Course: NGEN27 Student: Loa Andersson

GeoAI Project – Soil Classification with CNN

Examiner: Rachid Oucheikh Date: 2024-12-28

Introduction

As the population increases, so does the need for stable food production. Food production is dependent on arable land, which is being utilised for the expansion of urban environments to meet the growing population (Alexander et al., 2015). A paradox arises where increasing population also leads to less agricultural land. There is no simple solution to this problem, but one way to increase food production is to grow food in soils where specific crops thrive. Soil typing is described as a must to identify which crops to grow where and meet the increasing demand for food (Haider & Rehman, 2019). However, classifying soil types requires demanding field surveys that are expensive and time-consuming (Pandiri et al., 2024). In recent years, physical models have been developed that classify soil types based on environmental factors, but creating physical models that identify many similar soils has so far not been possible with great reliability (Gatiboni, 2018). Thus, there is a great need for cheap, time-efficient and reliable classification of soil types. One possible approach to the problem is to use computer vision (Pandiri et al., 2024).

Computer vision is a concept that has been researched for several decades and includes applications today such as object identification, organism identification and facial recognition in images (Bhatt et al., 2021). Traditional image classification is generally done by manually entering specific image features to search for in each image, which results in poorer generalisation for unseen data (Chen et al., 2021). With the development of AI applications in recent years, Deep Learning (DL) and Machine Learning (ML) algorithms have increased in use for computer vision. By allowing the computer to process images like human vision and process the image information like human neural networks, the possibility of generalising image classification and making it useful on unseen data is created (Voulodimos et al., 2018). One neural network DL algorithm which have been used since 1980 for image classification is the Convolutional Neural Network (CNN). The model was used sparingly due to lack of computer hardware until just a decade ago (Rawat & Wang, 2017). Since 2017, several different model architectures for CNN have been presented, showing time and again a superiority for image classification (Chen et al., 2021).

CNN has shown good results for the classification of soil types where the soil types are visually similar for the human eye. Pandiri et al. (2024) isolated the soil mounds on white or black background, photographed with RGB camera and used them as training and validation data. The structure of the soil, which depends on the grain size, was the type-specific patterns identified by CNN and the accuracy of the model was 97.2%. Since the model was trained on homogeneous backgrounds, the retrieval of the soil types to the lab environment is required for the practical use of the model. A model that classifies characteristic soil types in images without homogeneous backgrounds, such as in landscape or field images, would further increase the possibilities in soil type classification. The aim of this report is to investigate whether CNN can be used to classify four characteristic soil types in images without homogeneous background such as field images. The objectives of the report are to:

- Create a CNN architecture using Google's open-source DL framework TensorFlow.
- Tune the CNN model to find the best hyperparameters for the data and pre-processing for the images.
- Train the CNN model with pre-processed and normalised images of Alluvial soil, Black soil, Clay soil and Red soil that include images without homogeneous background.

- Evaluate the CNN model with evaluation metrics such as accuracy, precision, recall and F1-score.
- Train the model on Alluvial and Clay only and evaluate the results.

Method

This section discusses the methodology for data pre-processing, building of CNN architecture and fine-tuning hyperparameters. The hardware specifications and software used for the implementation are presented in Table 1. An overall workflow is shown in Figure 1.

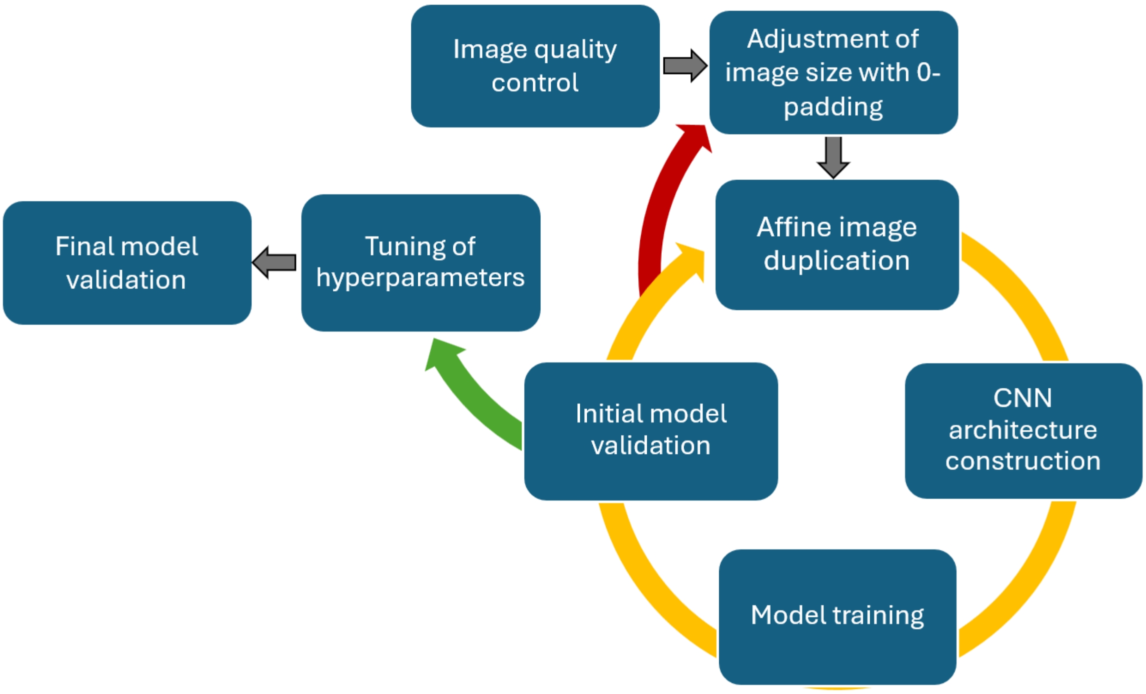

Figure 1: The figure shows an overall workflow for developing and training a CNN model to predict soil types in RGB images. Yellow arrow shows that the process was repeated with new architecture for each iteration. Red arrow shows that the image size was alternated occasionally to further evaluate CNN architectures. Green arrow shows that when an initial model met the goal of the development, the model was taken further for tuning of hyperparameters and validation.

Table 1: The table describes the hardware specifications and software used for the CNN training process. Hyperparameter tuning was particularly time-consuming; despite utilizing all CPU cores, the tuning process took nearly 6 hours. This is mentioned to highlight that repeating the work process described in this report with less powerful hardware may not be feasible due to time constraints.

Hard- and software	Specs
CPU	Intel Core i9 13900HX, 24 Cores, 32 Threads, 2,2GHz – 5,4GHz
GPU	NVIDIA GeForce RTX 4070, 12GB VRAM
RAM	32GB DDR5, 5600MHz
Programming language	Python 3.11.4
Software	TensorFlow 2.16.1, Keras 3.6.0

Pre-processing

RGB images from four different classes were downloaded from Lund University´s canvas for the course NGEN27. The folder contained two subfolders, one for test data and one for training data for the classes Alluvial soil, Black soil, Clay soil and Red soil. The folders with images were manually checked for errors and mistakes. Two images were removed from Red Soil that showed no soil but only a colour gradient ('Red_39' and 'Copy of images105'). The images varied in size and was not homogenous in X, Y direction. Therefore, the size of the images was normalised by scaling up and down (depending on the original size) and adding a padding of 0-values around each image to reach the correct size. Normalisation of the size varied between 256x256 and 512x512 (see section for the CNN architecture). There was a difference in the number of images in each class (Figure 2). To smooth out the differences in class size and at the same time create a more varied training data, all classes that had less than 300 images were duplicated with a randomised affine transformation for each duplicate. The band values of the images were normalised by min-max scaling. Pre-processing was finalised by splitting the data into 70% for training and 30% for validation and converting the labels to one-hot encoding.

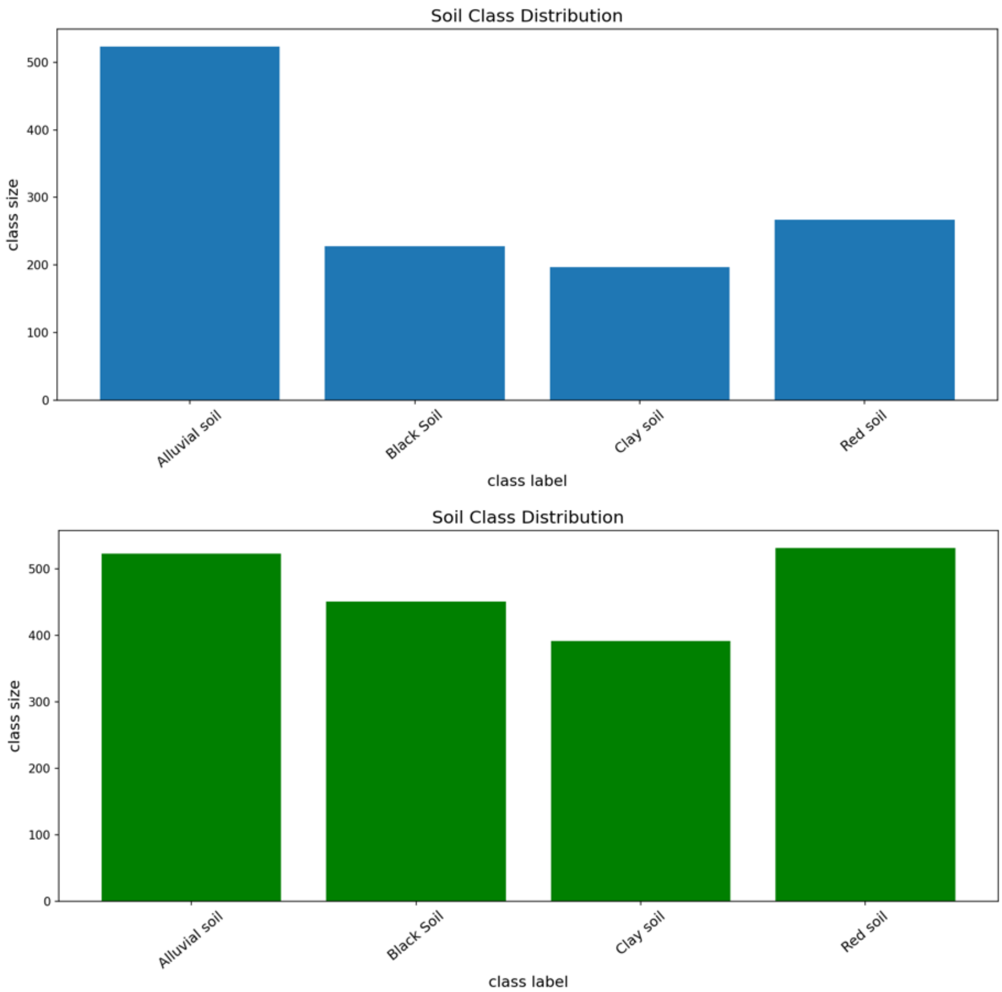

Figure 2: The figure shows the number of images in each soil class used for training the CNN model. The original dataset was decided to be too unbalanced with too large differences between Alluvial soil and the other three classes (blue). Therefore, the number of images in these classes was duplicated using an affine transformation (green)

CNN architecture

Next, an extensive task of identifying an optimal model architecture was initiated. Validation and training accuracy/loss were plotted for each tested architecture. The development goal was to identify an architecture that exhibited smooth and closely aligned curves between training and validation datasets. As there was still a difference in the class sizes after duplicating the images, weighting for the labelling was used in compiling the model so the effect of this difference was reduced during model training. To prevent overfitting, early stopping was also employed. This technique halts the training process if the validation accuracy does not improve for 5 consecutive epochs. The yellow arrow in Figure 1 visualises the process of finding a model architecture that meets the development goals. When a CNN architecture did not demonstrate adequate alignment between the training and validation data, the structure was modified, and the model was re-trained. Occasionally, the image size (as indicated by the red arrow in Figure 1) was also adjusted to evaluate different CNN architectures and explore new combinations of input data and neural layers. All CNN architectures built from scratch, with varying image sizes and model depths, encountered significant issues with overfitting. Batch normalisation, L2 regularization and dropout rates were added and adjusted without major improvements. A transfer learning approach was finally tested with the pre-trained model VGG16 trained on the ImageNet image database. VGG16 was added as a base layer to the existing CNN architecture (Figure 3). The base layer was frozen for modification during training so that the pre-trained weights would persist through all epochs. Since the data demonstrated a tendency towards overfitting, the architecture was designed with as few layers as possible after the VGG16 base layer. This approach aimed to minimize the depth of the model, thereby reducing the risk of the model learning excessive details from each image. This approach yielded a significant improvement in model validation after training with the specific dataset used in this report (Figure 4).

Model fine tuning

Once the goal of the CNN architecture development was achieved, a copy of the model was converted into a hypermodel where hyperparameters were set in ranges for fine-tuning with Keras Tuner. The hyperparameters set in ranges included dropout rate, L2 regularisation, learning rate and the number of units in the first dense layer (Figure 3). Optimisers were not tested with Keras Tuner as they had been evaluated during the initial construction of the CNN architecture. RMSprop exhibited the best results and was therefore chosen as the sole optimiser during Keras Tuning. A randomized search was conducted with 10 different trials, where each trial was run once with 20 epochs and early stopping, with the objective of achieving the highest possible validation accuracy. After tuning the hyperparameters, the best model hyperparameters were retrieved. The first four layers of the VGG16 model were then activated for training to further enhance accuracy while avoiding overfitting (hence only training of the first four layers) and the model was then compiled and trained a second time at a very low learning rate for 10 epochs with early stopping. For clarity, fine-tuning was thus performed in two rounds. Initially, Keras Tuner was used to identify the optimal hyperparameters, which were then employed in a subsequent fine-tuning process. During this second round, the first four layers of the base VGG16 model were unlocked and modified for learning, having been previously frozen during the initial fine-tuning round. The fine-tuned model was subsequently used to predict 25 randomly selected images from the test data folder in the dataset.

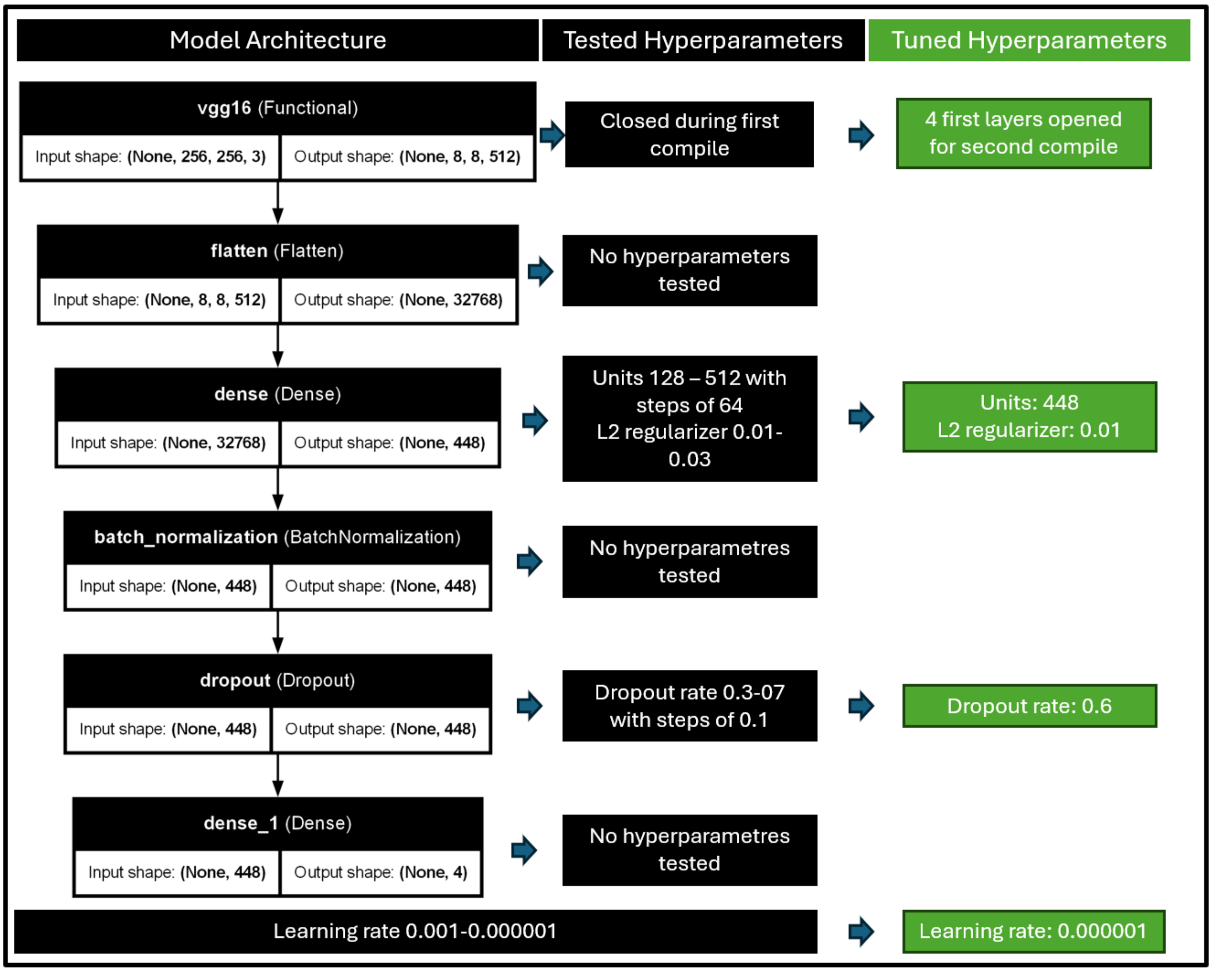

Figure 3: When the development of the CNN architecture reached the development goals, the hyperparameters were fine-tuned. This was done in two rounds where the Keras Tuner was used to search for the best hyperparameters. The best model was then trained with the VGG16 base model with the first four layers trainable. The figure shows the hyperparameters that were tested and which hyperparameters gave the best validation accuracy during the training phase.

Model training with only clay and alluvial soil

After fine tuning the hyperparameters, a separate model was also trained using only the Alluvial and Clay soil classes. This was done using the same architecture and hyperparameters as for the previous model.

Results

The goal of the development of the CNN architecture was to achieve training and validation data with as similar changes as possible in validation accuracy. This was considered achieved after the initial CNN model showed the curves in Figure 4. The validation accuracy shows a rapid increase, indicating that the model finds patterns quickly in the validation data. Since the curve flattens out without large fluctuations, Figure 4 shows that the model is stable and insensitive to variations in the validation data, and this is also confirmed by a smooth decreasing validation loss curve which means successively better predictions. The overfit observed throughout the model development can also be seen in the initial model where the validation accuracy falls below the training accuracy when the model is trained with more epochs (>10). The initial model shows validation metrics above 0.94 for Accuracy, Precision, Recall and F1 (Table 2). After the fine-tuning of hyperparameters and the recompilation with trainable layers in the VGG16 model, these increased to 0.972 for all validation metrics. When the refined model was trained with only Alluvial soil and Clay soil, all validation metrics increased to 0.986 (Table 2). However, the training and validation loss/accuracy curves (Figure 5) reveal a more unstable model. The model quickly identifies patterns in the validation data, as evidenced by the rapidly increasing accuracy curve. However, there is some volatility in the accuracy curve for the validation data, indicating that the model is sensitive to variations within the validation data. Consequently, this model can be considered more unstable compared to the model trained with all four soil classes.

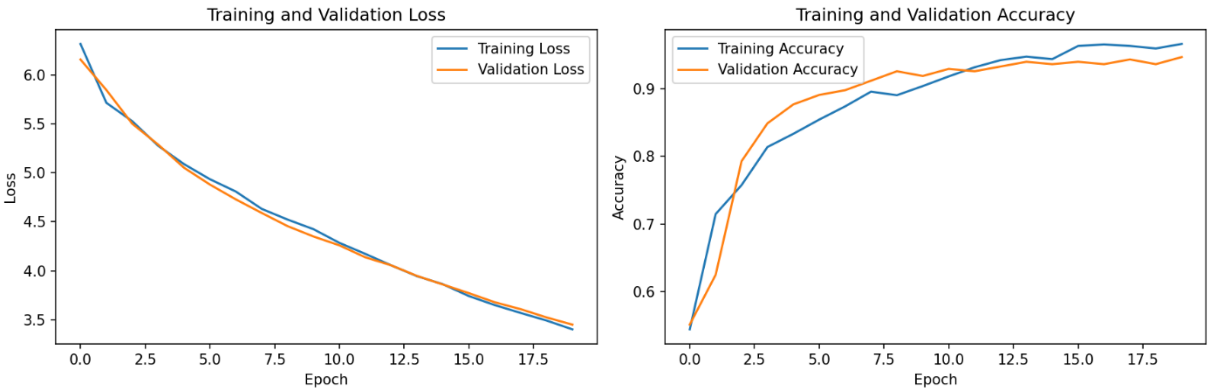

Figure 4: The figure illustrates the training and validation loss/accuracy curves for the fine-tuned CNN model, which was trained on the four soil classes: Alluvial soil, Black soil, Clay soil, and Red soil. For clarity, the curves shown represent the results after the first round of tuning with Keras Tuner. The model was further refined in a second round, where the first four layers of VGG16 were made trainable. Since the validation and accuracy curves after second round did not show significant information, reflecting only a small change from 0.944 to 0.972, these curves have been excluded from the results section.

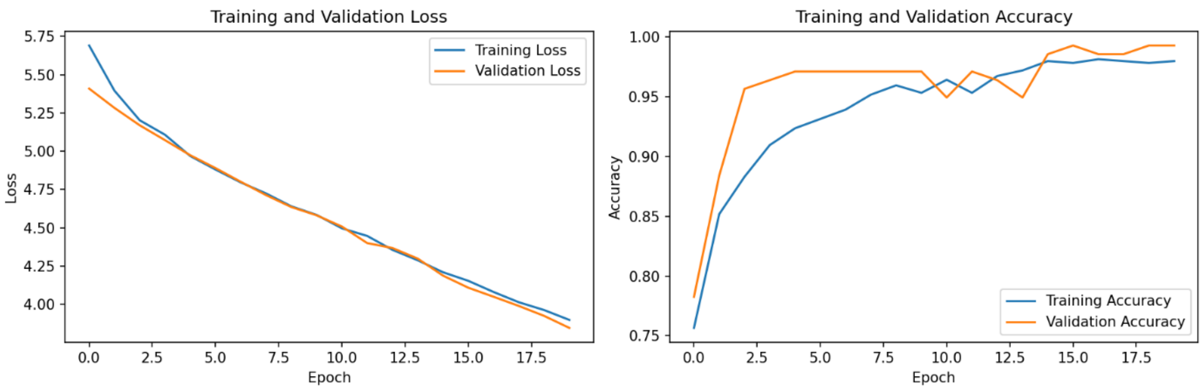

Figure 5: The figure shows the training and validation loss/accuracy for the fine-tuned CNN model trained with the two soil classes Alluvial soil and Clay soil.

Table 2: The table shows validation metrics for the CNN model trained with the four soil classes Alluvial soil, Black soil, Clay soil and Red soil before hyperparameter tuning, after hyperparameter tuning and with only the classes Clay soil and Alluvial soil.

Metric	Before hyperparameter tuning	After hyperparameter tuning	Only Clay and Alluvial
Accuracy	0.944	0.972	0.986
Precision	0.946	0.972	0.986
Recall	0.944	0.972	0.986
F1	0.944	0.972	0.986

The confusion matrix of the fine-tuned model trained with all four soil classes shows a high proportion of true positives for each class (Figure 6), which is consistent with the high validation metrics (Table 2). Apart from occasional misclassifications for some classes, Alluvial soil is misclassified as Red soil on two occasions, representing false negatives for Alluvial soil.

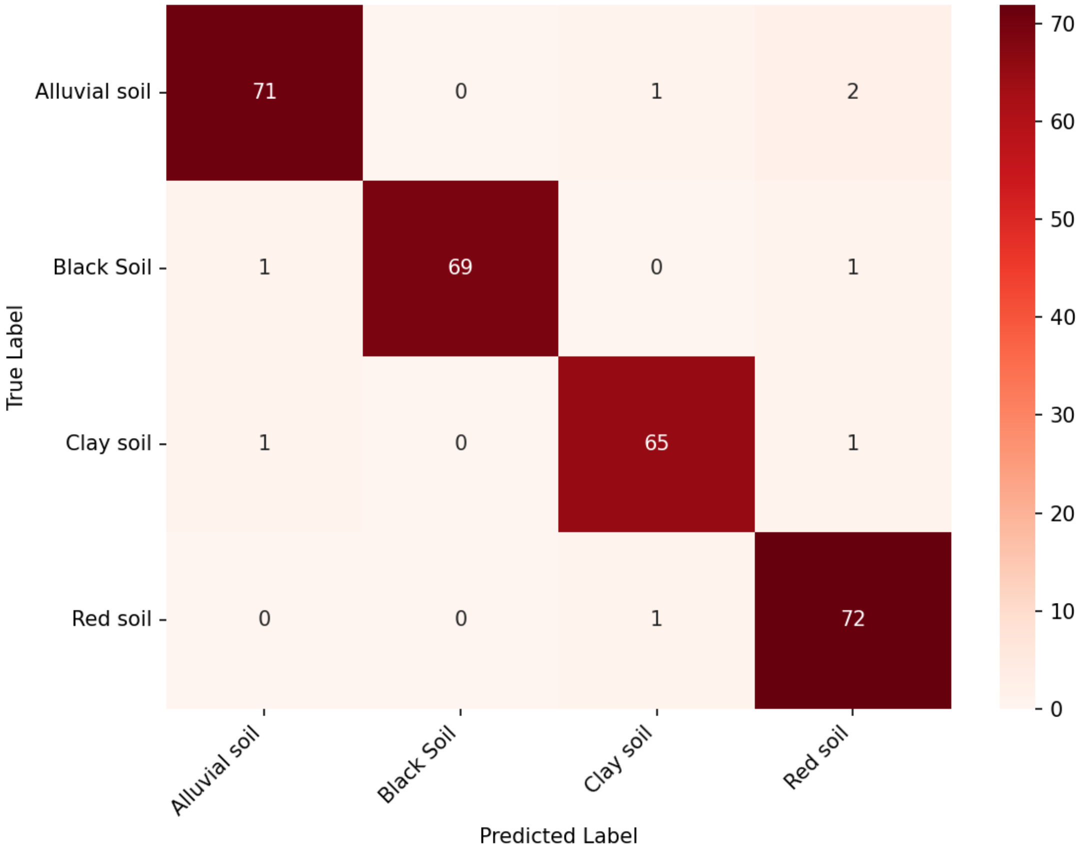

Figure 6: The figure shows a confusion matrix of the fine-tuned CNN model trained with the soil classes Alluvial soil, Black soil, Clay soil and Red soil.

The fine-tuned model was used to predict the test dataset (retrieved from Lund University´s canvas) in which 25 images were randomly selected. For these 25 predictions, Clay soil was classified as Alluvial soil twice, and Red soil was classified as Black soil once (Figure 7).

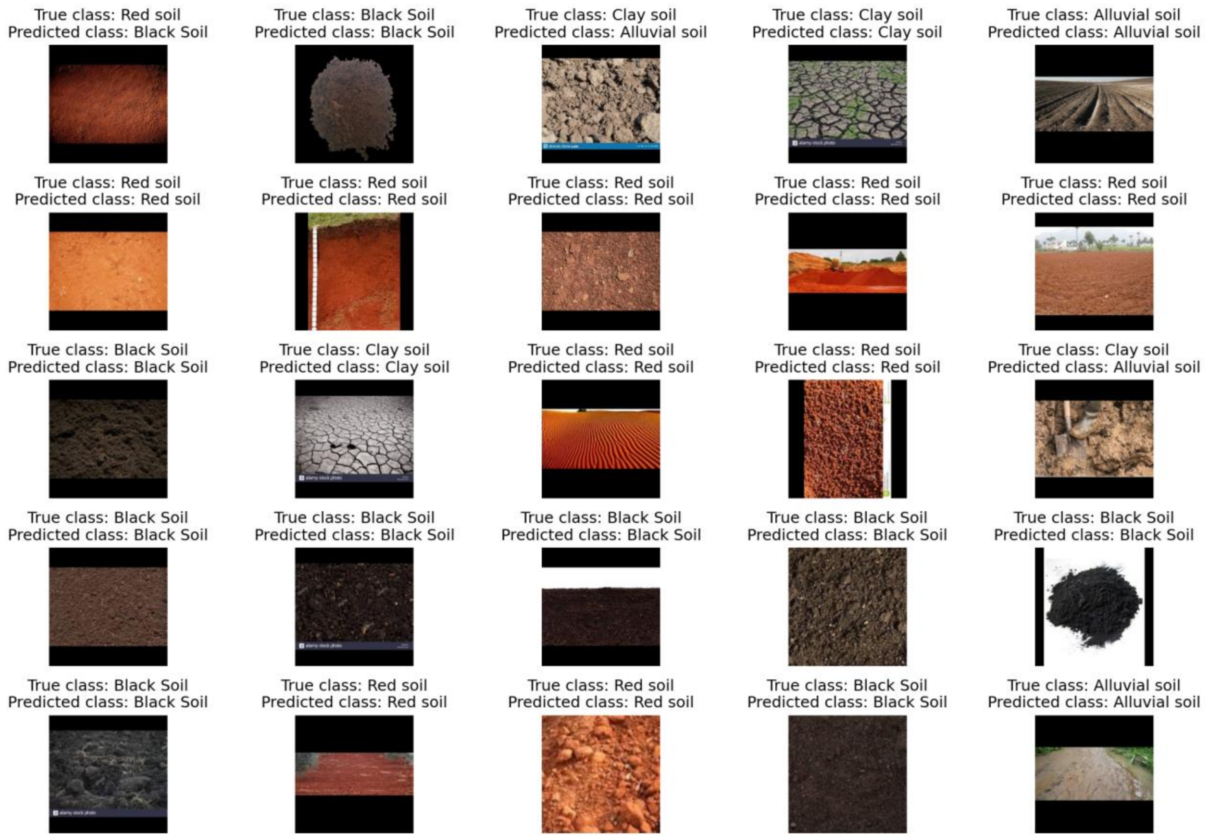

Figure 7: The figure shows soil class labels predicted by the fine-tuned CNN model.

Discussion

Training a CNN model with non-homogeneous images of soil types proved to be prone to overfitting despite using shallow model architectures and measures such as batch normalization, L2 regularization, and increased dropout rates. This is likely due to the heterogeneous background and non-normalized soil preprocessing, which creates high variance between the images, causing the model to adapt to specific details rather than general similarities. Pandiri et al. (2024) addressed this issue by isolating soil types in a lab environment and drying them so that the soil piles had the same water content. This preprocessing step reduced the variance between the images, resulting in more general similarities. This was essential as soil types with similar compositions, colors, and grain sizes had to be distinguished using CNN. In this report, only distinct soil types were used, but since the water content, photo angles and sunlight varied between the images, there remained intra-variance that contributed to additional variance that the model could overfit on.

Based on advice from Dr. Rachid Oucheikh, researcher at Lund University, a transfer learning approach was tested with VGG16 to reduce overfit and increase model performance. Transfer learning with frozen pre-trained models trained on large datasets such as ImageNet has been shown to work well for generalising models and reducing overfit (Iman et al., 2023, Sing et al., 2018). When VGG16 was added as a frozen base model together with an otherwise basic model with batch normalisation, L2 regularisation and dropout, the model's validation metrics increased and much less overfit was observed. To further improve the model, Keras Tuner was used to find the hyperparameters that gave the best validation accuracy. When the model was compiled with these hyperparameters, the first four layers of the base model were also opened for training while the remaining layers were kept closed for training during a second compile. Iman et al (2023) describe how freezing and unfreezing of layers in a pre-trained base model can be used to increase performance on data that the pre-trained base model has not seen before. The application of hyperparameter tuning and defrosting of layers in the VGG16 base model resulted in an increase of the accuracy from 94.4% to 97.2%.

The confusion matrix validates the accuracy with a very high proportion of true positive predictions. However, two false negatives are seen for Alluvial soil and occasional misclassification for most other classes (Figure 6). In the prediction with the test data (Figure 7), Red soil is misclassified as black soil. This image shows some vignetting and shading at the edges of the image which makes the soil appear darker in the image and more like black soil. Alluvial soil is not classified as Red soil during the prediction of the 25 images as in the confusion matrix, but Clay soil is classified as Alluvial soil on two occasions. When looking closely at these images, the Clay soil is not characteristically segmented, and the water content varies greatly giving it the Alluvial soil look. Similarly, this can also be interpreted by Clay and Alluvial soils showing an unstable fit between training and validation data (Figure 5). This is because the model is sensitive to the variance in soil type condition. Comparatively, images taken at unfavourable angles on Red soil and on Red soil with high water content may appear darker and more similar to Black soil and vice versa. As weighting was used to reduce the effect of different class sizes, these differences should not have had a major impact on the misclassifications. However, it is possible that a more normally distributed image variance within each class would have reduced these misclassifications by levelling out the intra-variance between the states of each soil type.

Training a CNN where the background, angles, and sunlight of the images are not normalised is part of this report's objective. Pandiri et al. (2024) achieved an accuracy of 97.2% with normalised soil conditions and background in the images, compared to 97.2% achieved by this report with other conditions for the images. The result should be seen in the light of characteristic soil types as opposed to the soil types classified by Pandiri et al. (2024). Achieving the same accuracy with a heterogeneous background and non-normalised soil conditions as with a homogeneous background and normalised soil conditions can be considered a successful model development. However, the application to field images should be further evaluated with more soil types and testing of more transfer learning models to hopefully create a model that can identify more soil types without homogeneous background to further justify its use. In addition to this, a standard should be developed for the field images that should follow a form of normalisation in terms of shading, time of day and photo angle in relation to the sun. Based on the hypotheses of this report, such normalisation could further improve the performance of the CNN model.

Conclusion

Using Google's open-source TensorFlow to create and train a CNN model for soil type classification in field images proved to provide a stable model with overall high accuracy (97.2%). Some problems in distinguishing Alluvial soil from Clay soil and Black soil from Red soil are found when the model is used on unseen data. This is probably due to intra-variance between the images in each class resulting from differences in water content, photo angle and differences in sunlight. For further development of the model, work should be done on two fronts. Firstly, to develop a more accurate model that can classify more soil types in the field, but also by developing a field standard for the field images to normalise the parameters that are possible in field environments.

Sources

Alexander, P., Rounsevell, M. D., Dislich, C., Dodson, J. R., Engström, K., & Moran, D. (2015). Drivers for global agricultural land use change: The nexus of diet, population, yield and bioenergy. *Global Environmental Change*, *35*, 138-147.

Bhatt, D., Patel, C., Talsania, H., Patel, J., Vaghela, R., Pandya, S., ... & Ghayvat, H. (2021). CNN variants for computer vision: History, architecture, application, challenges and future scope. *Electronics*, *10*(20), 2470.

Chen, L., Li, S., Bai, Q., Yang, J., Jiang, S., & Miao, Y. (2021). Review of image classification algorithms based on convolutional neural networks. *Remote Sensing*, *13*(22), 4712.

Gatiboni, L. (2018). Soils and plant nutrients. *North Carolina Extension Gardener Handbook. NC State Extension, Raleigh, NC*.

Haider, W., & Rehman, A. (2019). Knowledge based Soil Classification Towards Relevant Crop Production. *Int. J. Adv. Comput. Sci. Appl.*, *10*(12), 488-501.

Iman, M., Arabnia, H. R., & Rasheed, K. (2023). A review of deep transfer learning and recent advancements. *Technologies*, *11*(2), 40.

Pandiri, D. K., Murugan, R., & Goel, T. (2024). Smart soil image classification system using lightweight convolutional neural network. Expert Systems with Applications, 238, 122185.

Rawat, W., & Wang, Z. (2017). Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, *29*(9), 2352-2449.

Sing, M., Saxena, V., & Jain, A. (2018). The role of transfer learning in enhancing model generalization in deep learning. *International Journal of Applied Research 2018, 4(10),* 59-62.

Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, *2018*(1), 7068349.