Toronto Metropolitan University

Department of Aerospace Engineering
Faculty of Engineering & Architectural Science

Course Code:	AER 850
Course Title:	Introduction to Machine Learning
Semester:	F2024
Instructor:	Dr. Reza Faigehi
Submission:	Project 2
Due Date:	Sunday, November 17th, 2024

Title: Project 2

Section Number:	02
Submission Date:	Sunday, November 17th, 2024
Submission By:	Arjun Tripathi

Name	Student Number (XXXX99999)	Signature
Arjun Tripathi	XXXX21964	

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 Student Code of Academic Conduct, which can be found online at www.torontomu.ca/senate/policies/

Table of Contents

Introduction

This project aims to develop an accurate machine-learning model with a deep convolutional neural network to classify images. These images are of various cracks that are usually observed on aircraft structures. Such cracks are classified into 3 types, normal crack, missing-head, and paint-off. The following report will present the evaluation of the model and 3 images that were supplied to the model to test its prediction accuracy.

Results

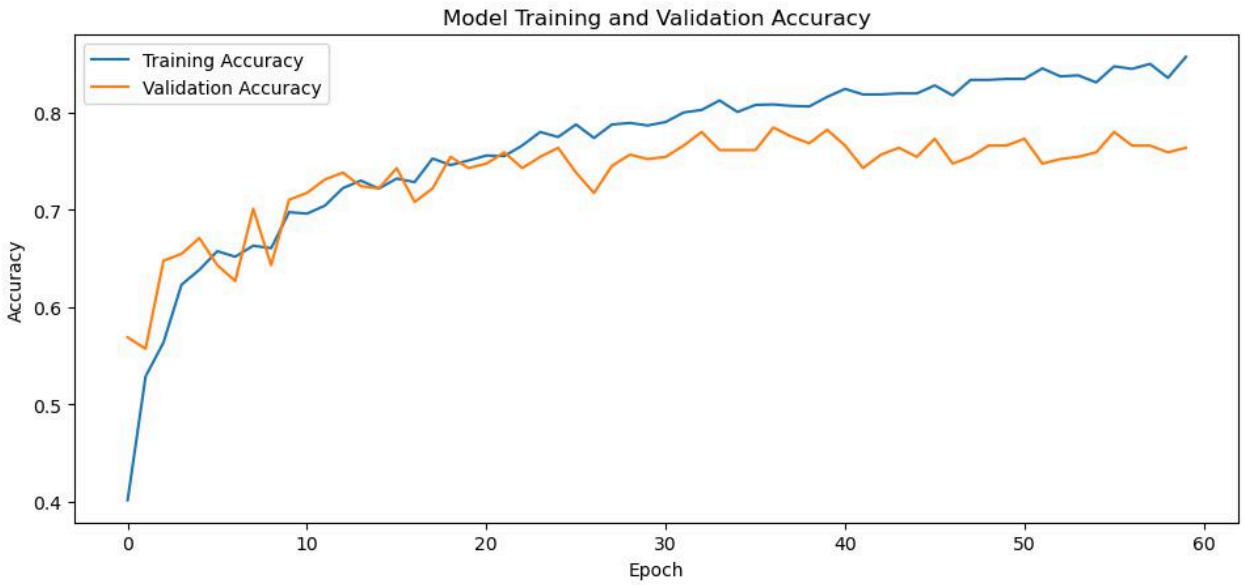

Figure 1.0: Model's Training and Validation Accuracy

Figure 2.0: Model's Training and Validation Loss

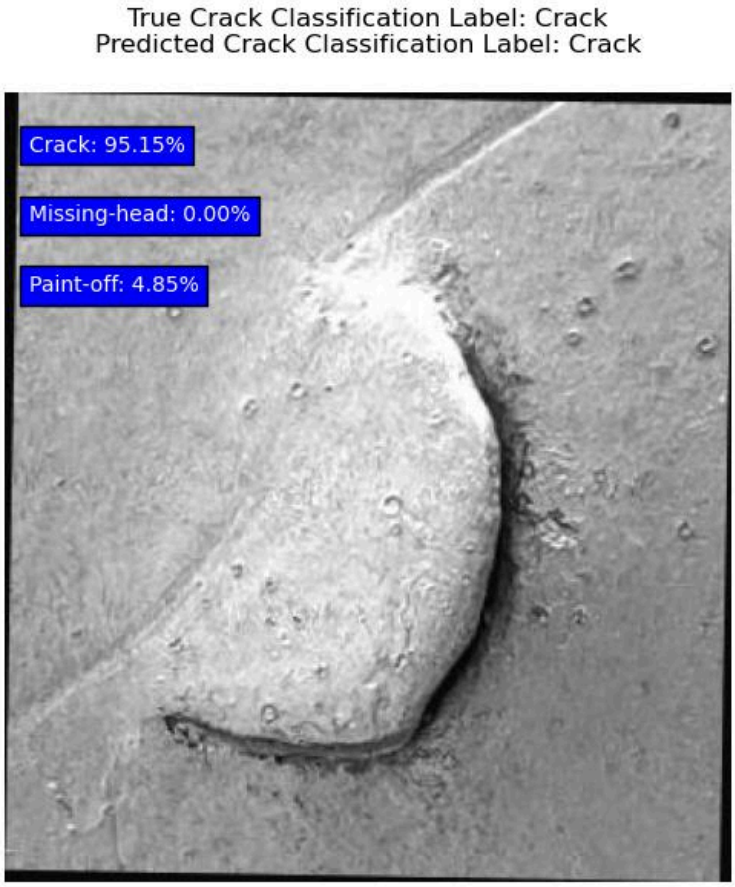

Figure 3.0: Test Image of Crack

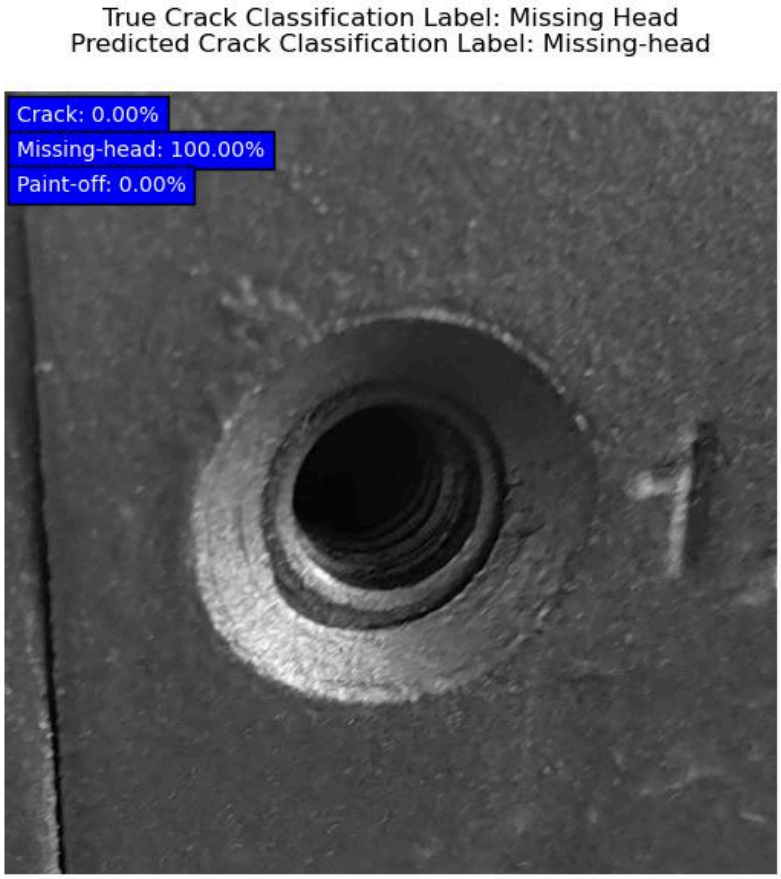

Figure 4.0: Test Image of Missing-Head

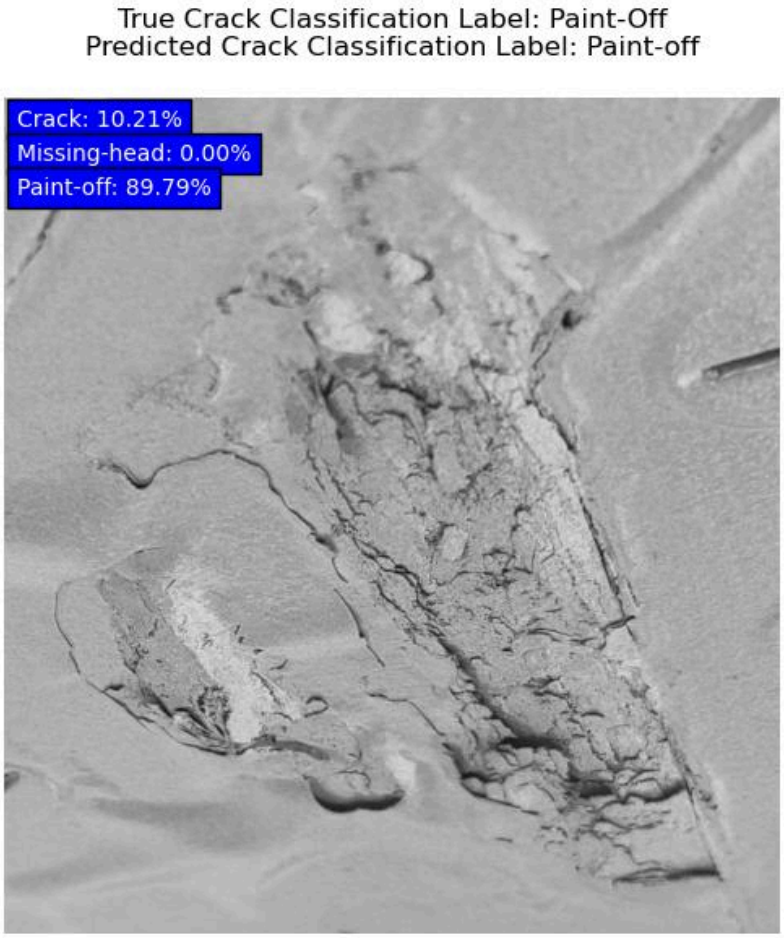

Figure 5.0: Test Image of Paint-Off

Discussion

This model consisted of 60 epochs with 4 convolutional layers, 1 flattened layer and 2 dense layers. Convolutional layers 1, 3 and 4 utilized the ReLU activation function, while the second layer utilized the LeakyReLU activation function. All convolutional layers except the input layer (the first one) had dropout layers. Furthermore, the first dense layer utilized the ReLU activation function, while the last dense layer, the output layer, utilized the softmax activation function. The model can be viewed in depth through the GitHub link found in Appendix A.1.

According to Figure 1.0, the model's accuracy with the validation and training datasets was about 77% and 85%, respectively. During the 60 epochs, some noise is observed in the train and validation dataset accuracies. The validation dataset has more noise than the training dataset. There is also a noticeable gap between the training and validation accuracy, mainly between the 30th and 60th epochs. This gap is not extreme, so it is not of concern but suggests room for improvement.

According to Figure 2.0, the model’s training and validation loss steadily declines after each epoch. The training loss declines to a final loss of around 0.3500, and the validation loss declines to around 0.6500. There are some fluctuations in the validation loss which suggests room for improvement.

Lastly, the model accurately predicted each image’s classification with the test images provided, as seen in Figures 3.0 to 5.0. The average accurate prediction percentage is around 90%, indicating that the model performed well. However, further improvements can be made to the model due to the lack of adequate resources, such as hardware performance and training time, which made these improvements unviable. If additional resources are given, the model could be improved through changes made to the dropout or convolutional layer parameters.

Conclusion

To conclude, this project effectively allows the student to learn the architecture of a deep convolutional neural network and understand how it is utilized in image classification. The model developed was fairly accurate and could predict the test images successfully. Further improvements to the model could be made to increase the accuracy and decrease losses. However, the lack of hardware resources and time-limited these improvements.

Appendix

Appendix A.1 - Project Code:

GitHub Link: https://github.com/Arjunt10/AER850Project2.git