Using Pre-Trained Deep Learning Models to Identify Underlying Diseases from Skin Lesion Image Data

Joshua Uhlfelder
GitHub Repo Link
https://github.com/JoshuaUhlfelder/project1-ACV

1. Introduction

Motivation

A 1998 study showed that dermatologists with >10 years of experience diagnosed moles and other skin lesions with an accuracy of 80%.[1] With the increase in the prevalence of the application of neural networks in a clinical setting, melanoma identification seems to be an obvious application of deep learning. Indeed, a 2022 study from Alwakid et. al. predicted skin diseases with a CNN-based model with an accuracy of 86%, significantly higher than was achievable through clinical diagnosis.[2] The recent development of transformers and more sophisticated deep learning models creates a potential opportunity to build even more accurate skin lesion identification models. Indeed, a 2021 study deployed a Masked Attention Transformer (MAT) model on the HAM10000 dataset to get an accuracy of 92.55% on test data.[3] The literature shows that there is high potential to apply deep-learning models to medical diagnoses.

Project Goal

The goal of this project is to develop a model that identifies the underlying disease (melanoma, dermatosisroma, carcinoma, etc.) from images of skin lesions. Ideally, the model will achieve an accuracy above 86% on test data (more accurate than Alwakid et. al.). To attain this goal, three different types of pre-trained neural network models were fine-tuned and applied to skin lesion image data from the HAM10000 dataset. The best-performing models on validation data were selected and evaluated with test data. The breadth of this study will allow me, as a deep-learning novice, to interact with a multitude of different models to better understand how neural networks operate on large datasets.

---

[1] Morton, C A, and R M Mackie. “Clinical accuracy of the diagnosis of cutaneous malignant melanoma.” *The British journal of dermatology* vol. 138,2 (1998): 283-7. doi:10.1046/j.1365-2133.1998.02075.x

[2] Alwakid, Ghadah, et al. “Melanoma Detection Using Deep Learning-Based Classifications.” *Healthcare*, vol. 10, no. 12, Dec. 2022, p. 2481. *Crossref*, https://doi.org/10.3390/healthcare10122481.

[3] L. Zhou and Y. Luo, "Deep Features Fusion with Mutual Attention Transformer for Skin Lesion Diagnosis," *2021 IEEE International Conference on Image Processing (ICIP)*, Anchorage, AK, USA, 2021, pp. 3797-3801, doi: 10.1109/ICIP42928.2021.9506211.

2. Overview

Task Summary

- A total of three unique neural network models were fine-tuned and trained. Each was modified to classify 7 different types of skin lesions (see "Data" section below):
  1. **CNN** - used as a baseline for the other two models. A variety of different ResNet and VGG models were trained and evaluated.
  2. **Vision Transformer (ViT)** - image data was tokenized and processed to classify skin lesion images.
  3. **Multimodal BiTransformer/BERT** - the traditional "caption" data was replaced by tokenized information regarding the patient to which the skin lesion belongs. This model, unlike the other two, combined demographic information with image data to classify skin lesions. Two other models (models 4 and 5) also used BERT classification but varied in terms of their classification and tokenization strategies.
- Each model was fine-tuned by adjusting pre-training, batch sizes, gradient updates, output layers, and other modifications to optimize validation error.

Data

Each model was trained, evaluated, and tested on 10,015 images from the HAM10000 dataset.[4] The data contains pictures of 7 different types of skin lesions (actinic keratoses, basal cell carcinoma, benign, dermatofibroma, melanoma, melanocytic nevi, and vascular lesions).

The dataset did not have a pre-determined split. Therefore, data was divided so that approximately 60% was used for training, 20% was used for validation, and 20% was set aside for testing. These numbers are imprecise, as the dataset contains multiple photos of the same skin lesion. Thus, the total lesions were divided by this split ratio, resulting in datasets of the following size:

- Training set: 6022 images
- Validation set: 2001 images
- Test set: 1992 images

[4] Tschandl, Philipp. The HAM10000 Dataset, a Large Collection of Multi-Source Dermatoscopic Images of Common Pigmented Skin Lesions. V4 ed., Harvard Dataverse, 2018, https://doi.org10.7910/DVN/DBW86T.

3. Experimentation

Data Preprocessing

I applied different image pre-processing procedures for the training and evaluation datasets. For each of the three models, the various transformations of the data remained mostly unchanged besides a few modifications marked in the appendix.

The HAM10000 dataset consists only of images that are 400x600 pixels. Each training image was randomly cropped and scaled with a range of 20% to 100% of the original image, resulting in an image of size 224x224 px. The scaling range occasionally varied depending on the model (see appendix). This scaling helped to prevent overfitting by introducing noise into the models. Then, training images were randomly flipped horizontally and vertically to create a more diverse training set. Each image was then normalized.

For the validation and test datasets, the images were resized to 256x384 px and then cropped, resulting in an image of size 224x224 px.

Model 1: CNN

Objective
The implementation of Convolutional Neural Networks on the HAM10000 dataset is well documented. However, I thought it might be interesting to explore the performance of different types of CNNs on the classification task. I trained and fine-tuned a pre-trained ResNet18, ResNet34, ResNet50, and VGG19 on the data.

Model Evaluation Strategy
For each type of CNN model (VGG, ResNet18, etc.), I applied a greedy method of analysis where I would isolate a parameter — learning rate, for instance — and optimize it using the model's accuracy. Then, I would optimize a different variable. Because many of the model's parameters are highly dependent, this strategy did not always result in the most accurate model.

Please see **Appendix A** for a complete list of the models and their respective parameters and output evaluations.

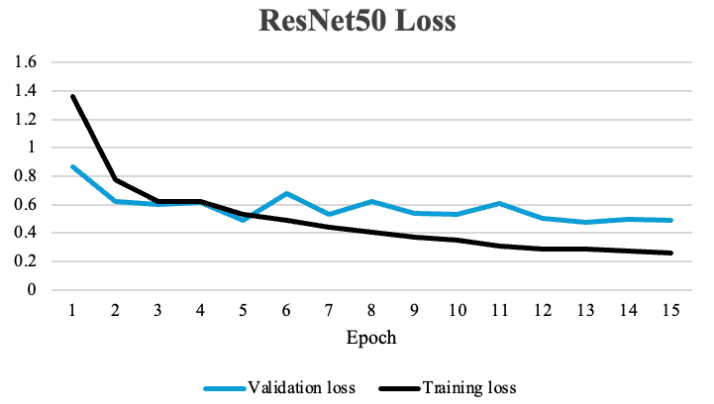
Final Model
The model that performed the best on validation data was a ResNet50 pre-trained on ImageNet.

Arguments

Model	ResNet50 pretrained on ImageNet-1k
Learning rate	0.00005

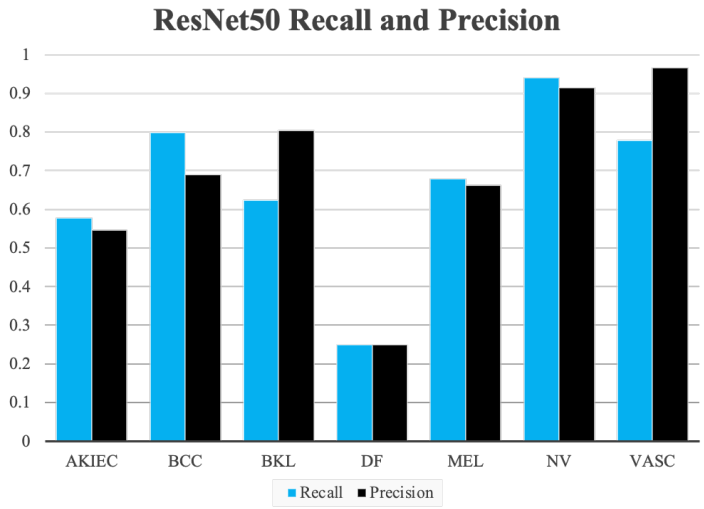
Schedule	Linear warmup for 5, exponential decay of 0.8 for 10
Epochs	15
Batch size	8

Validation and Training Loss



As the epochs proceed, the training loss continues to dip. However, the validation loss begins to settle after about the fifth epoch. This may be a sign of overfitting; however, the accuracy of the model continued to climb until the 13th epoch, reaching a maximum of 84.4%.

Precision and Recall on Test Data



Above, we see that the model had high recall and precision for the ‘nv’ class, which is ideal, as the data consisted primarily of this class. However, the model fails to reliably identify the ‘df’, ‘akiec, and ‘mel’ classes.

Model Summary

The ResNet50 pertained on ImageNet achieved a relatively high **accuracy of 84.6%** on test data. The model had 23,522,375 parameters. The model may be overfitting, as training loss continues

to descend far below validation loss as the model progresses. Given that Alwakid et. al. achieved an accuracy of 86% on a CNN, 84.6% is remarkable in comparison, especially given the difference in access to resources for training and evaluation. That said, the model has room for improvement in terms of achieving higher across-the-board precision and recall.

Model 2: Vision Transformer

Objective

Transformers are remarkably efficient and accurate while training with large quantities of data. Little scholarship has been applied to applying transformers (without additional demographic details) to skin lesion data. This exploratory analysis allowed me to do exactly that.

Model Evaluation Strategy

Much like for the CNN analysis, I generally isolated variables and optimized them in a greedy approach. Four different types of pre-trained transformers were tested: ViT, BEiT, DeiT, and MAE.

Please see **Appendix B** for a complete list of the models and their respective parameters and output evaluations.

Final Model

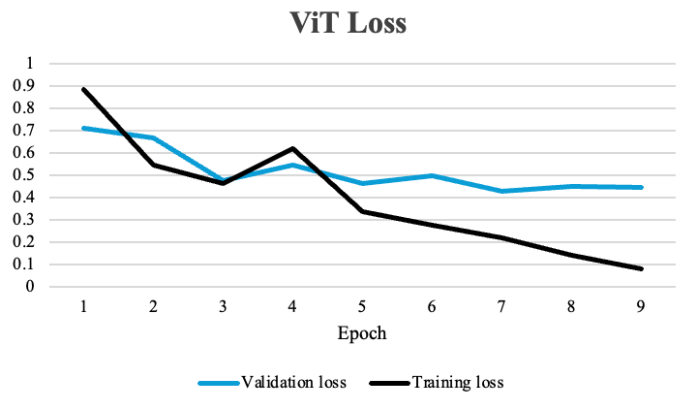
The model that performed the best on validation data was a standard Vision Transformer (ViT) that was pre-trained on ImageNet.

Arguments

Model	ViT pretrained on ImageNet-21k
Learning rate	0.00025
Schedule	Linear warmup for 5, cosine for 10
Epochs	9
Gradient updates	1
Batch size	64

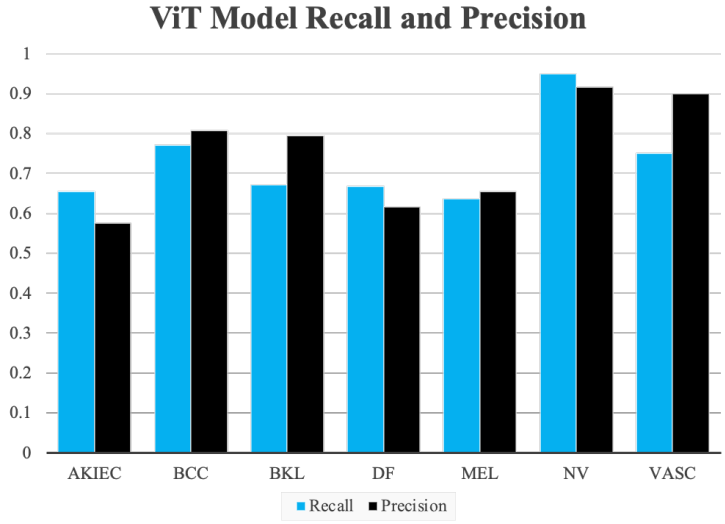
Valuation and Training Loss

As in Model 2, As the epochs proceed, the training loss continued to dip. The validation loss continues to descend until about the 7th epoch. Due to resource constraints, the model was not trained past epoch 9, so it is possible that with more epochs, this model could perform even better. The validation accuracy reached a maximum of 86.1% at epoch 8.



Precision and Recall on Test Data

Compared to the CNN model, the ViT performs extraordinarily better across the board for nearly all classes. There are still some classes where both metrics are low, but the model has no particular weaknesses.



Model Summary

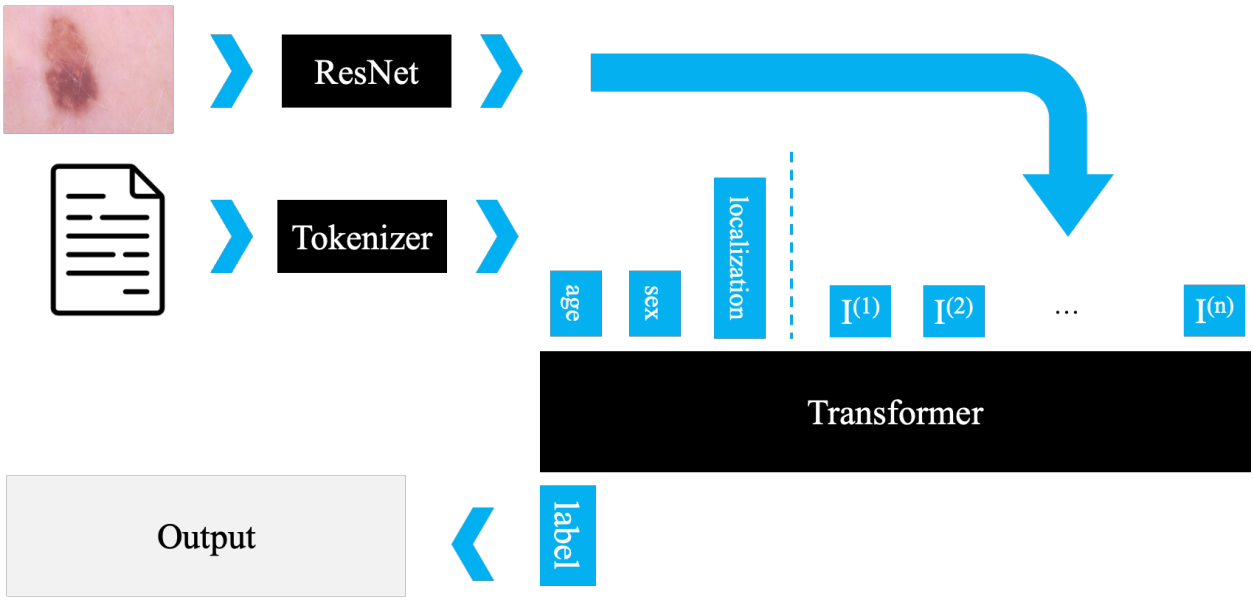
The ViT pertained on ImageNet got a relatively high **accuracy of 85.6%** on test data. This accuracy is very close to the one achieved by Alwakid et. al. Given more time and resources, the ViT model shows great promise for improvements in accuracy even beyond this benchmark. Indeed, the model shows adequate recall and precision across all classes, especially compared to Model 1.

Model 3: MultiModal BiTransformer (MMBT)

Objective

After reviewing literature, it seems that one of the most successful applications of neural networks on the HAM10000 dataset was the application of a MAT model that processed both demographic and localization information regarding the skin lesions and image data by Zhou et al.[5] Inspired by this, I wanted to incorporate this additional information into my model, so I turned to a BERT model that would tokenize the data and images and feed them into a transformer.

A diagram of the model is pictured below.



The model tokenizes the image with a ResNet50 (this was kept constant). Metadata was tokenized and fed into the transformer. Different tokenization strategies were used for different iterations (see below). Predictive labels were then derived.

Model Evaluation Strategy

Like for the other models, I attempted to isolate variables to maximize accuracy. This quickly became challenging once the evaluation accuracy, precision, and recall began to remain static despite any changes to the model arguments.

Please see **Appendix C** for a complete list of the models and their respective parameters and output evaluations.

---

[5] hL. Zhou and Y. Luo, "Deep Features Fusion with Mutual Attention Transformer for Skin Lesion Diagnosis," *2021 IEEE International Conference on Image Processing (ICIP)*, Anchorage, AK, USA, 2021, pp. 3797-3801, doi: 10.1109/ICIP42928.2021.9506211.

Final Model

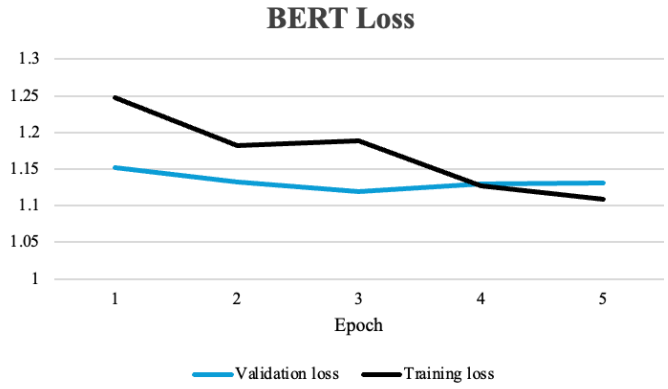
Nearly every potential model performed the same on validation data. However, the final model is a BERT with a ResNet50 image encoder and a custom tokenizer. The custom tokenizer maps the sex, age, and localization of the skin lesion to a tensor used in the forward process while training.

Arguments

Model	BERT with custom tokenizer and ResNet50 image encoder
Learning rate	0.00025
Schedule	Cosine
Epochs	5
Gradient updates	1
Batch size	64

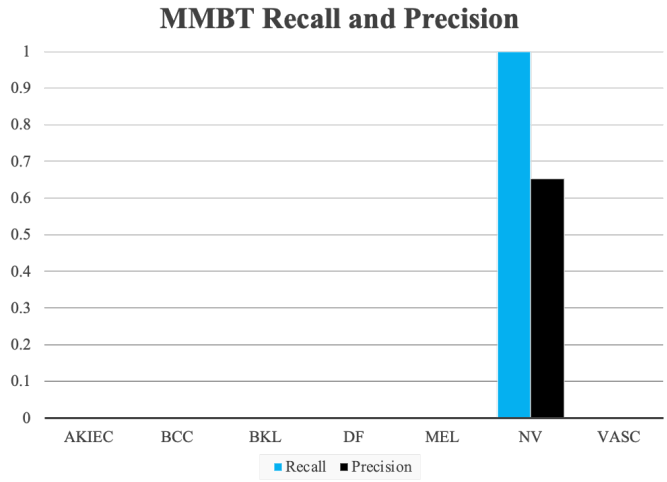
Validation and Training Loss

The model's training loss continues to diminish for every epoch. However, validation loss dips at epoch 3 while remaining relatively still. The accuracy of the model on validation data at every epoch remained at 67.2%.



Precision and Recall on Test Data

It is clear from the recall and precision data that this model generalizes poorly, as it labels every test case as 'nv'. Because 'nv' accounts for about two-thirds of all cases, the model is able to achieve an overall **accuracy of 65.3%**. This model is useless, however, as it only predicts one class.



Model 4

To try to relieve this single-class prediction issue, I generated a binary classifier that would identify if a skin lesion belongs to the class 'other' or to 'nv'. The model functions the same as Model 3, except there are only two classes. This model, too, achieved an **accuracy of 65.3% on test data**. It also revived the same precision and recall scores. It appeared that joining the classes made no difference.

Model 5

Assuming the issue was the custom tokenizer and not the out-of-the-box BERT model, I decided to join the demographic information as a string and feed it to the pre-trained BERT tokenizer. The only difference between this model and Model 3 was the tokenization strategy. This model had the same problem as the previous two.

Issue Diagnosis

After spending much time trying to identify the underlying problem, it seemed that occasionally, the model was able to break out of its 67.2% loop to reach an improved accuracy. When this happened, however, it quickly fell back. Although a variety of different learning rates and schedulers were used, it might be worth employing a dynamic scheduler that rapidly adjusts the learning rate in an attempt to break the model from its loop. It might also be worth applying a different loss equation that penalizes a skewed distribution to an even greater extent than cross-entropy.

Additional Test

With two functioning models (Model 1 and 2), and a model that delivers solely good news (Model 3), I decided to apply these models to some test data of my own. I ran an image of a mole on my toe through the model and received the diagnosis of 'nv', which is a benign, typical mole.

4. Conclusion

Summary

The most successful model I derived (Model 2: ViT) achieved an accuracy of 85.6% on test data with mostly high precision and recall scores across all classes. This accuracy is competitive against Alwakid et. al.'s accuracy of 86%. Given the time and resource constraints of this study, there is great potential to further optimize this vision transformer model by adjusting the loss function, implementing different optimizers, or modifying other learning parameters.

Model 1, the ResNet50, achieved a similarly high accuracy but failed to get as high recall and precision scores for all classes. Model 3, the modified MMBT BERT failed to generate any useful insights, as it appeared to fall into a broad local minimum. However, this model has the greatest potential for improvement by implementing new loss functions, learning rate schedulers, or preprocessing steps.

Overall, this brief study demonstrates the practicality of deep learning models in the medical space, as the models I trained can identify skin lesions with a 6% greater accuracy than experienced professionals.

Potential Further Analyses

Despite having some success developing competitive deep learning models on the HAM10000 dataset, there remains plenty of room for improvement and further study.

First, many attempts could be made to deter Model 3's tendency to converge to a local minimum. For instance, a learning rate scheduler that changes dynamically may prevent early convergence. Or, a different loss function could be applied that penalizes the labeling of only one class.

Across all models, timing limitations may have prevented convergence. The validation loss of many of the models tested continued to descend while the models reached their final scheduled epochs. Therefore, increasing the number of epochs might allow for more accurate models. However, this runs the risk of overfitting.

Another improvement for all models would be greater training data preprocessing. Adjusting the brightness, hue, shape, and filters of images would likely help prevent overfitting and allow training for a greater number of epochs. Some data modification was used to train these models, but this was rather limited.

For Model 1, Adam was selected as the optimizer for all trials. However, SGD might allow the accuracy of the model to supersede its current metric.
Additionally, k-fold cross-validation could be applied to the training of the models, as this would enable more accurate validation information. The HAM10000 dataset does not have pre-

determined training and evaluation sets, so this is an ideal application of cross-validation. Cross-validation was avoided due to the time implications.

4. Appendix

Note: all models use cross-entropy loss.

Appendix A: Model 1 Trials

Trial 1	
Model	ResNet18 pretrained on ImageNet-1k
Batch size	4
Epochs	10
Optimizer	Adam
Learning rate	0.00001
Scheduler	Every 7 epochs, multiply lr by 0.1
Validation accuracy	0.837851
Training time	13m 3s
Trial 2	
Model	ResNet34 pretrained on ImageNet-1k
Batch size	8
Epochs	5
Optimizer	Adam
Learning rate	3E-04
Scheduler	Every 3 epochs, multiply lr by 0.2
Notes	All weights frozen except classifier
Validation accuracy	0.744478
Training time	6m 28s
Trial 3	
Model	ResNet18 pretrained on ImageNet-1k
Batch size	8
Epochs	5
Optimizer	Adam
Learning rate	3E-04
Scheduler	Every 2 epochs, multiply lr by 0.2

Validation accuracy	0.8052
Training time	7m 28s
Trial 4	
Model	ResNet50 pretrained on ImageNet-1k
Batch size	8
Epochs	5
Optimizer	Adam
Learning rate	3E-03
Scheduler	Every 2 epochs, multiply lr by 0.1
Validation accuracy	0.684237
Training time	8m 215s
Trial 5	
Model	ResNet34 pretrained on ImageNet-1k
Batch size	4
Epochs	7
Optimizer	Adam
Learning rate	0.002
Scheduler	Linear warmup of 3, exponential decay of 0.5
Validation accuracy	0.75
Training time	NA
Trial 6	
Model	ResNet34 pretrained on ImageNet-1k
Batch size	4
Epochs	7
Optimizer	Adam
Learning rate	0.0001
Scheduler	Linear warmup of 4, exponential decay of 0.5
Validation accuracy	0.832329
Training time	12m 54s

Trial 7	
Model	ResNet50 pretrained on ImageNet-1k
Batch size	4
Epochs	7
Optimizer	Adam
Learning rate	0.0001
Scheduler	Linear warmup of 4, exponential decay of 0.5
Validation accuracy	0.8404
Training time	11m 26s
Trial 8	Chosen model
Model	ResNet50 pretrained on ImageNet-1k
Batch size	8
Epochs	15
Optimizer	Adam
Learning rate	0.00005
Scheduler	Linear warmup of 5, exponential decay of 0.8
Validation accuracy	0.846386
Training time	25m 56s
Trial 9	
Model	ResNet50 pretrained on ImageNet-1k
Batch size	8
Epochs	7
Optimizer	Adam
Learning rate	0.00005
Scheduler	Linear warmup of 5, exponential decay of 0.8
Notes	Images not normalized
Validation accuracy	0.8253
Training time	12m 29s
Trial 10	

Model	VGG19 pretrained on ImageNet-1k
Batch size	8
Epochs	15
Optimizer	Adam
Learning rate	0.00005
Scheduler	Linear warmup of 5, exponential decay of 0.8
Validation accuracy	0.78212
Training time	25m 7s

Appendix B: Model 2 Trials

Trial 1	
Model	ViT pretrained on ImageNet-21k
Batch size	8
Epochs	5
Optimizer	AdamW
Learning rate	1E-03
Scheduler	cosine
Gradient updates	4
Notes	Weights fixes except classifier
Validation loss	0.736172199249268
Validation accuracy	0.729919678714859
Training time	7m 55s
Trial 2	
Model	ViT pretrained on ImageNet-21k
Batch size	16
Epochs	5
Optimizer	AdamW
Learning rate	1E-03
Scheduler	cosine
Gradient updates	8
Validation loss	0.489103704690933
Validation accuracy	0.827309236947791
Training time	10m 56s
Trial 3	
Model	MAE pretrained on ‘facebook/vit-mae-large'
Batch size	16
Epochs	5
Optimizer	AdamW

Learning rate	1E-03
Scheduler	cosine
Gradient updates	8
Validation loss	0.713780462741852
Validation accuracy	0.725401606425703
Training time	23m 50s
Trial 4	
Model	BEiT pretrained on ‘microsoft/beit-base-patch16-224-pt22k’
Batch size	16
Epochs	5
Optimizer	AdamW
Learning rate	1E-03
Scheduler	cosine
Gradient updates	8
Validation loss	0.859493911266327
Validation accuracy	0.686244979919679
Training time	11m 02s
Trial 5	
Model	DeiT pretrained on ‘facebook/deit-base-patch16-224’
Batch size	16
Epochs	5
Optimizer	AdamW
Learning rate	1E-03
Scheduler	cosine
Gradient updates	8
Validation loss	0.566697061061859
Validation accuracy	0.791666666666667
Training time	11m 3s
Trial 6	

Model	ViT pretrained on ImageNet-21k
Batch size	64
Epochs	9
Optimizer	AdamW
Learning rate	1E-03
Scheduler	cosine, with 3-step warm up
Gradient updates	1
Validation loss	0.511732757091522
Validation accuracy	0.841867469879518
Training time	19m 33s
Trial 7	Chosen model
Model	ViT pretrained on ImageNet-21k
Batch size	64
Epochs	9
Optimizer	AdamW
Learning rate	1E-03
Scheduler	cosine
Gradient updates	1
Validation loss	0.592900395393372
Validation accuracy	0.849397590361446
Training time	19m 33s

Appendix C: Model 3 Trials

Trial 1	
Model	BERT with ResNet50 and custom tokenizer
Batch size	64
Epochs	3
Optimizer	AdamW
Learning rate	1E-04
Scheduler	cosine
Gradient updates	1
Validation loss	0.627010703086853
Validation accuracy	0.67218875502008
Training time	9m 21s
Trial 2	
Model	BERT with ResNet50 and custom tokenizer
Batch size	64
Epochs	3
Optimizer	AdamW
Learning rate	1E-02
Scheduler	cosine
Gradient updates	1
Validation loss	1.0984
Validation accuracy	0.67218875502008
Training time	10m 6s
Trial 3	
Model	BERT with ResNet50 and custom tokenizer
Batch size	64
Epochs	2
Optimizer	AdamW
Learning rate	1

Scheduler	cosine
Gradient updates	8
Validation loss	5.87114238739014
Validation accuracy	0.00953815261044177
Training time	8m 41s
Trial 4	
Model	BERT with ResNet50 and custom tokenizer
Batch size	64
Epochs	3
Optimizer	AdamW
Learning rate	1E-03
Scheduler	cosine
Gradient updates	1
Validation loss	1.11880075931549
Validation accuracy	0.67218875502008
Training time	9m 42s
Trial 5	Final Model - but metrics not replicable
Model	BERT with ResNet50 and custom tokenizer
Batch size	64
Epochs	5
Optimizer	AdamW
Learning rate	1E-03
Scheduler	cosine
Gradient updates	1
Notes	Model showed best accuracy but was not replicable due to random instantiation
Validation loss	0.848152101039887
Validation accuracy	0.718373493975904
Training time	12m 10s
Trial 6	

Model	BERT with ResNet50 and custom tokenizer
Batch size	64
Epochs	10
Optimizer	AdamW
Learning rate	2E-03
Scheduler	cosine
Gradient updates	1
Validation loss	1.12663996219635
Validation accuracy	0.67218875502008
Training time	20m 58s