Toronto Metropolitan University

Department of Aerospace Engineering
Faculty of Engineering
& Architectural Science

Semester (Term, Year)	**Fall, 2023**
Course Code	**AER850**
Course Section	**1**
Course Title	**Introduction to Machine Learning**
Course Instructor	**Dr. Reza Faieghi**
Submission	**Project Report**
Submission No.	**1**
Submission Due Date	**11/26/2023**
Title	**Project 2: Machine Learning**
Submission Date	**11/26/2023**

Submission by (Name):	**Student ID (XXXX1234)**	**Signature**
Mehtab Singh	**XXXX60754**	

By signing the above you attest that you have contributed to this submission and confirm that all work you contributed to this submission is your own work. Any suspicion of copying or plagiarism in this work will result in an investigation of Academic Misconduct and may result in a ”0” on the work, and “F” in the course, or possibly more severe penalties, as well as a Disciplinary Notice on your academic record under the Academic Integrity Policy 60, which can be found at www.torontomu.ca/senate/policies/

Aerospace Assignment Cover as of May 2022

Table of Contents

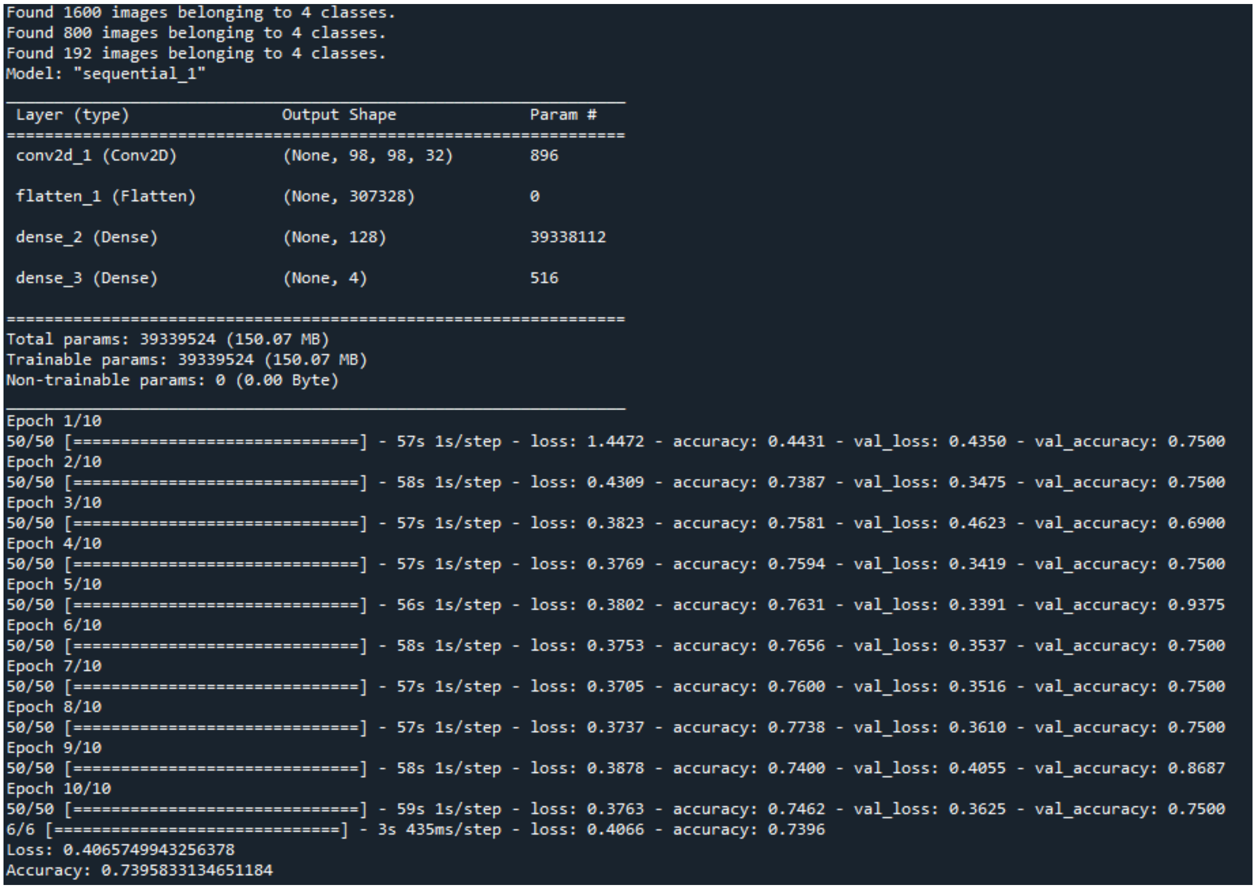
List of Figures

1. Introduction and Project Outline

The project is to detect cracks within a metal plate using Convoluted Neural Networks (CNN).

2. Results & Discussion

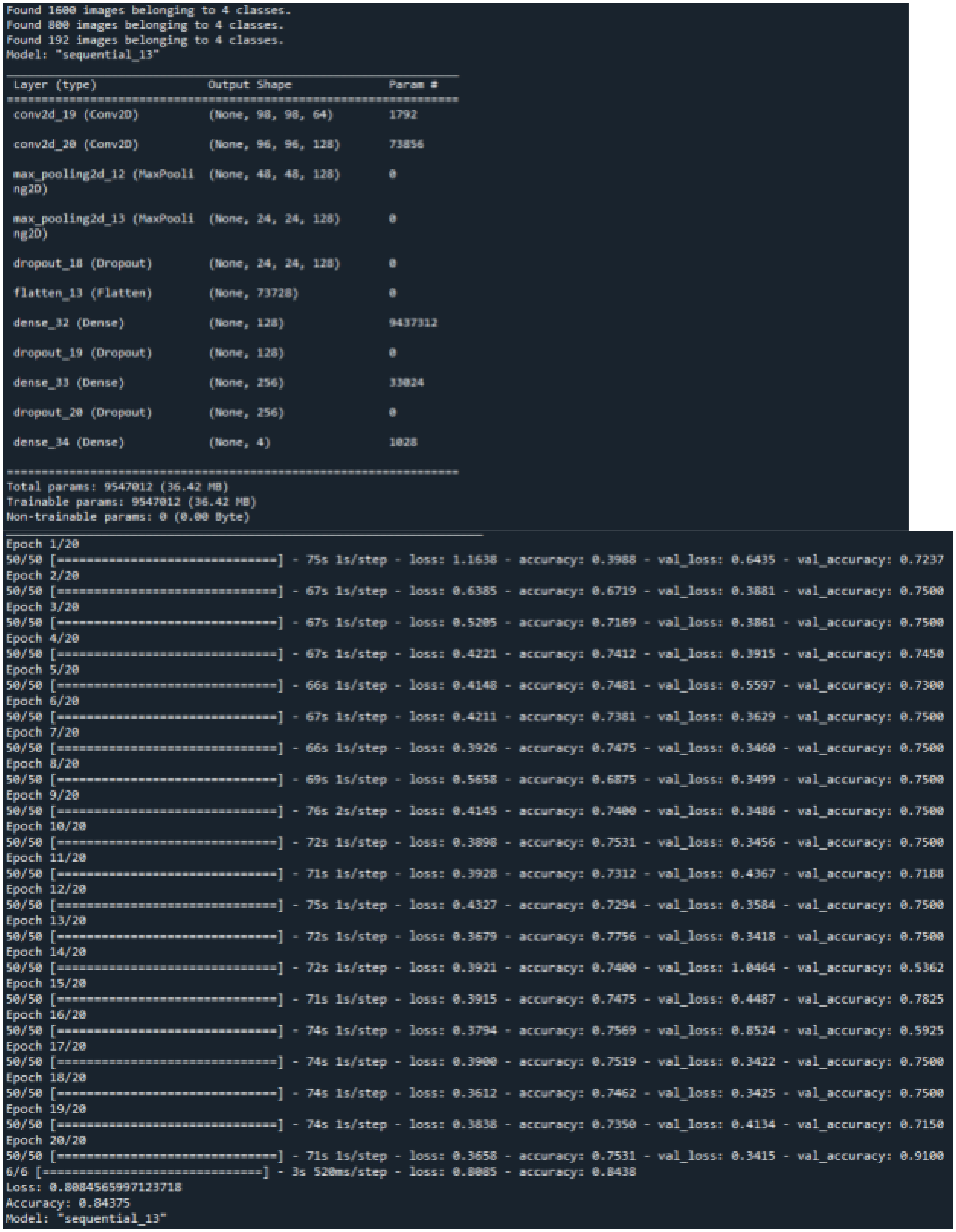
The following are the steps taken in the project, the code for which is uploaded in the GitHub repository which is linked in the Appendix as well as a zip file submitted alongside the report. First was setting up the code to be used for running and training the CNN with the train and validation data. The following below is the first run of the code.



```
Found 1600 images belonging to 4 classes.
Found 800 images belonging to 4 classes.
Found 192 images belonging to 4 classes.
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_1 (Conv2D)           (None, 98, 98, 32)        896

 flatten_1 (Flatten)         (None, 307328)            0

 dense_2 (Dense)             (None, 128)               39338112

 dense_3 (Dense)             (None, 4)                 516

=================================================================
Total params: 39339524 (150.07 MB)
Trainable params: 39339524 (150.07 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
Epoch 1/10
50/50 [==============================] - 57s 1s/step - loss: 1.4472 - accuracy: 0.4431 - val_loss: 0.4350 - val_accuracy: 0.7500
Epoch 2/10
50/50 [==============================] - 58s 1s/step - loss: 0.4309 - accuracy: 0.7387 - val_loss: 0.3475 - val_accuracy: 0.7500
Epoch 3/10
50/50 [==============================] - 57s 1s/step - loss: 0.3823 - accuracy: 0.7581 - val_loss: 0.4623 - val_accuracy: 0.6900
Epoch 4/10
50/50 [==============================] - 57s 1s/step - loss: 0.3769 - accuracy: 0.7594 - val_loss: 0.3419 - val_accuracy: 0.7500
Epoch 5/10
50/50 [==============================] - 56s 1s/step - loss: 0.3802 - accuracy: 0.7631 - val_loss: 0.3391 - val_accuracy: 0.9375
Epoch 6/10
50/50 [==============================] - 58s 1s/step - loss: 0.3753 - accuracy: 0.7656 - val_loss: 0.3537 - val_accuracy: 0.7500
Epoch 7/10
50/50 [==============================] - 57s 1s/step - loss: 0.3705 - accuracy: 0.7600 - val_loss: 0.3516 - val_accuracy: 0.7500
Epoch 8/10
50/50 [==============================] - 57s 1s/step - loss: 0.3737 - accuracy: 0.7738 - val_loss: 0.3610 - val_accuracy: 0.7500
Epoch 9/10
50/50 [==============================] - 58s 1s/step - loss: 0.3878 - accuracy: 0.7400 - val_loss: 0.4055 - val_accuracy: 0.8687
Epoch 10/10
50/50 [==============================] - 59s 1s/step - loss: 0.3763 - accuracy: 0.7462 - val_loss: 0.3625 - val_accuracy: 0.7500
6/6 [==============================] - 3s 435ms/step - loss: 0.4066 - accuracy: 0.7396
Loss: 0.4065749943256378
Accuracy: 0.7395833134651184
```

Figure 1: First Epoch run of CNN Code

To get better understanding and more accurate validation, extra layers of the Conv2D, MaxPooling, Dense and Dropout to increase the accuracy. The end result was spikes in inaccurate validation occurring in the run, which could be due to the dropout of the layers causing less data to be included in the validation.



```
Found 1600 images belonging to 4 classes.
Found 800 images belonging to 4 classes.
Found 192 images belonging to 4 classes.
Model: "sequential_13"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_19 (Conv2D)          (None, 98, 98, 64)        1792

 conv2d_20 (Conv2D)          (None, 96, 96, 128)       73856

 max_pooling2d_12 (MaxPooli  (None, 48, 48, 128)       0
 ng2D)

 max_pooling2d_13 (MaxPooli  (None, 24, 24, 128)       0
 ng2D)

 dropout_18 (Dropout)        (None, 24, 24, 128)       0

 flatten_13 (Flatten)        (None, 73728)             0

 dense_32 (Dense)            (None, 128)               9437312

 dropout_19 (Dropout)        (None, 128)               0

 dense_33 (Dense)            (None, 256)               33024

 dropout_20 (Dropout)        (None, 256)               0

 dense_34 (Dense)            (None, 4)                 1028

=================================================================
Total params: 9547012 (36.42 MB)
Trainable params: 9547012 (36.42 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
Epoch 1/20
50/50 [==============================] - 75s 1s/step - loss: 1.1638 - accuracy: 0.3988 - val_loss: 0.6435 - val_accuracy: 0.7237
Epoch 2/20
50/50 [==============================] - 67s 1s/step - loss: 0.6385 - accuracy: 0.6719 - val_loss: 0.3881 - val_accuracy: 0.7500
Epoch 3/20
50/50 [==============================] - 67s 1s/step - loss: 0.5205 - accuracy: 0.7169 - val_loss: 0.3861 - val_accuracy: 0.7500
Epoch 4/20
50/50 [==============================] - 67s 1s/step - loss: 0.4221 - accuracy: 0.7412 - val_loss: 0.3915 - val_accuracy: 0.7450
Epoch 5/20
50/50 [==============================] - 66s 1s/step - loss: 0.4148 - accuracy: 0.7481 - val_loss: 0.5597 - val_accuracy: 0.7300
Epoch 6/20
50/50 [==============================] - 67s 1s/step - loss: 0.4211 - accuracy: 0.7381 - val_loss: 0.3629 - val_accuracy: 0.7500
Epoch 7/20
50/50 [==============================] - 66s 1s/step - loss: 0.3926 - accuracy: 0.7475 - val_loss: 0.3460 - val_accuracy: 0.7500
Epoch 8/20
50/50 [==============================] - 69s 1s/step - loss: 0.5658 - accuracy: 0.6875 - val_loss: 0.3499 - val_accuracy: 0.7500
Epoch 9/20
50/50 [==============================] - 76s 2s/step - loss: 0.4145 - accuracy: 0.7400 - val_loss: 0.3486 - val_accuracy: 0.7500
Epoch 10/20
50/50 [==============================] - 72s 1s/step - loss: 0.3898 - accuracy: 0.7531 - val_loss: 0.3456 - val_accuracy: 0.7500
Epoch 11/20
50/50 [==============================] - 71s 1s/step - loss: 0.3928 - accuracy: 0.7312 - val_loss: 0.4367 - val_accuracy: 0.7188
Epoch 12/20
50/50 [==============================] - 75s 1s/step - loss: 0.4327 - accuracy: 0.7294 - val_loss: 0.3584 - val_accuracy: 0.7500
Epoch 13/20
50/50 [==============================] - 72s 1s/step - loss: 0.3679 - accuracy: 0.7756 - val_loss: 0.3418 - val_accuracy: 0.7500
Epoch 14/20
50/50 [==============================] - 72s 1s/step - loss: 0.3921 - accuracy: 0.7400 - val_loss: 1.0464 - val_accuracy: 0.5362
Epoch 15/20
50/50 [==============================] - 71s 1s/step - loss: 0.3915 - accuracy: 0.7475 - val_loss: 0.4487 - val_accuracy: 0.7825
Epoch 16/20
50/50 [==============================] - 74s 1s/step - loss: 0.3794 - accuracy: 0.7569 - val_loss: 0.8524 - val_accuracy: 0.5925
Epoch 17/20
50/50 [==============================] - 74s 1s/step - loss: 0.3900 - accuracy: 0.7519 - val_loss: 0.3422 - val_accuracy: 0.7500
Epoch 18/20
50/50 [==============================] - 74s 1s/step - loss: 0.3612 - accuracy: 0.7462 - val_loss: 0.3425 - val_accuracy: 0.7500
Epoch 19/20
50/50 [==============================] - 74s 1s/step - loss: 0.3838 - accuracy: 0.7350 - val_loss: 0.4134 - val_accuracy: 0.7150
Epoch 20/20
50/50 [==============================] - 71s 1s/step - loss: 0.3658 - accuracy: 0.7531 - val_loss: 0.3415 - val_accuracy: 0.9100
6/6 [==============================] - 3s 520ms/step - loss: 0.8085 - accuracy: 0.8438
Loss: 0.8084565997123718
Accuracy: 0.84375
Model: "sequential_13"
```

Figure 2: Final Epoch run of CNN Code

The following below is the first and final runthrough of the code for the model performance.

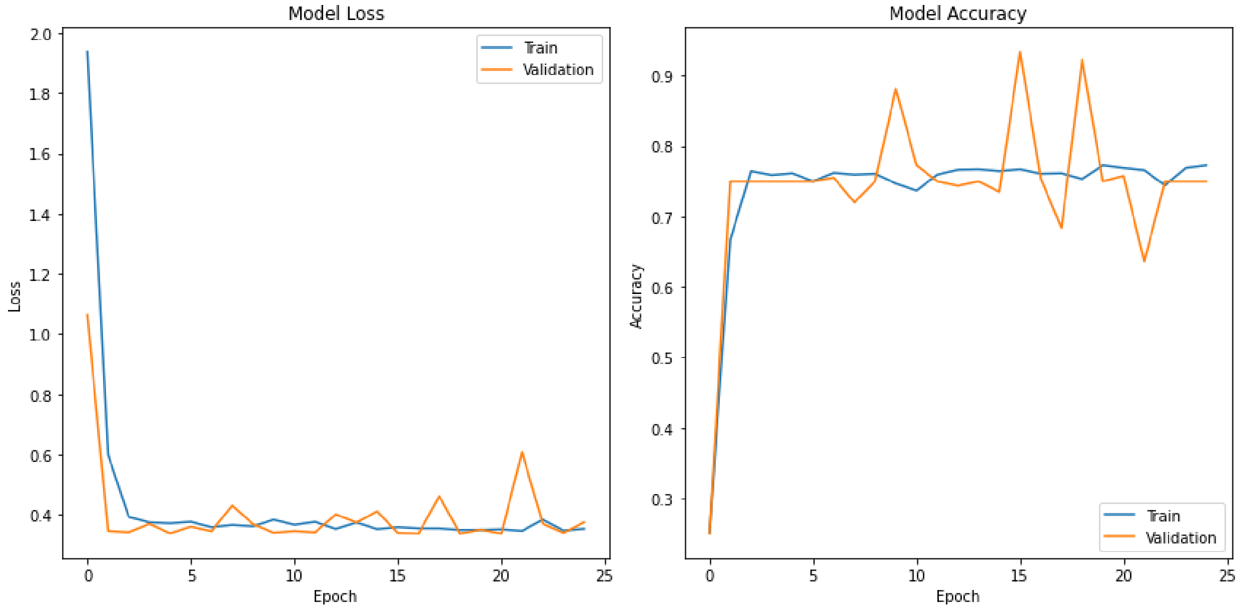


Figure 3: First Model Performance of the Code Run

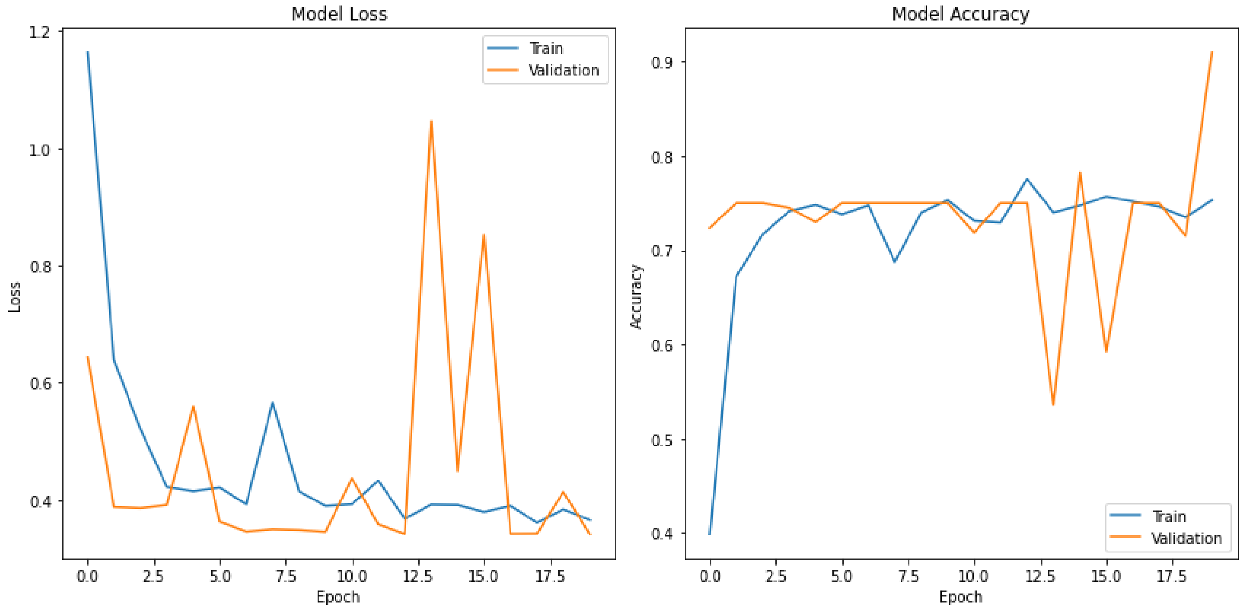


Figure 4: Final Model Performance of the Code Run

The figure below shows the medium crack test, on the left, and the large crack test, on the right.

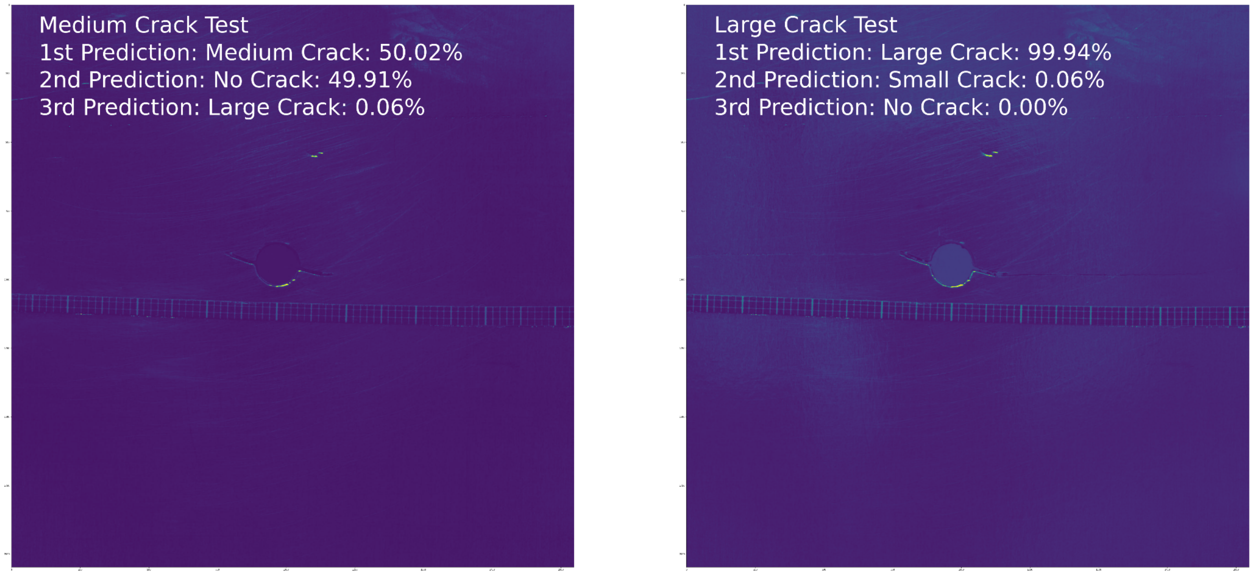

Medium Crack Test
1st Prediction: Medium Crack: 50.02%
2nd Prediction: No Crack: 49.91%
3rd Prediction: Large Crack: 0.06%
Large Crack Test
1st Prediction: Large Crack: 99.94%
2nd Prediction: Small Crack: 0.06%
3rd Prediction: No Crack: 0.00%

Figure 5: Final Crack Output for Testing

The large crack test was correct, The reason for the discrepancy for the medium crack is due to the validation spikes causing a great loss in accuracy.

3. Conclusion

This project helped understand the practical use of CNN and visual detection using metal cracks.

References

[1] Dr. Reza Faieghi “AER850 Project #2” https://courses.torontomu.ca/d2l/le/content/804341/viewContent/5398953/View [Accessed 11/15/2023]

Appendix

The following link is for the GitHub repository:
https://github.com/Mehtab0Singh/AER850_Project_2_MehtabSingh

The files with all the code and training, validation, and testing pictures are linked with this report in submission.