Project 2 Report

**Summary**

This project showcased the ways to develop an accurate Deep Convolutional Neural Network (DCNN) to classify images based on their defects into different categories like: Crack, Missing Head, and Paint Off.

**Learning Methods**

The DCNN consisted of four Conv2D Layers with32, 64, 128, 256 filters with ReLU activation and a maxpooling size of (2,2) and two dropout layers. Initially, L2 regularization was not used but due to its ability to effectively mitigating overfitting it was added. The dropout rate is at 60%, initially tested at 50%, then to 40% noticing that a higher dropout rate reduced the overfitting so kept at 60%.

Learning rate was set to 0.0005 for stable convergence. Since many first iterations showed a low accuracy epoch value was raised to 50, but included an EarlyStopping and ModelCheckpoint callback. These monitor the validation loss with patience at 3 provided while also saving the best-performing model (Many times where laptop would shut off or code would crash so this method allows autosave).

**Accuracy and Loss Evaluation**

Early iterations yielded an accuracy of ~45%, this was a terrible model and needed to improve the set. The layers were initially changed from three to four layers, and dropout from one layer to two, epochs to 50, and reduced the regularizer. The final model peaked at 75% accuracy and Validation Accuracy at 71.23%. The loss was reduced progressively for both training and validation showing a better generalization by epoch 24 as shown in Figure 1.

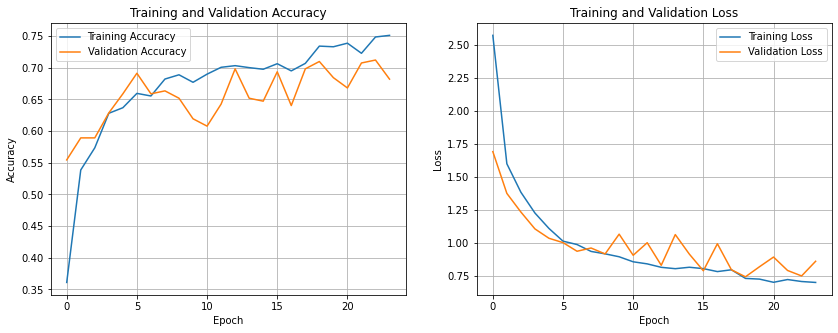


Figure 1: Training and Validation Accuracy for Final Iteration

Figure 2 classifies the probabilities for each defect category; image 1 confidently identified the correct defect as Crack showing a higher percentage of the classified model (69.9%). Image 2 had the best classified defect at 85% for Missing Head, whilst image 3 misclassified Paint Off as Crack with 67.8% confidence, indicating an overlap in features and a confusion.

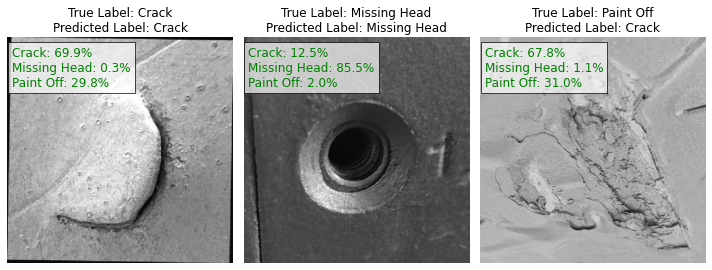


Figure 2: Classification Labelling for Final Iteration

**Discussion/Conclusion**

The challenges came from the plateau of the validation accuracy as it showed signs of overfitting at around 68-71%. For future improvements the code will be structured with higher dropout rates to reduce that overfitting, attempt at using a scheduler to adapt to learning rates dynamically, or even simplifying the architecture to balance training time and architecture. For classification, target data augmentation (brightness, cropping, rotation) to enhance classification.

The model was strong and had high accuracies, while it still struggled between Crack and Paint Off, the iterations taken to get to this point were learned from and the above cases are now learned to incorporate in future iterations to help improve the overall output.