|  |  |  |
| --- | --- | --- |
|  | **ResNet50** | **Xception** |
| **MINC with Baseline (Unfreeze weights)** | 0.7151 | 0.7548 |
| **MINC with Data Augmentation and Hyperparameter tuning of models** | 0.7610 | 0.7668 |

Summarised Report of Models

**Objectives:**

My project aims to tackle the issue of material recognition by implementing various state-of-the-art CNNs through transfer learning to classify the various images of the surfaces provided to its respective material category. I have referenced a deep learning architecture from Kaggle that utilises ResNet50 as its base model, and explored other machine learning techniques to finetune and optimise the model’s performance. Following, I will also use another well-known CNN architecture called Xception and implement similar techniques so that I can compare the performances of both models in classifying the different material categories.

**Conclusion:**

This huge difference between ResNet50 and Xception could be due to the vast difference in design architecture. While ResNet50 focuses more on adding layers to improve the model’s accuracy whilst minimising the percentage of errors, Xception focuses more on depthwise separable convolutions to improve computational efficiency. As such, despite having trained on the same ImageNet dataset as ResNet50, Xception might possess different weights that cater to the learning of the FMD dataset when training.

Performing data augmentation and hyperparameter tuning is proven to greatly enhance the model’s accuracy. This result is within my expectations since data augmentation is known to increase the generalizability of an overfitted data model by generating additional training data such as through Image resizing or Image rotation, thereby exposing the model to different versions of data. In terms of hyperparameters, I introduced dropouts at every single hidden layer. Dropout works by temporarily dropping different sets of neurons during each training iteration. This prevents overfitting as the neurons are not co-adapting with other neurons. Utilising the keras tuner and bayesian optimization, I managed to find the best learning rate, dropout probability and hidden nodes for each model.

In conclusion, depending on the nature of the architecture that is to be built and the constraints set on the neural network, Xception may perform better than ResNet50. This is seen when the weights are frozen and no data augmentation is done. However, if there are no constraints then after fine-tuning the hyperparameters, I can see that the ResNet50 yields similar results to Xception when Data Augmentation and Hyperparameters fine-tuning are done.