Capstone Project:
Speech Emotion Recognition with Spectrograms and Convolutional Neural Networks

Kris Tost
Machine Learning Nanodegree Program
Udacity

August 2, 2018

1 Definition

1.1 Project Overview

Emotion Recognition in computing involves the detection of human emotion from one or more modalities such as audio speech, facial features in images or videos, or the semantic and linguistic analysis of transcribed speech [1]. Speech Emotion Recognition (SER) utilises acoustic features present in speech to perform emotion classification through purely audio-based modalities [2]. In a future where robotics [3], assistive living devices [4, 5] and virtual companions are the norm [6, 7], it is imperative that such applications can detect and respond intelligently to human emotion [8, 9], particularly in situations where other modalities such as facial features and language semantics aren't available due to device constraints or even privacy restrictions.

1.2 Problem Statement

Emotion classification models used in SER typically use discrete categorical labels [10], often based on the big 6 emotions identified by Paul Ekman [11]. For example, the labels "anger", "happiness", and "sadness" are frequently-used categorical labels employed to describe emotional speech. Predicting and classifying emotional speech in such an instance can therefore be considered a multi-classification supervised learning problem.

The challenge for this capstone project is to apply supervised learning algorithms to develop a model capable of classifying audio recordings of emotionally-charged human speech by outputting a discrete emotion class label, and/or a set of probabilities for the emotion classes under consideration. The goal is to treat the SER problem as an image classification task by training a Convolutional Neural Network (CNN) on spectrogram images of the spoken utterances within the Emo-DB dataset [12, 13, 14].

At a high level, the general workflow for this consists of the following stages:

- Data collection, analysis and pre-processing
- Generating spectrograms and data augmentation
- Model Training
- Hyperparameter Tuning

- Evaluation and assessment against speaker-independent test data using k-fold cross validation

1.3 Metrics

In the field of SER, Weighted Average Recall (WA) and Unweighted Average Recall (UAR) are the typical metrics used in the community for establishing performance [3, 15]. WA is the mean of all the recall scores for each class, where each class' recall score is weighted by the number of instances for that class.

Let rc_i be the recall score where TP is true positives and FN is false negatives for class i:

$$rc_i = \frac{TP}{TP + FN}$$

and let WA be:

$$WA = \frac{(rc_i * |c_i|)...(rc_n * |c_n|))}{|c_i| + ...|c_n|}$$

where n is the number of classes, and $|c_i|$ is the total number of instances for a given class i. WA is effectively the same as the Accuracy metric, expressed as the number of correctly identified samples divided by the total number of samples:

$$\frac{\#\ correct\ predictions}{\#\ total\ samples}$$

The Unweighted Average Recall is used for unbalanced data sets where there isn't an even distribution of classes represented in the data set. It is defined as the mean value over all of the recall scores for each class:

$$\frac{1}{n}\sum_{i=1}^{n} rc_i$$

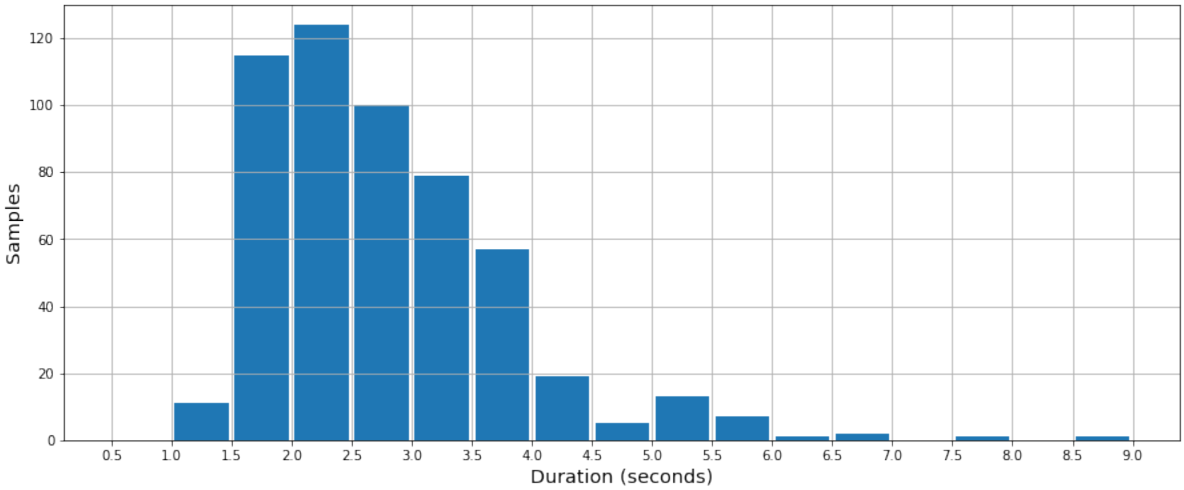
In keeping with the above metrics used in the literature and to allow for reasonable comparison between the proposed solution and the benchmark model(s), the WA and UAR metrics will be used to measure and evaluate the solution. In addition, a confusion matrix and classification report are both useful to show per-class performance on a multi-class problem.

2 Analysis

2.1 Data Exploration

This project will utilise the Emo-DB[1] emotional speech corpus [16]. It contains 535 utterances of acted emotional speech from 10 German-speaking actors (5 female, 5 male). A total of 7 emotion classes are represented in the corpus consisting of the emotions anger, boredom, disgust, fear, happiness, sadness, and neutral. Emo-DB is an oft-cited benchmark speech corpus [2, 17, 18] and is freely available for download with no license requirements. The emotions expressed within the EMO-DB corpus are highly acted and were recorded under ideal, noise-free acoustic conditions, making it a suitable benchmark dataset for establishing the viability of a classifier and relating its performance against other approaches documented in the literature [17].

Figure 1: Histogram showing EMO-DB utterance durations



2.2 Exploratory Visualisation

Pie chart/histogram/etc showing the emotion distribution for Emo-DB, maybe as a whole and also grouped by speaker.

Should visualise the emotions distribution and note the fact that Disgust is very poorly represented. Maybe show the emotion distributions by speaker and also by gender. Ideally disgust would be removed due to low number of samples, but is left in to be consistent with other studies using Emo-DB and to allow for fair comparison.

2.3 Algorithms and Techniques

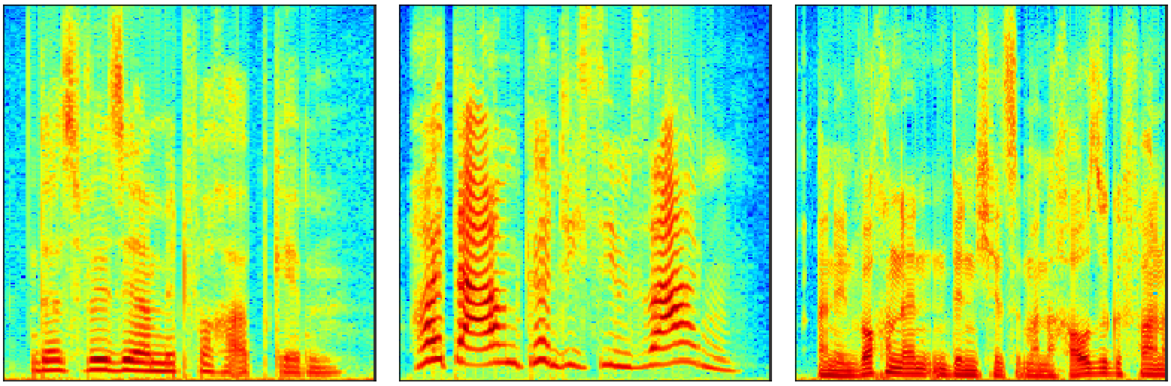
A common data extraction technique for SER is extracting Mel-scaled Frequency Cepstral Coeeficients (MFCCs) and Mel-scaled filter banks from an audio signal. This technique is prevalent in both speech recognition and SER as MFCCs and Mel-scale filter banks have proven sucessful in the extraction of prosodic features [17]. A more recent data extraction technique for audio processing tasks such as speech recognition and speech emotion recognition (particularly with Deep Learning and CNNs) is the creation of spectrogram images of audio [12, 19]. A spectrogram (see figure 2) is a 2D time-frequency representation of the changes in signal strength of various frequencies and how they vary in time [20]. Frequency is typically represented on the y-axis, with time on the x-axis. Amplitude at a given time and frequency is indicated by the intensity of colour, with low amplitudes represented by darker colours and higher amplitudes by lighter colours [12] [20]. Spectrograms can be generated from the raw audio by segmenting the signal into sequences of frames and then applying Short-time Fast Fourier Transforms (STFFT) to the audio frames [19]. For Mel spectrograms, Mel-scaled filter banks are applied to the transformed frames and finally rendered out as 2D RGB images of frequency vs time to produce a spectrogram [19].

This paper outlines the development of a CNN using a categorical cross-entropy loss function to compute accuracy in accordance with multi-classification coding conventions for CNNs and other deep learning models. CNNs are well-suited to image classification tasks, and therefore this paper will attempt to train a SER classifier using spectrogram images extracted from the Emo-DB dataset.

A multi-class classfier will be trained by using a CNN to perform feature extraction on the raw audio data as represented by spectrogram images produced from the audio files. In the

[1] http://www.emodb.bilderbar.info/download/download.zip

Figure 2: Example spectrograms from Emo-DB



case of spectrograms and image classification tasks, CNNs have been shown to be very suitable at SER classification tasks, surpassing results obtained by SVM using acoustic functionals on the same corpus [15, 21, 22]. The resulting classifier will be trained, tuned, and evaluated using k-fold cross-validation, and a comparative analysis with a benchmark model will be conducted to compare the developed solution against those in the literature.

2.4 Benchmark

Poria et al. (2017) report in their review [4] that to the best of their knowledge, the best *speaker-independent*[2] accuracy achieved on Emo-DB is 81% by Atassi and Esposito (2008)[3] using a mix of acoustic and prosodic vocal features engineered and extracted from the audio signal. Their review however precedes some of the most recent deep learning and CNN-based classifiers reported in the literature for Emo-DB, as well as alternative feature engineering techniques such as Bag-of-Audio-Words (BoAW) [23]. For example, using the BoAW [24] technique for engineering features and using an SVM classifier, the OpenXBOW [25] toolkit achieves roughly 81.4% accuracy on Emo-DB using Sequential Minimal Optimisation (SMO)[4] and Weka[5].

For CNN-based benchmarks using Emo-DB and spectrograms, Badshah et al. (2017) [12], Stoler et al. (2017) [13], and Weiskirchen et al. (2017) [14] establish benchmark performance metrics on Emo-DB using CNNs. Stolar et al. report an average accuracy of 79.68% and 76.79% for female and male speakers respectively, comparing their results against other papers utilising spectrograms and Emo-DB [26, 27]. Badshah et al. don't report overall accuracy or recall values but they do provide confusion matrices showing the precision on a per-class basis. Weiskirchen et al. report Unweighted Average Recall values (explained in the next section) of 71% for the top 1 class probability in a speaker-independent, Leave-One-Speaker-Out (LOSO) assessment.

3 Methodology

The proposed solution involves approaching SER as an image classification problem, solved by training a CNN model on spectrograms generated from raw audio inputs. The convolutional layers of the CNN are expected to perform the function of feature extraction, while the later

[2] Speaker-independence refers to models trained on one set of human speakers and tested on a different set of speakers.

[3] `http://ieeexplore.ieee.org/document/4669768/`

[4] `http://weka.sourceforge.net/doc.dev/weka/classifiers/functions/SMO.html`

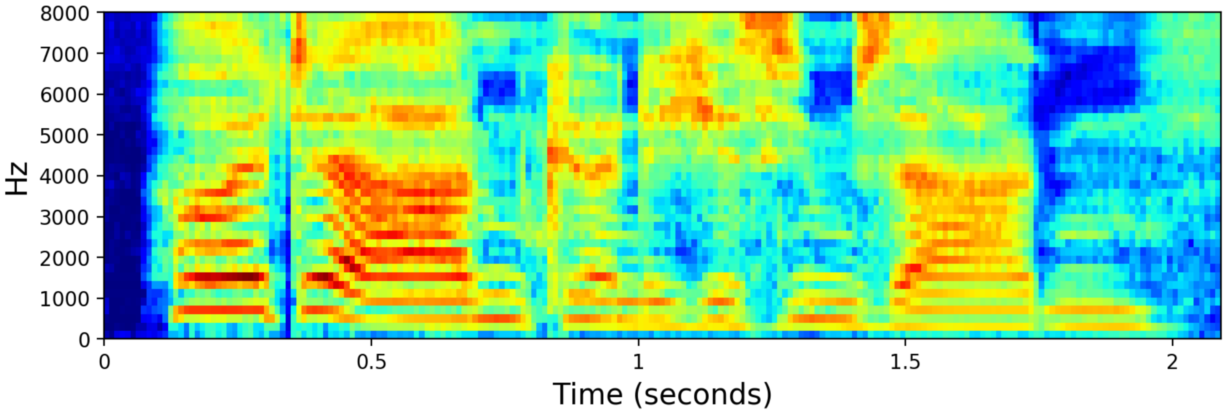
[5] `https://www.cs.waikato.ac.nz/ml/weka/`

fully-connected layers learn the patterns and adjust the network's weights to achieve minimal loss on the error function.

3.1 Data Preprocessing

The data pre-processing steps will utilise Mel-spectrogram images for input into the CNN [19] [21]. Spectrograms are generated using a 25 ms Hamming window and Short-term Fast Fourier Transform (STFFT) with a 10 ms hop length and log amplitude with the python librosa audio library, similar to the procedures used in [21], [12], [13] and [14]. For Mel-spectrograms, the image is generated by applying 40 Mel-scaled filter banks to the STFFT data before rendering the image [21]. Finally, the spectrogram is rendered to an image file in 3-channel RGB color mode with "jet" color heatmap using matplotlib and python.

Figure 3: Mel-spectrogram of speech utterance '08a04La.wav'



Female speaker saying the phrase "Tonight I could tell him" in the acted emotion of 'Boredom'.

3.1.1 Data Augmentation

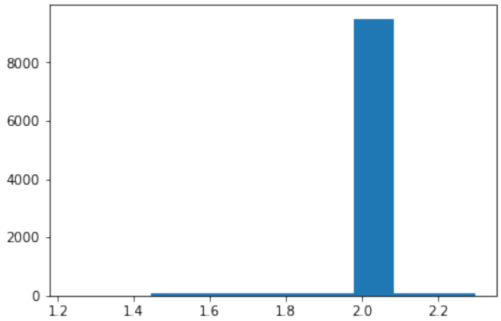
The Emo-DB dataset contains 535 audio samples. Deep Learning typically requires several thousand samples to train a model, otherwise the model is prone to overfitting due to the large number of weights in a typical network. Without an adequate number of training samples, a DNN learns the patterns within the training data but cannot generalise well to unseen data. To mitigate this problem, some form of data augmentation is required to boost the number of samples used for training, validation and testing. The benchmark papers by [12] and [13] mention the use of training sets larger than the 535 samples distributed with Emo-DB, but do not explain how they arrive at such training set sizes. However, Weiskirchen et al. ([14]) discusses the augmentation technique of slicing up the audio samples and generating spectrograms from the sub-samples. In addition, multiple spectrograms per sub-sample were generated by limiting the upper frequency limit of the spectrogram to a range of values such as 7000, 7500 and 8000 Hz.

In this paper, two different data augmentation techniques have been used to generate additional data by segmenting the speech samples: 1) segmentation of speech samples by dividing them at fixed intervals and 2) slicing up the speech samples by repeatedly sliding a fixed-length 'window' by a given step size over the entire utterance. Although the sub-sampling procedure varies slightly from that described in [14], the same frequency range limits of 7000, 7500 and 8000 Hz are used in this paper for both augmentation techniques. During the implementation phase of the project, it was found experimentally that the 'fixed interval' segmentation datasets performed poorly relative to the ‘sliding window’ segmentation datasets. For this reason, the focus of this paper is concerned primarily with the latter augmentation technique.

3.1.2 Sliding Window Segmentation

This technique involves use of a sliding window over the sample, where the window size (not to be confused with the STFFT or spectrogram window size) represents the number of contiguous frames taken from an audio sample to create a sub-sample. The window is then shifted by a given number of frames (the step size) and another sub-sample is taken. In this way, many sub-samples are created in order to increase the data samples available to train the system. For this paper, three different sliding window sizes were tested: 1.5, 2.0, and 2.6 seconds, which resemble the sizes used in the benchmark papers [21], [13], [14].

Figure 4: Example histogram of Emo-DB ‘sliding window’ segmentation using a 2 second sliding window size with a 10 frame step size



The advantage to the sliding window approach is that all sub-samples are the same length, however a disadvantage is that many of the sub-samples represent highly redundant data. Another advantage is that sliding window produces many more sub-samples, particularly if a small step size is selected and/or small window size is selected. On the other hand, a small window size may not be large enough to adequately capture salient speech features.

3.2 Implementation

The following software libraries and toolkits were used to complete the project:

- Python 3.5
- Keras
- TensorFlow
- Scikit-Learn
- Jupyter notebooks

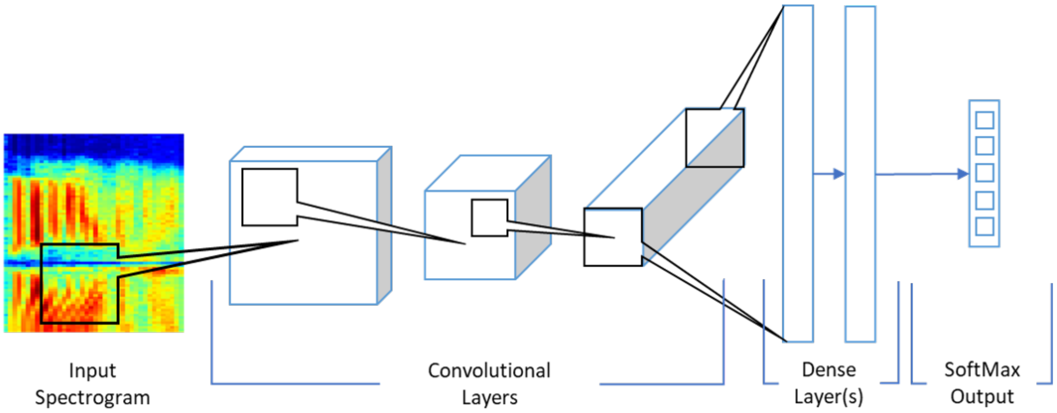
Spectrogram data for the project was generated on a personal computer, whereas model training and evaluation was executed on cloud-based servers due to the computational overhead involved in training a CNN. Audio files were converted to spectrogram images using python and the librosa audio processing package with Jupyter notebooks in an Anaconda virtual python environment. The resulting spectrogram datasets were then uploaded to Google CoLab or FloydHub to be used for training CNN models using GPU resources in the cloud. Python code for training CNN models on the datasets was executed in the cloud using Jupyter notebooks.

Spectrograms were input as numerical tensors into the training algorithms after being read from image files. As is typical for a multi-class problem such as SER, categorical labels for the spectrograms were one-hot encoded as part of the data processing steps before training any models.

3.2.1 CNN Implementation

The CNNs were implemented using the Keras library with a TensorFlow backend. The CNN models tested in this paper all consisted of at least one or more convolutional layers, followed by one or more fully-connected layers and a final output layer corresponding to the number of output classes represented in the dataset (see figure 5). The ReLU activation function was used on all layers except the final output layer which used the SoftMax activation function. All models were trained using Keras' Adam optimiser with default settings for learning rate and decay rate. Training used the categorical cross-entropy loss function as is appropriate for a multi-class classifier, with validation accuracy as the training evaluation metric. Training utilised Keras' callback functionality for saving model checkpoints (when validation accuracy had improved during training, helping to avoid using an overtrained model) and early-stopping. Early-stopping was used during the initial and intermediate experiments to assist with quickly iterating through various models without incurring excessive training overhead for poorly-performing models.

Figure 5: Overview of proposed Convolutional Network



Spectrogram images are input to a series of convolutional layers, optionally interspersed with other layers such as pooling or dropout layers. Each convolutional layer performs the role of a feature extractor. Feature maps are flattened before being passed into dense (fully-connected) layers. The final layer is a softmax output layer consisting of as many nodes as output classes. The final output is a list of probabilities that the input corresponds to each output class.

3.2.2 Data Splits

Preliminary models were trained and tested using a 66/33% train/test split, with 15% of the training data used for validation. Intermediate models were typically evaluated using a speaker-independent LOSO-like data split modelled after the OpenXBOW-style train/validation/test split [25]. Speaker-independence refers to models trained on one set of human speakers and tested on a different set of speakers. This data split (hereafter refered to as the 'XBOW' split) uses all speakers with an ID greater than 10 for training and the remainder (speaker IDs of 10 or below) for testing [25]. In addition, the author reserved 33% of the XBOW training data for validation during training. The choice of a larger validation data split for the development of intermediate models was due to the inherent difficulties in obtaining a good generalised model on speaker-independent data. Speaker-independence is a fundamentally challenge with SER as different speakers can exhibit highly variable characteristics when displaying emotion, making generalisation difficult. However, speaker-indpendence is a common metric for success in SER tasks [28], [29].

3.3 Refinement

The refinement phase was an iterative process of experimentation using various CNN architectures trained on a variety of resampled (data augmented) spectrogram datasets. Two different preliminary datasets were trialed; one using fixed-interval segmentation and the other using sliding window segmentation. This stage helped to verify the suitability of the augmented data techniques and to establish a workflow and data processing pipeline for training a variety of CNN models.

3.3.1 Initial Solutions

The model architecture initially tried was a simple 2-layer CNN. The first convolutional layer consisted of 16 filters with 5x5 kernel and a stride 3 using 'same' padding. Layer 1 was followed by a MaxPooling layer with 2x2 filter and stride 2. Convolutional layer 2 consisted of 32 filters with 3x3 kernel, a stride of 2, and 'same' padding. Next, 2 fully-connected layers of 512 nodes each and a final SoftMax output layer with 7 nodes (corresponding to the number of output classes). This network was trained for 25 epochs with mini-batch size of 64, using a 33% test split and 15% validation set for speaker-dependent evaluation. The initial models were tested using a simple speaker-dependent train/validation/test split in order to quickly evaluate relative performance and to reject poor performers early on.

Using the initial results as a guide, it was decided that the sliding window segmentation technique yielded better results and two additional datasets were created using 1.5 and 2.6 second segmentation windows. Upon further experimentation, the 1.5 second segmented data was found to perform worse than the others so was omitted from further experimentation. No additional variants of fixed interval segmentation datasets were tested.

Table 1: Initial results of speaker-dependent models trained on segmented and sliding window datasets.

WxH	Segmentation Type	# Samples	Train WA (%)	Val WA (%)	Test WA (%)
179x174	2 sec seg	2247	99.92	94.25	92.72
256x128	2 sec slide, 10 frame step	14460	99.91	96.08	97.35

Two different preliminary datasets were trialed, one using fixed-interval segmentation and the other using sliding window segmentation. The image dimensions are displayed in the WxH column, and training, validation (Val) and testing accuracies (WA) are shown.

3.3.2 Intermediate Solutions

Following on from the initial stage of implementation, various CNN architectures were tried and tested, by replicating the CNN models described in the benchmarks and literature. Candidate architectures and hyperparameters were subsequently tested on a speaker-independent XBOW data split, which has an established LOSO-like benchmark performance. This was done to quickly iterate and test various models without the computational overhead of full-blown k-fold LOSO. These models did not yield results similar to those of the benchmark CNN models.

Table 2: Intermediate Results

The author's model is structured as follows: convolutional layer 1 consists of 32 filters with 5x5 kernel and a stride 3 using 'same' padding. Convolutional layer 2 consists of 64 filters with 3x3 kernel, a stride of 2, and 'same' padding. Layer 2 is followed by a MaxPooling layer with 2x2 filter and stride 2. Next, 2 fully-connected layers of 512 nodes each and a final SoftMax output layer with 7 nodes.

Model	Type	# Samples	Train WA (%)	Val WA (%)	Test WA (%)	Test UAR (%)
Author	Sliding	14682	99.75	98.59	69.40	65.76
Author	Segment	2247	98.46	94.39	61.36	59.13
AlexNet	Sliding	14682	98.73	99.45	61.90	55.91
Badshah et al.*	Sliding	8859	97.64	99.05	68.50	56.80

*With the exception of Badshah, all results are based on a 2 second segmentation (fixed interval or windowed). In the case of Badshah, the segment length is 2.6 seconds.

The results obtained with the fixed interval segmented data continued to perform poorly relative to sliding window segmented data, therfore the remainder of the experiments utilises only data augmented with the latter technique. Based on results achieved in table 2, the 2.6 second sliding windows dataset was chosen for further experimentation. The reasoning and justification behind this choice is that it contains more temporal data and therefore should capture more prosodic and emotional features. In addition, the 2.6 second dataset container fewer samples with less feature redundancy than the 2.0 second dataset, making it suitable for faster experimentation and iteration while still having a sufficient training size for DNN and CNN.

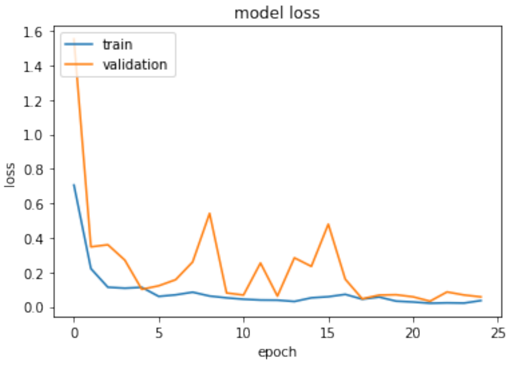
Final Solution As seen previously in table 2, AlexNet (as used by [13] and [14]) performed very poorly on the 2.0 second dataset, so no further training was conducted on the 2.6 dataset. This decision was made based on the length of training time required by such a complex network as AlexNet, and the observation that it did not produce results above that of the author's simpler network. The CNN model proposed by Badshah et al. [12] was simpler than AlexNet but it also performed poorly. As a smaller network that was seen as having intermediate complexity between the author's initial network and that of AlexNet, the 2.6 dataset was tested against the model and variants thereof. However, the performance of those was worse than the simple initial network so Badshah's architecture was not pursued further.

Table 3: Final Candidate Models

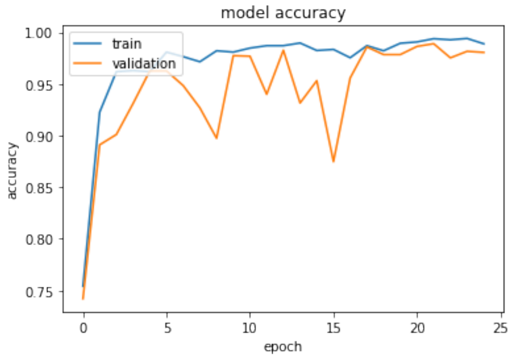
Model	Secs.	# Samples	Epochs	Train WA (%)	Val WA (%)	Test WA (%)	Test UAR (%)
CV(32 x 5 x 5) - CV(64 x 3 x 3) - FC(716)x2	2.0	14682	25	99.40	98.84	75.14	66.30
CV(64 x 5 x 5) - CV(128 x 3 x 3) - CV(256 x 3 x 3) - FC(4096)x2	2.6	8859	40	99.12	99.74	75.04	69.91
CV(64 x 5 x 5) - CV(128 x 3 x 3) - CV(256 x 3 x 3) - FC(4096)x2	2.0	14682	50	99.69	99.69	72.38	67.37
CV(64 x 5 x 5) - CV(128 x 3 x 3) - FC(1024)x2	2.6	8859	25	98.70	98.58	**75.75**	**71.31**

The CNN architecture proposed by Fayek et al. [21] was very similar to the model created by the author 2 that had yielded better performance than any of other models thus far. Fayek et al. introduced batch normalisation layers after each convolutional layer, as well as dropout layers between the fully-connected (FC) layers. It was found that the author's model performed much better when batch normalisation was added, even without the use of dropout layers. Adding batch normalisation layers helped to improve the accuracy (WA) and UAR scores by an additional 4-5% on models evaluated using the XBOW data split. Further iterations using dropout layers in between the FC layers, and increasing the number of filters and FC nodes helped to make small improvements to these results. The best-performing model achieved WA

Figure 6: Learning Curves for final model trained on XBOW LOSO data split



(a) Model Validation Loss



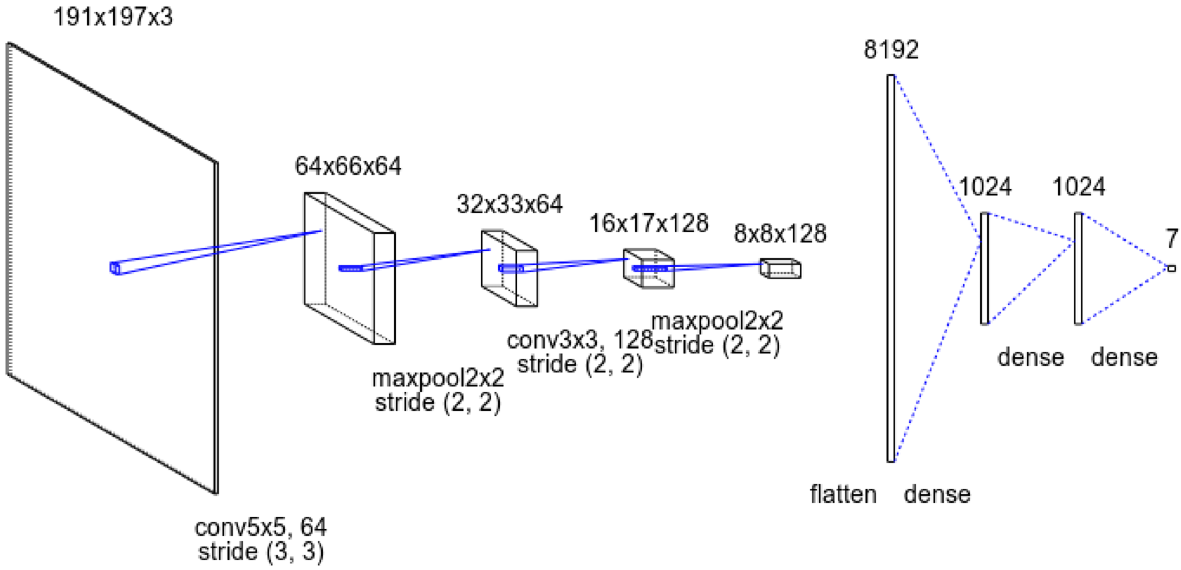
(b) Model Validation Accuracy

Results of *CV(64 x 5 x 5) - CV(128 x 3 x 3) - FC(1024)x2* model trained on 2.6 second sliding window dataset, evaluated using XBOW data split

and UAR scores of 75.75% and 71.31% respectively. The learning curves obtained for that model are shown in figure 6. The affect of the batch normalisation (batchnorm) and dropout layers is visible in the curves when the loss increases sharply (and accuracy drops); these are likely the epochs when the model's weights are adjusted drastically by batchnorm and dropout. This is in contrast to the learning curves observed without batchnorm that had smoother curves without the prominent spikes.

The final model is structured as follows (see figure 7): convolutional layer 1 consists of 64 filters with 5x5 kernel and a stride 3 using 'same' padding. Layer 1 is followed by a BatchNormalisation layer and MaxPooling layer with 2x2 filter and stride 2. Convolutional layer 2 consists of 128 filters with 3x3 kernel, a stride of 2, and 'same' padding. Layer 2 is followed by a BatchNormalisation layer and a MaxPooling layer with 2x2 filter and stride 2. Next, 2 fully-connected layers of 1024 nodes each interspersed with Dropout Layers having 20% dropout probability. The FC+Dropout layers are completed with a SoftMax output layer having 7 nodes. The best performing model was trained without early-stopping for 25 epochs.

Figure 7: Final CNN model



4 Results

The use of batchnorm and dropout in the final model were beneficial to avoid overfitting, and allowed the use of additional filters and FC nodes to the model. Prior to using batchnorm with dropout, additional filters and/or FC nodes with dropout contributed to increased complexity and training times without a corresponding increase in the metric scores. Batchnorm appears to have been the right kind of normalisation technique needed for this model to significantly improve the results, pushing the XBOW-based evaluation above 70% for both WA and UAR metrics.

4.1 Model Evaluation and Validation

For evaluation purposes, a Leave-One-Speaker-Out (LOSO) speaker-independent k-fold cross-validation strategy will be applied. Each training fold will consist of n-1 speakers with the validation fold containing samples from a single speaker, such that no training fold will contain samples of the speaker contained within the test fold. As there are 10 speakers in the Emo-DB speech corpus, the resulting evaluation will be a 10-fold LOSO cross-validation. This will ensure that the model generalises to a speaker-independent test set, as is typical for SER tasks. For the purposes of this project, the UAR result of 71% achieved by Weiskirchen et al. will be considered the primary metric for comparing results as the 10-fold cross-validation LOSO evaluation is most similar to the evaluation approach used in this paper.

4.2 Justification

One clear observation made during the implementation phase, and borne out by the results is that for the spectrogram datasets used in this project, the choice of kernel size (feature map) has a significant impact on the results. Smaller kernel sizes (i.e. 5x5, 3x3) proved to be much better than larger filters particularly in the first convolutional layer, which is in contrast to the models exhibited in the literature where 10x10 and 11x11 kernel sizes are frequently used.

Table 4: Final Results

Model	Evaluation	Epochs	Test WA (%)	Test UAR (%)
CV(64 x 5 x 5) - CV(128 x 3 x 3) - FC(1024)x2	XBOW	25	75.75	71.31
CV(64 x 5 x 5) - CV(128 x 3 x 3) - FC(1024)x2	5-fold (non sp-ind)	25	97.84	96.72
CV(64 x 5 x 5) - CV(128 x 3 x 3) - FC(1024)x2	LOSO	25	**71.19**	61.68
CV(64 x 5 x 5) - CV(128 x 3 x 3) - FC(1024)x2	LOSO	50	70.75	**63.19**
BoAW [25]	XBOW		81.40	75.11
Badshah et al. [12]	5-fold (non sp-ind)		84.30	-
Stolar et al. [13](female)	LOSO (female)		79.68	-
Stolar et al. [13](male)	LOSO (male)		76.79	-
Weiskirchen et al. [14]	LOSO		-	71.00

Judging by the results obtained using the benchmark architectures (table 2) and the low scores achieved relative to the benchmarks, it is likely that the spectrogram images used in this project are significantly dissimilar to those used in the benchmark papers. The spectrograms may not be generated using the exact same parameters as in the benchmark papers, or the spectrograms don't quite capture the same audio qualities. There are many libraries available for audio-processing and the options for spectrogram-related parameters aren't well described in the referenced papers making it difficult to accurately replicate the spectrogram data. In contrast, the CNN architectures used are well-documented therefore replicating the CNN models is relatively easier.

A comparison to Badshah et al. shown in table 4 demonstrates that the author's model tested using a 5-fold speaker-dependent cross-validation evaluation achieves a WA score of 97.84%, a result that is far above that achieved by Badshah et al., while using a much less complicated CNN network. The author's model also achieves WA and UAR scores of 70.75% and 63.19% respectively on a full 10-fold LOSO evaluation. This is still lower than the UAR of 71.00% achieved by Weiskirchen et al. ([14]), but has been achieved with a much simpler network with fewer parameters. Similarly, the XBOW evaluation for the author's model achieves WA and UAR scores of 75.75% and 71.31% respectively compared to 81.40% and 75.11% for BoAW. The UAR delta here is much smaller at 3.8% as opposed to 7.81% in the case of Weiskirchen et al, and the XBOW evaluation is still a LOSO-like validation of the model's speaker-independence and generalisation ability across speakers and gender.

Table 5: Classification report for XBOW LOSO-evaluated final model

	Precision	**Recall**	**f1-score**	**Support**
Anger	0.92	0.73	0.81	591
Boredom	0.64	0.67	0.66	450
Disgust	0.70	0.71	0.70	285
Fear	0.40	0.75	0.52	102
Happy	0.38	0.64	0.48	183
Neutral	0.65	0.63	0.64	282
Sad	0.95	0.86	0.90	1209
Avg/ Total	**0.80**	**0.76**	**0.77**	**3102**

5 Conclusion

5.0.1 Summary

In this project, the author has tested and evaluated a number of different CNN architectures trained on spectrograms extracted from the Emo-DB speech corpus. A best-effort has been made to faithfully replicate the results of benchmark models found in the literature and the scores, although mixed, are close to the benchamrks while having been achieved using simpler and less complex architectures. Furthermore, the final CNN model selected is similar to that of Fayek et al. ([21]), demonstrating that a similar CNN architecture can be applied to different speech corpora while still achieving reasonable results.

5.1 Free-Form Visualisation

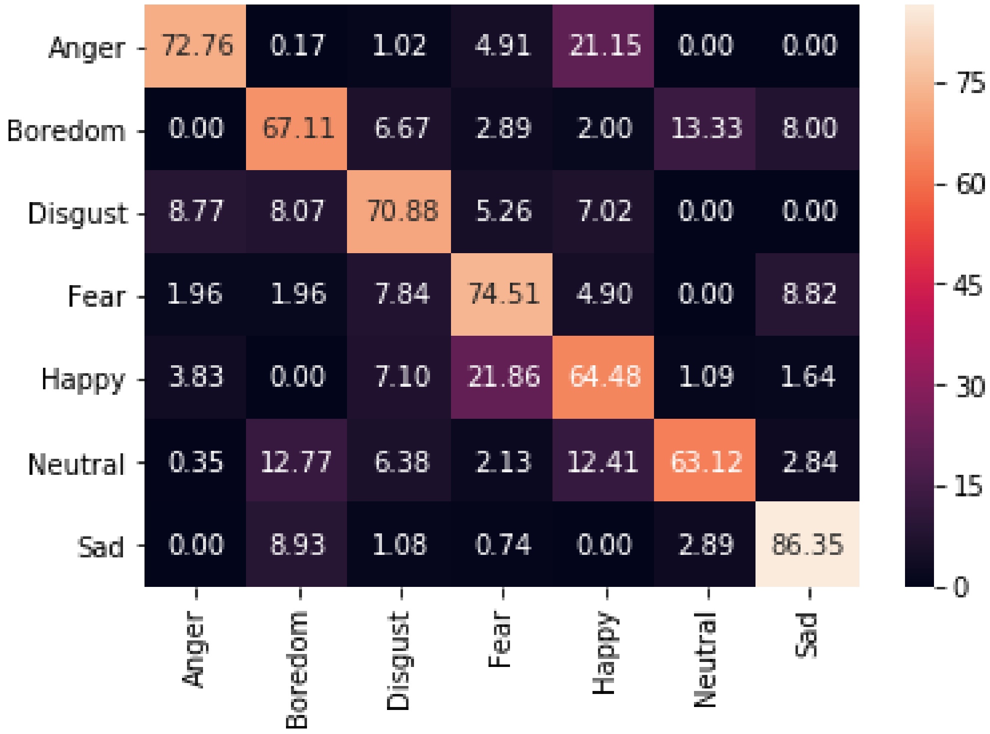
Badshah et al. 2017 ([12]) utilise transfer learning against an AlexNet CNN model and train it on Emo-DB. Although Badshah et al. do not provide a LOSO-based evaluation or UAR value in their results, it's possbile to make a comparison using confusion matrices. As can be seen by comparing the results in figures 8 and 9, the per-class true-positives achieved by this project are an improvement over those of Badshah et al., with the exception of the 'anger' and 'sad' emotions.

5.2 Reflection

Overall, it appears that data preparation is paramount as it is clear that despite the relative ease of replicating CNN architectures and the readily available documentation in the literature describing model architecture and parameters, the results of this paper are very different to that of the benchmark models. The most likely difference in approach is that of the data

Figure 8: Confusion Matrix (percent correct) for EMO-DB sliding 2-sec window lengths/durations (w/ 10-frame stride)

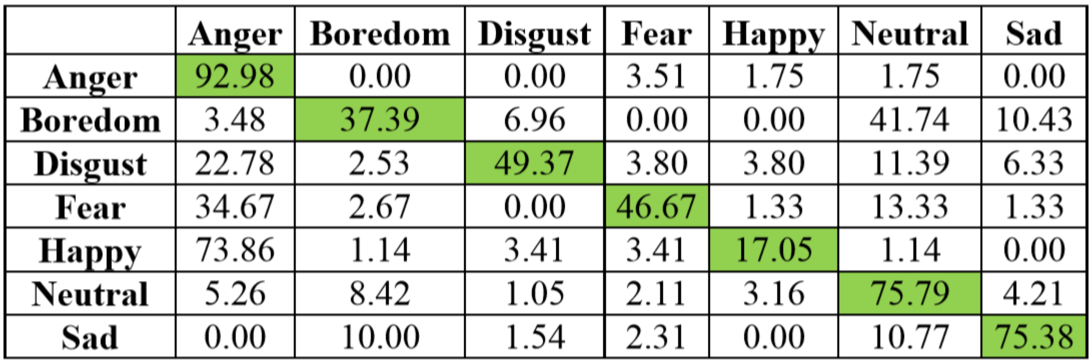
The x-axis is the Predicted values, and the y-axis is the Actual values.



	Anger	Boredom	Disgust	Fear	Happy	Neutral	Sad
Anger	72.76	0.17	1.02	4.91	21.15	0.00	0.00
Boredom	0.00	67.11	6.67	2.89	2.00	13.33	8.00
Disgust	8.77	8.07	70.88	5.26	7.02	0.00	0.00
Fear	1.96	1.96	7.84	74.51	4.90	0.00	8.82
Happy	3.83	0.00	7.10	21.86	64.48	1.09	1.64
Neutral	0.35	12.77	6.38	2.13	12.41	63.12	2.84
Sad	0.00	8.93	1.08	0.74	0.00	2.89	86.35

Figure 9: Confusion Matrix from AlexNet-based model (from Badshah et al. 2017 ([12])

The x-axis is the Predicted values, and the y-axis is the Actual values.



	Anger	Boredom	Disgust	Fear	Happy	Neutral	Sad
Anger	92.98	0.00	0.00	3.51	1.75	1.75	0.00
Boredom	3.48	37.39	6.96	0.00	0.00	41.74	10.43
Disgust	22.78	2.53	49.37	3.80	3.80	11.39	6.33
Fear	34.67	2.67	0.00	46.67	1.33	13.33	1.33
Happy	73.86	1.14	3.41	3.41	17.05	1.14	0.00
Neutral	5.26	8.42	1.05	2.11	3.16	75.79	4.21
Sad	0.00	10.00	1.54	2.31	0.00	10.77	75.38

preparation step, more specifically that of the creation of the spectrograms. In this paper, the use of data augmentation and spectrogram creation is a best effort at replicating the work done in the literature, but unlike the CNN and model arch, the data processing steps are not easily discerned or replicated. Indeed, the descriptions are sometimes inadequate or glossed over in the literature. It is the opinion of the author that the data processing is the single largest factor in the attempts taken to replicate the results and successes of the benchmark's authors.

5.3 Improvement

5.3.1 Silence: Labeling or Removal

After segmentation, many of the spectrograms are made up of predominantly silent frames. All segments that are derived from a given utterance and therefore have the same emotion label as the 'parent' utterance. Segments containing predominantly silent frames don't capture salient prosodic details. This could potentially skew the learner, teaching it that silent frames are associated with certain emotions. The highly acted and artificial nature of the audio samples in Emo-DB means that]qsad samples are often punctuated by long pauses (silence) between vocalisations. In contrast, excited emotions like anger and happiness exhibit high activations and very few silent frames relative to other emotions such as sadness or boredom. The end result could be that silence is mislabeled as sadness. Since sadness is one of the emotions that is most easily detected in audio-only SER, it's possible that the learner's accuracy on certain emotions is less accurate than could otherwise be if silent frames were distinguished.

5.3.2 Transfer Learning

Many of the benchmark models discussed in the literature use an AlexNet-like model architecture, and in one case ([13]) use a pre-trained AlexNet model with Transfer Learning. A future development or experiment could be to use a pre-trained AlexNet model and transfer learning to learn the Emo-DB dataset rather than try to learn it from scratch. This would also provide a good comparison against the literature.

References

[1] S. Mariooryad and C. Busso, "Exploring Cross-Modality Affective Reactions for Audiovisual Emotion Recognition," *IEEE Transactions on Affective Computing*, vol. 4, no. 2, pp. 183–196, Apr. 2013.

[2] T. Vogt, E. André, and J. Wagner, "Automatic Recognition of Emotions from Speech: A Review of the Literature and Recommendations for Practical Realisation," C. Peter and R. Beale, Eds. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 75–91. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-85099-1_7

[3] M. Tahon and L. Devillers, "Towards a Small Set of Robust Acoustic Features for Emotion Recognition: Challenges," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 1, pp. 16–28, Jan. 2016.

[4] S. Poria, E. Cambria, R. Bajpai, and A. Hussain, "A review of affective computing: From unimodal analysis to multimodal fusion," *Information Fusion*, vol. 37, pp. 98–125, Sep. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1566253517300738

[5] A. Hong, Y. Tsuboi, G. Nejat, and B. Benhabib, "Affective Voice Recognition of Older Adults1," *Journal of Medical Devices*, vol. 10, no. 2, pp. 020 931–020 931–2, May 2016. [Online]. Available: http://dx.doi.org/10.1115/1.4033226

[6] G. Acampora, D. J. Cook, P. Rashidi, and A. V. Vasilakos, "A Survey on Ambient Intelligence in Healthcare," *Proceedings of the IEEE*, vol. 101, no. 12, pp. 2470–2494, Dec. 2013.

[7] P. Rashidi and A. Mihailidis, "A Survey on Ambient-Assisted Living Tools for Older Adults," *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 3, pp. 579–590, May 2013.

[8] K. Johnson, "Amazon's Alexa wants to learn more about your feelings," Dec. 2017. [Online]. Available: https://venturebeat.com/2017/12/22/amazons-alexa-wants-to-learn-more-about-your-feelings/

[9] H. Pérez-Espinosa, J. Martínez-Miranda, I. Espinosa-Curiel, J. Rodríguez-Jacobo, and H. Avila-George, "Using acoustic paralinguistic information to assess the interaction quality in speech-based systems for elderly users," *International Journal of Human-Computer Studies*, vol. 98, pp. 1–13, Feb. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S107158191630129X

[10] c. Wikipedia, *Emotion classification — Wikipedia, The Free Encyclopedia*, 2017. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Emotion_classification&oldid=802307984

[11] P. Ekman, "An argument for basic emotions," *Cognition & emotion*, vol. 6, no. 3-4, pp. 169–200, 1992.

[12] A. M. Badshah, J. Ahmad, N. Rahim, and S. W. Baik, "Speech Emotion Recognition from Spectrograms with Deep Convolutional Neural Network," in *2017 International Conference on Platform Technology and Service (PlatCon)*, Feb. 2017, pp. 1–5.

[13] M. N. Stolar, M. Lech, R. S. Bolia, and M. Skinner, "Real time speech emotion recognition using RGB image classification and transfer learning," in *2017 11th International Conference on Signal Processing and Communication Systems (ICSPCS)*, Dec. 2017, pp. 1–8.

[14] N. Weißkirchen, R. Böck, and A. Wendemuth, "Recognition of Emotional Speech with Convolutional Neural Networks by Means of Spectral Estimates," Oct. 2017.

[15] N. Cummins, S. Amiriparian, G. Hagerer, A. Batliner, S. Steidl, and B. Schuller, "An Image-based Deep Spectrum Feature Representation for the Recognition of Emotional Speech," in *Proc. of the 25th ACM International Conference on Multimedia, MM*, 2017.

[16] F. Burkhardt, A. Paeschke, M. Rolfes, W. Sendlmeier, and B. Weiss, "A database of german emotional speech," in *in Proceedings of Interspeech, Lissabon*, 2005, pp. 1517–1520.

[17] C.-n. Anagnostopoulos, T. Iliou, and I. Giannoukos, "Features and classifiers for emotion recognition from speech: a survey from 2000 to 2011," *The Artificial Intelligence Review; Dordrecht*, vol. 43, no. 2, pp. 155–177, Nov. 2012. [Online]. Available: http://search.proquest.com.proxy.library.dmu.ac.uk/docview/1647316503/abstract/19FE966497E74B1BPQ/1

[18] B. Schuller, S. Steidl, A. Batliner, F. Burkhardt, L. Devillers, C. Müller, and S. Narayanan, "Paralinguistics in speech and language—State-of-the-art and the challenge," *Computer Speech & Language*, vol. 27, no. 1, pp. 4–39, Jan. 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0885230812000162

[19] H. Fayek, "Speech Processing for Machine Learning: Filter banks, Mel-Frequency Cepstral Coefficients (MFCCs) and What's In-Between," Apr. 2016. [Online]. Available: http://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html

[20] Wikipedia, "Spectrogram," Mar. 2018, page Version ID: 830276272. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Spectrogram&oldid=830276272

[21] H. M. Fayek, M. Lech, and L. Cavedon, "Evaluating deep learning architectures for Speech Emotion Recognition," *Neural Networks*, Mar. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S089360801730059X

[22] A. Satt, S. Rozenberg, and R. Hoory, “Efficient Emotion Recognition from Speech Using Deep Learning on Spectrograms,” 2017.

[23] M. Schmitt, F. Ringeval, and B. Schuller, “At the Border of Acoustics and Linguistics: Bag-of-Audio-Words for the Recognition of Emotions in Speech,” Sep. 2016, pp. 495–499. [Online]. Available: http://www.isca-speech.org/archive/Interspeech_2016/abstracts/1124.html

[24] M. Schmitt and B. W. Schuller, “openXBOW - Introducing the Passau Open-Source Crossmodal Bag-of-Words Toolkit,” *arXiv:1605.06778 [cs]*, May 2016, arXiv: 1605.06778. [Online]. Available: http://arxiv.org/abs/1605.06778

[25] openXBOW, “openXBOW: the Passau Open-Source Crossmodal Bag-of-Words Toolkit,” Jan. 2018, original-date: 2016-05-21T23:32:39Z. [Online]. Available: https://github.com/openXBOW/openXBOW

[26] Z. Huang, M. Dong, Q. Mao, and Y. Zhan, “Speech Emotion Recognition Using CNN,” in *Proceedings of the 22Nd ACM International Conference on Multimedia*, ser. MM ’14. New York, NY, USA: ACM, 2014, pp. 801–804. [Online]. Available: http://doi.acm.org/10.1145/2647868.2654984

[27] W. Lim, D. Jang, and T. Lee, “Speech emotion recognition using convolutional and Recurrent Neural Networks,” in *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, Dec. 2016, pp. 1–4.

[28] B. Schuller, S. Reiter, R. Muller, M. Al-Hames, M. Lang, and G. Rigoll, “Speaker Independent Speech Emotion Recognition by Ensemble Classification,” in *2005 IEEE International Conference on Multimedia and Expo*, Jul. 2005, pp. 864–867.

[29] J. Chang and S. Scherer, “Learning Representations of Emotional Speech with Deep Convolutional Generative Adversarial Networks,” *arXiv:1705.02394 [cs, stat]*, Apr. 2017, arXiv: 1705.02394. [Online]. Available: http://arxiv.org/abs/1705.02394