ENHANCING ROAD DAMAGE SURVEY WITH DEEP LEARNING

CLASSIFYING GRAVEL ROAD DAMAGE USING RESNET50 AND TRANSFER LEARNING

Tommy Nielsen

EC Utbildning

Graduation Thesis

2024-02

Abstract

The objective of this thesis is to explore the integration of machine learning into the current manual process of gravel road damage surveying conducted by Ramboll RST Finland. The project aims to develop a model that can classify road damage with a high degree of accuracy. This study leverages data provided by Ramboll, consisting of annotated images of gravel roads, to train and test a deep learning model based on ResNet50 architecture and transfer learning techniques. The thesis examines the effectiveness of this model in accurately identifying road damage, aiming to achieve at least 85% accuracy in classification. The research addresses challenges associated with adapting the existing survey data for machine learning and explores the potential of this technology to enhance road maintenance strategies.

The study culminates in the development of two models through different data selection processes: a robust model designed for general applicability with a modest performance accuracy of 75.2%, and a highly specialized model achieving an impressive accuracy of 97.5% under specific conditions. These models offer valuable insights into the strengths and weaknesses within the data, guiding future enhancements by identifying key areas for development.

Key words: Gravel Road Damage Classification, ResNet50, Deep Learning, Image Classification, Transfer Learning, Data Preparation

Acknowledgments

I would like to express my sincere gratitude to Ramboll for the invaluable opportunity, resources, and support they provided, crucial for the realization of this project as the subject of my thesis. Special thanks to the project manager, Harri Ahola, for his guidance and assistance.

I am also deeply thankful to my classmates and project collaborators, Nicklas Mattisson and Márk Mészáros for their tenacity, insights, and passion throughout this journey.

I would also like to extend my deepest appreciation to my family and friends for their unwavering support and encouragement. A special thanks to my brother, whose support has been a cornerstone of my personal well-being and success throughout the duration of this project and my academic journey.

Lastly, would like to extend my heartfelt thanks to Antonio Prgomet. His role extends beyond supervising this thesis; as a mentor and educator in many of my Data Science courses, he has been a guiding star in my journey into the realm of Data Science. His support and wisdom have been invaluable in shaping my understanding and passion for this field and life in general.

Contents

1 Introduction

Whether you are traveling for personal or commercial purposes, a key factor in your safety is the condition of the road. Imagine you are in one of the safest vehicles available, but the road ahead is rough, full of big rocks and large holes in it. Even if you adapt your speed and driving style to avoid crashing or even getting a flat tire, the road conditions will not only impact the time and comfort of your journey but also likely increase the wear and tear on the vehicle, putting a strain on its otherwise good safety features. This will most likely result in higher upkeep costs or the need for earlier replacement of your otherwise so safe vehicle. This is why maintaining the roads on which we intend to travel safely is relevant to study.

Due to their low building and maintenance cost, gravel roads are a preferred alternative in many rural areas and their importance can be described as follows:

> *Gravel roads are an essential part of the "blood vascular system" of the transportation infrastructure. They provide access to many rural communities, and they act as a transportation route for products to markets. For instance, in most cases the beginning of the transportation routes for products of the farming, forest and aggregate industries has gravel surfaces. In addition, gravel and forest roads have a critical role in forest fire management as well as defence training. And finally gravel roads have a great role in recreational, social and tourism activities. (ROADEX, 2023)*

However, a major drawback of gravel roads is that they are not as durable as paved roads. The deterioration rate of the gravel roads is considerably affected by rain and snowfall. Therefore, it is crucial to survey road conditions during seasons with heavy rain and snowfall, such as fall and winter in the Nordic countries and in the spring. So that the correct maintenance can be planned in a timely manner.

For this purpose, Ramboll RST Finland has been tasked with surveying the gravel road surface conditions across a large portion of the Finnish gravel road network. The survey is conducted biannually, once in the fall and again in the spring. Currently, the assessment is performed manually, with each section (every 20 meters) being documented by taking digital images. To enhance this process, Ramboll is embarking on a project to explore the potential of implementing Machine Learning for the detection and possibly grading the severity of road damages. The initial objective of the project is to develop a model that can classify whether a road is damaged or not with at least 85% accuracy. For this reason, Ramboll RST Finland have provided us with annotated data in form of images and a csv-file, from their survey made during the end of 2023.

1.1 Purpose and Research Question

The purpose of this thesis is to explore the prerequisites and challenges associated with integrating machine learning into Ramboll's current road surveying process. This investigation aims to provide valuable insights and recommendations for successful implementation and further development.

Searching for the answer to the research question: Can the data provided by Ramboll be leveraged through machine learning to achieve the project goal of 85% accuracy in road damage classification?

1.2 Limitations

Due to time constraints, we have prioritized the utilization of existing annotations, categorizing each image as either "damage" or "no damage", which limited us to models with the suitable architecture. Additionally, we have constrained ourselves to using one pre-trained model that has been proven to perform well on this type of task while demanding minimal time and computational resources.

1.3 Outline of the Thesis

This thesis begins with an introduction and description of the problem. Subsequently, it details the relevant general and specific theory required for understanding this thesis. Following this, we describe the experiments and methods used. We then present the results and discuss the findings. Finally, we summarize our conclusions and suggest avenues for further research and model development.

2 Theory

In this chapter, we describe the theory needed to understand the applied ResNet model, retraining a pre-trained model, also called transfer-learning and properly evaluate training and end results.

2.1 Neural Networks

Machine learning is the utilization of available data to train computers for problem-solving, using algorithms to learn patterns and relationships within the data. Whether the algorithms are used on their own or have been assembled in some way, they are commonly referred to as "models". One area where ML has proven to be effective is computer vision, where algorithms analyse pixel patterns and intensity in digital images or videos. Common examples include unlocking your phone by facial recognition and object detection used by self-driving cars. A group of models that has shown to have been particularly successful in solving computer vision problems is Neural Networks (NN).

2.1.1 Basics of Neural Networks

Neural Network models have been inspired by the neurological structure of a biological brain and consist of a set of layers, where each layer contains a set of neurons (nodes) that are interconnected to the nodes in the next layer, much like the neurons in a human brain. Figure 1 illustrates a single neuron, its placement, and connections to other neurons in a neural network. Note that a network always has one input and one output layer, beside at least one hidden layer with a set of nodes, where the computation and transforms are conducted, before the results are passed on to the nodes in the next layer. When the model is trained, the resulting predictions after each batch of examples, are compared to the ground truth values in the training data, and an error (loss) is calculated. Then the algorithm adjusts the weights by a factor (learning rate) in the opposite direction of the gradients of the loss, layer by layer starting from the last layer moving backwards through the network to minimize the loss, this method is called backpropagation (Géron, 2019, pp. 289-292).

A layer using the nodes described earlier are called fully connected layers (dense layers), however other type of layers has been developed to tackle different type of problems and data types. For example, convolutional filters used in convolutional layers and recurrent units (like LSTM and GRU cells) in recurrent layers. These types of layers share common characteristics. They are trainable, using backpropagation and offer tuneable hyperparameters, such as the number of nodes or filters and the choice of activation function.

In addition to these main types of layers, a network may require or benefit from transformation and regularization layers that are non-trainable but may have hyperparameters to configure the effect it has on the network.

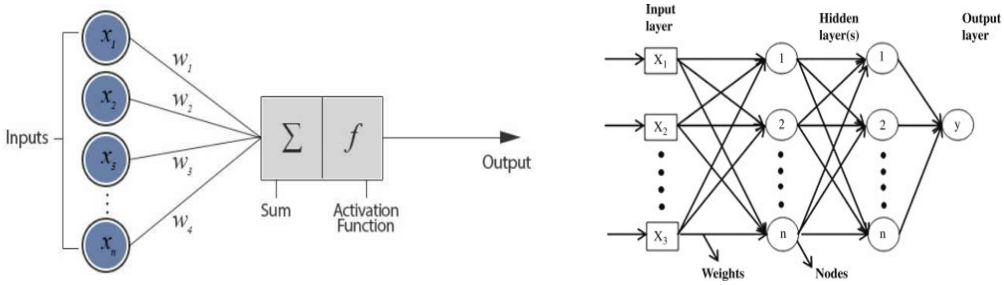

Figure 1: Overview of a single neurons structure (left) and connections between neurons in different layers within a neural network (right)

Over the years it has been proven that increasing the number of layers in a neural network can effectively improve the model´s ability to extract different levels of features from complex data. This phenomenon has developed into a sub-field of Neural Networks that has been given the name "Deep Learning" (DL), because of the often, high number of hidden layers used in the network.

Within the DL field models with convolutional layers, also called Convolutional Neural Networks (CNN) has been proven to be very useful for high dimensional data classification and computer vision problems (Krizhevsky, Sutskever, & Hinton, 2012).

2.2 Pre-trained models

Rather than starting on the resource-intensive and time-consuming task of developing a new model from scratch, we explored the potential of pre-trained models. This approach offer a significant advantage by utilizing networks that have already been trained on extensive datasets to perform well in various tasks, and adapting them to our own data with the help of transfer learning (Géron, 2019, pp. 481-483).

Our research indicated that the ResNet50 architecture, with highly optimized classification training using the open-source ImageNet, would be a good starting point and is susceptible to transfer learning in the field of road classification (Saeed, Nyberg, & Alam, 2022). Both the model and weights from its training on ImageNet were provided by the Keras library, which also offers easy-to-use tools for data preparation, transfer learning, and evaluating models.

2.2.1 Transfer Learning with Fine Tuning

Utilizing a pre-trained model, which is already well-trained to perform a specific task such as classifying different objects in an image, saves significant time and resources. However, this does not automatically ensure that the model will excel at any new classification task you give it. Pre-trained models are excellent starting points and, as Géron (2019, p. 346) notes "… also require significantly less training data." compared to training a model from scratch.

When adapting a pre-trained model to a new task, it's often necessary to replace the output layer, as the number of nodes in the original layer may not match the requirements of the new task. Finding out how many other layers to replace or modify requires experimentation.

A key consideration during transfer learning is that the weights in the newly replaced or added layers are not initially as well-tuned to the task as the weights in the pre-trained layers. These less optimized weights can cause disturbances in the pre-trained layers. Because the error gradients generated during training, are more erratic for the new layers. To prevent this, it's a good idea to initially freeze the pre-trained layers. This allows the model to focus on training the new layers with the new data without the risk of disturbing the established, valuable features in the pre-trained layers.

Once the model has been "transferred" to the new data, it is many times a good idea to apply some additional training also called Fine Tuning. This is done by experimenting with unfreezing some or all the pre-trained layers. This is often done at a lower learning rate to fine-tune, the entire model to optimize performance for the new task.

Throughout these processes, continuously monitoring the model's performance is crucial to identifying the best configuration and to prevent overfitting.

2.3 ResNet

In this section, we will focus on the underlying theory, architecture, and components common to ResNet models. While all ResNet variants primarily utilize the same techniques and components, they do differ in aspects such as the number of layers and specific layer configurations. This section aims to provide a general overview, applicable to most ResNet models. Figure 2 provides an example of layer composition within a ResNet model.

2.3.1 Residual Learning

In the article "Deep Residual Learning for Image Recognition", He, Zhang, Ren, and Sun (2015) introduced the concept of residual learning through the application of skip connections. Their groundbreaking research, successfully demonstrated that residual learning not only enhanced the efficiency and optimization of neural networks but also effectively addressed critical challenges encountered in deeper networks. These challenges including the vanishing gradient problem and degradation issues, which had previously been significant obstacles in the development of deep learning models.

Residual learning uses the skip connection to add the input from previous layer to the output of a layer deeper in the network before this combined result is passed on as input to the next layer. The layers surpassed by a skip connection are collectively referred to as a residual unit or block. By doing so, the skip connection forces a unit to model the residual function $f(x) = h(x) - x$, instead of conventional identity function $h(x)$. These blocks are stacked together between the input and output layers often with other types of layers adapted for the intended use.

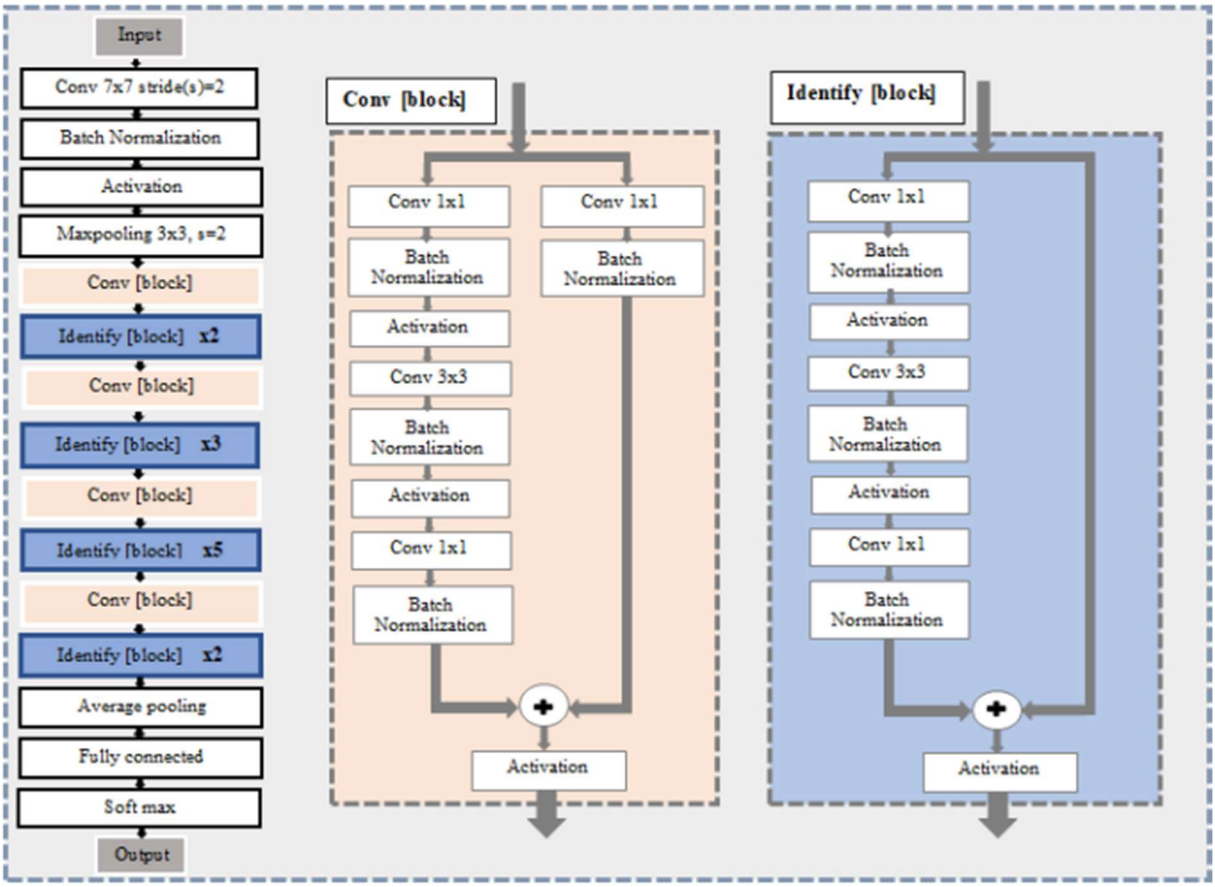

Figure 2: Layer composition within a ResNet model and its building blocks.

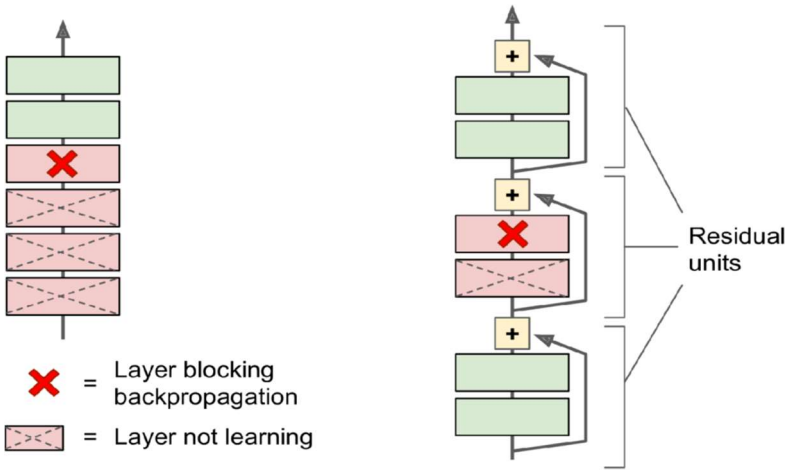

Figure 3: Regular deep neural network (left) and deep residual network (right)

One of the main strengths of ResNet models has efficiently been described by Géron as "The deep residual network can be seen as a stack of residual units (RUs), where each residual unit is a small neural network with a skip connection" (2019, p. 471). Giving the networks the ability to maintain learning in sections deeper into the network even some part has not yet started or stopped learning, illustrated by Figure 3.

2.3.2 Convolutional Layers

The main distinctions of convolutional layers in Convolutional Neural Networks (CNNs) are the use of kernels and feature maps (also called filters). Kernels can be described as a window that is moved across the data and the feature maps are represented as weight matrices corresponding to the kernel size. These filters are then applied to the input values within the kernel by element-wise multiplication, for each stride (movement). The resulting products are then summed up to produce an output matrix of convoluted values one for each specific feature map used. This process also known as a convolution is illustrated in Figure 4.

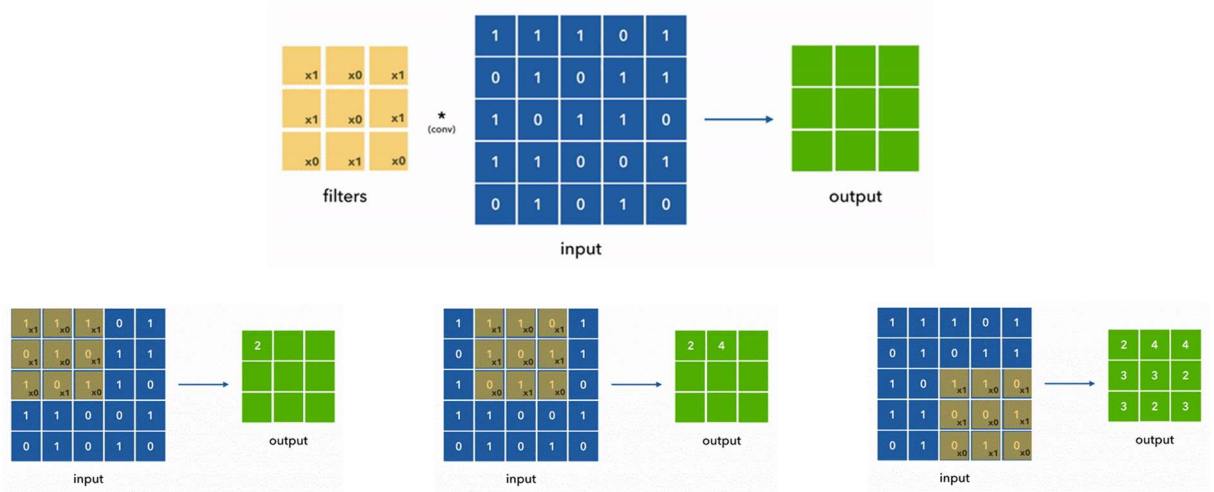

Figure 4: Illustration of a convolution operation in a CNN, showing how filters are used on input data to create output feature maps.

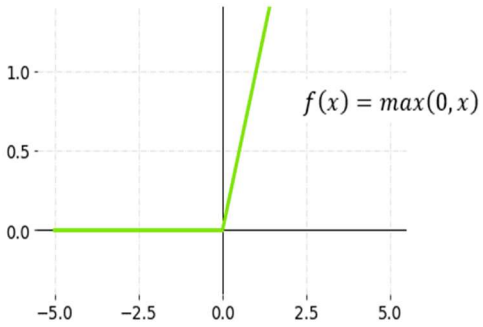

Figure 5: Graph of the ReLU (Rectified Linear Unit) activation function, including its definition.

2.3.3 Activation Function

The activation function introduces non-linearity by applying a function to the results from a neuron, in CNN the convoluted values before they are passed as output to the next layer. The necessity of an activation function, especially when dealing with complex data as is often the case in image processing, has been explained by Aurélien Géron (2019, p. 292).

As mentioned earlier, the activation function used in ResNet is ReLU (Rectified Linear Unit). ReLU has gained vast popularity since its introduction to the field of neural networks by Nair and Hinton (2010). It not only addresses the issue of vanishing gradients, explained by Géron in his book (2019, pp. 332-345) which can be problematic with other activation functions. Additionally, ReLU is known for its computational efficiency, contributing to reduced training time. ReLU sets all negative values to zero and ranges between 0 and infinity as illustrated by its graph in Figure 5 along with its definition.

It´s simplicity allows ReLU to reach convergence more efficiently during model training compared to other activation functions. Unfortunately, this advantage does not come without a downside and is called "dying ReLU´s", again, described by Géron (2019, p. 335). Note that as previously mentioned ResNet models already addresses this problem with the use of residual learning.

2.3.4 Batch Normalization

Batch Normalization (BN), proposed by Sergey Ioffe and Christian Szegedy (2015), effectively addresses the vanishing gradient problem and allows for the use of larger learning rates without compromising performance. BN also introduces a regularizing effect, potentially reducing the need for other regularization methods like dropout. Additionally, it makes the model less sensitive to the initial weight settings, positively impacting the speed of convergence and overall computational efficiency.

The main idea of Batch Normalization (BN) is to normalize the input values for each layer in the network. It does this by zero-centering and scaling the values either before or after the activation function, depending on the implementation. BN calculates the mean and standard deviation within the current batch to perform this normalization. For a more in-depth explanation, Aurélien Géron's book provides a detailed discussion on Batch Normalization in the specified section (2019, pp. 338-345).

2.3.5 Pooling Layers

Pooling layers play a crucial role in neural networks by reducing the spatial dimensions of images or feature maps, with regards to computational efficiency and effective feature extraction (Géron, 2019, pp. 456-458). The mechanics of pooling layers closely resemble convolutional layers, utilizing a kernel (or pool) and stride for processing. However, unlike convolutional layers, pooling layers do not use

weight filters; instead, they apply aggregation functions such as max or mean. Among these, max pooling is the most popular and widely used. Where the max value within the pool is picked and passed on to the next layer.

Within the category of pooling layers, there exists a subcategory known as "Global" layers. These layers function similarly to regular pooling layers; the main difference is that their kernel size matches the dimensions of the feature maps they are presented with. For global pooling, average pooling is commonly used and is often considered the most suitable aggregation function. The primary role of a Global layer is to condense the data within each feature map it receives into a single aggregated value. This not only helps the model during convergence but also prepares the outputs for interpretation, all while eliminating the need for the traditionally used flatten layer.

2.3.6 Dense Layers

Dense layers are layers in which every neuron is fully connected to each neuron in the previous layer. By default, ResNet50 includes only one such layer, which is the last layer, also known as the output layer. The output layer plays a crucial role in interpreting the predictions made by a model. It achieves this by transforming the output from the previous layers into probabilities. Different activation functions can be applied, with "softmax" for multi-class classification, and "sigmoid" for binary classification scenarios.

2.4 Evaluation Metrics

In this study, we used Accuracy as the primary evaluation metric. However, we recognize that Accuracy alone may not be sufficient to uncover various issues like imbalance or bias in the data, model robustness, and potential bias in its predictions. Therefore, we also used other widely accepted metrics for classification problems, including Precision, Recall, F1-score, and the Confusion Matrix. These metrics are calculated by in different ways comparing the correctly classified instances of the positive and negative classes, known as TP (True Positive) and TN (True Negative), with the corresponding FP (False Positive) and FN (False Negative) for the incorrectly classified instances. When conducting binary classification and in our case knowing what roads damages are is of interest, this class will normally represent the Positive class and the un-damaged roads the Negative class.

In the following sections, we will provide descriptions of each metric and their respective calculations.

2.4.1.1 Accuracy

Accuracy assesses the model's correctness by calculating the ratio of correctly predicted images to the total number of test images.

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

2.4.1.2 Precision

Precision quantifies the accuracy of a model's correct predictions for a specific class by calculating the ratio of true positives of the sum of positive predictions made by the model for the specific class.

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

2.4.1.3 Recall

Recall quantifies how well the model captures all instances of the positive class among all actual instances of the specific class.

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

2.4.1.4 F1-score

F1-score is the harmonic average of precision and recall, that provides a single metric to assess the overall performance. That is useful when the balance between precision and recall is important and help in making trade-offs when necessary.

$$F1 = \frac{2 \times TP}{2 \times TP + FP + FN} \qquad F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

2.4.1.5 Confusion matrix

A confusion matrix can be described as a visual representation that provides a clear way to assess precision and recall. By displaying the quantities of the predicted outcomes in relation to the actual values, the confusion matrix is a good way to understand the types of errors the model is making. Below is a simple representation of a confusion matrix.

	Predicted Positive	Predicted Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

3 Metod

In this chapter, we describe the origin of, and preparations made to the data, model selection and choice of limitations how we evaluate model performances and explaining the experiments conducted.

3.1 Tools

The programming was done using Python with the Keras API (Keras) and TensorFlow (TensorFlow) in a Google Colab environment using the T4 GPU unit (Colab). Keras provided a simplified framework not only for training and evaluating models but, in our case, also supplied us with the ResNet-50 architecture and the weights pretrained on the ImageNet dataset.

3.2 Data Collection

The data we received was collected between September 21, 2023, and October 11, 2023. The data consisted of .jpg images and a csv file.

Images were automatically captured from every 20-meter section of the roads using GoPro HERO9 cameras installed inside the windshields of six different vehicles. Each vehicle was operated by one out of a total of seven trained operators. The assessments were conducted by the operator in real-time and recorded using tablets equipped with V-tracker software, which allowed for the synchronization of GPS coordinates with the current road section.

3.3 Data Exploration

A total of 44499 samples (images and annotations) were made available.

3.3.1 Data Quality

The use of the same type of cameras and settings resulted in that all images had the same high image resolution 5184 x 3888.

During our initial exploration of the annotations, we found a systematic inconsistency between the annotations and images. This turned out to be caused by an unexpected delay in the tracking system. After this was re-solved Ramboll provided us with adjusted annotations.

3.3.2 Filtering

To limit the data to use in our study a subset of 21125 instances from the first (Run1) and fourth (Run4) run was chosen. Mainly this subset was chosen since Run1 was a part of the POC made before starting the project and Run4 as it contained a larger number of damaged roads, ensuring that we would be able to end up with a balance between classes in our dataset.

While browsing the images, we noticed that some images had potentially problematic characteristics, such as items like windscreen wipers, vehicles, and animals blocking the road surface, or compromised image quality due to factors like blurriness, strong glare from the sun, or darkness. Figure 6 exhibits a few examples of this.

Figure 6: Examples of problematic images.

After manually removing these "problematic" images, we ended up with a substantial dataset of 15,559 images, with a balanced distribution between the damage (46%) and no-damage (54%) classes. Both desirable features for successfully applying Machine learning. This dataset represents our base line dataset and was labelled "full_ds".

After completing our initial training and testing phases, we decided to create a high-quality dataset. The images in this dataset that we labelled "best_ds" where manually selected from the full_ds and we ensured that it exclusively contains images where the distinction between damage and no damage can be unequivocally identified. We have also made sure that all the images in the dataset share similar weather and lighting conditions. Specifically, we have excluded images with overly strong sunlight or those where the sun creates distracting shapes and shadows. Our additional research into the realm of transfer learning, indicated that it could be done with considerably fewer images than when training a model from scratch. It was our decision to keep the number of images to a 1000 due to that this process was manually executed.

3.4 Preparing Images

For the images to be compatible with our pre-trained version of ResNet, it is necessary to reduce the image dimension to 224x224. Now, we will describe the four methods that emerged from our explorations in this field and explain why we choose to test them. Our name for the methods became: Original crop, Square, Padded and Stretched, see Figure 7.

3.4.1 Resize and Cropping

The images were cropped to isolate and retain only the area of interest, i.e. road surface. To minimize the risk of irrelevant elements, such as the sky, car hoods, and roadside terrain, introducing unwanted bias. For instance, if one vehicle predominantly surveyed roads with damages, the model might learn the placement or colour of that specific vehicle's hood as a feature, leading it to predict that all images recorded with that same vehicle were damaged.

Since resizing images with different proportions between the height and length of the original and target size may result in more data distortion than when the proportions are the same, we explored different cropping methods. Since the target format is a square 224x224 we started off by identifying squared areas, to minimize the distortion. This is the case for both Original crop and Square, where the difference is in the size of the area of interest. By keeping as much as possible of the road, but maybe introduce a lot of useless or potentially harming information (Original crop) or by **only** including road surface (Square) with the risk of having removed to much useful information.

Additionally, we examined whether similar trade-offs could be achieved when maximizing the size of the area of interest, resulting in a rectangular area. This was explored while keeping the proportion ratio (Padding) and adding padding to fit the format or allowing the resize to stretch the dimensions to fit the target format (Stretched) with the risk of distorting the image.

Figure 7: Examples of different crop and resize methods.

During this process we notice that the documented operator name was not vehicle specific as we previously had assumed, rendering it an unreliable indicator for determining which coordinates to use during cropping. However, since it turned out that only one operator had changed vehicle and only recorded a very small number of instances and the effect was a barely visible car hood, we did not address this issue further at this point.

3.5 Data Augmentation

In the exploration of various augmentation techniques to enhance our model's performance, we found that contrast adjustment was particularly effective in improving detail visibility for human subjects, see Figure 8. As a result, we decided to incorporate adjustments to contrast and brightness, along with random image flips and rotations, into our training data augmentation strategy. Additionally, we experimented with applying contrast adjustments as a "filter" during both training and testing phases.

However, due to time constraints, we had to narrow down the number of configurations to test. Consequently, we focused on the following four configurations:

- **no_aug:** This configuration involved no augmentation or contrast adjustment.
- **no_contrast_aug:** Augmentations were applied, but no contrast adjustment.
- **contrast:** Only contrast adjustment was applied as a "filter" during both training and evaluation.
- **full_aug:** Both contrast adjustment and augmentations were applied during training.

With these configurations we were able to explore the impact they had on our model's performance.

Figure 8: Different contrast factors applied to the same image.

3.6 Training and Testing

To mitigate the risk of overfitting, we employed a train, validation, and test split methodology. This involved partitioning each "base" dataset, into distinct segments as shown in Table 1. The validation dataset served the purpose of monitoring the training performance, while the test dataset remained reserved solely for evaluating the final, most promising models.

Table 1: Total, balance and sample distribution for each dataset.

	Total	Balance	Train	Validation	Test
full_ds	15559	46/54%	9956	2490	3113
best_ds	1000	50/50%	640	120	200

After the split, we proceeded to apply the different crop techniques including, Original crop, Square, Padded, and Stretched, to the base datasets. Resulting in a total of eight datasets. Enabling us to assess the effect of each data collection and preparation method separately have on our model.

Each dataset and augmentation configurations were then used for transfer learning (TF) and that resulting model was then fine-tuned (FT) with the same dataset. The batch size was set to 32 and length of each phase used was 5 epochs. For the best_ds when augmentation was applied, we also tested to repeat the dataset 5 times per epoch. The performance was evaluated after each phase with help of the validation metrics, to determine if one or both steps were most effective.

Once the best models for each base dataset was identified, the models were tested on their corresponding test data. The model using the best_ds for training, was also tested the test data from full_ds, to see how well it would perform on unfiltered data.

4 Results and Discussion

In this chapter, we will present and discuss the outcomes of our study, with a focus on the best-performing model from each dataset approach. It is our decision to highlight only the best-performing models from each dataset approach is rooted in the observation of minimal performance variations among other parameter configurations. However, for readers seeking a more in-depth analysis, you can find all the training results for various configurations in Appendix A.

Table 2: Composition of the best performing models, presented in Table 3.

	Dataset	Crop	Augmentation
Model A	full_ds	Original crop	contrast
Model B	best_ds	Square	full_aug

Table 3: Results from the best models. (*) denotes the optimal training method (TL or FT) for each model, used in evaluating the metrics on the left. (**) represents the test accuracy achieved with the full_ds test dataset.

	TL Val Accuracy	FT Val Accuracy	Best (*)	Test Accuracy	Precision	Recall	F1-Score	(**)
Model A	72.9 %	74.8 %	FT	75.2 %	70.3 %	75.7 %	72.9 %	-
Model B	95.0 %	95.6 %	FT	97.5 %	95.9 %	94.0 %	94.9 %	64.0 %

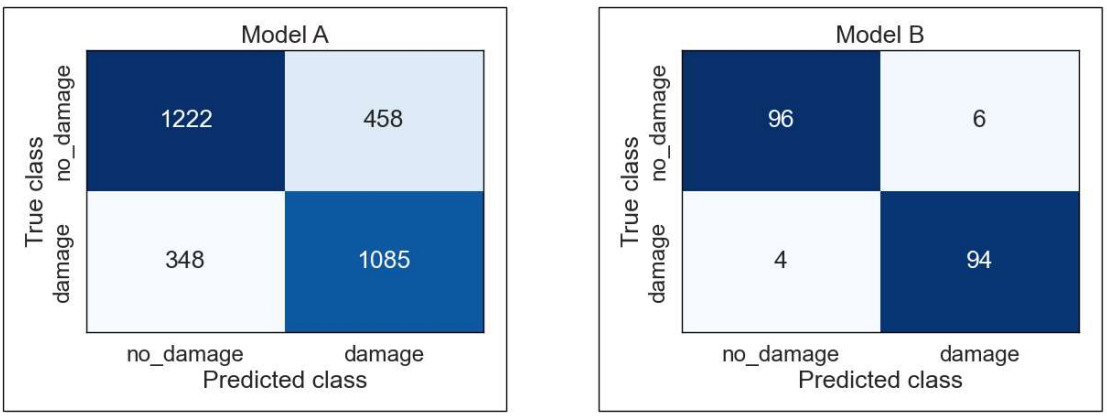

Figure 9: Confusion Matrices for Models A and B. Based on the test datasets corresponding to each model's training dataset approach.

4.1 Data quality

Even by only achieving an accuracy of 75.2% and falling short of the 85% target set for this study. Model A still provided us with a positive answer to one of our underlying research questions, whether the data from current survey method is valuable for machine learning. By demonstrating that with the full_ds setup, i.e. without any alterations of the images and only essential preparations, such as filtering out images with no or small amount of visible road surface and ensuring accurate annotations, a result although modest was achievable.

4.2 Data Filtering

We conducted experiments with the aim to improve our model's performance by adjusting augmentations, contrast filtering, and hyperparameters along with the different crop and resize methods. When surprisingly, none of the combinations of between these things, had minimal or even negative effects on the model's performance. We shifted our focus to the data itself.

What originally was a focus on trying to identify poor annotations or very minor damage cases resembling those without any damage. Soon shifted to image characteristics, as we observed that under various conditions such as wet or dry road surfaces, sunny or shady lighting, and dark or bright environments, similar damages could look different to the human eye. This led us to develop a hypothesis.

Our hypothesis was that since humans rely on visual cues to distinguish between these conditions, the model, which also learns from visual features, might get confused by these varying conditions. We were able to visually confirm our hypothesis, by applying edge detection filter to images as seen in Figure 10. Where the difference in the number of edges is detected is quite large. This experiment is of importance since the application of filters that edge detection uses is similar to the feature maps the convolutional layers utilize to extract features to learn from.

This led us to manually collect the dataset (best_ds) that was the base for Model B, only containing images with similar characteristics.

4.3 Method and Model

With robust results showing only slight differences between the training and test accuracy, as well as Precision, Recall, and F1-score for both models, we have been able to confirm that using ResNet50 with the ImageNet training is transferable to Ramboll's data. Therefore, in our opinion, it serves as a reasonable starting point for further development for both dataset approaches.

4.4 Model Performance

While Model A fell short of the 85% accuracy goal, achieving 75.2%, it remains far from being without value. It has not only provided valuable insights in our study, but Model A also have some positive attributes. It is "ready to go" without the need for extensive efforts and its performance level would

Figure 10 Example of edge detection applied on images with no damage but different characteristics.

be expected to be robust, due to its prior exposure to a diverse range of conditions. Furthermore, the full_ds approach applied to Model A allows for most of the data to be used for training but also when making predictions. Which is important when trying to make sure that the whole road network is surveyed in a timely manner.

While Model B's impressive 97.5% performance is noteworthy, it comes with trade-offs. This model relies on the "best_ds" approach, where carefully filtered images share similar characteristics, aiding the model in learning relevant road surface features effectively. However, when faced with images that differ significantly from the training data, the model's generalization ability diminishes, as evidenced by a drop in accuracy to 64.0% (see column (**) in Table 3).

Surprisingly the largest limitation of this method: by manually choose images that have similar characteristics, lies not in the reduced size of the training data. Contrary to training a model from scratch, the necessity for extensive data in transfer learning is minimal, a fact supported by Géron (2019, p. 346) and confirmed by our experiments where the best_ds had merely 1000 images. The real challenge emerges when attempting to ensure comprehensive coverage of all road sections within the survey area. Relying on this selection process can result in significant gaps; particularly, if a substantial fraction of the images does not meet the specific conditions required, it may leave only a few viable sections (images) for classification.

5 Conclusions and Future Work

In this chapter, we present our conclusions and provide recommendations for future work aimed at developing a robust model for gravel road damage classification.

5.1 Conclusion

Our first conclusion is that our study proves that the data, particularly in terms of image quality and camera stability, can indeed be utilized for machine learning purposes. However, we emphasize the critical importance of ensuring precise alignment between images and annotations for effective application.

In addition to our first conclusion, we will now address our research question: Can the data provided by Ramboll be leveraged through machine learning to achieve the project goal of 85% accuracy in road damage classification? Although we do not consider ourselves quite there yet due to the limitations of Model B, which currently makes it impractical, we are of the opinion that our findings suggest that the method applied in our study has shown great potential for further development to achieve the goal, particularly in the area of ensuring similar image characteristics.

5.2 Future work

As we conclude our study, we are glad to see that the methods explored have shown potential in some key areas of Machine Learning. This gives us the opportunity to identify key areas to focus on for future development using these methods, as well as to make general suggestions of preparations required for more advanced application of Machine Learning.

5.2.1 The "bare" essentials

Starting off with a few, but crucial things to implement into the current documentation process. To make any future Machine Learning development easier and ensuring that performance is reliable.

- **Annotation alignment:** It is crucial for the annotations line up with correct image. If not, the model will be confused on what is a damage and not. Ensuring correct annotations during the documentation stage or having them adjusted after the decision-making stage, where it is ultimately decided what section (image) is damaged or not. Makes any further Machine Learning development, easier and will produce models with more robust performance.
- **Vehicle-ID:** In addition to the operator's name, it would be beneficial to have a vehicle ID, as this would be more consistence indicator on how the camera is mounted. Allowing for reliable automated adjustments to alight the area of interest (road surface). While we consider adding a **Vehicle-ID**, would be the bare minimum. For the same beneficial reasons adding a **Camera-ID** and **Camera-settings** should also be taken into consideration.

5.2.2 Data filtering

In our opinion, focusing on the best_ds approach would be the best start for further development. Areas to start researching would be.

- **Image similarity:** Experimenting with software manipulation of images, physical filters, or different camera settings to reduce the effects of unwanted characteristics. For this purpose, it might be beneficial to consult a photographer or expert in image manipulation for advice. Since even if software manipulation most likely is the most efficient way to do this, physical filters or different camera setting might offer some things software cannot or a combination is most effective.
- **Specialized models:** By using dividing images into different groups, what have certain characteristics and features. Specialized models could be developed for each group. To get

this to work an efficient segmentation method is needed. Finding the right balance of number of segments i.e. number of models to develop and the overall performance is important, as well as find the best strategy for handling if it turns out that the models will not be able to perform at the same levels.

5.2.3 Different models

Given that our initial research findings indicate the minimal impact of the choice of model on performance when applied to the same data, with the results table (Table II) in (Saeed, Nyberg, & Alam, 2022, p. 6)) supporting this observation, and confirmed by the results of this study, we would emphasize the importance of focusing on data filtering, before or at least in parallel with exploring different models. But when doing so, our recommendations for exploring other models would be.

- **Different models and pre-training:** Experimenting with different architectures: ResNet, VGG, InceptionV3, MobileNet, DenseNet, Xception.
- **Different pre-training:** Different pre-trainings with same or different architectures (above) may also have different effects, examples: ImageNet, CIFAR-10/100, Places365, OpenAI's CLIP
- **Deeper Networks:** For the full_ds approach it might be beneficial to explore models with deeper networks for example ResNet152. As they often have a better performance when the data is very complex and with small nuances in the data. But then often demand a large amount of data. Which this approach over short time would be able to provide vast amount of data, with very low effort.
- **More classes:** Withing the project the wish to explore grading different degree or types of damages has been expressed. Adapting the current model (ResNet50) should be easy to do code wise, during transfer learning process. Beside dividing annotations into different classes, this might introduce new types of issues that needs to be addressed. However, most of them would most likely be common issues arise when dealing with multiclass classification.

5.2.4 Models to be explored.

Finally, we would like to explore models that, in this study, could be likened to "The Road Not Taken." With a positive confirmation that the data is valuable for enhancing and automating the survey process with Machine Learning, it may be a good idea to investigate models beyond strictly classification models. One type of model worth considering is object detection models; here are some examples: Faster R-CNN, YOLO, SSD, and RetinaNet.

One noticeable difference to keep in mind when starting with these kinds of models is that they will require more resources for annotations. This is because they demand not only class labels but also bounding boxes. However, the rewards for the increased annotation cost and research requirements are expected to be realized in the form of high-performance levels and flexibility in determining what features and damage frequency, that contribute to classifying a road section as damaged or not.

Appendix A

Table 4: Validation accuracy for all models experiments.

dataset	crop	steps_per_epoch	augment	TL	FT
best	original	1	no	0.9	0.9125
best	square	1	no	0.9125	0.93125
best	padded	1	no	0.88125	0.91875
best	stretched	1	no	0.90625	0.925
best	original	1	contrast	0.9	0.925
best	square	1	contrast	0.9125	0.95
best	padded	1	contrast	0.85	0.90625
best	stretched	1	contrast	0.91875	0.93125
best	original	1	non-contrast	0.875	0.9125
best	square	1	non-contrast	0.90625	0.9375
best	padded	1	non-contrast	0.9	0.925
best	stretched	1	non-contrast	0.8375	0.90625
best	original	1	full	0.85625	0.85625
best	square	1	full	0.925	0.91875
best	padded	1	full	0.85625	0.71875
best	stretched	1	full	0.85625	0.925
best	original	5	full	0.90625	0.94375
best	square	5	full	0.95	0.95625
best	padded	5	full	0.85625	0.8625
best	stretched	5	full	0.9375	0.9375
full	original	1	no	0.732932	0.722892
full	square	1	no	0.724096	0.723695
full	padded	1	no	0.7249	0.733735
full	stretched	1	no	0.72249	0.733735
full	original	1	contrast	0.729317	0.748193
full	square	1	contrast	0.704016	0.745382
full	padded	1	contrast	0.700402	0.714056
full	stretched	1	contrast	0.734137	0.731727
full	original	1	non-contrast	0.728112	0.746586
full	square	1	non-contrast	0.734137	0.741365
full	padded	1	non-contrast	0.697992	0.708434
full	stretched	1	non-contrast	0.694377	0.728514
full	original	1	full	0.689558	0.702811
full	square	1	full	0.698795	0.718474
full	padded	1	full	0.684739	0.68996
full	stretched	1	full	0.718474	0.719679
full	original	5	full	0.722892	0.732932
full	square	5	full	0.719679	0.718474
full	padded	5	full	0.698795	0.694377
full	stretched	5	full	0.704819	0.724498

References

Colab. (2024). *https://colab.research.google.com/*.

Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow.* (R. R. Tache, Ed.) Canada: O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472. Retrieved from http://oreilly.com

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. *ArXiv*. Retrieved from https://arxiv.org/abs/1512.03385

Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *arXiv*. Retrieved from https://arxiv.org/abs/1502.03167

Keras. (2024). Retrieved from https://keras.io/.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional. *Advances in Neural Information Processing Systems 25* (pp. 1097–1105). Neural Information Processing Systems Foundation, Inc. (NeurIPS). Retrieved from https://proceedings.neurips.cc/paper/2012

ROADEX. (2023). Retrieved from roadex.org: https://www.roadex.org/e-learning/lessons/gravel-and-forest-road/general/

Saeed, N., Nyberg, R. G., & Alam, M. (2022). Gravel road classification based on loose gravel using transfer learning. *International Journal of Pavement Engineering*. Retrieved 2024, from https://www.tandfonline.com/doi/full/10.1080/10298436.2022.2138879

TensorFlow. (2024). Retrieved from https://www.tensorflow.org/.