DATS6203 - Machine Learning – II

Multi-Label Classification

Individual Project Report

Author(s): Varun .R. Shah

12/13/2022

Overview of the Project

The goal of this project was to implement Multi-Label Image Classification using Convoluted Neural Network. We used various other pretrained models such ResNet101, EffecientNet-B3 and RegNet_x_800.

The data consists of approximately 10,000 images, with each image having between 0 to 16 different labels. Its taken from the IDD (Indian Driving Dataset) dataset, which is a dataset for road scene understanding in unstructured environments used for semantic segmentation and object detection for autonomous driving.

We use different fine-tuning methods such as data augmentation, changes in model definition, changes in optimization algorithm, changes in batch size and other parameters.

For metrics, we use Accuracy Score and Hamming Loss.

THE GEORGE WASHINGTON UNIVERSITY
WASHINGTON, DC

Roles and Responsibilities

In this project, my role was to improve Convoluted Neural Network model definition and fine tune the EfficientNet-B3 model.

I used different data augmentation techniques such as Random Rotation, Affine transformation, Normalization and resizing of the image to improve the model. While defining the Convoluted Neural Network, I tried to improve the model by adding multiple Convoluted Layers and MaxPooling layers and also make changes and add Dense layers to the model definition.

In addition to this, I implemented different fine-tuning techniques to improve the EfficientNet Pretrained Model. With changes in Learning Rate, Batch Size, Number of Epochs and Optimization algorithms.

I was also involved in preparing the Group Project Report while my team members managed other aspects of the submission requirements.

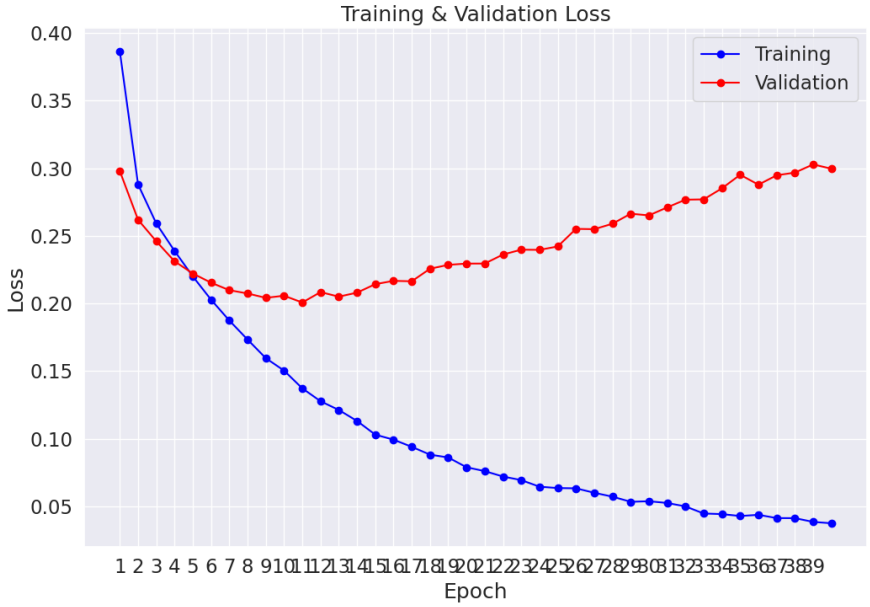
The results were as follows from Fine-Tuning:

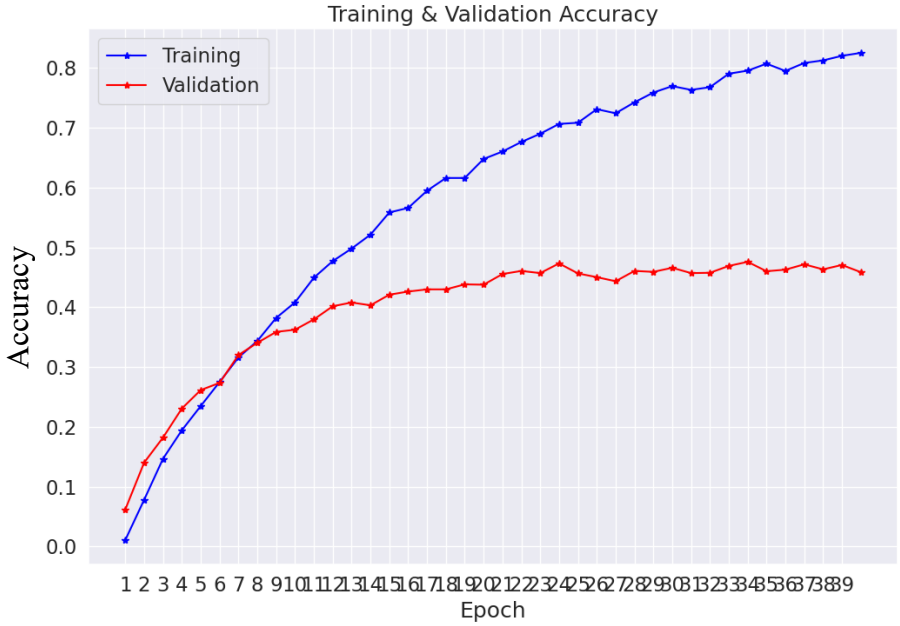
Epoch	Batch Size	Augmentation	Optimazation	Learning Rate	CNN Layers	Pretrained Model	Accuracy
10	32	RandomRotation(degrees=45 Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])]	Adam	0.0001		efficient_netb3	0.2
10	32	RandomRotation(degrees=45 Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])]	Adam	**0.001**		efficient_netb3	0.39
10	32	RandomRotation(degrees=45 Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])] RandomFlip	Adam	0.0001	(with 2 dense layers)	FALSE	0.05
10	32	RandomRotation(degrees=45 Normalize([0.485, 0.456, 0.406], [0.229, 0.224, 0.225])] RandomAffine(30, 70)	Adam	**0.001**	(with 3 dense layers)	FALSE	0.05

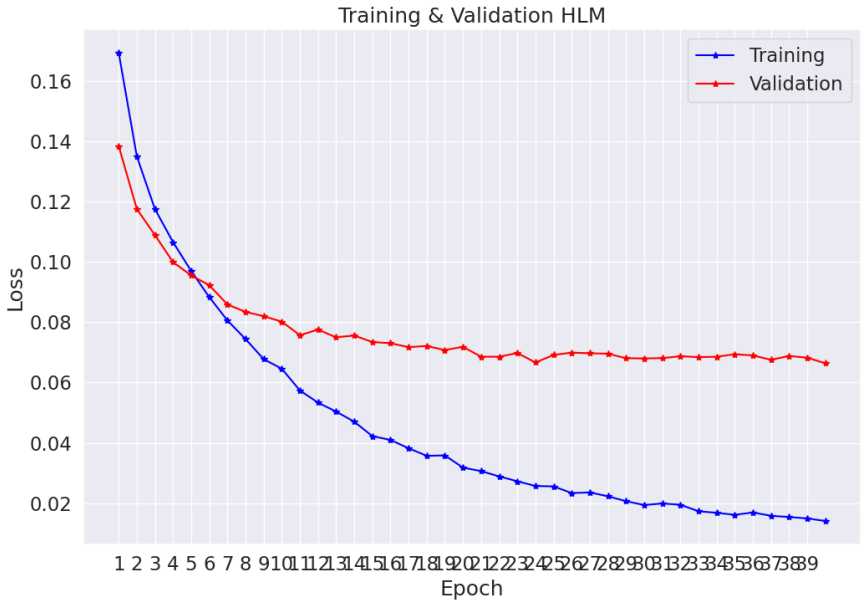
Metrics on 40 Epochs:

Test Accuracy: 46% and Test Hamming: 0.06681
Average Individual Class – F1: 0.68


THE GEORGE
WASHINGTON
UNIVERSITY
WASHINGTON, DC


Training & Validation Loss
Training
Validation
Loss
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39
Epoch


Training & Validation Accuracy
Training
Validation
Accuracy
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39
Epoch


Training & Validation HLM
Training
Validation
Loss
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39
Epoch

Conclusion

In conclusion, the project gave me the opportunity to work on CNN model architecture, learn the limitations of it and also work on fine-tuning the EfficientNet pretrained model.

Appendix

GitHub Link - https://github.com/IshanKuchroo/IDD-Indian-Driving-Dataset

Datasource:

https://paperswithcode.com/dataset/idd

References

Dr. Amir Jafari, George Washington University

https://ai.googleblog.com/2019/05/efficientnet-improving-accuracy-and.html

https://pytorch.org/vision/stable/transforms.html