Emotional Voice Conversion using multiple Deep Learning Methods

Final Report

Ying-Lun Cheng

Supervisor: Prof Kit Wong

Second Assessor: Dr Kenneth Tong

April 2021

DECLARATION

I have read and understood the College and Department's statements and guidelines concerning plagiarism.

I declare that all material described in this report is all my own work except where explicitly and individually indicated in the text. This includes ideas described in the text, figures and computer programs.

Name: Ying-Lun Cheng ……

Signature: Ying-Lun Cheng ……

Date: 14/April/2021 ……

Abstract

Emotional voice conversion is a technique that converts the emotional features of a given speech, while preserving the speaker's identity and the linguistic content. Voice conversion has vast potential in various fields, such as style transferring in audio-related content creations, serving as medical aids for people who have difficulties in voicing, etc. Most of the previous studies deal with voice conversion without focusing on specific features such as speech emotions. Therefore, this project aims to carry out a voice-conversion study that only emphasises converting speech emotion. In this project, two methods are proposed so as to deal with the subject matter under investigation, including convolutional neural networks (CNNs) and cycle-consistent generative adversarial networks (CycleGAN). The results of CNNs score high marks by human evaluation when trained by parallel data while the performance of CycleGAN are highly proportional to the training time provided. Then, objective and subjective evaluations are carried out. The final results indicate that performing voice conversion to recognise emotion is feasible.

Chapter 1: Introduction

The progress of human civilisation can be attributed to the fact that humans can communicate effectively. One distinctive feature of human interaction is emotion. Without needing to involve text or speech, emotion itself can transmit intended messages. For example, people with language barriers might still be able to communicate via physical or verbal emotional expressions. Moreover, delivering text or speech with different emotional states might dramatically change its inherent meaning. According to the arguments mentioned above, analysing human expressions of emotion is a vital field of study, yet past limitations in computing power and the complexity of the details inherent in emotions had made relevant research impractical.

Restrictions have been gradually eliminated as of late, with various emotion-related studies having already contributed to changes in lifestyles. Before emotion was introduced, the production of synthesised speech was not appealing at all. Metallic sounds came off as uncomfortable and hostile, despite being originally made by humans. These days, however, everyday life is flooded with synthesised speech due to the advent of virtual assistants. These speech synthesisers, such as Google Assistant or Siri, include emotional features which have made their speech samples significantly more natural and acceptable. One approach to utilising an emotional speech synthesiser is labelling the text with its corresponding emotion [1]. When needed, the machine can synthesise speech with predefined emotional features. Furthermore, there are studies that attempt to perform speech emotion recognition (SER). Performing SER requires a full understanding of emotional features, since the quality of the feature extraction directly affects the recognition results. The research carried out in [2] was a build of a SER model with a high accuracy using a machine learning algorithm such as Support Vector Machine (SVM) and Convolutional Neural Network (CNN). Reaching over 80% accuracy, the author raises a noteworthy point regarding the misclassified audio. A “calm” audio is predicted as a “neutral” emotion which caused an error, but the difference is so subtle that even humans have difficulties in classifying the audio. This reminds one to inspect the database before applying any sort of research, as errors might come from the dataset, rather than the model or the algorithm.

Another relevant study is emotional voice conversion (EVC). Similar to the traditional voice conversion (VC) technique, the goal is to modify the acoustic features while preserving the linguistic content of a speech. VC can be seen as approximating a mapping function between the source and target audio [3]. Past studies have successfully performed VC using different approaches such as the restricted Boltzmann machine (RBM) [4], the Gaussian mixture model (GMM) [5], the CNN [6], and the generative adversarial networks (GANs) [3] [7]. However, most of these research

projects focus on converting the speakers' identities, while our goal is to convert the emotions only.

This project aims to build a model that converts the emotional features of a given speech. The speaker identity and the content should both remain unchanged. I expect to achieve the following:

- Perform EVC: The main goal of this project is to convert the emotion and the emotion only of a speech.
- Attempt multiple methods: Perform EVC using at least two methods.
- Have a deeper understanding of deep learning: Solidify skills in performing deep learning.
- Build audio analysing skills: Being able to extract audio features and interpret the meanings.

After going through numerous articles and projects online, I decided that I should first perform a simpler version of EVC using a CNN model, and then build another model based on GANs if the CNN model is a success.

To perform deep learning algorithm on audio files, it is common that we transform the audio files into spectrograms or extract additional features to create input data for training the model. The raw data format of the audio files appears simply as magnitudes along the time axis which provide very little information [8]. The first approach that was chosen for execution uses the spectrogram as the input data, which is similar to [8] and [9]. In research [8], the spectrogram and the fundamental frequency are extracted as the input training data, and the model they build is based on CycleGAN architecture. The aim and objective of [9] is not related to VC, but the methodology they present is helpful for spectrogram conversion. Their project is about converting the style of images using CNN, and my goal is to perform EVC by a CNN model. Demonstrations that they provide clearly show that they are capable of converting a photograph into the painting style of Vincent Van Gogh, which will be useful since converting spectrogram style is the same as converting photograph style.

The second approach I have chosen will be to use models related to GANs due to the robust results shown in the studies. GANs is a machine learning class that includes a generative network and a discriminative network. The generative network generates output data while the discriminative network evaluates the similarity between the output data and the true data. The goal of a GAN is to find a mapping algorithm producing outputs that can fool the discriminator. There are several GAN-based models, namely CycleGAN [7] [8], VAW-GAN [10], and StarGAN [11]. Among all the GAN-related research I read through, most of it mentioned CycleGAN as the fundamental GAN model while other GAN models are developed based on CycleGAN. Therefore, I set my research direction to CycleGAN. To reduce the computational demand, the

CycleGAN model we use waives the RNN structure and replaces it with gated CNNs. The main issue of GANs is that they produce outputs regardless of the input data, but we want to keep the content features of a given speech and convert the emotional features only. Therefore, two sets of generators and discriminators are included in CycleGANs [8]. While evaluating the quality of the output data, the network is also assessing the similarity between the output data and the original inputs so that some features such as the speech content can be preserved. To reduce the computational demand, the CycleGAN model we use waived the RNN structure and replaced it by gated CNNs.

Most of the models mentioned above focus on converting the speaker identities, but my goal will be converting the emotion of speech while preserving the identities of the speakers. Moreover, the mean opinion score (MOS) of the results mentioned above do not seem too satisfying, which points out that the converted audios in those researches are not too satisfying to human listeners. Hence, improving the quality of the converted speech is also one of our aims.

Chapter 2: Goals and Objective

According to the completed research and the given studying time, I aim to create two models that perform EVC, the CNN model and the CycleGAN model. Since an earlier study showed that the quality of the dataset and the subtle difference between emotional features might cause unexpected errors when performing EVC, I decided to strictly gauge the quality of the dataset and start the experiment using emotions with noticeable differences, which should increase my success rate. Furthermore, I will tune the parameters inside the models to find the optimal combination.

There are a lot of applications for EVC. It could enhance the user experience of text-to-speech systems by adding emotional features to the synthesised speech. The technology could aid people with difficulties in voicing, helping people who might have damages to their vocal cord. EVC might also be helpful in telecommunications, intensifying the soft voice coming out from the speaker to ensure the user experience of the listener.

This is a software modelling and design project. All the algorithms will be run on python with GPU acceleration. A new hardware acceleration option called TPU has been released recently and should be much more powerful than GPU acceleration. Yet, the algorithm structures are not the same, and there are still certain limitations on TPU, so we will still mainly focus on GPU acceleration.

Chapter 3: Theory and Analytical Bases for the Work

This is a project that explores the potential of Artificial Intelligence (AI), specifically in its subset, deep learning. AI is a broad category that refers to machines that have the ability to learn and respond to relevant situations [12]. Much like the capabilities of humans and the animals in their care, these machines can learn new things through self-study or with external guidance. Thus, when facing new situations that have never been seen before, these machines can make their own decisions based on the knowledge they acquired. Among all the applications, the most well-known AI advancements belong to a category of algorithm called machine learning [12].

Machine learning algorithms rely highly on statistics and probability. Using statistics, these algorithms intake massive amounts of data and try to find any patterns or hidden relationships between the data. These algorithms will then make the highest probable predictions based on the information they have discovered from the data [13]. Machine learning has become a popular field of research, with a steep rise in the number of related publications due to the versatility and adaptability of machine learning algorithms. Any data that can be digitalised becomes a potential research subject for machine learning, such as numbers, words, and images. Furthermore, we can classify the algorithms according to the different learning methods they utilise, namely supervised learning, unsupervised learning, and reinforcement learning [13]. The sole difference between supervised learning and unsupervised learning is whether the data come with labels. If the data are labeled and we expect the algorithm to find specific patterns, this learning method can be categorised as supervised learning. On the other hand, if the algorithm is fed with unlabeled data and we only expect to find random patterns, the process is called unsupervised learning. Lastly, the latest frontier of machine learning is known as reinforcement learning. This is a method that is more likely to "learn" from the data, instead of simply sorting and categorising it all. A reinforcement learning algorithm goes through trial and error, receiving rewards and penalties depending on its actions, much like how dogs are trained [14]. As one can imagine, reinforcement learning produces the most flexible model with the highest potential. A well-known example of reinforcement learning is the creation of AlphaGo, the program that had defeated the top human player of the most complex chess-type strategy game, Go.

Recently, a machine learning technique that was invented in the 1980s dubbed deep learning has become feasible due to improved computing power [15]. Deep learning has attained more impressive results than ever before, making certain high-risk applications such as autonomous vehicles much more practical, since the accuracy of the AI system has a direct impact upon user safety. Most deep learning algorithms are

based on neural network architecture consisting of a significantly larger number of hidden layers than traditional machine learning methods [15], another reason why deep learning requires the significantly greater computing power that has only become possible in recent years. Instead of processing the labels and the manually extracted features, the mass hidden layers of deep learning are capable of extracting detailed information from the data to perform various learning methods on its own [16]. Another privilege afforded to deep learning holds is that performance typically improves as the size of the data increases [15]. Conversely, for traditional machine learning methods, the performance curve will eventually reach a plateau even if the dataset continues growing. In view of the strengths of deep learning, we will be using such a technique to accomplish our goals.

3.1 Deep Learning with Audio Signals

In the previous section, we revealed that any data that could be digitalised can benefit from the machine learning technique. Therefore, the first and foremost step of applying deep learning to audio files is by digitalising the audio. Audio signals can be plainly understood as vibrations. When the source makes a sound, the source is simply generating energy to compress media such as air or solids. In other words, audio signals represent changing energy in different frequencies along with time. However, it is impossible and impractical to record every detail in a piece of audio, since it will require infinite memory; that is why we have sampling frequency, the sampling rate used to record audio information per second.

The goal of deep learning is to train a model with given data under the expectation that the model will perform specific tasks. Hence, we must perform feature extraction, going from low-level audio data which merely contains amplitudes to a high-level representation of the audio signals [17]. Intuitively, we know that audio signals have some basic features such as amplitude and frequency. The easiest way of visualising an audio file is to plot its signal waveform. However, we want the machine to grasp exactly what the human is hearing and processing, otherwise the machine might work on finding patterns that are not influential to the human ear [18]. The following presents a straightforward example. Figure 3.1 shows the waveforms of different sounds that might be heard in the city. Note that the waveforms of sirens, idling engines, and jackhammers are visually very similar, but from our experience we know that there is no difficulty distinguishing between the three. Consequently, we can imagine that if we provide waveforms as the input data to train the models, the results may well be unsatisfactory. Instead of examining the time domain, the more commonly used method is to transform the audio into the frequency domain so as to perform further analysis. More audio features will be demonstrated in detail in the following section.

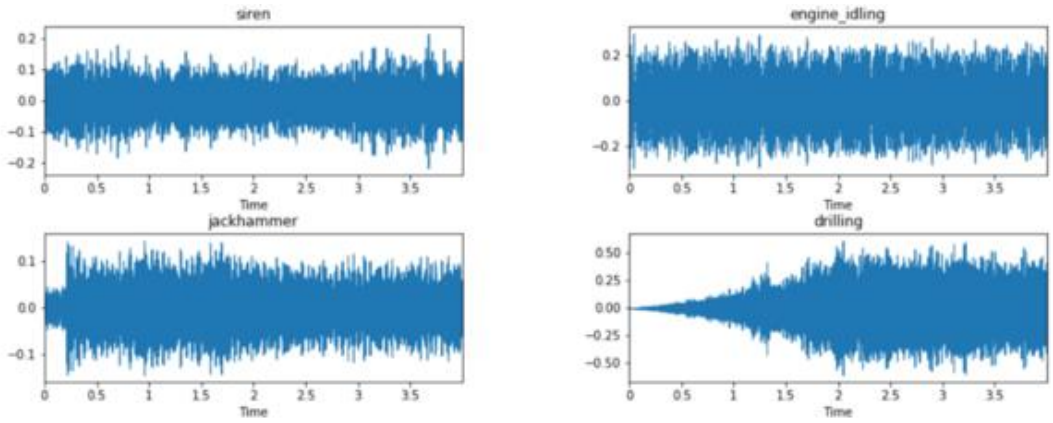

Fig. 3.1 Waveforms of different urban sound. [19]

3.1.1 Spectrogram

The goal of this project is not simply classifying audio files; therefore, the features that we choose to extract should also be editable. Recently, the generating and converting of images through deep learning methods has achieved significant progress [20]. As such, we can benefit from these successes if we can transform audio files into images. This is why the spectrogram is introduced. An audio and machine learning researcher, Adam Sabra, has a very precise explanation for spectrograms: "A spectrogram is a figure which represents the spectrum of frequencies of a recorded audio over time [21]." A corresponding spectrogram of a given soundwave is illustrated in Fig. 3.2.

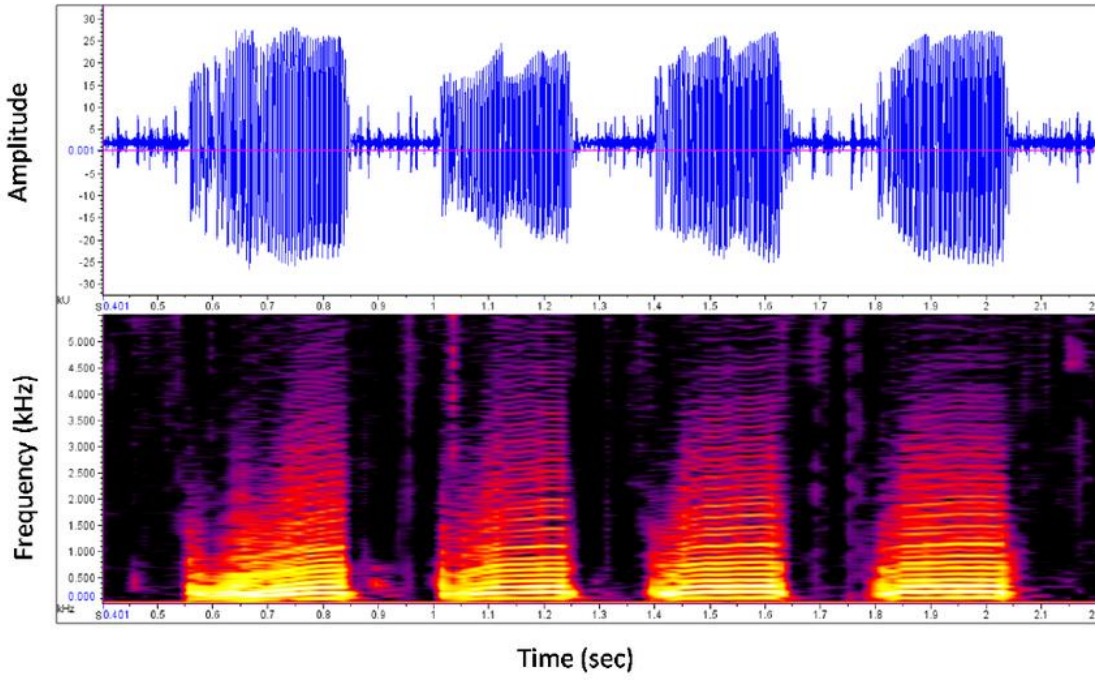

Fig. 3.2. Soundwave and the spectrogram of the boatwhistle call of a toadfish [22]

When performing the Fourier transform on a signal, all the frequencies involved in the audio are extracted into providing a great deal of information; however, these data cannot be easily edited. To attain a spectrogram, the audio first has to be split into tiny windows and the Fourier transform performed on each window. Taking the absolute values and combining all the results from different windows, we get an image that can be processed through deep learning algorithms. The brighter parts in the spectrogram demonstrate that more energy is concentrated in the matching frequency range. In several periods of time, the entire frequency range is relatively darker than other sections. This means that the actual sound is also quite silent compared to other segments, which can also be easily observed in Fig. 3.2. It is worth mentioning that a spectrogram provides us with a fairly strong understanding of an audio sample without even listening to it. From the shape and the structure of the colour blocks in the spectrogram, we can speculate upon the vocal prosody of the audio sample. Vocal prosody is an expression of speech denoting loudness, pitch, and timing [23], all important features when doing emotion conversion. In comparison with the original soundwaves, spectrograms are more ideal as input data since they contain higher-level information and details.

3.1.2 Mel Scale

After visualising the frequency in a piece of audio, there is still a vital step before feeding the spectrogram into the model and beginning the training process. The core idea is to provide the machine with a sense of hearing identical to that of the human ear, yet the human perception of audio is not linear. Instead, humans are more sensitive to lower frequencies [24]. For instance, 500 Hz and 1000 Hz will sound dramatically different to humans, but 7500 Hz and 8000 Hz will sound nearly identical. The Hertz scale in such a situation is no longer meaningful, and as such will be substituted with the Mel scale. The following shows the relationship between the Hertz scale (f) and the Mel scale (m) [24].

$$m = 1127\ln\left(1 + \frac{f}{700}\right) \tag{1}$$

There is a comparison of spectrograms of different scales in Fig. 3.3. The spectrogram on the right is in the Mel scale, meaning that it shows more significant details than human hearing, especially regarding how information below 4000 Hz is amplified. The resolutions and the data sizes are identical for both spectrograms; however, half of the information in the Hz-scaled spectrogram is redundant, which will only cause errors during training.

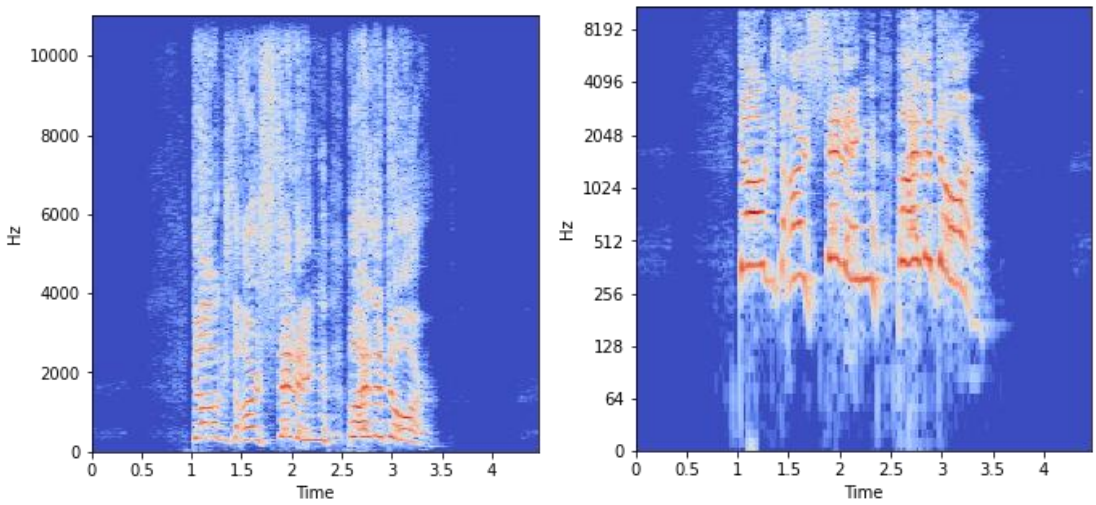

Fig. 3.3. Spectrograms of a sample speech in Hertz scale on the right and Mel scale on the left

3.2 Convolutional Neural Network (CNN)

3.2.1 The structure of the Convolutional Neural Network

Having consciousness and the ability to think has made humans the most intelligent creature on Earth. Accordingly, replicating the way humans use their minds to think and process information has become one of the research highlights of AI. Humans excel in the skill of creating the same content across different styles. For instance, different painters often portray the same scene in completely different ways, which should be nonsensical considering the information going into the painters' eyes is identical. However, the mechanism of the human brain has made that possible. Inspired by humans, researchers have come up with an algorithm called Deep Neural Networks [25]. As shown in Fig. 3.4, a deep neural network is based on node layer, including an input layer, multiple hidden layers, and an output layer. Each node stands for a neuron, and every node is connected just like the neurons of the human brain. The connections between the nodes have corresponding weights and thresholds, so one node will only be activated if the output of the previous node fulfills the limitations [25]. To fine tune the weights and thresholds, deep neural networks rely heavily on training to increase accuracy over time. Since the final classification is done inside the hidden layers with multiple nodes, deep neural networks can achieve very high accuracy even when classifying non-linear relationships.

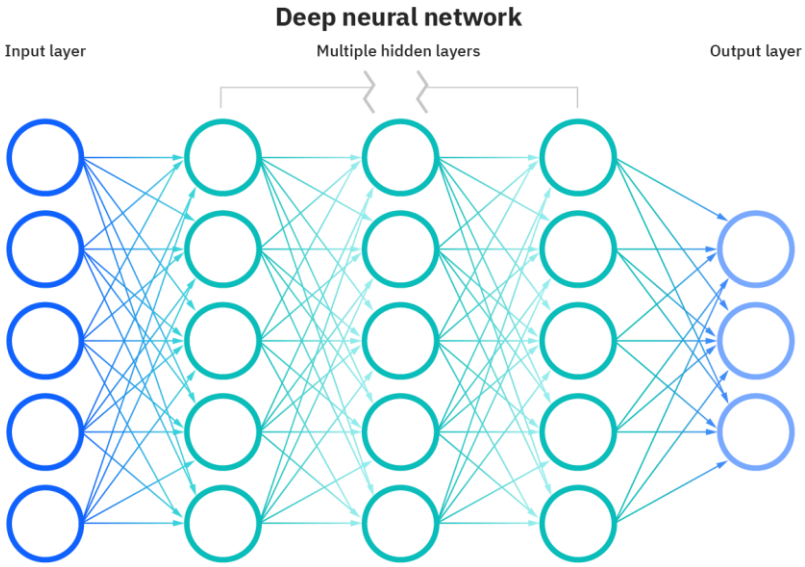

Fig. 3.4. Structure of a Deep Neural Network [25]

One robust class of deep neural network that masters image processing tasks is the Convolutional Neural Network (CNN) [9]. CNN is built by adding feature learning layers before a typical deep neural network. Inside these feature learning layers are computational units that analyse different aspects of images. These units act like filters that capture potentially useful information, with each unit responsible for extracting distinct features, very similar to how our brain processes images.

The role of CNN is to maintain the critical features of images while reducing resolution for easier further processing [26]. In reference to Fig. 3.5, the convolutional layers contain filters (or kernels) to carry out computational processes which will eventually produce convoluted feature outputs. The reason that a fully designed CNN holds multiple convolutional layers is that early convolutional layers can only extract low-level features such as edges, color, or outlines. Adding further layers will allow the network to dive deeper into the input image and explore more high-level features. The feature learning section shown in Fig. 3.5 provided the network with the opportunity to understand the input images, just like how our brains process information—although we rarely notice it.

Following the convolutional layers, we have the pooling layers. Although the use of pooling layers also reduces image resolution to save on computing power, the computational process is slightly different from convolutional layers. While convolutional layers extract the features, pooling layers amplify the extracted features and eliminate the features that are less dominant [26]. When performing the preferred type of pooling, max pooling, we scan the image with kernels and return the maximum value inside each kernel [26]. By doing this, we reduce the dimension of the image and

erase the noisy activations. Thus, max pooling can also be considered to be a noise suppressant.

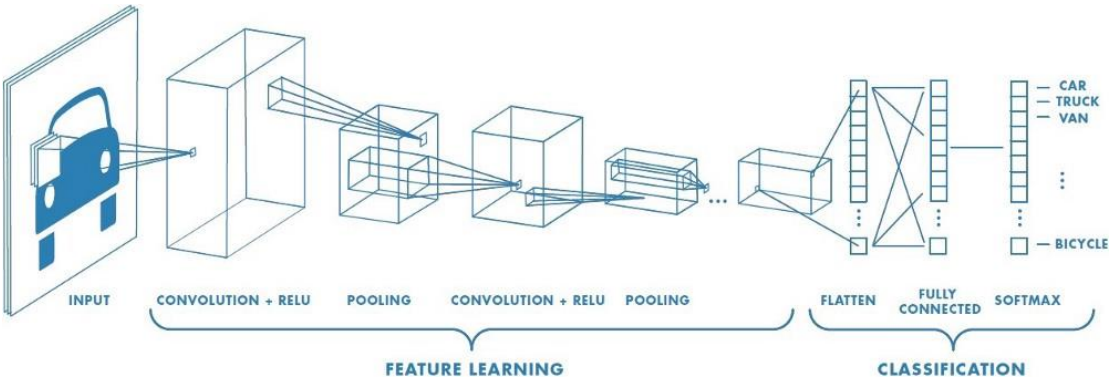

Fig 3.5. Structure of a Convolutional Neural Network. [26]

After completing the feature learning process shown in Fig. 3.5, the model has gathered enough features to analyse the image. These features should then be flattened into column vectors to be fed into a regular feed-forward deep neural network while applying backpropagation during every training iteration [26]. It usually requires several series of epochs before getting a reliable model; one commonly used output layer is the SoftMax Classification technique.

3.2.2 Applying the Convolutional Neural Network for image style transfer

The objective of performing image style transfer is to retain the input image content while adding the style from the respective image. In our case, we will be converting the spectrograms of the audios. The first problem is to find the rough sketch of the content but not the texture inside it, and low-level feature maps collected in early convolutional layers fully match the requirements [27]. From Fig. 3.6 we observed that by using features from earlier layers to reconstruct the image, the content will be sharper. As we dig deeper into the convolutional layers, the feature maps start to focus on the style and texture of the image instead of the content, therefore the reconstructed image has a much richer style but a rather blurry outline.

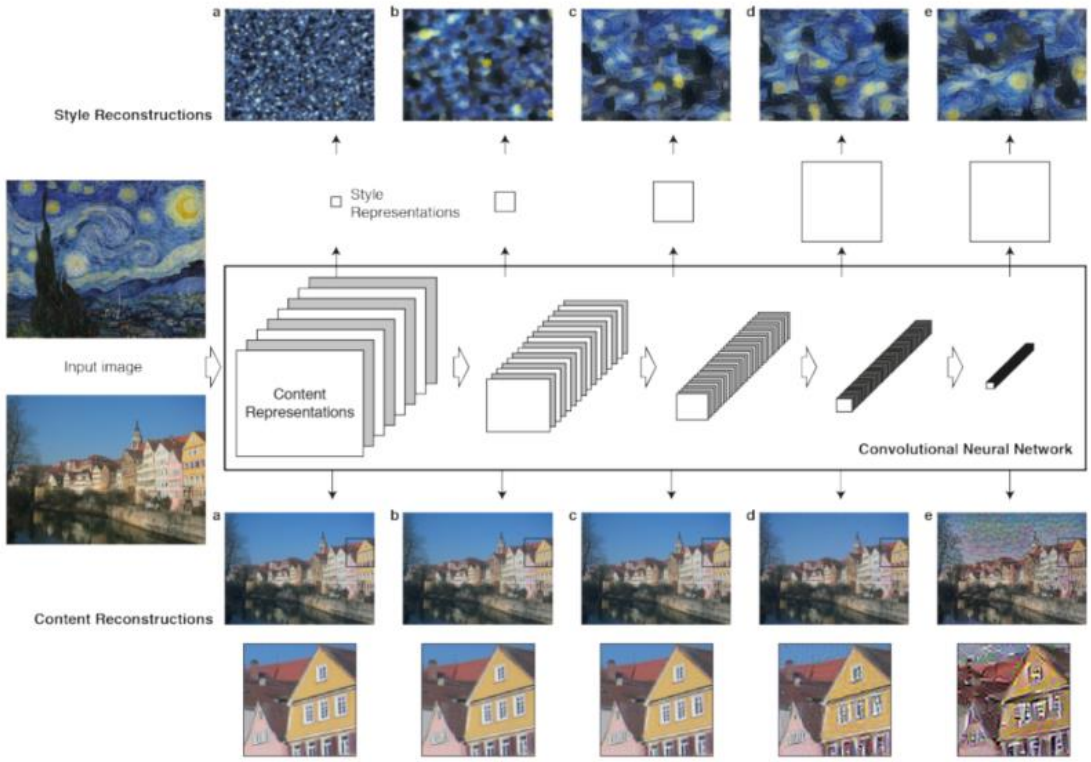

Fig. 3.6. Simple demonstration of how a Convolutional Neural Network works on images. [9]

The second problem is to extract the style of the target image and add the style to the content image obtained earlier. This can be achieved using Gram matrix [27]. Gram matrix is simply a matrix that contains the dot products between all the elements in the feature maps; this is a powerful technique because of the properties of dot products. A dot product is the multiplication of the magnitudes of vector a and vector b times the cosine of the angle between them. Intuitively, we find that a dot product indicates how similar two vectors are. The angle between vector a and vector b is smaller if they are more similar, and the dot product will be larger because of the cosine term. Since the feature maps are flattened into column vectors at the end of the feature learning stage, we can easily carry out dot products between the feature maps. From a feature map of depth N, we take all N flattened vectors and calculate the dot products of all the elements to obtain a Gram matrix of size $N \times N$. The corresponding equation is listed below [27], where G is the Gram matrix and F is the feature map.

$$G_{ij}^{l} = \sum_{k} F_{ik}^{l} F_{jk}^{l} \tag{2}$$

Finally, after computing the loss between the Gram matrix of the input and the style image, the required image style transfer can be done [27]. However, different layers in the network give different degrees of feature extraction. Too much emphasis

on the content will eliminate the style that we want to add onto the image, and in the case of spectrogram conversion, this will result in no change in the emotion style. Too much emphasis on the style will destroy the outline of the content, which is a disaster in spectrogram conversion—the transformed output audio will merely contain emotional human sounds, but no comprehensible content. There is one solution that can solve this issue. The loss function in our algorithm contains two terms: one for the content, one for the style. Therefore, the model will try to reach a balance between the two loss terms and gradually smoothen out the synthesised image so that neither the content nor the style will be overly emphasised.

3.3 Cycle-Consistent Generative Adversarial Network (CycleGAN)

Training an image-related deep learning model usually requires a large dataset with parallel data. However, it is expensive and time-consuming to create such datasets. Even when these datasets are created, they might be restricted to specific usage, meaning that building these datasets is not cost-effective. The CycleGAN is a method that automatically trains image-to-image conversion without the need for paired data [28].

Building a CycleGAN might be complicated, but the concept of CycleGAN is relatively straightforward. Instead of manually telling the generator model whether the generated image is satisfying or not, we build a discriminator model that automatically judges the quality of the output. Inside the CycleGAN architecture are two generator models and two discriminator models [28], as shown in Fig. 3.7. The variables x and y represent the features that belong to domains X and Y, respectively. One generator model takes the input from X domain and outputs images that belong to Y domain, while the other generator model does the opposite. The discriminator models will then examine whether the images are sent directly from the corresponding domain or the images are generated. To be more specific, the generator learned the mapping $G_{X \to Y}$ by introducing two losses, adversarial loss and cycle-consistency loss [3].

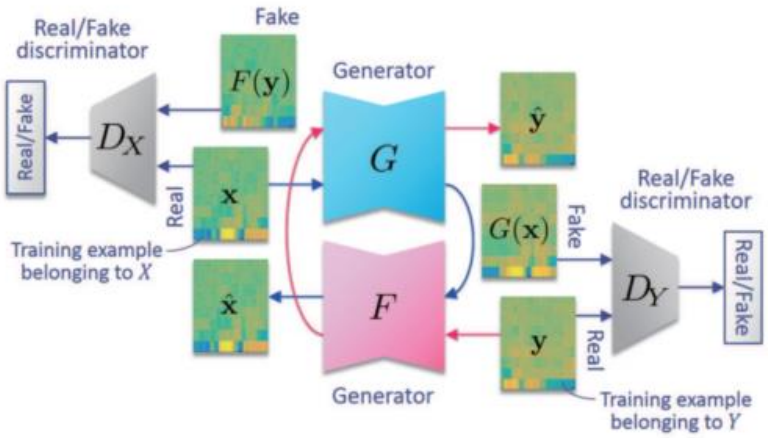

Fig. 3.7. Basic structure of a CycleGAN [11]

3.3.1 Adversarial loss [3]

Considering mapping $G_{X\to Y}(x)$, adversarial loss $\mathcal{L}_{adv}(G_{X\to Y}, D_Y)$ measures how similar the generated images are when compared to real images in the *Y* domain. Thus, the loss becomes smaller when the converted images $P_{G_{X\to Y}(x)}$ become more alike to the target images $P_{Data(y)}$. This can be written as [3]:

$$\mathcal{L}_{adv}(G_{X\to Y}, D_Y) = \mathbb{E}_{y\sim P_{Data(y)}}[\log D_Y(y)] + \mathbb{E}_{x\sim P_{Data(x)}}[log(1 - D_Y(G_{X\to Y}(x)))] \quad (3)$$

The goal of generator $G_{X\to Y}$ is to deceive discriminator D_Y by minimising adversarial loss, while the goal of discriminator D_Y is to distinguish between the generated images and the real images by maximising this loss.

3.3.2 Cycle-consistency loss [29]

Optimising adversarial loss $\mathcal{L}_{adv}(G_{X\to Y}, D_Y)$ simply tells us mapping $G_{X\to Y}(x)$ is successful, but the original contextual information of *x* might not be preserved. Therefore, CycleGAN presents two supplementary terms to solve the problem. One is the adversarial loss of inverse mapping $G_{Y\to X}$, which can be expressed as $\mathcal{L}_{adv}(G_{Y\to X}, D_X)$, while the other one is cycle-consistency. Cycle-consistency is the idea that for a complete model, any converted image should be able to be converted back to the original image, i.e. $F(G(x)) \approx x$. Cycle-consistency loss can be written as [29]:

$$\mathcal{L}_{cyc}(G, F) = \mathbb{E}_{x\sim P_{Data(x)}}[\|F(G(x)) - x\|_1] + \mathbb{E}_{y\sim P_{Data(y)}}[\|G(F(y)) - y\|_1] \quad (4)$$

By adding trade-off parameter λ_{cyc}, we can get the full objective [29]:

$$\mathcal{L}_{full} = \mathcal{L}_{adv}(G_{X\to Y}, D_Y) + \mathcal{L}_{adv}(G_{Y\to X}, D_X) + \lambda_{cyc}\mathcal{L}_{cyc}(G, F) \quad (5)$$

CycleGAN has achieved significant success in image processing. A well-known example is the conversion between horse and zebra photographs as shown in Fig. 3.8. The results are visually impressive. Applying CycleGAN on spectrum conversion should also give us reasonable outcomes.

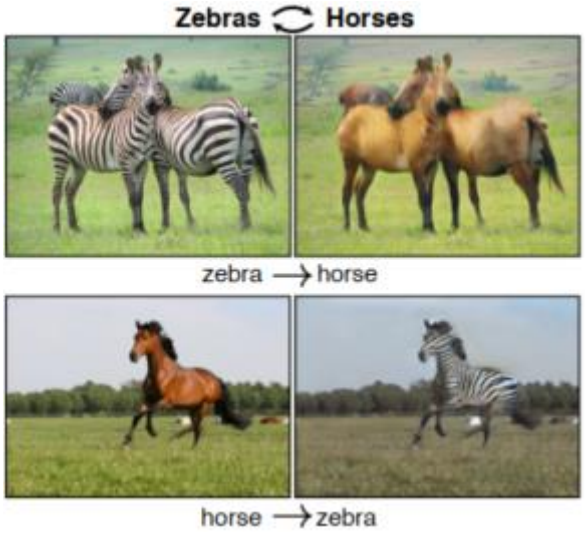

Fig. 3.8. Examples of image translation using CycleGAN [29]

3.4 Evaluation Methods

In this project, we use CNN and CycleGAN to perform emotional voice conversion. Previous studies discussed how evaluating the quality of voice conversion can be quite challenging [7]. Since the main approach we adopt is converting spectrums, we have collected numerous relevant studies and found that the most commonly used objective evaluation method is the Mel-cepstral distortion (MCD) [3], [7] – [30], where MCD is defined as [8]:

$$MCD[\text{dB}] = \frac{10}{ln10}\sqrt{2\sum_{i=1}^{24}\left(mcep_i^{target} - mcep_i^{convert}\right)^2} \quad (6)$$

In (6), $mcep$ stands for Mel-cepstral coefficients (MCEPs), which is also the input that we use to train the CycleGAN model. MCD analyses the similarity between the MCEPs of two audio data [7], which provides a fair view of how close these two audio pieces are based on their extracted features. Since we are calculating the distance between the two converted MCEPs, the smaller the MCD value, the smaller the difference between the two audios will be. Showing the MCD results of the target audios and the generated audios does not provide enough information; therefore, the MCD results of the input audios and the generated audios will also be given. In comparing the MCD values, we can know whether the features of the generated audio are more similar to the features of the target audios or the original ones.

Moreover, some researchers inspect the audio pitch, which is also called the fundamental frequency, F0 [8], [30]. This serves to evaluate the quality of the converted audios from a more intuitive perspective to the human auditory experience while remaining objective. Pitch is an acoustic feature that could easily be observed and classified even with standard human listening ability. In general, the voices of females

have higher pitch than their male counterparts, and speech with stimulating emotions such as anger also have higher pitch than speech with stable emotions such as neutrality.

Finally, since this is a project that focuses on converting the emotional styles in given speech, it is of paramount importance to evaluate the performance by human listeners. Although it cannot be quantified, most studies will carry out subjective evaluation by holding a listening test [3], [11], [7] – [30]. To perform the subjective evaluation scientifically, the Mean Option Score (MOS) is one of the most commonly accepted methods [31]. MOS provides the listeners with a scale of 1 to 5 having corresponding labels such as bad to excellent. The average score of the listeners will then form a reasonable evaluation. The following includes some MOS results from previous studies: the models built in [3] scored roughly 2.4; the model built in [7] scored between 2.2 to 3.1, while also mentioning a model existing in another study which scored between 1.8 to 2.4; and the models built in [30] scored between 2.0 to 4.0, depending on the type of the emotion converted. Most of the results ranged between 3.0 to 3.5.

Chapter 4: Technical Method

The quality of the dataset directly affects the results of performing emotional voice conversion. Two frequently used databases with reliable audio quality are the Toronto Emotional Speech Set (TESS) [32] and the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) [33]. However, the audio samples in those datasets are sorted into different formats, which will cause certain inconvenience. In the end, we decided to use a sorted database [34] which contained audio files from four separate databases, namely RAVDESS, CREMA-D, SAVEE, and TESS. There were six predominant emotional categories in this database: angry, happy, sad, neutral, fearful, and disgusted.

From previous studies, we found that emotional voice conversion was quite challenging, since MOS swung between 2.0 to 3.5 for most of the models. MOS was the main measurement of the quality of telecommunications, and according to the industry standard [35], scoring below 3.6 indicated that many of the users were not satisfied. To maximise the success rate of the experiment that we were going to conduct, we decided to discern between the two most disparate emotions so that any successful conversion between the speech samples could easily be identified. Hence, we carried out pitch analysis, attempting to distinguish the relevant emotion by calculating F0.

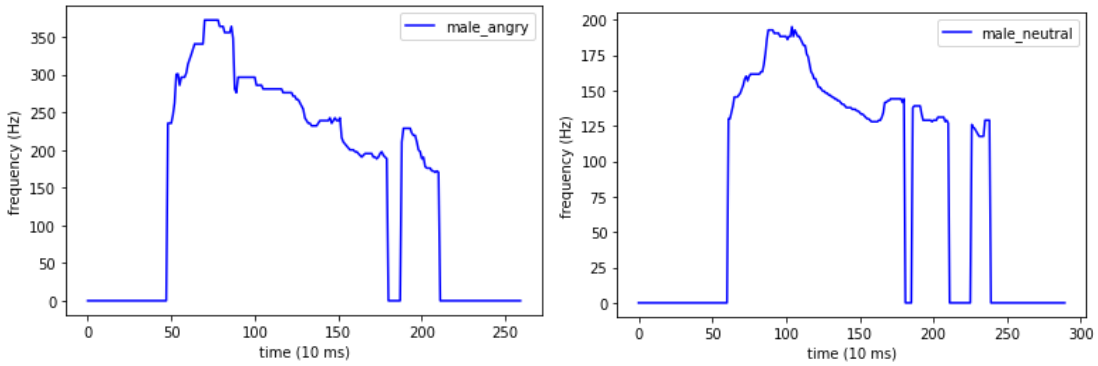

Fig. 4.1. Pitch analysis for two different emotions. The content and the speaker were identical; the only difference was the speech emotion.

The YAAPT algorithm was chosen because it outperformed the YIN and Praat algorithms [36]. Each analysis frame length was set to 40ms and the minimum F0 value was set to 75 to minimise interference. The noise and the pausing periods would not only be meaningless to display, but they could have negative effects on neighbouring frames. An analysis sample is shown in Fig. 4.1. The sample audios were recorded by the same male speaker saying the exact same sentence “Maybe tomorrow it will be cold” in two different emotional states, angry and neutral. Since the only difference was the emotion, the F0 trends should be very similar but with different magnitudes, meaning

that the speaker was saying two identical sentences with various pitches, in this case related to the speech emotion. For each emotion, 100 samples were randomly selected, including both male and female voices. The F0 of these samples were then extracted and the zero values were removed. Finally, we plotted the results into distributions, as Fig. 4.2. shows.

From the results, we could easily observe that neutral, disgusted, and sad emotions had very similar low pitch range while angry, happy, and fearful emotions shared similar high pitch range. It would be relatively difficult to convert the emotions in similar pitch range, since one of their features was no longer representative. Thus, we decided to perform conversion between neutral and angry speech samples.

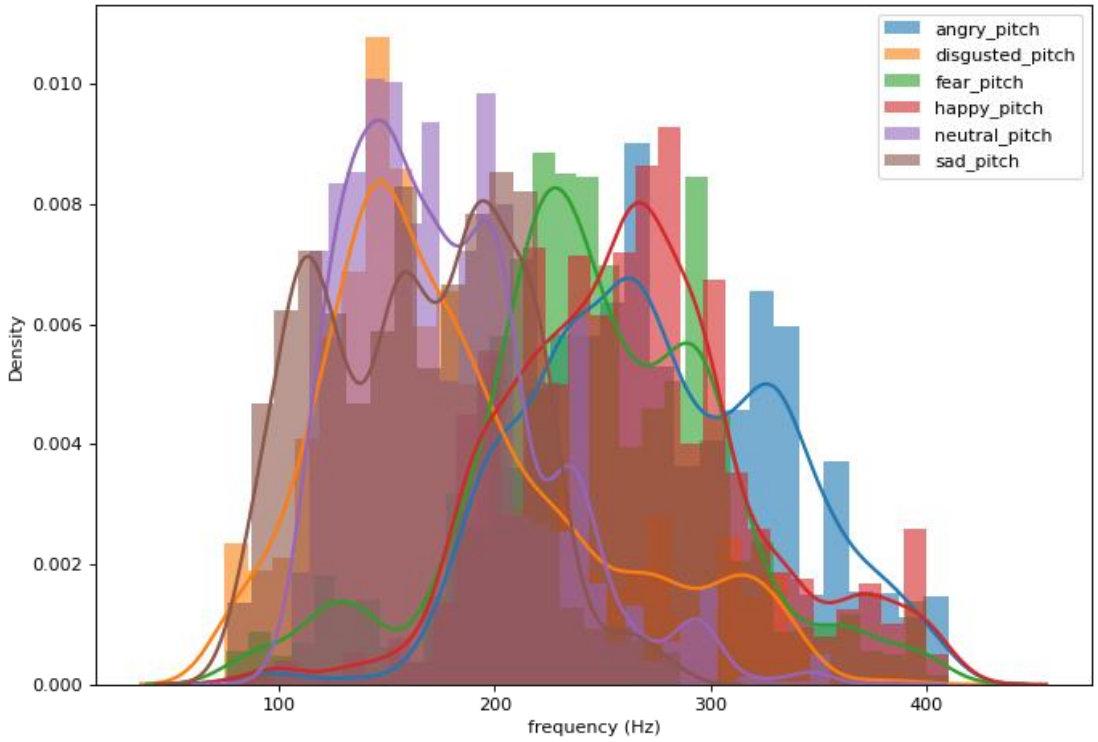

Fig. 4.2. Pitch analysis for six different emotions; each emotion contained 100 samples

4.1 Experimental Setup using CNN Method

The CNN model that we built takes two inputs: one will be the input audio file that we want to convert, and one will be the target audio file with the emotion that we desire. Since it does not require a large dataset, the training process is relatively fast compared to that of other models. There are also limitations related to the type of the data. As mentioned before, this model directly copies the style of the target audio and pastes it onto the input file. Consequently, the more similar the structure of the two given audio files, the better the final output should be. This will be more comprehensively discussed in the next chapter.

4.1.1 General Structure of the CNN model

We built the CNN model based on PyTorch and boosted the training speed using the GPU accelerator provided by Colaboratory. The audio processing mainly relied on the Python package, Librosa. First, the model loaded the two audio files and transformed them into spectrograms. The number of output frequency bins, which was referred as n_fft in librosa, was recommended to be set to 512 for speech processing [37]. Since we were dealing with spectrograms, a 2D convolution structure was implemented. Moving on to generating the new spectrum, the ideal method would be starting by producing a spectrogram that was very similar to the input audio spectrogram, and then gradually modify the generating mechanism to fit the style of the target audio spectrogram. Thus, the initial weight of the content was set rather high at 100 and the initial weight of the style was set fairly low at 1.0. The number of epoch iterations was originally set to 20000, but later experimentation proved that 10000 would be sufficient, as the lost curves in most of the attempts would stabilise before 10000 epochs. Finally, we would have liked to transfer the generated spectrogram back to the audio, but it required some sort of phase information to obtain an audible transferred audio. To solve this problem, we extracted the phase information from the input audio file and provided it for the spectrogram-audio transfer. Although the generated spectrogram and the input audio phase information did not have a perfect match, this was the best solution, according to our research.

4.1.2 Parallel Data Training

The two audio files fed into the model in this section were very similar. The speaker and the content were both identical, so the only difference would be the emotional style. Any possible variables that might cause errors were minimised, meaning that the chance of getting solid results was very high. Since the results of parallel data training were the most reliable, this was the optimal training method to examine the tuning of other parameters, such as the kernel size and the number of training epochs.

Kernel size: A larger kernel size usually depicted a smoother output image, which was important for generating spectrograms since a rough spectrogram would result in a blurry audio. The goal of this experiment was to examine whether a kernel size of (3,3) performs better than a kernel size of (3,1). The speaker of the input audio and the target audio was the same female speaker and the content was identical. The emotion of the input audio was neutral, and the emotion of the target audio was angry. The kernel size with the better performance was later used in all the other experiments.

Audio length: Longer audio resulted in larger spectrograms which would increase the size of the training data. We wanted to examine if the size of the training data affects the required training epochs. Several parallel data trainings were carried out and the

input audios were all set to neutral while the target audios were all set to angry.

Other emotions: We were initially focusing on converting neutral speeches into angry speeches to maximise the chance of success while tuning the unsure parameters. After these numbers and details were found, we did some experiments to see how our model performs when converting other emotions. However, we still followed the rule mentioned when analysing F0 in the previous section. The emotions we tried to convert came from different pitch ranges, such as sad and happy.

4.1.3 Semi-parallel Data Training

Before moving on to use the ideal form of training, non-parallel training, we examined the quality of the emotional conversion between speeches with different content. The speakers were still the same, but the emotions and the content of the input and target audios were different.

4.1.4 Non-parallel Data Training

Finally, we carried out the training method that was more practical in real life scenarios. Usually, there would not be a parallel dataset, so an algorithm that could be trained using non-parallel data would have the maximum potential. However, there were limitations in the CNN architecture that were unsolvable, so applying non-parallel data training on CNN-based EVC in theory would not perform too well. Still, we tried to improve our model by tuning the parameters such as the weights of the content and the style. These parameters were not considered to be modified in previous stages, as they were values suggested from other studies. Similar examples were the variables used in the spectrogram transformation, as they were values suggested by the developers of the Python packages. Tuning these values might not be useful, or could even worsen the quality of the conversion.

The input audio we used was neutral female speech, and the target audio was male angry speech. We changed four sets of different content weights and style weights to examine the difference. The original [content weight, style weight] was [100, 1] and we tried [100, 0.5], [200, 0.5], and [200, 0.25].

4.2 Experimental Setup using CycleGAN Method

To break the limitation of the CNN model that we built, we tested the CycleGAN method. This CycleGAN model was trained on non-parallel data, which was a major improvement upon the CNN model. In contrast to using spectrograms as the training data, we extracted 24-dimensional Mel-cepstral coefficients (MCEPs) to train the CycleGAN. The generators and the discriminators that we used were based on the architecture shown in Fig. 4.3, where h, w, and c are height, width, and number of

channels; k, c, and s are kernel size, number of channels, and stride size. This architecture was inspired by [3].

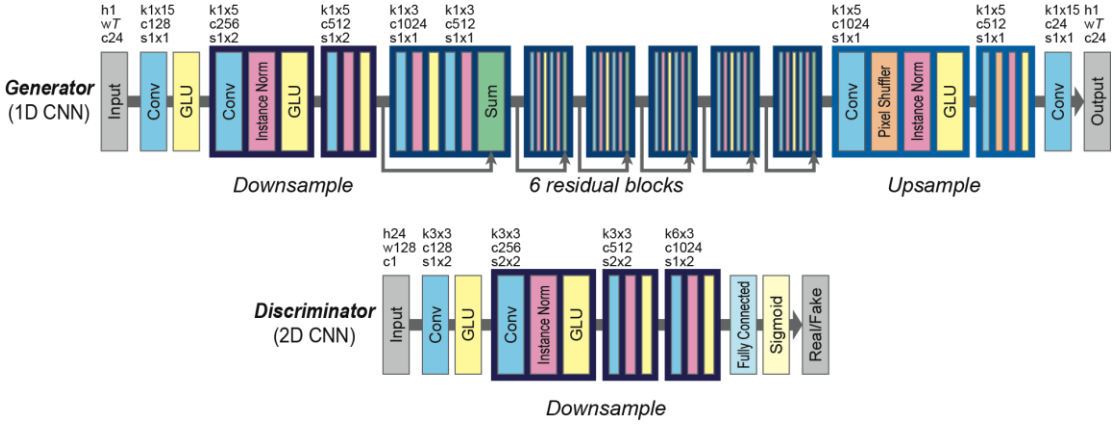

Fig. 4.3. Architectures of the generators and the discriminators. [3]

To train the model, we randomly selected 250 neutral audios and 250 angry audios. This ensured that all the training data were non-parallel. Since the advantage of CycleGAN was that the training data were not required to be equal in length, no further pre-processing was needed. The network was trained by the Adam optimiser and the batch size was set to one. The initial learning rates of the generators and the discriminators were set to 0.0002 and 0.0001, respectively. We set the number of epochs to 1000 due to the limitations of our training environment. The usage of GPU acceleration was limited to 12 hours or even less per day in Colaboratory, forcing us to finish the training session in less than 10 hours to ensure that the training would not be forced to stop.

4.1.1 Reduced Training Time

We executed several attempts using the same setup parameters mentioned previously. The independent variable of this section was the number of epochs. Although in theory, the longer the training session was, the better the trained model would be. However, we would like to examine the difference between a barely-trained model and a well-trained model. After evaluating the results of this section, the number of epochs was fixed at 1000.

4.1.2 16-dimensional MCEPs Training

This section we examined the relationship between the resolution of the training data and the quality of the trained model. We extracted 24-dimensional MCEPs as the input training features earlier since it was suggested in other studies. A new model was trained by using 16-dimensional MCEPs while other parameters remained the same.

4.1.3 Reduced Dataset Training

It was apparent that a large set of data was needed to create a well-performing model. Still, we inspected the result of training the model with reduced size dataset. The parameters of the setup were identical to 4.1.1 while the number of epochs was set to 1000.

4.3 Objective Evaluation

As discussed in previous chapter, we intended to evaluate the performance of our models objectively by MCD and F0. Note that only the best model was being evaluated. To evaluate the final CNN model, we chose three training samples that converted different emotions: angry to neutral, neutral to angry, and sad to happy. To evaluate the final CycleGAN model, we used the best training result obtained from section 4.2. that performed neutral to angry EVC.

For each sample, we first calculated the MCD between the input audio and the generated audio, and then calculated the MCD between the actual emotion audio and the generated audio. Note that the actual emotion audio was not the target audio. Instead, the actual emotion audio had the same speaker and content of the input audio, but it was recorded in the target emotion. Such an evaluation method required parallel data when calculating the MCD. Similar to our evaluation method using MCD, we found the F0 of the input, target, and generated audio. Finally, we determined whether the F0 distribution of the generated audio was closer to the input audio or the target audio.

4.4 Subjective Evaluation

We found 10 listeners to participate in the MOS test. Each listener was provided with the input, target, and generated audios. They were asked to mark the converted audio out of a scale of 1.0 to 5.0, and the corresponding labels were: 1.0, no emotion converted; 2.0, something was converted but the difference was barely discernable; 3.0, emotion conversion was subtle and might cause confusion; 4.0, emotion was recognisable; and 5.0, emotion was perfectly converted.

Chapter 5: Results and Analysis, Discussion

5.1 Training with CNN

To make the analysis easier to follow, we first analyse a successful EVC example to provide a general idea of the results of the CNN model we built. Following up, we analyse the result in the sequence of the experiments we performed. Note that we represent the audio samples in the form of spectrograms, since that is the only way to show the features of audio samples in paper reports.

5.1.1 Successful Neutral to Angry EVC using parallel data

The model in this section was trained with parallel data. The input audio was neutral male and the target audio was angry male. According to Fig. 5.1., we could observe that the generated audio spectrogram had similar structures and outline with the input audio, meaning that the content was the same. Note that we were using parallel data, so the structure of the target audio spectrogram was also similar to the generated audio spectrogram, since they share the same content. The style of the generated audio spectrogram had wavy features, which was different than the style of the input audio spectrogram (flat features) but similar to the target audio spectrogram (wavy features). In combining the listening experience, this was a successful EVC.

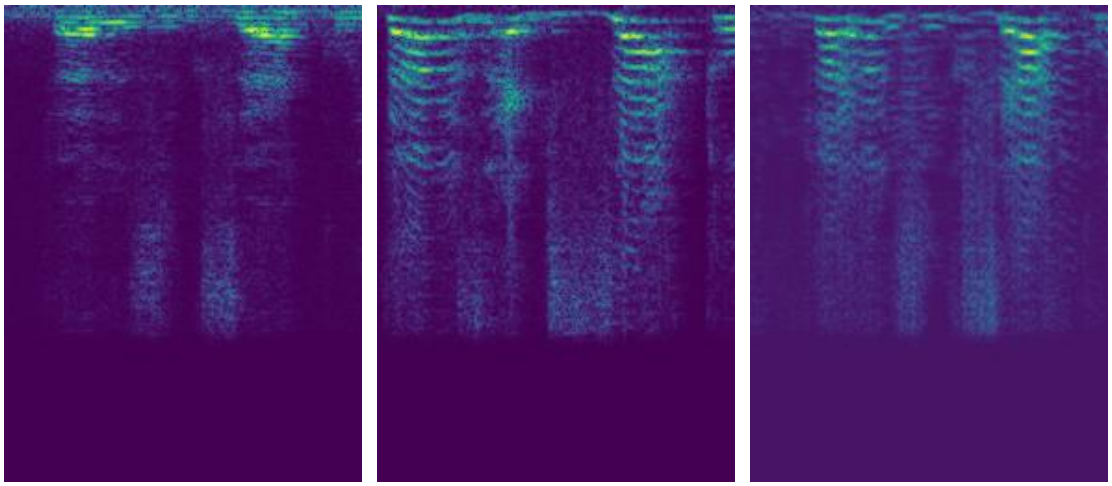

Fig. 5.1. Result of neutral to angry EVC using parallel data. Spectrograms of the input audio (on the left), the target audio (in the middle), and the generated audio (on the right)

5.1.2 Different Kernel Size

We assumed that a kernel size of (3,3) would outperform a kernel size of (3,1), and the result had proven this correct. According to Fig. 5.2, training the model with a kernel size of (3,3) provided a better result. The image on the left of Fig. 5.2 could be described as having smooth lines with wavy features, while the image on the right was full of

jagged lines. Although both audio samples were comprehensible, these jagged lines resulted in glitter noises.

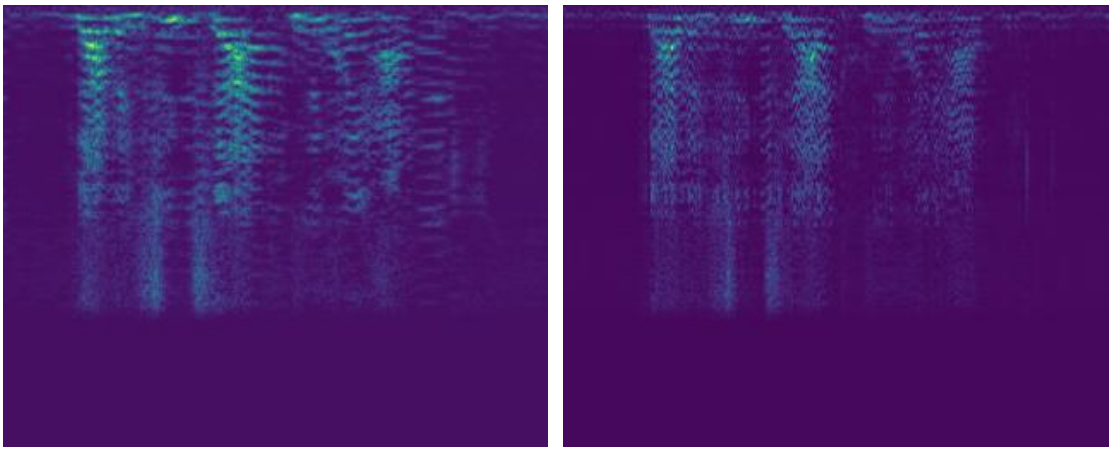

Fig. 5.2. Spectrogram of the generated audio trained by model with a kernel size of (3,3) (on the left), and the spectrogram of the generated audio trained by model with a kernel size of (3,1) (on the right)

5.1.3 Different Audio Length

We originally believed that longer audio would require more training epochs to reach a flat loss function. However, the only thing that increased was the training time. The training audio samples were double the length of previous experiments, but their loss functions were all relatively low and stable after 5000 epochs. However, since each epoch was responsible for training more data, the training session was 50% longer than that of previous experiments.

5.1.4 Successful Sad to Happy EVC using Parallel Data

The results in this section showed the limitations of performing EVC using CNN. The target audio was a happy female speech sample, and the speaker ended the speech with a rising tone, as it fit the happy situation in this specific sample. This rising tone can be seen in the middle image of Fig. 5.3. The EVC was only successful because we were using parallel data and the speaking context was identical. Obviously, not all the happy samples had a rising tone at the end. The generated audio in this experiment had an excited tone instead of a monotonous voice, which formed a successful EVC without neutral or angry emotions involved. However, the restrictions of CNN were also revealed.

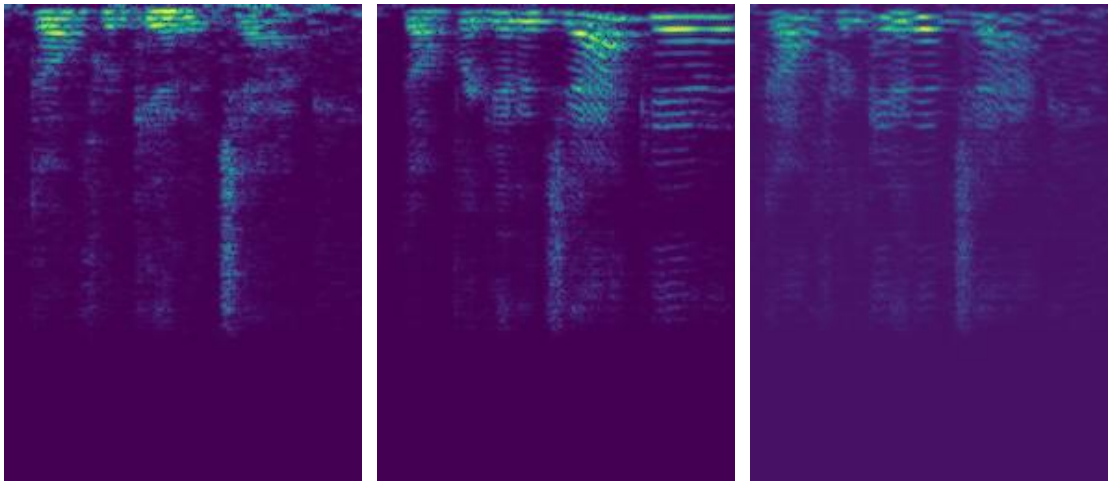

Fig. 5.3. Result of sad to happy EVC using parallel data. Spectrograms of the input audio (on the left), the target audio (in the middle), and the generated audio (on the right)

5.1.5 Successful Neutral to Angry EVC using semi-parallel data

Three complete spectrograms were provided, as the focus of this section was semi-parallel data training and it would be clearer to provide the features of the audios from the beginning to the end. By inspecting the difference between Fig. 5.4. and Fig. 5.5, it is obvious that the content of the two audio samples were totally different. We would like to convert a neutral speech of a male into an angry speech using another angry speech of the same speaker. In other words, we would like to add the wavy features onto the input audio spectrogram while preserving the content. From Fig. 5.6 we could see that the main structure of Fig 5.4. was retained, while the wavy style from Fig. 5.5 was added. Although quite a lot of noise appeared, the generated audio was still comprehensible.

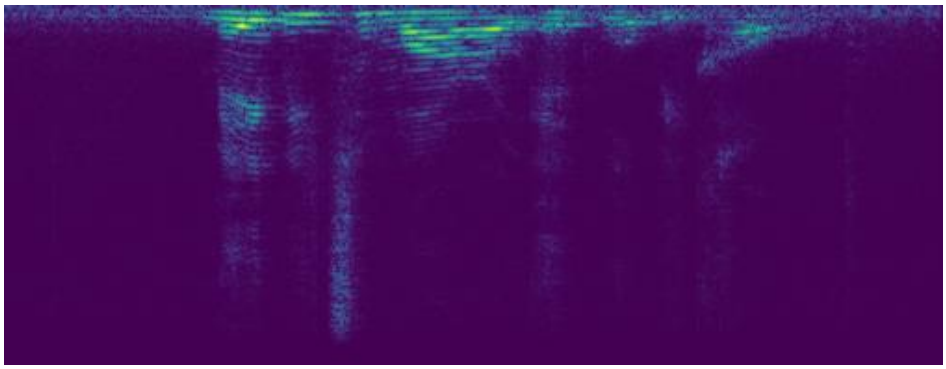

Fig. 5.4. Spectrogram of the input audio from neutral to angry EVC using semi-parallel data.

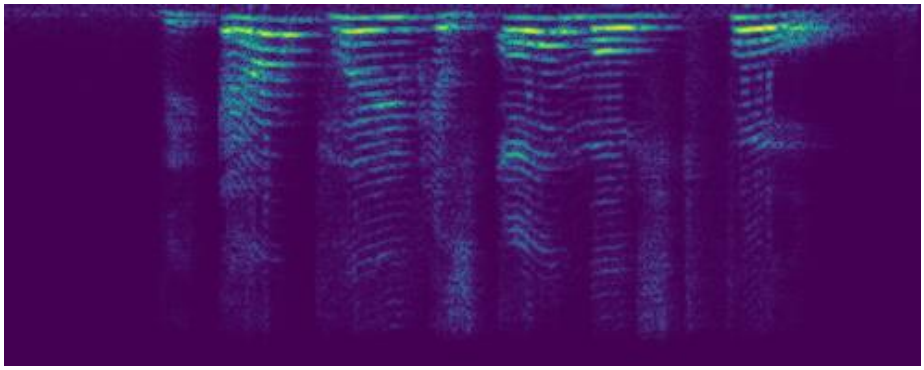

Fig. 5.5. Spectrogram of the target audio from neutral to angry EVC using semi-parallel data.

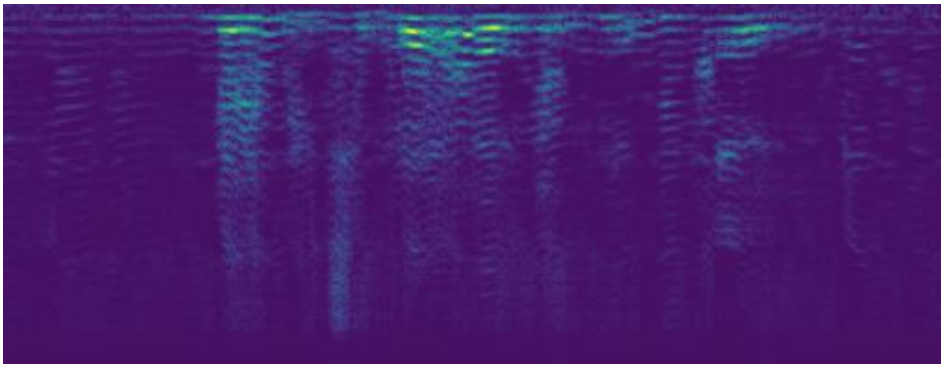

Fig. 5.6. Spectrogram of the generated audio from neutral to angry EVC using semi-parallel data.

5.1.6 Neutral to Angry EVC using non-parallel data

Knowing that training the CNN model using non-parallel data was impractical, we did not expect this training would give any helpful result. However, changing the content weights and the style weights improved the performance, although the overall performance was still not qualified as a successful EVC. From Fig.5.7. we could see that the content and the style of the two audios were totally different. The input audio speaker was a female and the target audio speaker was a male. The goal was to generate a spectrogram that preserved the outline on the left while combining the wavy style on the right.

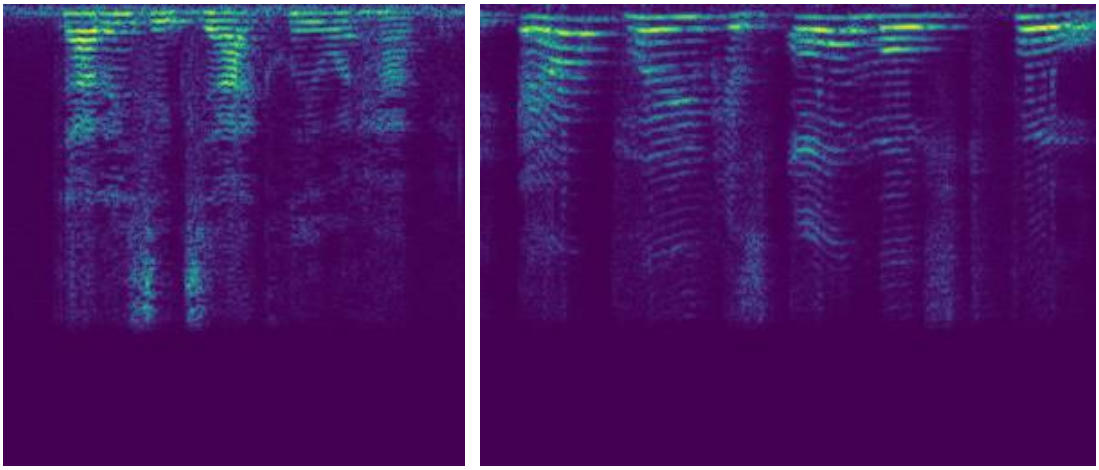

Fig. 5.7. Spectrograms of the input audio (on the left) and the target audio (in the middle) used for the training of neutral to angry EVC using non-parallel data.

The results of changing the content weight and the style weight were shown in Fig. 5.8. The changes were very subtle visually inspecting the spectrogram but there were some obvious differences. Keeping the same values as before, the top left result in Fig. 5.8. successfully converted the emotion but the speaker identity was also converted, therefore we increase the content weight and reduce the style weight. There was some information on the speaker identity in Fig. 5.7., the female speech had a higher average pitch while the male speech mainly focused on lower frequencies. This could be found in Fig. 5.8., as the lower style weight we assigned, the more speaker information was kept. The bottom right result in Fig. 5.8. preserved more speaker identity than the top left result in Fig. 5.8. The former result sounded more like a female speaking because the frequency were higher while the latter result sounded like a male speaking as it emphasised the lower frequencies.

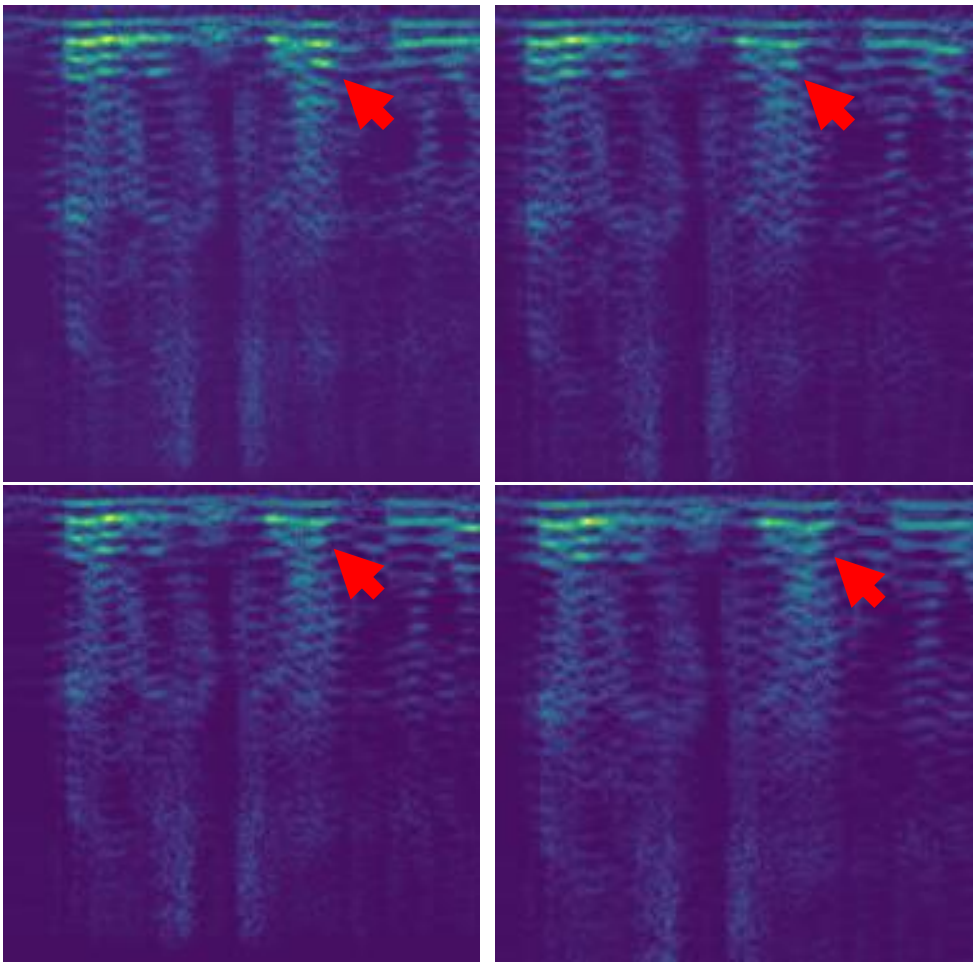

Fig. 5.8. Result of neutral to angry EVC using non-parallel data. The top left result was trained by using content weight=100 and style weight=1. The top right result was trained by using content weight=100 and style weight=0.5. The bottom left result was trained by using content weight=200 and style weight=0.5. The bottom right result was trained by using content weight=200 and style weight=0.25.

5.2 Training with CycleGAN

To evaluate the different models we trained using CycleGAN, we tested all the models using the same audio show in Fig 5.9., which was a neutral female speech. All the CycleGAN models we built perform neutral to angry EVC.

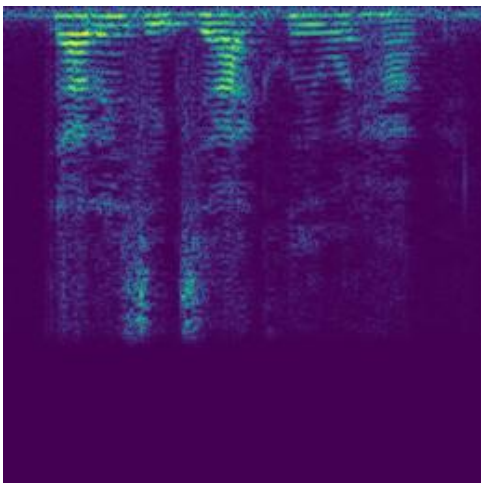

Fig. 5.9. Spectrogram of the input audio that we used to examine the performance of all the CycleGAN models

5.2.1 Training Epochs

The model trained with 100 epochs was able to convert some of the emotional features, but the quality was not as good as the model trained with 100 epochs. We could observe this by referring to Fig. 5.10. Both spectrograms were added certain wavy styles which means that they were successfully converted to angry speech in some degree. However, the noise and the unsmooth transformation on the left lowered the listening experience. Also, the result on the right was able to generate more speech power difference, which made the angry style even more obvious.

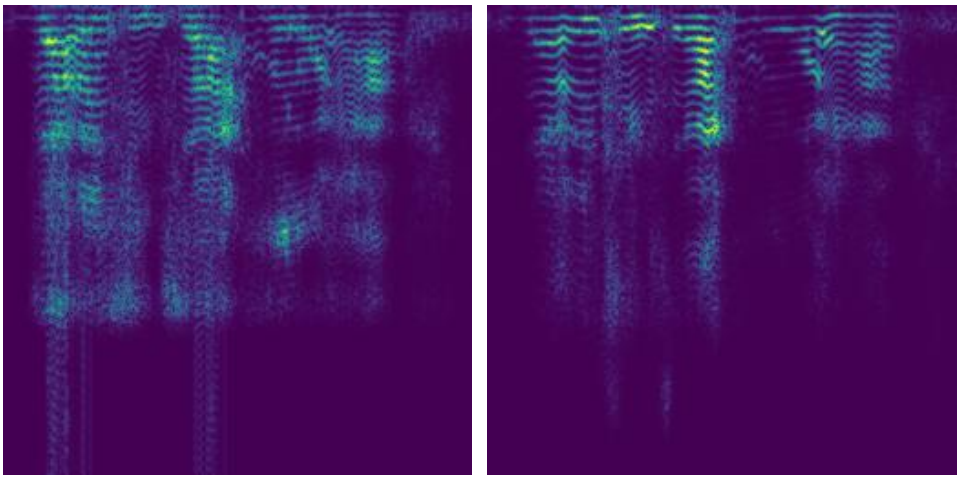

Fig. 5.10. Results of EVC using CycleGAN with different training epochs. Spectrogram of the result trained by 100 epochs (on the left), and spectrogram of the result trained by 1000 epochs (on the right)

5.2.2 Dimensions of the MCEPs

The model using 24-dimensional MCEPs performed slightly better than the model using 16-dimensional MCEPs. Both models produced fairly smooth transformations that provided a good listening experience. Still, from Fig. 5.11. we can see that the model trained by 24-dimensional MCEPs generated more angry tone by increasing the speech power difference and was able to suppress more noise.

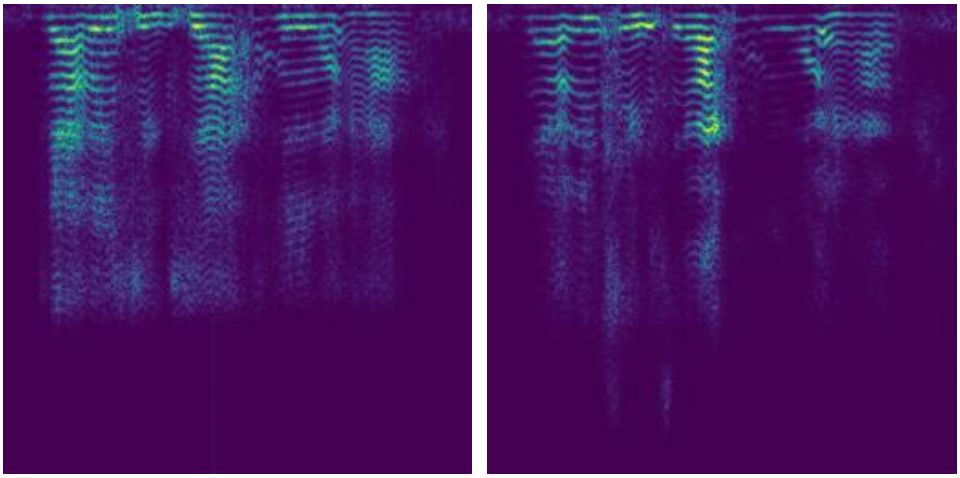

Fig. 5.11. Results of EVC using CycleGAN with different dimensions of the MCEPs. Spectrogram of the result trained by 16-dimensional MCEPs (on the left), and spectrogram of the result trained by 24-dimensional MCEPs (on the right)

5.2.3 Dataset Sizes

Surprisingly, the model trained on a significantly smaller dataset size performed way better than we expected according to Fig 5.12. The reason for this might be that angry speeches had very distinctive feature that could easily be captured by the network we built. The converted angry emotion could be observed in both models, while the model trained by 500 samples produced less noise.

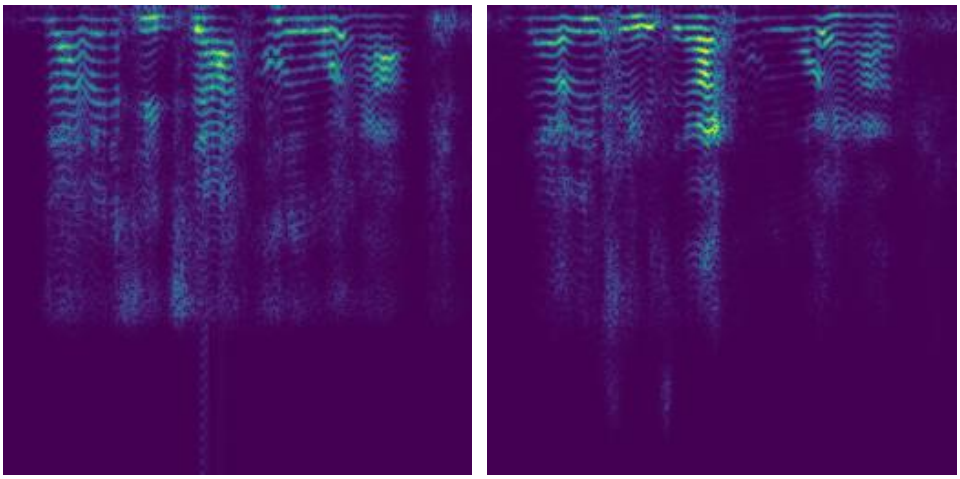

Fig. 5.12. Results of EVC using CycleGAN with different dataset size. Spectrogram of the result trained by 20 samples (on the left), and spectrogram of the result trained by 500 samples (on the right)

5.3 Objective Evaluation

5.3.1 MCD Evaluation

The MCD results were shown in Table. 5.1. Note that lower MCD values indicated better performance since MCD measures the distance between two sets of data. It was clear that CycleGAN produced the best result since the MCD value was the lowest. All the MCD results between the generated audio and the actual emotion audio were lower than the MCD results between the generated audio and the input audio, meaning that all the converted audios' features were closer to the prescribed emotions than the original emotions. It is worth mentioning that if the quality of the parallel training data was relatively high, the performance of CNN might be comparable to the performance of CycleGAN (refer to CNN angry to neutral conversion).

Table. 5.1. Comparison of MCD results of different conversions

	MCD (dB)	
Generated audio compared with	Input audio	Actual emotion audio
CNN neutral to angry	8.29	8.16
CNN angry to neutral	8.06	6.98
CNN sad to happy	9.27	8.26
CycleGAN neutral to angry	6.76	6.01

5.3.2 F0 Evaluation

F0 analysis evaluated the pitch of a speech, which directly reflected human listening experience. We witnessed that the pitches of the generated audios were all shifted toward the actual emotion audios. The result of CNN angry to neutral VC even had a nearly perfect match. The pitch shifting of CNN neutral to angry VC was really close to the pitch range of the actual angry audio, but according to Fig 4.2., the pitch range of fear audios was also merely below the pitch range of angry audios. Therefore, some listeners felt that the converted audio sound more familiar to fear emotion.

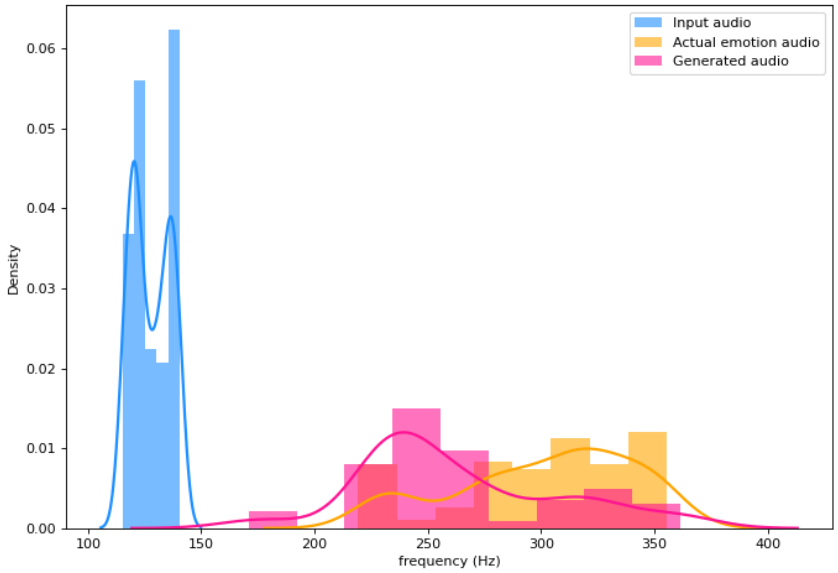

Fig. 5.13. Pitch analysis of CNN neutral to angry VC

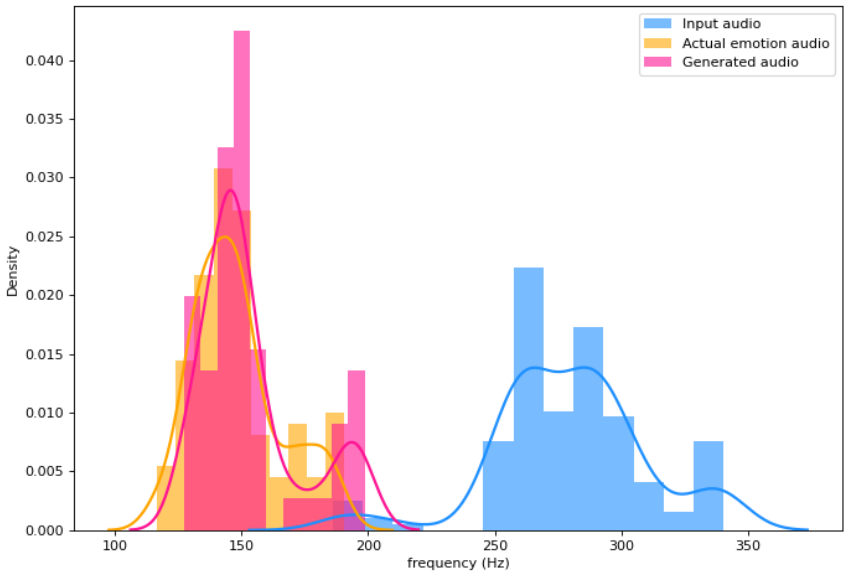

Fig. 5.14. Pitch analysis of CNN angry to neutral VC

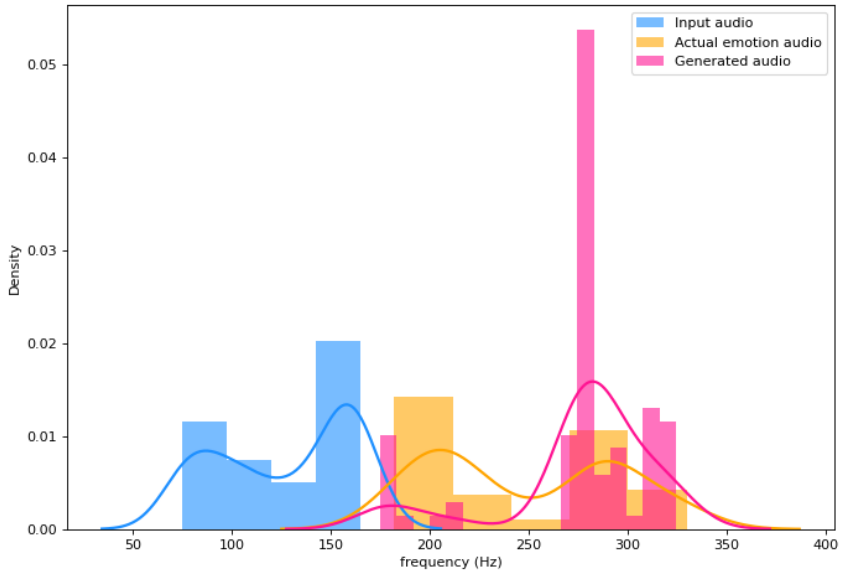

Fig. 5.15. Pitch analysis of CNN sad to happy VC

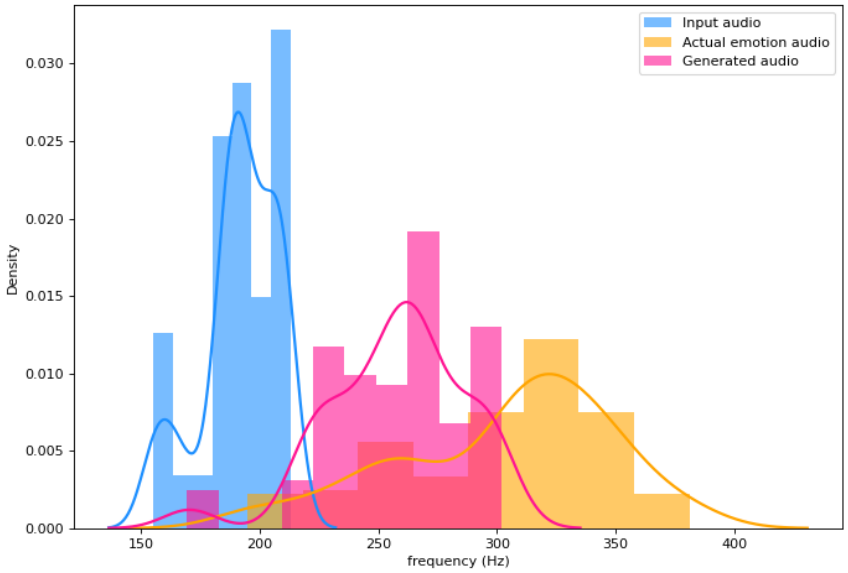

Fig. 5.16. Pitch analysis of CycleGAN neutral to angry VC

5.4 Subjective Evaluation

The results of subjective evaluation were presented in Table. 5.2. Out of our expectation, the converted audio using CycleGAN received the lowest MOS. Most of the listeners mentioned that although the noise generated by the CycleGAN model was minimized and the content was smooth, the converted features were too subtle to notice the difference in the emotion. The sample that acquired the highest MOS, CNN sad to happy EVC, received this result because of parallel data training. The parallel target happy audio had a rising tone at the end of the speech, which was converted in the generated audio. This was the key reason that made most of the listeners decided to give a high score.

Table. 5.1. Comparison of MOS results of different conversions

	MOS
CNN neutral to angry	3.2
CNN angry to neutral	3.5
CNN sad to happy	3.6
CycleGAN neutral to angry	2.5

This emotional voice conversion project was successful in comparing the results of other studies, which ranged from 3.0 to 3.5.

Chapter 6: Conclusion

This project builds up two deep learning models, i.e., CNN and CycleGAN, to converse emotional voice successfully. Two types of audio features, including the spectrogram and MCEPs, are used to train these two proposed models. The CNN model can be trained within a very short period and the performance is appealing to human hearing when adequate training data are provided. The CycleGAN model has a lower dependency on training data compared to the CNN model, while its performance has more potential when the training time increase. Given that the two EVC models are developed and their performances are acceptable, the aim of this project to perform voice conversion on specific features rather than copying all the information from the target data is accomplished.

Based on the research results, several possible implications in practice are discussed. As the two models have totally different advantages which can meet and satisfy the needs of various clients. For content creators such as singers or video producers, there will be chances that certain pieces of recordings contain flaws. Instead of reproducing the whole creation, it could be more efficient and cost saving to apply voice conversion. This could be tuning the acoustic features or fixing the technical problems occurred in the recording session. Also, in terms of medical use, voice conversion can assist those who have voicing problems. Our EVN models can help patients who damage their vocal cord but still be able to make slurring speech by converting their voice back to normal speech that contains correct emotions.

In spite of the contributions this project provides, there exists some limitations which can be taken for future research. First of all, it should be noted that emotional features are not fully isolated and speaker identities may be modified during the conversion. Further research can be implemented so as to solve this problem. Second, due to the nature of the research design, there leaves room for improving the specified-feature extraction. It is therefore encouraged to focus on enhancing the structure of our CNN model, thereby bringing the performance of CNN closer to CycleGAN. Thus, the improved CNN might reach a perfect balance between the performance and the training time. Further, it is suggested to implement TPU acceleration on training models to further minimise the training time since the major downside of CycleGAN is the long training sessions. Last but not least, it might be possible to combine deep learning and advanced voice conversion to study whether to translate the vibrations produced from throats and tongues into emotional speeches as some people can only do such an action.

References

[1] C. Heejin, P. Sangjun, P. Jinuk, H. Minsoo , "Emotional Speech Synthesis for Multi-Speaker Emotional Dataset Using WaveNet Vocoder," in *2019 IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, 2019.

[2] A. Gupta, "Speech-Emotion-Recognition-using-ML-and-DL," 2020. [Online]. Available: https://github.com/abhay8463/Speech-Emotion-Recognition-using-ML-and-DL. [Accessed 2 December 2020].

[3] T. Kaneko and H. Kameoka, "Parallel-Data-Free Voice Conversion Using Cycle-Consistent Adversarial Networks," arXiv, 2017.

[4] T. Nakashika, T. Takiguchi and Y. Ariki, "Voice conversion based on speakerdependent restricted Boltzmann machines," *IEICE Trans. Inf. Syst,* vol. 97, no. 6, pp. 1403-1410, 2014.

[5] Y. Stylianou, O. Capp´e and E. Moulines, "Continuous probabilistic transform for voice conversion," *IEEE Trans. Speech and Audio Process,* vol. 6, no. 2, 1998.

[6] T. Kaneko, H. Kameoka, K. Hiramatsu and K. Kashino, "Sequence-to-sequence voice conversion with similarity metric learned using generative adversarial networks," *Proc. INTERSPEECH,* pp. 1283-1287, 2017.

[7] T. Kaneko, H. Kameoka, K. Tanaka and N. Hojo, "CycleGAN-VC2: Improved CycleGAN-based Non-parallel Voice Conversion," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, 2019.

[8] K. Zhou, B. Sisman and H. Li, "Transforming Spectrum and Prosody for Emotional Voice Conversion with Non-Parallel Training Data," 1 February 2020. [Online]. Available: https://arxiv.org/abs/2002.00198. [Accessed 27 March 2021].

[9] L. A. Gatys, A. S. Ecker and M. Bethge, "A Neural Algorithm of Artistic Style," *Journal of Vision ,* vol. 16, no. 12, 2016.

[10] K. Zhou, B. Sisman, M. Zhang and H. Li, "Converting Anyone's Emotion: Towards Speaker-Independent Emotional Voice Conversion," 13 May 2020. [Online]. Available: https://arxiv.org/abs/2005.07025. [Accessed 22 March 2021].

[11] H. Kameoka, T. Kaneko, K. Tanaka and N. Hojo, "StarGAN-VC: Non-parallel many-to-many voice conversion with star generative adversarial networks," in *2018 IEEE Spoken Language Technology Workshop (SLT)*, Athens, 2018.

[12] K. Hao, "What is AI? We drew you a flowchart to work it out," MIT Technology

Review, 10 November 2018. [Online]. Available: https://www.technologyreview.com/2018/11/10/139137/is-this-ai-we-drew-you-a-flowchart-to-work-it-out/. [Accessed 29 March 2021].

[13] K. Hao, "What is machine learning?," MIT Technology Review, 18 November 2018. [Online]. Available: https://www.technologyreview.com/2018/11/17/103781/what-is-machine-learning-we-drew-you-another-flowchart/. [Accessed 1 April 2021].

[14] "What Is Reinforcement Learning?," The MathWorks, [Online]. Available: https://www.mathworks.com/discovery/reinforcement-learning.html. [Accessed 1 April 2021].

[15] "What Is Deep Learning? 3 things you need to know," The MathWorks, [Online]. Available: https://www.mathworks.com/discovery/deep-learning.html. [Accessed 20 March 2021].

[16] J. Brownlee, "What is Deep Learning?," Machine Learning Mastery, 14 August 2020. [Online]. Available: https://machinelearningmastery.com/what-is-deep-learning/. [Accessed 29 March 2021].

[17] T. Giannakopoulos, "Intro to Audio Analysis: Recognizing Sounds Using Machine Learning," HACKERNOON, 13 September 2020. [Online]. Available: https://hackernoon.com/intro-to-audio-analysis-recognizing-sounds-using-machine-learning-qy2r3ufl. [Accessed 20 March 2021].

[18] K. Choi, "Deep Learning with Audio Signals: Prepare, Process, Design, Expect," InfoQ, 10 July 2019. [Online]. Available: https://www.infoq.com/presentations/dl-audio-signal/. [Accessed 23 March 2021].

[19] G. Mendels, "How to apply machine learning and deep learning methods to audio analysis," Medium, 18 November 2019. [Online]. Available: https://towardsdatascience.com/how-to-apply-machine-learning-and-deep-learning-methods-to-audio-analysis-615e286fcbbc. [Accessed 21 March 2021].

[20] D. Rothmann, "What's wrong with CNNs and spectrograms for audio processing?," Medium, 26 March 2018. [Online]. Available: https://towardsdatascience.com/whats-wrong-with-spectrograms-and-cnns-for-audio-processing-311377d7ccd. [Accessed 10 March 2021].

[21] A. Sabra, "Learning from Audio: Spectrograms," Medium, [Online]. Available: https://towardsdatascience.com/learning-from-audio-spectrograms-37df29dba98c. [Accessed 9 March 2021].

[22] P. S. Lobel and K. E. Kovitvongsa, "Convenient Fish Acoustic Data Collection in

the Digital Age," in *Proceedings of the American Academy of Underwater Sciences 28th Symposium*, Dauphin Island, 2009.

[23] Crumpton and C. Bethel, "A Survey of Using Vocal Prosody to Convey Emotion in Robot Speech," *International Journal of Social Robotics,* vol. 8, no. 2, pp. 271-285, 2015.

[24] D. Gartzman, "Getting to Know the Mel Spectrogram," Medium, 20 August 2019. [Online]. Available: https://towardsdatascience.com/getting-to-know-the-mel-spectrogram-31bca3e2d9d0. [Accessed 2 March 2021].

[25] "Neural Networks," IBM, 17 August 2020. [Online]. Available: https://www.ibm.com/cloud/learn/neural-networks. [Accessed 14 March 2021].

[26] S. Saha, "A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way," Medium, 16 December 2018. [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53. [Accessed 4 March 2021].

[27] R. Asokan, "Neural Networks Intuitions: 2. Dot product, Gram Matrix and Neural Style Transfer," Medium, 25 January 2019. [Online]. Available: https://towardsdatascience.com/neural-networks-intuitions-2-dot-product-gram-matrix-and-neural-style-transfer-5d39653e7916. [Accessed 15 March 2021].

[28] J. Brownlee, "A Gentle Introduction to CycleGAN for Image Translation," Machine Learning Mastery, 5 August 2019. [Online]. Available: https://machinelearningmastery.com/what-is-cyclegan/. [Accessed 25 March 2021].

[29] J.-Y. Zhu, T. Park, P. Isola and A. A. Efros, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks," in *IEEE International Conference on Computer Vision*, Venice, 2017.

[30] Z. Luo, J. Chen, T. Takiguchi and Y. Ariki, "Emotional voice conversion using neural networks with arbitrary scales F0 based on wavelet transform," *Journal on Audio, Speech, and Music,* vol. 18, 2017.

[31] P. Kamp, "Mean Opinion Score (MOS)," Twilio Docs, [Online]. Available: https://www.twilio.com/docs/glossary/what-is-mean-opinion-score-mos. [Accessed 2 April 2021].

[32] K. Dupuis and M. K. Pichora-Fuller, "Toronto emotional speech set (TESS) Collection," University of Toronto, 2010. [Online]. Available: https://tspace.library.utoronto.ca/handle/1807/24487. [Accessed 10 December

2020].

[33] S. R. Livingstone and F. A. Russo, "The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English.," 5 April 2018. [Online]. Available: https://zenodo.org/record/1188976. [Accessed 10 December 2020].

[34] U. Valainis, "Audio emotions: Sorted audio emotions from 4 data sets.," kaggle, June 2020. [Online]. Available: https://www.kaggle.com/uldisvalainis/audio-emotions. [Accessed 16 January 2021].

[35] "Understanding Mean Opinion Scores," Broadcom, 17 July 2019. [Online]. Available: https://techdocs.broadcom.com/us/en/ca-enterprise-software/it-operations-management/unified-communications-monitor/4-3-1/reference/understanding-mean-opinion-scores.html. [Accessed 30 January 2021].

[36] N. Lab, "Pitch-Tracking, or How to Estimate the Fundamental Frequency in Speech — on the Examples of Praat, YAAPT, and YIN Algorithms," Medium, 3 August 2018. [Online]. Available: https://medium.com/@neurodatalab/pitch-tracking-or-how-to-estimate-the-fundamental-frequency-in-speech-on-the-examples-of-praat-fe0ca50f61fd. [Accessed 10 December 2020].

[37] "librosa.stft," librosa , [Online]. Available: https://librosa.org/doc/0.8.0/generated/librosa.stft.html. [Accessed 15 January 2021].