Song Genre Classification using ViT-B16

By Nick Rommel and Alex Cerullo

Outline

- Introduction and motivation
- Related work
- Our Methodologies
  - Dataset
  - Transfer Learning
- Experiment and Analysis
  - Data Pre-Processing
  - Hyperparameter Sweep
  - Results
- Conclusion
  - Limitations and Ideas for Future Work

Introduction and Motivation

- With millions of songs being created continually, the need arise to efficiently classify these songs into genres
- Used a Pretrained ViT_B_16
- Fine tuned prediction head to classify the audio files of the GTZAN dataset that were converted to mel-spectrograms

Related Work

- Song genre classification via computer vision is not a new idea
  - However, most existing work uses CNNs and not ViTs
- Researchers by Niizumi et al is the most similar to our own
  - They devised a new self-supervised training with Masked Modeling Duo
  - Achieved 83% test accuracy using a baseline ViT
  - However, they fully trained the ViT, we employed transfer learning

Methods

The Dataset

- Used a modified version of GTZAN
- 1000 total tracks, 10 genres, 100 samples per genre
- Hosted on Kaggle
- The audio tracks of the original dataset were pre-converted into mel-spectrograms

Transfer Learning

- Used the built in ViT_B_16 model from Pytorch
- The model was pretrained on the ImageNet dataset
- Replaced prediction head with 3 layer MLP with 10 output classes
- “Froze” the pretrained gradients of the ViT body, and only trained the MLP prediction head

Experiments and Analysis

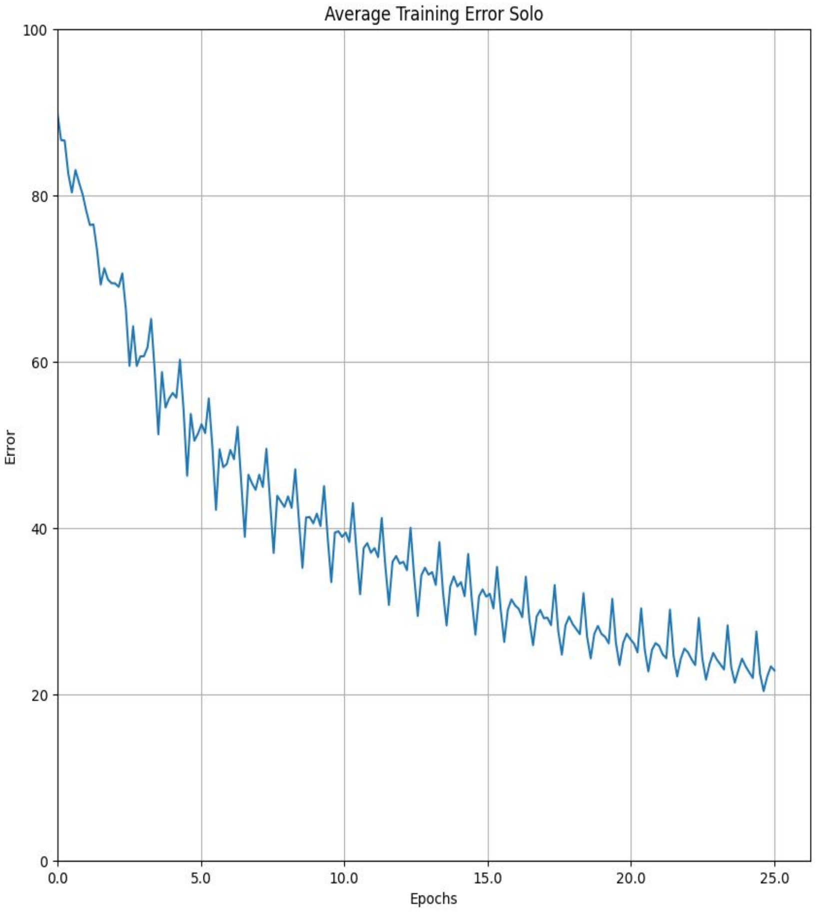
Data Pre-Processing

- The conversion to mel-spectrograms had already been done.
- Center cropped and then resized the images
- Normalized the resized images following recommendations in Pytorch documentation
- Image flattening into patches was taken care of by the ViT model

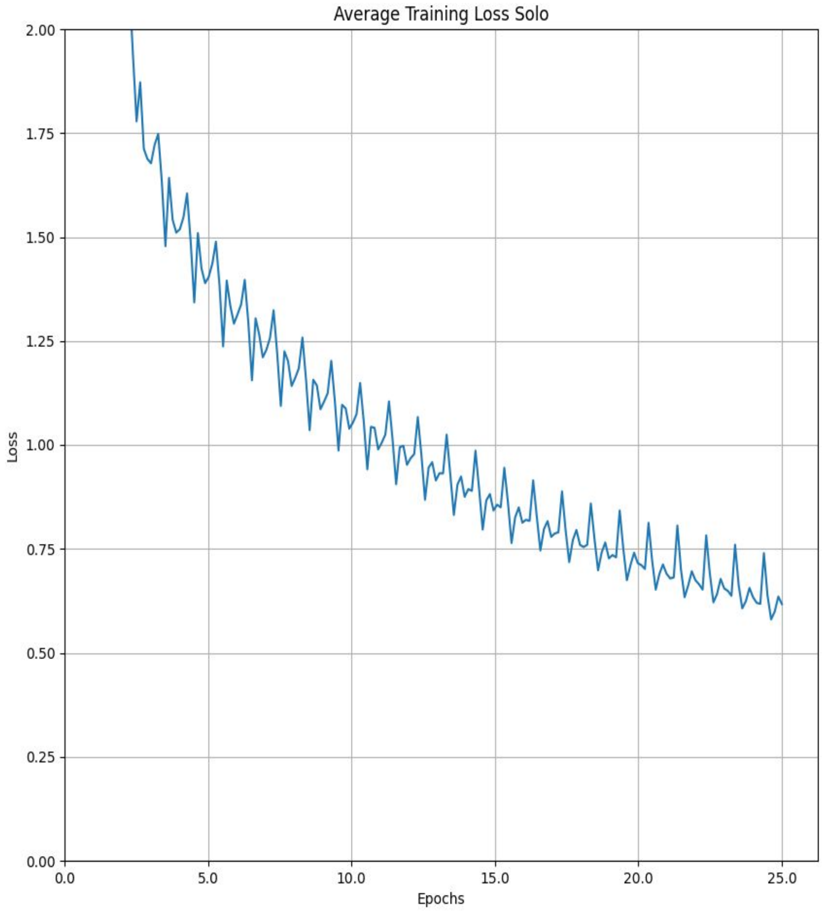
Hyperparameter Sweep

- All model training conducted using a NVIDIA RTX 3080ti GPU.
- Performed two sweeps, using the Learning Rate and Weight Decay as the hyperparameters:
  - 1st sweep trained 506 different models with 25 training epochs each
  - 2nd sweep trained 236 different models with 15 training epochs each
- First sweep took ~8.5 hours, the second sweep took ~2.5 hours.
- Saved training/validation losses and accuracies to both text files and Weights and Biases

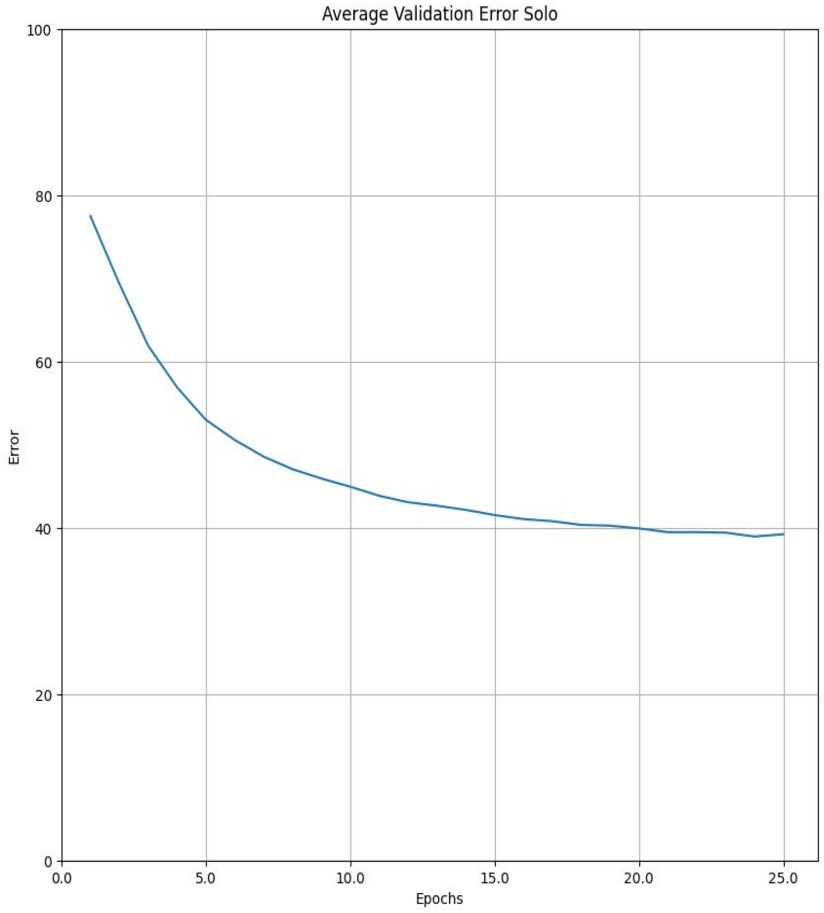
Results



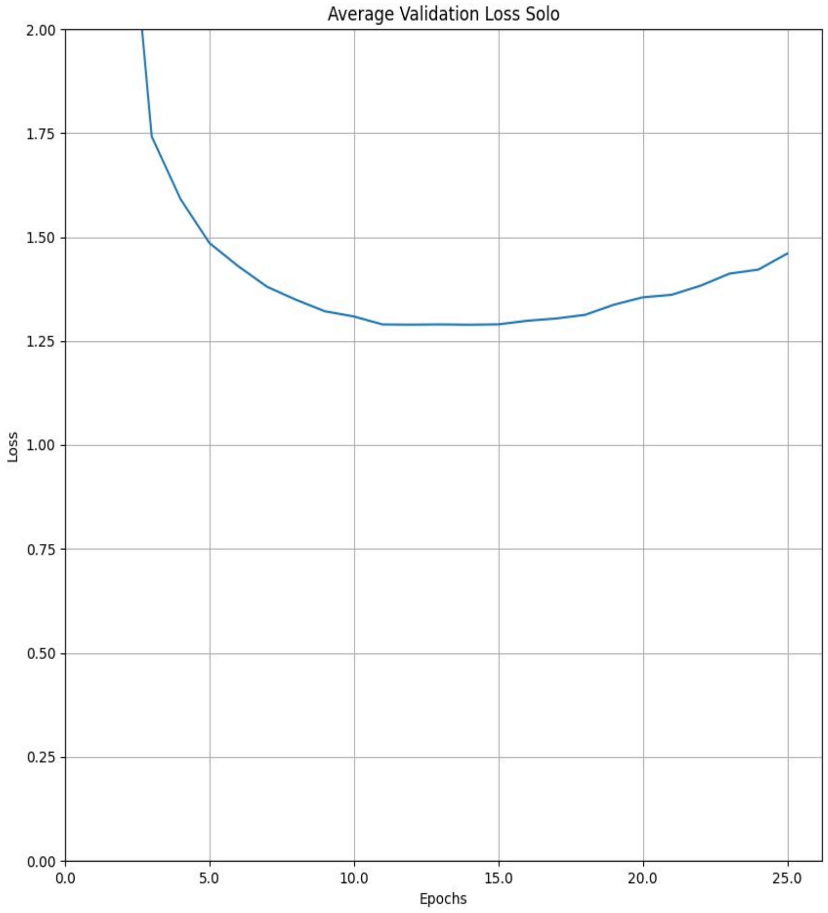
Average Training Error Solo
Error
Epochs
0
20
40
60
80
100
0.0
5.0
10.0
15.0
20.0
25.0



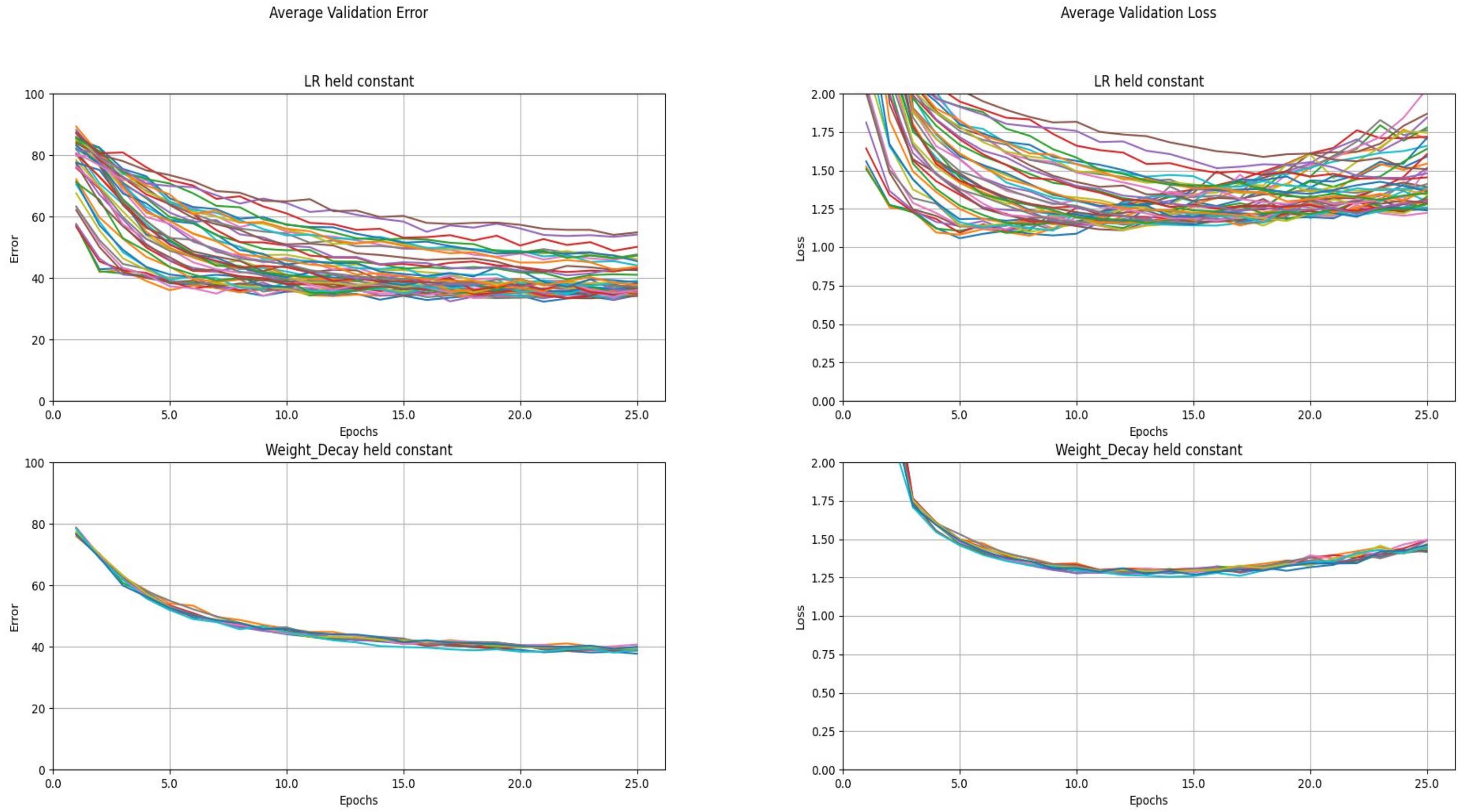
Average Training Loss Solo
Loss
Epochs
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
0.0
5.0
10.0
15.0
20.0
25.0

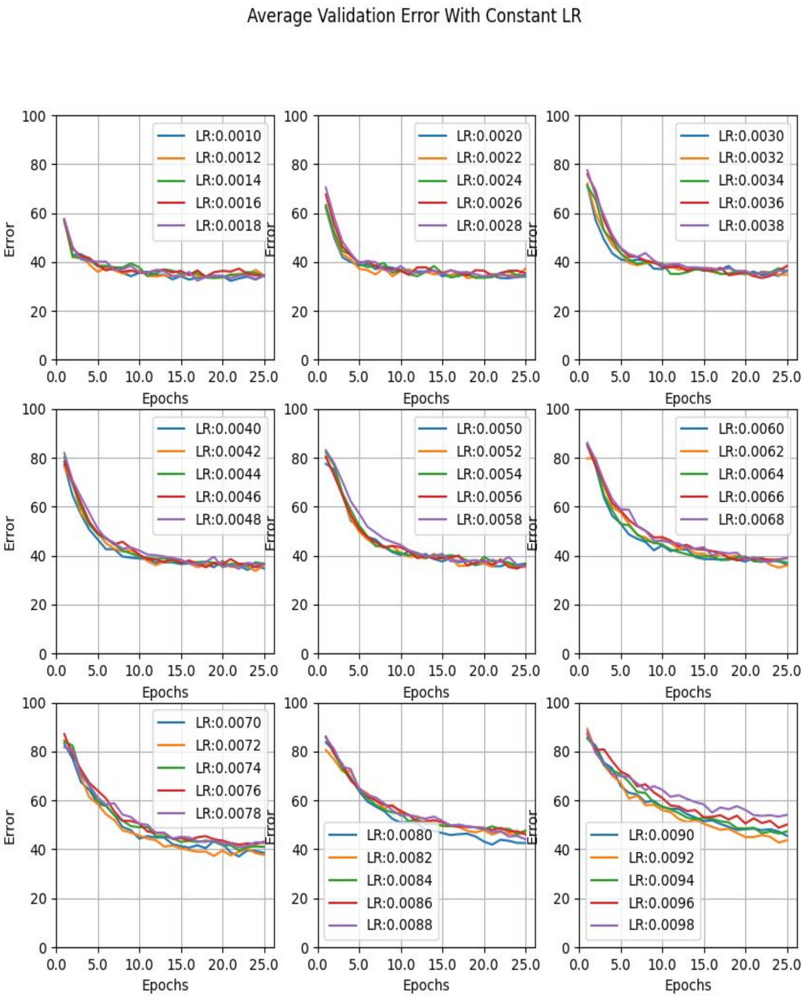


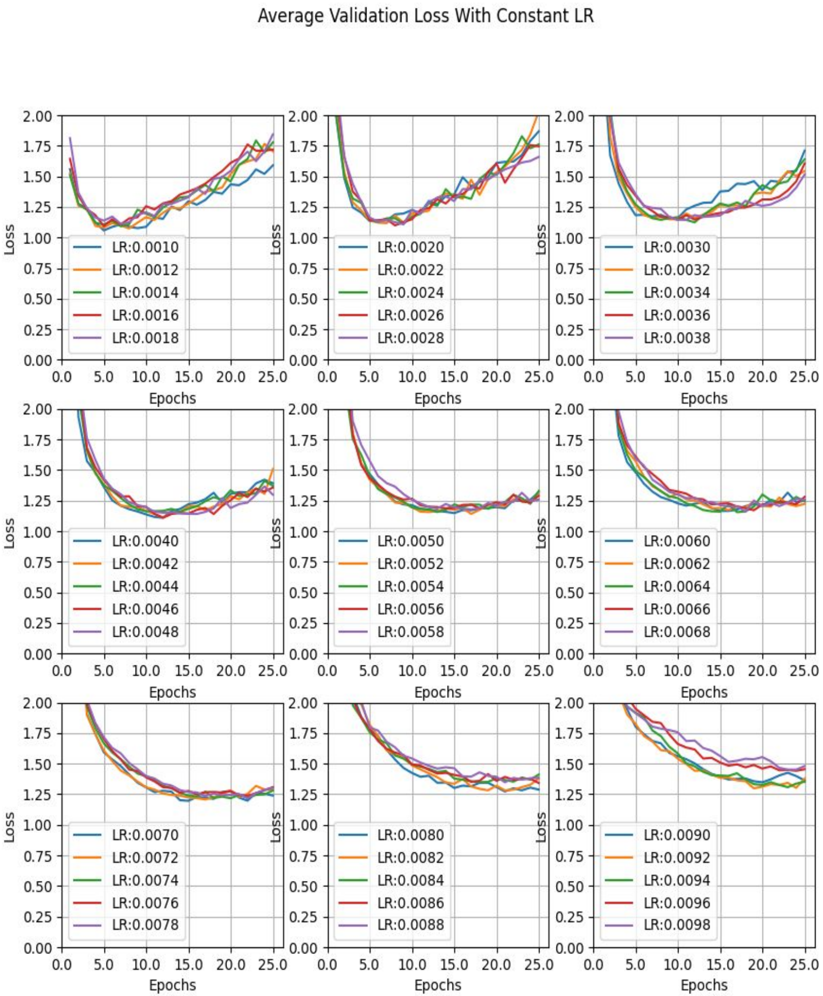
Average Validation Error Solo
Error
Epochs
100
80
60
40
20
0
0.0
5.0
10.0
15.0
20.0
25.0

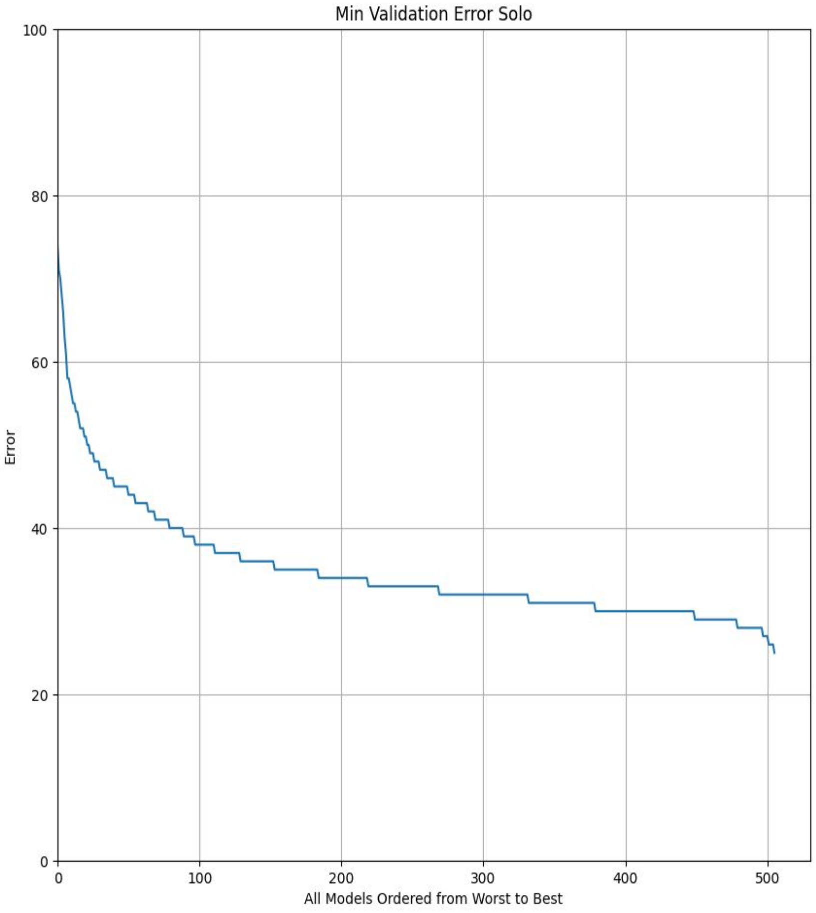


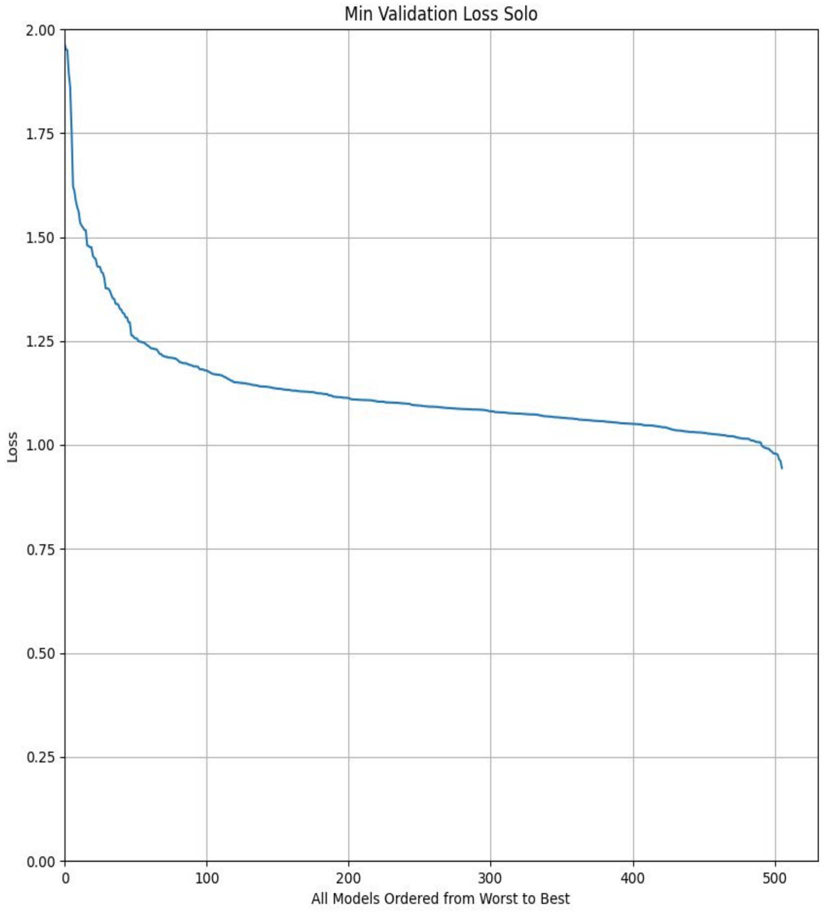
Average Validation Loss Solo
Loss
Epochs
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0.0
5.0
10.0
15.0
20.0
25.0


Average Validation Error
Average Validation Loss
LR held constant
Error
Epochs
Loss
Weight_Decay held constant
0.0
5.0
10.0
15.0
20.0
25.0
0
20
40
60
80
100
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00


Average Validation Error With Constant LR
Error
Epochs
LR:0.0010
LR:0.0012
LR:0.0014
LR:0.0016
LR:0.0018
LR:0.0020
LR:0.0022
LR:0.0024
LR:0.0026
LR:0.0028
LR:0.0030
LR:0.0032
LR:0.0034
LR:0.0036
LR:0.0038
LR:0.0040
LR:0.0042
LR:0.0044
LR:0.0046
LR:0.0048
LR:0.0050
LR:0.0052
LR:0.0054
LR:0.0056
LR:0.0058
LR:0.0060
LR:0.0062
LR:0.0064
LR:0.0066
LR:0.0068
LR:0.0070
LR:0.0072
LR:0.0074
LR:0.0076
LR:0.0078
LR:0.0080
LR:0.0082
LR:0.0084
LR:0.0086
LR:0.0088
LR:0.0090
LR:0.0092
LR:0.0094
LR:0.0096
LR:0.0098


Average Validation Loss With Constant LR
Loss
Epochs
LR:0.0010
LR:0.0012
LR:0.0014
LR:0.0016
LR:0.0018
LR:0.0020
LR:0.0022
LR:0.0024
LR:0.0026
LR:0.0028
LR:0.0030
LR:0.0032
LR:0.0034
LR:0.0036
LR:0.0038
LR:0.0040
LR:0.0042
LR:0.0044
LR:0.0046
LR:0.0048
LR:0.0050
LR:0.0052
LR:0.0054
LR:0.0056
LR:0.0058
LR:0.0060
LR:0.0062
LR:0.0064
LR:0.0066
LR:0.0068
LR:0.0070
LR:0.0072
LR:0.0074
LR:0.0076
LR:0.0078
LR:0.0080
LR:0.0082
LR:0.0084
LR:0.0086
LR:0.0088
LR:0.0090
LR:0.0092
LR:0.0094
LR:0.0096
LR:0.0098


Min Validation Error Solo
Error
All Models Ordered from Worst to Best
0
20
40
60
80
100
0
100
200
300
400
500


Min Validation Loss Solo
Loss
All Models Ordered from Worst to Best
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
0
100
200
300
400
500

Conclusion

Best Model

Sweep 1	Sweep 2
LR:0.0026, WD:0.05	LR: 0.0016, WD:0.012
Error: 37.37%	Error: 34.34%
Loss: 1.86	Loss: 1.13

Limitations and Ideas for Future Work

- Lack of available data and compute
  - In lieu of finding a better dataset, we could create our own using the YT-DLP tool
- If we had the time and compute, we could employ the M2D methodologies of Niizumi et al to train on the target dataset
- Could explore how the distribution of data affects the accuracy

Repo and Links

- GitHub repo found at:
  https://github.com/nick-rommel/CSC561-Final-Project/
- Weights and Biases Project link:
  https://wandb.ai/foxx-skulk/CSC561-Final-Project/overview