Research Thesis

A thesis submitted in partial fulfillment of the requirements for the degree of master of computing

Automatic Assessment of Dysarthric Severity Level Using Audio-Video Cross-Modal Approach in Deep Learning

Han Tong

Unitec Institute of Technology

tongh02@myunitec.ac.nz

Supervisor

Associate Professor Hamid Sharifzadeh

Co-supervisor

Professor Ian McLoughlin

Abstract

Dysarthria is a speech disorder disease that can have a significant impact on a person's daily life. Early detection of the disease can put the patient into therapy sessions more quickly. Researchers have established various approaches to detect the disease automatically. Traditional computational approaches commonly analysed acoustic features like Mel-Frequency Cepstral Coefficients (MFCC), Spectral Centroid, Linear Prediction Cepstral (LPC) coefficients and Perceptual Linear Prediction (PLP) from speech samples of patients to detect dysarthric speech characters like slow speech rate, short pauses, mis-articulated sounds, etc. Recent research has shown that some machine learning algorithms can also be deployed to extract speech features and detect the severity level automatically.

In machine learning, feature extraction is a crucial step in dealing with classification and prediction problems. For different data formats, different well-established frameworks have been developed to extract and classify the corresponding features. For example, for an image data processing system, Convolution Neural Network (CNN) can provide the underlying network structure for the system to analyse the video data to obtain the visual features. In contrast, for audio data processing system, Natural Language Processing (NLP) algorithms can be applied to obtain acoustic features. Therefore, the selection of the framework to be used mainly depends on the modality of the input. As early steps in development of machine learning approaches for automatic assessment of dysarthric patients, classification systems based on audio features have been considered in literature; however, recent research efforts in other fields have shown that using an audio-video cross-modal framework can improve performance of the classification systems.

In this thesis, for the first time, an audio-video cross-modal framework is proposed using deep-learning algorithm that the network takes both audio and video data as input to detect severity levels of dysarthria. Within the deep-learning framework, we also propose two network architectures using audio-only or video-only input to detect dysarthria severity levels automatically. Comparing with current one-modality systems, the deep-learning framework yields a satisfying results. More importantly, comparing with systems based only on audio data for automatic dysarthria severity level assessment, the audio-video deep-learning cross-modal system proposed in this research can accelerate the training speed, improve accuracy and reduce the amount of required training data.

Acknowledgement

I would like to express my deep and sincere gratitude to my research supervisor, associate professor Hamid Sharifzadeh, Unitec Institute of Technology, for providing me the opportunity to do research. His dynamism, vision, sincerity and motivation have extremely inspired me. He has taught me the methodology to carry out the research and to present the research works as clearly as possible. It was such a privilege and honor to do research under his guidance. I am extremely grateful for what he has offered me. I would also like to thank him for his friendship, empathy, and great sense of humor. My sincere gratitude also goes to professor Ian McLoughlin, the co-supervisor of this research. His deep insights helped me at various stages of my research. I really appreciate his selfless support, guidance, and patience he provided to me during this research.

I am deeply grateful to my parents and all my family members for their love, caring and sacrifices for educating and preparing me for my future.

I would like to extend my thanks to my friends Ali Keivanmarz and Xavier Francis for their constant encouragement. I express my special thanks Dr Iman Ardekani, School of Computing and Information Technology, Unitec Institute of Technology, for his genuine support throughout this research work. I am extending my thanks to Cynthia Natalia Almeida, postgraduate academic administrator, Unitec Institute of Technology, for her support during my research.

Finally, my thanks go to all the people who have supported me to complete the research work directly or indirectly.

Han Tong

Table of Contents

List of Figures

List of Tables

1 Introduction

Dysarthria is a speech disorder disease, and people with dysarthria have difficulties for controlling their lips, tongues and other speech generation muscles. Therefore, the speech rate, intonation, and stress are not articulated as normal speech. Thus, dysarthria patients have lower intelligibility rate, which creates obstacles for the patients' daily communication. To evaluate the patient's progression in the root cause of the disease, doctors first need to know the severity level of the dysarthria. Depends on the severity levels, clinical decisions can be made regarding the medicines taken or the progress of therapy, as well as to arrange appropriate speech recovery sessions.

However, current assessment of dysarthria is mainly auditory based using subjective tests. The traditional subjective tests require the presence of a Speech-Language Pathologist (SLP), which is costly and time-consuming, and due to the assessment procedures, the reliability and validity may suffer.

The UASPEECH dataset [1] is an audio-visual database of dysarthric speech. To assess the severity levels of the patients participated in the UASPEECH recording, 5 native speakers joint the process. The native speakers listened to all the tapes of every patient and scored all the word spoken by the patients. Based on the overall score, the SLP can assign different severity levels to the patients accordingly. This process took a vast amount of human resources, and the result can still be unreliable due to potential human error may exist in the experiments. However, with the help of high computational power from computers and some practical machine learning algorithms, we should be able to find more effective and more reliable ways to accomplish this task. This motivates researchers to find new computational ways to assess the dysarthria severity levels.

In this research we use deep-learning framework to assess the severity levels of dysarthric speech. The proposed model have applied in three types of data in this research: audio-only, video-only and audio-video combined. Therefore, we first discuss the current techniques used in audio and video processing, as well as their adaptations in machine learning fields. For dysarthria assessment using audio data, a common approach is to extract acoustic features from the speech samples and use those features to detect the disease [2][3][4]. Additionally, some research has shown that the algorithms used in machine-learning areas can also be integrated into the system and help to improve the effectiveness and accuracy of the system [5][6].

Using audio data to assess the severity levels of dysarthria are closely related to some of the researches in the natural language processing field. So, some techniques used in natural language processing field can also be applied to this research. Some of the widely used algorithms, such as Support Vector Machine (SVM), Artificial Neural Network (ANN), Convolutional Neural Network (CNN), and Recurrent Neural Network will be introduced in the literature review chapter. The same algorithms have also been widely used in computer vision system to deal with visual-based data. Since our research used data contains both audio and video data, these research related to audio and image processing are also covered in the literature review chapter. In our study, based on the algorithms used in previous researches, we propose a system that for the first time takes both audio and video data at the same time The proposed system helps the severity levels of dysarthria patients to be assessed automatically. Our experimental results show that the proposed method works effectively and accurately.

Furthermore, a traditional neural network-based algorithm requires a vast amount of training data and a long training period. However, in general, dysarthric speech data is hard to be collected because of the sparsity of dysarthric speakers in the general population and the fact that it is hard for a dysarthric speaker to speak for a very long time. Although the UASPEECH dataset provided us with the training samples during the network training phase, comparing with to the traditional networks, the amount of training data is still limited. Therefore, to address data limitation issue, our proposed method using an audio-video cross-modal framework utilise a limited number of audio and video data more effectively comparing with current audio-based methods.

Patients' speech in UASPEECH dataset has been recorded using both cameras and microphones, which makes it possible for us to make use of both audio and video data. Our proposed system is divided into two different phases, the feature extraction phase and the classifier training phases. In the feature extraction phase, the acoustic and visual features are extracted from audio files and video files, respectively to form acoustic and visual feature vectors. Then, in the classification phase, classifiers are trained using the vectors generated in the feature extraction phase to predict the severity levels of dysarthric speech.

The rest of this thesis is organised as follows: chapter 2 is a literature review that introduces dysarthria related topics and researches, including some techniques of the disease classification. Then some widely used algorithms used in the NLP field for automatic dysarthric speech detection is reviewed. Then the algorithms used in computer-vision area for image/video processing is covered in the same chapter. Additionally, some audio-video cross-modal applications are discussed as well. Chapter 3 discusses implementation experiences in the proposed method realization process. Then, chapter 4 covers the implementation of the network in this research, which introduces our cross-modal architecture, and how it is used to detect the severity levels of dysarthria automatically. Finally, chapter 5 shows the experimental result and chapter 6 covers the conclusion of this research and some possible directions of future research.

2 Literature Review

This chapter reviews the research efforts relative to dysarthria and natural language processing and the techniques used in the machine-learning field for audio and video modalities.

2.1 Dysarthria

Verbal communication is the most common oral communication in our daily life. By sending air from our lungs to the larynx, various sounds or phonations can be generated. The detailed process of how speech generated in our brain can be found in [7][8][9]. The entire organ area behind the brain helps us interpret the text, including the angular gyrus structure of the parietal lobe of the brain (commonly referred to as the Wernicke area, island lobe leather, and basal ganglia). These areas can complement each other as a system network to process text and word order to determine the context and meaning of the text. This will train our ability to understand language. It also expresses the ability to understand language, which may be a way of expression. In addition to language comprehension, it also has the ability to generate language [10][11]. To express an opinion, we have to consider some words to express the meaning or information from the brain, and then form them into a sentence through some grammatical rules, and then use the human lung, vocal cords and mouth to articulate it. The frontal, high blood pressure and parietal regions of the brain help us shape what we want to say, and the motor cortex of the frontal lobe allows us to express language by coordinating the related muscles [12]. It can further be divided into 3 stages. The first stage is the conceptual preparation. At this stage, speakers will form concept to describe the object, and then to consider to associate a certain characteristic of the object with the current narrative [13]. This stage can be seen as a concept-action matching stage. The second is the syntax encoding phase. Pre-prepared semantic information will activate corresponding entries in the mental dictionary. Otherwise, they will be separated according to specific word order. In the process, the speaker must extract grammatical information, including part-of-speech grammar. The third stage is the speech coding stage. In this stage, the speaker gradually converts the semantic grammatical information into speech coding, which is finally executed by the motor system [14][15][16].

To generate a speech, we have to articulate the sound of phonations first. The phonations are the basic elements of a speech. Dysarthria is a speech disorder disease where the muscles responsible for speech-generation are too weak, or the speaker has difficulty controlling them [17]. The main symptom of dysarthria is that patients have trouble speaking normally, such that their speech has low intelligibility [18]. The root causes of dysarthria can be some impairment to the structures of the central nervous cells or peripheral nervous cells which are responsible for controlling the muscles related to speech generation. The symptoms of dysarthria may have increased respiration frequency, small pauses in speech, and some may have a hoarse voice, slower speech rate, pitch and volume variations, and mis-articulated sounds[19]. Moreover, some severe physical injuries or disabilities can also be related to dysarthria [20]. Based on the locus and the symptoms of the disease, dysarthria can be categorised into the following types, spastic, flaccid, ataxic, hyperkinetic, hypokinetic and mixed [21]. Specifically, dysarthria can be categorised into different types, including flaccid, spastic, ataxic, hyperkinetic, hypokinetic and mixed. Spastic, which is caused by bilateral damage to the upper motor neurons [22]; flaccid, which is a result of bilateral or unilateral damage to the lower motor neuron; ataxic, which results from damage to the cerebellum unilateral upper motor neurons [23][24]. Hyperkinetic and hypokinetic which often results from damage to parts of the basal ganglia, such as in Parkinsonism [25].

Nevertheless, the most common type of dysarthria is mixed dysarthria [26], as neural damage including multiple strokes and traumatic brain injury resulting in dysarthria that often influence more than one nervous systems. Moreover, certain degenerative illnesses, such as amyotrophic lateral sclerosis (ALS), often cause damages to various areas of the nervous system[27][28]. Some researchers also indicate that the disease can also have influences on other areas, such as early childhood recognition, sensory defect, and intellectual disability

[29][30][31][32]. However, depends on the severity levels of the damage to different areas, we can still define the leading root cause of the disease, and based on that we can group dysarthria into different types. Although in [33], authors indicate that there are more types of dysarthria (the authors listed 8 different types of dysarthria in the research), it is more common to classify dysarthria into the standards described in the literature review. Since the classification of dysarthria type is not in the scope of this research, we will also the more commonly used dysarthria type classification.

A summary of dysarthria types is shown in Table 2.1.

Table 2.1: Different dysarthria types

Dysarthria Types[34][35]		
Type	**Locus**	**Pathology**
Flaccid	Lower motor neurons	Breathy
Spastic	Upper motor neurons	Strangled or breathy
Ataxic	Cerebellar control circuit	Guttural
Hyperkinetic/Hypokinetic	Basal Ganglia	Reduced Sensitivity
Mixed	More than one	More than one

Table 2.1 shows the different types of dysarthria and their corresponding locus, which cause the disease and the voice pathologies.

Dysarthria can be a result of the damage of one nervous group, yet it is more common to see that damage is reflected in multiple speech-related motor-controlling systems [36]. The disease strongly influences individual's quality of lives. Depends on the severity of the disease, the symptom can be different. For patients with light dysarthria, they will have occasional articulation; however, for patients with severe dysarthria, the speeches can be completely incomprehensible [37]. Treatment at the early stages of the disease can have better performance and to reduce the negative influences of the disease, so patients should be put into therapy sessions as soon as possible. Early detection of the disease can put the patient into therapy sessions more quickly. Therefore, researchers have implemented automatic dysarthria detection systems, which will be introduced in the next chapter.

2.2 Traditional Computational Approach

Recent studies have shown that the techniques used in the NLP field can also be deployed to resolve the problems encountered in automatic dysarthria detection.

In 2006, Gunderson et al. proposed an automated digit recognition systems to recognize the isolated spoken digits by talkers suffering from spastic dysarthria [38]. The research explores the voice of three speakers with spastic dysarthria caused by cerebral palsy. The symptoms of low intelligibility are similar to all three experiment subjects, but reasons vary. The primary study shows that the three speakers are likely to lower or eliminate word-initial consonants; one speaker eliminates all consonants, and one speaker shows a severe stutter. Two different approaches were deployed to build an isolated-digit recognition system for the three speakers. A speaker-dependent phone-based hidden Markov model was used as the underlying structure for the first approach. The second approach was realized by using speaker-dependent support vector machines. Both models are trained and validated with the help of the HTK toolkit. As a result, the HMM-based recognizer was sufficient for two speakers but failed to recognise the speaker removing all the consonants. Comparatively, the recognizer using the SVM approach was effective for two speakers but failed with the speaker with a stutter. Although this research only used 10-digits as training input, it gives us a good direction for future research and shows us an example that we can also use computational approach for dysarthria detection/recognition.

Mujumdar and Kubichek [5] implemented a dysarthria system where global statistics of speech have been used as objective features. The low level, frame-based speech features such as pitch period and line-spectral frequencies are obtained first. Thus the local features of the audio data are computed. Then the high-level global features such as the mean, variance, skewness, and kurtosis of the features are used for speech quality predictions. They used 109 raw acoustic features such as MFCCs, LPC, Mel-filter bank, etc. and for each of the raw feature, five global statistics features were computed. A Bayes based classifier was trained using those features to detect dysarthric speech automatically.

Chmurzynska et al. suggested that computer of speech isolating sounds will contribute to the detection of neurological-related metrics [39]. This study describes the initial outcomes of the dysarthria-induced physiological changes. The vocabulary content collection was distinguished by the location and method of the articulation within the Polish phonetary structure. The clinical examination findings allowed specific markers of neurodegenerative disorders, such as dysarthria, which are the basis for developing an accurate examination model. The detection by machines of speech isolating sounds will contribute to the detection of neurological-related metrics. The research used MATLAB to help them to analyse voice stability and the changes in sound realization. The findings in this paper can be used as the feature descriptors of automatic voice recognition systems for future research.

Carmichael [6] implemented a computer-based application, the Computerised Frenchay Dysarthria Assessment Procedure (CFDA), to detect symptoms of dysarthria. The system has a graphical user interface that enables the patients to use them without the presents of an SLP. The program will take the users voice as input and generate information about the severity of dysarthria as output. Mainly, the application used digital signal processing (DSP) techniques to process the speech and effectively detect two symptoms of dysarthria, respiration at rest and sustained phonation. By combining the two tests, the CFDA system was able to achieve a diagnostic accuracy of 85%. This research shows that some computational approaches can be deployed in the automatic dysarthria assessment system to improve the performance and reduce the manual interventions needed.

Ijitona et al. used extended speech feature extraction techniques which refer to the process of calculating the weighted averages of the formants extracted from an audio file [40]. The averaged formants are called centroid formants, and their weights are calculated based on the peak energies of the bands of frequency resonance, formants. After re-sampling, amplitude normalisation, and framing, Linear Prediction Coding (LPC) is applied to encode the signal and to extract the audio features. Although the experiment shows a satisfying result, other studies point out that the LPC filter will decrease the quality of the speech by reducing the bit rate [41]. Low or medium bit rate LPC can still provide reliable a method [42] for establishing the essential components of frequency characteristics of audible and comprehensible speech [43][44].

The acoustic and lexical models are also adapted in [45] to improve the performance of automatic dysarthric speech detection [46]. The analysis of articulatory and pronunciation errors is characterised and follows the intra-speaker patterns. The result shows that the word error rate (WER) is reduced by 36.99%.

The application of MFCC to detect dysarthric patients who have Parkinson's disease (PD) can be seen in [47]. Instead of using time-frequency based features as used in [48], MFCC was calculated to get the fingerprint for each patient. As mentioned in [49][50][51], the cepstral coefficients are uncorrelated and are very small for higher orders, therefore, in [47] authors used parameter lifter to enlarge the coefficients. A support vector machine with different types of kernels was used as a classifier and as a result, the experiment was able to achieve a classification accuracy of 91.17%

Hosom et al. implemented a structure using LPC to extract the frame-level voice features called centroid formants. Essentially, the centroid formants are computed through weighted averages of the speech frame-level formants [52]. Using this algorithm, they were able to achieve an accuracy of 75.6%. The training data consisted of 400 voice samples gathered from 10 dysarthric speakers and 10 healthy speakers. Another research [53] yielded a recognition rate of 98% and achieved 87% classification accuracy for dysarthric speech utterances. The features of the speech were obtained through MFCCs as well as other attributes such as peak amplitude, skewness, kurtosis, fundamental frequency and formants. The features are selected genetic algorithms, and the chosen features finally feed into a support vector machine for classification.

Other research has shown that some machine learning algorithms can also be used to detect dysarthria automatically. The machine learning approach for dysarthric speech detection will be introduced in the next section.

2.3 Audio-Based Predictive Model

Traditionally, a speech-language pathologist (SLP) analyses and categories normal and disordered speech by collecting information about the patients from speech ability measurement method such as subjective listen and reading test. These speech measurement methods are time-consuming, costly and need additional human intervention since usually the presents of a professional SLPs are required [54]. Therefore, there is a demand for a new algorithm that can assess and classify dysarthric speech automatically and provide result instantly. The new method should also offer economically-viable alternatives to patients which means to reduce the experts demand in this process.

Diwakar et al. constructed a system for detecting the repeated words in dysarthric speech [3]. In the research, both formants and MFCCs were used as acoustic feature vectors. To detect repetition in spoken dialogue environment, they used a neural network structure for phoneme recognition. The network was first constructed in [55]. Two scores were calculated to decide the repetition. The first score was obtained by measuring the cosine distance between the phoneme posterior vector of different frames. The second score was measured by the dynamic time warping (DTW) based alignment score. A deep neural network structure, Deep Belief Network-Deep Neural framework (DBN-DNN), is trained as a classifier. A similar approach can be seen in [56], where the authors used a DBN-DNN as a classifier to detect the disfluency in dysarthric speech. Instead of using formants and MFCCs, they used linear predictor coefficients (LPC) to form large dimensional feature vectors which could be analysed by the DBN-DNN. The method for exploiting pause duration, repeated and non-repeated words are reported in [57]. After establishing the mathematical model, they used Polynomial Curve Fitting (PCF)[58], which finds the best-fitting set data through matching the coefficients of a mathematical model that minimises the difference between the model and the data.

Nakashika et al. [59] proposed a machine learning approach for extracting dysarthric speech features based on a ConvNets which was implemented by Lee et al. in [60]. The network is called Convolution Bottleneck Network (CBN) which extends the ConvNet to extract dysarthria-specific features. Instead of using traditional acoustic features like LPC, MFCC and formants, they insert various types into the CBN including convolution layer, sub-sampling layer and bottleneck layer for acoustic feature extraction. As a result, the CBN-based feature extraction method achieved a much better result than the conventional feature extraction method and achieved a word-accuracy of 88%.

Carmichael implemented a computerised algorithm explicitly designed for dysarthric speech to measure the syllables-per-minute rate [61]. The system is integrated into the Computerised Frenchay Dysarthria Assessment (CFDA) suite for dysarthric speech assessment, but the system alone can be an assessment engine. The author suggests that in dysarthria assessment, the syllables-per-minute analysis has better performance than the phoneme level detections. This research shows that the acoustic features of dysarthric speech can be extracted at a higher level of a speech compared to that method that are extracting acoustic features from the frame-based voice unit.

Sayan and Seyed constructed an Automatic Speech Recognition (ASR) system that is dedicated to dysarthria patients [62]. They suggested in the research that features between normal speech, and dysarthric speech is different, so the ASR system needs to be changed adaptively to recognise the dysarthric speech better. They used the 10 digits recording from UASPEECH dataset to test the network. The network has an ANN-based array structure and takes fixed-length, word-based, speaker-dependent dysarthric speech as input. The idea is to use the ANN arrays as an ASR system. 12 MFCCs are extracted and fed into the ANN arrays. Compared to a single-learner approach, the algorithm in this research improves the recognition rate by 10.41% and reduce the error rate by 4.84%.

Another machine learning approach for automatic dysarthria assessment can be seen in [63]. The authors used a collection of auditory descriptors, including pitch, loudness, duration, spatial position and timbre. Timbre, as introduced in [64], is a multi-dimensional attribute in addition to other auditory descriptors. They used Short Term Fourier Transform (STFT) and harmonic based features as audio descriptors which were calculated through multi-taper spectral estimation. Then, a three-layer ANN has been trained as a classifier. The research used both UASPEECH and TORGO as a dataset. As a result, the average accuracy for the classification of 96.44% and 98.7% were achieved, respectively.

The above research show that machine learning algorithms are promising for the dysarthric speech detection problem. However, all of the study only used audio data to detect the problem. The detection of dysarthric speech using video data is currently absent. However, a visual based machine learning algorithm is widely used in solving other computer-visioning problems, such as action recognition [65], emotion analysis and lip-reading [66][67], image classification [68][69], etc.

2.4 Video-Based Predictive Model

Using CNN or a Recurrent Neural Network (RNN) as a basic structure to develop a system to extract features from video data is still an active topic in the computer vision field [70][71]. In this section, some algorithms used in the computer vision field will be introduced.

A huge amount of research has been done in recent years, trying to solve image/video related problems using CNN as an underlying algorithm from different perspectives. It can be applied to solve action recognition problems, such as crowd behaviour recognition and sports recognition, as shown in [72][73][74]. Some networks can also help to locate regions of interest (ROI), so that only the portion that contains the interesting features will be analysed, as shown in lip movements/lip reading problems [75][76]. In [77], authors implemented a combined network of DCNN and RNN to understand the content in a video scene to classify the sports types. The system was based on the VGG-16 network, and similar research can be seen in [78] and [79]. Some multi-task/Hybrid CNN Structure can be seen in [80], and three-dimensional CNN combines with the RNN structure [66] for emotion detection. The CNN structure acts as an underlying foundation for many complex structures and improves the performance significantly on solving problems in many areas.

The You Only Look Once (YOLO) algorithm is the state-of-the-art technology used in current computer vision-related video processing problems. Since the size of an object in an image can vary, the performance of object detection task on one static image will suffer. The sliding window technique can solve this problem by applying a window scanning through the whole image to detect the object in the image. Once finished a sliding window with different size will be applied again to scan the image again. This process is repeated until the whole set sliding windows are applied. Thus, all the objects in the image will be captured. However, some windows may overlap with each other because one object might be captured by different sliding windows or capture by the same window for more than one time. Also, it requires a very high computation power to solve real-time problems. The YOLO algorithm can set a different threshold on the number of overlapping windows of an object. Only when the number of the overlapping window is larger than the threshold, the object can be identified. Additionally, since the YOLO network structure is based on CNN structures, so lots of shared parameters will be generated in the window sliding process. Thus, compared to other region proposal algorithm, such as fast RCNN, the amount of calculations needed is reduced, and the system can react faster. YOLO algorithm has applied on fire detection system on CCTV videos [81], pedestrian detection [82], intelligent tracking system for curling sport [83], and automatic detection of melanoma [84].

Ullah et al. implemented a novel action recognition system of handling video data through a CNN and a deep bidirectional LSTM (DB-LSTM) network structure[65]. In this paper, some very low-level features are first extracted on the frame level. The features are discontinuously extracted every sixth frame of the recording. Since the correlations between two adjacent frames are high, this step lowers the redundancy and complexity of the feature vector. Then, frame-based sequential knowledge is obtained by a DB-LSTM network, where multiple network layers are concatenated in both forward passes and backward passes so that the dimension of the DB-LSTM network is increased. The implemented system is able to obtain some long-term sequences features and able to handle videos with long duration by analysing the features for a certain time window. The experimental outcomes of this research yield remarkable improvements in the action recognition system. The proposed approach has been verified on three benchmark data sets including UCF-101 [85], YouTube 11 Actions [86], and HMDB51[87].

Ye et al. have constructed an improved CNN to classify video scene with complex backgrounds [88]. In the improved CNN system, the features are extracted from video scenes with complex background. The network was based on CNN [89][90][91] and Histogram Oriented Gradient (HOG), proposed by Dalal et al in [92], and scale-invariant feature transform (SIFT),proposed by Lowe in [93], algorithms. As a result, this proposed method was able to classify video scenes successfully. Similarly, in[94], researchers also proposed a solution to detect crowd behaviour also using SIFT algorithm. It is implemented via four steps, statistical computations of video, background information filtering, frame segmentation and classification. The system used a genetic algorithm to utilize the obtained frame feature set and used Matlab image processing toolbox and finally achieved precision near 0.99 and accuracy of 95%.

Ouyang et al. used a novel Multi-task Learning (MTL) algorithm to recognise the action in video clips [95]. The study has developed the MTL architecture in such a way that 3D convolutional neural networks (3D-CNN) and the long-term memory (LSTM) network are combined. In the series of those videos, each video was divided into several clips, and the deeper hybrid model 3D-CNN and LSTM was applied. The MTL model can thereby share visual knowledge more efficiently among various categories based on these video clip features. The MTL method can be based on tasks that are not identical, so that disjoint working groups are presumed. This led to the question, utilizing a regularisation process, that tasks within each group lie in a low subspace and do not overlap the subspaces shared by each group. They used two methods to overcome this problem: they permitted tasks in various groups to overlap by representing all tasks as the linear combinations of certain latent basic tasks as implemented in [96] and enforce standardisation of latent task parameter vectors to prevent too much systemic information sharing as shown in [97]. Three public action data sets have been used to test the proposed method: KTH dataset [98], UCF101 [85] and HMDB51 [87]. The experimental results showed that the MTL architecture could share detailed data across multiple action categories in video clips effectively and outperform other multi-task approaches.

Some non-CNN based systems are also implemented to solve video related problems. In [99] the author implemented system for sports detection. In the system, the athlete action database is used to detect the athletes from backgrounds. SIFT was then used to define the local features of the video sequences, to remove the point pairs and for two neighbouring video sequences, the residual mistakes are determined. Histograms of feature vectors are used to represent the video sequences to find similarity. The sports can then be identified by adding the scores of all histogram similarity tests from the previous step. Similarly non-CNN approach can be seen in [94] and [100]. Whereas in [94], SIFT was used as a basic algorithm in a real-time crowd behaviour detection system and in [100] Kanade-Lucas-Tomasi (KLT) was used to compute the optical flow on compensated frames to detect the group action in soccer videos.

Recent research shows that a system can take more than one modality as input, and the correlation between the two modalities can be calculated and used to make better predictions for the system. In the next section, the audio-video cross-modal system used in the deep learning field will be introduced.

2.5 Audio-Video Cross-Modal Systems

Currently, no audio-video cross-modal system has found to detect dysarthric speech. However, the cross-modal system has already been widely used in computer vision fields and often delivering state-of-the-art results.

Zheng et al. implemented a cross-modal of an audio-video embedding algorithm through Supervised Deep Canonical Correlation Analysis (S-DCCA) [101]. In this model, audio and video are projected into a shared area to address the semantic distance between audio and video. The system first only picked a trunk of an audio file that better synthesises the audio properties. This step was realized using an attention-based LSTM model. The long audio sequence was divided into small sequences with the same length. Each audio was broken into 72 sections, each three seconds apart. Then, each chunk was placed on the bi-directional LSTM platform. Later the property score of each chunk was computed, and the top scores were identified and selected to represent the whole audio section. A previously trained network was used directly to extract image features. Canonical Correlation Analysis (CCA) was used to find the correlation between the audio and video data. This is a classic approach for correlation computation between two or more modalities. Then Surprised Deep CCA projected the audio and video feature into the same domain and maximised the correlation between them. Through comparing, the system was able to retrieve the whole music video from the representative audio chunks.

An audio-video cross-modal structure was used to retrieve remote sensing images and audio in [102]. In the network, a large-scale remote sensing image dataset with a massive amount of human-labelled spoken audio captions was constructed for the cross-modal information retrieval task. A deep audio-visual system was built in a way that it can study the correspondence between audio and image directly. As a result, the network, Deep Visual-Audio Network (DVAN), has successfully retrieved visual data from audio data using the computed correspondence.

In [103], an audio-visual predictive cross-modal system was implemented. The system was able to predict the shape of the person's mouth from the corresponding speech signal, which gives a direction of how the audio and video can be correlated. Brutti and Cavallaro [104] has implemented a cross-modal model adaptation system for the audio-visual target recognition task. The framework instinctively and iteratively adapted, using the back of the other modality, the time-based models of each model to varying target appearance and environmental conditions. The modification process was unsupervised and carried online; thus, as new unlabeled data becomes available, the models can be enhanced. In the multimodal model adaptation process, the classification probabilities were represented as scores with a normalised scale for different modalities. In the score fusion process, the combined score will be accepted only when the joint score exceeded the threshold. The trained multimodal system successfully performed person recognition task over the QM-Seminars dataset [105] and the QM-GoPro Dataset [106].

Roy et al. used speaker-specific information that is common to audio and video modalities to compare the speaker in audio recording to the same speaker in a video recording, when both recordings were made at different sessions [107]. In the research, the task was to align the unknown voice X audio recording with two unknown speakers video recording, one being x, with two unknown speakers, A and B, and vice-versa. This process refers to the XAB task. The XAB task consists of two critical phases: the learning phase, which refers to the process of mapping the speaker's identity between audio and video modalities, and the matching phase, which refers to the audio-to-video or video-to-video matching process. The video is framed to get 2D-DCT features, and MFCC is used to obtain the audio features.

Torfi et al. implemented a coupled 3D Convolution Neural Network (3D CNN) architecture for Audio-Visual Recognition (AVR) task [108]. The aim of this work is to find a solution to the problem of speech recognition when audio data become corrupted, or in a multi-speaker environment, to identify the right speaker. The essential idea is to calculate the correspondence between audio and video data via the selected features. For the speech modality, the traditional models like HMM and GMM with MFCC were not used for audio feature extraction, because in the last operation of MFCC, the DCT, which will eliminate the correlations between the coefficients of energies as well as the order of the filter-bank energies. Therefore, the last step of MFCC calculation, the DCT transform, will disturb the locality property of the data. Instead, they used MFEC, which is similar to MFCC but got rid of the DCT operation in the last step. To form an audio-image cube, they used 20ms non-overlapping windows for generating the spectrum features, which were later used as images to train the DNN. For audio-video matching problems, the traditional CCA and CoIA were not used, they used DNN as classifiers. The lip's motion was selected as an image feature, and the extracted energy features for audio files were selected as audio features.

A cross-modal may make the best usage of the available data, therefore, compared to current existing diagnostic/predictive systems that usually require a considerable amount data to train the system, a cross-modal system may reduce the amount of training data needed. Therefore, high performance can be expected from a cross-modal system when the amount of training data is limited.

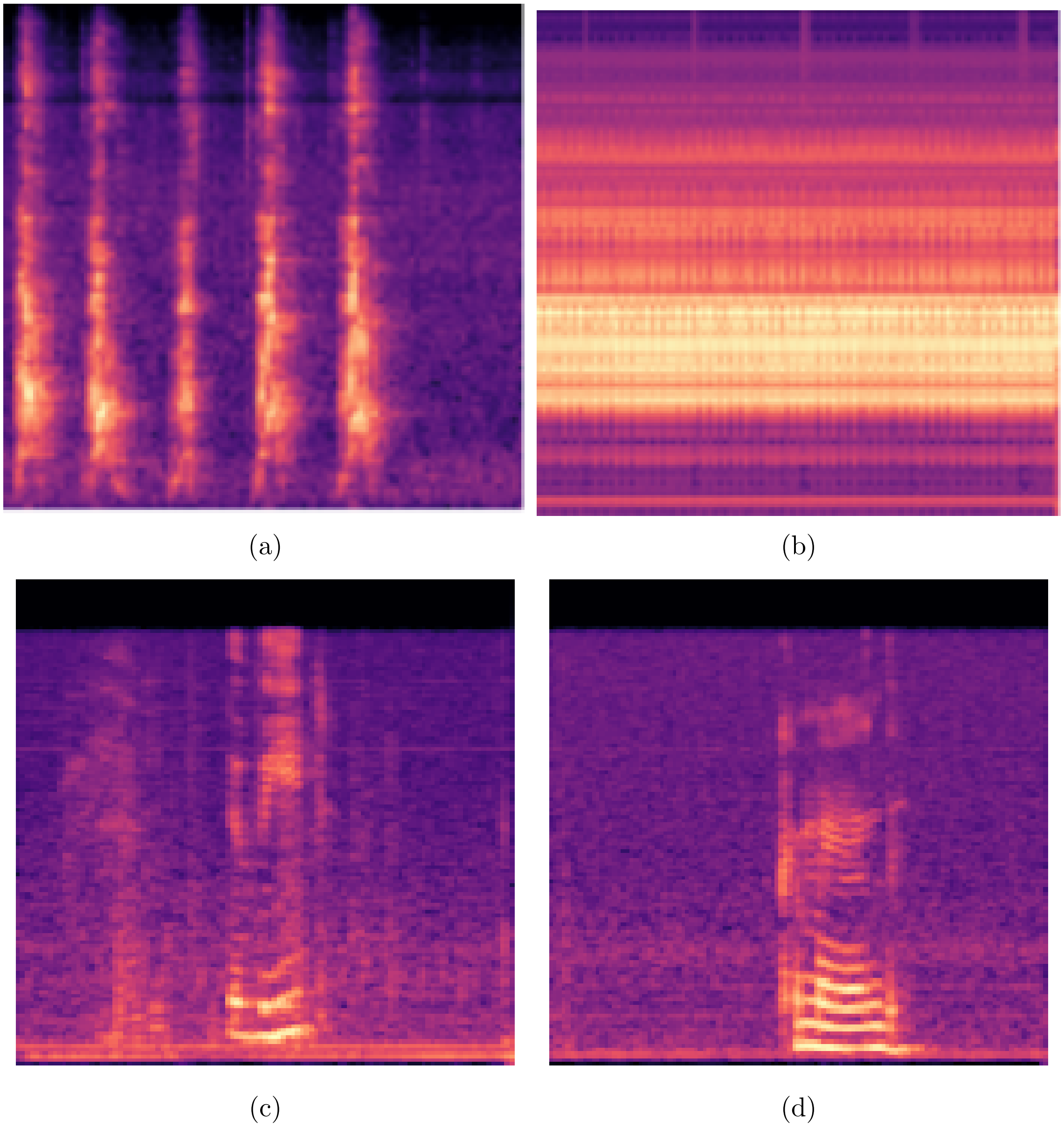
3 Research Gap and Approaches

As mentioned in the previous chapters, the automatic dysarthria severity level detection systems have received some research efforts, however, both the traditional approaches and the machine learning approaches used only one form of data as a sole resource for feature extraction. The most recent cross-modal architectures like in [101][104][107][108], are emphasising on the task of person recognition or information retrieval from audio and video files. Our proposed cross-modal system will extract features from both audio and video data and use the joint acoustic-visual features as training samples to train the system to detect the dysarthria severity levels automatically. The training samples for dysarthric speech severity level are usually restricted compared with other classification problems where training samples are plentiful. We are the first to introduce a cross-modal audio-video system to classify the severity levels of dysarthria. The UASPEECH dataset consists of both audio and video recordings which would be used for the purpose of this research. In this chapter, some of the approaches been used to achieve our research objectives are introduced. Some of the approaches did not offer positive results explicitly, but learning these approaches gives one a clear understanding of the algorithm/network properties. These experiences are beneficial when deciding the research directions. These research pathways should also be helpful in similar research projects. Therefore, in this chapter, some of the approaches have been tested within this research will are discussed while chapter 4 focuses on our proposed method which delivers the best result.

3.1 Discrete Signal and Voice Data

The speech data used in this research is saved in '.wav' format in computer storage device. The sampling frequency refers to how many samples are saved in one second, and the bitrate refers to how many bits in the memory space to represent one value in the data sample. The value of the data sample represents the amplitude of the voice. We used 16kHZ and 16-bit as our sampling frequency and bitrate. The research object of most NLP or ASR tasks is sound or speech recording [109][110]. Due to the quantum nature of computer storage hardwares, sound signals are stored discretely. Therefore, essentially, any sound signals saved on computers are discrete signals. Hence, some techniques used in discrete signal processing can also be applied to deal with voice signal processing.

To extract the features or properties from a voice signal, one technique that often used is Fourier Transform or Fast Fourier Transform (FFT) for discrete signals. The FFT technique can be seen as a technique that transfers discrete data points (voice recordings) from the time domain to frequency domain so that some properties that cannot easily be obtained from time domain will be easier to get in the frequency domain. Based on the Fourier transform, the energy spectrum can be computed. The format of the energy spectrum of voice data will be a 2D matrix. Thus, the energy spectrum of an audio recording can be plotted and visualized as an image. The intensity of the pixels represents the energy of the signal at the corresponding time and frequency. For different types of voices, for example, dog bark sound and siren sound will generate very distinctive plots. Based on the patterns of the spectrum, we can train a classifier to recognise the sound-making object, i.e., a dog or a car. The idea here is that the time-domain data is not directly used, but the information is transferred into the frequency domain, so the features are easier to capture. We have constructed a 2D-CNN network to classify the spectrums of speech recordings. However, by just providing energy spectrums to the network, the network was not able to classify the dysarthric speeches into different severity levels. This is because the system might be able to tell the difference between human talking and siren, but with all the data being speech recordings, it might not be able to tell the difference between two words. Figure 3.1 shows some visualised MFCC patterns of different types of sounds. The original sample files of the dog-bark sound and siren sound are downloaded from URBANSOUND8K DATASET [111]. This dataset contains 8732 labelled sound recordings (with duration less than 4 seconds) of urban sounds from 10 classes.



(a) (b) (c) (d)

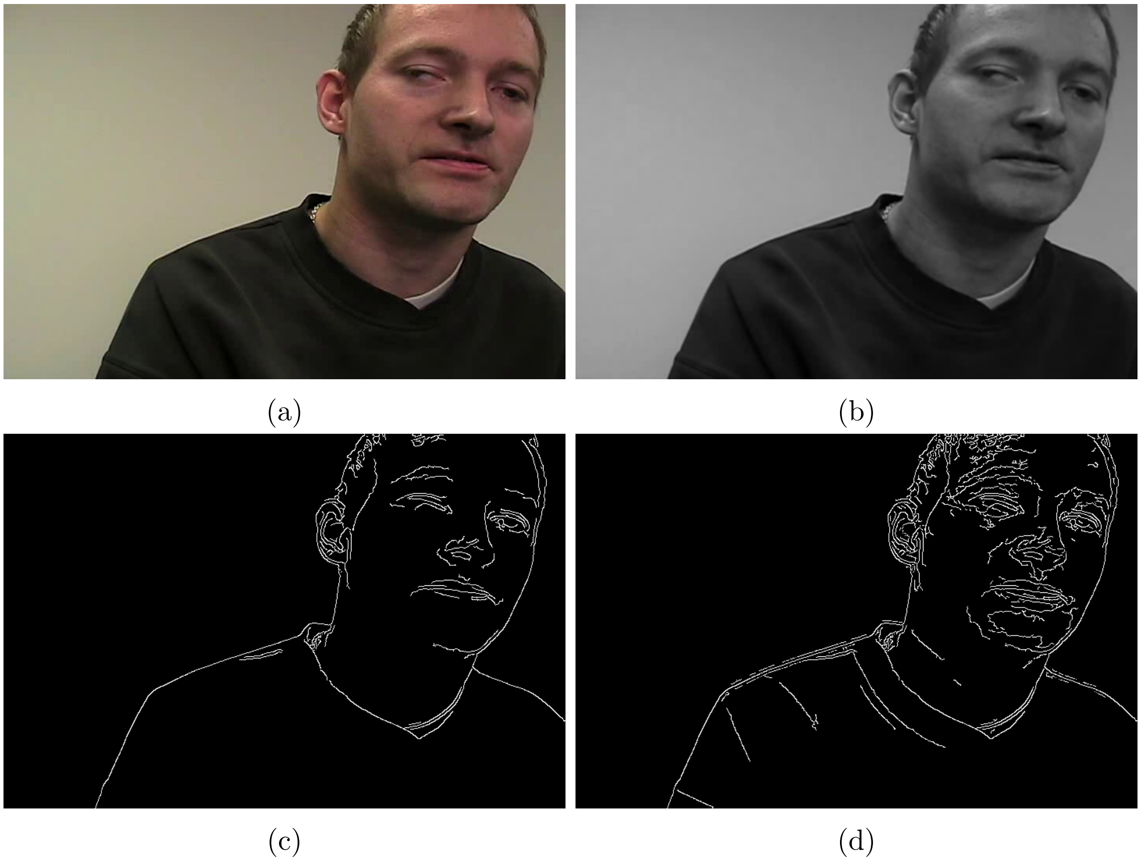
Figure 3.1: (a) Visualised patterns for dog-bark sound (b) Visualised patterns for siren sound (c) A visualised pattern for sample speech 1 (d) A visualised pattern for sample speech 2

As shown in Figure 3.1a, Figure 3.1b, and Figure 3.1c, the first figure shows the visualised MFCC pattern of a dog bark sound and the second one shows the pattern of a siren sound, and Figure 3.1c shows a visualised pattern for a speech sample used in this research. We can see from the figures that they have very distinctive and recognisable plotting styles. By speculating the figures, one can make predictions to identify the sound source with confidence. However, this does not hold between Figure 3.1c and Figure 3.1d. The patterns between two speech sample are almost identical, even though some details in the pattern are different. The similarity patterns apply to the speech data used in this research. All the recordings are from dysarthria patients, and they will have similar patterns. To be able to tell the differences between speech recordings, we need a new algorithm to extract more detailed information about the recordings.

Due to the phonetic nature of words, a different approach has to be taken to overcome this problem. The basic unit of speech is phoneme, and by combining them differently, we can get different recognizable words. In the similar way, people have implemented speech recognition systems [112][113][114][115]. In some speech recognition systems, they have a dataset or dictionary that maps the sound data to a phoneme. This way, the first-step recognition happens in the phoneme level, and the duration of the phoneme is in the magnitude of 10-100ms. By putting the phonemes together, the network then can make predictions about the input sound signal. However, the system will not have any other information about the speaker, such as gender, age, speed of talking, etc. Thus, algorithms work on low-level (short-time period) voice data can effectively obtain local features such as MFCC, LPC, and PLP, but usually lose some high-level global features. Some high-level statistic data such as peak frequency, skewness, and kurtosis contains information about the global feature, but they contain less local information compared to MFCC features. The purpose of this research is to assess the severity levels of dysarthria; thus, the recognition of word or identification of a speaker is not in our focus. The speech features extracted from speakers need to characterise the dysarthria severity levels. Thus, we implement a system that can extract the desired features to train a classifier to detect the severity levels of dysarthric speech. The discussion and implementation of the system are covered in the Research Methodology chapter.

3.2 Image Processing and Video Data

The video recordings used in this research have fps rate of 30, i.e., in each second of a video file; we can obtain 30 frames. Frames are the fundamental unit of a video file, and each frame can be processed as a static image. Thus, to extract information from a video file, we can start from image processing. An effective way to extract features from an image is to obtain the shape of objects in the image. Gaussian smoothing or Gaussian blur is widely used for image boundary detections. Some boundary detections techniques are highly sensitive to noises [116][117]. The Gaussian blur can reduce the noise and the details of an image. Essentially, it smears the intensity changes of an image. Then by calculating the intensity changes or gradient amplitudes and applying a threshold to those values, we can obtain the edges in the image. Boundary detection with different margins can outline the gaussian-blurred image differently [117]. In Figure 3.2, Figure 3.2a and Figure 3.2b show a frame image from a video sample and the gaussian-blurred version of that image. Figure 3.2c and Figure 3.2d show the images after boundary detections with different margin settings.


(a)

(b)

(c)

(d)

Figure 3.2: (a) An original sample frame obtained from video recording (b) Gaussian-blurred version of the captured frame (c) Boundary detection with high margin (d) Boundary detection with low margin

However, this also brings a problem as shown in Figure 3.2c and Figure 3.2d. A higher margin edge detection can filter out a lot of unwanted background information, but it also loses some information in the region that we are interested in. In Figure 3.2c, we can see that we have a rough shape of the speaker, but some critical facial information, such as the shape of the mouth is not clearly shown. An alternative way is to use a lower to margin for edge detection to capture those details so that we do not lose critical information on the image. Yet, as shown in Figure 3.2d, it outlines too many undesired details that will interfere with the facial expression analyse severely.

Additionally, even if we can get finely-lined boundaries of the speakers' facial expressions, we still need to further process the images. For the interest of this research, we need to focus on the facial expressions of speakers when they were doing the speech test. Label and identify the region of interest in these images are incredibly time-consuming and require a huge amount of manual intervention. A 10-mins video recording with 30 fps rate generate 180,000 images, so manually label all the images in a video recording is not a practical approach. Also, manual labelling is subjective to human errors which may lower the accuracy of the experiment. Thus, an automatic computational approach to recognise and label the critical facial points of speakers is needed to perform this task.

Despite the CNN structures have the proven abilities to extract features from images in various computer vision tasks, it demands enormous computational power. Reduce the dimensions of input images is a sufficient way to reduce the computational loads. A regular computer usually can not manage to accomplish massive graphical computing within a reasonable time. Originally captured videos often contain a vast amount of useless background area and noise signals in the image. Thus, a very crucial step in image processing is to find ways to capture the region of interest in the image and drop the extra areas to reduce the size of the image so that we can extract features more effectively.

3.3 Early Fusion vs. Late Fusion

For a classification system that take two types of input media, combining the extracted features of two inputs can greatly enhance the feature selection process leading to more promising results. The feature fusion can happen in the early stages of the late stages of the feature extraction process. One potential problem with early fusion is that the feature may not be normalised that their values are not in the same magnitude. The features in lower magnitude become negligible, or the feature values are in the same magnitude, but the size of the feature vectors have a huge difference. Similarly, the features with smaller feature vector size become trivial, and the final result merely rely on one type of data.

More discussions about early and late fusion can be found in [118][119]. In [118], the authors evaluated the influences of early and late fusion for plant species classification problem using flower images. The authors point out that some high-level features such as colour information can be obtained inside a pre-defined descriptor vector and computed at each colour channel. Then those features are concatenated after that. This process is the 'early' fusion because some of the simple features are combined at the earliest possible stage. Similarly, the late fusion refers to the idea to combine the obtained image-represent features at the latest stage before the features feed into a classifier. The authors also suggest that local spatial correlations are preserved in early fusion method. In contrast, in late fusion, the information is lost, and thus the features are expected to be less discriminative. Which fusion method to use depends on how important the spatial information is to the feature extraction process.

The early and late fusion can also be applied in generic indexing research [119]. The authors constructed two network structures with different feature group numbers to compare the effect of early and late fusion have to the generic semantics indexing problem. The result shows that different structures of the network and fusion stages will have a different influence on the effectiveness and robustness of the system. In this research, the features are first extracted using deep-learning architectures, and then late fusion is used to form audio-video joint features to train the classifier for dysarthria severity level assessment.

3.4 Data Pre-processing

The data pre-processing is a vital step in data preparation and sometimes underestimated by researchers. Some systems are highly sensitive to noises in data, so if the noise is not removed in the data pre-processing step, the system will not generate the expected result. Also, data pre-processing step can decide the attributes of data, such as dimensions of the image, size of the input vector. We used CNN structure for feature extraction for both audio and video data in our research. Most CNN network can not handle variable-length input data. Thus, we have to format out inputs to have identical dimensions so that they can feed into CNN networks. In this section, some approaches implemented in this research are discussed to show how data pre-processing can affect the experimental results.

- Data Formatting

For MFCC features, we use the delta and the delta-delta features together with MFCC. However, there are different ways of formatting the data, as discussed in [120]. In the implementation, the MFCC features and its delta and delta-delta's are stacked vertically. The MFCCs are computed to have a dimension of 13 by 650, i.e., 13 MFCC coefficients and 650 frames. After stacked them vertically, the dimension of the new matrix becomes 39 by 650 as shown in Figure 3.3.

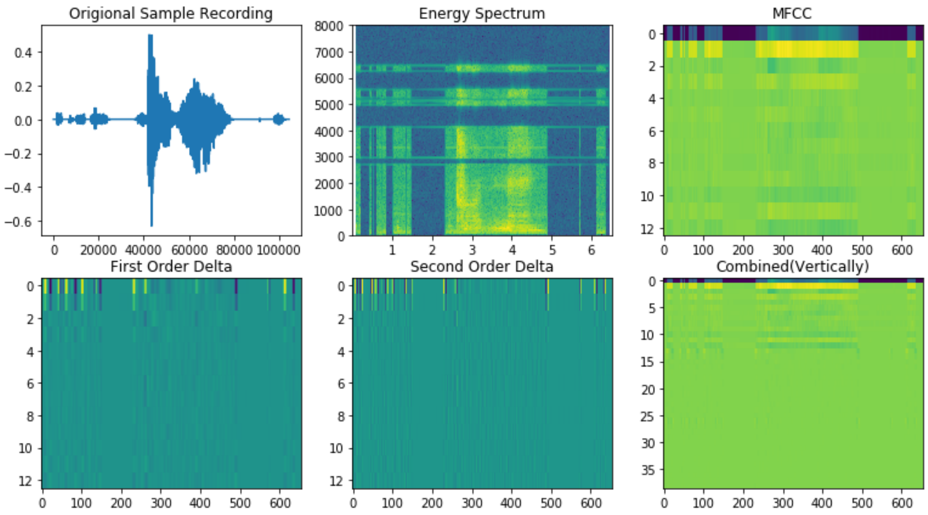


Figure 3.3: The waveform of an sample dysarthric recording, together with its energy spectrum, MFCC, delta coefficients of the MFCC, delta-delta coefficients of MFCC, and the vertically stacked MFCCs and its first and second order derivatives

The first graph on the top-left corner of Figure 3.3 shows the waveform of a random audio file from UASPEECH dataset. The rest figures are the energy spectrum of the file, the MFCCs of the file, the first-order derivatives of the MFCC, the second-order derivatives of the MFCC, and the vertically-stacked plot of MFCCs and its derivatives, respectively.

Different CNN network structures are tested for this data formatting (MFCCs + the first and second-order derivatives, stacked vertically), but none of them has generated promising result. The colours in the graph indicate the amplitudes of the pixels. The yellowish pixels represent larger values, and blueish pixels represent smaller values. From Figure 3.3, we can see that only the first few coefficients in the MFCCs got relative larger values. However, since the data was not normalised to the same level before concatenation, the values in the delta and delta-delta vector become trivial. Thus when the network is trained, use the combined matrix, the delta and delta-deltas' contribution to the network are almost negligible. Moreover, in the waveform and energy spectrum, we have some zero values. In some research, those areas can be eliminated to reduce the dimension of input data.

This shows an example of how audio data can be pre-processed before feeding into the network for feature extraction. Similarly, we evaluated different 3D-CNN network structures by inputting raw video recordings to the network and did not get a satisfying result. Therefore, the input data must be processed properly before the network can extract features from it.

- Phoneme Extraction

As shown in the literature review, frame-level acoustic features, such as phonemes, are giving good recognition results. We also attempted to extract phonemes in this research. As shown in [40], different phonemes have different shapes of formants. Thus, it is more reasonable to compare the acoustic features of the same phonemes. Therefore, we attempted to extract the same phonemes from speech recordings. For example, the [aɪ] sound in 'like', 'bike', 'five', 'right', 'nine' and etc. from the wordlist of the UASPEECH dataset.

However, this is hard to accomplish practically. Since we used a CNN network structure, the input data has to have identical dimensions. To extract the target phoneme with the same length is a very time-consuming task and requires enormous manpower. More importantly, the boundaries of phonemes in a speech is not well defined. So it is hard to just cut the target phoneme, it will contain pieces of information from adjacent phonemes or lose part of its own information. Besides, the acoustic characters can be affected by many factors, for example, the duration of a phoneme can be affected by the speed the speaker talks; the tone of the same phoneme can be different in a statement sentence than in a question... Additionally, the phoneme contains lots of local features but very limited global features of a speech. Thus we have to find a new method for acoustic feature extraction.

In this research, we considered both local and global characters of the speech and then used a late fusion technique to combine the acoustic and visual features. The magnitudes of the vectors are normalised to the same level, and the system is designed to let the feature vectors to have a similar size. The detailed implementation will be introduced in the next chapter.

4 Research Methodology

The purpose of this research is to automatically detect the severity levels of dysarthric speech. As mentioned, the dataset used in this research contains both video and audio files of the recordings of dysarthria patients. Since the video and audio data have different formats and stored differently in computer, different procedures should be taken to process them as described in the previous chapter. Audio data have are segmented into small fragments that each fragment contains only one spoken word from the wordlist. Video data are a series of images recorded when speakers read through the wordlist. After pre-processing, those data are ready for feature extraction. CNN networks are used for feature extraction for both video and audio data. Different dysarthria severity level features first are extracted from audio and video data and then are combined to form joint features. A classifier then is trained for dysarthria severity level detection using audio-video joint features. This chapter elaborates on how this proposed framework is implemented.

4.1 Objectives

- Data preparation

Speech recordings and video files can not directly used to train the system, they must be pre-processed to filter out unnecessary information and the input data has to be formatted before feeding into the system.

- Feature Extraction

Within deep-learining framework, apply acoustic and visual feature extraction techniques to obtain acoustic and visual features from audio and video file, respectively.

- Feature combination

After feature extraction, the acoustic and visual features are combined to form joint feature vectors. The joint feature vectors contain both audio and video features of dysarthric speech.

- Classifier training

Using audio-video joint features to train a fully-connected classifier to automatically detect different dysarthria severity levels.

4.2 Dataset

UASPEECH dataset [1] is used in this research. The UASPEECH is a dataset of dysarthric speech for research purposes. This dataset was created in 2013 at Illinois University. It contains isolated-word recordings of speakers with spastic dysarthria.

- Recording equipment

Data were collected in a lab environment. Subjects read isolated words from a computer monitor. A seven-channel microphone was used for recording audio data, and an additional 8th channel microphone was used to record the Dual-Tone Multi-Frequency signalling (DTMF) tone which later was used to segment the files. A digital camera was set up to record the speech process.

- Prompts

15 speakers (4 females and 11 males) were recorded in the experiment. A summary table of all available speakers' conditions is shown below in Table 4.1. As shown in Table 4.1, M and F in the speaker codes indicate the gender of the speaker: M for male, and F for female. The two digits after the letter are the code to identify the speaker. For example, F02 represents the second female speaker. M02, M03, M06, and F01 are not included in the dataset because either they were recorded under a different protocol or they did not approve redistribution of their data. For each speaker, 765 words were recorded in 3 blocks of 255. In each block, 155 common words are repeated, and the rest 100 are uncommon words. The 155 repeated words include 10 digits, 26 alphabet letters, 19 computer commands, and 100 most common words. The following are some examples of the prompt words:

Speaker Intelligibility Scores				
Speaker Label	Age	Speech Intelligibility (%)	Dysarthria Diagnosis	Severity Level
M01	>18	very low (15%)	Spastic	Severe
M04	>18	very low (2%)	Spastic	Severe
M05	21	mid (58%)	Spastic	Mild
M07	58	low (28%)	Spastic	Moderate
M08	28	high (93%)	Spastic	Low
M09	18	high (86%)	Spastic	Low
M10	21	high (93%)	Mixed	Low
M11	48	mid (62%)	Athetoid	Mild
M12	19	very low (7.4 %)	Mixed	Severe
M14	40	high (90.4%)	Spastic	Low
M16	-	low (43%)	Spastic	Moderate
F02	30	low (29%)	Spastic	Moderate
F03	51	very low (6%)	Spastic	Severe
F04	18	mid (62%)	Athetoid	Mild
F05	22	high (95%)	Spastic	Low

Table 4.1: Speaker intelligibility scores and severity levels

- Digits (10 words repeated 3 times): ”one, two, three, ...”

- Letters (26 words repeated 3 times): ”alpha, bravo, charlie,...”

- Computer Commands (19 words repeated 3 times): ”command, line, paragraph, enter,...”

- Common Words (100 words repeated 3 times): ”the, of, and,...”

- Uncommon Words (300 words): ”naturalisation, faithfulness, frugality,...”

- Intelligibility score

The speech intelligibility score is based on average scores given in listening test by 5 native speakers. The intelligibility ratings fall between 2% and 95%. Aligned with literature, we have classified the dysarthric speakers into four groups based on the speech intelligibility score, i.e. very low for 0-25%, low for 25-50%, medium for 50-75%, and high for 75-100%, the corresponding dysarthria levels are: severe, moderate, mild, and low. Table 4.1 below shows the detailed intelligibility scores of all the available speakers of the dataset.

From table 4.1, we can see that the intelligibility score ranges from 2% up to 95%. And based on those scores, one the dysarthria severity level will be asserted from the four predefined severity level. For example, M12 has a very low intelligibility score (7.4%), so the detected dysarthria level will be severe.

4.3 System Specifications

Python 3.0 was used as the programming language and deployed in windows 10 environment. A NVIDIA GeForce GTX 1080 Ti GPU was used to perform the computational tasks. The following packages/libraries are also used in this research: Tensorflow, dlib, librosa, Keras, and OpenCV.

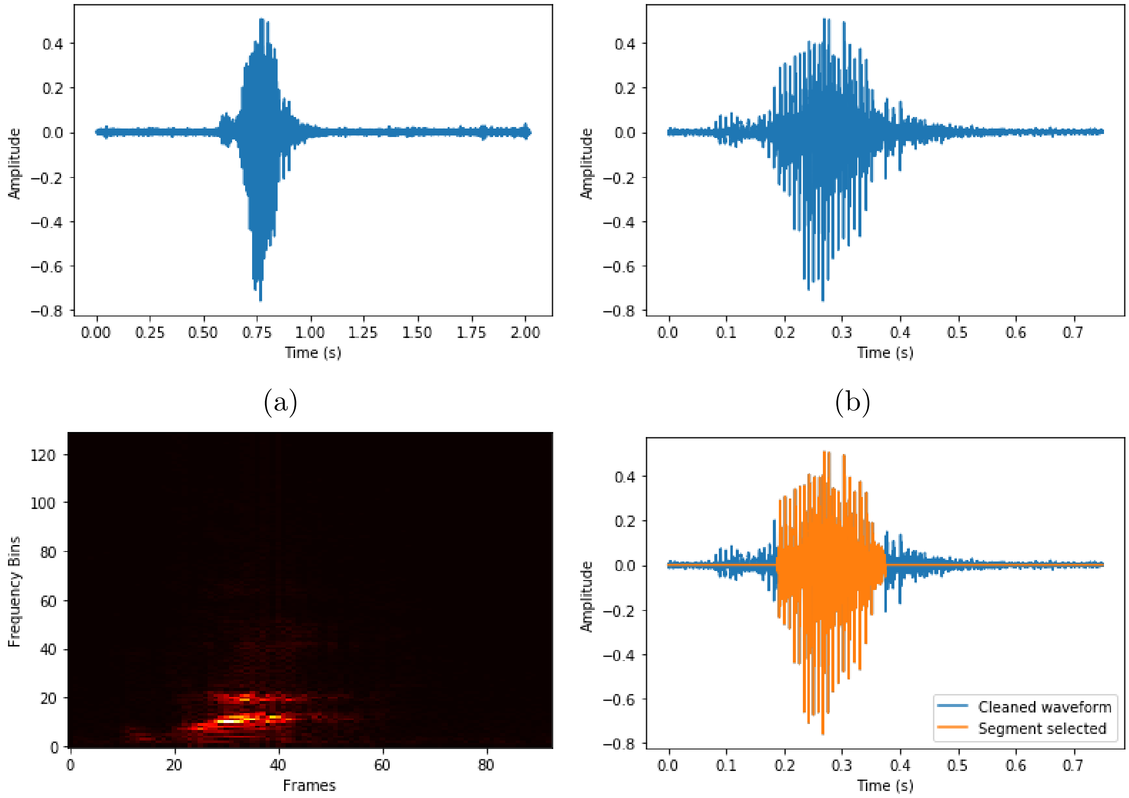
4.3.1 Audio Processing

- Data preparation

For each speaker, the records are recorded in 3 blocks; the specific recordings for each individual word are already separated by the researchers using DTMF tone. The recordings are saved in wave format(.wav sound files). The durations of the files range from 1 second to 20 seconds. Although the audio recording of each speaker was already segmented into fragments that each file contains only one word, the audio file still needs to be processed to remove unwanted information. From inspection, we find that most of the sound files contain a large portion of silences. For example, in the recording of word ‘Mike’ from speaker M11, the speaker was not making any pronunciation in both the beginning and the end of the file. Thus, those portions of the file contain only background noise which is useless and may cause errors in our analysis. Figure 4.1a shows the waveform of the word ’Mike’ from speaker M11. From Figure 4.1a, we can see that the useful information only occupied a very limited portion of the whole file. The beginning and ending part of the sound wave file consists of long silences. That information can be safely removed without affecting the performance of feature extraction. Figure 4.1 shows the waveform after removing those silence regions and

Figure 4.1c shows a plot of the MFCCs of the cleaned waveform. The brighter pixels means that the corresponding local waveform has higher energy.

The cleaned waveform shows in Figure 4.1b still contains some background noise. For some other files in the dataset, after dropping the long silences in the beginning and in the end, we still got some null regions in the middle. To remove that information, we calculated the energy of local regions. Based on the total energy of the local regions, we selected only the highest ones to make sure that that useless information is excluded for feature extraction steps. Figure 4.1d shows the selected segment. As shown in Figure 4.1d, the blue-coloured waveform is the cleaned waveform; the orange-coloured waveform is the waveform selected from local energy based calculation.



(a)

(b)

(c)

(d)

Figure 4.1: (a) Sample file: word 'Mike' from speaker M11 (b) Cleaned waveform of word 'Mike' (c) MFCCs of the cleaned waveform (d) Selected segment based on local energy calculation

- Features extraction

MFCC features are effective in most of the voice-related tasks. Thus, we used MFCCs as the main feature to analyse our data. The audio file has to be framed first to obtain the MFCC features. A very common setup for MFCC calculation is to use 40ms as window length and 15-25ms step length. We used 40ms window length and 25ms step size. The sampling rate when reading the wave files is 16kHz, so we have 640 data points per window and 400 points per step.

In addition to the MFCCs, the delta, and the delta-delta of the MFCC is also commonly used in the feature extraction step. The delta and delta-delta feature can be obtained by taking the first and second derivatives of the MFCCs. Our feature set includes MFCCs, the delta and the delta-delta of the MFCCs. There are different ways of putting these features together. As cussed in [120], the calculated result can be simply stacked horizontally or vertically together to form a feature vector, or they can be sorted by frame numbers. We used the same method suggested by researchers in [120], which the feature vectors are sorted by frame numbers.

4.3.2 Video Processing

- Data preparation

Similar to the audio recordings, the video files are divided by different blocks: 3 blocks for each speaker. The video is recorded at 29.969 fps. Since videos essentially are frame sequences, our basic operations are done at the frame level.

To process each frame, we need to first define the region of interest Which is the face of each speaker as we need to work on movements of facial muscles such as lip or eye movements. Thus, the first task is to identify the speaker's face from frames. To perform this, we use the face detector (a CNN based face detector with support vector machine) in the dlib library and deployed it in python environment to help us to capture the faces of the speakers in all the frames in the video recording.

- Features extraction

After the region of interest is defined, we can filter out other regions and focus on the facial region. The muscle movement is a significant factor when tracing the facial movement. In the Facial Action Coding System (FACS) [121][122], researchers summarised some important regions in a human face to recognise different facial emotions. We use similar analysis method for this research. We define two regions as our primary feature regions, mouth and eyes. Specifically, these regions are upper and lower lips, eyes, and eyebrows. To track facial muscle movement, we use facial landmarks. With the help of dlib library, we are able to draw facial landmarks in all captured faces. We use 68 landmarks to track the facial muscle movement, as shown in the figure below.

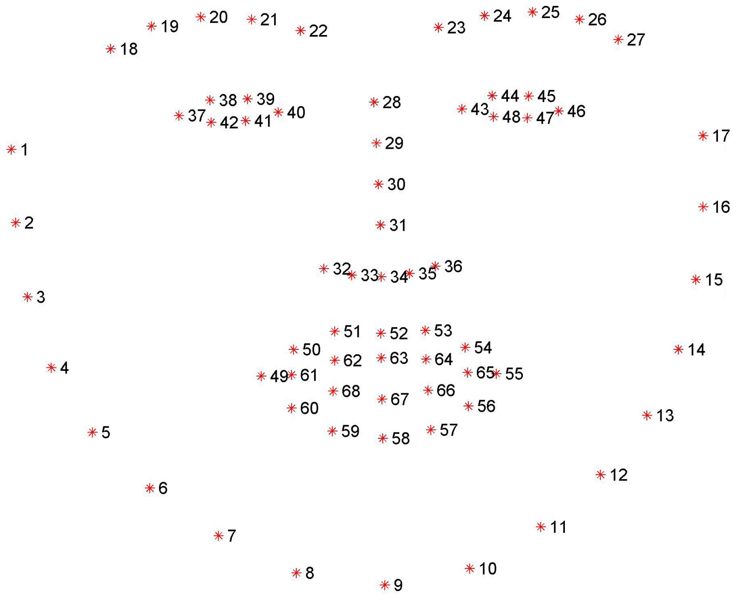


Figure 4.2: 68 Facial landmarks

Figure 4.2 shows the 68 facial landmarks, those landmarks outlines the shape of face and emotions on the face can be obtained accordingly. Key regions include jaw, eye and eyebrow, nose, and mouth. Here is an example of how this process is carried out in this research:

The following frame, as shown in Figure 4.3 is obtained from the video recording block 1 of speaker F02.

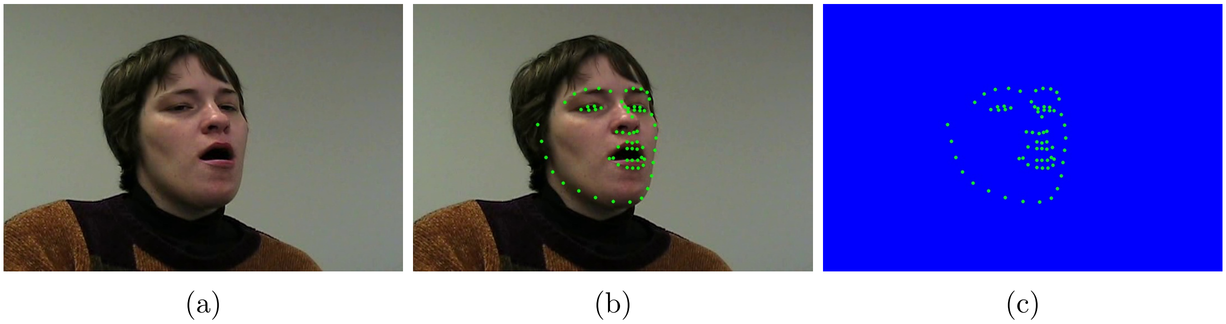


Figure 4.3: (a) An original frame from video recordings of F05 (b) Facial landmarks of speaker F05 (c) Background information is filtered, only key information about facial expression are left

For each frame of the video recording, the face capture function will be deployed and to capture the faces in the video stream. Then, landmarks are drawn on the frame to show the facial features. From this frame, first, the facial region is captured, and then the facial landmarks are drawn to track the facial movement. Figure 4.3b shows the frame after the face detection and the landmarks are outlined in the same frame.

Figure 4.3c shows a simplified version image. From the figure we can see that only the region of interest in the positions of facial landmarks (the green stars) are kept and the background information is filtered.

This same process can be applied to all the frames across all videos. And we can track the facial movement by using those facial landmarks. Then we divided the frames into different groups to form frame segments. Inside each segment, we can compare two adjacent frames to obtain the facial movement. After preparing the video and audio data, they can now be sent to the CNN network for feature extraction.

4.3.3 Combined Features

The MFCC features will be delivered to a 2D-CNN network for feature extraction. The CNN network has the structure of the input layer, convolutional layer, pooling layer and softmax layer. The features are extracted to generate an acoustic feature vector for each audio segment.

Similarly, to obtain visual features from video files, the facial landmarks features are sent to another CNN network that contains a similar structure. The video feature extraction channel is designed only for visual feature extraction. After feature extraction, the visual feature vector is obtained.

Depending on the experimental setup, the audio and video features can be used a sole or be used jointly, i.e. the audio features can be solely used for training the network for dysarthria severity level detection, and the same applies to the video features. Further details on different training arrangements are discussed in chapter 5, while our main focus here remains on joint feature vectors.

The two vectors (audio feature vector and video feature vector) can also be combined to form an acoustic-visual feature vector that contains both video and audio information from dysarthric speeches. Therefore, we need to train a neural network classifier to work on the combined feature vectors.

The classifiers we use to train the obtained features have similar structure. As shown in Figure 4.5, the layers on the left are the input layer which takes the feature vector, and the output layer generates a 4-digits one-hot coded vector to indicate the severity level of the dysarthric input speech, as [0001] for severe, [0010] for moderate, [0100] for mild and [1000] for low levels.

4.4 Network Diagram

This section describes the overall data flow of the network. We also provide the mathematical expression of the core structure of our proposed network architecture - the CNN networks to show how audio data and video data are processed. Additionally, some examples are given to show how some of the network parameters are tuned in the implementation process.

As shown in Figure 4.4, the audio and video files go through different paths through the network. After pre-processing, the MFCCs are calculated for all the audio files to form acoustic feature cube.

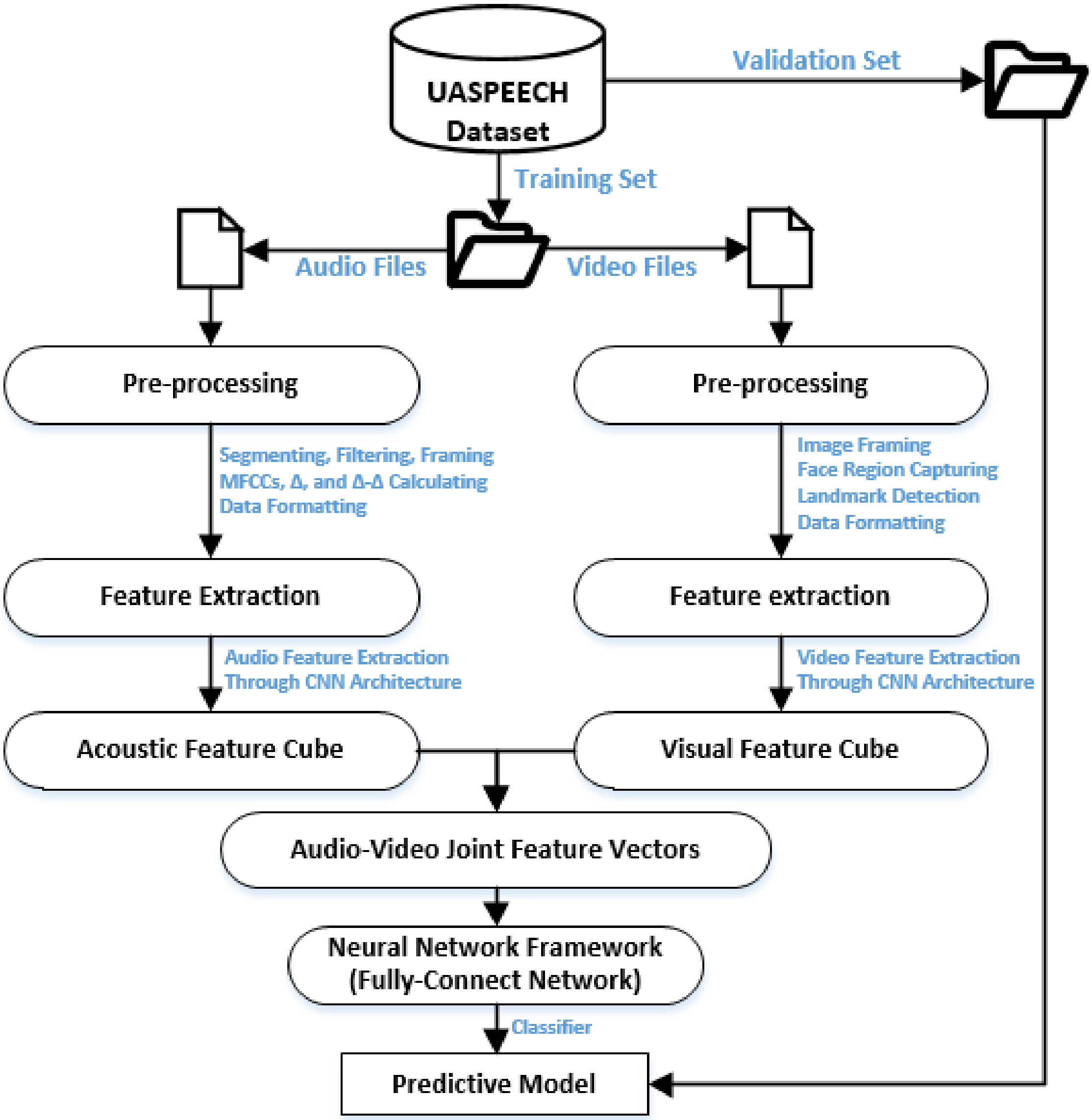


Figure 4.4: Overall network diagram

Similarly, the visual feature cube are also formed after feature extraction. For a cross-modal classifier, the acoustic feature cube and the visual feature cube are combined to form an audio-video feature vector. Finally, through a supervised learning approach, a fully-connected neural network classifier is trained to detect the severity levels of dysarthria patients based on joint vectors.

- Convolutional Neural Networks

Since the Convolutional Neural Networks (CNNs) are the underlying networks that being used in both feature extraction and classification of our proposed architecture, we would like to give more information about the CNNs. However before going deep into the network, we provide some fundamental mathematical expressions of the CNNs. More discussions about the CNN networks can be find in [123][124][125][126].

A CNN network typically consists of an input layer, convolutional layers, pooling layers, fully-connected layers, and a output layer. Convolutional layers and pooling layers are often used together, and are sometimes referred to convolution layers.

The input layer can take inputs with more than one dimensions, such as 2-D images and audio data. As with other neural network algorithms, the Gradient Descent (GD) algorithm is used in the learning process, so the input features of the convolutional neural network need to be normalised. Specifically, the input data must be normalised before the learning data is fed into the convolution neural network. For example, for grayscale images where the input data is a collection of pixel values from [0-255], the normalisation process can convert the values to the [0-1] interval to enhance the calculation efficiency and the performance of CNNs.

Then, input data will be forwarded into convolution layers. The function of the convolution layer is to extract features from the input data. Each convolution layer contains many convolution kernels or convolution filters. Each number that composes the filter corresponds to a weighted coefficient and a bias vector, similar to a feedforward neural network. Each neuron in the convolutional layer is connected to multiple neurons in an adjacent region in the previous layer. The size of the region depends on the size of the convolution kernel. The following equation is used in convolutional layers when extracting features from input data:

$$Z^{l+1}(i,j) = \sum_{k=1}^{K_l} \sum_{x=1}^{f} \sum_{y=1}^{f} \left[Z_k^l \left(s_0 i + x, s_0 j + y \right) w_k^{l+1}(x,y) \right] + b \qquad (1)$$

In equation 1, b is the bias vector, Z^l and Z^{l+1} represent the input and output of the convolution layer, they are often refered as feature maps. $Z^{l+1}(i,j)$ corresponds to the pixels in the feature map. k is the channel number in the feature map.f, s$_0$, and p are the parameters of the convolution layer, namely, they are the size of the convolution kernel, the stride of the convolution, and the number of padding layers, respectively.

Those parameters in equation 1 can be manually selected when implementing the network. There is no fixed rule for how to select some of the parameters such as number of network layers and the size of filter used in each layer. Trial and error approach is used to check the perfromance of different parameter settings. In our implementation, we have tested differnet numbers of convolutional layers for different input data. As shown in table 5.1, for audio data, we have tested 2 and 3 convolution layers for audio feature extraction process. For video data we tested both 3 and 4 convolutional layers for video feature extraction. We used 3 by 3 convolution kernels throughout our entire network structure since 3 by 3 is a common setting for filter size in CNN based network structures.

In some neural networks, each node receives the input data and forwards the input value to the next layer, and the input node directly passes the input attribute value to the next layer (hidden layer or output layer). However, in neural networks, there is a functional relationship between the input and output of the hidden layer and output layer nodes. This function is called the activation function. The mathematical equation expression of the activation function is

$$A_{i,j,k}^{l} = f\left(Z_{i,j,k}^{l}\right) \tag{2}$$

In above equation, k is the channel number in the feature map. l is the layer number. The most commonly used activation function is the Rectified Linear Unit (ReLU) function. We also used the ReLu function in our research as activation function. This function can be written as following:

$$f(x) = max(0, x) \tag{3}$$

where x is the input value. Before ReLu is commonly used, people have used sigmoid and hyperbolic tangnt as activation function as well. In recent studies, other functions are also used as activation functions such as Leaky ReLU (LReLu), Parametric ReLU (PReLU), Randomized ReLU (RReLU), Exponential Linear Unit (ELU) and etc.[127][128][129][130].

The activation function usually are placed after the convolution calculation, but in some early research, such as LeNet-5 [131], the activation is calculated after the pooling.

The output feature map will be passed to the pooling layer for feature selection and information filtering. The pooling layer contains a pre-configured pooling function whose function is to replace the result of a single pixel in the feature map with the feature map statistics of its adjacent areas.

Those above equations summarised how the CNN works and how different parameter settings are applied in our proposed architecture. We have implemented different feature extraction channels for different input data types. In the next section, the data flow in different featrue extraction channel and the classifier training will be discussed.

4.4.1 Cross-Modal Network Structure

In this research, we first tested the network performance using only one type of data, i.e., a network structure that uses only audio or video data to make predictions about the severity levels of dysarthric speech. Then, we construct a network architecture as shown in Figure reffig:1nw to process audio-video data. The structure of one-type data predictive model is shown below.

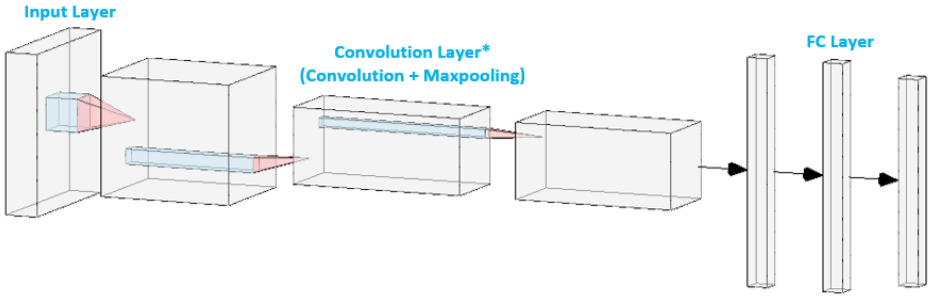


Figure 4.5: The audio/video processing network structure. The actual number of convolution layer sets and FC layers can differ from this figure

As shown in Figure 4.5, similar structures have been used in the audio-only/video-only predictive model. The first larger grey cube on the left of the network is the input layer.

Other grey cubes are the convolutional layer sets. One convolution layer set refers to the combination of convolution layer plus a maxpooling layer. In literature, this combination is commonly referred to as convolution layer. The last part of the network is the fully-connected (FC) layers as labelled in the figure. The smaller blue cubes are the filters used in each layer. The first part of the entire network is the CNN layer group in which data extraction is carried out. Then feature is fed into FC layers so that the network is trained. The number of CNN layers and FC layers used in actual network implementation may vary from the figure indication.

This network structure takes only one type of data as input: audio data or video data. The audio data and video data processing network have similar architectures, although they might have different numbers of CNN layers and FC layers. Figure 4.6 shows the audio-video cross-modal network structure.

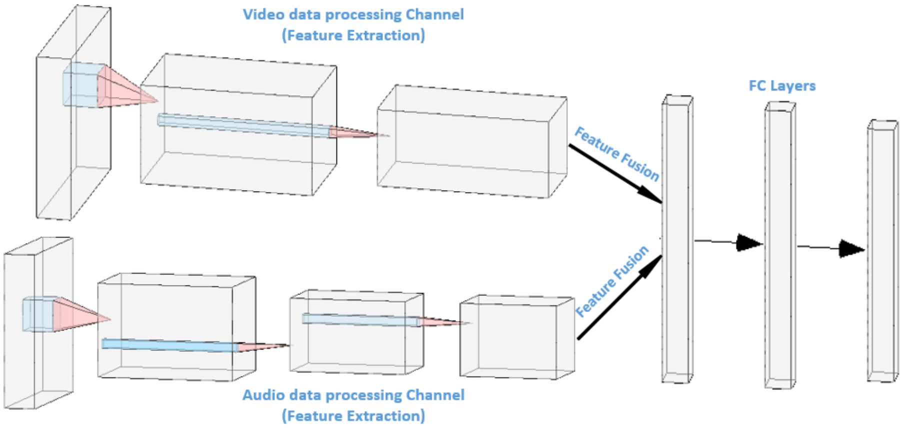


Figure 4.6: The audio-video cross-modal network structure. The actual number of convolution layers and FC layers can differ from this figure

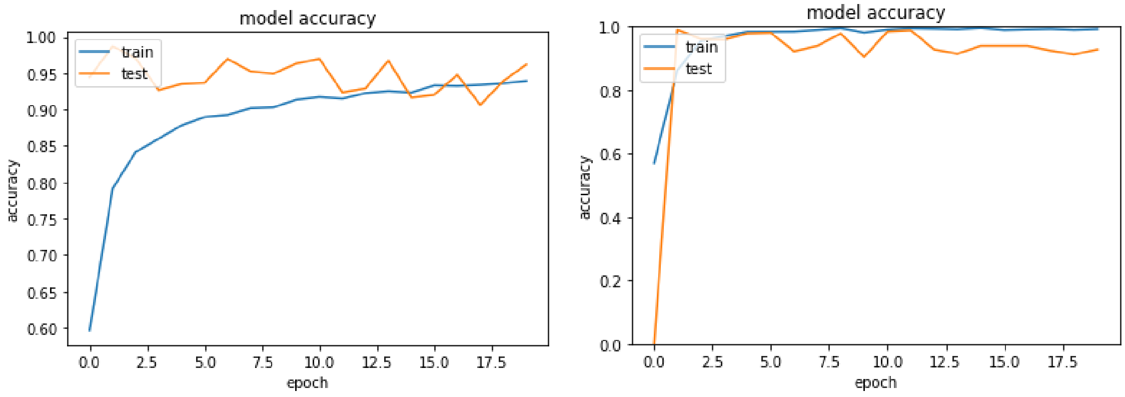
As shown in Figure 4.6, the audio data and video data flow through two separated channels. The acoustic and visual features are extracted in the audio processing channel and video processing channel, respectively. After the feature extraction stage, data fusion is applied to combine the acoustic and visual features to form a joint-feature vectors, as shown in the figure. Then, the combined feature vector is fed into FC layers as input to train the parameters in FC layers. Finally, the network generates a prediction vector which is fed into a neural network classifier to classify dysarthria severity levels. Note that in the actual network implementation, the number of CNN layers and FC layers can be different from the numbers shows in Figure 4.6. Some actual implementation examples with different number of network layers will be shown and discussed in next chapter.

5 Results and Discussion

5.1 Experimental Results

This chapter discusses the results of this research obtained from the proposed method in chapter 4. In this research, we have trained three different predictive models using three different training input setups: audio-only predictive model that uses only audio data as input data, video-only predictive model that uses only video data as input data, and the audio-video cross-modal uses both audio and video data. For each setup, we show at least two results for different network structures to give the idea of how different architectures might change the performance of the proposed deep-learning model.

Figure 5.1 shows the training process and the result of the predictive model for two network structure setups using audio files only, the network is tested on 16,000 samples and validated on 1,600 samples.



(a) (b)

Figure 5.1: (a) Training process and result with fewer network layers - audio-only (b) Training process and result with more network layers - audio-only

As we can see from Figure 5.1, using different parameter setups give us different results. The result shows in Figure 5.1a uses less network layers compared to the network shows in Figure 5.1b. The network in Figure 5.1a has a network setup parameters of 64-32 to 64-32, which means that the network has two convolutional-max pooling layers in the feature extraction architecture and two neural networks in the classification architecture. The number presents the number of filters in the layer, i.e., the two convolutional network layers have 64 and 32 filters, respectively, and the two layers in the classifier network have 64 and 32 filters as well. The second network has network parameters of 128-64-32 to 128-64-32 and training process repeated for 20 epochs. In the first network, the average training accuracy reaches 91.6% after 15 epochs, and the average testing accuracy reaches 93.0% after 12 epochs. In the second network, average training accuracy reaches 99.6% after 10 epochs, and the average testing accuracy reaches 92.6% after 12 epochs. Moreover, during testing the network parameters, reducing the network layers can make the training process faster but sometimes over reduction of the network structure can lead to testing accuracy close to 0. Figure 5.2 summarises the testing accuracies in a box-and-whisker plot for two different structures using audio data only.

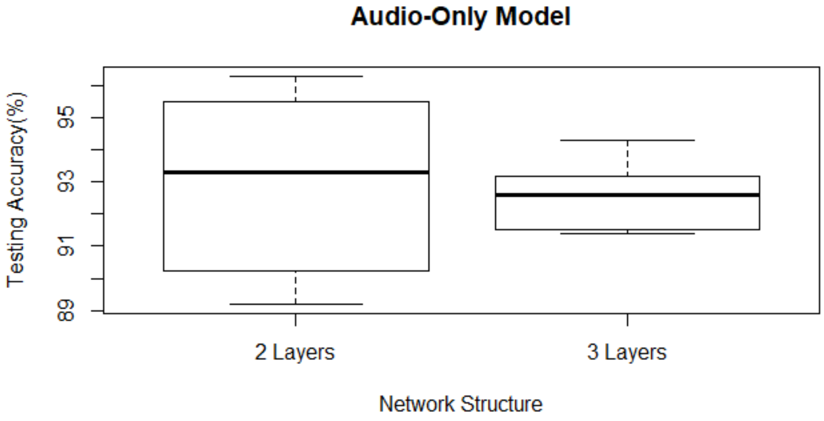
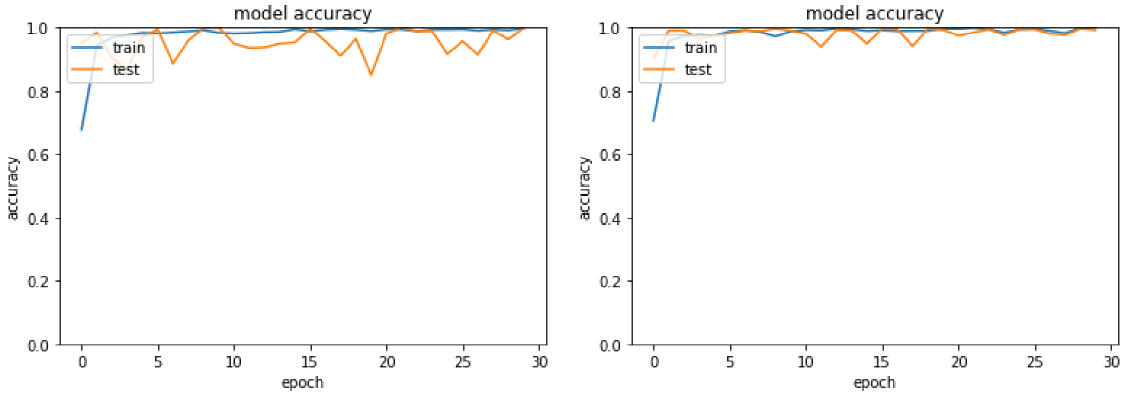


Figure 5.2: Box plots summarising testing accuracies of two network structures for audio-only model

Note that in Figure 5.2, '2 Layers' refers to the network structure that uses 2 CNN layers in the feature extraction process. Similarly, 3 layers means that 3 CNN layers are used for feature extraction in the corresponding network structure. As clearly evidenced in the figure, the median values of testing accuracies for 2-layers and 3-layers network structure are very close. However, the accuracy variations or the value ranges are smaller for the 3-layers network structure. This is because the testing result are more stable for 3-layers network as showing in Figure 5.1.

Figure 5.3 shows the training process and the result of the predictive model using only video feature, the network is trained on 6,000 samples and tested on 1,000 samples.



(a)

(b)

Figure 5.3: (a) Training process and result with less network layers - video-only (b) Training process and result with more network layers - video-only

The first video-only predictive model has a network structure of 128-64-32 to 64-32 and the second model has parameters of 256-128-64-32 to 128-64-32. Both networks are trained for 30 epochs. In the first network, the average training accuracy reaches 99.8% after 13 epochs, and the average testing accuracy reaches 96.1% after 20 epochs. In the second network, average training accuracy reaches 99.5% after 11 epochs, and the average testing accuracy reaches 98.9% after 19 epochs. Figure 5.4 summarises the testing accuracies in a box plot for two different structures using video data only.

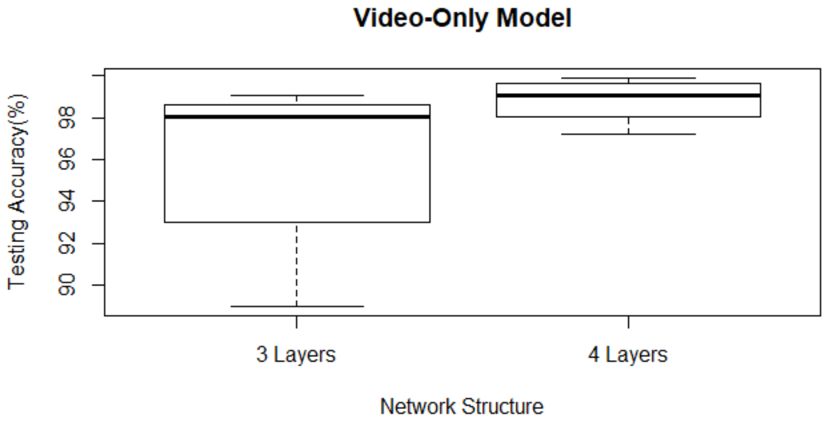
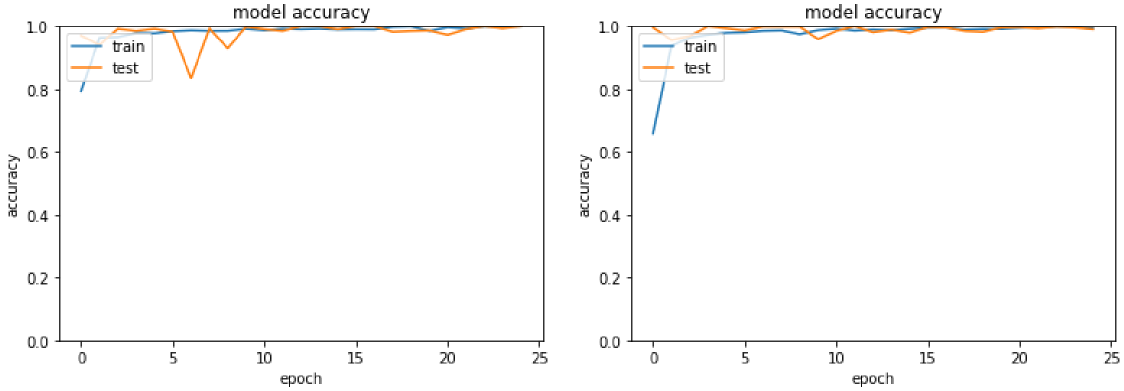


Figure 5.4: Box plots summarising testing accuracies of two network structures for video-only model

Note that 2 different network structures have been used for video-only model: the one on shown on the left of Figure 5.4 has 3 CNN layers in the feature extraction process and the one shown on the right has 4 CNN layers for feature extraction. And from figure 5.4 we can see that the 4-layers structure has a silghtly higher median value than the 3-layers structure. Also, the value range of accuracies for 3-layers are smaller than the 4-layers network, which indicates that the 4-layers network is more stable than the 3-layers.

As we see from predictive systems trained using deep-learning framework, the architecture using audio-only and the architecture used video-only data can both fulfil the objective of automatic dysarthric speech severity level assessment with high accuracy. However, to make the system work faster and with less amount of audio and video data, the cross-modal network is also considered. The cross-modal network has two separated processing channels, one for audio data feature extraction and one for video data feature extraction. After both acoustic and visual features are obtained, they are combined to form joint features. Both networks are trained on 6000 training sample and 1000 validation samples and trained for 25 epochs. Figure 5.5 shows the combined network training process and results.



(a)

(b)

Figure 5.5: (a) Training process and result with less network layers - combined (b) Training process and result with more network layers - combined

Figure 5.6 shows the summary of testing accuracies for audio-video cross modal.

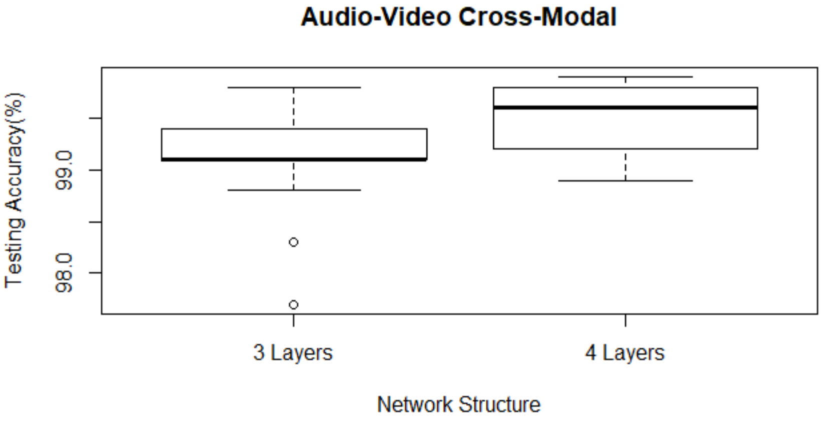


Figure 5.6: Box plots summarising testing accuracies of two network structures for cross modal

Similarly, the numbers in the x-axis indicates the number of CNN layers used in the feature extraction process. As evidenced in Figure 5.6, for the proposed audio-video cross modal, the two network structures have achieved very high accuracy with both accuracy average and accuracy median values above 99%. Table 5.1 summarised the experimental results for different network structures used in this research.

Table 5.1: Summary table of experimental results

Experimental Results Summary					
Input Data	Network Structures (Number of filters and layers)		Epochs	Training Accuracy	Testing Accuracy
	Feature Extraction	Classification			
Audio-Only	64-32	64-32	20	91.6	93.0
Audio-Only	128-64-32	128-64-32	20	99.6	92.6
Video-Only	128-64-32	64-32	30	99.8	96.1
Video-Only	256-128-64-32	128-64-32	30	99.5	98.9

Cross-Modal	128-64-32(Audio)
	128-64-32(Video)
		128-64-32	25	99.8	99.1

Cross-Modal	256-128-64-32(Audio)
	256-128-64-32(Video)
		128-64-32	25	99.8	99.5

As shown in Table 5.1, feature extraction and classification are two critical phases in those architectures, and the numbers the feature extraction and classification columns reflect the network structures. For example, the numbers of 128-64-32 (Audio) and 128-64-32 (Video) in the feature extraction column and the first cross-modal architectures shown in Table 5.1 indicate that both audio and video feature extraction channels have 3 CNN layers with 128, 64, and 32 filters in each layer, respectively. Then for the classification phase, the classifier has 3 fully-connected layers with 128, 64, and 32 filters in each layer.

In the first architecture, the audio processing channel has a network structure of 128-64-32 in the feature extraction step, and the video processing channel has a structure of 128-64-32. After feature extraction, the features will be combined as a joint feature and uses as an input to the classifier. The classifier has a network structure of 128-64-32 as well. Figure 5.5a shows the training outcome of this network. The average training accuracy reaches 99.8% after 9 epochs, and the average testing accuracy reaches 99.1% after 11 epochs. In the second architecture, both audio and video feature extraction channel has the structure of 256-128-64-32, and the classifier has the structure of 128-64-32. The result is shown in Figure 5.5b, the average training accuracy reaches 99.8% after 13 epochs and the average testing accuracy reaches 99.5% after 14 epochs.

Figure 5.7 summarises the testing accuracies for all the 3 different models.

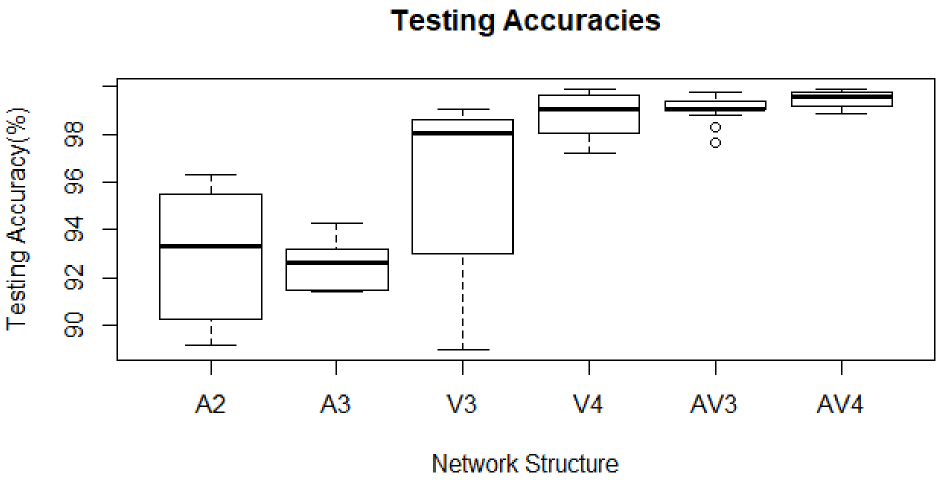


Figure 5.7: Box plots summarising testing accuracies of two network structures for all models

Note that the letters and numbers in the x-axis implies the predictive modal type and the network structure. For example, V3 means video-only modal and 3 CNN layers structure in the feature extraction process and AV4 means that audio-video cross modal and both audio and video feature extraction networks have 4 CNN layers. From Figure 5.7 we can see that for one-modality system, although different network may have similar testing accuracies, the network structures with more number of layers in the training and testing process have more stable result, i.e., less variations in the testing accuracies. Moreover, comparing to the one-modality predictive model, the cross modal is more accurate and stable as evidenced from Figure 5.7, the accuracy value variations are very small. The detailed network specifications can be found in Table 5.1.

Most of the current research related to automatic dysarthria classification are using audio data as sole data source and they are mainly emphasising on the the phoneme-level dysarthric speech recognition task. There are less research about high-level dysarthria severity level classification. However, we find some research closely related to our research. Here we are comparing some results of those research to our results using the proposed deep-learning network architecture.

As discussed in the literature review, research [63] trained an ANN-based network architecture to detect the dysarthric speech severity levels using the speech utterances from UASPEECH dataset. In this research, they first computed 31 audio descriptors to represent the speech utterances. By combining different audio descriptors, they obtained different acoustic feature sets. Those feature sets are used to train the ANNs for dysarthric speech severity level detection. Compare to the network architecture implemented this research, our proposed deep-learning framework for automatic dysarthria severity level classification has improved in the following aspects:

- Higher accuracy

They have used 11 different feature sets and different classification accuracies are obtained using the UASPEECH dataset. The highest accuracy is 96.44% from their research. Using the same dataset, our proposed deep-learning network is able to improve the accuracy by 3.06%.

- Simpler network structure

They have implemented 8 different ANN configurations for different feature sets since different feature sets will generate feature vectors with various lengths. Each of the ANN structures has to be modified specifically to be able to process the input data. This complicates the training process. With our proposed network architecture, all the training data are processed to have the same dimension so there is no need to change the network structures during training. The overall training process is easier to configure.

5.2 Discussion

Three network structures within deep-learning framework are proposed in this research, and the audio and video predictive-model have a similar architecture, as shown in Figure 4.5. The input data was first fed into the CNN networks for feature extraction; then, those extracted features are used to train the FC layers to generate predictive results about the severity levels of dysarthric speech. The number of CNN and FC layers in network structure, as shown in 4.5 can be different, and it has some influences on the result. Similar architecture can be seen in the audio-video cross-modal network, with different steps of processing after the feature extraction stage, as shown in 4.6. After obtaining the acoustic features for audio file and visual features for video file, the two types of features are combined together to form new joint-feature. The new joint features are served as training input for the FC classifier. After training, the network can generate predictions about the dysarthria severity levels from the input data.

Additionally, from the result, we can see that the network complexity (number of CNN or FC layers in the network) do have some influence on network accuracy. Increase the number of CNN or FC layers will slightly increase the accuracy. Oversimplified network structure leads to a result that sometimes the network will not be able to assess the severity level at all. As our experiments show the efficiency of our proposed method for automatic dysarthria assessment, we did not systematically set up some parameters to check the performance of the network, and here we just provided two representative cases for each network structure, one simpler and one slightly more complicated. From the result, we can see that the slightly more complex networks have better training and testing accuracies. However, the network is trained in a lab environment in which there is no time limitation in terms of response time.

Moreover, the result of the combined audio-video predictive model is a more accurate comparing with the system uses only audio or video data, and the cross-modal requires less amount of training data. So cross-modal network structure has the optimised performance among the three network structures, especially in the case when the number of training data is limited. On the side, the cross-modal network also has the most complex network structure amoung all three proposed networks. The influences that network structure complexity brings to different network architecture may be systematically compared in future studies.

6 Conclusion and Future Work

Dysarthria refers to speech disorders caused by illness in the nerve area, paralysis of muscles associated with speech, weakened contractility, or uncoordinated movement. Stresses changes in breathing, resonance, pronunciation, and rhythm, and pathological changes from the brain to the muscles themselves can cause these dysarthric speech symptoms. A simple and efficient way of evaluating dysarthria is critical to place dysarthria patients in therapy sessions more quickly. A conventional auditory testing approach typically involves an SLP presence and an expensive and time-consuming testing procedure. This motivates researchers to develop new computational methods for automatic evaluation of dysarthria severity levels. For the first time, we introduced the the use of deep-learning framework for this purpose. CNN based systems have provided state-of-the-art outcomes in the field of computer vision, in which image/video processing is the fundamental problem. However, to apply those algorithms for automatic assessment of dysarthria severity levels, we have to adapt them accordingly to fit our research interest.

With the proposed method yielded, high training and validation accuracies, we still see some areas for improvement and some potential problems for future study:

- Data pre-processing

The data feature extraction was completed in the CNN framework, but before feeding into the CNN network, the data has to be prepared manually. If the data pre-processing step can be automated, the entire process will be faster.

- Computational power needed

In the implemented network structures, multi-layers CNN and fully-connected layers were heavily used, which generated a huge amount of trainable parameter, and the system must be trained using a dedicated GPU systems. If we can optimise the network structure and reduce the number of parameters in the system, the system will be easier to be embedded in other applications.

- Accessibility

The entire research implementation was completed in a Python environment that does not have a graphical user interface (GUI). If we want to change a parameter, we have to go into the source code and edit the source code. Adding a GUI will make the implementation more user-friendly.

- Structure and parameter fine-turning

As discussed in the previous chapters, we can have a systematic study on the relationship between the number of network layers and network accuracies. However, there are more parameters that can be changed to alter the experiment results, such as the number of filters used in each layer, size of filters, batch size of training data, different activation functions and etc.

- Advanced models

The core structure of the current model is CNN framework. Future studies may use other deep-learning network structures to process data more effectively. An alternative network may be trained to give not only the dysarthria severity level but also the type of the dysarthria, for example.

References

[1] "Uaspeech database," *Statistical Speech Technology Group of University of Illinois*, 2013.

[2] J. B. Mathew, J. Jacob, K. Sajeev, J. Joy, and R. Rajan, "Significance of Feature Selection for Acoustic Modeling in Dysarthric Speech Recognition," in *2018 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, Mar. 2018, pp. 1–4.

[3] G. Diwakar and V. Karjigi, "Repetition detection in dysarthric speech," in *2017 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)*, Mar. 2017, pp. 1150–1154.

[4] a. A. Kostov, "Optimization of dysarthric speech recognition," in *Proceedings of the 19th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. 'Magnificent Milestones and Emerging Opportunities in Medical Engineering' (Cat. No.97CH36136)*, vol. 4, Oct. 1997, pp. 1436–1439 vol.4.

[5] M. V. Mujumdar and R. F. Kubichek, "Design of a dysarthria classifier using global statistics of speech features," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, Mar. 2010, pp. 582–585.

[6] J. Carmichael, "Dysarthria diagnosis via respiration and phonation," in *2015 International Conference and Workshop on Computing and Communication (IEMCON)*, Oct. 2015, pp. 1–5.

[7] O. Ghitza and S. Greenberg, "On the Possible Role of Brain Rhythms in Speech Perception: Intelligibility of Time-Compressed Speech with Periodic and Aperiodic Insertions of Silence," *Phonetica*, vol. 66, no. 1-2, pp. 113–126, 2009. [Online]. Available: https://www.karger.com/DOI/10.1159/000208934

[8] P. Bedenbaugh, D. K. Sarko, H. L. Roth, and E. M. Martin, "Prosody-preserving voice transformation to evaluate brain representations of speech sounds," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 5, pp. 1017–1029, July 2010.

[9] O. Hazrati, S. Ghaffarzadegan, and J. H. L. Hansen, "Leveraging automatic speech recognition in cochlear implants for improved speech intelligibility under reverberation," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015, pp. 5093–5097.

[10] R. A. Sharon, S. Narayanan, M. Sur, and H. A. Murthy, "An empirical study of speech processing in the brain by analyzing the temporal syllable structure in speech-input induced eeg," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 4090–4094.

[11] K. H. Wee, L. Turicchia, and R. Sarpeshkar, "An articulatory silicon vocal tract for speech and hearing prostheses," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 5, no. 4, pp. 339–346, Aug 2011.

[12] H. Kang, D. Kim, and J. Lee, "Are there brain regions related to speech perception? evidence from a functional mri study," in *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Oct 2012, pp. 1100–1102.

[13] F. Aboitiz, "A brain for speech," *A Brain for Speech: A View from Evolutionary Neuroanatomy*, 07 2017.

[14] M. Cernak, A. Asaei, and A. Hyafil, "Cognitive speech coding: Examining the impact of cognitive speech processing on speech compression," *IEEE Signal Processing Magazine*, vol. 35, no. 3, pp. 97–109, May 2018.

[15] E. M. Mugler, M. Goldrick, J. M. Rosenow, M. C. Tate, and M. W. Slutzky, "Decoding of articulatory gestures during word production using speech motor and premotor cor-

tical activity," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Aug 2015, pp. 5339–5342.

[16] A. Lacroix, "Speech production-physics, models and prospective applications," in *ISPA 2001. Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis. In conjunction with 23rd International Conference on Information Technology Interfaces (IEEE Cat.*, June 2001, pp. 3–.

[17] K. J. Kluin, S. Gilman, M. Lohman, and L. Junck, "Characteristics of the Dysarthria of Multiple System Atrophy," *Archives of Neurology*, vol. 53, no. 6, pp. 545–548, Jun. 1996.

[18] R. J. Morris, "Vot and dysarthria: A descriptive study," *Journal of Communication Disorders*, vol. 22, no. 1, pp. 23–33, Feb. 1989.

[19] H. Christensen, I. Casanueva, S. Cunningham, P. Green, and T. Hain, "homeService: Voice-enabled assistive technology in the home using cloud-based automatic speech recognition," in *Proceedings of the Fourth Workshop on Speech and Language Processing for Assistive Technologies.* Grenoble, France: Association for Computational Linguistics, Aug. 2013, pp. 29–34.

[20] M. Fried-Oken, "Voice recognition device as a computer interface for motor and speech impaired people." *Archives of physical medicine and rehabilitation*, vol. 66, no. 10, pp. 678–681, Oct. 1985.

[21] W. B. Matthews, "Paroxysmal symptoms in multiple sclerosis." *Journal of Neurology, Neurosurgery & Psychiatry*, vol. 38, no. 6, pp. 617–623, Jun. 1975.

[22] S. N. Masaki Nishio, "Speaking rate and its components in dysarthric speakers," *Clinical Linguistics & Phonetics*, vol. 15, no. 4, pp. 309–317, Jan. 2001.

[23] Kent Raymond and Netsell Ronald, "A Case Study of an Ataxic Dysarthric: Cineradiographic and Spectrographic Observations," *Journal of Speech and Hearing Disorders*, vol. 40, no. 1, pp. 115–134, Feb. 1975.

[24] S. E. Swedo, H. L. Leonard, B. J. Casey, G. B. Mannheim, M. C. Lenane, D. C. Rettew, and M. B. Schapiro, "Sydenham's Chorea: Physical and Psychological Symptoms of St Vitus Dance," *Pediatrics*, vol. 91, no. 4, pp. 706–713, Apr. 1993.

[25] M. S. Hawley, S. P. Cunningham, P. D. Green, P. Enderby, R. Palmer, S. Sehgal, and P. O'Neill, "A Voice-Input Voice-Output Communication Aid for People With Severe Speech Impairment," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 21, no. 1, pp. 23–31, Jan. 2013.

[26] K. Rosen and S. Yampolsky, "Automatic speech recognition and a review of its functioning with dysarthric speech," *Augmentative and Alternative Communication*, vol. 16, no. 1, pp. 48–60, Jan. 2000.

[27] Kent Ray D., Kent Jane Finley, Duffy Joe R., Thomas Jack E., Weismer Gary, and Stuntebeck Sarah, "Ataxic Dysarthria," *Journal of Speech, Language, and Hearing Research*, vol. 43, no. 5, pp. 1275–1289, Oct. 2000.

[28] J. Noyes and C. Frankish, "Speech recognition technology for individuals with disabilities," *Augmentative and Alternative Communication*, vol. 8, no. 4, pp. 297–303, Jan. 1992.

[29] B. Kashyap, P. N. Pathirana, M. Horne, L. Power, and D. Szmulewicz, "Quantitative Assessment of Syllabic Timing Deficits in Ataxic Dysarthria," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Jul. 2018, pp. 425–428.

[30] S. Perez-Lloret, L. Nègre-Pagès, A. Ojero-Senard, P. Damier, A. Destée, F. Tison, M. Merello, and O. Rascol, "Oro-buccal symptoms (dysphagia, dysarthria, and sial-

orrhea) in patients with Parkinson's disease: preliminary analysis from the French COPARK cohort," *European Journal of Neurology*, vol. 19, no. 1, pp. 28–37, 2012.

[31] T. B. Ijitona, J. J. Soraghan, A. Lowit, G. Di-Caterina, and H. Yue, "Effects of acoustic features modifications on the perception of dysarthric speech — Preliminary study (Pitch, intensity and duration modifications)," in *?IET 3rd International Conference on ??Intelligent Signal Processing (ISP 2017)*, Dec. 2017, pp. 1–6.

[32] J. Millet and N. Zeghidour, "Learning to Detect Dysarthria from Raw Speech," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 5831–5835.

[33] E. Castillo Guerra and D. F. Lovey, "A modern approach to dysarthria classification," in *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE Cat. No.03CH37439)*, vol. 3, Sep. 2003, pp. 2257–2260 Vol.3.

[34] Simpson Marianne B., Till James A., and Goff Anne M., "Long-Term Treatment of Severe Dysarthria," *Journal of Speech and Hearing Disorders*, vol. 53, no. 4, pp. 433–440, Nov. 1988.

[35] G. L. Dorze, L. Ouellet, and J. Ryalls, "Intonation and speech rate in dysarthric speech," *Journal of Communication Disorders*, vol. 27, no. 1, pp. 1–18, May 1994.

[36] H. Ackermann and I. Hertrich, "Voice Onset Time in Ataxic Dysarthria," *Brain and Language*, vol. 56, no. 3, pp. 321–333, Feb. 1997.

[37] Yorkston Kathryn M., "Treatment Efficacy," *Journal of Speech, Language, and Hearing Research*, vol. 39, no. 5, pp. S46–S57, Oct. 1996.

[38] M. Hasegawa-Johnson, J. Gunderson, A. Perlman, and T. Huang, "Hmm-based and svm-based recognition of the speech of talkers with spastic dysarthria," in *2006 IEEE*

International Conference on Acoustics Speech and Signal Processing Proceedings, vol. 3, May 2006, pp. III–III.

[39] T. Orzechowski, A. Izworski, R. Tadeusiewicz, K. Chmurzynska, P. Radkowski, and I. Gatkowska, "Processing of pathological changes in speech caused by dysarthria," in *2005 International Symposium on Intelligent Signal Processing and Communication Systems*, Dec 2005, pp. 49–52.

[40] T. B. Ijitona, J. J. Soraghan, A. Lowit, G. Di-Caterina, and H. Yue, "Automatic detection of speech disorder in dysarthria using extended speech feature extraction and neural networks classification," in *?IET 3rd International Conference on ??Intelligent Signal Processing (ISP 2017)*, Dec. 2017, pp. 1–6.

[41] J. G. Zapata, J. C. D. Martín, and P. G. Vilda, "Fast formant estimation by complex analysis of LPC coefficients," in *2004 12th European Signal Processing Conference*, Sep. 2004, pp. 737–740.

[42] H. Martens, G. V. Nuffelen, M. D. Bodt, T. Dekens, L. Latacz, and W. Verhelst, "Automated assessment and treatment of speech rate and intonation in dysarthria," in *2013 7th International Conference on Pervasive Computing Technologies for Healthcare and Workshops*, May 2013, pp. 382–384.

[43] J. M. Elvira, F. J. Dickin, and R. A. Carrasco, "A comparison of speech feature extraction employing autonomous neural network topologies," in *IEE Colloquium on Systems and Applications of Man-Machine Interaction Using Speech I/O*, Mar. 1991, pp. 9/1–9/5.

[44] N. Kamaruddin, A. W. Abdul Rahman, and N. S. Abdullah, "Speech emotion identification analysis based on different spectral feature extraction methods," in *The 5th International Conference on Information and Communication Technology for The Muslim World (ICT4M)*. Kuching, Malaysia: IEEE, Nov. 2014, pp. 1–5.

[45] K. T. Mengistu and F. Rudzicz, “Adapting acoustic and lexical models to dysarthric speech,” in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2011, pp. 4924–4927.

[46] H. Christensen, I. Casanueva, S. Cunningham, P. Green, and T. Hain, “Automatic selection of speakers for improved acoustic modelling: recognition of disordered speech with sparse data,” in *2014 IEEE Spoken Language Technology Workshop (SLT)*, Dec. 2014, pp. 254–259.

[47] A. Benba, A. Jilbab, A. Hammouch, and S. Sandabad, “Voiceprints analysis using MFCC and SVM for detecting patients with Parkinson’s disease,” in *2015 International Conference on Electrical and Information Technologies (ICEIT)*, Mar. 2015, pp. 300–304.

[48] B. E. Sakar, M. E. Isenkul, C. O. Sakar, A. Sertbas, F. Gurgen, S. Delil, H. Apaydin, and O. Kursun, “Collection and Analysis of a Parkinson Speech Dataset With Multiple Types of Sound Recordings,” *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 4, pp. 828–834, Jul. 2013.

[49] A. Benba, A. Jilbab, and A. Hammouch, “Voice analysis for detecting persons with Parkinson’s disease using MFCC and VQ,” p. 5.

[50] J. Martinez, H. Perez, E. Escamilla, and M. M. Suzuki, “Speaker recognition using Mel frequency Cepstral Coefficients (MFCC) and Vector quantization (VQ) techniques,” in *CONIELECOMP 2012, 22nd International Conference on Electrical Communications and Computers*, Feb. 2012, pp. 248–251.

[51] P. C. S. Kumar, H. O. Department, and D. P. M. Rao, *Design Of An Automatic Speaker Recognition System Using MFCC, Vector Quantization And LBG Algorithm.*

[52] J. Hosom, A. B. Kain, T. Mishra, J. P. H. v. Santen, M. Fried-Oken, and J. Staehely, “Intelligibility of modifications to dysarthric speech,” in *2003 IEEE International Con-*

ference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03)., vol. 1, Apr. 2003, pp. I–I.

[53] G. Vyas, M. K. Dutta, J. Prinosil, and P. Harár, "An automatic diagnosis and assessment of dysarthric speech using speech disorder specific prosodic features," in *2016 39th International Conference on Telecommunications and Signal Processing (TSP)*, Jun. 2016, pp. 515–518.

[54] F. Xiong, J. Barker, and H. Christensen, "Phonetic Analysis of Dysarthric Speech Tempo and Applications to Robust Personalised Dysarthric Speech Recognition," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 5836–5840.

[55] K. M. Ravikumar, B. Reddy, R. Rajagopal, and H. C. Nagaraj, "Automatic Detection of Syllable Repetition in Read Speech for Objective Assessment of Stuttered Disfluencies," *International Journal of Electrical and Computer Engineering*, vol. 2, no. 10, p. 4, 2008.

[56] S. Oue, R. Marxer, and F. Rudzicz, "Automatic dysfluency detection in dysarthric speech using deep belief networks," in *Proceedings of SLPAT 2015: 6th Workshop on Speech and Language Processing for Assistive Technologies.* Dresden, Germany: Association for Computational Linguistics, Sep. 2015, pp. 60–64.

[57] M. Kaushik, M. Trinkle, and A. Hashemi-Sakhtsari, "Automatic Detection and Removal of Disfluencies from Spontaneous Speech," p. 5.

[58] "Intro. to Signal Processing:Curve fitting."

[59] T. Nakashika, T. Yoshioka, T. Takiguchi, Y. Ariki, S. Duffner, and C. Garcia, "Dysarthric speech recognition using a convolutive bottleneck network," in *2014 12th International Conference on Signal Processing (ICSP)*, Oct. 2014, pp. 505–509.

[60] H. Lee, P. Pham, Y. Largman, and A. Y. Ng, "Unsupervised feature learning for audio classification using convolutional deep belief networks," in *Advances in Neural Information Processing Systems 22*, Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, Eds. Curran Associates, Inc., 2009, pp. 1096–1104.

[61] J. N. Carmichael, "Enhancing speech rate estimation techniques to improve dysarthria diagnosis," in *2017 8th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, Oct 2017, pp. 309–313.

[62] S. R. Shahamiri and S. K. Ray, "On the use of array learners towards automatic speech recognition for dysarthria," in *2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA)*, June 2015, pp. 1283–1287.

[63] C. Bhat, B. Vachhani, and S. K. Kopparapu, "Automatic assessment of dysarthria severity level using audio descriptors," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2017, pp. 5070–5074.

[64] B. Peeters, M. Trinkle, and A. Hashemi-Sakhtsari, "Exteacting acoustic descriptors from musical signal," p. 5.

[65] A. Ullah, J. Ahmad, K. Muhammad, M. Sajjad, and S. W. Baik, "Action Recognition in Video Sequences using Deep Bi-Directional LSTM With CNN Features," *IEEE Access*, vol. 6, pp. 1155–1166, 2018.

[66] Y. Fan, X. Lu, D. Li, and Y. Liu, "Video-based Emotion Recognition Using CNN-RNN and C3d Hybrid Networks," in *Proceedings of the 18th ACM International Conference on Multimodal Interaction*, ser. ICMI '16. New York, NY, USA: ACM, 2016, pp. 445–450, event-place: Tokyo, Japan.

[67] I. Fung and B. Mak, "End-To-End Low-Resource Lip-Reading with Maxout Cnn and Lstm," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2018, pp. 2511–2515.

[68] S. Zha, F. Luisier, W. Andrews, N. Srivastava, and R. Salakhutdinov, “Exploiting Image-trained CNN Architectures for Unconstrained Video Classification,” *arXiv:1503.04144 [cs]*, Mar. 2015, arXiv: 1503.04144.

[69] A. Abbas, A. Chadha, Y. Andreopoulos, and M. Jubran, “Rate-Accuracy Trade-Off in Video Classification with Deep Convolutional Neural Networks,” in *2018 25th IEEE International Conference on Image Processing (ICIP)*, Oct. 2018, pp. 793–797.

[70] X. Chen, X. Zhang, H. Tan, L. Lan, Z. Luo, and X. Huang, “Multi-granularity Hierarchical Attention Siamese Network for Visual Tracking,” in *2018 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2018, pp. 1–8.

[71] B. Banerjee and V. Murino, “Efficient pooling of image based CNN features for action recognition in videos,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2017, pp. 2637–2641.

[72] X. Wang, L. Gao, J. Song, and H. Shen, “Beyond Frame-level CNN: Saliency-Aware 3-D CNN With LSTM for Video Action Recognition,” *IEEE Signal Processing Letters*, vol. 24, no. 4, pp. 510–514, Apr. 2017.

[73] Z. Xu, Y. Yang, and A. G. Hauptmann, “A Discriminative CNN Video Representation for Event Detection,” 2015, pp. 1798–1807.

[74] Bo Li, Yuchao Dai, Xuelian Cheng, Huahui Chen, Yi Lin, and Mingyi He, “Skeleton based action recognition using translation-scale invariant image mapping and multi-scale deep CNN,” in *2017 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, Jul. 2017, pp. 601–604.

[75] Y. Lu and H. Li, “Automatic Lip-Reading System Based on Deep Convolutional Neural Network and Attention-Based Long Short-Term Memory,” *Applied Sciences*, vol. 9, no. 8, p. 1599, Jan. 2019.

[76] S. NadeemHashmi, H. Gupta, D. Mittal, K. Kumar, A. Nanda, and S. Gupta, "A Lip Reading Model Using CNN with Batch Normalization," in *2018 Eleventh International Conference on Contemporary Computing (IC3)*, Aug. 2018, pp. 1–6.

[77] M. A. Russo, L. Kurnianggoro, and K. Jo, "Classification of sports videos with combination of deep learning models and transfer learning," in *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, Feb. 2019, pp. 1–5.

[78] J. Chen, S. Liu, and Z. Chen, "Gender classification in live videos," in *2017 IEEE International Conference on Image Processing (ICIP)*, Sep. 2017, pp. 1602–1606.

[79] J. Li, "Parallel two-class 3d-CNN classifiers for video classification," in *2017 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)*, Nov. 2017, pp. 7–11.

[80] X. Chen and Y. Han, "Multi-task CNN Model for Action Detection," in *2018 IEEE Visual Communications and Image Processing (VCIP)*, Dec. 2018, pp. 1–4.

[81] D. P. Lestari, R. Kosasih, T. Handhika, Murni, I. Sari, and A. Fahrurozi, "Fire hotspots detection system on cctv videos using you only look once (yolo) method and tiny yolo model for high buildings evacuation," in *2019 2nd International Conference of Computer and Informatics Engineering (IC2IE)*, Sep. 2019, pp. 87–92.

[82] W. Lan, J. Dang, Y. Wang, and S. Wang, "Pedestrian detection based on yolo network model," in *2018 IEEE International Conference on Mechatronics and Automation (ICMA)*, Aug 2018, pp. 1547–1551.

[83] S. Zhang, S. Lan, Q. Bu, and S. Li, "Yolo based intelligent tracking system for curling sport," in *2019 IEEE/ACIS 18th International Conference on Computer and Information Science (ICIS)*, June 2019, pp. 371–374.

[84] Y. Nie, P. Sommella, M. O'Nils, C. Liguori, and J. Lundgren, "Automatic detection of melanoma with yolo deep convolutional neural networks," in *2019 E-Health and Bioengineering Conference (EHB)*, Nov 2019, pp. 1–4.

[85] "Ucf101 - action recognition data set," *UCF Center for Research in Computer Vision*, 2013.

[86] "Youtube action dataset," *UCF Center for Research in Computer Vision*, 2009.

[87] "Hmdb51: A large video database for human motion recognition," *A Brown University Research Group*, 2011.

[88] O. Ye, Y. Li, G. Li, Z. Li, T. Gao, and T. Ma, "Video scene classification with complex background algorithm based on improved CNNs," in *2018 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*, Sep. 2018, pp. 1–5.

[89] Z. Zheng, Z. Li, and A. Nagar, "Compact Deep Neural Networks for Device-Based Image Classification," in *Mobile Cloud Visual Media Computing: From Interaction to Service*, G. Hua and X.-S. Hua, Eds. Cham: Springer International Publishing, 2015, pp. 201–217.

[90] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[91] Z. Dong, Y. Wu, M. Pei, and Y. Jia, "Vehicle Type Classification Using a Semisupervised Convolutional Neural Network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 2247–2256, Aug. 2015.

[92] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," Jun. 2005.

[93] D. G. Lowe, “Distinctive Image Features from Scale-Invariant Keypoints,” *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[94] S. Choudhary, N. Ojha, and V. Singh, “Real-time crowd behavior detection using SIFT feature extraction technique in video sequences,” in *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, Jun. 2017, pp. 936–940.

[95] X. Ouyang, S. Xu, C. Zhang, P. Zhou, Y. Yang, G. Liu, and X. Li, “A 3d-CNN and LSTM Based Multi-Task Learning Architecture for Action Recognition,” *IEEE Access*, vol. 7, pp. 40 757–40 770, 2019.

[96] A. Liu, Y. Su, W. Nie, and M. Kankanhalli, “Hierarchical Clustering Multi-Task Learning for Joint Human Action Grouping and Recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 1, pp. 102–114, Jan. 2017.

[97] Q. Zhou, G. Wang, K. Jia, and Q. Zhao, “Learning to Share Latent Tasks for Action Recognition,” in *2013 IEEE International Conference on Computer Vision*, Dec. 2013, pp. 2264–2271.

[98] “Recognition of human actions - kth dataset,” *School of Electrical Engineering and Computer Science at KTH Royal Institute of Technology*, 2005.

[99] L. Yue-Hui, “A Novel Local Features Based Athlete Detection Method in Sports Video,” in *2014 Fifth International Conference on Intelligent Systems Design and Engineering Applications*, Jun. 2014, pp. 55–58.

[100] and and a. W. H. and, “Group action recognition in soccer videos,” in *2008 19th International Conference on Pattern Recognition*, Dec. 2008, pp. 1–4.

[101] D. Zeng, Y. Yu, and K. Oyama, “Audio-Visual Embedding for Cross-Modal Music Video Retrieval through Supervised Deep CCA,” in *2018 IEEE International Symposium on Multimedia (ISM)*, Dec. 2018, pp. 143–150.

[102] G. Mao, Y. Yuan, and L. Xiaoqiang, "Deep Cross-Modal Retrieval for Remote Sensing Image and Audio," in *2018 10th IAPR Workshop on Pattern Recognition in Remote Sensing (PRRS)*, Aug. 2018, pp. 1–7.

[103] R. R. R. and, "Cross-modal prediction in audio-visual communication," in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, vol. 4, May 1996, pp. 2056–2059 vol. 4.

[104] A. Brutti and A. Cavallaro, "Online Cross-Modal Adaptation for Audio–Visual Person Identification With Wearable Cameras," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 1, pp. 40–51, Feb. 2017.

[105] "Qm-seminars dataset," *School of Electronic Engineering of Queen Mary University*, 2017.

[106] "Qm-gopro dataset," *School of Electronic Engineering of Queen Mary University*, 2017.

[107] A. Roy and S. Marcel, "Crossmodal Matching of Speakers Using Lip and Voice Features in Temporally Non-overlapping Audio and Video Streams," in *2010 20th International Conference on Pattern Recognition*, Aug. 2010, pp. 4504–4507.

[108] A. Torfi, S. M. Iranmanesh, N. Nasrabadi, and J. Dawson, "3d Convolutional Neural Networks for Cross Audio-Visual Matching Recognition," *IEEE Access*, vol. 5, pp. 22 081–22 091, 2017.

[109] P. D. Polur and G. E. Miller, "Experiments with fast Fourier transform, linear predictive and cepstral coefficients in dysarthric speech recognition algorithms using hidden Markov model," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 13, no. 4, pp. 558–561, Dec. 2005.

[110] M. Dhanalakshmi and P. Vijayalakshmi, "Intelligibility modification of dysarthric speech using HMM-based adaptive synthesis system," in *2015 2nd International Conference on Biomedical Engineering (ICoBE)*, Mar. 2015, pp. 1–5.

[111] "Urbansound8k datasets," *Music and Audio Research Laboratory (MARL), New York University*, 2014.

[112] R. Rasipuram and M. . Mathew, "Integrating articulatory features using kullback-leibler divergence based acoustic model for phoneme recognition," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2011, pp. 5192–5195.

[113] P. Kannadaguli and V. Bhat, "A comparison of gaussian mixture modeling (gmm) and hidden markov modeling (hmm) based approaches for automatic phoneme recognition in kannada," in *2015 International Conference on Signal Processing and Communication (ICSC)*, March 2015, pp. 257–260.

[114] N. Uma Maheswari, A. P. Kabilan, and R. Venkatesh, "Speaker independent speech recognition system based on phoneme identification," in *2008 International Conference on Computing, Communication and Networking*, Dec 2008, pp. 1–6.

[115] D. Vazhenina and K. Markov, "Phoneme set selection for russian speech recognition," in *2011 7th International Conference on Natural Language Processing and Knowledge Engineering*, Nov 2011, pp. 475–478.

[116] J. J. Hwang and K. H. Rhee, "Gaussian forensic detection using blur quantity of forgery image," in *2019 International Conference on Green and Human Information Technology (ICGHIT)*, Jan 2019, pp. 86–88.

[117] Zheng-Jian Ding, Yang Zhang, A-Qing Yang, and Dai-Li, "Image matching of gaussian blurred image based on sift algorithm," in *2012 International Conference on Wavelet*

Active Media Technology and Information Processing (ICWAMTIP), Dec 2012, pp. 121–124.

[118] M. Seeland, M. Rzanny, N. Alaqraa, J. Wäldchen, and P. Mäder, "Plant species classification using flower images—a comparative study of local feature representations," *PLOS ONE*, vol. 12, p. e0170629, 02 2017.

[119] Y. Dong, S. Gao, K. Tao, J. Liu, and H. Wang, "Performance evaluation of early and late fusion methods for generic semantics indexing," *Pattern Analysis and Applications*, vol. 17, no. 1, pp. 37–50, Feb. 2014. [Online]. Available: https://doi.org/10.1007/s10044-013-0336-8

[120] O. Abdel-Hamid, A. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 10, pp. 1533–1545, Oct 2014.

[121] T. R. Brick, M. D. Hunter, and J. F. Cohn, "Get the facs fast: Automated facs face analysis benefits from the addition of velocity," in *2009 3rd International Conference on Affective Computing and Intelligent Interaction and Workshops*, Sep. 2009, pp. 1–7.

[122] R. Amini, C. Lisetti, and G. Ruiz, "Hapfacs 3.0: Facs-based facial expression generator for 3d speaking virtual characters," *IEEE Transactions on Affective Computing*, vol. 6, no. 4, pp. 348–360, Oct 2015.

[123] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[124] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014.

[125] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2016.90

[126] B. Liu, W. Zhao, and Q. Sun, “Study of object detection based on faster r-cnn,” in *2017 Chinese Automation Congress (CAC)*, Oct 2017, pp. 6233–6236.

[127] K. C. Kirana, S. Wibawanto, N. Hidayah, G. P. Cahyono, and K. Asfani, “Improved neural network using integral-relu based prevention activation for face detection,” in *2019 International Conference on Electrical, Electronics and Information Engineering (ICEEIE)*, vol. 6, 2019, pp. 260–263.

[128] H. Hu, “vrelu activation functions for artificial neural networks,” in *2018 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, 2018, pp. 856–860.

[129] Z. Hu, Y. Li, and Z. Yang, “Improving convolutional neural network using pseudo derivative relu,” in *2018 5th International Conference on Systems and Informatics (ICSAI)*, 2018, pp. 283–287.

[130] J. Si, S. L. Harris, and E. Yfantis, “A dynamic relu on neural network,” in *2018 IEEE 13th Dallas Circuits and Systems Conference (DCAS)*, 2018, pp. 1–6.

[131] Y. LeCun, C. Cortes, and C. J. Burges, “The mnist database of handwritten digits, 1998,” *URL http://yann. lecun. com/exdb/mnist*, vol. 10, p. 34, 1998.

Appendix A - Sample Source Codes

Below, we provide some source codes developed in our research, including the code to extract and visualise MFCC features from an audio file, edge detection code for images, landmark detections. Since the audio-only training model has similar network structure as the video-only model, we only attached the source code we developed for video-only networks training. The source code of the cross-modal network training also attached.

Source Code A.1: MFCCs.py

This code shows an example of how MFCC features can be extracted and plotted from an audio file. In research, for-loop or similar interation control statement can be deployed to extract MFCCs of all audio data.

```
import numpy as np
import librosa
import librosa.display
import matplotlib.pyplot as plt

audio_file = ’F02MFCC.wav’
samples, sample_rate = librosa.load(audio_file)
fig = plt.figure(figsize=[5,5])
ax = fig.add_subplot(111)
ax.axes.get_xaxis().set_visible(False)
ax.axes.get_yaxis().set_visible(False)
ax.set_frame_on(False)
S = librosa.feature.melspectrogram(y=samples, sr=sample_rate)
K = librosa.power_to_db(S, ref=np.max)
librosa.display.specshow(K)
```

Source Code A.2: VideoOnlyModel.py

This code shows how a network using video-only data is trained. From the code we can see the number of filters used in each layer, the activation function, size of filters, etc.

```python
import numpy as np
import visualDataPrep as vdp
import matplotlib.pyplot as plt
from keras.models import Model
from keras.layers import Input, Dense, Flatten
from keras.layers import Conv2D, MaxPooling2D

V2data = vdp.prepvData('videoFolder)

VX2 = []
VY2 = []
i=0
for frames, mlabels in V2data:
    VX2.append(frames)
    VY2.append(mlabels)
    i += 1

VX2 = np.array(VX2).reshape(-1,18,54,1)
VY2 = np.array(VY2).reshape(-1,4)

###Feature Extraction
audin = Input(shape=(18, 54, 1))
audp = Conv2D( filters =128 ,kernel_size=(3,3), strides=(1,1),
              padding='same', data_format='channels_last',
              activation = 'relu')(audin)
audp = MaxPooling2D(pool_size=(2,2), strides=(2,2) )(audp)

audp = Conv2D( filters =64 ,kernel_size=(3,3), strides=(1,1),
              padding='same', data_format='channels_last',
              activation = 'relu')(audp)
audp = MaxPooling2D(pool_size=(2,2), strides=(2,2) )(audp)

audp = Conv2D( filters =32 ,kernel_size=(3,3), strides=(1,1),
              padding='same', data_format='channels_last',
              activation = 'relu')(audp)
audp = MaxPooling2D(pool_size=(2,2), strides=(2,2) )(audp)

###Classification
audp = Flatten()(audp)
audp = Dense(128,activation='relu')(audp)
audp = Dense(64,activation='relu')(audp)
audp = Dense(16, activation='relu')(audp)
audp = Dense(4,activation='softmax')(audp)
output = audp

model = Model(inputs=audin, outputs=output)
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=[
   'accuracy'])
model.summary()
```

Source Code A.3: CrossModal.py

This code shows the training process of the cross-modal network. In the feature extraction phase, the audio and video data pass through different channels. Then the extracted features are combined and feed into the fully-connected classifiers.

```
import numpy as np
import visualDataPrep as vdp
import matplotlib.pyplot as plt
from keras.models import Model
from keras.layers import Input, Dense, Flatten, concatenate
from keras.layers import Conv2D, MaxPooling2D


V2data = []
trainingdata = []
audX = []
audY = []
vidX = []
vidY = []

for frames, vlabels in V2data:
    vidX.append(frames)
    vidY.append(vlabels)

for aud, alabels in trainingdata:
    audX.append(aud)
    audY.append(alabels)


def dumbcounter(myarray):
    a = 0
    b = 0
    c = 0
    d = 0
    for i in range(len(myarray)):
        if myarray[i][0] == 1:
            a += 1
        elif myarray[i][1] == 1:
            b += 1
        elif myarray[i][2] == 1:
            c += 1
        elif myarray[i][3] == 1:
            d += 1
    return(a,b,c,d)

vidX = np.array(vidX).reshape(-1,18,54,1)
vidY = np.array(vidY).reshape(-1,4)

audX = np.array(audX).reshape(-1,30,45,1)
audY = np.array(audY).reshape(-1,4)

aa, ab, ac, ad = dumbcounter(audY)
```

```
va, vb, vc, vd = dumbcounter(vidY)

classa = min(aa, va)
classb = min(ab, vb)
classc = min(ac, vc)
classd = min(ad, vd)

audXT = list(audX[0 : classa])
audXT.extend(list(audX[aa : (aa + classb )] ))
audXT.extend(list(audX[ab : (ab + classc )] ))
audXT.extend(list(audX[ac : (ac + classd )] ))
audXT = np.array(audXT).reshape(-1,30,45,1)


###Audio Feature Extraction
audin = Input(shape=(30, 45, 1))
audp = Conv2D( filters =64 ,kernel_size=(3,3), strides=(1,1),
              padding='same', data_format='channels_last', activation = '
   relu')(audin)
audp = MaxPooling2D(pool_size=(2,2), strides=(2,2) )(audp)

audp = Conv2D( filters =32 ,kernel_size=(3,3), strides=(1,1),
              padding='same', data_format='channels_last', activation = '
   relu')(audp)
audp = MaxPooling2D(pool_size=(2,2), strides=(2,2) )(audp)

audp = Conv2D( filters =16 ,kernel_size=(3,3), strides=(1,1),
              padding='same', data_format='channels_last', activation = '
   relu')(audp)
audp = MaxPooling2D(pool_size=(2,2), strides=(2,2) )(audp)

###Video Feature Extraction
vidin = Input(shape=(18, 54, 1))
vidp = Conv2D( filters =64, kernel_size=(3,3), strides=(1,1),
              padding='same', data_format='channels_last', activation = '
   relu')(vidin)
vidp = MaxPooling2D(pool_size=(2,2), strides=(2,2) )(vidp)

vidp = Conv2D( filters =32, kernel_size=(3,3), strides=(1,1),
              padding='same', data_format='channels_last', activation = '
   relu')(vidp)
vidp = MaxPooling2D(pool_size=(2,2), strides=(2,2) )(vidp)

###Flatten and Feature Fusion layers
audp = Flatten()(audp)
vidp = Flatten()(vidp)
combined = concatenate([audp, vidp])

###Classifier
combined = Dense(128,activation='relu')(combined)
combined = Dense(32,activation='relu')(combined)
combined = Dense(4,activation='softmax')(combined)
output = combined
```

```
model = Model(inputs=[audin,vidin], outputs=output)
model.compile(optimizer=’adam’, loss=’categorical_crossentropy’, metrics=[
   ’accuracy’])
model.summary()

history = model.fit([audXT,vidX], vidY, epochs=25, verbose=1,
                    validation_split=0.2, shuffle = True)

print(history.history.keys())
# summarize history for accuracy
plt.plot(history.history[’acc’])
plt.plot(history.history[’val_acc’])
plt.title(’model accuracy’)
plt.ylabel(’accuracy’)
plt.xlabel(’epoch’)
plt.legend([’train’, ’test’], loc=’upper left’)
plt.ylim([0, 1])
plt.show()
# summarize history for loss
plt.plot(history.history[’loss’])
plt.plot(history.history[’val_loss’])
plt.title(’model loss’)
plt.ylabel(’loss’)
plt.xlabel(’epoch’)
plt.legend([’train’, ’test’], loc=’upper left’)
plt.show()
model.save(’VIDnetwork2.h5’)
```

Source Code A.4: Landmarks.py

This code shows how facial landmarks can be extracted from videos. Using the pre-trained face-region recognition and face landmark detection network from dlib, we can extract 68 (0-67) facial landmarks from video recordings of dysarthria patients.

```
import cv2
import dlib

frontfacedetector = dlib.get_frontal_face_detector()
frontfaceshape = dlib.shape_predictor('shape_predictor_68_face_landmarks.
   dat')
loadv = cv2.VideoCapture('landmark.divx')

while(loadv.isOpened()):
    ret, frame = loadv.read()
    if ret == True:
        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        faces = frontfacedetector(gray)
        for face in faces:
            landmarks = frontfaceshape(gray, face)

            for lms in [x for x in range (0, 67) if x not in
   [16,21,26,30,35,41,47,59]]:
                    cv2.line(frame, (landmarks.part(lms).x , landmarks.
   part(lms).y),
                                     (landmarks.part(lms+1).x , landmarks.
   part(lms+1).y),
                                     (0,0,255), 1)
            cv2.line(frame,(landmarks.part(36).x , landmarks.part(36).y),
                                     (landmarks.part(41).x , landmarks.
   part(41).y),
                                     (0,0,255), 1 )
            cv2.line(frame,(landmarks.part(42).x , landmarks.part(42).y),
                                     (landmarks.part(47).x , landmarks.
   part(47).y),
                                     (0,0,255), 1 )
            cv2.line(frame,(landmarks.part(48).x , landmarks.part(48).y),
                                     (landmarks.part(59).x , landmarks.
   part(59).y),
                                     (0,0,255), 1 )
            cv2.line(frame,(landmarks.part(60).x , landmarks.part(60).y),
                                     (landmarks.part(67).x , landmarks.
   part(67).y),
                                     (0,0,255), 1 )
        cv2.imshow('frame', frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
                break
    else:
        break
loadv.release()
cv2.destroyAllWindows()
```

Source Code A.5: EdgeDetection.py

This code was developed in our research to study different approaches to extract information from images. Gaussian blur is used then different thresholds can be set to detect the boundaries of an input image.

```python
import cv2
import numpy as np

def canny(img):
    gray = cv2.cvtColor(myimg, cv2.COLOR_RGB2GRAY)
    blur = cv2.GaussianBlur(gray, (5,5), 0)
    canny = cv2.Canny(blur,30, 90)
    return (blur,canny)

img = cv2.imread('inputImage.jpg')
myimg = np.copy(img)
myblur, mycanny = canny(myimg)
cv2.imshow('result', myblur)
cv2.waitKey(0)

cv2.imwrite('blur.jpg', myblur)
cv2.imwrite('edge.jpg', mycanny)
```