Emotion Recognition in Speech

John Koumentis, Sotiris Panopoulos

Overview

The project evaluates the performance of traditional Machine Learning techniques and Deep Learning methods over predicting the correct emotion of the given human speech audio example

Datasets

SAVEE

- Recordings from four male actors
- Seven different emotions (anger, disgust, fear, happiness, pleasant surprise, sadness, and neutral).
- Recordings consist of 15 phonetically balanced TIMIT sentences per emotion (with additional 30 sentences for neutral state)
- Corpus of 480 British English utterances.

TESS

- Two actresses
- Seven emotions (anger, disgust, fear, happiness, pleasant surprise, sadness, and neutral).
- Set of 200 target words spoken in the carrier phrase 'Say the word _' portraying each emotion.
- 2800 data points (audio files) in total.

Note: In order to avoid overfitting on actor voices, the test set contains samples only for one of the dataset actors, not present at the training set.

Feature extraction

- Utilizing MidTermFeatures directory extraction module from the pyAudioAnalysis library to extract audio features.
- The sound file is divided into mid-term segments.
- For each segment, the short-term processing stage takes place.
- The feature sequence, extracted from a mid-term segment, is used for computing feature statistics.
- Each mid-term segment is represented by a set of statistics which correspond to the respective short-term feature sequences.
- The used mid-term window and step were set to 1 second and the short-term window and step to 0.1 seconds.

Selected classifier

Support Vector Classifier with RBF kernel was used for the experiments.

Accuracy and weighted F1 score were the used metrics.

Other classifiers like Logistic Regression, Decision Trees and Gradient Boosting were tested but the metric results were worse.

Methods

- Speaker dependent analysis - Overfitting
- Speaker independent analysis

Speaker dependent analysis - Overfitting

- Train a classifier by splitting the train set using a stratified train test split method (0.75 - 0.25) and check the performance
- Samples from the same speakers at both train and validation sets.
- For SAVEE dataset, it appears that the validation metrics (accuracy, weighted f1-score) and the test ones are similar. Probably due to small sample size, the classifier cannot generalize properly.
- In TESS dataset, the overfitting consequences are significant, due to the fact that both train and validation sets contain audio samples only from one speaker.

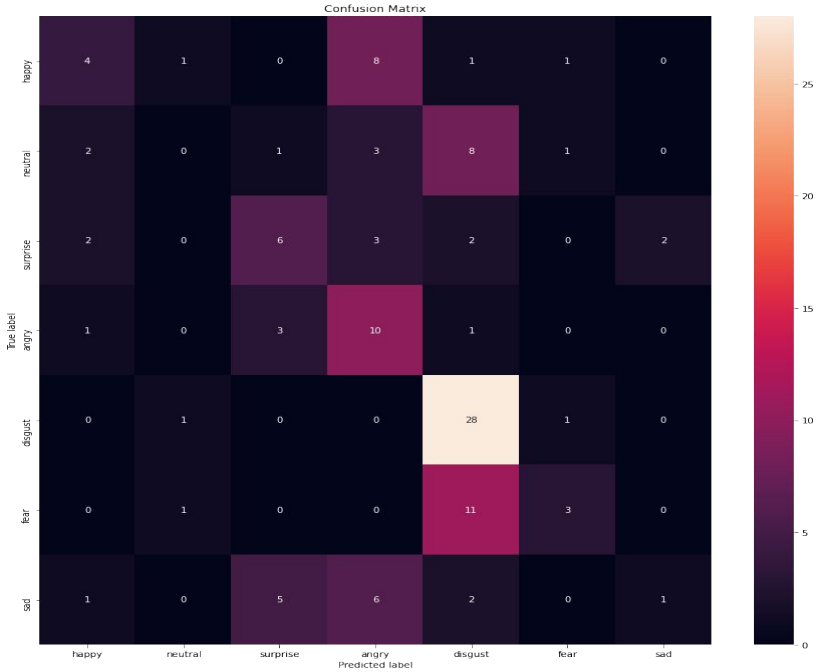
Speaker dependent analysis - Overfitting SAVEE

Validation Accuracy: 0.4444444444444444

Validation F1 Score: 0.41031156257660656

Test Accuracy: 0.43333333333333335

Test F1 Score: 0.36383342895391085



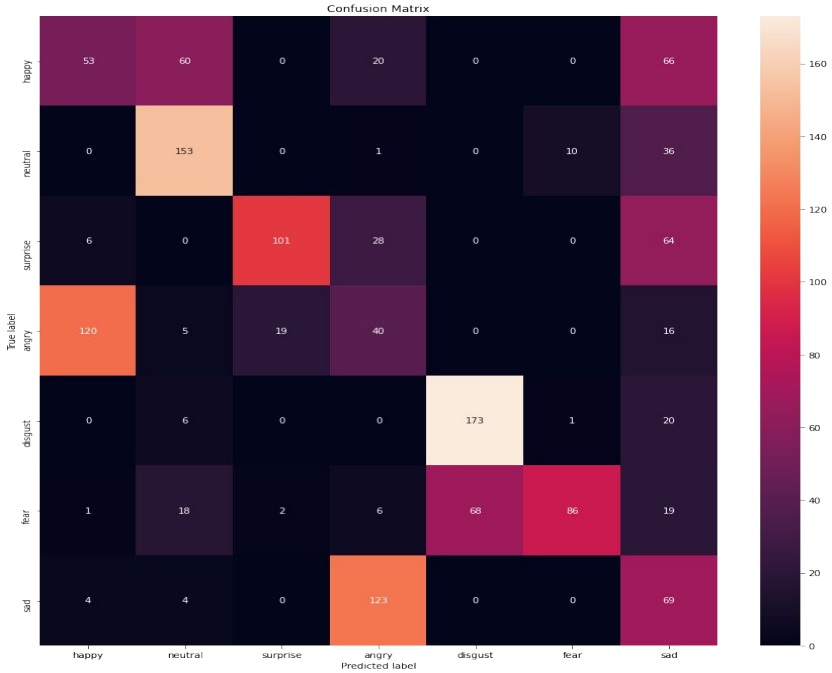
Speaker dependent analysis - Overfitting TESS

Validation Accuracy: 0.9714285714285714

Validation F1 Score: 0.9718660239201761

Test Accuracy: 0.48283261802575106

Test F1 Score: 0.4898912170670342



Speaker independent analysis

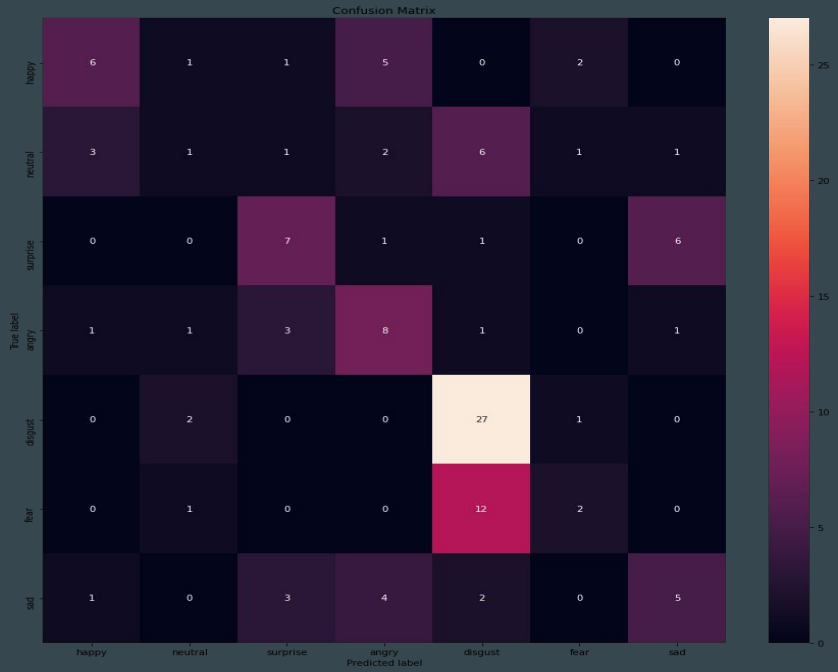
- Performance of the Support Vector Classifier, over predicting the correct class on the speakers that belong to the test set at SAVEE and TESS datasets, respectively.
- SAVEE: Accuracy score and F1 score are slightly better than those achieved at the speaker dependent analysis. This makes sense, since more training examples are fed to the classifier in this approach.
- TESS: Similar test set scores to the speaker dependent analysis are achieved at the speaker independent analysis. As long as this dataset is larger than SAVEE, the split that took place during the speaker dependent analysis does not affect the classifier performance significantly.

Speaker independent analysis

SAVEE

Test Accuracy: 0.4666666666666667

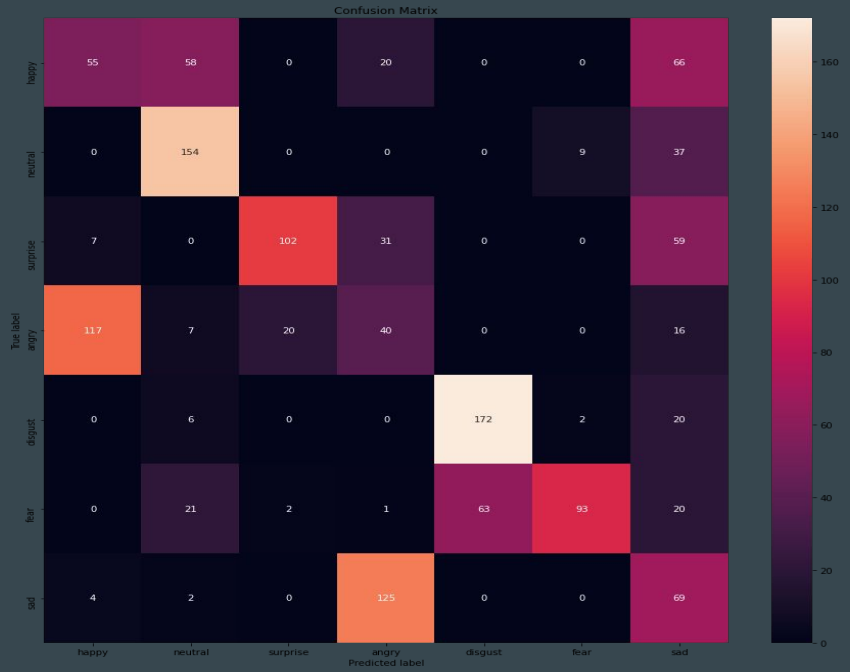
Test F1 Score: 0.42441171697500807



TESS

Test Accuracy: 0.4899856938483548

Test F1 Score: 0.49789701851976587

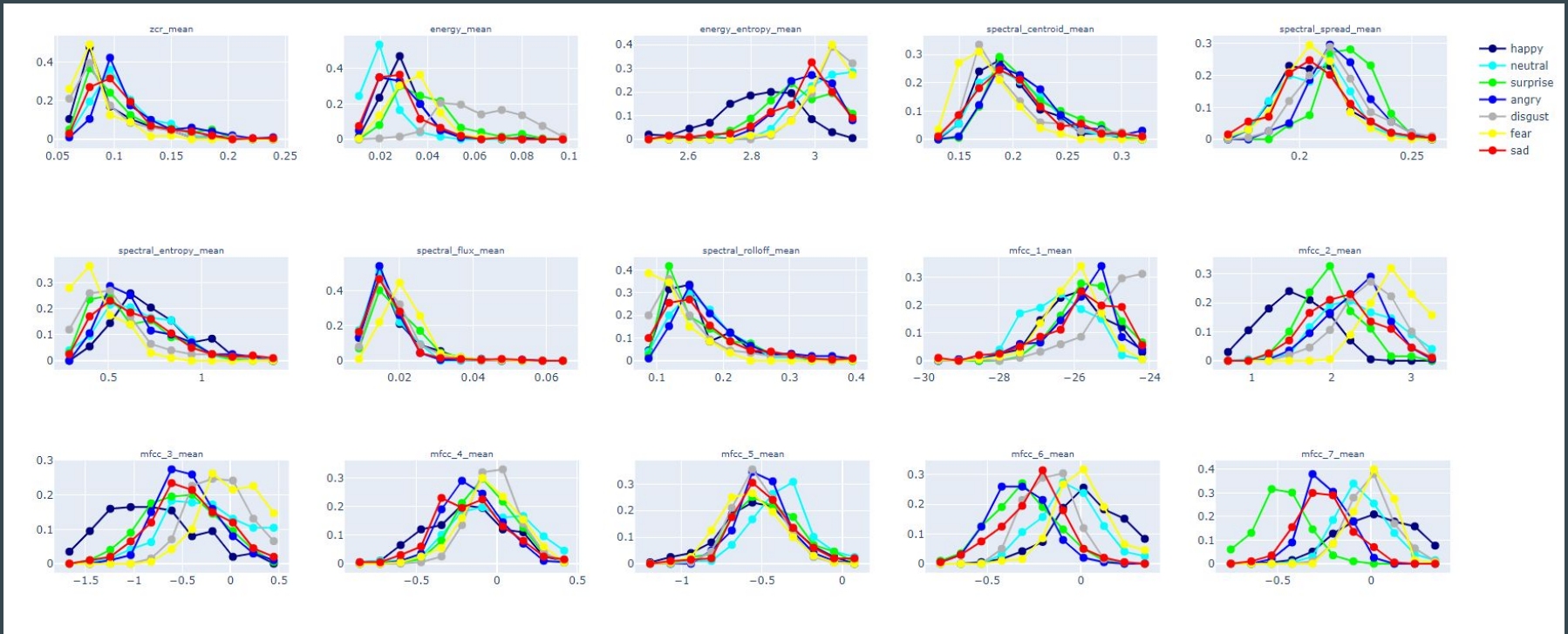


Hyperparameter tuning

- The purpose of the hyperparameter tuning is to select the best parameter combination for the classifier, in order to achieve better scores at the selecter metrics and hence, better classification results.
- The chosen hyperparameters were the C parameter (regularization parameter) and the decision function shape. The tuning was performed at the accuracy score metric.
- It appears that the tuned model overfit the train set, hence the scores at the test set are worse that those returned by the unoptimized model.
- The tuning that took place for the TESS train data seems that it slightly improved the metric scores.

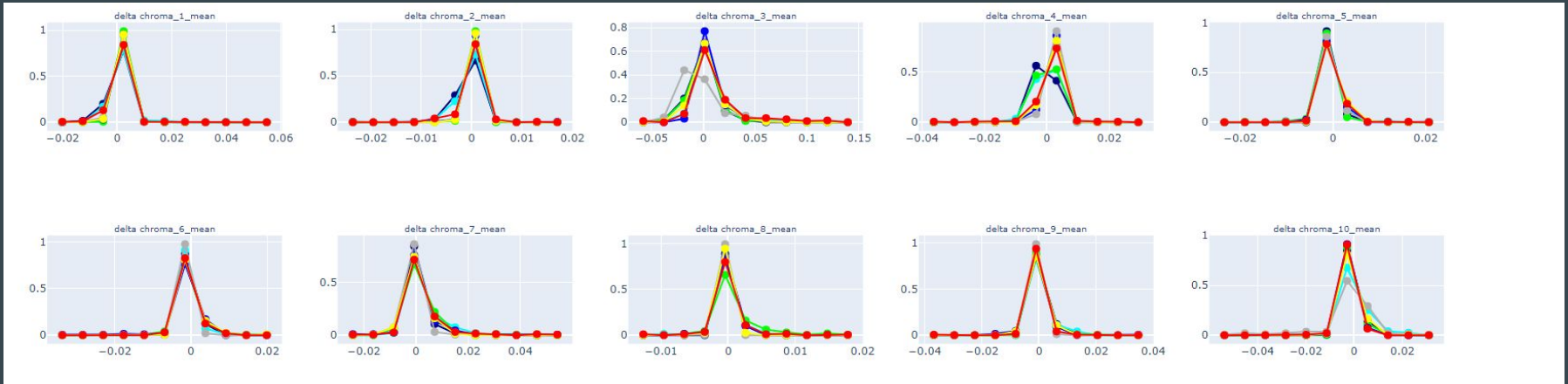
	SAVEE Unoptimized	SAVEE Best Model	TESS Unoptimized	TESS Best Model
Accuracy	0.4666666666666667	0.4166666666666667	0.4899856938483548	0.5028612303290415
F1 Score	0.4244117169750080 7	0.4017809585744 3685	0.4978970185197 6587	0.5067867142409 216
Hyperparameters: C	1.0	10.0	1.0	10.0
Decision function shape	ovr	ovo	ovr	ovo

Feature Histograms



Features like mfcc mean and mfcc std have significant variation among the classes.

Feature Histograms



However, features like the energy entropy mean and the delta chroma mean have almost identical values at all classes.

Feature Selection

Variance Threshold method was used, to remove features with variance below the specified threshold.

SAVEE

Initial features: 138

Remaining features: 73

Test Accuracy: 0.5 (0.46666667)

Test F1 Score: 0.4760206613914178 (0.4244117)

TESS

Initial features: 138

Remaining features: 64

Test Accuracy: 0.437768240343347 66 (0.4899856938483548)

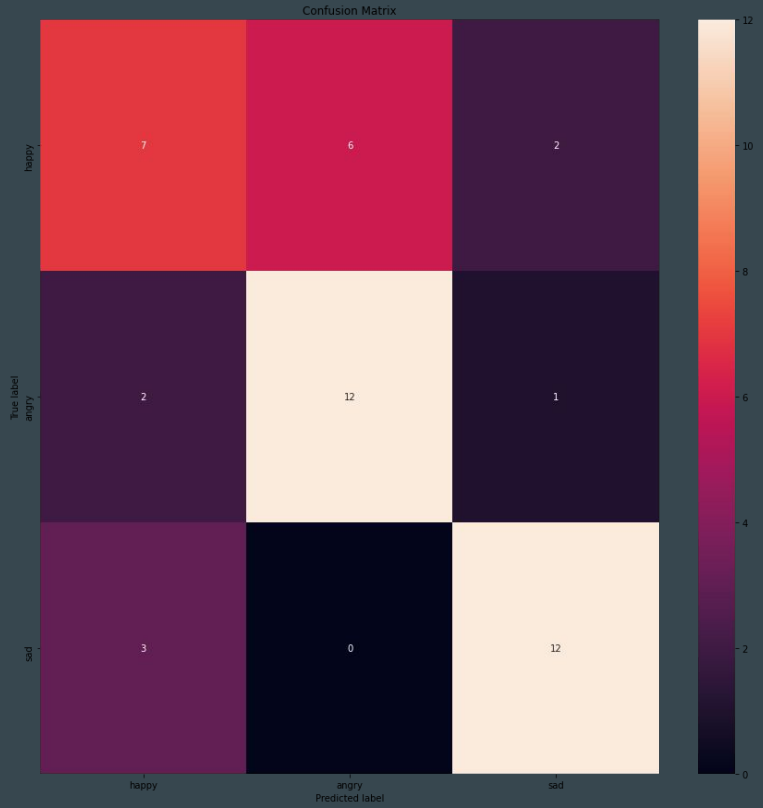
Test F1 Score: 0.4511318218770213 (0.49789701851976587)

Simplify problem - Reduce classes

SAVEE

Test Accuracy: 0.6888888888888889

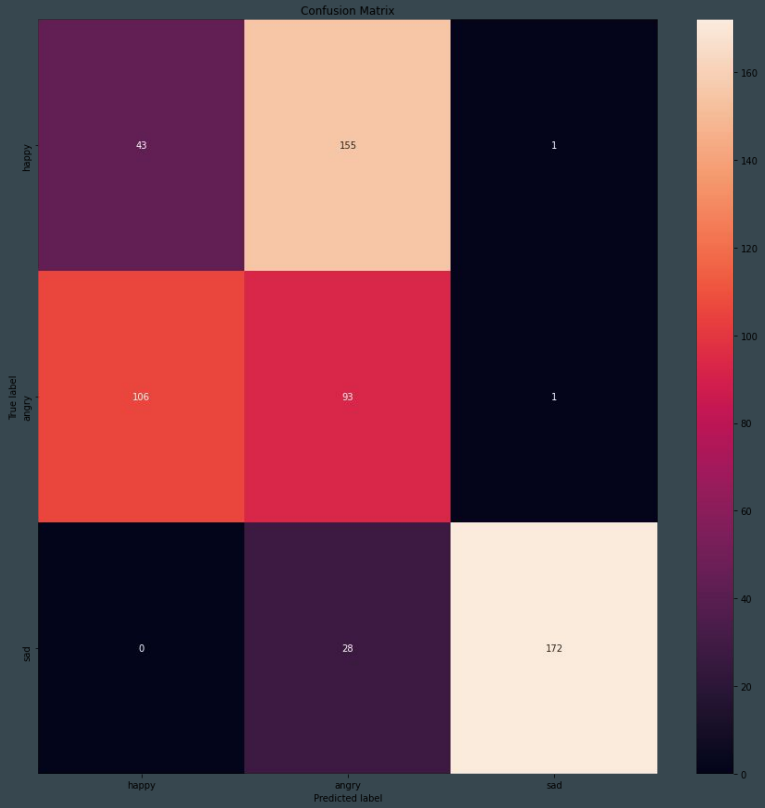
Test F1 Score: 0.6819304152637485



TESS

Test Accuracy: 0.5141903171953256

Test F1 Score: 0.5196771964525523



Dataset bias investigation

Checked classifiers trained on SAVEE at the test set of TESS and vice versa.

SAVEE

Accuracy on TESS test set: 0.27968526466380544

F1 Score on TESS test set: 0.294783568995891

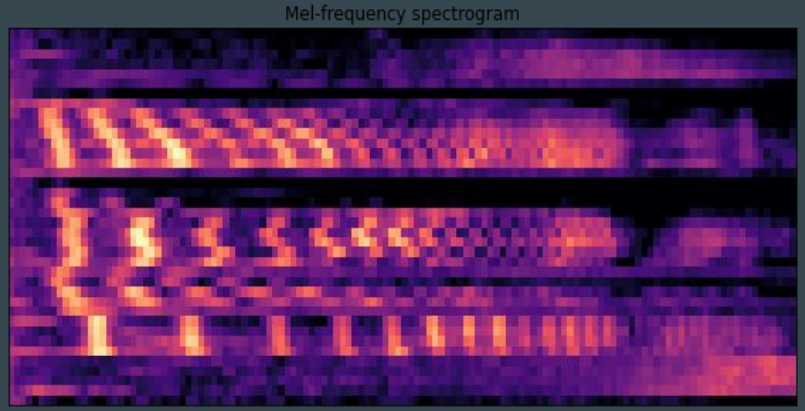
TESS

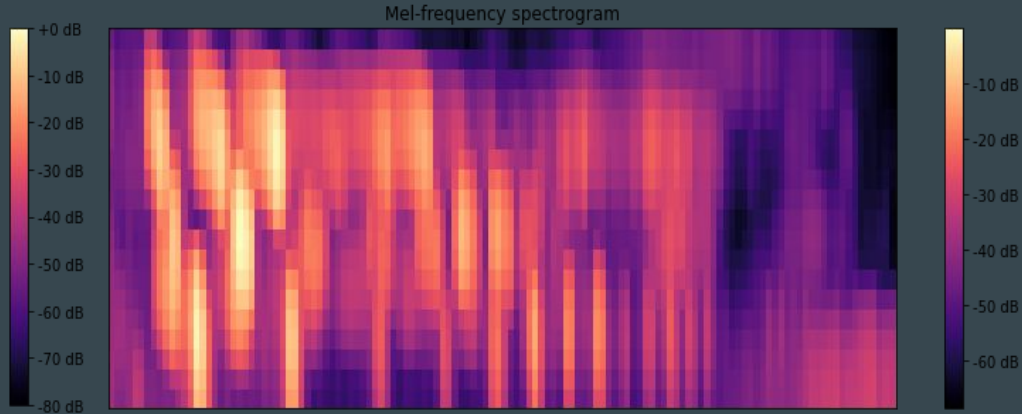
Accuracy on SAVEE test set: 0.19166666666666668

F1 Score on SAVEE test set: 0.16619719395621338

Deep learning approach

- The approach utilizes the deep_audio_features library to classify the emotions via training a CNN with the respective Mel spectrograms of the audio files as input.
- We tested sampling settings:
  - WINDOW_LENGTH = (50 * 1e-3), HOP_LENGTH = (50 * 1e-3)
  - WINDOW_LENGTH = (1), HOP_LENGTH = (0.1)



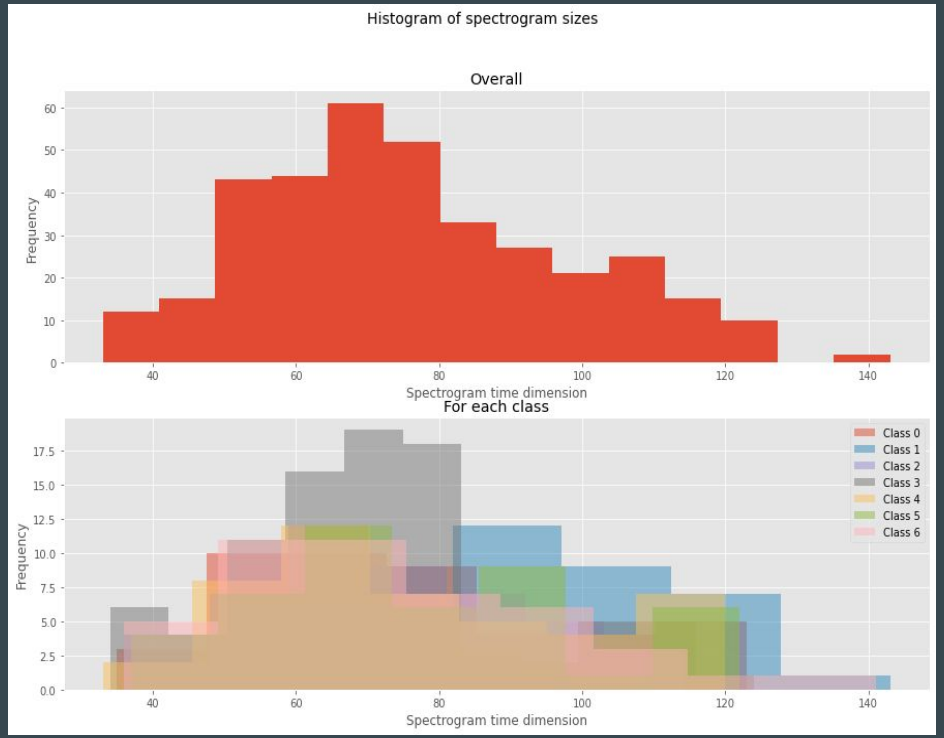


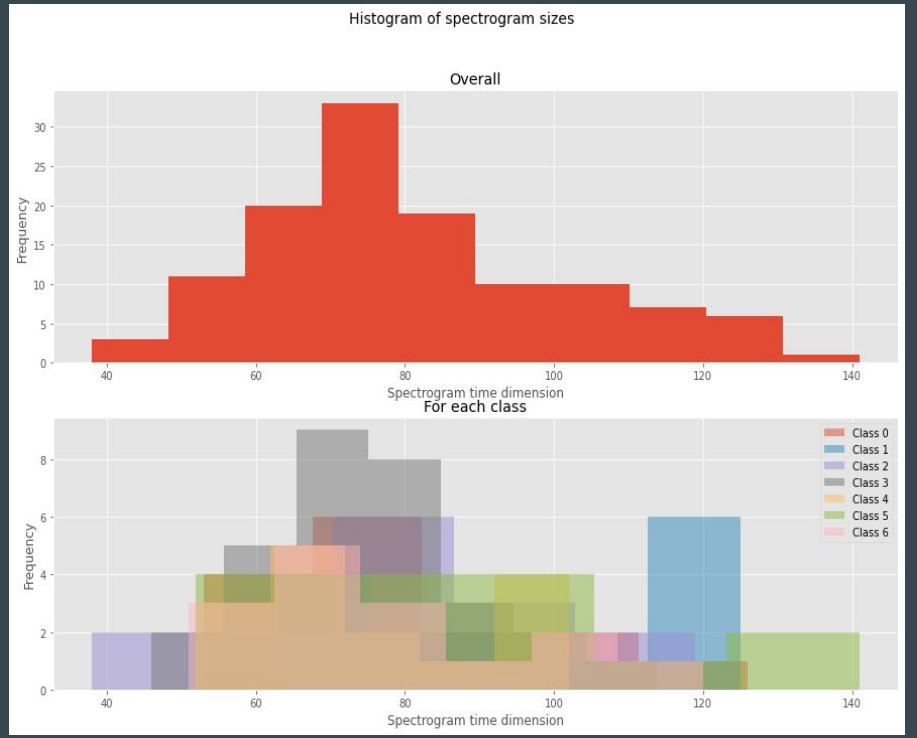
Deep learning approach- SAVEE

Best model's validation accuracy: 0.325

Best model's validation f1 score: 0.20864435393276057

Best model's validation loss: 0.10319621463616689



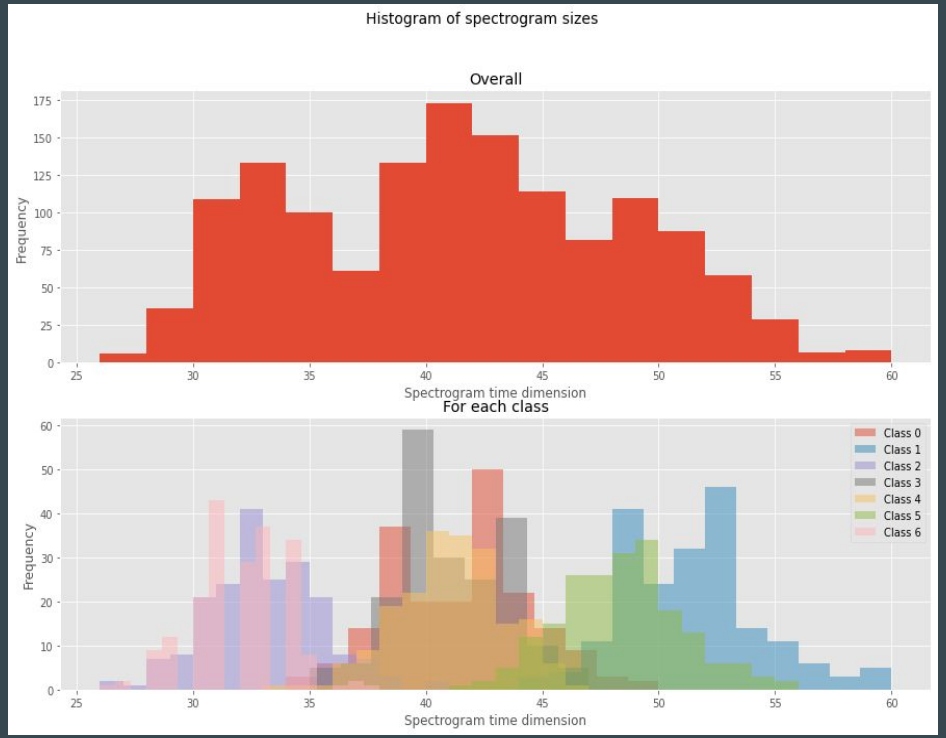


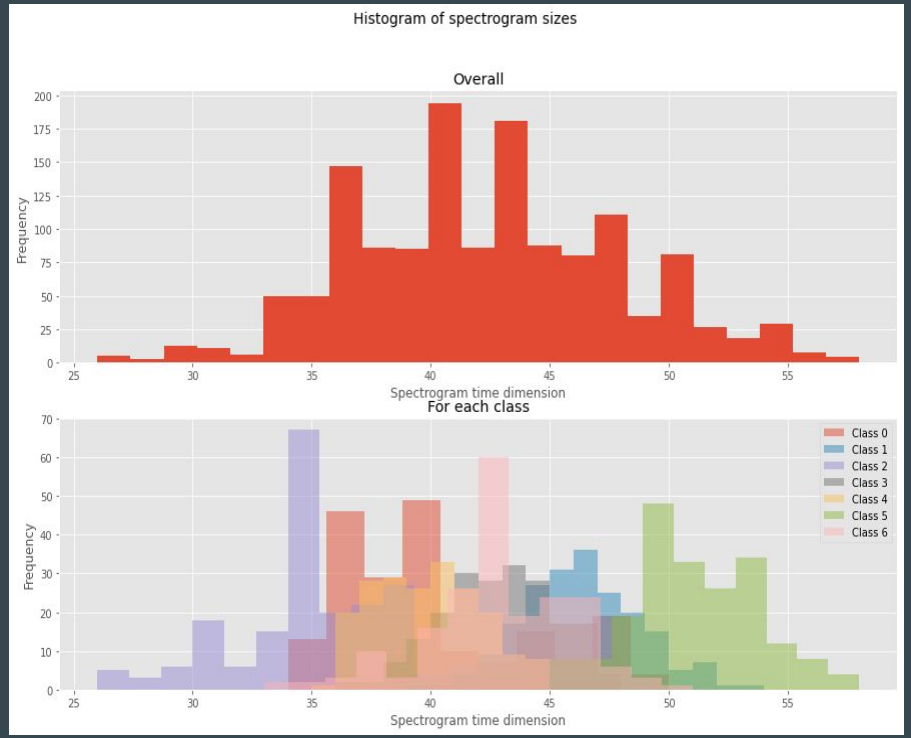
Deep learning approach- TESS

Best model's validation accuracy: 0.5586552217453505

Best model's validation f1 score: 0.4941300333481178

Best model's validation loss: 0.13344837181557914



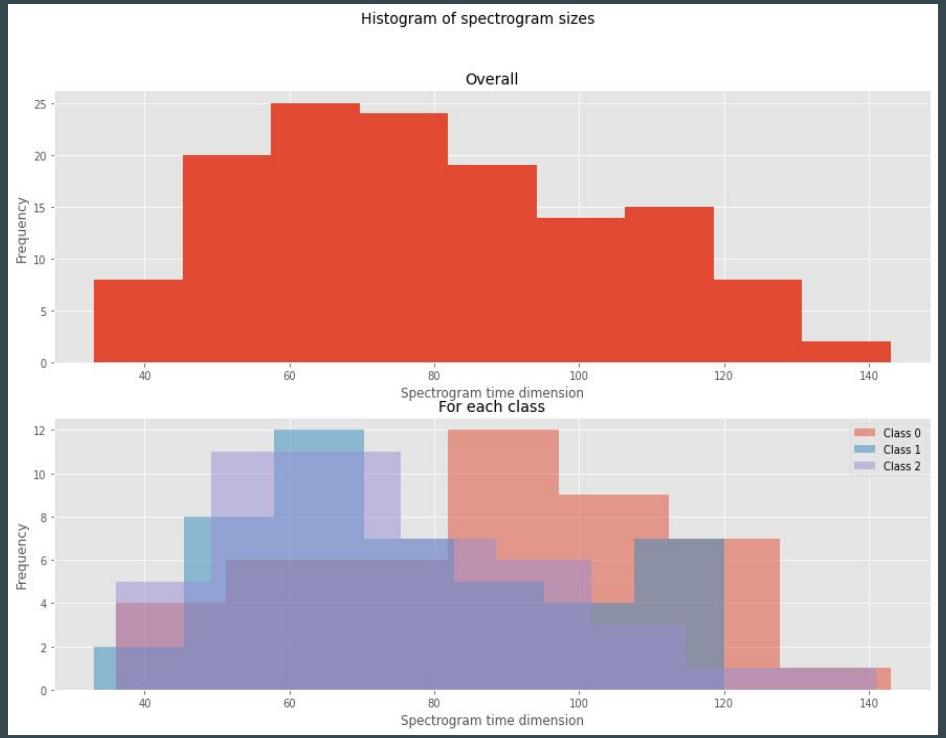


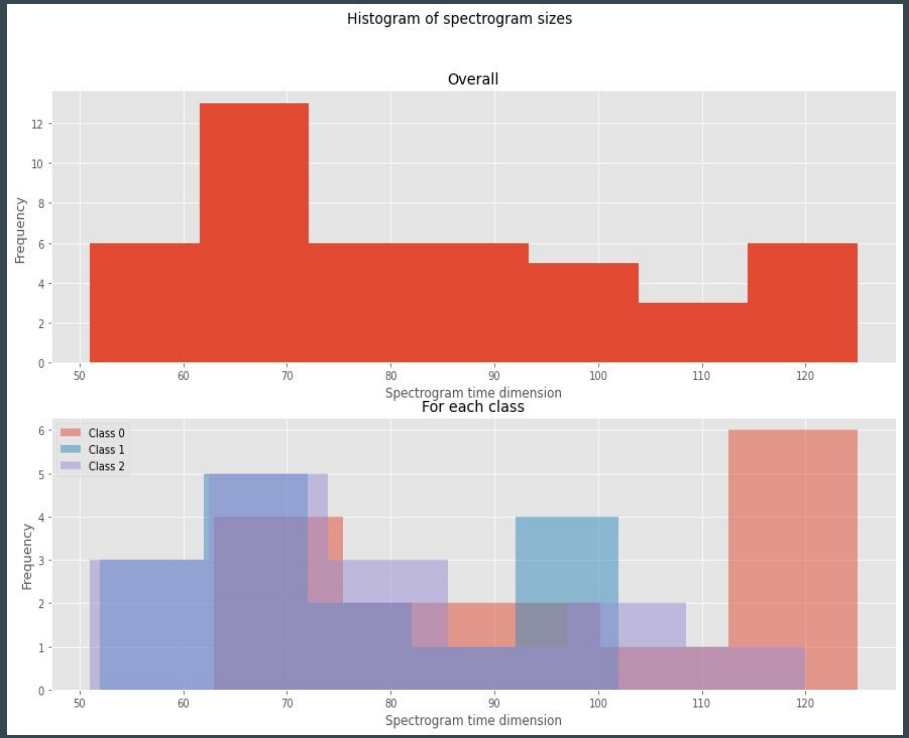
Deep learning approach- SAVEE (3 classes)

Best model's validation accuracy: 0.4444444444444444

Best model's validation f1 score: 0.6388888888888888

Best model's validation loss: 0.04166735013326009



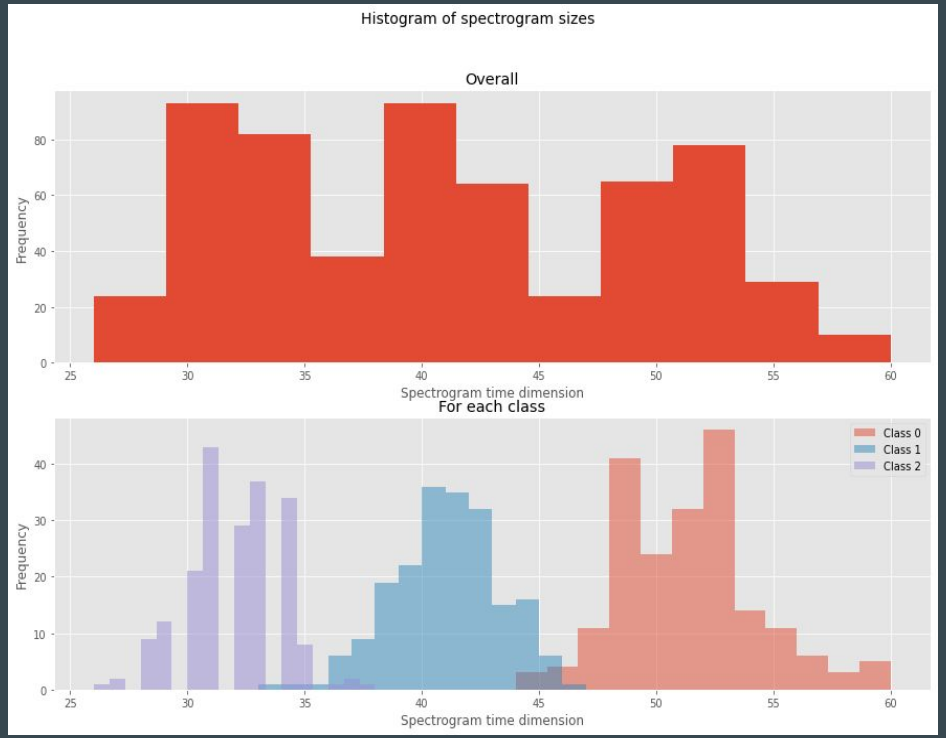


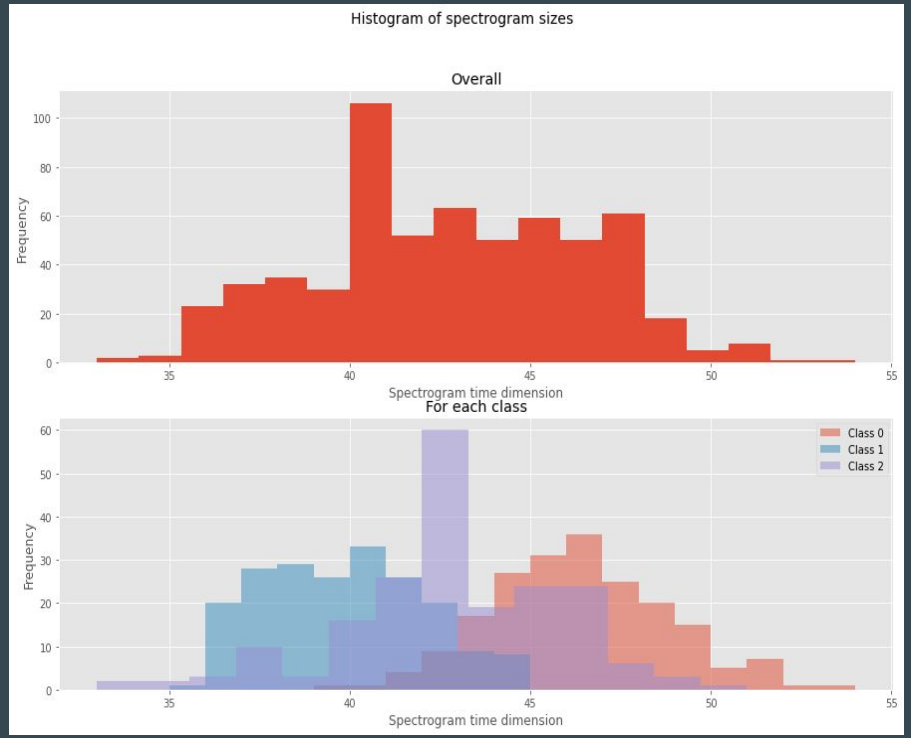
Deep learning approach- TESS (3 classes)

Best model's validation accuracy: 0.5075125208681135

Best model's validation f1 score: 0.4132753928758694

Best model's validation loss: 0.132855205583652





Deep learning approach

- The results from the Mel spectrogram classification using Convolutional Neural Networks indicate that the method requires a larger amount of data to generalise properly.

- The validation loss gets decreased only for a small number of epochs, after which the models overfit the training data and the training stops due to the early stopping method.

Conclusions

- Both classical machine learning approach with feature extraction and deep learning approach could not achieve accuracy score and F1 score larger than 51% at the 7 classes problem and larger than 69% at the simplified 3 class problem.
- Classes are not easily separable.
- Both datasets contain recordings from actors trying to express the respective emotion. Due to this, bias is inserted since every actor express an emotion in their own way.
- Future improvements would include:
    - Upsampling the datasets by introducing noise to the audio samples.
    - Investigate more datasets, to exclude the dataset bias.
    - Investigate audio samples from different languages.
    - Use a pre-trained deep neural network and transfer learning for fine tuning.
    - Examine not only audio, but face expressions (emotion detection through Computer Vision) as well.