Dara Golden

Student number: 19432242

Role number: 1

Contribution: Conducted and wrote the Literature review, utilised the acquired knowledge to come up with the proposed model. Completed the implementation of the proposed model using pytorch as a combination of ideas from constituent papers. Analysed the dataset distribution, recreated a more balanced dataset that was used for training, and wrote the dataset section of the report. Completed references for the report.

Lukasz Szemet

Student number: 19502109

Role number: 2

Contribution: Designed a front-end webapp that allows interaction with our trained models using the Streamlit framework. Abstract and Webapp sections of report.

Aaron Brennan

Student number: 19402624

Role number: 3

Contribution: I took over from Dara after he investigated the proposed model(s). I trained these various models and gathered training, validation and testing losses. I documented, graphed and analysed the results after the models finished training. Wrote the Introduction and Conclusion also.

Abstract

In this report we present our unique machine learning model for road surface classification, where we distinguish between 3 classes: Asphalt, Gravel and Concrete. The ability to distinguish different road types can improve the safety and reliability of autonomous vehicles.
We propose a convolutional model with additional attention layers to enhance its effectiveness. We compare our model against various state-of-the-art classification models, showing that our model can surpass their performance, or display a similar performance while having less trainable parameters, making it smaller and faster. We also provide a web-app that allows users to interact with our model and compare it against other models.

Introduction

This case study focuses on predicting road surfaces using a machine learning model. This task is important for improving road safety, such as predicting surface conditions. The customer has asked us to create a machine learning model that is unique and can be used in a desktop or web application. The model must be unique to avoid any legal issues with intellectual property. The model will take images as input and classify them into one of three categories: Asphalt, Concrete or Gravel.

We first examined the literature to understand previous approaches taken to the problem of road surface classification. We weren't surprised to find that deep convolutional models such as ResNet or Inception were commonly used for this image classification problem, as these types of models have become the state of the art within image classification. We also investigated the use of attention blocks within a deep convolutional network as a potential innovation to a standard model such as ResNet.

In section 3 we examined and processed the dataset. After visually inspecting and collecting statistics on the data we addressed imbalances in the dataset by sampling the three classes with an equal frequency.

Next, we began experimenting with different model architectures. To get a baseline result we first tried a shallow CNN, which consisted of a few convolutional layers followed by a few fully connected layers. Then from the literature review we tried some well-known image classification models tried by others in this domain. We experimented with ResNet, Inception and VGG models. Finally, we tried our unique custom model which combines standard convolutional blocks with attention blocks. Since the customer insists on a unique model, this will be the one used in production.

In section 5 we evaluate the results of our various models for differing hyper-parameters. We analysed the results and explained the difference in performance among the models. In the next section we cover all the details of making a functional web application which provides a user-interface to experiment with the model. This involved exporting the model to ONNX format, writing the required HTML and JavaScript code and deploying the server.

Finally, we conclude the case study in section 7, where we go over what worked well, what we learned and the future directions in which this study can be taken.

Literature Review

The problem of road surface classification is being raised in the field of autonomous driving and intelligent vehicles for the estimation of conditions to enhance intelligent control. Some methods implemented work reactively through measuring sound, vibrations, speed, location, and other features based on the road currently being driven on. This allows vehicles to adjust to the current conditions but not to proactively plan for future road sections. This is where front facing image-based approaches can detect road sections out ahead of the vehicle and classify the road type to make proactive decisions about vehicle control. [1]

Traditional computer vision-based methods in this area utilise simple machine learning classification methods based on hand crafted features extracted from regions of interest. For example, the work of Qian et al utilised pixel luminance and texture information taken from an estimated region of interest to classify road surfaces. In their work they worked on tasks with 2, 3 and 5 classes achieving accuracies of 80%, 68% and 46% respectively. This work however was conducted on a very small private dataset of only 100 images which would be an insufficient amount for most computer vision or deep learning methods [2].

An early work applying deep learning to road surface classification was done by Slavkovikj et al in 2014. In their work they utilised pretrained residual based deep learning models in Inception V3 and ResNet50 and retrained them on road type classification. They achieved an accuracy score of about 90% on a dataset with 6 different classes that had a base of 700, 500 and 300 images in the training, validation and testing sets respectively [3]. This dataset was still small relative to what is required for training a deep neural network from scratch, but it shows the use case of transfer learning and careful optimisation to produce good results despite a smaller dataset.

In recent years a number of datasets that are sufficiently large enough for deep learning have been made publicly available such as the Oxford RobotCar dataset with over 20 million images [4] or the Road Surface Classification Dataset with over 1 million images (RSCD) [5]. This massive increase in available data allowed for the development of deep learning based solutions to the road surface classification problem.

For example, Dewangan et al achieves an accuracy and precision of 99.90% in a 5 class classification task over a subset of the Oxford RobotCar dataset using a deep convolutional neural network. The five classes they aim to categorise are: curvy, dry, ice, rough, and wet roads [6]. The main proposal of this work was the introduction of a model architecture designed with extensive empirical analysis of the optimiser, batch size and kernel size. The authors proposed this model as a base line for the intelligent vehicle road surface classification field.

Attention mechanisms have become ubiquitous in the field of machine learning and are showing promise in the field of computer vision as a method of improving model performance. For example, Woo et al propose a convolution block attention module as an intermediary layer for any convolutional layer. They propose modules for both spatial and channel-based attention that can be combined together. In their work they verified that the implementation of CBAM into various state of the art models outperformed the base models on the ImageNet-1K, MS COCO and VOC 200 benchmark datasets [7].

There are some examples of attention-based methods being implemented in the road surface classification space such as the work of Guo et al who implemented an attention based ReXNet on the RSCD dataset. This work proposes a new model implementing attention and a new balanced softmax cross entropy criterion to improve performance and address imbalances in the RSCD dataset. This model achieved 87.67% precision and 88.52% top1 accuracy on the RSCD dataset [8].

Road surface classification is an emerging field with relatively little published literature to date that can have a huge impact on the future development of intelligent vehicles. The current benchmark datasets dominating in the space are the Oxford RobotCar and RSCD datasets. Detailed above are some examples of state-of-the-art deep learning methods for this problem. Given the relative sparsity of research there are many opportunities for future research but the one we will be pursuing in the development of a novel model is the implementation of attention into deep convolutional networks.

Dataset

Original dataset.

The dataset we are working with is a subset of the RSCD dataset specifically 3 classes of it which are:

- Wet Asphalt Smooth
- Wet Concrete Smooth
- Wet Gravel

The original dataset that was provided to us had the following distribution.

Class	Train	Validation	Test	Total Samples
Wet Asphalt Smooth	79404	79	821	80304
Wet Concrete Smooth	66955	160	821	67936
Wet Gravel	36515	2351	821	39687

Table 1: Number of Samples in each original class and set.

Class	Train	Validation	Test
Wet Asphalt Smooth	98.8793%	0.0984%	1.0224%
Wet Concrete Smooth	98.5560%	0.2355%	1.2085%
Wet Gravel	92.0075%	5.9239%	2.0687%

Table 2: Percentage of each class in original Train, Test and Validation sets.

Class	Train	Validation	Test
Wet Asphalt Smooth	43.42006%	3.050193%	33.33333%
Wet Concrete Smooth	36.61264%	6.177606%	33.33333%
Wet Gravel	19.9673%	90.7722%	33.33333%

Table 3: Percentage of each original Set made up by each class.

Training set

We can see both from the total number of samples in each class and the percentage makeup of the training set that it is imbalanced. This is an issue we will have to address if we want the model to be able to accurately learn the minority classes, especially gravel.

Validation set

The validation set makes up less than 1% of the asphalt and concrete classes and roughly 6% of the gravel class. These percentages are too low and show that the validation set does not make up a large enough percentage of the total data. For the validation set to be an accurate representation of the data distribution it should contain 10% or more of the data.

Another issue with the validation set is the massive imbalance in the number of samples from each class with gravel making up over 90% of the total samples. Given we will be using the validation set to select the best performing models a much more balanced set of samples will have to be chosen.

Testing set

Similar to the validation set the testing set needs to be an accurate representation of the dataset. 2% or less of each class has been assigned for the testing set which is insufficient for testing metrics to be an accurate measure of model performance.

However, the equal distribution of data in the testing set with each class having an equal number of samples is the ideal situation for both the test and validation sets to allow them to accurately model the different classes.

There are several issues that we needed to address before we could use this data as a means for training deep learning models. First the amount of data in the validation and test sets needs to be increased to 10% or more to provide an accurate test and validation metrics. Secondly, the balance of data in the training and validation sets must be improved.

Proposed dataset

To address the issues identified in the original dataset analysis, several measures were implemented to ensure a balanced distribution of classes within the validation and test sets and a balanced distribution across all 3 sets. To achieve equal representation across classes in these sets, 10% of the minority class, which is Wet Gravel in this case, was selected as the target number of samples from each class to constitute these sets.

This approach ensures that each class is equally represented in both the testing and validation sets, thereby providing an accurate representation of all classes, and facilitating precise evaluation metrics during testing and validation. This strategy represents an improvement over the alternative of selecting 10% from each class, as such an approach would have resulted in imbalanced testing and validation sets, skewing the evaluation process.

Furthermore, to tackle the class imbalance within the training set while still utilising the entirety of available samples for training, a weighted random sampler was employed within the PyTorch training data loader. This weighted random sampler, facilitates the retrieval of samples with replacement during each epoch in a manner that approximates an equal representation of each class. By incorporating this sampler, the training process is optimized to effectively learn from all classes, mitigating biases towards overrepresented classes and enhancing the model's generalisability.

Class	Train	Validation	Test	Total Samples
Wet Asphalt Smooth	72366	3969	3969	80304
Wet Concrete Smooth	59998	3969	3969	67936
Wet Gravel	31749	3969	3969	39687

Table 4: Number of Samples in each proposed class and set.

Class	Train	Validation	Test

Wet Asphalt Smooth	90.1151%	4.9425%	4.9425%
Wet Concrete Smooth	88.3155%	5.8423%	5.8423%
Wet Gravel	79.9985%	10.0008%	10.0008%

Table 5: Percentage of each class in proposed Train, Test and Validation sets.

Class	Train	Validation	Test
Wet Asphalt Smooth	44.09523%	33.33333%	33.33333%
Wet Concrete Smooth	36.55896%	33.33333%	33.33333%
Wet Gravel	19.34582%	33.33333%	33.33333%

Table 6: Percentage of each proposed Set made up by each class.

Data Preprocessing

Normalisation

To combat the issue of exploding gradients and make it easier for the models to learn from the data all the pixels in the images were normalized between the values of 0 and 1.

Resizing

The different models utilised in the experiments section of this report both the pretrained models and models defined by us had different resolution requirements. The resolutions required are detailed in the table below. In order to improve the speed of training a version of the proposed dataset was written at a resolution of 96x64.

Model	Resolution
shallow_cnn_model	96x64
inceptionV3_model	299x299
vgg_model	224x224
resnet_model	96x64
RCNet_model	96x64
RCNet_attention_model	96x64

Table 1: Image resolutions required by each classifier

Proposed Model

It is clear from reviewing the literature deep convolutional neural networks are the path forward in the context of road surface classification. Models like RCNet [6] have shown great promise on the Oxford RobotCar dataset [4] and given the volume of data available to us in the dataset we are proposing to train a deep convolutional neural network to address the problem presented. We propose a network inspired by the architecture proposed in the RCNet[6] paper but with the additional novelty of enhancement through the implementation of attention. The specific attention mechanisms we propose utilising are the Convolutional block attention modules (CBAM) proposed by Woo et al [7]. These attention modules were chosen for a number of reasons, firstly because they attend both based channels and based on spatial information. They were chosen secondly for their ease of implementation into existing convolutional architectures and their ability to be trained at the same time as the main network.

A total of 4 CBAMs were added to our implementation of the RCNet network along with several additional dropout layers. These dropout layers were added based on empirical evidence in the training of the RCNet that showed it was overfitting on the training set in an attempt to improve the validation and test performance of the proposed model. The 4 CBAMs were added after Convolution 2, 4, 6, and 8 allowing the model to consistently attend to the important information.

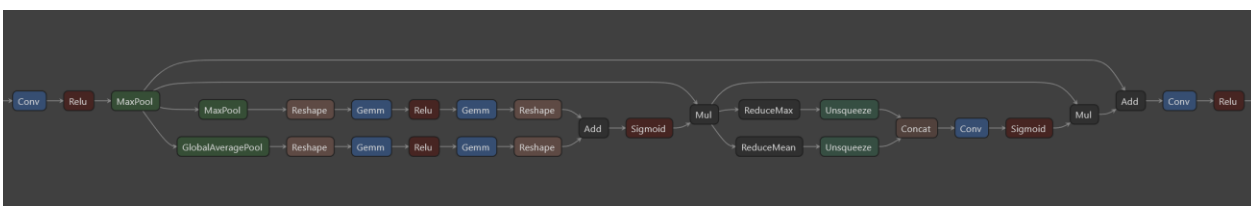


Figure 1: Visualisation of the operations in a CBAM between 2 convolutional layers.

The CBAMs are made up of two parts each of which produce a mask. The first of these masks attends to the channels and is multiplied with the output of the convolutional layer. The second mask attends spatially and is multiplied by the output from the first mask. These are finally combined with an addition residual connection from before the convolutional block. The reason the blocks are organised in the order channel then special is due to empirical testing done by the original model authors. [7]

Experiments/results & Analysis

We evaluated 6 different models as part of our experimentation:

1. Shallow CNN
2. Inception net
3. VGG16
4. ResNet18
5. RC Net
6. RC Net with attention

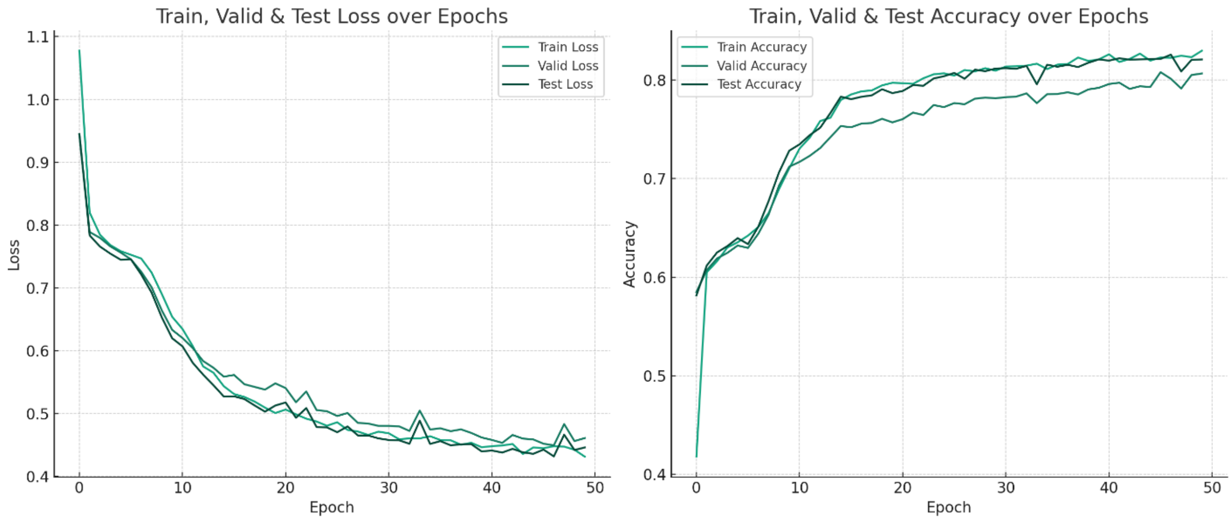
I will go through the results of each model one by one

Shallow CNN:

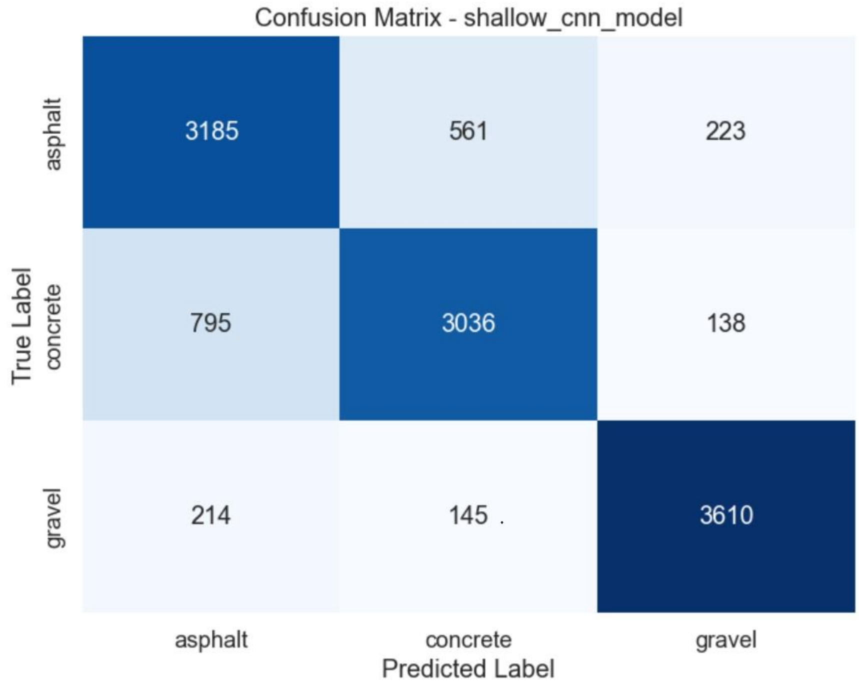
The model had the following training details:

- Learning Rate: 1e-05
- Epochs: 50
- Batch Size: 64
- Total parameters: 2492835
- Trainable parameters: 2492835

The loss and accuracy graphs looked as follows:



The confusion matrix on the test set:

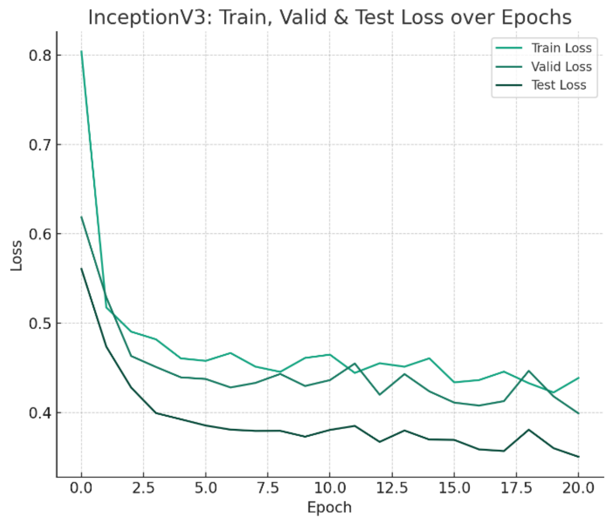


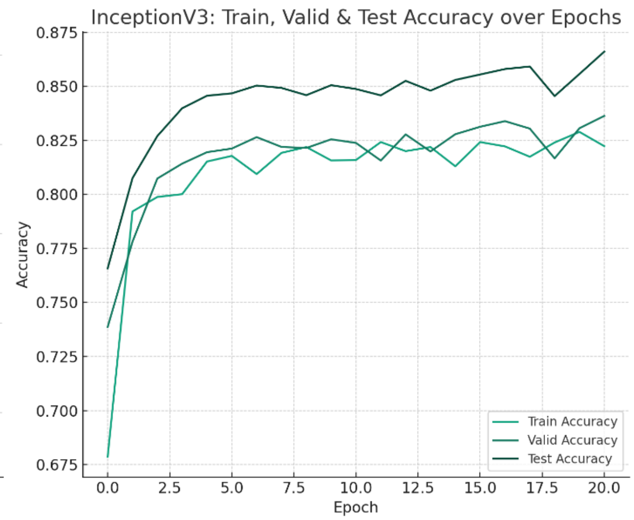
InceptionV3[9]:

The model had the following training details:

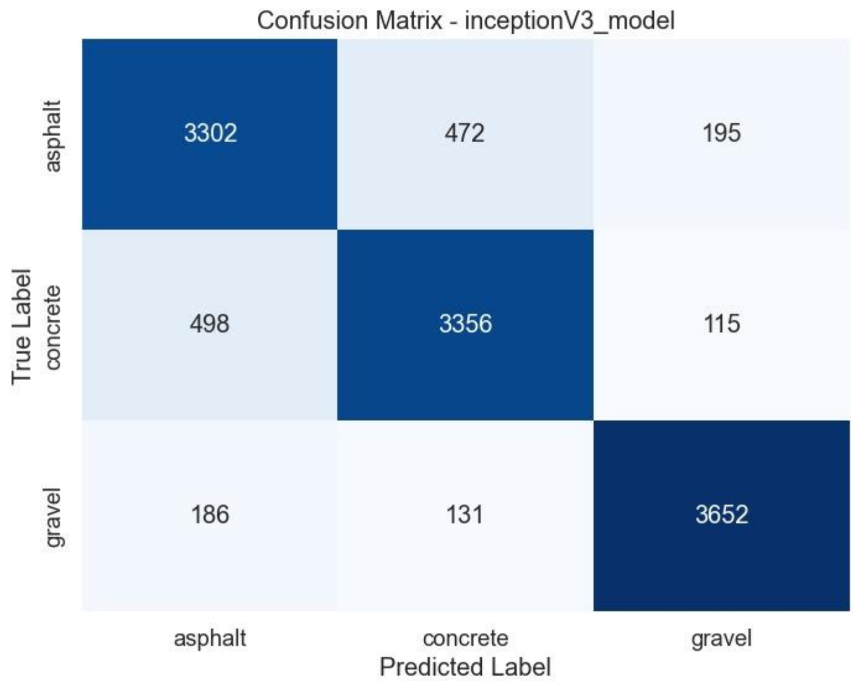
- Learning Rate: 0.0001
- Epochs: 20
- Batch Size: 128
- Total parameters: 26293451
- Trainable parameters: 1181187

The loss and accuracy graphs looked as follows:





The confusion matrix on the test set:



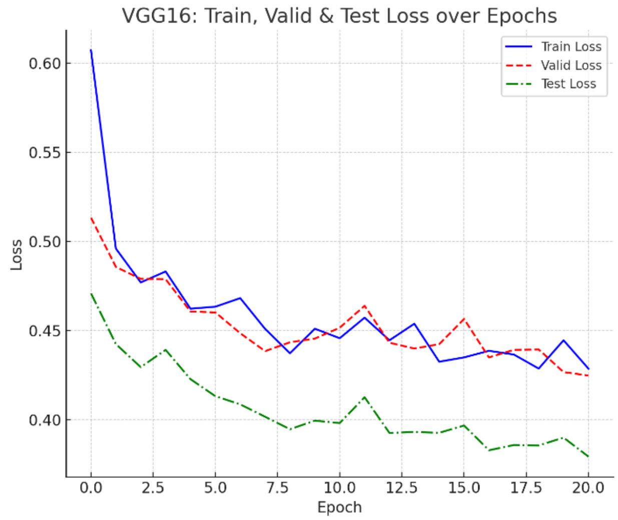
VGG16[10]:

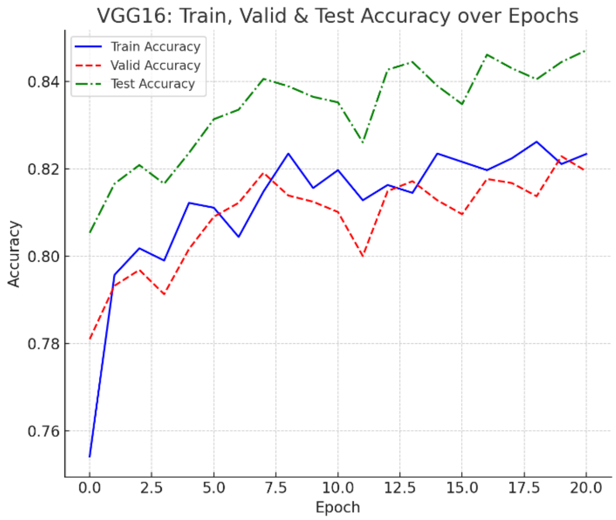
The model had the following training details:

- Learning Rate: 0.0001
- Epochs: 20
- Batch Size: 64

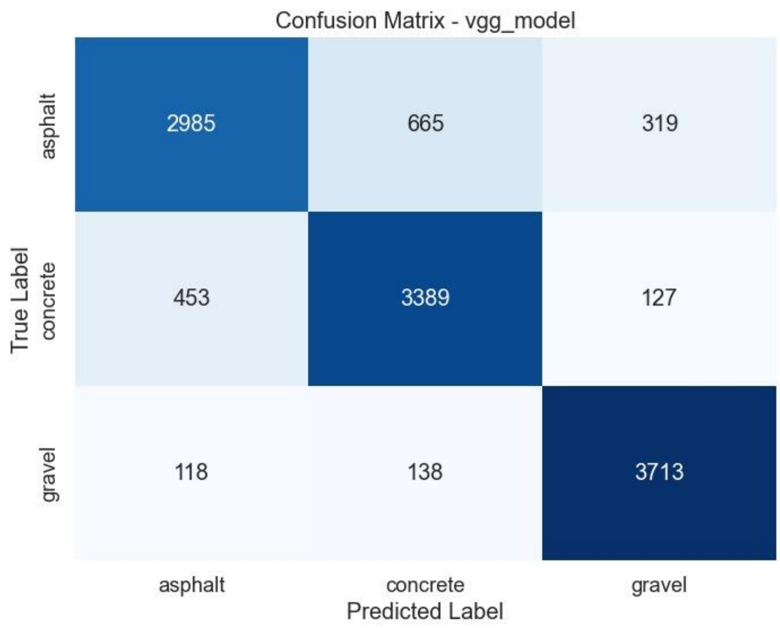
- Total parameters: 136490307
  Trainable parameters: 2229763

The loss and accuracy graphs looked as follows:





The confusion matrix on the test set:



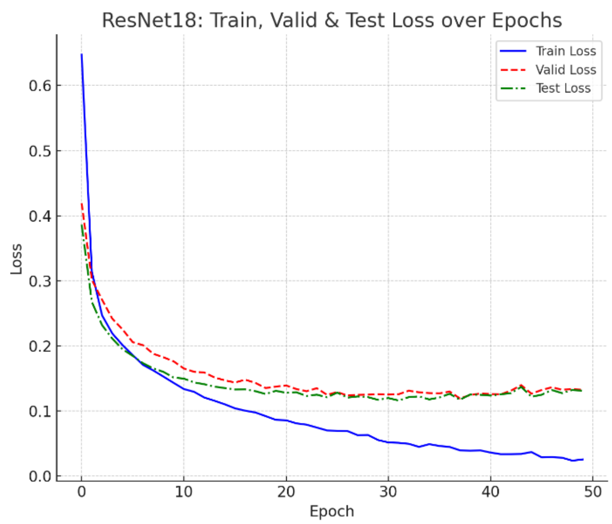
ResNet18 [11]:

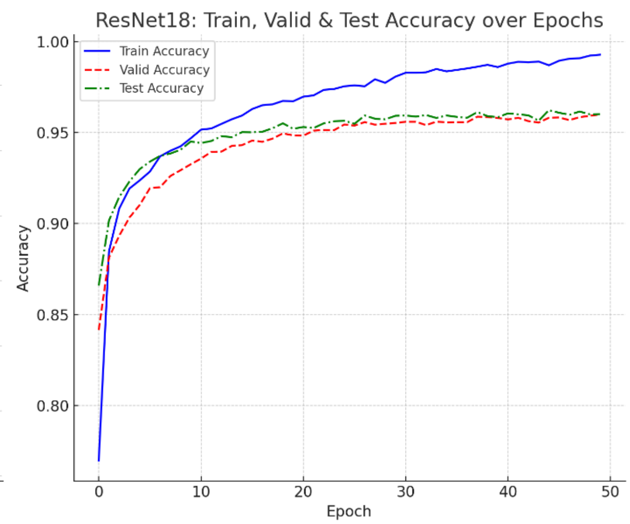
The model had the following training details:

- Learning Rate: 1e-05

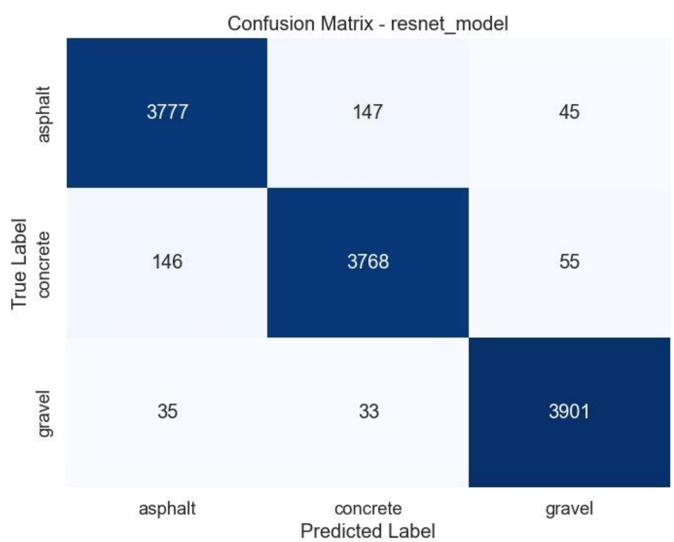
- Epochs: 50
- Batch Size: 128
- Total parameters: 11440707
- Trainable parameters: 11440707

The loss and accuracy graphs looked as follows:





The confusion matrix on the test set:



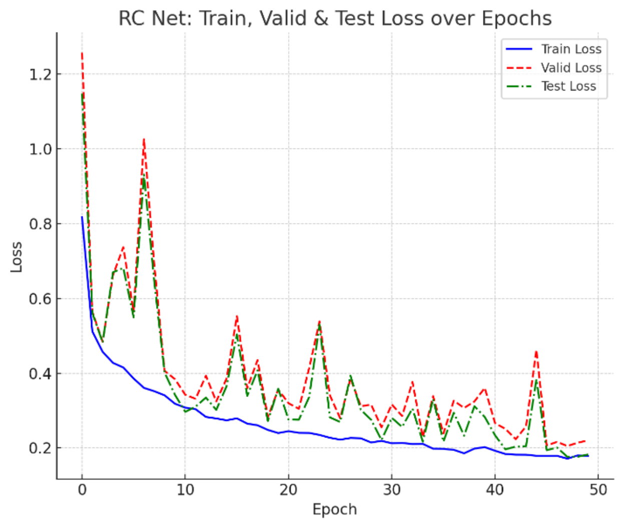
RC Net:

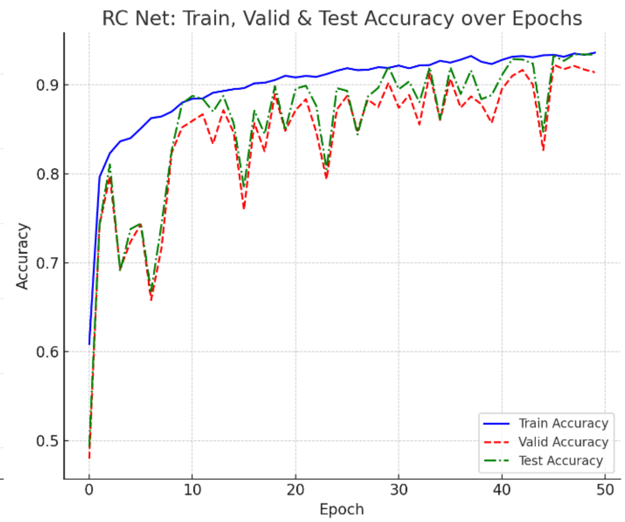
The model had the following training details:

- Learning Rate: 0.0001

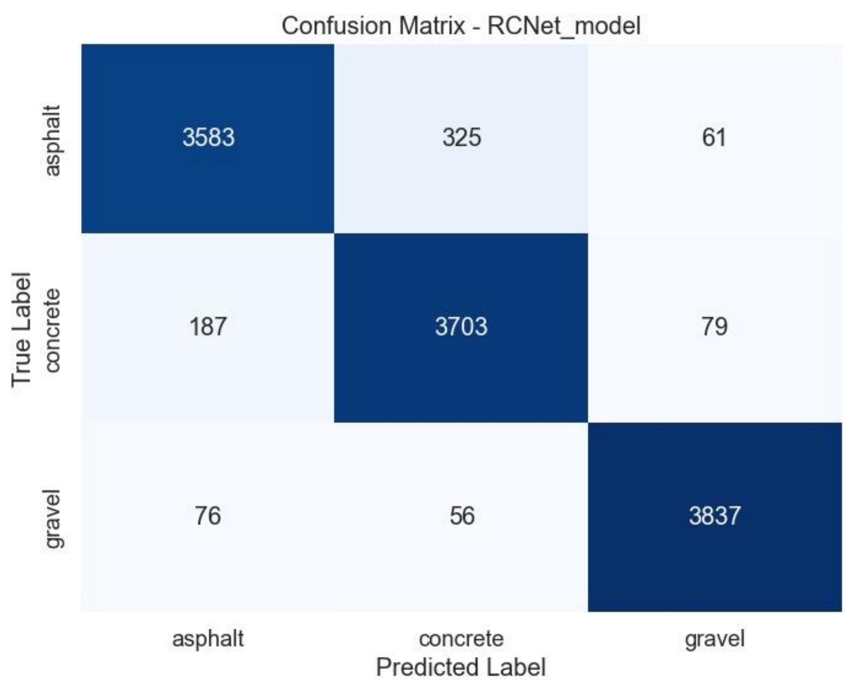
- Epochs: 50
- Batch Size: 128
- Total parameters: 4829731
  Trainable parameters: 4829731

The loss and accuracy graphs looked as follows:





The confusion matrix on the test set:

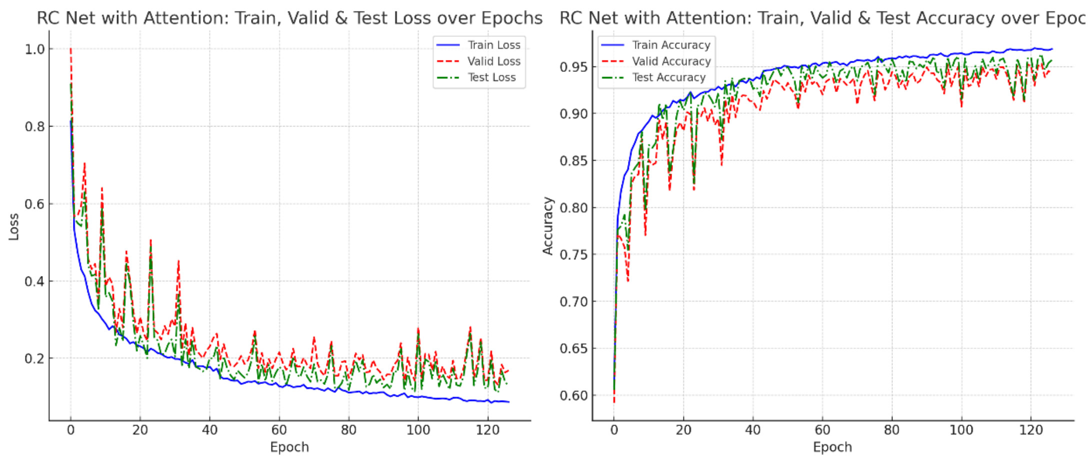


RC Net with Attention:

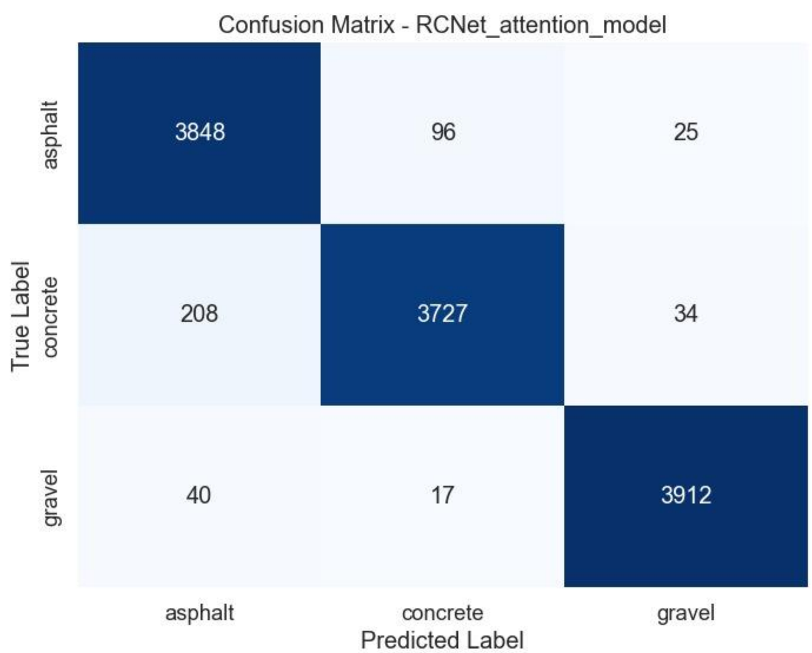
The model had the following training details:

- Learning Rate: 0.0001
- Epochs: 125
- Batch Size: 128
- Total parameters: 4849507
- Trainable parameters: 4849507

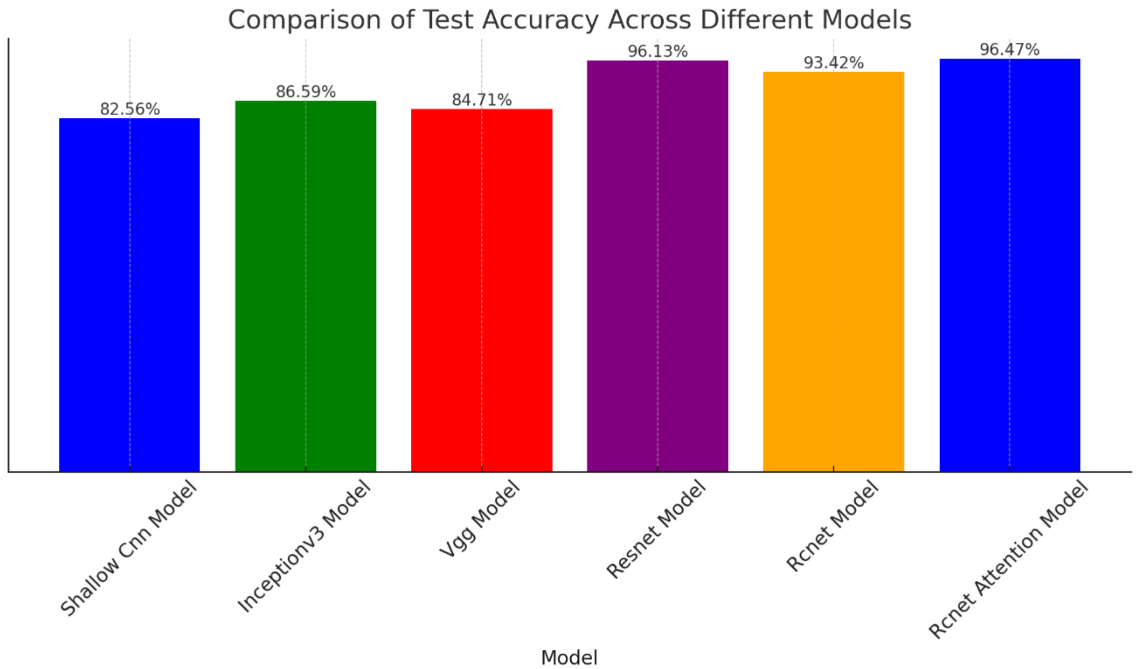
The loss and accuracy graphs looked as follows:



The confusion matrix on the test set:



Comparison of test accuracies:



Comparison of model parameter size

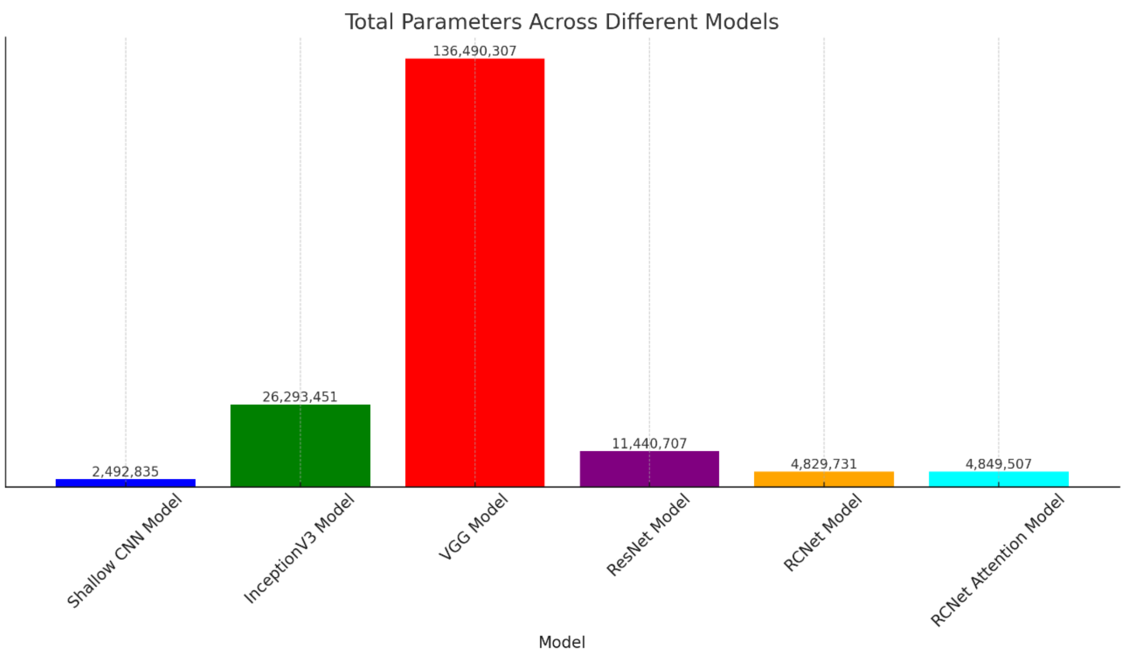


Table of results:

Model	Test Loss	Test Accuracy
shallow_cnn_model	0.432218599	0.825648778
inceptionV3_model	0.350453882	0.865877215
vgg_model	0.379357033	0.847148736
resnet_model	0.117266169	0.961283279
RCNet_model	0.176128799	0.934156379
RCNet_attention_model	0.106373388	0.964726631

Analysis of Results:

From the graphs above we can see that the two best performing models are the ResNet model and the RCNet model with attention which both score 96% accuracy on the test set. From the training graphs we can see that all the models successfully trained to convergence apart from the VGG model and Inception model. With a total parameter count of over 100 million parameters VGG took too long to train and had to be stopped early. This explains its poor performance as it didn't fully train for enough epochs. The Inception model with over 20 million parameters itself also had to be stopped prematurely after 20 epochs since it was too computationally expensive to keep running and was hence stopped early. We believe that both VGG and Inception would have scored an accuracy of over 90% had it been trained fully.

When looking at total parameter count for each model the shallow CNN is by part the smallest. However, this model was used as a baseline for performance and reported the worse scores among all the models, for this reason the shallow CNN will not be included in the conversation of comparing models.

The two RC Net models have a remarkably small size of roughly 5 million parameters. In particular, the best performing model, the RC Net with attention only has 4.8 million parameters which is more than half as small as its close competitor ResNet18 which has a size of 11.4 million parameters.

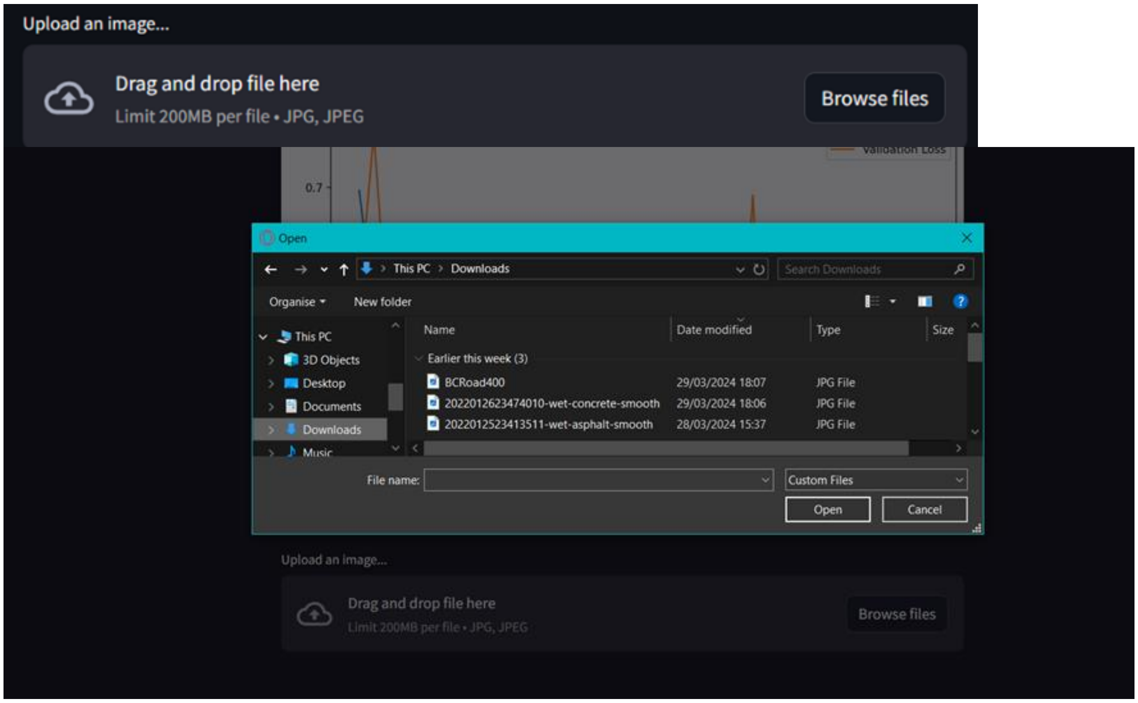
It is clear to see that the optimal model for this case study is the RC Net model with attention due to its best performance on the test set and its small architecture.

Web Application

We chose to use Streamlit as our framework for a front-end app. Streamlit is an open-source python library designed specifically for creating web apps for data science and machine learning projects. It has recently become a very popular framework among data scientists, with many employers looking for candidates that have experience utilising it. Since it is a python framework

Streamlit lets us easily re-use code previously written for our experiments, such as image pre-processing or loading in our models and provides an easy way to integrate pyplot graphs into the page.

Our app was made to be simple and easy to navigate. A dropdown menu gives the choice of one of 6 models: our design (RCNet with attention layers), RCNet, InceptionV3, VGG16, ResNet, and a shallow CNN. Upon choosing a model, a training curve giving the training and validation loss over each epoch is displayed. The user can submit an image to the chosen model to get the prediction for it.


Upload an image...
Drag and drop file here
Limit 200MB per file • JPG, JPEG
Browse files
Open
This PC › Downloads
Search Downloads
Organise
New folder
This PC
3D Objects
Desktop
Documents
Downloads
Music
Name
Date modified
Type
Size
Earlier this week (3)
BCRoad400
29/03/2024 18:07
JPG File
2022012623474010-wet-concrete-smooth
29/03/2024 18:06
JPG File
2022012523413511-wet-asphalt-smooth
28/03/2024 15:37
JPG File
File name:
Custom Files
Open
Cancel
Upload an image...
Drag and drop file here
Limit 200MB per file • JPG, JPEG
Browse files


Drag and drop file here
Limit 200MB per file • JPG, JPEG
Browse files
2022012623474010-wet-concrete-smooth.jpg 11.1KB
Uploaded Image.
The model predicts: wet_concrete_smooth


Choose model:
RCNet_attention
Training and Validation Loss for RCNet_attention
Training Loss
Validation Loss
Loss
Epoch
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0
5
10
15
20
25
30
35
40

Conclusion

In conclusion, our case study has shown impressive findings by using a unique implementation of the RC Net model augmented with attention blocks and dropout on road images. This model outperformed all others in our experiments, achieving a great accuracy of 96.5% on the test set while maintaining a relatively low parameter count of only 4.8 million. This demonstrates the efficiency of integrating attention blocks within CNNs, providing a substantial improvement in model performance without a huge increase in complexity. This is a completely new model architecture within the domain of road surface classification and there will be no intellectual property issues for the customer. Furthermore, the model can work in real time having been successfully implemented on the web server.

Future Directions:

Looking forward, there are several avenues for expanding this research. First, investigating the integration of additional situational features could further enhance the model's sensitivity to varying environmental conditions, which would improve accuracy in diverse settings. Also, developing a more robust system for real-time feedback and dynamic model updating could ensure the model remains effective as new data and road conditions emerge. These directions can improve the current model's capabilities and also to broaden the scope of its applicability beyond just road surface prediction.

References

[1] J. A. Kovilpillai J, S. Dhar, U. K. Singh, S. Prakash, R. K. Saidala, and S. Jayanthy, “Advancements in road surface classification: a comprehensive review of computer vision and IoT approaches,” in *International Conference on Computer Vision and Internet of Things 2023*, Institution of Engineering and Technology (IET), Jan. 2023, pp. 31–36. doi: 10.1049/icp.2023.2849.

[2] Y. Qian, E. J. Almazan, and J. H. Elder, “Evaluating features and classifiers for road weather condition analysis,” in *IEEE International Conference on Image Processing* , 2016, pp. 4403–4407. doi: 10.1109/ICIP.2016.7533192.

[3] Marcus Nolte, Nikita Kister, and Markus Maurer, “Assessment of Deep Convolutional Neural Networks for Road Surface Classification,” in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 381–386.

[4] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, “Year, 1000km: The Oxford RobotCar Dataset,” *Int J Rob Res*, vol. 36, no. 1, pp. 3–15, 2017, doi: http://dx.doi.org/10.1177/0278364916679498.

[5] T. Zhao, J. He, J. Lv, D. Min, and Y. Wei, “A Comprehensive Implementation of Road Surface Classification for Vehicle Driving Assistance: Dataset, Models, and Deployment,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 8, pp. 8361–8370, Aug. 2023, doi: 10.1109/TITS.2023.3264588.

[6] D. K. Dewangan and S. P. Sahu, “RCNet: road classification convolutional neural networks for intelligent vehicle system,” *Intell Serv Robot*, vol. 14, no. 2, pp. 199–214, Apr. 2021, doi: 10.1007/s11370-020-00343-6.

[7] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, “CBAM: Convolutional Block Attention Module,” in *European conference on computer vision*, Jul. 2018, pp. 3–19. [Online]. Available: http://arxiv.org/abs/1807.06521

[8] Y. H. Guo, J. R. Zhu, C. C. Yang, and B. Yang, “An Attention-ReXNet Network for Long Tail Road Scene Classification,” in *Proceedings of 13th IEEE International Conference on CYBER Technology in Automation, Control, and Intelligent Systems, CYBER 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 1104–1109. doi: 10.1109/CYBER59472.2023.10256482.

[9] C. Szegedy, V. Vanhoucke, S. Ioffe, and J. Shlens, “Rethinking the Inception Architecture for Computer Vision,” in *IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[10] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *arXiv preprint*, Sep. 2014, [Online]. Available: http://arxiv.org/abs/1409.1556

[11] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *IEEE conference on computer vision and pattern recognition*, Dec. 2016, pp. 770–778. [Online]. Available: http://arxiv.org/abs/1512.03385