Performance Metrics Overview

Batch Size	Test Accuracy	Test Loss
128	**0.6125**	**1.0799**
80	**0.6484**	**1.2878**
30	**0.6433**	**1.1785**

Larger Batch Sizes (128): Faster convergence but higher risk of overfitting and poorer generalization.

Medium Batch Sizes (80): Best trade-off between convergence speed and generalization, yielding the highest test accuracy.

Smaller Batch Sizes (30): Increased noise can aid generalization but may require more training epochs and careful tuning to avoid instability.

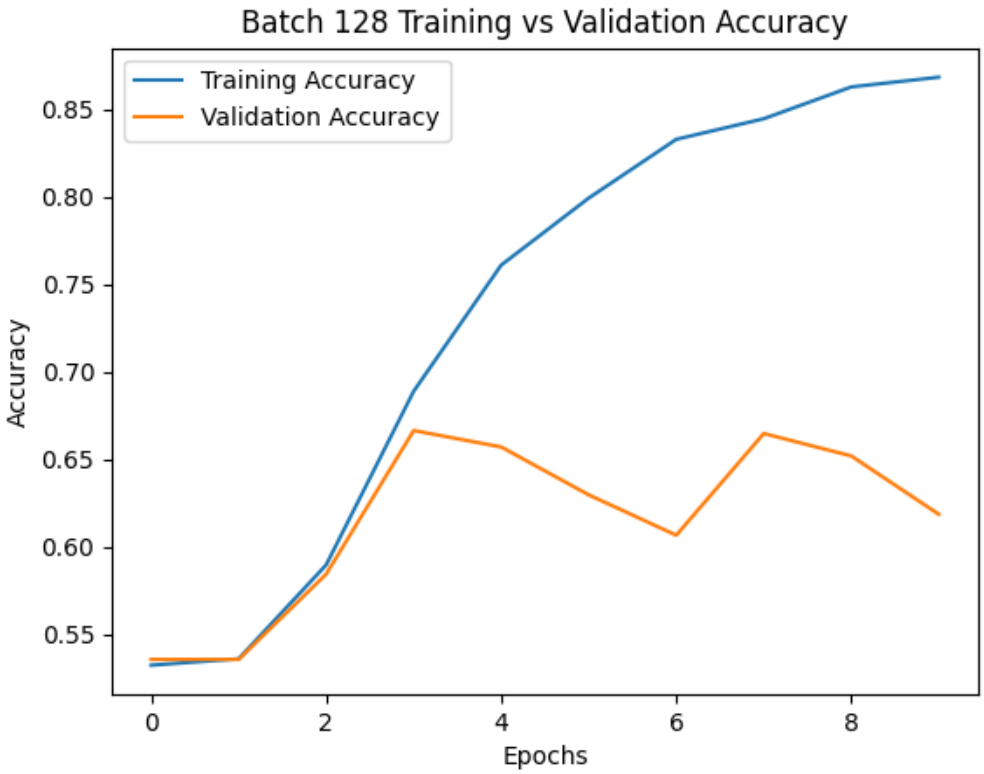
It seems the model benefits from moderate batch sizes that balance computational efficiency and the ability to generalize well to unseen data.

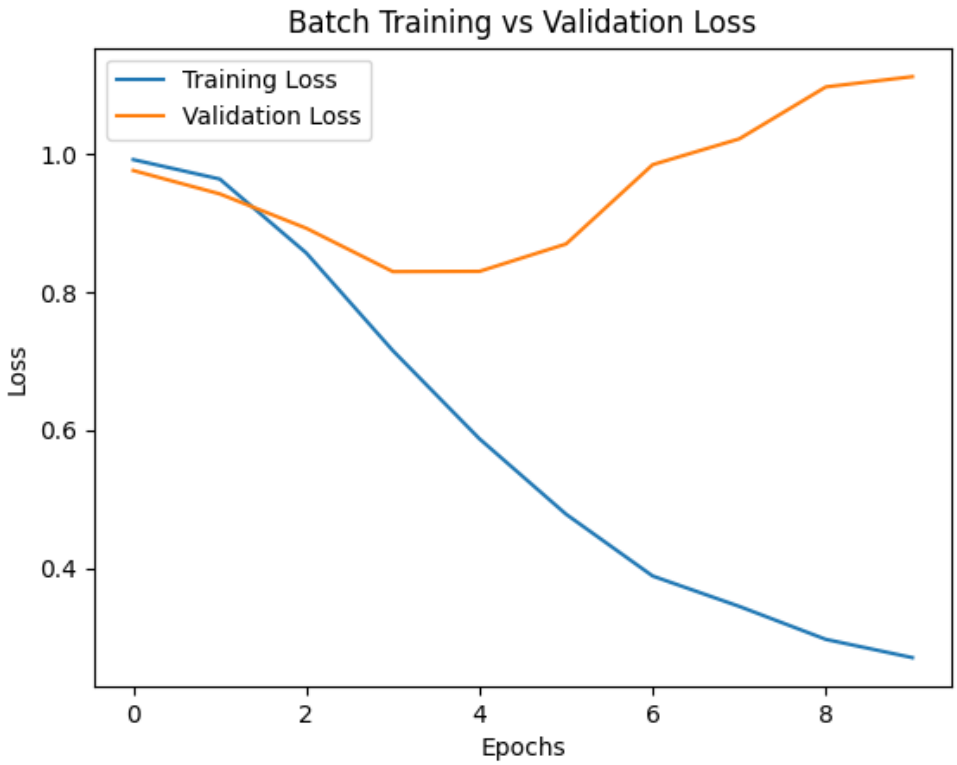
Overfitting is more pronounced with larger batch sizes, as evidenced by the higher training accuracy and lower validation accuracy.

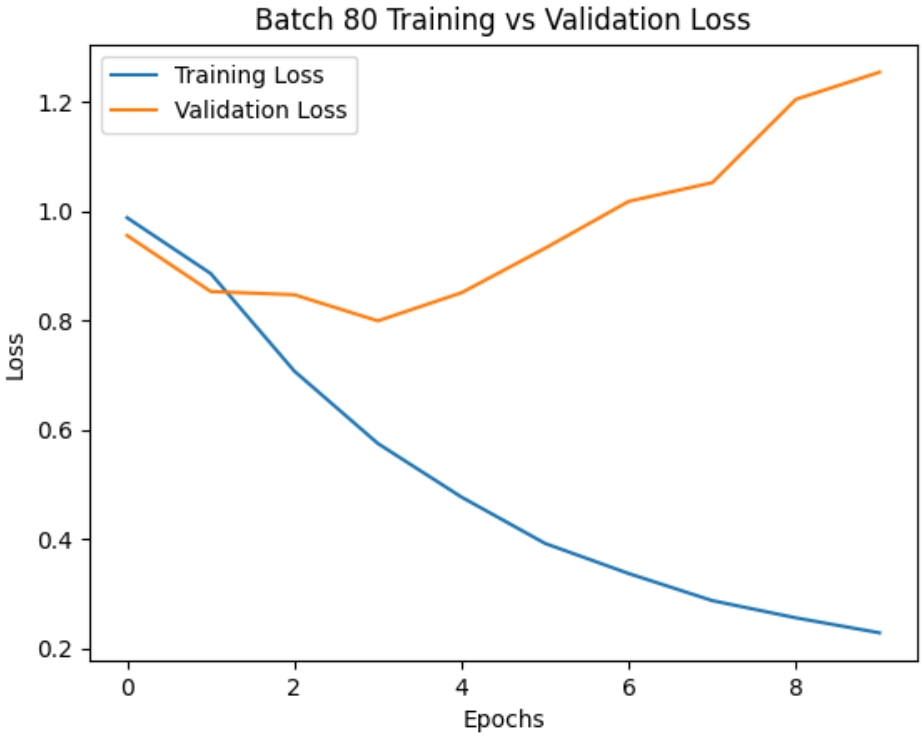
It seems that to further improve the model, we need to adjust the model to prevent overfitting possible altering the dropout rate in the future

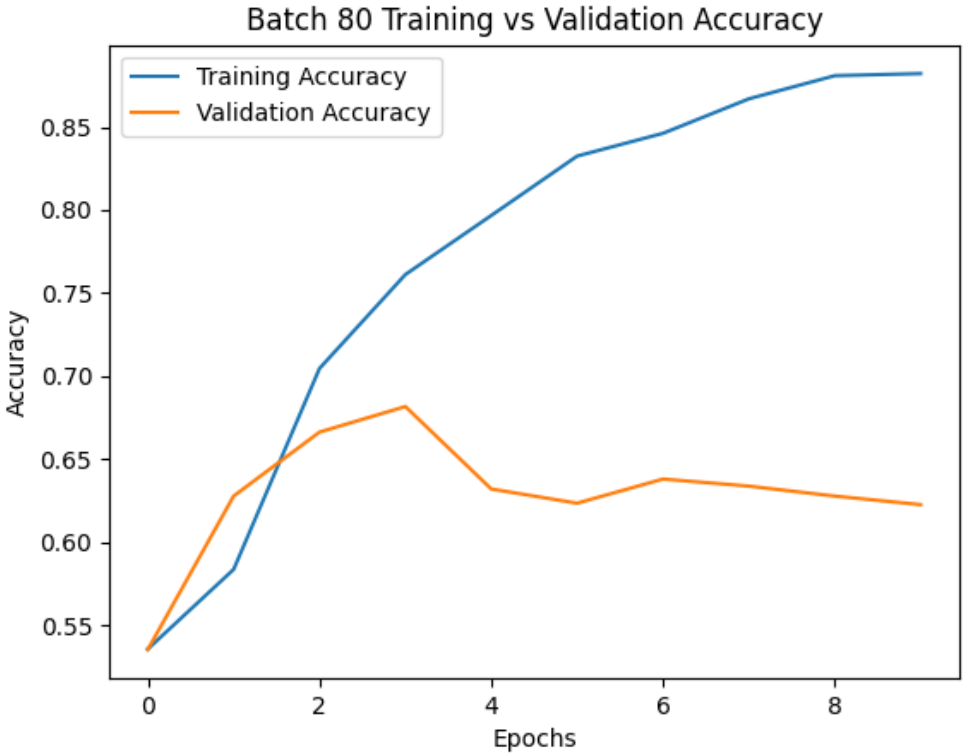
Linux Machine:

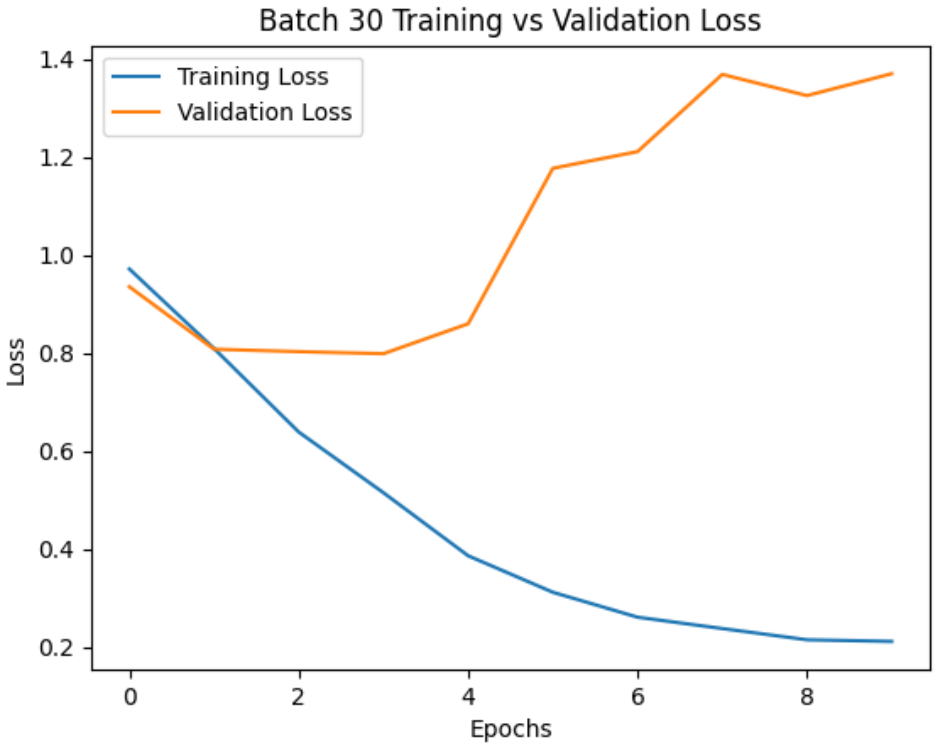
I was able to run the process on the Linux Machine however it was tedious since you needed to have specific package versions and it took a bit of effort to figure out which packages were needed

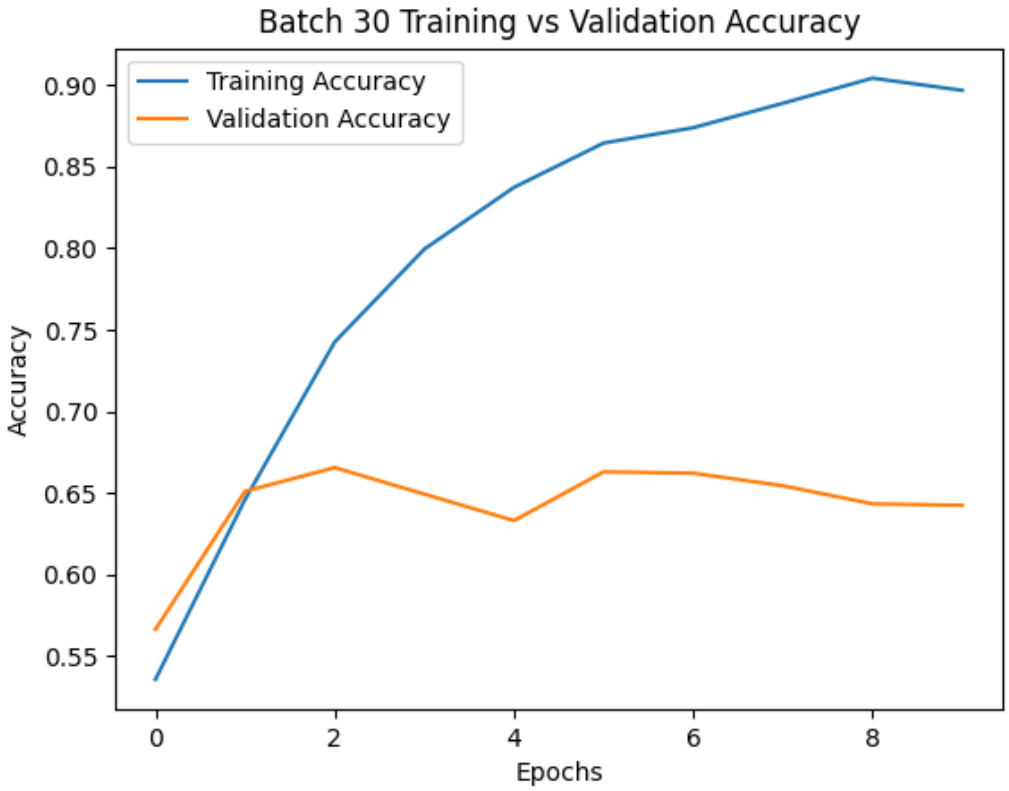

Batch 128 Training vs Validation Accuracy
Training Accuracy
Validation Accuracy
Accuracy
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0
2
4
6
8
Epochs


Batch Training vs Validation Loss
Training Loss
Validation Loss
Loss
1.0
0.8
0.6
0.4
0
2
4
6
8
Epochs


Batch 80 Training vs Validation Loss
Training Loss
Validation Loss
Loss
Epochs
1.2
1.0
0.8
0.6
0.4
0.2
0
2
4
6
8


Batch 80 Training vs Validation Accuracy
Training Accuracy
Validation Accuracy
Accuracy
Epochs
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0
2
4
6
8


Batch 30 Training vs Validation Loss
Training Loss
Validation Loss
Loss
Epochs
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0
2
4
6
8


Batch 30 Training vs Validation Accuracy
Training Accuracy
Validation Accuracy
Accuracy
Epochs
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0
2
4
6
8