Emotion Classification using Various Transformer Architectures

Daniel Pak
UNI: dcp2149
Data Mining
IEORE4540

May 4th 2022

Work Responsibility

I worked by myself, so I did everything from the programming (preprocessing data, creating machine learning models), and writing the report.

1 Introduction

Humans are able to tell emotions from a variety of different media like books, songs, or television. It is something humans learn by interacting with other people and their environment. However, attempting to translate to a computer is a task that requires the computer to learn what aspects are important to predict what emotion is occurring. To try to develop this emotion artificial intelligence, labeled data is necessary to learn various emotions. With a sufficient dataset, various machine learning models can be used to generalize and predict what emotions are being elicited by a human. An audio dataset is used for this kind of sentiment analysis, by testing and benchmarking with a variety of transformer architectures. Transformers use attention mechanisms to weigh the significance of an input sequence, and extract the meaningful portions that can correctly predict what emotion is occurring [1].

2 Exploratory Data Analysis

The dataset used is Variably Intense Vocalizations of Affect and Emotion (VIVAE) [2]. It contains 1085 unique audio recordings of human non-speech emotion vocalization, where there are three positive emotions (achievement, pleasure, and surprise) and three negative emotions (anger, fear, and pain). The audio recordings were measured from 11 different subjects at various intensities, which ranged from low, moderate, peak, and strong. The audio recordings were all recorded at 44100 Hz at 16-bit resolution and monaural (single channel).

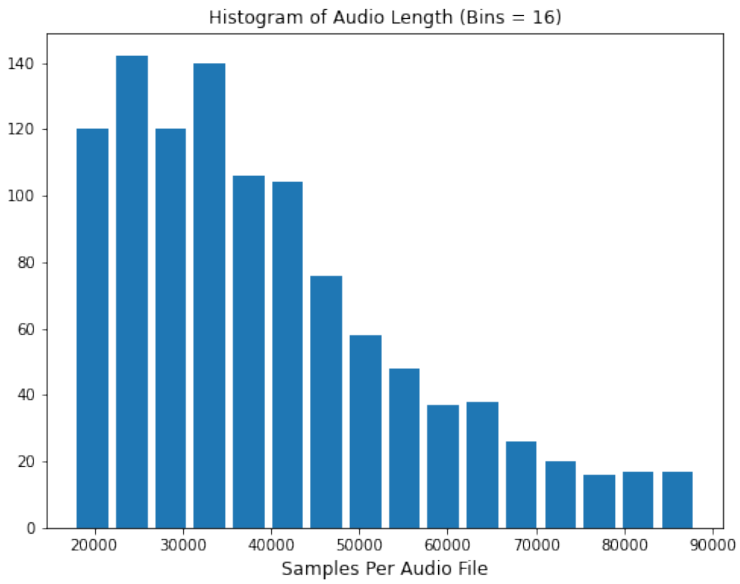


Figure 1: Distribution of Audio Lengths in Samples

The audio recordings were not uniform in length, with the shortest being 0.3995 seconds (17619 samples), the longest being 2 seconds (88200 samples), the average being 0.9029 seconds (39819 samples), and the median being 0.8102 seconds (35732 samples). The histogram shown in Figure 1 followed a right skewed distribution.

3 Processing Audio into Datasets

3.1 Features

The features to be used in different machine learning architectures can be represented by a 1-D representation using the time series of the recordings as a sequence, or as a 2-D representation using a Mel spectrogram as an image which contains both frequency and time series information. A Mel spectrogram differs from a regular spectrogram as it models human hearing perception using the Mel scale, which is approximately logarithmic, whereas a regular spectrogram places equal importance on all frequencies contained by the audio signal.

The time series representation is only preprocessed to be scaled from $[-1, 1]$ compared to the maximum and minimum value from the whole dataset, and then padded with 0 or spliced accordingly to be within a predetermined max length. This max length was chosen to be 1 second, so a sampling rate of 44100 Hz corresponds to 44100 samples. Incorporating the batch dimension n, the dimension of my 1-D representation dataset is $[n, 44100]$. An example follows in Figure 2.

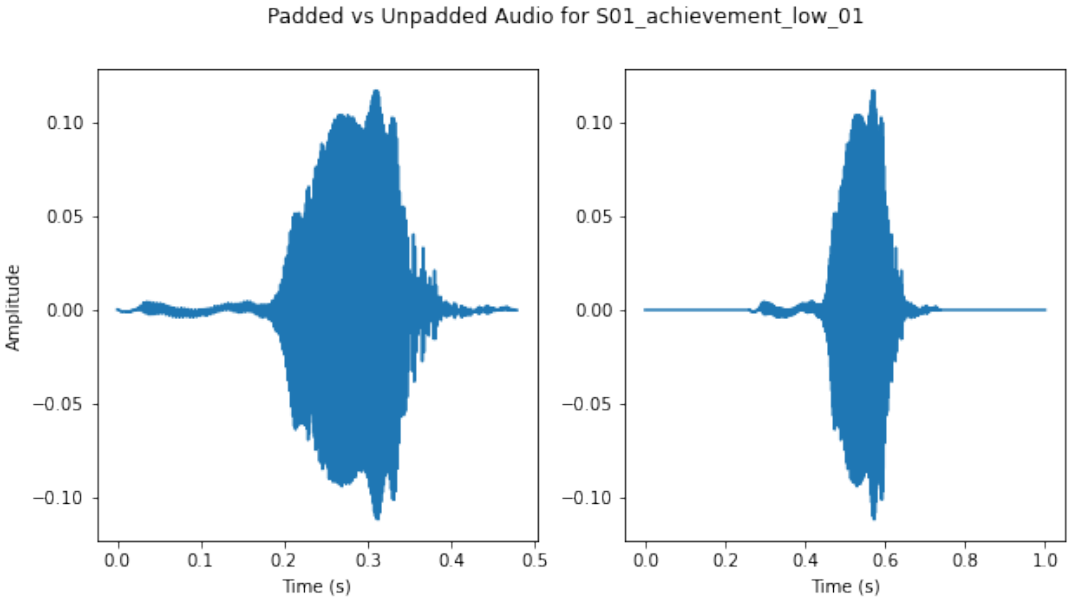


Figure 2: Padded Time Series

Before an audio signal is transformed into a Mel spectrogram, it is padded or spliced accordingly as outlined by the process from the 1-D representation. Compared to the 1-D representation for the features, the 2-D representation has multiple hyperparameters. The ones tuned for are the window length, hop length, the size of the fast fourier transfer (FFT) bins, and the number of mel filterbanks. These parameters allow for the short time fourier transform (STFT) of the audio signal to be taken and then transformed into the Mel spectrogram.

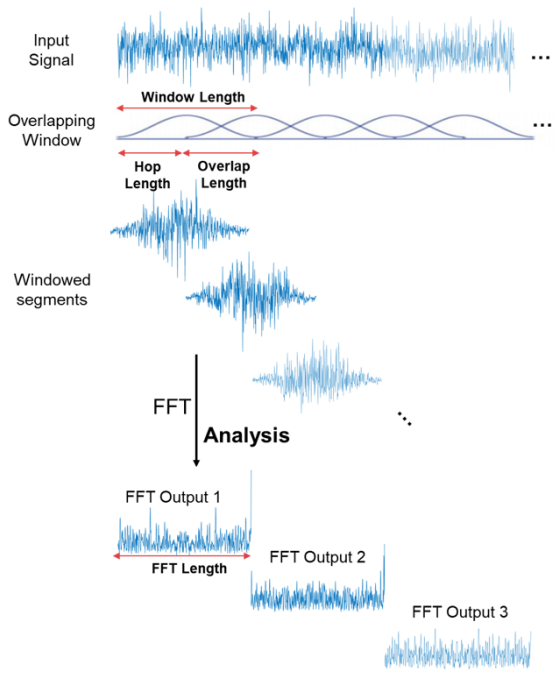


Figure 3: STFT Diagram [3]

In Figure 3, the window length and hop length dictates what part of the signal is taken to perform the STFT. Once these two parameters are set, the overlap length can be calculated. The FFT bins determines possible frequency resolution based off the sampling rate and Nyquist Theorem. Mel filterbanks are the frequency bins on the Mel scale to give better resolution for lower frequencies, and worse resolution at higher frequencies. Once these parameters are chosen to create the Mel spectrogram, it is then convert from an amplitude scale to a decibel scale, relative to a chosen top decibel (dB) value. The chosen parameters are as follows.

Hop Length	64
Window Length	1024
n_fft	1024
n_mels	64
Top dB	80

Table 1: Chosen Parameters to Generate Mel Spectrogram

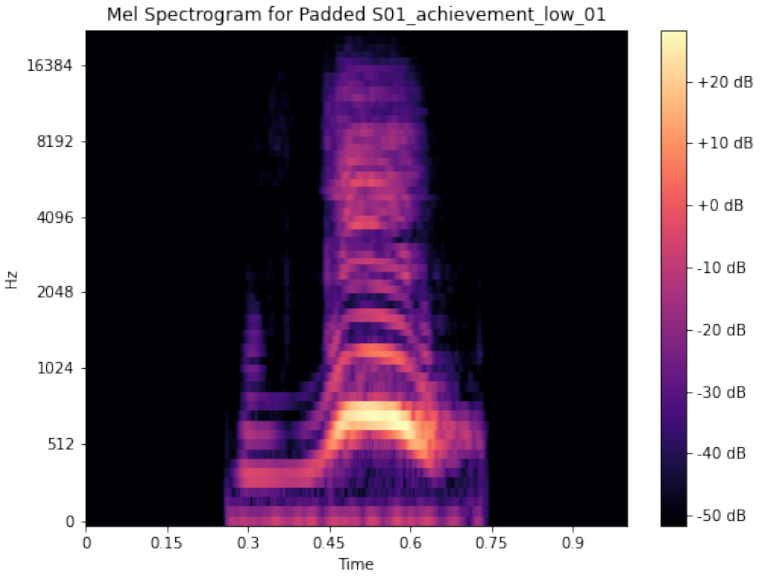


Figure 4: Padded Mel Spectrogram

In Figure 4, the heatmap shows that the majority of the signal lies between 0.25 seconds and 0.70 seconds, as this was the original signal before it was padded. The padded portions had an amplitude of 0, so it corresponds to the lowest decibel of -50 dB.

The dimensions of the mel spectrogram is $[n_{mels}, \lfloor sr/hoplength \rfloor]$ where sr is the sampling rate. Plugging in values, the dimensions of the spectrograms to construct the dataset are $[64, 690]$. However, to be processed as an image, channel information is also required. As the recordings are all monaural, this is just 1. Incorporating the batch dimension n, the dimension of my 2-D representation dataset is $[n, 1, 64, 690]$.

3.2 Labels

The chosen class label to analyze was just the emotions for a total of 6 possible class labels, as the emotions and intensities would create 24 unique classes, with an average of 45 recordings in each class. These labels would be encoded into numerical values ranging from 0 to 5 and kept in a master dictionary to associate the output of a model to the specific emotion. Figure 5 shows that the distribution of the emotion classes are balanced to not require any data augmentation like SMOTE. From this, using accuracy and area under curve (AUC) would be valid metrics to determine how well the various machine learning models are performing on a test set.

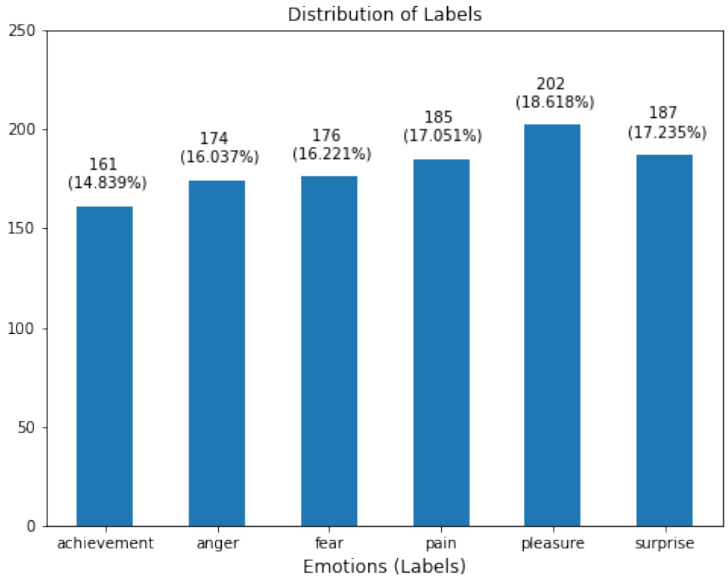


Figure 5: Distribution of Emotion Classes

4 Model Architectures

Various machine learning models were trained and tested using an 80-20 training-testing split of the two proposed datasets, depending on what kind of input the models would take. The training set would have 868 recordings while the testing set would have 217 recordings.

4.1 Logistic Regression

Logistic regression was used as a simple baseline model. This was chosen as it was expected that the transformer architectures would outperform logistic regression, as logistic regression has no capacity to use sequential data. It used the 1-D representation as an input and predicted an output of probabilities for the 6 labels.

4.2 Multi Layer Perceptron (MLP)

A vanilla MLP neural network was also used as a second baseline model as it is more lightweight and computationally efficient compared to the other various transformer architectures used, as it is matrix multiplication and vector addition under the hood. It used the 1-D representation as an input and predicted an output of probabilities for the 6 labels. The architecture is shown in Table 2.

Layers
Linear(44100,100)
Dropout(0.25)
Linear(100,200)
Dropout(0.25)
Linear(200,60)
Linear(60,60)

Table 2: Architecture for MLP Model

This model has 4.4 million trainable parameters. As this problem was a multi-label classification problem, the MLP network used cross entropy as the loss function. The optimizer was the Adam optimizer with a constant learning rate of 2e-5 with a L2 penalty of 1e-6. The activation function for each output of the Linear layers was ReLu. The batch size was 30 for both training and testing.

4.3 Vision Transformer (ViT)

ViTs were originally developed for image classification tasks. Transformers were built to be permutation invariant, so an image represented by a grid has to be encoded as a sequence. ViT does this by processing the original image into image patches and then flattens them to be used as the sequence tokens [4]. These flattened patches are then projected into a lower dimensional linear embedding using a linear transformation layer. There is no fixed position embedding, but instead a learned position embedding. This sequence is then used as the input to an encoder. Compared to a regular transformer, ViT does not need to use a decoder, but has a classification head to predict the labels.

Using the ViT architecture, 4 different models were trained using the VIVAE dataset.

Name	Image Size (B,C,H,W)	Patch Size	Patches	Pretrain	Fine-Tune	Trainable Parameters
ViT 32,30	[n, 1, 64, 690]	(32,30)	46	No	No	4.9 million
ViT 16,69	[n, 1, 64, 690]	(16,69)	40	No	No	4.9 million
ViT 8,23	[n, 1, 64, 690]	(8,23)	240	No	No	4.8 million
ViT-Base	[n, 3, 224, 224]	(16,16)	196	ImageNet-21k	ImageNet 2012	8.6 million

Table 3: Models Created with ViT Architecture

In Table 3, the first 3 models were trained from the ground up, with no pretraining or fine tuning done on separate datasets. The model hyperparameters are as follows.

Parameter	Description	Value
Dim	Linear Projection Dimension	200
Depth	Number of Transformer Blocks	5
Heads	Number of heads in Multi-head Attention Layer	16
MLP Dim	Dimension of MLP Layer	300
Dropout	Dropout Rate	0.05
Embedding Dropout	Embedding Dropout Rate	0.05

Table 4: Hyperparameters for ViT 32,30, ViT 16,69, and ViT 8,23

The last model, ViT-Base, was the same exact one that the original researchers for ViT proposed [4]. ViT-Base was already pretrained using ImageNet-21k, a dataset which contains 14 million

images with 21,843 classes, and then fine tuned on ImageNet 2012, a dataset which contains 1 million images and 1,000 classes. Knowing the features and labels, it is clear that ViT-Base is a model formed from supervised learning. Using this model is a form of transfer learning. The idea behind transfer learning is that if the original model is trained on a large and general dataset, it can serve as a generic model to perform well on other datasets by extracting features useful for these downstream tasks. Modifications had to be made on the Mel spectrograms to use ViT-Base [5]. The spectrograms were converted into an RGB image by repeating the original single channel image three times. The RGB images were then resized so that either the height or width would be of size 248. The resulting image would then be cropped from the center into a square image of resolution 224 by 224. Finally, the image would then be normalized to have 0.5 mean and 0.5 standard deviation for all 3 channels.

All of the ViT models used cross-entropy as the loss function. The optimizer was the Adam optimizer with a decaying learning rate of 5e-6 with a L2 penalty of 1e-6. The learning rate would decay every 50 epochs by 0.98. The batch size was 30 for both training and testing.

4.4 wav2vec2

The final transformer model used on VIVAE is wav2vec2, a framework which extracts new types of input vectors for acoustic models from raw audio, using pre-training and self-supervised learning [6]. Like ViT-Base, wav2vec2 is pretrained for audio tasks instead of image classification. However, wav2vec2 uses self-supervised learning, where the label is generated by itself from the input data. wav2vec2 was pretrained on Librispeech, a corpus of approximately 1000 hours of read English speech, which was unlabeled and then fine tuned on 960 hours of labeled data from Librispeech [7].

The original researchers added a classification head to the transformer architecture. For Librispeech, it was 29 tokens for the character targets. This can be modified to fine tune wav2vec2 to be used for a variety of downstream tasks, like using VIVAE for sentiment classification. Changing the classification head results in having no pretrained weights, so fine tuning is required for future inference.

wav2vec2 was trained with 16 kHz audio data, so audio data from VIVAE had to be resampled. Again, keeping the same max length of 1 seconds, this is reducing the number of samples from 44100 to 16000, while trying to retain the majority of the amplitude information. This was done with the same feature extractor as proposed by the original researchers. Figure 6 shows an example of resampled padded audio.

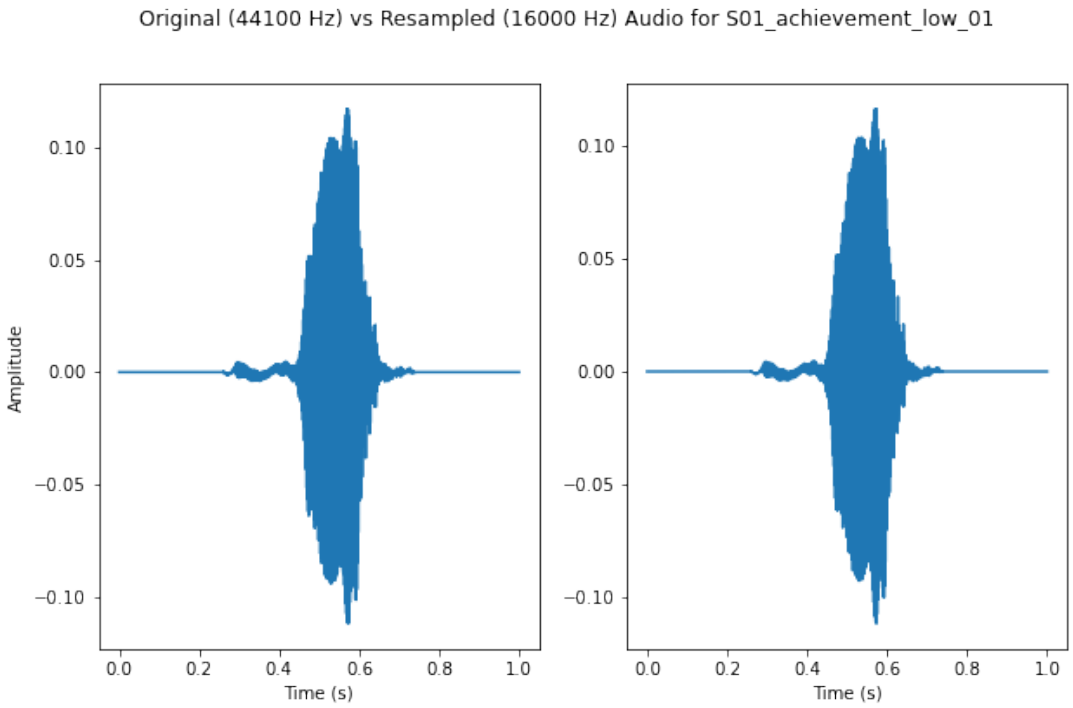


Figure 6: Resampled Audio

The transformer architecture used the exact same as the fined tuned one that the researchers for wav2vec2 developed, besides the changing of the classification head to a Linear layer of output dimension 6. The model had 95 million trainable parameters. The input sequence is of length 16000, as the audio is resampled to 16000 Hz and padded to 1 second. The optimizer was the AdamW optimizer with a constant learning rate of 1.5e-5 with a L2 penalty of 1e-6. The loss function was connectionist temporal classification loss. The batch size was 40 for both training and testing.

5 Results and Discussion

Model	Epoch	Training Loss	Testing Loss	Training Accuracy	Testing Accuracy
Logistic Regression	N/A	N/A	N/A	95.51%	20.73%
MLP	45	35.18	14.41	73.84%	19.82%
ViT 32,30	260	35.60	11.63	54.61%	47.47%
ViT 16,69	365	38.55	11.82	49.65%	47.00%
ViT 8,23	275	46.37	13.17	35.60%	33.64%
ViT-Base	3	47.77	13.93	31.80%	26.73%
wav2vec2	131	0.85	1.66	72.81%	47.00%

Table 5: Loss and Accuracy for All Models

The appendix contains all of the individual loss plots and accuracy plots.

Table 5 shows various losses and accuracies for a selected for each model. This epoch was selected where the testing loss was decreasing, as to indicate the model hasn't yet overfit to the

training data, and the test accuracy was highest. From all of the models, it is apparent that ViT 32,30 performed the best when viewing accuracy as the main metric. All of the transformer based models outperformed the logistic regression and MLP models, which were considered to be the baselines for this experiment. When viewing Figures 8.2 and 8.2, it is apparent for MLP that is was overfitting on the training set as the testing loss increased, with no change in test accuracy. However, what was surprising was the ViT-Base immediately started to overfit on the training data, which is why the best model was selected at epoch 3. In Figures 8.6 and 8.6, ViT-Base reached 0 training loss and 100% accuracy, while testing loss increased and testing accuracy remained unchanged. Although ViT-Base was not pretrained or fine tuned on spectrograms, it only did well during training, but was not generalizable for data it has never seen, seemingly guessing random predictions. When comparing the other ViT models, they all showed signs of underfitting the data. These models were not pretrained or finetuned before using the VIVAE dataset with it. The patch size of (32,30) would contain the most frequency information from 0 Hz to 2048 Hz, compared to (16,69) which only has 0 Hz to 768 Hz, and (8,23) has 0 Hz to 256 Hz. The smallest patch size of (8,23) would results in the most amount of patches of 240, but could not generalize well compared to the other two as it encapsulates the smallest frequency information. Comparing (16,69) and (32,30), an ideal patch size would have been (32, 69) as it would also contain the most temporal information of the audio. wav2vec2 also did well as it achieved a testing accuracy of 47%. It is not comparable to judge the loss of wav2vec2 to the other models as they were using different loss functions, but Figures 8.7 and 8.7 show that even though the testing loss remained constant on average as the number of epochs increased, the testing accuracy increased until it started to flatline. wav2vec2 did have the largest training accuracy when excluding the baseline models, so a better tuning of the hyperparameters seems optimal to develop a better model, especially the learning rate as it remained constant. However, wav2vec2 took 4 hours to achieve these results when training, compared to the ViT models which took around 1 hour each.

Model	Class 0 (Achievement)	Class 1 (Anger)	Class 2 (Fear)	Class 3 (Pain)	Class 4 (Pleasure)	Class 5 (Surprise)
Logistic Regression	0.390	0.472	0.524	0.568	0.549	0.419
MLP	0.914	0.814	0.878	0.834	0.858	0.827
ViT 32,30	0.781	0.811	0.809	0.814	0.878	0.841
ViT 16,69	0.826	0.811	0.810	0.815	0.845	0.838
ViT 8,23	0.711	0.728	0.681	0.707	0.667	0.740
ViT-Base	0.708	0.746	0.675	0.738	0.812	0.713
wav2vec2	0.741	0.808	0.697	0.674	0.745	0.785

Table 6: AUC for All Classes for Each Model

The appendix contains all of the individual ROC curves for each label. The ROC curves were created by doing a one versus rest for each class. When look at the AUC values, it is apparent that logistic regression performed the worst, not even better than a random classifier, but MLP was the best. As this dataset was found to be balanced in the class labels, it is apparent that MLP had good performance on predicting the positive class at the cost of high false negative rates. It was already known that MLP by itself could not interpret sequential or image data without some sort of flattening or preprocessing, and that it had the lowest accuracy as shown in Table 5. Looking at the other transformers, ViT 32,30 and ViT 16,69 were the next best models. Overall, based on looking at testing accuracy, AUC, and loss plots, ViT 32,30 was the best model on the VIVAE dataset.

6 Conclusion

Using transformers with the VIVAE dataset achieved reasonable results. It was shown that vision transformers can be applicable to audio data when representing audio as a Mel spectrogram. It was also shown that using pretrained models as a downstream task for audio classification did not do well on ViT-Base, which was pretrained on images of scenes and items, not spectrograms, but did considerably well on wav2vec2 as it pretrained on audio data of spoken text. In the developing field of machine learning, it would not be surprising in the future for a better pretrained model to be available to the public to extend and make better predictions for sentiment classification.

7 Future Work

- Dynamic splicing of the audio recordings to remove silence from the beginning and the end
- Dynamic batching based on the longest sequence length in a batch
- Better hyperparameter tuning for wav2vec2 at the cost of long runtimes
- Use performers to handle even longer audio sequences

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

[2] Natalie Holz, Pauline Larrouy-Maestri, and David Poeppel. The Variably Intense Vocalizations of Affect and Emotion Corpus (VIVAE), October 2020.

[3] Short-time fft. https://www.mathworks.com/help/dsp/ref/dsp.stft.html.

[4] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2020.

[5] pytorch-image-models. https://github.com/rwightman/pytorch-image-models.

[6] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *CoRR*, abs/2006.11477, 2020.

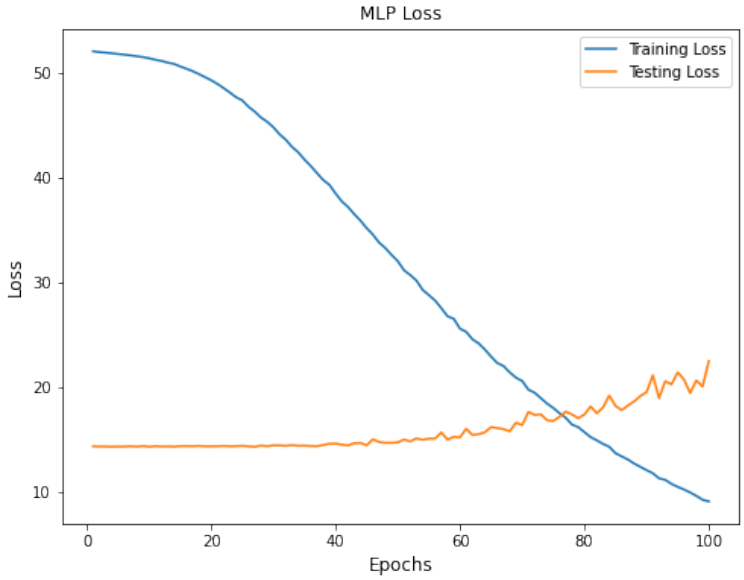
[7] fairseq. https://github.com/pytorch/fairseq/tree/main/examples/wav2vec.

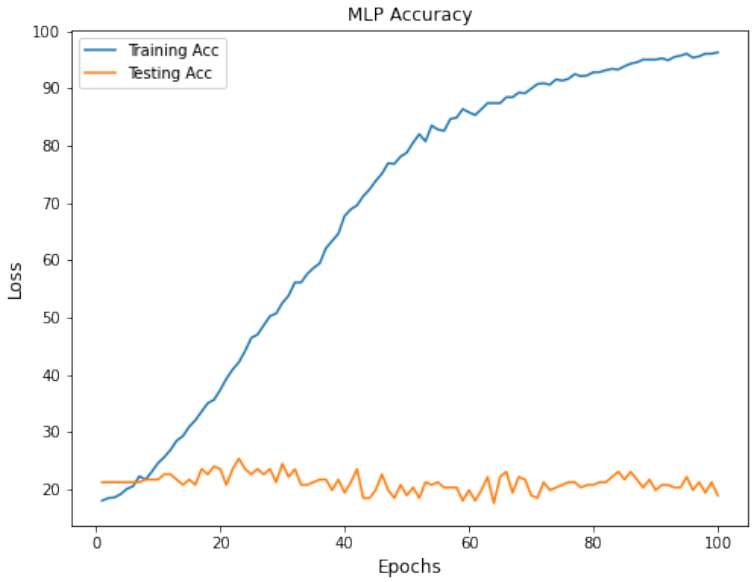
8 Appendix

8.1 Code

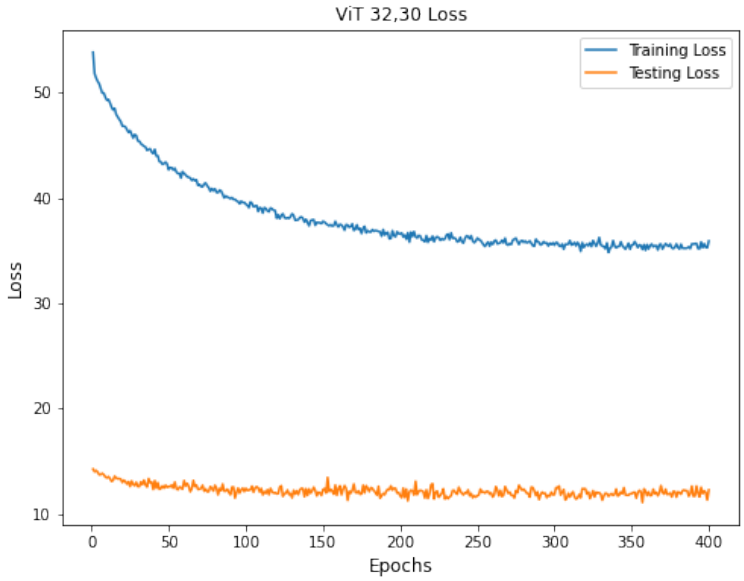
https://github.com/pakdaniel/DataMiningFinal

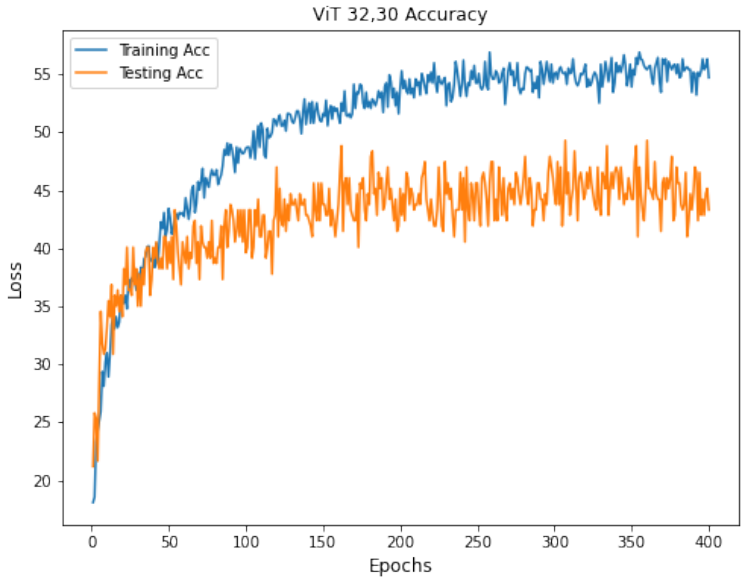
8.2 MLP Plots



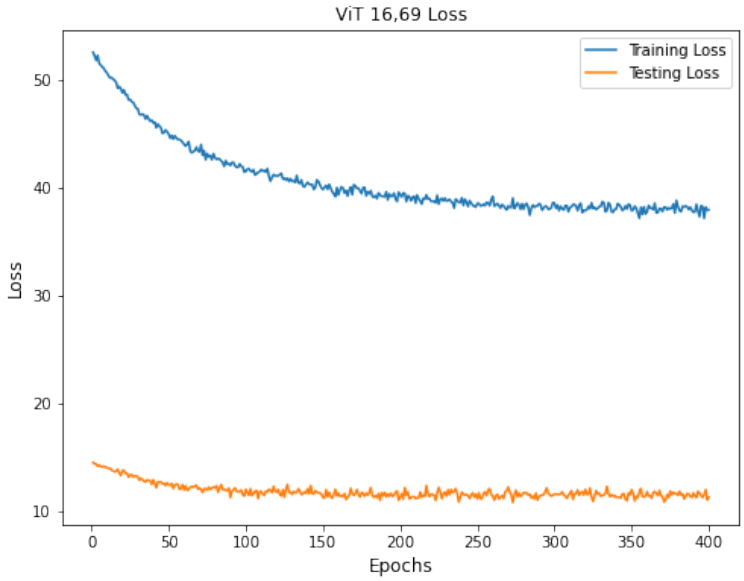


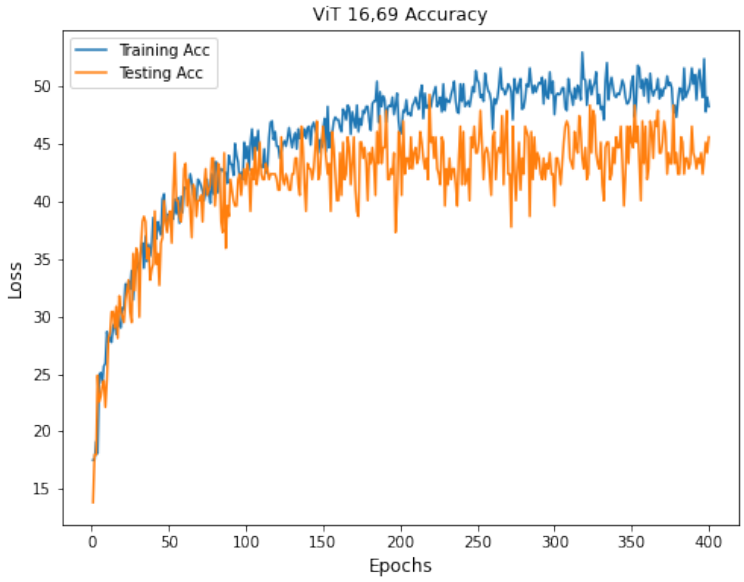
8.3 ViT 32,30 Plots



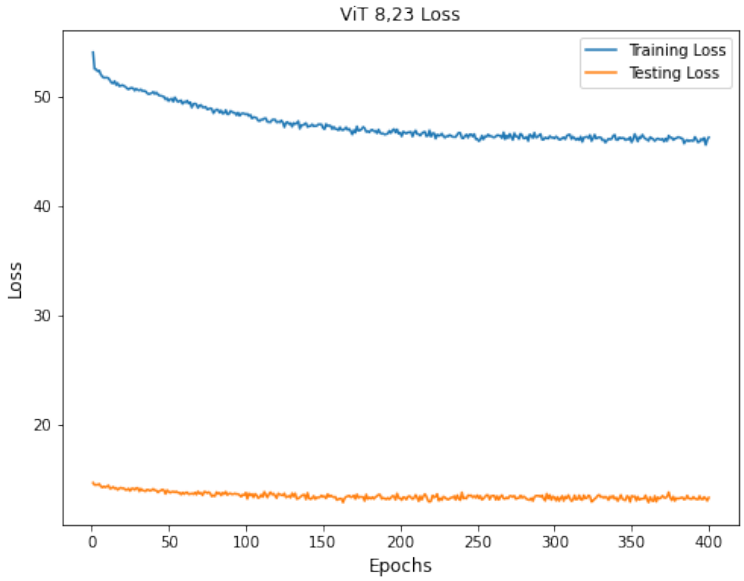


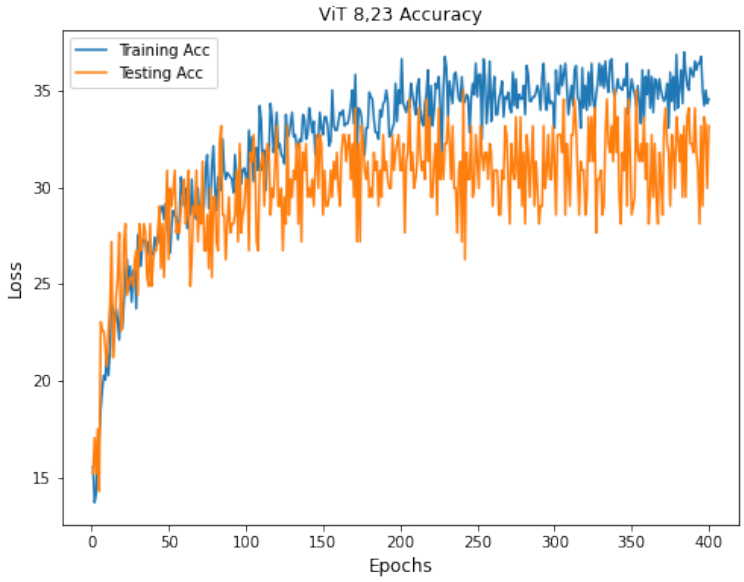
8.4 ViT 16,69 Plots



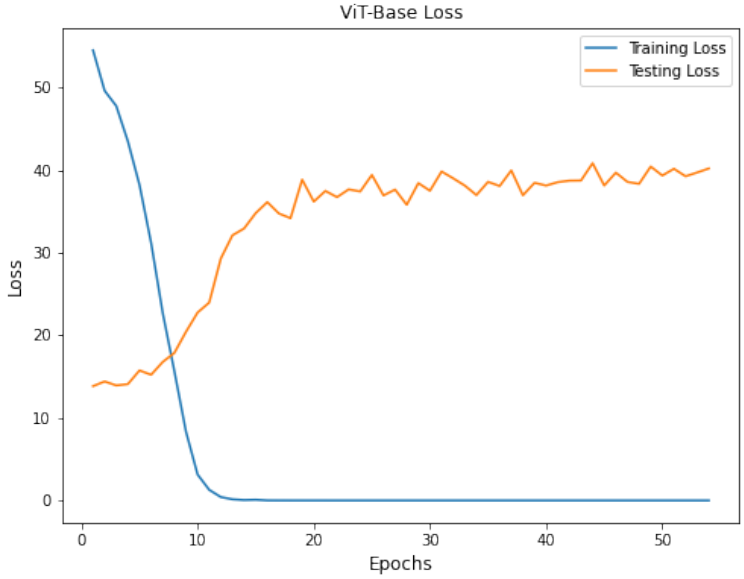


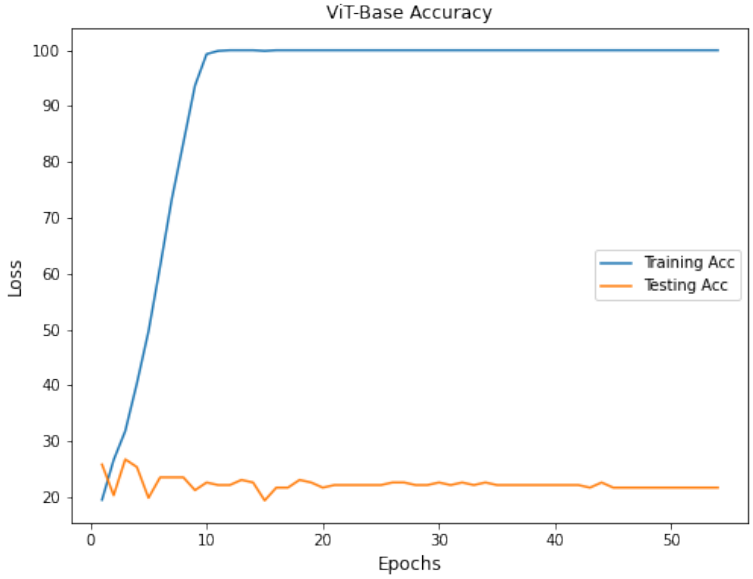
8.5 ViT 8,23 Plots



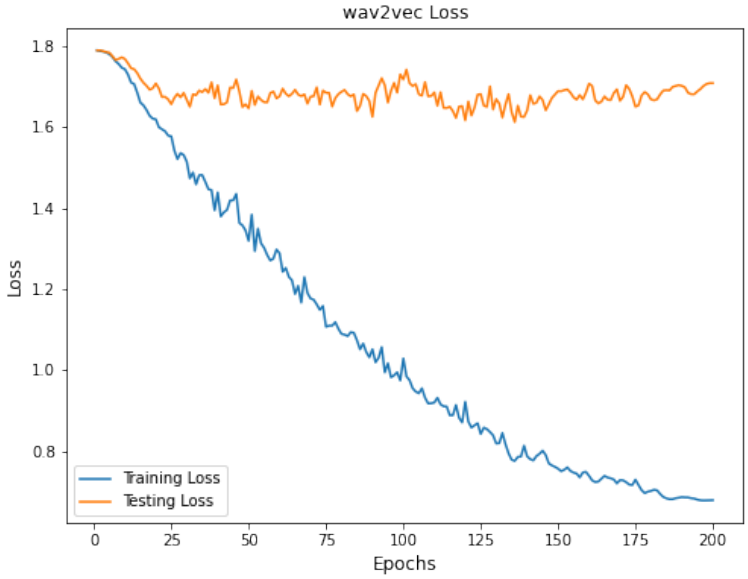


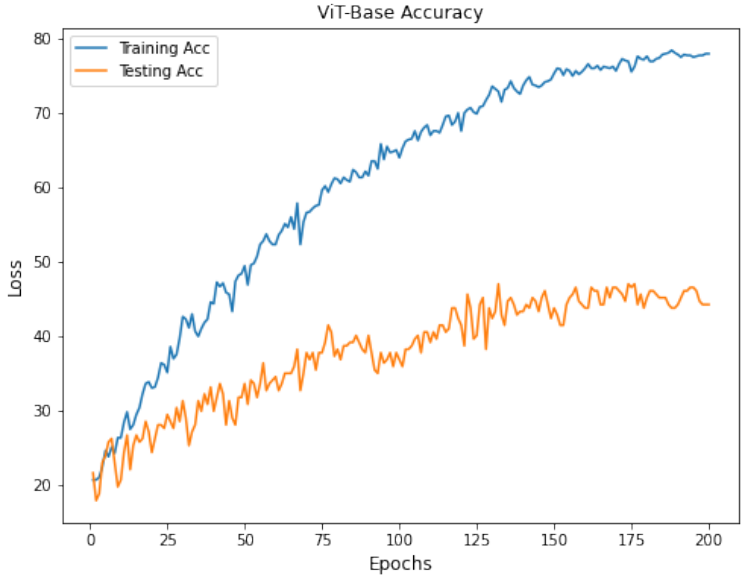
8.6 ViT-Base Plots



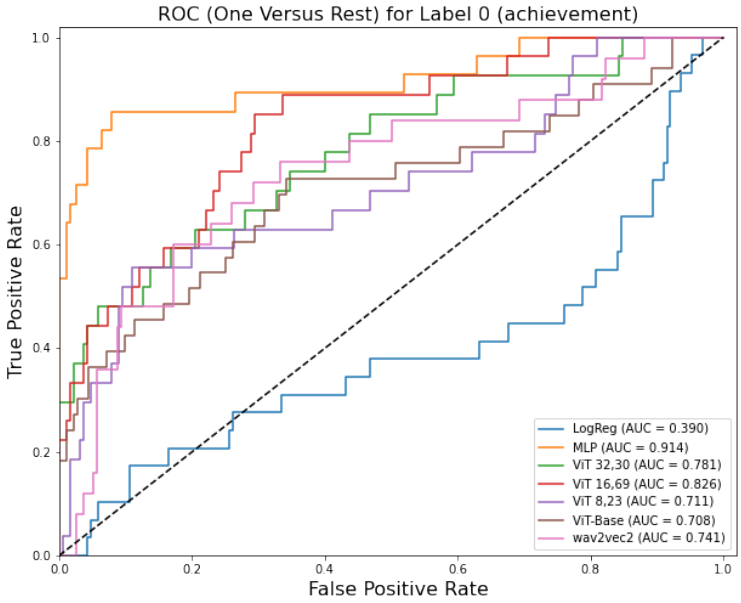


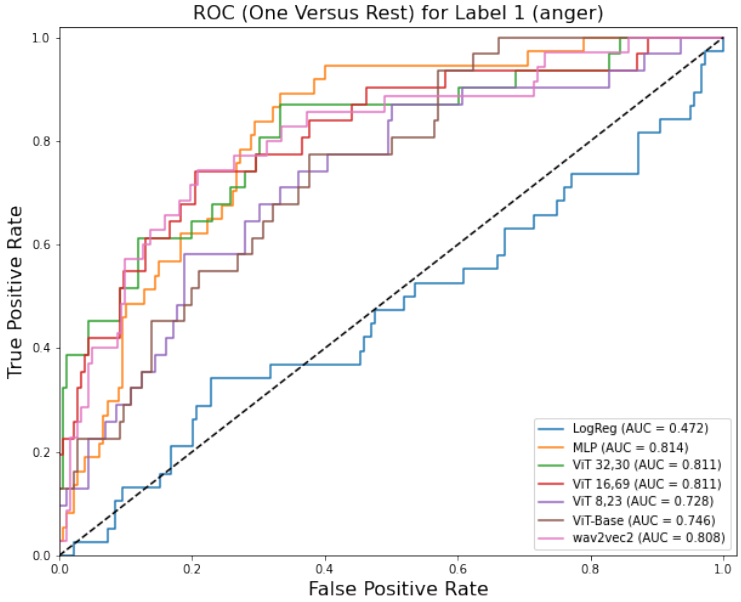
8.7 wav2vec2 Plots

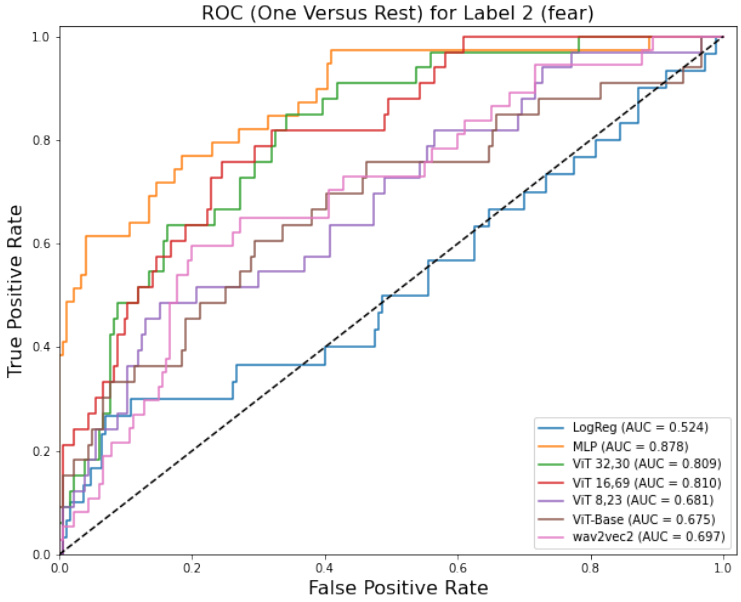


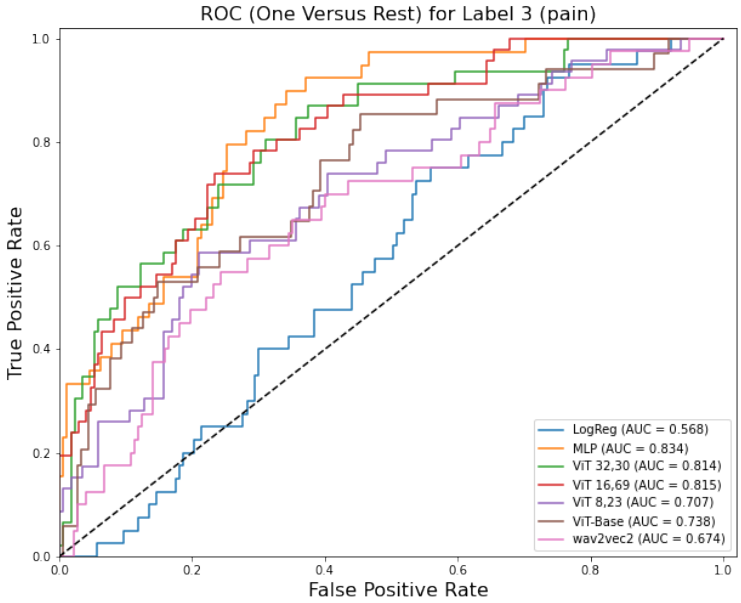


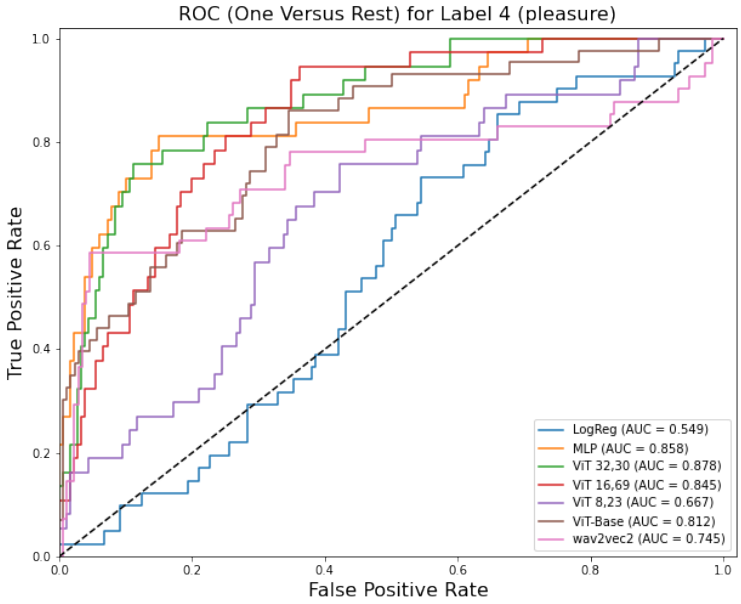
8.8 ROC Curves

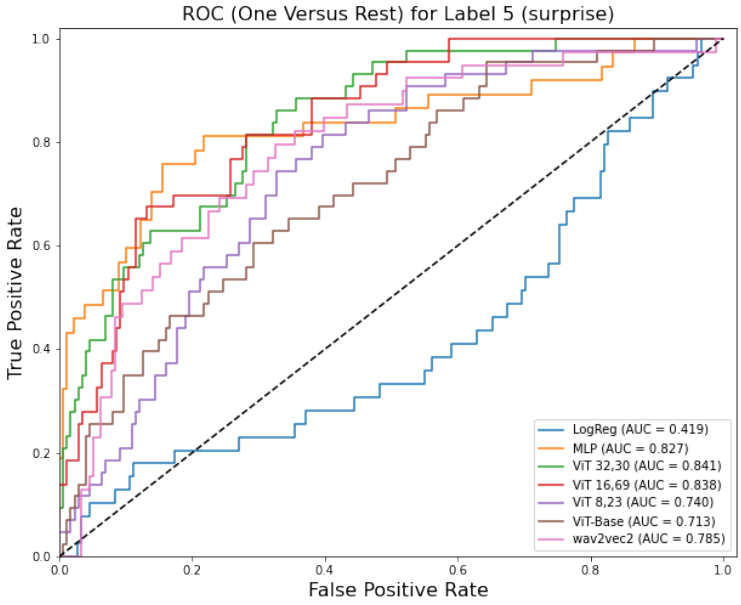





ROC (One Versus Rest) for Label 2 (fear)
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
LogReg (AUC = 0.524)
MLP (AUC = 0.878)
ViT 32,30 (AUC = 0.809)
ViT 16,69 (AUC = 0.810)
ViT 8,23 (AUC = 0.681)
ViT-Base (AUC = 0.675)
wav2vec2 (AUC = 0.697)


ROC (One Versus Rest) for Label 3 (pain)
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
LogReg (AUC = 0.568)
MLP (AUC = 0.834)
ViT 32,30 (AUC = 0.814)
ViT 16,69 (AUC = 0.815)
ViT 8,23 (AUC = 0.707)
ViT-Base (AUC = 0.738)
wav2vec2 (AUC = 0.674)


ROC (One Versus Rest) for Label 4 (pleasure)
True Positive Rate
False Positive Rate
LogReg (AUC = 0.549)
MLP (AUC = 0.858)
ViT 32,30 (AUC = 0.878)
ViT 16,69 (AUC = 0.845)
ViT 8,23 (AUC = 0.667)
ViT-Base (AUC = 0.812)
wav2vec2 (AUC = 0.745)


ROC (One Versus Rest) for Label 5 (surprise)
True Positive Rate
False Positive Rate
LogReg (AUC = 0.419)
MLP (AUC = 0.827)
ViT 32,30 (AUC = 0.841)
ViT 16,69 (AUC = 0.838)
ViT 8,23 (AUC = 0.740)
ViT-Base (AUC = 0.713)
wav2vec2 (AUC = 0.785)