Toronto Metropolitan University

Department of Aerospace Engineering
Faculty of Engineering
& Architectural Science

Semester (Term, Year)	F23
Course Code	AER850
Course Section	01
Course Title	Intro to Machine Learning
Course Instructor	Prof. Faieghi
Submission	Project
Submission No.	01
Submission Due Date	26/11/23
Title	Project 2
Submission Date	26/11/23

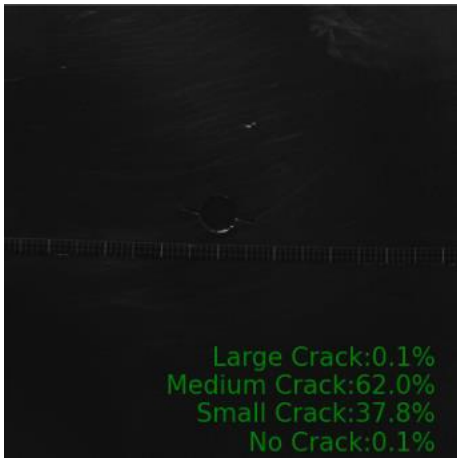
Submission by (Name):	**Student ID (XXXX1234)**	**Signature**
Emily Birkenhead	6788	Emily Birkenhead

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/

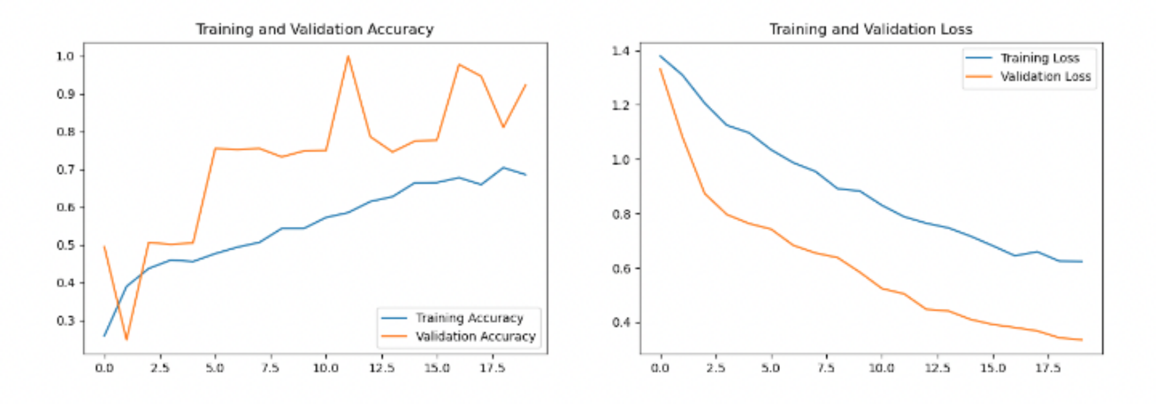
Github: https://github.com/ebirkenhead/AER850_Project2.git

This project report uses image recognition to determine the level of fatigue on an aircraft component. The program is tested on various sizes of cracks, learning how to classify them. It is then given two sample images and is required to determine their classification.

Expected image:



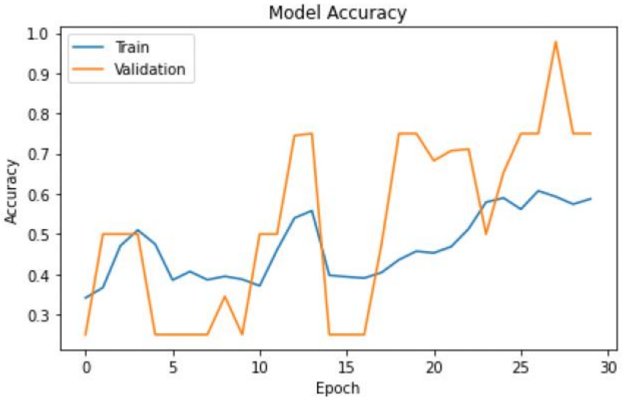
Expected graphs (trend):

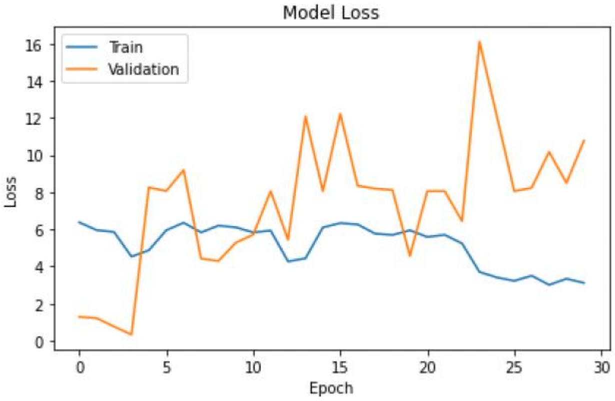


For the purposes of this report, the program was run twice for both images. The second time the program was run, changes were made to the code in order to improve the accuracy and reduce the loss.

First Run

The program run resulted in the following plots.

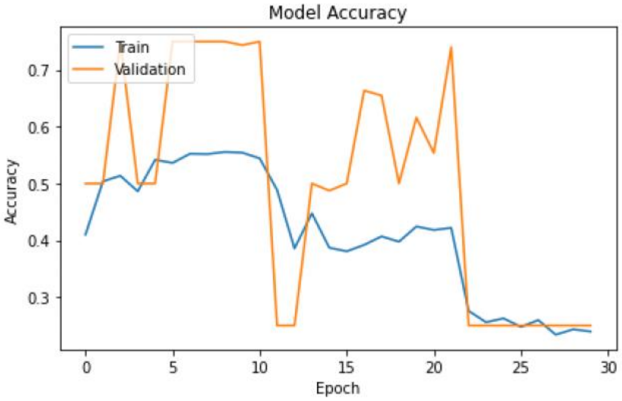


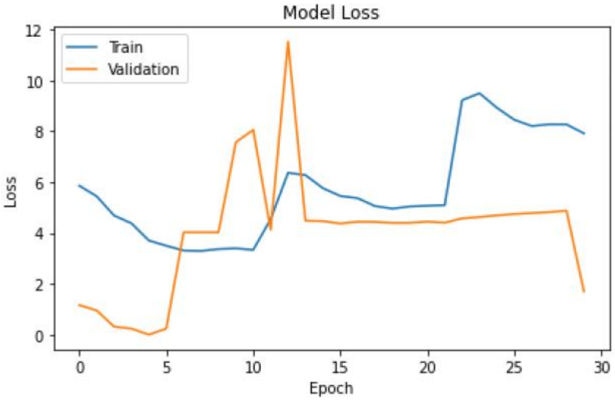


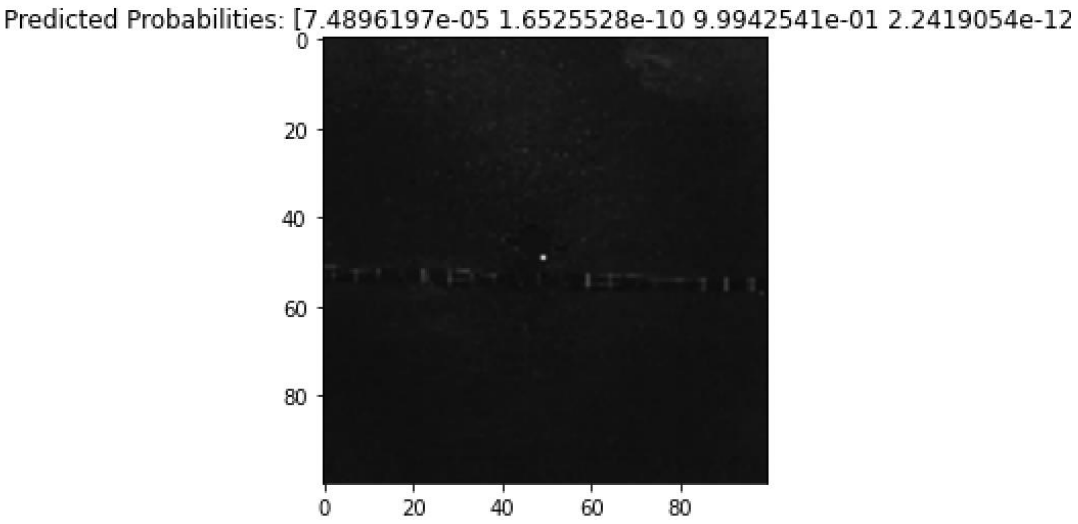
loss: 12.2565 - accuracy: 0.7396

These plots resulted in a high accuracy but also an unexpectedly high loss. It was expected for the accuracy plot to increase linearly, and the loss to decrease linearly, however, this did not happen.

Crack__20180419_06_19_09,915.bmp (Medium Image)







```
Predicted Probabilities: [7.4896197e-05 1.6525528e-10 9.9942541e-01 2.2419054e-12
```

The predicted values of this image are completely wrong. The values themselves are too small.

Crack__20180419_13_29_14,846.bmp (Large Image)

When run for the Large image, an error occurred, implying that the probabilities of the image classification could not be calculated. Nothing was changed between the medium and large image code other than than the source. Therefore, changes are needed in order for an acceptable prediction to be made.

Changes made

Based on the results of the first test run, changes were made to the program in order to improve the results, in accuracy, loss, and outputted image.

First, the epoch was increased from 30 to 45. This will allow the program to spend more time reaching its accuracy and loss.

The following lines were added:

```
    model.add(Dense(128, activation=activation))
    model.add(Dropout(0.2))
```

(Line 99-100)

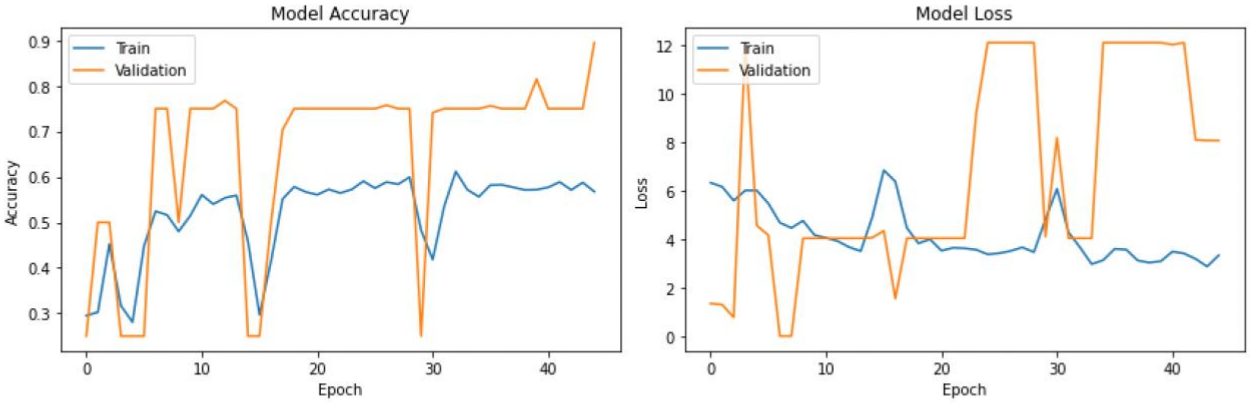
```
    model.add(Conv2D(64, (3, 3), strides=(1,1), activation=activation))
```

(Line 84)

This was in order to give the program more hidden layers to work with. It did increase the runtime but also increased precision of the code, creating more consistent results.

Second run

With the changes made, the following was produced:

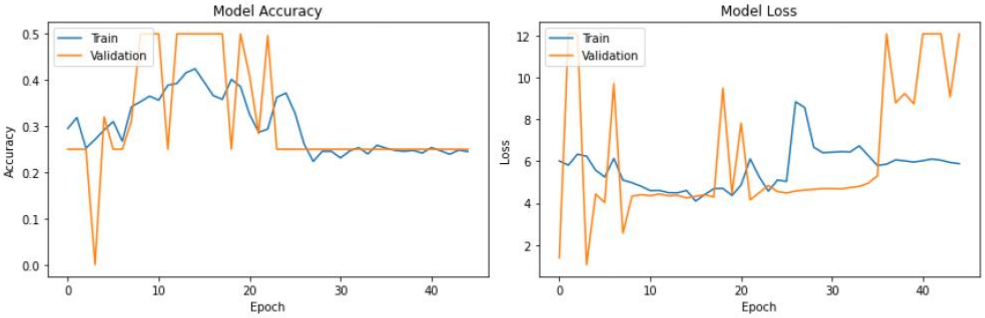


This resulted in a Validation Accuracy of: 0.8541666865348816

The validation accuracy was not a linear increase as expected. Instead, it began increasing linearly before dropping suddenly. It continued this pattern throughout, until around 30 epochs, where it became quite linear, before suddenly increasing to a very high accuracy for its 45th epoch. The Train Accuracy remained quite linear, at around 0.55, with multiple sudden jumps to a much lower accuracy.

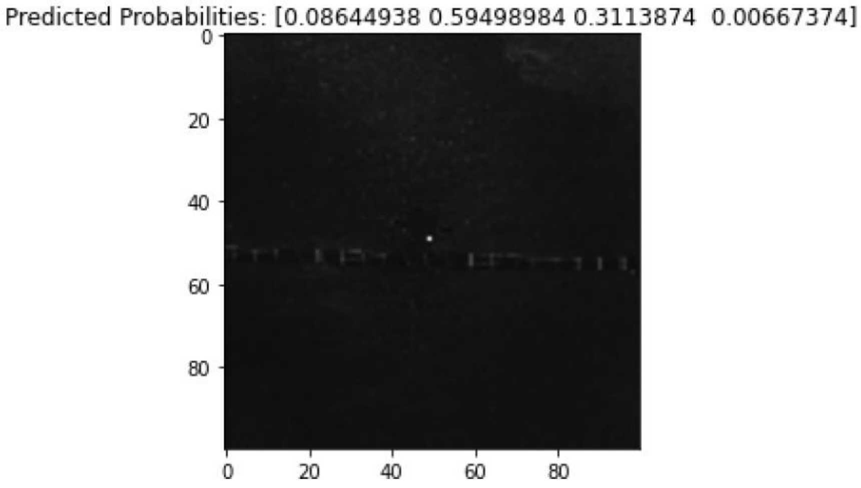
The loss also did not behave as expected. The Validation loss started very low, but jumped all over the place, ending with a very high loss of approximately 8. The Trian loss did have a steady decrease, as expected, but also experienced multiple sudden peaks, and did not decrease below approximately 4.

Crack__20180419_06_19_09,915.bmp (Medium Image)


Model Accuracy

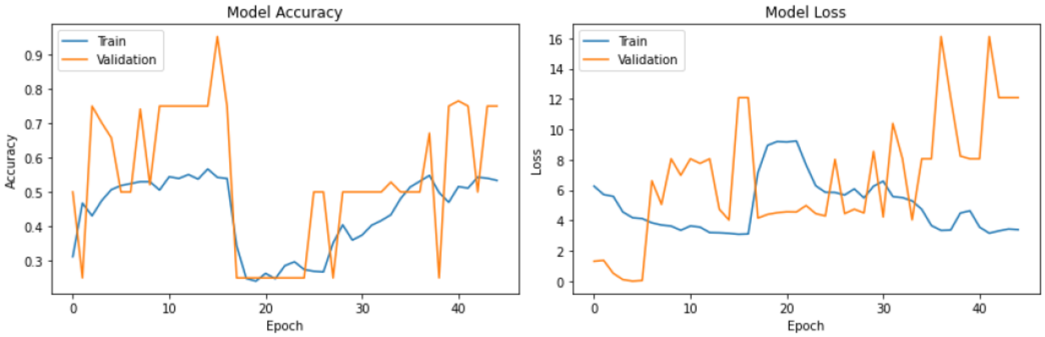
Model Loss

The following was the resulting image:

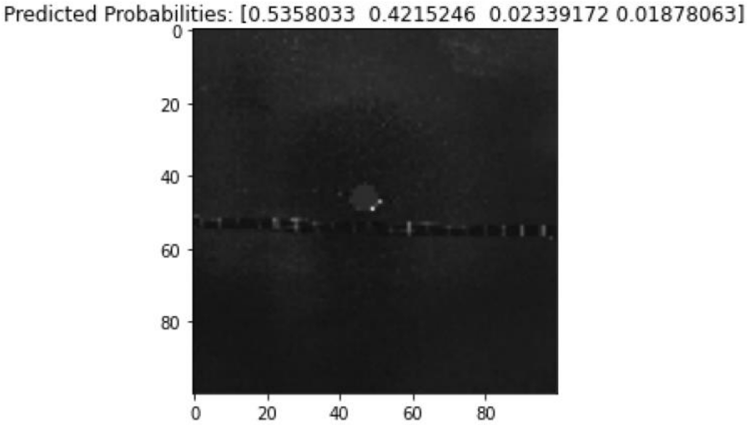

Predicted Probabilities: [0.08644938 0.59498984 0.3113874 0.00667374]

This image was predicted to be medium (59.4%). This correct. However, a higher percentage of probability was expected.

Crack__20180419_13_29_14,846.bmp (Large Image)



The result was the following image:


Predicted Probabilities: [0.5358033 0.4215246 0.02339172 0.01878063]
0
20
40
60
80
0
20
40
60
80

This image was predicted to be small (53.5%). This is incorrect. However, the prediction for medium is 42.1%, as these values are very close together, likely something in the classification method was wrong.

When run for both size images, the accuracy experienced the same unusual trend. The loss behaved in much the same way as before.

Conclusion

The second run was deemed to be the better of the two options. However, for the large image, the first run held a much higher accuracy.

Overall, the program behaved as expected in terms of classifying images. However, the trend of accuracy and loss that were expected in order for the program to make the correct classifications were not achieved, and the loss consistently remained too high.