STR459: Artificial Intelligence and Robotics

Cracking down on road cracks: Using machine learning for safer and more efficient road maintenance

Date submitted: **8 April 2025**

Total number of pages: **25**

Candidate Numbers: **15, 20, 32, 33, 69**

Do you allow your report to be used as an example in the future: **Yes**

Table of Contents

1 Introduction

Road maintenance accounts for a substantial share of public infrastructure spending. According to the Australian Bureau of Infrastructure and Transport Research Economics (2023), most countries spend between 20% and 40% of total road expenditure on maintenance. In Norway, 10.8 billion NOK are directly allocated to road maintenance in 2025, an increase of 1.1 billion NOK from the year before (Regjeringen, 2025). This trend of increased road maintenance spending in real terms has been consistent over the past decade (SSB, 2025), with reasons such as traffic safety, climate effect and military mobility most recently cited by The Norwegian Ministry of Transport (Regjeringen, 2024).

In addition to being a matter of economic efficiency, road maintenance is also crucial for public safety. Road quality is essential for reducing traffic-related fatalities, with road traffic injuries being the world's leading cause of death for children and young adults (WHO, 2023). In Norway, one quarter of traffic deaths were fully or partially due to faulty roads, further highlighting the importance of timely road maintenance (NAF, 2024) The high safety risk of faulty roads necessitates frequent monitoring. However, traditional road inspection methods remain labor-intensive, time-consuming, and costly.

This paper investigates the use of convolutional neural networks (CNNs) for automated and more cost-efficient crack detection in asphalt surfaces. By applying machine learning (ML) to visual data gathered from road monitoring, such systems can enable earlier intervention and more efficient allocation of maintenance resources. We propose that integrating CNN-based detection into road management systems can improve public spending efficiency while also enhancing road safety and durability.

2 Data Analysis

2.1 Business Idea

Efficient road maintenance depends on accurate and timely monitoring of road conditions. Without a clear understanding of the current state of infrastructure and its rate of deterioration, it becomes difficult to allocate maintenance resources effectively. In Norway, this maintenance is the responsibility of Statens Vegvesen (Statens Vegvesen, 2025). To address the increasing expenditures on road maintenance, we propose that they deploy ML models to automatically detect cracks in roads, providing a continuous and data-driven overview of Norway's road infrastructure.

Traditional road inspections, which rely on manual surveys or specialized vehicles, are time-consuming, labor-intensive, and often limited in geographic coverage and frequency. By equipping frequently used vehicles (e.g. buses and postal service vehicles) with GPS and cameras, roads can be continuously monitored. ML models can process the collected data to identify and map cracks in real time, providing a comprehensive overview of Norway's road network. This enables predictive maintenance by identifying at-risk roads and estimating when they are likely to require intervention. For example, based on historical trends, the system might flag that road X will breach acceptable thresholds in Y months, helping prioritize resources effectively and address issues before they worsen.

In developing this overarching system, the flagging done by the ML-models outlined in this report will play an integral part. The proposed model will identify whether a crack exists at a given location or not. Once the infrastructure is established, the model could be further developed to also assess the severity of cracks, for example by following the methodology outlined in Ha et. al. (2022). With widespread data collection across Norwegian roads and greater system sophistication, the resulting insights could enable significant cost savings.

2.2 Dataset and Choice of Tools

2.2.1 Dataset

The performance of our ML models will inherently be dependent on the quality and size of the underlying dataset (Goodfellow, Bengio, & Courville, 2016). One key consideration in dataset selection is the nature of variance intended for model training. Because Statens Vegvesen can employ a standardized data collection setup, the ML model does not need to accommodate variance introduced by different camera setups. Therefore, we are looking for a dataset with a consistent capture methodology, thus reducing the wrong sort of variance (e.g., from differing camera settings or mounting techniques) and allowing the model to focus on natural variations such as differences in lighting, weather condition, asphalt types and other noise usually found on the road.

While exploring common sources for datasets, such as Kaggle, Mendeley Data, and others, we identified several candidates designed for road crack classification and segmentation. However, many datasets lacked detailed documentation regarding their collection methodologies, making them unreliable for our use case. Among those with sufficient documentation, they would commonly suffer from limited sample sizes, and larger datasets were often aggregated from multiple smaller sources, introducing undesirable variance. Moreover, many of the datasets predominantly featured concrete surfaces, which are not representative of Norway’s primarily asphalt-based road network (Statens Vegvesen, 2024).

After evaluating several candidates, we selected the "Asphalt Cracked and Uncracked" dataset collected by Baduge et al. (2023) for our project. The dataset follows a rigorous and replicable image capture process, where images were captured using a smartphone mounted 850 mm above the road surface behind a car traveling at 50 km/h (see details of the setup in Figure 1).

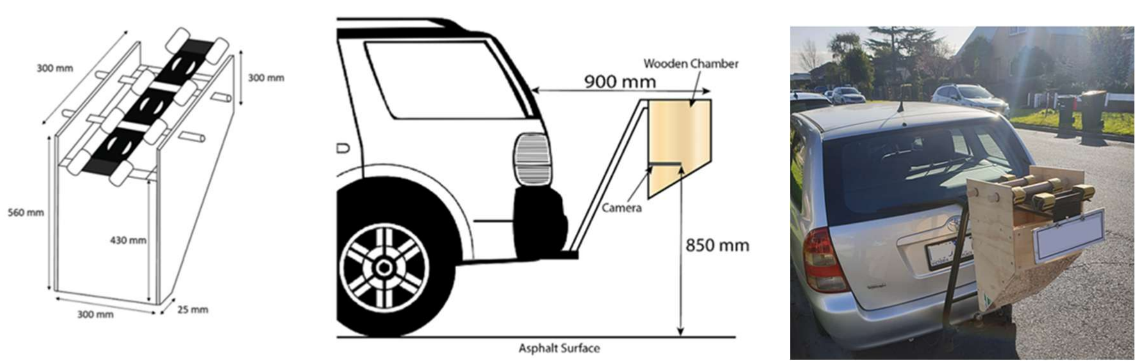

Figure 1: The data collection setup from Baduge et. al. (2023)

The final dataset contains a total of 2000 high-resolution images of cracked and uncracked asphalt pavement. These images were then split and separated by the authors into 8000 labelled images of 360x360 pixels with RGB Channels.

To structure the dataset for model training, we wrote a Python script randomly assigning images from the dataset into training, validation, and test sets using a 60/20/20 split, whilst ensuring the datasets were balanced across classes. This approach is recommended in machine learning best practices (Goodfellow et al., 2016). It ensures that the model has sufficient data to learn from the training set, while allowing for tuning of hyperparameters using the validation set and evaluation of performance on unseen data from the test set.

The final datasets have the following characteristics:

	Cracked Images	**Uncracked Images**	**Resolution**	**Color Channels**
Training	2679	2679	360 x 360	3
Validation	893	893	360 x 360	3
Test	894	894	360 x 360	3

Table 1: Dataset Summary Statistics

2.2.2 Libraries and tools

We rely on several key libraries to build and evaluate the classification model. Python's built-in *os* and *shutil* modules ensure efficient file handling, while *NumPy* provides fast numerical computations (NumPy, n.d.). Data visualization is performed using *Matplotlib*, known for its robust plotting capabilities (Hunter, 2007), and *Seaborn*, which simplifies the creation of aesthetically pleasing statistical visualizations (Waskom, 2021).

The core deep learning pipeline is constructed using *TensorFlow* (Abadi, et al., 2016), along with its high-level API *Keras* (Chollet, 2015). These powerful libraries are favored for their flexibility and scalability in large-scale machine learning.

Finally, *Scikit-learn* complements these tools by facilitating dimensionality reduction with t-SNE and providing reliable evaluation metrics such as confusion matrices and ROC curves, ensuring thorough model assessment (Pedregosa, et al., 2011). Together, these libraries enable efficient and reproducible machine learning workflows.

2.3 Exploratory Data Analysis

2.3.1 Exploring the dataset

We started our analysis with a manual inspection of images from the dataset:

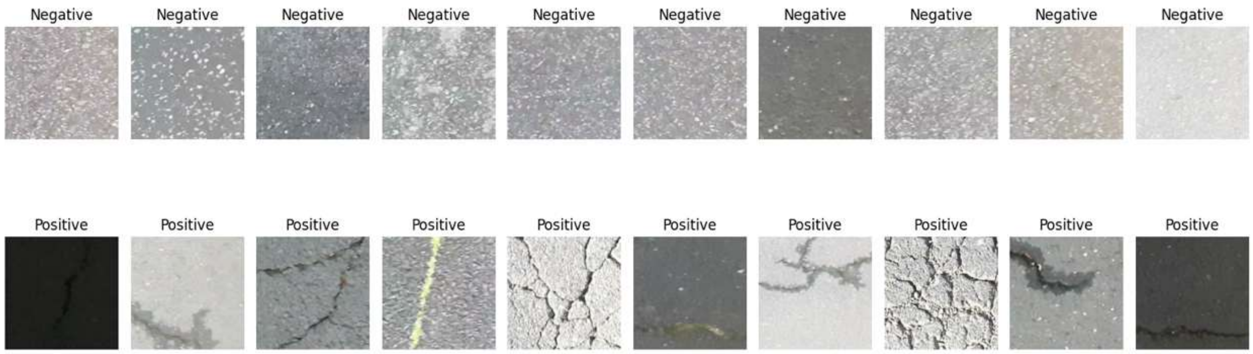

Figure 2: A grid of 10 negative and 10 positive randomly sampled images.

The uncracked (negative) images generally display uniform textures with minimal irregularities, which are crucial for the model to learn a baseline for intact asphalt. Conversely, the cracked (positive) images exhibit distinct fractures, edges, and breaks in the surface, providing clear patterns for classification. Some cracks are subtle, appearing as thin lines, while others are more pronounced, with wider separations and edge erosion. In addition, some images include noise like vegetation and shadows which naturally occur on the road. For instance, see the fallen leaves featured in *Figure 3* from the negative class. Including this sort of naturally occurring noise will likely make the models more robust.

Figure 3: Two images from the negative class where leaves are visible.

2.3.2 Bias and dataset limitations

The dataset presents a few limitations. Most importantly, all images have been captured under relatively consistent weather and lighting conditions on Australian roads. This could limit the model's ability to perform robustly in different climates or seasons, such as Norway's snowy or icy roads during winter.

To compensate for our dataset not being directly comparable to Norwegian roads, we introduce more variance artificially through data augmentation. As cracks can appear in any direction, we introduced random horizontal and vertical flips, as well as random rotations. To adjust for the expected reduced contrast from Norwegian winter roads, we also added a +/- 20% contrast adjustment. Data augmentation techniques like these have been shown to improve model generalization by reducing overfitting (Shorten & Khoshgoftaar, 2019). This is especially important when datasets are relatively small, as in our case.

2.3.3 T-distributed Stochastic Neighbor Embedding

Before using a dataset for classification tasks, it may be useful to examine its suitability by checking whether there are distinguishable clusters in the data. This can be achieved using unsupervised learning techniques to extract features and plot them using T-distributed Stochastic Neighbor Embedding (t-SNE). With this method, we reduce the high-dimensional data in the images into a two-dimensional space, aiming to preserve local structure and similarities between data points. The resulting t-SNE scatter plot can provide insight into whether the extracted features effectively separate the dataset's classes. (Van der Maaten & Hinton, 2008)

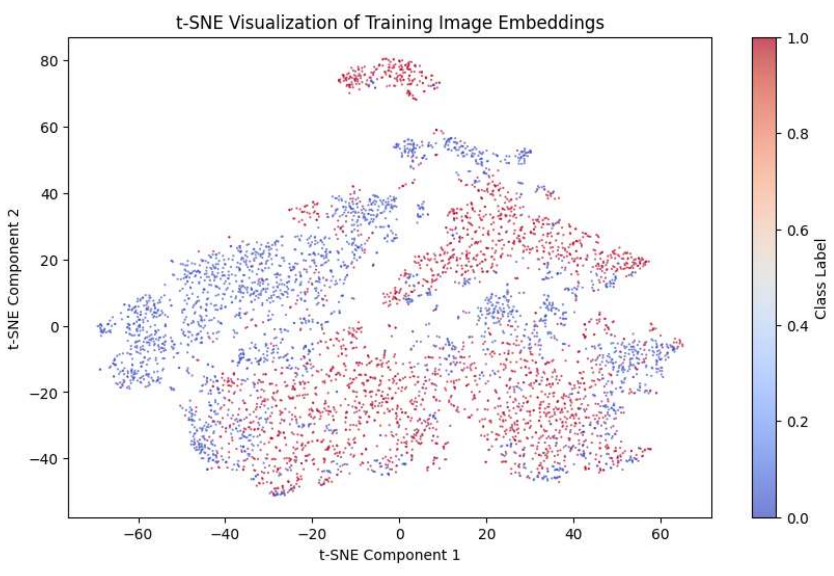

Figure 4: t-SNE Visualization of Training Image Embeddings

Inspecting the t-SNE visualization from our chosen dataset, it appears that the feature extraction process has captured meaningful patterns, as some clusters are clearly formed. Some areas show overlap between classes, suggesting potential misclassification risks. However, we do not believe this is enough to make the dataset unsuitable for our use case.

3 Building AI models

3.1 Why Convolutional Neural Networks?

Convolutional Neural Networks (CNNs) are the gold standard for image-based tasks like classifying asphalt cracks, efficiently extracting spatial features such as edges and textures through their convolutional and pooling layers (Alzubaidi, et al., 2021; Yamashita, et al., 2018). Their strength lies in learning hierarchical patterns directly from raw pixel data, making them great for distinguishing cracks from intact asphalt by using both local details and broader contextual cues (Xu & Xu, 2023).

Because CNNs learn and prioritize features directly from raw input, they eliminate the need for labor-intensive and error-prone manual feature engineering required by methods like Support Vector Machines (Zhao, et al., 2024). This is favorable as manually selecting features can result in discarding features that appear redundant in training data due to high correlation with other variables, yet may be critical for generalization in new conditions, such as varying asphalt textures or lighting. As Guyon & Elisseff (2003) noted, "Very high variable correlation […] does not mean absence of variable complementarity", highlighting that seemingly overlapping features can still offer unique contributions. For instance, a feature omitted during training because it correlates strongly with others, might be essential for detecting cracks in unseen data like texture shifts under different lighting. CNNs avoid these pitfalls by automatically learning and prioritizing a broad set of features from the raw input, ensuring robustness and superior generalization (Abdellatef, Al-Makhlasawy, & Shalaby, 2025).

Moreover, a CNN model is a more reasonable option in this context than the powerful alternative Vision Transformers (ViT), as ViTs require huge datasets and computational power (Ahmad, 2024).

3.2 Choice of CNN Models

Given that CNN models were our ideal choice for image classification, we decided to train three CNN models and evaluate which was the best fit for our project. The first was fully customized from scratch, and the other two models used pre-trained bases (ResNet50V2 and EfficientNetB1).

The rationale for creating a CNN from scratch is that it serves as a pure baseline to assess the learnability of the dataset, free from transfer learning assumptions. Limited computational power does, however, limit the ability to build a very deep model from scratch. A shallower model will likely be less robust, as it is less capable of extracting general features. This makes the model more sensitive to natural objects that appear in the dataset such as the leaves from *Figure 3*. However, this should not impact our results too drastically as our dataset is quite homogeneous and mostly free of noise.

A practical approach to building models with strong generalization capabilities with limited computational power is through transfer learning. This technique involves importing a larger pre-trained model and training a new customized classification head on top of the pre-trained base to tailor the model for a specific task, such as classifying asphalt cracks in our case. As our pre-trained bases are trained on the large ImageNet database, they are better equipped to filter out noise such as leaves on the road (Donges, 2024).

For the first pre-trained model we used ResNet50V2 as a base. This base was chosen due to its use of residual learning, which mitigate the accuracy degradation problem of deep models, outperforming shallower networks like AlexNet (Sharma, Jain, & Mishra, 2018). Additionally, its proven success on the ImageNet dataset, achieving an error rate comparable to human performance, highlights its robustness and reliability for complex image classification tasks (Kundu, 2023).

For the second pre-trained model, the EfficientNetB1 model was chosen due to its compound scaling coefficient that optimizes network depth, width, and resolution, achieving a superior balance of accuracy and efficiency compared to traditional scaling approaches (Le & Tan, 2019).

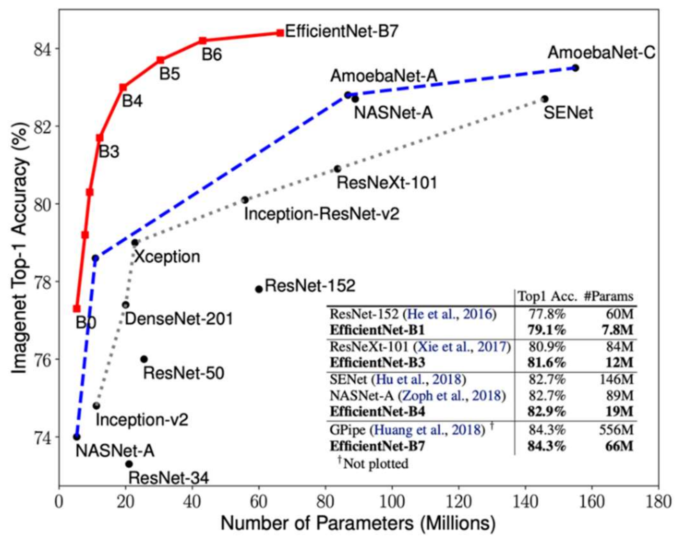

Figure 5: Comparison of popular models' accuracy on ImageNet versus number of parameters from Le & Tan (2019).

As seen in *Figure 5* the EfficientNet models represent an "efficient frontier" in a selection of common pre-trained CNN models. For both ResNet50V2 and EfficientNetB1 there exists significantly larger (and more accurate) models, but for our use case with computational constraints, the smaller versions are more suitable.

3.3 Model Architectures

All three models follow the same overarching architecture outlined in *Figure 6*:

Figure 6: General structure of the CNN models that were developed

The input layer defines the model's structure by specifying the dimensions of the input data and acts as the starting point for processing. In our models, the dimensions are 227x277 pixels with 3 color channels (RGB).

The preprocessing layer prepares the raw data for more efficient learning. Here we apply the data augmentation outlined in Section 2.3.2 to introduce variability to the training set and improve the model's robustness. We also normalize the data to prevent certain features from dominating the training process due to scale differences (Ioffe & Szegedy, 2015). For the custom model, a simple rescaling is sufficient. For ResNet we call the built-in preprocessing method. In the case of EfficientNet, no additional preprocessing steps are needed as the normalization happens automatically within the model pipeline (Keras, 2024). Finally, we prefetch all datasets and use TensorFlow's AUTOTUNE functionality to optimize the data pipeline and improve training efficiency (Abadi, et al., 2016).

The hidden layers extract features of increasing complexity, enabling the model to detect patterns ranging from simple edges to higher-level textures that are crucial for understanding the structure of asphalt cracks. In the custom model, this is achieved through three convolutional layers (with 16, 32, and 64 filters, respectively) that sequentially increase the capacity of the model to learn abstract features. BatchNormalization layers stabilize and accelerate training by normalizing activations (Ioffe & Szegedy, 2015), and MaxPooling2D (2x2) layers reduce spatial dimensions, thus enhancing computational efficiency by focusing on the most prominent features for binary classification.

In the classification head of the custom model, the input is first flattened because it converts the 2D feature maps from the convolutional layers into a 1D vector, enabling the dense layers to process the features for binary classification. In the pre-trained models, global average pooling is used instead of flattening as it reduces the spatial dimensions (height and width) to a single value per feature map, significantly decreasing the number of parameters. For all models, the classification head then includes a dense layer with 64 units and ReLU activation to learn complex, non-linear combinations of the extracted features, specialized for the task of detecting road-cracks. This is followed by a dropout layer to prevent overfitting by randomly deactivating some of the neurons during training. Experimenting with different dropout rates, we found that deactivating 35% of the neurons gave a good trade-off between robustness and accuracy. Finally, there is a dense layer with 1 unit and sigmoid activation to output a probability score for the binary classification of asphalt cracks.

3.4 Training and Compiling

Each model was compiled using the *Adam* optimizer with a learning rate of 1×10^{-3} and optimized for binary classification via the *binary crossentropy* loss function. To enhance training efficiency and robustness, especially given our limited computational resources, a set of base callbacks was implemented. *EarlyStopping* monitors the validation loss and halts training if no improvement is observed after seven consecutive epochs, restoring the best weights to avoid overfitting and unnecessary computations (Prechelt, 1998). *ReduceLROnPlateau* dynamically lowers the learning rate if the validation loss plateaus for three epochs to fine-tune model adjustments (Ioffe & Szegedy, 2015). Together, they accelerate convergence, adapt to learning dynamics, and ensure robust performance, making the model better suited for effective training (Brownlee, 2019).

4 Evaluating AI models

To evaluate the performance of our three models we will use Confusion matrices and ROC-curves.

4.1 Confusion Matrix

The confusion matrix includes True Positives (TP) and True Negatives (TN) for correct predictions of cracked and uncracked asphalt, respectively, as well as False Positives (FP) and False Negatives (FN) for misclassified instances. Based on the relation between these figures, we can calculate key performance metrics including *accuracy*, which indicates the proportion of correct predictions, and *precision*, which shows the proportion of correctly identified cracks among all flagged images. We also report *Recall*, which measures the proportion of actual cracks detected, and the *F1* score, which balances precision and recall, though these are considered less critical for our use-case. The dataset's balanced class distribution (50/50 cracked vs. uncracked images) means there is no need for weighted averages in performance calculations. (Goodfellow, Bengio, & Courville, 2016)

In the trade-off between precision and recall, we find high precision to be especially important for our use case. Testing a subset of misclassified pictures revealed that many of the false negatives were very thin and likely inconsequential cracks. These will likely be detected at a later stage before they can have large economic or safety-related consequences. However, false positives present a more serious challenge, as they undermine the system's reliability and make it harder to achieve the cost savings outlined in our proposed business case. If flagged cracks cannot be trusted, the data becomes unreliable for optimizing maintenance schedules or costly manual oversight will be required.

The three models have the following Confusion Matrices and Performance metrics:

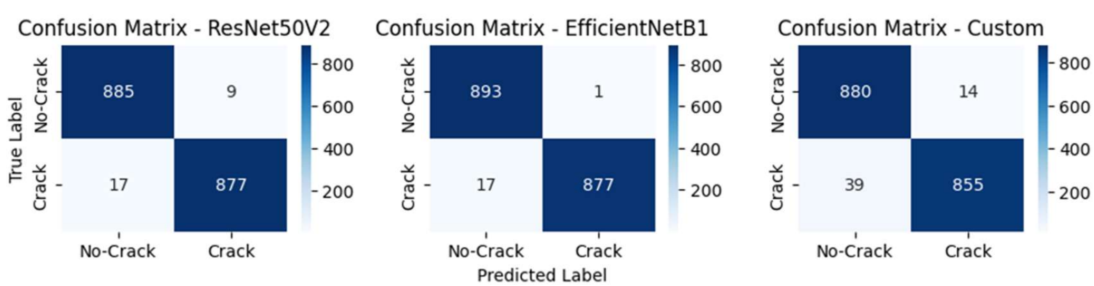

Figure 6: Confusion Matrices for the three models

	Accuracy	Precision	Recall	F1
ResNet50V2	0.9855	0.9898	0.9810	0.9854
EfficientNetB1	0.9899	0.9989	0.9810	0.9898
Custom CNN	0.9704	0.9839	0.9564	0.9699

Table 2: Performance metrics for the three models

Evaluating the three models, we see that EfficientNetB1 outperforms across all performance metrics except recall. With an accuracy of 0.9899 and precision of 0.9989 it is capable of distinguishing between cracked and non-cracked asphalt near perfectly and with only one false positive from our test set. ResNet50V2 also performs very well, but marginally worse than EfficientNetB1. These results indicate that the findings of Tan & Le (2019) are evident also for our dataset of asphalt images, where EfficientNetB1 displays superior performance with fewer parameters.

The custom CNN also performs remarkably well, with performance metrics only slightly lower than its pre-trained peers. We do however expect this discrepancy to increase as the dataset gets bigger and less homogenous. In this case, the pre-trained models will most likely benefit greatly from their deeper networks and ability to extract general features.

Regarding potential overfitting, we do not have reason to believe the models' performance should improve by further increasing dropout or regularization, as all three models exhibit very similar validation and test accuracies. Rather, the exceptionally high performance may be tied to what is in-domain versus out-of-domain, which will be discussed in section 4.4.

	Training Accuracy	Validation Accuracy	Test Accuracy
ResNet50V2	0.9566	0.9860	0.9855
EfficientNetB1	0.9830	0.9916	0.9899
Custom CNN	0.9624	0.9687	0.9704

Table 3: Training, Validation and Test accuracies for the three models

4.2 ROC Curves

ROC (Receiver Operating Characteristic) curves are graphical representations of a model's performance across different classification thresholds. They plot the rate of TPs to FPs, which provides insight into the trade-off between correctly identifying positive instances and incorrectly classifying negatives as positives. The Area Under the Curve (AUC) summarizes this into a single value, ranging from 0 to 1, where a higher AUC indicates better discrimination between the positive and negative classes. (Fawcett, 2006)

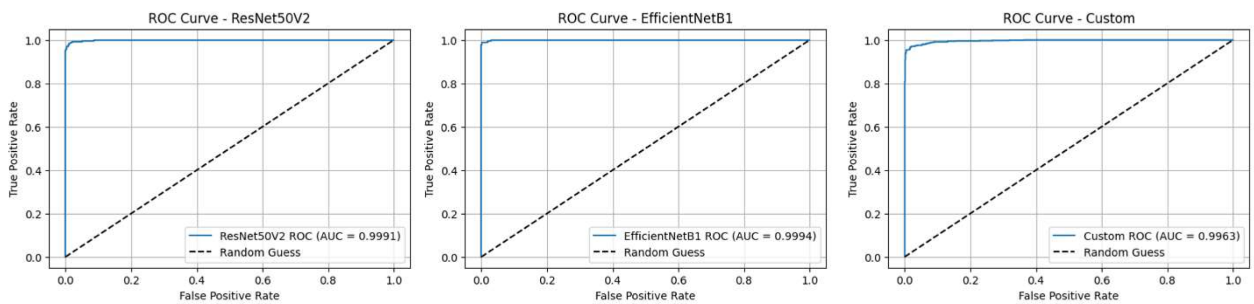

Figure 7: ROC-curves for the three models

Evaluating the models' ROC curves, all models achieve near-perfect performance with an AUC close to 1, reflecting exceptional discrimination capabilities at all thresholds. EfficientNetB1 has the greatest performance with an AOC of 0.9994, followed closely by ResNet50V2 (AUC = 0.9991) and then the custom CNN (AUC = 0.9963). This indicates that EfficientNetB1 has a slightly superior rate of true positives to false positives, which aligns with the models' previously observed accuracy and precision metrics.

4.3 Summary

In summary, we find that all three models, and especially EfficientNetB1, perform exceptionally well at classifying asphalt cracks across all performance measures. The custom CNNs performance is especially impressive given its shallower architecture.

To explain this exceptional performance, it is worth noting that detecting cracks in asphalt is a relatively simple task, especially when dealing with our relatively homogeneous dataset. While the use of a full, deep, pre-trained network may be excessive for this dataset, it positions the models well for future scalability. As Statens Vegvesen collects larger and more diverse datasets over time, we expect the deeper, pre-trained models to better adapt to increased complexity and variability, while the simpler custom CNN likely will see a worsened relative performance.

4.4 Limitations

The evaluation of performance measures shows that the three models, and especially EfficientNetB1, perform well in-domain. However, as the models have been trained on a relatively small Australian dataset, variance specific to Norwegian roads are currently out-of-domain. For instance, none of the pictures in the current dataset include ice, snow, grit or wet leaves, which are all common in Norway during fall and winter. To partially compensate for this, we introduced noise through data augmentation and used moderately high dropout rates for all models and L2 regularization for the pre-trained models to reduce overfitting. However, this does guarantee that the models will have exceptional performance on Norwegian roads if implemented as is.

To overcome these limitations, we propose that Statens Vegvesen retrain the models with labelled images after one year of data collection to control for Nordic conditions throughout all seasons. With a larger dataset including more naturally occurring noise, there is less need for artificial data augmentation, meaning this step may be skipped. Moreover, with a larger sample size, Statens Vegvesen may experiment with reducing the dropout rates and L2 Regularization to optimize for the new dataset. If they encounter problems with low precision, they may consider increasing the classification threshold above 0.5 to avoid false negatives. If they encounter problems with low accuracy, upgrading to one of the larger EfficientNet models like B6 or B7 may solve the issue, granted that the increased accuracy can justify the more computationally heavy training.

5 Business Application

5.1 Potential Cost Savings

Automated road crack detection offers a practical solution to reduce the cost and inefficiencies of manual inspections. With Norway, as previously mentioned, allocating 10.8 billion NOK to road maintenance in 2025, there is significant potential for cost savings (Regjeringen, 2025). Utilizing ML-models to detect road damage allows for early intervention, optimized resource allocation and reduction in labor cost. Early interventions are beneficial because they address minor damages before they escalate and become major repairs. Studies demonstrate that preventive maintenance strategies can save up to 50% in road repair costs (BITRE, 2023).

Beyond financial benefits, a streamlined maintenance process will also benefit private vehicle owners and companies operating on roads. According to NAF, 40% of drivers in Norway have experienced damages to their vehicle due to faulty communal roads. The related repair costs averaged 17 000 NOK per vehicle (Castle, 2023). Major road repairs often necessitate closures that reduce lane availability, generate congestion, divert traffic onto irregular routes, and increase the risk of accidents. By contrast, timely intervention extends the lifespan of road infrastructure and minimizes the need for such large-scale, traffic-disrupting repairs.

As a result, improved road quality enabled by data-optimized maintenance strategies can generate cost savings on two fronts: reducing public expenditure on infrastructure and lowering private costs associated with vehicle wear, delays, and accidents.

5.2 Practical Implementation

The practical implementation of using ML for road inspection first involves collecting high quality data. Once trained, the models can efficiently identify cracks and automate the inspection process. Cameras for continuous monitoring can be mounted on frequently used vehicles used such as public transport, delivery vehicles and taxis (Tello-Cifuentes, Acero, Marulanda, Thomson, & Barona, 2024). The data is used by road maintenance bureaus to accurately map damages, optimizing their schedules and projects. By becoming integrated into monitoring systems, maintenance crews can also rapidly respond to damages and prioritize by urgency in real time.

To gain access to a larger fleet of vehicles for road-monitoring, Statens Vegvesen could collaborate with private companies operating bus, delivery, rental and taxi vehicles. Since their businesses depend on roads, it is in their best interest to collaborate on improving road conditions. Once implemented, Statens Vegvesen and other transport bureaus can plan and prioritize spending more accurately within constrained budgets. The ML model can also be used in all areas where road quality is essential, such as for military mobility and urban infrastructure planning. In the future, a large fleet of vehicles may not even be required, as ML-models could potentially be trained on satellite imagery to assess road conditions on a broader scale (Thegeya, et al., 2022).

5.3 Viability

For the project to be a viable business case, its contributed value should be higher than its cost to develop and operate. In section 5.1 we have argued that the potential cost savings are significant, including positive externalities from well-maintained roads such improved accessibility, reduced vehicle repair costs, and better road safety. With public road maintenance expenditures steadily increasing, the efficiency gains from this solution will likely continue to grow over time. Meanwhile, the implementation costs for Statens Vegvesen are likely minimal, covering tasks like mounting cameras, partnering with private companies, and managing data collection and analysis. As Statens Vegvesen already has a practice of collecting and organizing large amounts of data in their data portal (Statens Vegvesen, 2025), they are well- suited for successfully incorporating the proposed system into their organization and unlocking more efficient road maintenance across Norway.

6 Conclusion

This project evaluates the potential of using machine learning to continuously monitor road cracks and optimize public road maintenance. Using a dataset of images depicting cracked and uncracked asphalt collected by Baduge et. al. (2023), we trained and tested three models: two pre-trained models using ResNet50V2 and EfficientNetB1 as bases, and a custom CNN model. Evaluating the performance using confusion matrices and ROC-curves, EfficientNetB1 was clearly the highest performing model, achieving near perfect accuracy and precision, closely followed by ResNet50V2 and the custom CNN model. When deployed on a larger and more varied dataset, we expect these differences in performance to be further extenuated, with the shallower custom CNN falling further behind.

The proposed ML-models present an attractive business case, as they enable cheaper and more efficient road monitoring, which is crucial for optimizing road maintenance overall. By transitioning from traditional inspection methods to ML-based systems, Statens Vegvesen could use continuously updated data to prioritize road repairs more effectively, reducing infrastructure deterioration and accident risks. Implementation would involve equipping vehicles with cameras, automating image collection, and integrating trained models into existing workflows for real-time monitoring. Over time, the system would continuously improve by adapting to larger datasets and include environmental complexities unique to Norwegian roads, such as snow and ice degradation. As Statens Vegvesen are well positioned to implement such a system, integrating ML models for road monitoring is a low hanging fruit with significant positive potential like reduced costs, improved road quality and safety.

7 References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., . . . Steiner, B. (2016). TensorFlow: A system for large-scale machine learning. *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16)*.

Abdellatef, E., Al-Makhlasawy, R. M., & Shalaby, W. A. (2025). Detection of human activities using multi-layer convolutional neural network. *Scientific Reports*.

Ahmad, A. (2024). *What is a Vision Transformer (ViT)? What are ViT's Real-World Applications?* Retrieved from SJ Innovation: https://sjinnovation.com/what-vision-transformer-vit-what-are-vits-real-world-applications

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., . . . Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*.

Baduge, K. K., Thilakarathna, S., Perera, ,. J., Ruwanpathirana, G., Doyle, L., Duckett, M., . . . Mendis, P. (2023). Assessment of Crack Severity of Asphalt Pavements using Deep Learning Algorithms and Geospatial System. *Construction and Building Materials*. Retrieved March 23, 2025, from https://data.mendeley.com/datasets/88kdyyc73h/1

Baduge, K. K., Thilakarathna, S., Perera, J. S., Ruwanpathirana, G., Doyle, L., Duckett, M., . . . Mendis, P. (2023, August 8). *Asphalt Cracked and Uncracked Image Dataset.* Retrieved March 20, 2025, from Mendeley Data: https://data.mendeley.com/datasets/88kdyyc73h/1

BITRE. (2023). *The Economics of Road Maintenance.* Canberra: BUREAU OF INFRASTRUCTURE AND TRANSPORT RESEARCH ECONOMICS.

Brownlee, J. (2019). *A Gentle Introduction to Early Stopping to Avoid Overtraining Neural Networks - MachineLearningMastery.com*. Retrieved March 31, 2025, from MachineLearningMastery.com: https://machinelearningmastery.com/early-stopping-to-avoid-overtraining-neural-network-models/

Castle, E. (2023). *Har vegvesenet ansvar for hull i veien?* Retrieved from NAF: https://www.naf.no/trafikksikkerhet/lover-regler/veimyndighetenes-ansvar-for-skade

Chollet, F. (2015). *Keras*. Retrieved April 3, 2025, from GitHub: https://github.com/fchollet/keras

Cifuentes, L. T. (2025). Detection and classification of pavement damages using wavelet scattering transform, fractal dimension by box-counting method and machine learning algorithms. *Road Materials and Pavement Design*. doi:10.1080/14680629.2023.2219338.

Donges, N. (2024). *What Is Transfer Learning? Exploring the Popular Deep Learning Approach.* Retrieved March 25, 2025, from BuiltIn: https://builtin.com/data-science/transfer-learning

ei Zhang, F. Y. (2016). Road Crack Detection Using Deep Convolutional Neural Network. *2016 IEEE International Conference on Image Processing (ICIP)*. Retrieved from http://doi.org/10.1109/ICIP.2016.7533052

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters, 27*, 861-874. Retrieved April 5th, 2025, from https://www.sciencedirect.com/science/article/pii/S016786550500303X

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning.* The MIT Press.

Guyon , I., & Elisseff, A. (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*.

Ha, J., Kim, D., & Kim, M. (2022). Assessing severity of road cracks using deep learning-based segmentation and detection. *Journal, 78*, 17721–17735. doi:https://doi.org/10.1007/s11227-022-04560-x

Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science and Engineering, 9*, 90-95. Retrieved April 6, 2025, from https://ieeexplore.ieee.org/document/4160265

Ioffe, S., & Szegedy, C. (2015). *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift*. Retrieved March 23, 2025, from Cornell University ArXiv: https://arxiv.org/abs/1502.03167

Jayanth Balaji A, T. B. (2019, April 26). *Mendeley Data*. Retrieved from Asphalt Crack Dataset: https://data.mendeley.com/datasets/xnzhj3x8v4/2

Keras. (2024, 06 07). *Github.* Retrieved from TensorFlow v2.16.1 Documentation: https://www.tensorflow.org/api_docs/python/tf/keras/applications/efficientnet/preprocess_input

Kundu, N. (2023). *Exploring ResNet50: An In-Depth Look at the Model Architecture and Code Implementation*. Retrieved from Medium: https://medium.com/@nitishkundu1993/exploring-resnet50-an-in-depth-look-at-the-model-architecture-and-code-implementation-d8d8fa67e46f

Le, Q., & Tan, M. (2019). *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.*

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (2002). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 2278 - 2324. doi:10.1109/5.726791

NAF. (2024). *NTB Kommunikasjon*. Retrieved from Dårlige veier øker risikoen for dødsulykker: https://kommunikasjon.ntb.no/pressemelding/18341500/darlige-veier-oker-risikoen-for-dodsulykker?publisherId=2126680&lang=no

NumPy. (n.d.). *NumPy documentation Version 2.2*. Retrieved April 6, 2025, from https://numpy.org/doc/2.2/

Özgenel, Ç. G. (2018). Performance Comparison of Pretrained Convolutional Neural Networks on Crack Detection in Buildings. *ISARC*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . P. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*.

Prechelt, L. (1998). Early Stopping - But When? In G. Orr, & K. Müller, *Neural Networks: Tricks of the Trade.* Berlin: Springer. doi:https://doi.org/10.1007/3-540-49430-8_3

Regjeringen. (2024). *1,1 milliardar kroner meir til drift og vedlikehald av riksvegane*. Retrieved March 28, 2025, from Statsbudsjettet 2025: https://www.regjeringen.no/no/aktuelt/statsbudsjettet-2025-11-milliardar-kroner-meir-til-drift-og-vedlikehald-av-riksvegane/id3056057/#:~:text=Regjeringa%20foresl%C3%A5r%20%C3%A5%20l%C3%B8yva%2010,kroner%20til%20vedlikehald%20av%20riksveg.

Regjeringen. (2025). *Statsbudsjettet 2025: A-Å*. Retrieved March 31, 2025, from https://www.regjeringen.no/no/statsbudsjett/2025/a-aa/id3052994/

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4510-4520. doi:10.1109/CVPR.201

Sharma, N., Jain, V., & Mishra, A. (2018). An Analysis Of Convolutional Neural Networks For Image Classification. *Procedia Computer Science*.

Shorten, C., & Khoshgoftaar, T. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data|, 60*.

Statens Vegvesen. (2024). *Asfaltering av riksveier*. Retrieved from https://www.vegvesen.no/trafikkinformasjon/langs-veien/drift-og-vedlikehold-av-vei/asfaltering-av-riksveier/

Statens Vegvesen. (2025). *Drift og vedlikehold av vei*. Retrieved from https://www.vegvesen.no/trafikkinformasjon/langs-veien/drift-og-vedlikehold-av-vei/#:~:text=Statens%20vegvesen%20har%20ansvar%20for%20drift%20og%20vedlikehold%20av%20st%C3%B8rstedelen%20av%20riksveinettet.

Statens Vegvesen. (2025, April 7). *Om Dataportalen*. Retrieved from Dataportalen: https://dataut.vegvesen.no/about

Szegedy, C., & Ioffe, S. (2015). Batch normalization: accelerating deep network training by reducing internal covariate shift. *Association for Computing Machinery*.

Tello-Cifuentes, L. (2025). Detection and classification of pavement damages using wavelet scattering transform, fractal dimension by box-counting method and machine learning algorithms. *Road Materials and Pavement Design*. doi:10.1080/14680629.2023.2219338.

Tello-Cifuentes, L., Acero, S., Marulanda, J., Thomson, P., & Barona, J. J. (2024). Implementation of a low-cost comprehensive pavement inspection system. *Transportation Engineering*.

Thegeya, A., Mitterling, T., Martinez, J. A., Bulan, J., Durante, R. L., & Mag-atas, J. (2022). APPLICATION OF MACHINE LEARNING ALGORITHMS ON SATELLITE IMAGERY FOR ROAD QUALITY MONITORING. *ADB Economics Working Paper Series*. Retrieved April 07, 2025, from ADB Economics Working Paper Series: https://www.adb.org/sites/default/files/publication/849926/ewp-675-machine-learning-satellite-image-road-quality.pdf

Van der Maaten, L., & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research, 9*, 2579-2605.

Waskom, M. L. (2021). Seaborn: Statistical Data Visualization. *The Journal of Open Source Software*, 3021.

WHO. (2023). *Road traffic injuries*. Retrieved from World Health Organization: https://www.who.int/news-room/fact-sheets/detail/road-traffic-injuries

Xu, G., & Xu, G. (2023). Using A CNN to Solve the Problem of Asphalt Pavement Crack Detection. *Association for Computing MAchinery*.

Yamashita, R., Nishio, M., Do, R. K., & Togashi, K. (2018). Convolutional neural networks: an overview and application in radiology. *Insights into Imaging*.

Zhao, X., Wang, L., Zhang, Y., Han, X., Deveci, M., & Parmar, M. (2024). A review of convolutional neural networks in computer vision. *Artificial Intelligence Review*.