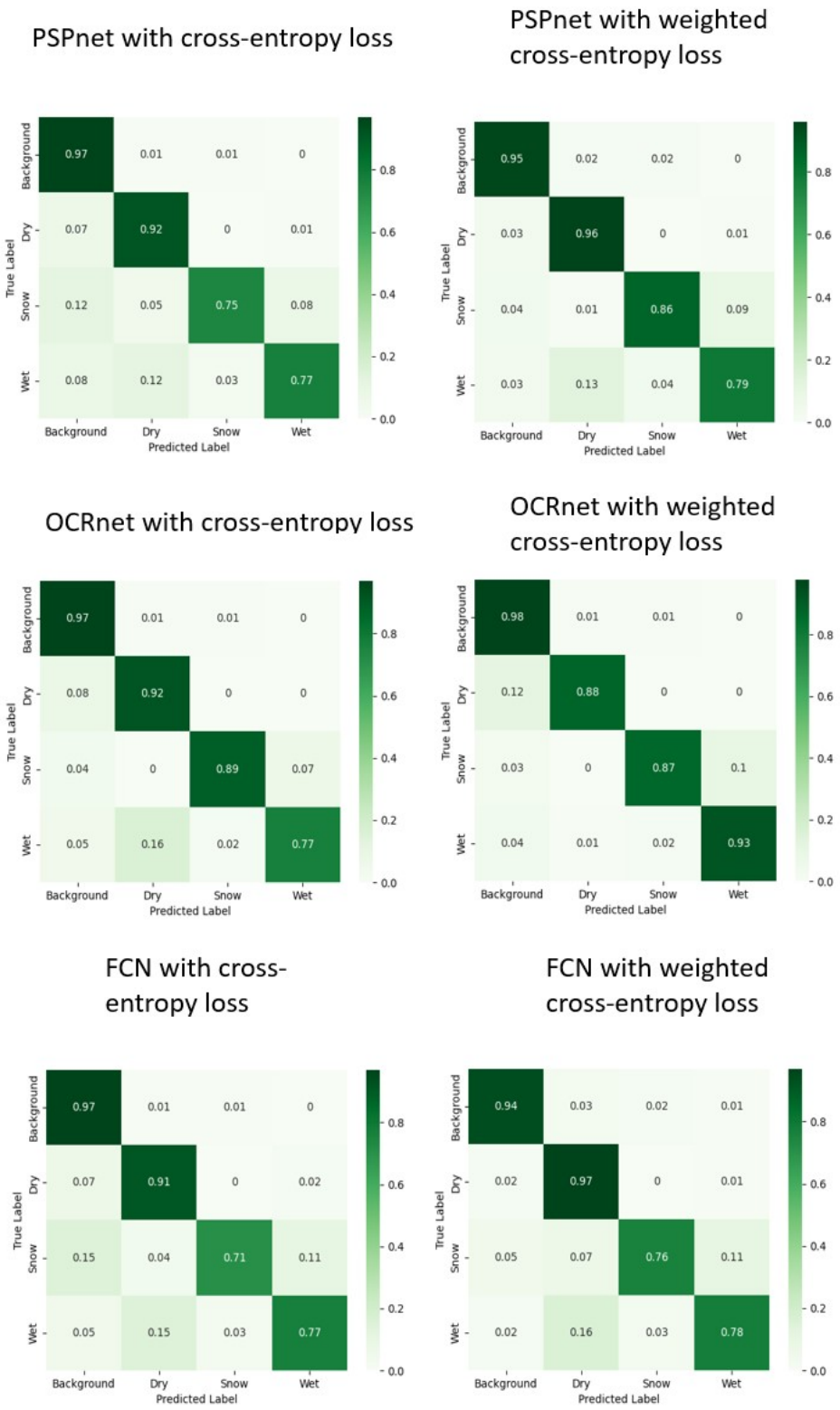


Figure 4.3: Confusion matrices for all three models using cross-entropy and weighted cross-entropy loss.

4.2.2 Optimizing PSPnet: Different Learning Rates

PSPnet is selected for further optimization and the learning rate is chosen to be, in separate experiments, divided by 10 in one case and multiplied by 10 in another. The parameters of the learning rate optimizer are not changed.

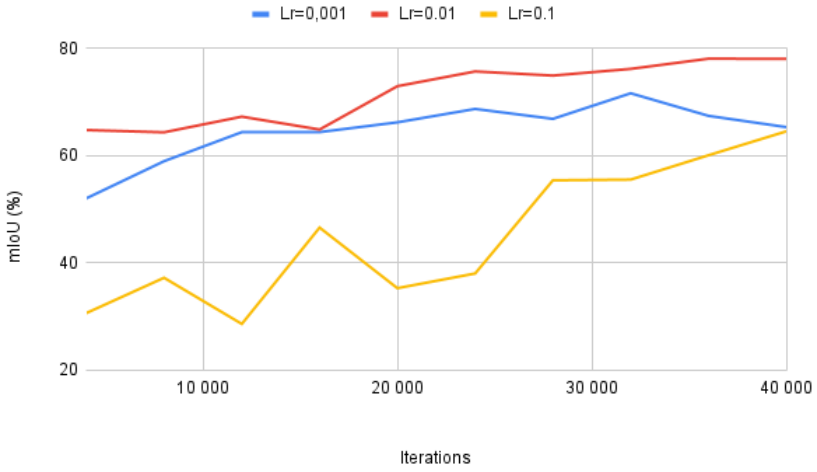


Figure 4.4: mIoU (%) of PSPnet, with learning rates (Lr) of 0.001, 0.01 and 0.1.

Table 4.9: mIoU (%) for PSPnet with learning rates (Lr) of 0.001, 0.01 and 0.1.

	Learning rate		
Iterations	0.001	0.01	0.1
4000	52.06	64.80	30.66
8000	58.99	64.38	37.25
12000	64.41	67.30	28.64
16000	64.42	64.90	46.61
20000	66.24	72.98	35.31
24000	68.74	75.73	38.05
28000	66.89	74.96	55.43
32000	71.66	76.21	55.56
36000	67.45	78.11	60.10
40000	65.35	78.08	64.56

Observing the results from both Table 4.9 and Figure 4.4 it is evident that PSPnet performs best utilizing a learning rate of 0.01. Therefore, in separate experiments, the learning rate is chosen to be divided by 2 in one case and multiplied by 2 in another. This is then displayed in Figure 4.5 and Table 4.10 and shows how a learning rate of 0.01 still is superior.

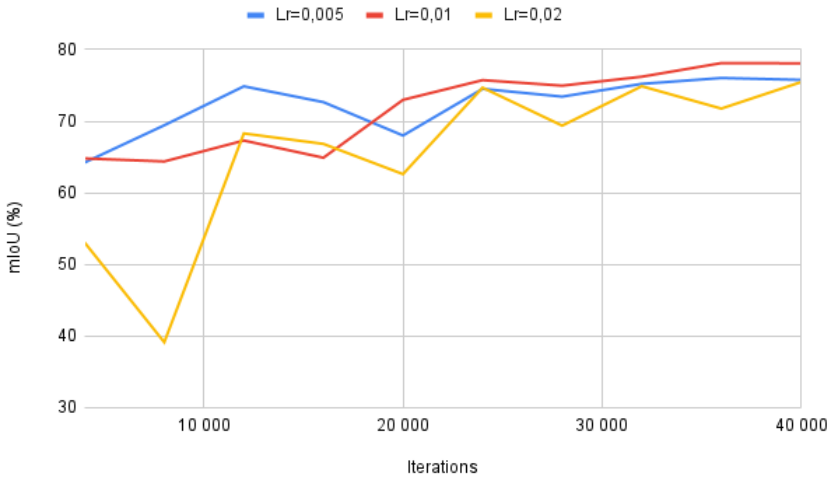


Figure 4.5: mIoU (%) of PSPnet, with learning rates (Lr) of 0.005, 0.01 and 0.02.

Table 4.10: mIoU (%) for PSPnet, with learning rates of 0.005, 0.01 and 0.02.

	Learning rate		
Iterations	0.005	0.01	0.02
4000	64.23	64.80	53.05
8000	69.43	64.38	39.14
12000	74.87	67.30	68.29
16000	72.66	64.90	66.83
20000	68.00	72.98	62.61
24000	74.51	75.73	74.68
28000	73.43	74.96	69.38
32000	75.21	76.21	74.88
36000	76.03	78.11	71.76
40000	75.77	78.08	75.45

4.2.3 Assessing Performance through Cross-validation for PSPnet

In order to ensure the performance of the model we utilized cross-validation. The training set and validation is divided into another split, with the same size as previously and the same test set as previously. The new split is displayed in Table 4.11 and 4.12. The cross validation is trained with a learning rate of 0.01, SGD with momentum of 0.9 and weighted cross-entropy loss with updated weights that match the new training set. The cross validation model is further tested with the test set.

Table 4.11: Training set for cross validation consisting of 969 images.

	Pixels		Images	
Class	Amount (millions)	Percentage	Amount	Percentage
Background	2311.5	75.56%	969	100%
Dry	374.1	12.23%	290	29.9%
Snow	140.1	4.55%	428	44.2%
Wet	234.2	7.66%	560	57.8%

Table 4.12: Validation set for cross validation containing of 200 images.

	Pixels		Images	
Class	Amount (millions)	Percentage	Amount	Percentage
Background	499.13	75.14%	200	100%
Dry	90.23	13.58%	76	38%
Snow	39.14	5.89%	102	51%
Wet	35.77	5.39%	102	51%

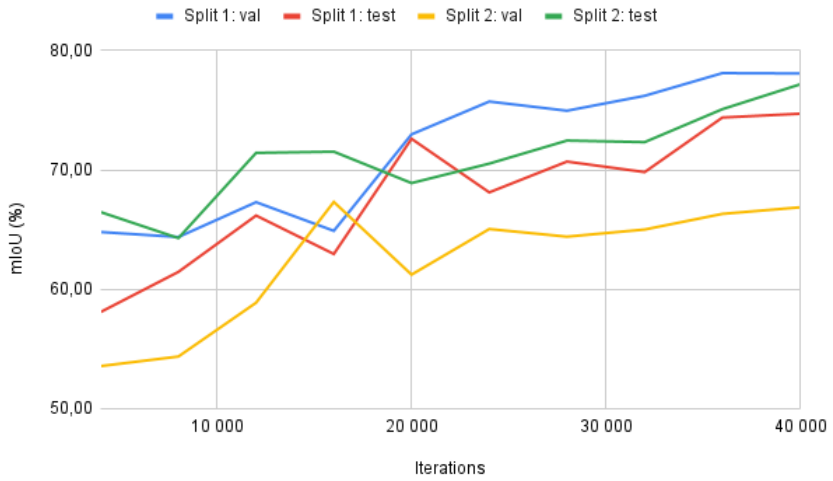


Figure 4.6: mIoU (%) of PSPnet with cross validation.

The model trained on the cross validation split (split 2) and the model trained on the original split (split 1), both are evaluated on their respective validation set and

also with the test set, the mIoU of the evaluations is shown in Figure 4.6. The final values of the mIoU are shown in Table 4.13. The model performs worse on the new validation set, but reaches a similar mIoU when evaluating with the test set.

Table 4.13: Final mIoU (%) of the three evaluations.

Split 1: validation	Split 1: test	Split 2: validation	Split 2: test
78.08	74.71	66.87	77.17

4.2.4 Optimizing PSPnet: Extended Training

The model is finally trained for 80,000 iterations to analyze convergence.

Figure 4.7: mIoU (%) of PSPnet with cross validation.

The mIoU of the model is shown in Figure 4.7. On the validation set the model performs best at 76,000 iterations, with an mIoU of 80.19%. The model performs slightly worse on the test set, it reaches an mIoU of 75.90% at 52,000 iterations and it's second highest mIoU of 75.60% is at 76,000 iterations. Moving forward with the qualitative analysis the model from 76,000 iterations are used, as it performs well at both the validation and test set.

4.3 Qualitative Analysis

Figure 4.7 and 4.8 showcases the output of 14 different images using inference with the best performing model where Figure 4.7 displays where the model performs good and Figure 4.8 displays where the model performs poorly.

Class	Color
Dry	
Snow	
Wet	

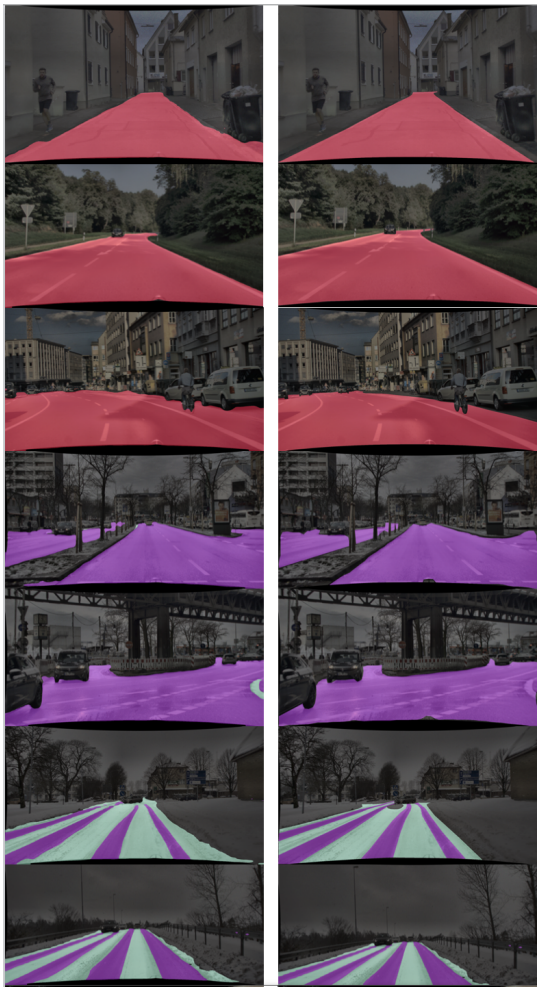


Figure 4.8: Good predictions (left) and the ground truth (right), images taken from the Dense dataset [1].

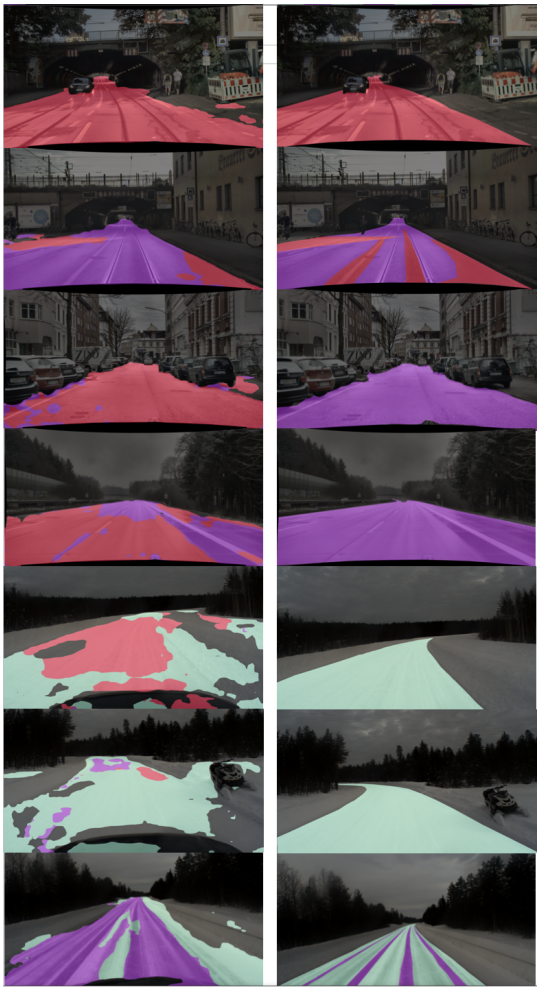


Figure 4.9: Bad predictions (left) and the ground truth (right), images taken from the Dense dataset [1].

Overall the the performance of the model is good with an mIoU of 80.19%. For the dry class, the model performs the best with good results. One problem with this class is that the model has some difficulties with finding where the road ends and

thus classifying some of the background class as dry. In some cases the model has problems distinguishing dry with wet.

For the wet class the model manage to find the the road very well and in the best cases the outcome is almost perfect but the main problem for this class is that it is hard for the model to tell the difference between dry and wet and in some cases snow as well.

For roads which are completely covered in snow and the only thing differentiating background and the drivable area is the height of the snow or similar, the model can have issues finding the exact outlines of the drivable area and therefore classifying some background as the snow class.

When there are tracks in the snow, if there are several tracks next to each other and the snow in between is not clearly defined the model has a hard time differentiating between wet and snow. This could be due to the tracks snow class being merged with the wet class where some tracks could have a slim amount of snow inside of them and thus making the model have a harder time differentiating these in some scenarios.

Roads where different road types appear and where lines and object appear on the road affects the performance of the model. During these circumstances, the model has a hard time knowing what is background and what is not.

4.4 Ethical aspects

If the images taken by the camera mounted on the cars using this RSC system gets saved somewhere, it might intrude on personal privacy for the people captured in the footage. The capturing of video and images must follow the laws and regulations from the country where it is operating.

This model or a similar model might in the future be incorporated into vehicles, either as driving assisting function or part of a self-driving system. A problem in this case is when an accident happens. If the model did not detect a slippery surface, there is a risk that an accident occurs. This raises concerns about how reliable a system that relies on this type of model is. An mIoU of 80% means that a lot of pixels are classified wrong and there is a significant risk that slippery spots can be missed. Therefore it is important to further improve the model and also proceed with caution when implementing it into vehicles.

5
Conclusion and Future work

5.1 Conclusion

This thesis presents the development of a dataset and training of a model for classifying road surface condition (RSC) in a high resolution. The dataset is developed by manually labeling 687 images and combining these with 682 already labelled images from a previous project at Chalmers University [2].

Devloping a model for classifying RSC in a high resolution is achieved by training and evaluating convolutional neural networks suitable for semantic segmentation. Three different models are selected and trained with different loss functions: PSPnet [5], OCRnet [6] and FCN [7]. The weighted cross entropy loss proved to produce more stable models and improve regularization. The best performing model, PSPnet, is then selected for further optimization. The model is evaluated with various learning rates and the best performing learning rate, 0.01, is selected for cross validation. Additionally, this model is trained for an extended duration and achieved an mIoU of 80.19% at 76,000 iterations. The model performed well when classifying the different RSCs for most images but sometimes misclassified snow as wet in some cases and wet as dry for images with poor vision. Images showing more uncommon scenes made it more difficult for the model to classify what belonged to the background and what was drivable area.

Both autonomous vehicles and vehicles with drivers could benefit from using a semantic segmentation road surface condition classifier. For both vehicles it helps with decision making such as being notified of low friction on the road ahead.

5.2 Future Work

The future work sections presents three different approaches for further improvement of this project.

5.2.1 Improvement of the dataset

To further improve performance, the dataset can be expanded to increase the sample size for both the training data and for the evaluation/testing data by adding annotated images to the dataset. Expanding the training set would lead to bet-

ter performance and regularization of the model, as it learns from bigger and more diverse data.

5.2.2 RSC Segmentation with Roadtype classification

As this projects dataset is already annotated with the road type (asphalt, cobblestone or undefiend), a new head of one of the selected models could be made to classify the road type. This head would make use of multi-task learning and have an image classification output which outputs the road type of the image at the same time as the other heads output the predicted segmentation mask.

5.2.3 Further fine-tuning of the model

To improve model performance, there may be great benefit using more powerful hardware, capable of training the network at a bigger batch size and a greater number of iterations. This is especially true if the dataset is to be extended and diversified. A greater dataset will need further training to converge effectively.

Additional hyperparameters can be tuned for the network and optimizer, such as weight decay and momentum. A different learning rate optimizer may also yield better result as it dictates how big of an update the model's parameters get during training. A different optimizer may fit this model better and more effectively find a superior local minimum.

By further examining different loss functions, it would be possible to determine which one is best suited for this task. A better suited loss function could improve model performance by better alignment with task objectives or encouraging desired model behavior.

Bibliography

[1] Bijelic M, Gruber T, Mannan F, Kraus F, Ritter W, Dietmayer K, and Heide F. Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2020.

[2] Bergman A, De Geer C, Lindell T, and Lundgren F. Create a data set for road surface condition segmentations. *MPSYS, Chalmers*, January 2023.

[3] Neuhold G, Ollmann T, Rota Bulò S, and Kontschieder P. Mapillary vistas dataset for semantic understanding of street scenes. In *In International Conference on Computer Vision (ICCV)*, 2017.

[4] CVAT.ai Corporation. Computer vision annotation tool (cvat), September 2022. If you use this software, please cite it using the metadata from this file.

[5] Zhao H, Shi J, Qi X, Wang X, and Jia J. Pyramid scene parsing network. *CoRR*, abs/1612.01105, 2016.

[6] Yuan Y, Chen X, and Wang J. Object-contextual representations for semantic segmentation. *CoRR*, abs/1909.11065, 2019.

[7] Long J, Shelhamer E, and Darrell T. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1411.4038, 2014.

[8] Zhang J. Humaidi A.J. Alzubaidi, L. et al. Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. *Journal of Big Data*, 8(53), Mar 2021. doi: 0.1186/s40537-021-00444-8.

[9] Cityscapes. Examples. https://www.cityscapes-dataset.com/examples/.

[10] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020. (Accessed on 2023-05-10).

[11] Cordts M, Omran M, Ramos S, Rehfeld T, Enzweiler M, Benenson R, Franke U, Roth S, and Schiele B. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Apr 2016.

[12] Federal Highway administration U.S. department of transportation. How do weather events impact roads? https://ops.fhwa.dot.gov/weather/q1_roadimpact.htm, 2023. (Accessed on 19-04-2023).

[13] Kordani A, Rahmani O, Nasiri A, and Boroomandrad S. Effect of adverse weather conditions on vehicle braking distance of highways. *Civil Engineering Journal*, 4:46, Feb 2018.

[14] Busch A, Fink D, Laves M, Ziaukas Z, Wielitzka M, and Ortmaier T. Classification of road surface and weather-related condition using deep convolutional neural networks. In Klomp M, Bruzelius F, J Nielsen, and Hillemyr A, editors, *Advances in Dynamics of Vehicles on Roads and Tracks*, pages 1042–1051, Cham, 2020. Springer International Publishing.

[15] Novikov A, Novikov I, and Shevtsova A. Study of the impact of type and condition of the road surface on parameters of signalized intersection. *Transportation Research Procedia*, 36:548–555, Jan 2018.

[16] Ognjanovski G. Everything you need to know about neural networks and backpropagation — machine learning easy and fun. https://towardsdatascience.com/everything-you-need-to-know-about-neural-networks-and-backpropagation\-machine-learning-made-easy-e5285bc2be3a, Jan 2019. Towards data science, (Accessed on 2023-04-14).

[17] McNeill G Anderson D. Artificial neural networks technology. Technical Report F30602-89-C-0082, Kaman Sciences Corporation, Utica, New York, USA, Aug 1992. Availiable https://csiac.org/wp-content/uploads/2021/06/Artificial-Neural-Networks-Technology-SOAR.pdf.

[18] Baheti P. Activation functions in neural networks [12 types amp; use cases]. https://www.v7labs.com/blog/neural-networks-activation-functions#h1, Mar 2023. (Accessed on 2023-04-22).

[19] Brownlee J. A gentle introduction to the rectified linear unit (relu). https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/, Aug 2020. MachineLearningMastery, (Accessed on 2023-04-22).

[20] Zhou Z, Huang H, and Fang B. Application of weighted cross-entropy loss function in intrusion detection. *Journal of Computer and Communications*, 9(11):1–21, Nov 2021. doi: 10.4236/jcc.2021.911001.

[21] Dabbura I. Gradient descent algorithm and its variants. https://towardsdatascience.com/gradient-descent-algorithm-and-its-variants-10f652806a3, Dec 2017. Towards data science, (Accessed on 2023-05-05).

[22] Musstafa. Optimizers in deep learning. https://medium.com/mlearning-ai/optimizers-in-deep-learning-7bf81fed78a0, May 2021. Medium, (Accessed on 2023-05-05).

[23] Kukačka. J, Golkov. V, and Cremers. D. Regularization for deep learning: A taxonomy. *arXiv preprint arXiv:1710.10686*, 2017.

[24] Brownlee J. A gentle introduction to k-fold cross-validation. https://machinelearningmastery.com/k-fold-cross-validation/, Aug 2020. MachineLearningMastery, (Accessed on 2023-05-05).

[25] Kuen J Ma L Shahroudy A Shuai B Liu T Wang X Wang G Cai J Chen T Gu J, Wang Z. Recent advances in convolutional neural networks. *Pattern Recognition*, 77, May 2018. doi: 10.1016/j.patcog.2017.10.013.

[26] O'Shea K and Nash R. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, Nov 2015.

[27] Nishio M. Do R.K.G. et al Yamashita, R. Convolutional neural networks: an overview and application in radiology. *Insights Imaging*, 9, Jun 2018. doi: https://doi.org/10.1007/s13244-018-0639-9.

[28] Sohail A. Zahoora U. et al. Khan, A. A survey of the recent architectures of deep convolutional neural networks. *Artif Intell Rev*, 0(53), Apr 2020. doi: https://doi.org/10.1007/s10462-020-09825-6.

[29] Pranshu S. Understanding transfer learning for deep learning. https://www.analyticsvidhya.com/blog/2021/10/understanding-transfer-learning-for-deep-learning/, Oct 2021. Analytics Vidhya, (Accessed on 2023-04-23).

[30] Brownlee J. A gentle introduction to transfer learning for deep learning. https://machinelearningmastery.com/transfer-learning-for-deep-learning/, Sep 2019. MachineLearningMastery, (Accessed on 2023-04-23).

[31] Matcha A. C. N. A 2021 guide to semantic segmentation. Available at https://nanonets.com/blog/semantic-image-segmentation-2020/, 2021. (Accessed on 24 April 2023).

[32] Sun Y, Wong A, and Kamel M. Classification of imbalanced data: a review. *International Journal of Pattern Recognition and Artificial Intelligence*, 23, 11 2011. doi: 10.1142/S0218001409007326.

[33] Agrawal S. How to split data into three sets (train, validation, and test) and why? https://towardsdatascience.com/how-to-split-data-into-three-sets-train-validation-and-test-and-why-e50d22d3e54c, May 2021. Towards data science, (Accessed on 2023-05-05).

[34] Chollet F et al. Image segmentation metrics. https://keras.io/api/metrics/segmentation_metrics/, 2015.

[35] Rosebrock A. Intersection over union (iou) for object detection. https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/, Dec 2022. PyImageSearch, (Accessed on 2023-04-12).

[36] Elharrouss O, Akbari Y, Almaadeed N, and Al-Maadeed S. Backbones-review: Feature extraction networks for deep learning and deep reinforcement learning approaches. *arXiv preprint arXiv:2206.08016*, 2022.

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY