University of Essex

School of Mathematics, Statistics
and Actuarial Science

MA981 DISSERTATION

Speech recognition as a classification problem on audio or video or both

DA COSTA ANRIS ROQUE

Supervisor: **Alexei Vernitski**

September 18, 2024
Colchester

ACKNOWLEDGMENT

I am incredibly grateful to my dissertation supervisor, Vernitski, Alexei, for relentlessly supporting, encouraging, and guiding me with the presented lipreading topic. Even making it possible for me to make the right decisions, your gratitude and assistance will never be forgotten.

ABSTRACT

This report concentrates on visual lip and audio recognition, Visual speech recognition, and speech recognition based through the empirical knowledge from the background study which displays several possible techniques for classifying phonemes through audio classification and visemes through visual lip gestures classification, which would contribute to identifying the spoken word. This classification method for visemes and phonemes can be achieved through several neural network algorithms, especially CNN, LSTM, pre-trained, and machine learning models. Also focusing on prepossessing techniques of images like removal of image noises, image edge detection, gray image conversion, extracting frames from videos, face recognition and lip recognition.

Contents

CHAPTER 1

Introduction

1.1 Background theory

Lipreading is a demanding technology used to interpret words based on the gestures of the lip movement. The estimated value for an individual to interpret lip movement would be around 30 percent because lip movement is decoded, though human vision would primarily depend on factors like the movement of the lips, face, and tongue. However, considering linguistics, spoken English has 44 phonemes, and it is hard to integrate visual units, so-called "visemes," into complete phonemes, as per the grounds of the lip gestures. For instance, some words have nasal sounds, some being noiseless while some being voiced in the same tone. There is also the possibility of homophones where the words have different sounds but the same lip movement for that particular word, which causes ambiguity in interpreting the spoken word, for example, "spit, sip, sit, stick", and so on [70], reflecting on the methods different researchers used variety of techniques to solve lipreading for different applications, some being successful, however having some limitations.

1.2 Dataset Overview

Consequently, the dataset used here is the famous "The Oxford-BBC Lip Reading in the Wild LRW (Lip reading in the wild) dataset", where videos are recorded from the BBC

news telecast for 500 specific words and 1000 utterances from different broadcasters [11]. Additionally, the video will be about 1.16 seconds in length. The data set also consisted of training, validation, and testing sets, which made things easier because it's hard to record each word from any video, and it would take a long time to create such a dataset. Indeed the dataset "The Oxford-BBC Lip Reading in the Wild LRW (Lip reading in the wild) dataset" is only used for non-commercial individual research and private study use only. The BBC content included courtesy of BBC. To achieve this dataset there was a need for an agreement for a year of use and thereby following the protocols specified in the agreement were required to be followed. Indeed the permission to use this dataset was granted on 4TH of July 2024 with the password to download the dataset. The dataset were present in seven parts in the form of tar file, where a Linux command was required to extract the video mp4 format.

1.3 Scope

For the initial task or method 1, the model was trained only for two words, which are 'Workers' and 'Politics' for the first two months, since the dataset was huge and not suffice to be used on a smaller working system. Moreover, for the last two months some methods were used to improve the accuracy for the visual content, which is called by method 2 and the later methods used were for homophones and some hand on practice to gain some ideas which is present after the literature review.

1.4 Application

Lipreading interpretation consists of several applications found explicitly in assistive technology [30], Healthcare [38], Bio metric authentication; for instance, to identify speech and convert spoken audio-visual speech to text for deaf people, the following approach to lipreading in Healthcare is to diagnose speech delays in individuals. While also, Lip reading is used in speech recognition in a noisy environment, to detect long-distance speeches.

1.5 Systems and platforms

To analyse this dataset, A Mac operating system MacBook Pro with apple M2 chip consisting of 8-core CPU with 4 performance cores and 4 efficiency cores, 10-core GPU, 16-core Neural Engine and 100GD/s memory bandwidth, But the limitation on this, is that it cannot handle a huge dataset of 72 GB video while the google colab also was used. [1].

CHAPTER 2

Background study

Informative background study is approached through the referred information from the research papers mentioned below in the literature review that will be based on audio, visual, edge detection and the trends in lipreading.

2.1 Literature Review

- In the context of the research paper 'Lip Reading in the Wild' by Joon Son Chung and Andrew Zisserman from Visual Geometry Group, Department of Engineering Science, University of Oxford, wherein exclusive information about the LWR database and its functionality is provided. The author describes several challenges, for example homophones that ensemble the same lip movement for different words, such as pat, bat, and mat or mark, bark, and park, display some lip movement. While also asserting about audio ambiguity, for example, the alphabet 'm' and 'n' are hard to distinguish on the audio but would have different lip movement that classifies them in a different state. Perhaps the research points out an adverse effect called the 'McGurk effect.' The McGurk effect states that stimuli perception information depends on the facial expression and incoming word frequency [61] [37]. In addition, further factors affect lip reading quality, for example, different speech accents, speed of speaking, and mumbling, while also contributing to video quality, such as the ratio of poor lighting, shadows, motion, resolution, etc. Since the research is based on the foundations of perceiving only

lip reading rather than audio thereby, it concentrates on the empirical use of a convolutions neural network for classifying each word, while also uses temporal sequencing through the use of sequence models like Hidden Markov Models or Recurrent Neural Network. The research paper also provides an in-depth sequel of several datasets and their ranks over preferences. The research paper also displays the key sequel to generating a precise dataset. Relatively, the genre of the dataset is generally inclined to news and current affairs. Several analyses were produced to extract the video using optical character recognition to display the projected words over the video. However, the spoken words weren't in sync with the optical characters on the screen and brought ambiguous results mentioned in the research paper. Specific package HTK toolkits were implemented to determine the maximum likelihood of their verbatim to bring about normality. Several feature detection's were also constructed to obtain the face and lip movement using face landmark detection, which helped create a preferable dataset. In contrast, the dataset was trained over several models, including CNN and VGG-16, while also including data augmentation and the required augmentation scales that enhanced the readability of the model. A report about the words that were hard to classify are mentioned by the author. [12] [11].

- The research paper Lip Reading Sentences in the Wild, by Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman, from the department of enginerering science, university of Oxford, introduces an update on lrw datast and present's its pattern. To overcome certain ambiguities from the previous dataset "lrw" for homophones, the paper suggests adding additional words in a sentence to overcome homophones. The unique feature of this data set is that it is recorded over thousands of hours from the recorded BBC telecast and consists of subtitles for what the speaker speaks, while the previous paper lrw didn't have specific subtitles while they were alight using some toolkit. The interesting point about this LRS dataset is that it has several challenges in extracting information from a broader perspective. For instance, the broadcast contains faces in the Wild, considering it has several faces that may require face detection synchronised with the speech detected by the individual portraying to speak. The research paper includes its work using visual gesture movement of the lip and speech while also

solely speech or lip movement. Additionally, The paper points out the architecture and training strategy, firstly considering the watch architecture that is an image encoder; from the study interprets, features are extracted from convolution layers for each pixel, further locating this set of sample sequences into the LSTM model with a defined timestamp. This would be why the shape of the image sequence must match the audio sequence for a particular timestamp, or else one of them might overlap during the training purposes. Secondly, about listening, ideally, an audio encoder constructed through MFCC (MEL-frequency cepstral coefficient) at a particular timestamp with preferred dimensions. For example, through empirical knowledge, it would be (13, sizes of frames). Thirdly, while implementing spell architecture, ideally known as a character decoder, in the research papers, there is an emphasis on fusing the models through an attention-to-attention mechanism with MPL for the output. The research paper focuses on training approaches for each word rather than sentences during the training process. If a sentence is trained over a timestamp, the outcome mentioned in the research paper shows that it is not converging. However, when using short words, the consequence does converge and has less overfitting. In addition, regarding the limitation, in the multi-model, the audio classification dominates over the visual classification. For training noisy audio data, since the data is taken from the wild, it is necessary to have noise tolerance for faster convergence for the WLAS (Weighted latent Alignment search) model by introducing Additive White Gaussian noise about 10 dB and 0 dB. Compilation took around ten days for this model at a learning rate of 0.1. The paper summarises its results, including visualisation and a summary, like the prior paper of lrw, for extracting clear distinct videos [51].

- The implementation of the dlib library in deep learning face recognition by Dujuan Zhang, Jie Li, Zhenfang, and Shan contributed their work to the recognition of images, published in 2020 international conference on Robots and intelligent system (ICRIS), the use of this research paper was to have focus on the use of dlib library. The author primarily initiates by mentioning the effect of OpenCV due to the ambiguity produced over missing detection and poor recognition effect, Thereby countering these effects by using dlib face recognition using the ERT algorithm for detection. The dlib is a cross-platform library software written

mainly in C++. It is used in linear algebra, networking, threads, graphical user interface, machine learning, image processing, data mining, XML, and text parsing [65]. In brief, the authors suggest the process of Face recognition, which is done by first detecting the face in an image; then, feature points are landmarked and calibrated, and states that the resultant face recognition is carried out based on Euclidean distance analysis, the same method must be applied to lipreading, instead the lip coordinated must be extracted. Furthermore, the author mentions the architecture of the neural network through which the images are convolved and pooled and mentions the SoftMax function, which provides classification if the same image exists, and thereby facilities the number of faces present over a video stream is detected using the neural network regression algorithm. Later, it addresses the use of affine transformation for feature extraction. The author indeed provides a figure that elaborates the procedure. The research paper emphasises in-depth information about OpenCV and Dlib. The face detection classifier for the OpenCV library is known as the Cascade Classifier, which is based on the HAAR algorithm. The cascade classifier acts like a scanner that scans over the detected intensities from grayscale image histogram equalisation. The grey scale histogram is known to measure the intensity of the image. The image is converted to monochrome for grey scales to see the intensity of each pixel present in an image [68]. Thereby paper addresses the problem of missed detections and false detection through dlib and passes openCV for image processing. The paper elaborates on the environment required for the dlib library and mentions its complexity. The dlib requires several implementations over the Python interface. For example, it needs access to CMake and BOOST. In contrast with obtaining the library it was important to know how to implement the syntax, from method used for the provided dlib class (lib.getfrontalfacedetector()) helps get the required set of points for detecting the face. The histogram of the oriented gradient orients the gradients according to the movement towards the fringes. It provides a direction that is then scaled and normalised to a block of pixels, which helps map the image sections [69]. Shape predictor 68 face landmarks must be drawn through cv2.circle to obtain the necessary point for the image section. The research paper stresses on the algorithm in reference of Euclidian distance measurement for face recognition.

It implements that if a given distance is less than the specific threshold value, there is a high chance of similarity, whereas if the distance is more than the threshold, it is not similar [76].

- Through the survey conducted over lip-reading, it was essential to gain insight into using the Librosa library; therefore, a search was required to gain empirical knowledge over this, as a need for exploring audio informatics. The libROSA assessment tool for the Music Information Retrieval System essentially research article would provide an idea regarding the use of this library over lipreading. In the perception of music, it is essential to know Several streams that provide examinations for musical instruments. Every teacher would have a different analysis of musical notes, and they may vary according to human cognisance. The librosa becomes an important tool in recognising audio signal classification. The research paper over here provides information regarding evaluating student's musical performance based on the teacher's musical recorded benchmarks. The essential features required for assessments are loudness, tempo, rolloff frequency, kurtosis, skewness, and centroid based on the piano. For better performance, the paper provides information on noise cancellations, which is an essential objective of the study. Over here, a matrix consists of the beats required for that particular music sample, where a student can evaluate their beats over a music sample, using chord estimation classification based on the musical notes. The interesting point of view is that a chord has several sounds and will consist of noise. In this regard, it is essential to reduce the noise presence as per the empirical cognisance of music, for example, consonance and dissonance [62]. The pattern detection of the music as there are several timestamps for the musical beat and structural segmentation based on that particular beat. For example, a certain silence or a decrease in speech and melody extraction. The research paper provides the required strategy to neglect noise by increasing the intensity of the pitch, which helps to detect the notes played by the student. The algorithm is based on the onset; The onset includes the note played at that particular time, which may be in microseconds or longer. This will help in the study of lip reading, where analysis is based on the order over that particular time stamp, for example, phonemes and visemes at that particular timestamp, and the noise resistance may deteriorate the performance to

extract a particular sequence. So, it is essential to increase the loudness so that the noise can also be seen. After producing the loudness, the next step shown in the research paper is the removal of noise present in that particular time stamp, which is done through the use of statistical techniques using the z score and finding the mean of the noise and the current pitch of that particular sound. The robust package librosa is used to detect the beats through MFCC, which also helps in the analysis of the spectrogram while also displaying the musical entities. The paper also provides information on the Django web development tool, which provides internet interfaces for musical tests [48].

- A similar update over lip reading can be analysed in SpeakingFaces, the advanced dataset containing a combination of visual, thermal, and audio streams. The paper provides its uses in different domains, specifically human-computer interaction, biometric authentication, recognition analysis, domain transfer, and speech recognition. The model provided by the speaking face emphasises its optimal performance using thermal and visual pictures with an audio source. Through this research paper, it is interesting to learn that several companies use thermal cameras to detect facial features. For example, FLIR developed the FLIP ONE Pro thermal camera to connect to Android or iOS smartphones, At the same time, the company Caterpillar also introduced CAT S62 Pro integrated with FLIR lepton 3.5 professional graded sensor for Android phones which can be used for providing the edges factors for images. What makes the speakingFace data set popular is its ability to overcome the limitation of multimodal datasets, such as lrw, lrz and other datasets. The speakFace consists of 142 subjects, providing a gender-balanced analogy. What is interesting is that the image is processed from different angle viewpoints. For example, its varied proximity is placed in 9 angles, producing a 3D image, wherein 900 frames were collected for each position, about 29 frames per second for two cameras, which is about 32 seconds of video. The problem study used in LRW dataset is that the person moves over different angles, making it hard to detect lip movement and not containing thermal data, as a fact, conisdering this limitation for lrw. Therefore, the speaking face provides a proper infrastructure to pattern the lip and overcome losses of lip interpretations. In addition, the dataset consists of about 100 valuable phrases. Certain concepts brought

about curiosity regarding the preprocessing techniques as the author stresses that there was a need to detect facial artefacts that contributed to image objectives, such as First, camera freeze(in thermal), which was detected using the method Structural similarity index of scikit-image library [60]. Second, Blur detection was controlled using the OpenCV library by using the Laplacian method for edges [64]. Third, Flickering was detected through the dlib library by considering the facial boundaries, . Fourth, there is a need for sequence processing of frames. In regards to the methods applied, the author mentions the period of such occurrence. For example, the author mentions that flickers happen only at the beginning of the video capture. Therefore, the corresponding flickering frames were deleted. The two types of images (thermal and visual images) were aligned using the estimation of a planar homography, which involved adding the pixel coordinates corresponding to the features present in both the thermal and visual images. For insignificant edges detection, the author uses two approaches to improve edge detection by using a composite chessboard of two different materials or a board of fix pattern of holes. The author also mentions the use of ArUco markers in the figure, mentioning the combination of a gray scaled image with ArUco to create a distinct image. In the results, the author frames the use of Gender classification through the use of LipNet model, while the model consists of an encoder and a classifier. Thereby the author also mentions some of the limitations on speaking face. [3].

- The lip reading in the wild provides an account of 5 papers based on their accuracy and the data published over this website Landmark-based lipreading in the wild. The first research paper, SynVSR: Data-Efficient Visual Speech Recognition with end-to-end Cross modal Audio Token Synchronisation with an 80.3 per cent accuracy, was published in 2024. In contrast, the paper specifically focuses on full synchronization of audio and visemes and contradicts the previous methods applied had no means of synchronization of this means (visemes and audio) . While the author states about the method used by terming synchronising time-dependent audio and visual data using a non-autoregressive method. The author asserts the importance of lip reading and provides examples like the use of lip reading in speech disorder, benefiting individuals with hearing disorder

and fortifying security systems. In the methodology section, the author provides information about some training objectives, thereby elaborating some loss methods, namely the word classification loss where the cross entropy measures the difference of the predicted class and the ground truth labels of the word, thereby understand how well the synchronisation is established.

For sentence level, the joint CTC Attention loss that integrates two methods namely the connectionist temporal classification CTC loss for encoder and language modelling loss for decoder.

In contrast about Audio reconstruction loss the author prescribes some methods, in the first method the author elaborates the alignment of specific number of audio token with respect to certain number of frames. This synchronisation is done through some specific sample rate of frames and audio mentioned in the paper to be (16kHz) and (25 fps) undergoing through some hop sizes then indicate to align the points between the audio and video waves. Thereby for the following process the author informs that about 4 vector quantized audio tokens were aligned with a unite frame at 100Hz. This method can also be used in DTW, where each hop of the audio is connected to a frame for synchronising.

Furthermore, to generate discrete audio signal the author performs cross entropy estimations and the last performing total loss model through the sum of word classification loss and audio reconstruction loss. For the experimental setup done, the author uses two datasets namely the lrw dataset for English and CAS-VAS-WIK for Chinese to evaluate the recommended task. For the preprocessing steps the author uses media pipe to identify the region of interest with a specific image size of 128X128, and thereby the author states that the extracted points were served as an input feature to point landmark cloud, thereby the later steps were like using augmentation and a random horizontal flip was applied. For the phase of model architecture the word level encoder implanted thought the use of 3D CNN, ResNet18 and some transformers. The author provides the training recipe [5].

- The research paper Another Point of View on Visual Speech Recognition by Baptiste Pouthier, Laurent Pilati, Giacomo Valenti, Charles Bouveyron, and Frederic Precioso displays a vital approach, unlike the other paper, using graphical out-

lines for the face. Also, the authors mention the drawbacks of the latter research papers regarding statistical models. For example, an image is prone to certain biases, such as gender, age, skin tone, and illumination. The author uses graphical landmarks extracted from a facial image and processed to a Graphical Convolution Network (GCN). A complex set of adjacency matrices is studied using the Adaptive Graph Convolution Network to attain that set's relationship between the graphical coordinates. The author also uses several features, which, in turn, determine which region would acquire the likelihood of obtaining the best results. The methodology used in this paper allows the study to gain cognisance of Point Cloud detection, which becomes an essential aspect of this case study. The face produces about 478 landmarks through a media pipe network. The images are aligned taking the nose's tip as point of reference and are normalised. Through the interest of study, a point cloud is a discrete type of a set that provides data points over a region in space [73]; in contrast, those points contain extra attributes like colour (RGB), normals, timestamps and other attributes. The author provides two specific interests; the first is identifying different regions of the face using cloud points and so using sub-sampling points, which are fewer points, maybe cause it takes extra computational time. To make this paper enjoyable, it must provide information about audio inclusion by aligning the point using the timestamps for the audio signals, as the time is inversely proportional to frequency [45].

- The adaptive semantic-spatial-temporal graph convolution network for lip reading uses a semantic-spatial-temporal graph convolution network to analyse the semantic appearance of the face, specifically the landmarks outlined over the face, such as eyes, nose, and lips. In contrast, the spatiotemporal graph investigates the relationship between the movements of the coordinates over time during an individual's speech.

  Adaptive learning provides optimal graph structure when the model is trained for the number of epochs. The author points out some demerits of the concept that it would not harness the accuracy using graph theory because it's different from the ordinary graph as it displays no correlation with the lip-reading landmark points even after using predefined adjacency matrix, it would not provide perfect since the connection of each landmark points are static, the requirement is that it

should be dynamic as the person speaks. As a result, adaptive learning is being performed in this research model. Secondly, during the learning process, the GCN provides a fixed topology, where it will learn the vital relation of lip and later some complex points which are considered infeasible by the author; therefore, due to this limitation, the author finds Adaptive semantic spatial to be reliable due to its capacity and flexibility of learning across different layers.

Also, the author points out that there must be a requirement for two particular graphs, semantic and spatial. The author also provides information that would be used in this study, for example, using appearance-based methods, for example, Pixel value for the region of interest, of the lips since its lip reading and the dimensional reduction methods, for example, PCA, discrete Cosine transform (DCT), Linear discriminate analysis, Maximum likelihood linear transformation. Other methods include optical flow [72], Even Local Gabor Binary patterns from three orthogonal planes were used for automatic facial expression recognition present in the paper mentioned to how lip reading is going about [7], and Nonlinear dimensional [71] reduction [49].

- Lipreading by Locality Discriminant Graph, published in 2007, displays its visual feature extraction using a classification algorithm called Locality Discriminant Graph, unlike the usual methods used for feature extraction, DCT, PCA, and LDA. The LDA provides an advantageous application using manifold learning and Fisher criteria. These maintain non-linear dimensional reduction by perceiving local neighbour affinity within the same classes while discriminating neighbours with other class factors. The data set used in this research paper is AVICAR (Audio-Visual-speech corpus in a Car). The machine learning model used over here is Knn. The lipreading framework design in this paper uses analysis of the region of interest through face tracking done through Adaboost-based face tracker and lip tracker. The estimated region of interest is provided by extracting its features using classical linear transformation algorithms and manifold learning algorithms. This helps track the list of dimensions present around the area of interest While the data is trained using the HMMs algorithm. The research asserts that LDG provides better results than its counterparts [24].

- Dating back to lip reading in 1984, Petajan and Eric David published a paper on

Automatic lipreading to enhance speech recognition. This paper was influenced by research conducted in IBM by Dr Ernie Nassimbene in 1965, where he analysed the reflected beam produced from the tooth displaying speech recognition. For this study, the author produced video samples of around 16 seconds consisting of 16 frames, and the author pointed out that the audio data wasn't aligning with the lip gestures since they were one-sixteenth of a second, which is very quick. So, indeed, to align this problem, the video was presented at a rate of 60 Hz. This was the earliest operation performed in the IBM labs and help gain exciting insights into the early tools used in this lipreading study report. The tools used for acoustic recognition were Voterm for discrete utterance recogniser, for visualisation required good lighting and cameras and for scaling the images, grayscale thresholding is implemented, thereby raster smoothing. The further processes are contour coding for tracking facial demographics to measure pixels, region coding, and matching, which require huge spaces. Early lip reading provides challenging insights and involves a lot of updates with real-time lip-reading capabilities, such as using LSTM, HMMs, and DTW, the required foundation of encoding schemes and the foundation of further development [44].

- Adriana Fernandez-Lopez and Federico M. Sukno published a survey on automatic lip reading in the era of deep learning, which provides insight into the early techniques used from 2007 to 2018. The author argues the problems associated with the latter research papers, for instance, head pose variations, illumination conditions, poor temporal resolution, efficient encoding of spatiotemporal resolution and speaker dependency. The research paper discusses the traditional systems employed using appearance-based features with hidden Markov models. Through the perception, the region of interest can be correlated to appearance-based methods like Template Matching, Histogram-based methods, Eigenfaces and Fisher faces, Bag of Visual words, Colour-based Methods, Texture-based Methods, Deep Learning-based Appearance Models, Support Vector Machines, Appearance-based Object Tracking, Histogram of Oriented Gradients, Active Appearance Models and Visual Bag of Patches and since the data is manipulated over a considerable dataset its essential for dimensional reduction for example PCA, DCT, LDA and locality discriminant graph. The author stresses providing the

appropriate dataset, including the number of speakers, vocabulary size, recording settings, and total duration. The author states that, traditionally, at the start of the nineties, automatic lip reading was constructed through research on a dataset of alphabets and number recognition. In contrast with the lip-reading model in the nineties, the author points out its limitations, like a lower number of subjects and a limited number of recorded data. The most extensive database is the LRS, which consists of about 100,000 utterances spoken by thousands. The author describes several datasets that represent the year it was generated, the number of cites, language, speakers, tasks, classes, utterances, resolution, Duration and the kind of sentence phrases. For earlier datasets for instance, alphabet recognition, AVletter is one of the considerable datasets produced in 1998; each video is around 25 long and 376*288 pixels in resolution. However, AVletter is remarkably low in quality compared to AVletter2 due to its lack of resolution and the number of speakers with several utterances; for example, the pixels size is (1920*1080) for 50 50-second video. The next type of recognition is XM2VTS, one of the most extensive multi-speaker databases, with 295 participants pronouncing two continuous digital strings and one phonetically balanced phrase. VALID and BANCA are like XM2VTS; This provides insights into the earlier data sets used for lip reading. Other than the alphabetical dataset is the multi-view data set, for example, the movement of the head views at an orientation of 90 degrees, for example, lrs and lrw. The best model was using Bi-LSTM using autoencoder. [23].

- The research paper, Audiovisual Synchrony Detection with Optimised Audio Features, focuses on context window length's impact on delta feature computation and compares the MCFFs with energy-based features for lip reading. The methods that enhanced the audiovisual speech recognition were using the selected state-of-the-art handcrafted lip-synchronised visual features, space-time autocorrelation of gradients and deep canonical correlational analysis. This deep canonical correlation is an exciting feature of unsupervised learning of acoustic features. The author of the former research paper promotes the idea of avoiding spoofing through text that is aligned with visual detection, which helps identify the person based on time-related constraints rather than text independently, where a person portrays a picture and voice tone. Through the methodology, the author uses STACOG to

extract the geometrics of the moving object, dlib to extract useful facial features like lips, eyes, and nose, and, for audio, MCFF components are analysed. Both audio and visual data are placed in canonical correlation analysis [50] [38].

- To enhance the knowledge of preprocessing in this case study, informative research conducted by Saban Ozturk and Bayram Akdemir on the effects of histopathological image preprocessing on convolution neural networks determines how the convolution method will vary based on using preprocessing techniques or without using pre-processing techniques. This research was conducted on cancer images and stresses computer-aided diagnosis. In general, the author implies that some research has been done to overcome the noises in the image and address facial recognition by an author named Xu et al. This is a crucial point for the study of lip reading in the process of preprocessing. Here is the list of methods implemented by the author: 1. Find the median value of the original image 2. Remove the median value from the original image 3. Apply waiver filter with 3x3 neighbourhood 3. The final image is substracted from the original image. The wiener filter removes the unwanted noise from the corrupted signal mentioned in Wikipedia [74] The second method is by 1. Find background using image opening 2. the background is removed from the original image 3. Apply median filter with 5x5 neighborhood 4. Apply adaptive histogram equalizer The 3 method applied is: 1. Apply adaptive threshold 2. Apply medianfilter with 5x5 neighborhood to remove small regions The result not much significant change after employing the preprocessing techniques [42].

- The research on a pipeline to data preprocessing for lip reading and audio-visual speech recognition displays the preprocessing used in lip reading. The data sets used here are GRID, MIRACL-VC1 and the pre-trained lipNet model. Additionally, lipNet provides an exciting learning feature over predefined data. The preprocessing methods are extracting the frames from video and grayscale conversion, facial landmark, lip region cropping using Dlib, image sequencing and audio alignment. The limitation concerning LRW is that in the LRW dataset, the subject would tilt the head, making it challenging to capture the data over a tilted face [40]

- The research paper Edge Detection for Discontinuity in Reflectance, Illumination, Normal Depth addresses the four categories of edges based on the discontinuity in surface reflectance, illumination, surface normal and depth. The author mentions using a neural network to detect the four kinds of edges present in the image and suggests the three stages for feature detection. The author mentions the uses of edge detection, for example, pavement crack detection, which uses reflectance discontinuity, and Shadow edge detection, which is necessary to eliminate shadows in the image. Otherwise, using all these four techniques would produce a high-yield image with more information. Additionally, this shows that the author is criticising generic edge detection since it treats the edges individually. Moreover, the dataset used over here is BSDS-RIND, and so mentions the source code for the procedure. This related paper provides examples for addressing the moving or tilted faces and the necessary edges required in the image. [46]

- LipNet End to End Sentence-Level Lipreading details its precision lip reading model to be around 95.2 percent of accuracy then the traditional word level lip reading. Additionally, the author prescribes that the model is operated at a character level using spatio-temporal convolutions neural network, recurrent neural network and connectionist temporal classification loss. The above results are based on the GRID data set which displays a higher accuracy, the dataset consist of 34 subject speakers, 1000 sentences while excluding those corrupted or missing, training set consist of 28, 775 videos. However, the author states that errors in the prediction are occurred within vismes. The architecture for the lipreading is t-frames are chosen and added on to STCNN with Spatial Pooling continued with Bi-GRU and so continued by Linear with a result of CTC losses. Additionally, the author uses augmentation techniques to avoid over fitting, the delays in the video and the varying motion speed by deletion and duplication of the frames present in that dataset. [8].

- A user evaluation of speech/phrase recognition software in critically ill patients: a DECIDE-AI feasibility study, developed for patients with neurological pathologies for examples mention in the research paper are head and neck tumours, Laryngectomies, Vocal cord injury and tracheostomies problems, that have lost their voice. The author mentions that the designs developed in the app are for the

applications of mouthing words, gestures, nodding to say yes or no, writing on paper, visual characters, alphabet boards, and eye gaze boards, among others. At the same time, the author asserts the app's limitations, such as the inability to some identify patterns. Implementing the model for the app was collaborative process by different organisations in the UK. The methods implemented here are Dynamic time wrapping for feature extraction and a DNN neural network for performance [38].

- A research paper on image noise reduction using Linear and non-linear filtering techniques state that noises are present in the photos, specifically during transmission, coding, processing step and image capture. These factors contribute to image sensing from different environmental perspectives. The author gives an example of CCD cameras, where light levels and sensor temperatures are the major factors for noise production in an image. The author mentions different kinds of noise, such as amplifier noise (Gaussian noise), Poisson noise, salt and pepper noise, short noise, Quantization noise (uniform noise), film grain, on-isotropic noise, speckle noise (Multiplicative noise) and Periodic noise. The noise removal is done through the filtering mechanism, such as a smoothed derivative filter, either applied linearly or non-linearly. In a non-linear method, a statistical order is measured by analysing each pixel's median value and replaying it. This non-linear method specifically reduces noises like salt and pepper noise. While in the linear approach, the pixels are replaced by the average value of the number of pixel. The adaptive method is a versatile class of noise filtering techniques. It regards it as adaptive in using both linear and non-linear approaches in replacing the pixel with the average or median, depending on the local statistical property of the pixel values [75].

- LipType, a silent speech recogniser augmented with an independent repair model, The research paper illustrates the use of LipType, an upgraded tool of lipNet for improved speed and accuracy, which was also developed under low-intensity lighting scenarios and improvement on potential error at the output. The author provides a chronological survey of related works that intersect with the research's approach, such as detecting silent speech recognition, low light image enhancement, recognition error correction, and silent input and interaction on mobile

devices. In contrast, through the network architecture, the author displays the structural format of the LipType model, which consists of two sub-models, specifically the spatiotemporal feature extraction and sequence model. The feature extraction mechanism of the mouth uses a pre-processed Dlib to plot the facial landmarks. Towards the following flow, the paper mentions that an ibug face landmark predictor was established with 68 facial landmarks, and thereby, these models are combined with Kalman filtering. Furthermore, a centred mouth crop was used using an affine transformer. A resulting method is applied through 3D CNN to capture the mouth shape, and temporal features are then aligned with 2D Se ResNet, Converting to a single dimension per time step. For model sequencing, bidirectional gated recurrent units are used, thereby producing the resultant based on CTC (connectionist temporal classification). Above all, the dataset used here is the GRID dataset on which the LipNet model was also trained. The only setting drawback would be with relation to LRW: there is not much tilting of the face at different positions. The author highlights that the frame depth for 3D CNN is not uniform; thereby, longer image sequences were truncated while the shorter image sequences were padded with NumPy zeros. The author mentions different performance matrices used to evaluate the model's performance, such as Word error rate, Words per minute, and Computation time and emphasises the performance results. The author addresses different techniques used for other subjects to enhance performance. For instance, a GLAD Net model was implemented for low-lighted images under light enhancement and error reduction, while the research paper displays the workings of GLADNet. The models used for error reduction are Deep denoising Autoencoder, spell checker, and custom language model. The paper's result emphasises a significant decrease in error rate due to the error reduction approaches performed at different stages [43].

2.2 Manipulation check

Some of the hands on practices i had to perform to have a brief idea about the use of 68 face landmarks

- Face Featuring

- This was done in the first week to understand the implementation of facial point landmarking on an single image [76]. The syntax for this is over here

Figure 2.1: Shape predictor

Google Colab code Example for face annotations.

- Time synchronise methods

  - Several methods exist for identifying the time-related constraints over two parameters, thereby facilitating the process of lip reading; it was essential to understand the technical aspects of the time-related framework for distinguishing two parameters.

For learning purposes, I have worked on a single video and audio and tried to find out how to synchronise both.

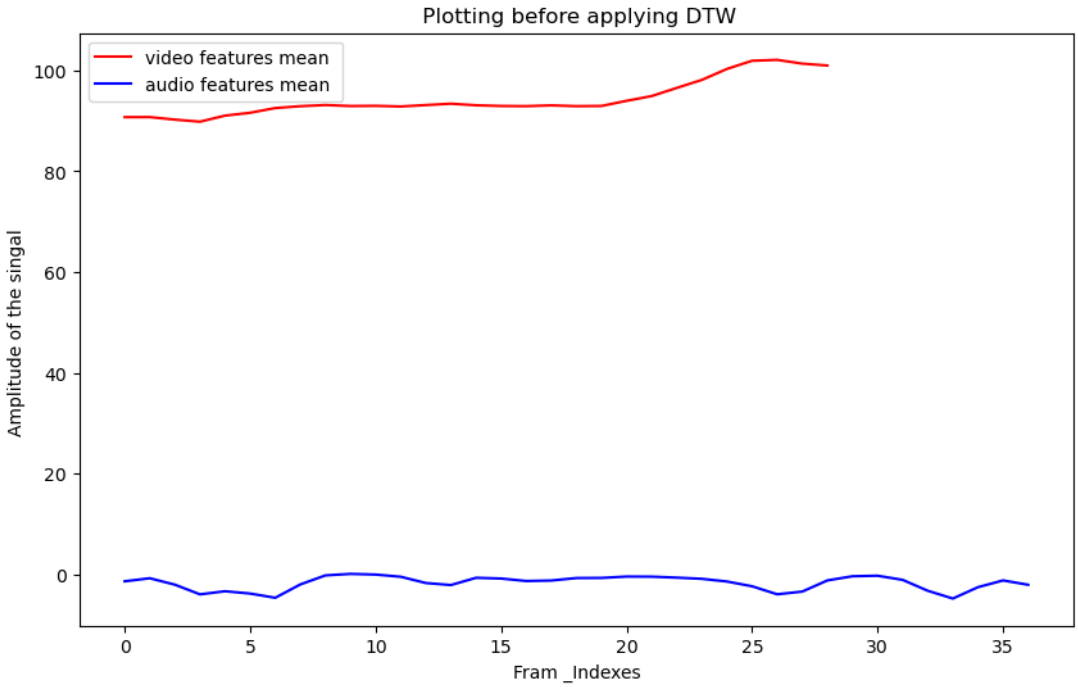

Figure 2.2: Distinguishing the number of frames of audio and video frames

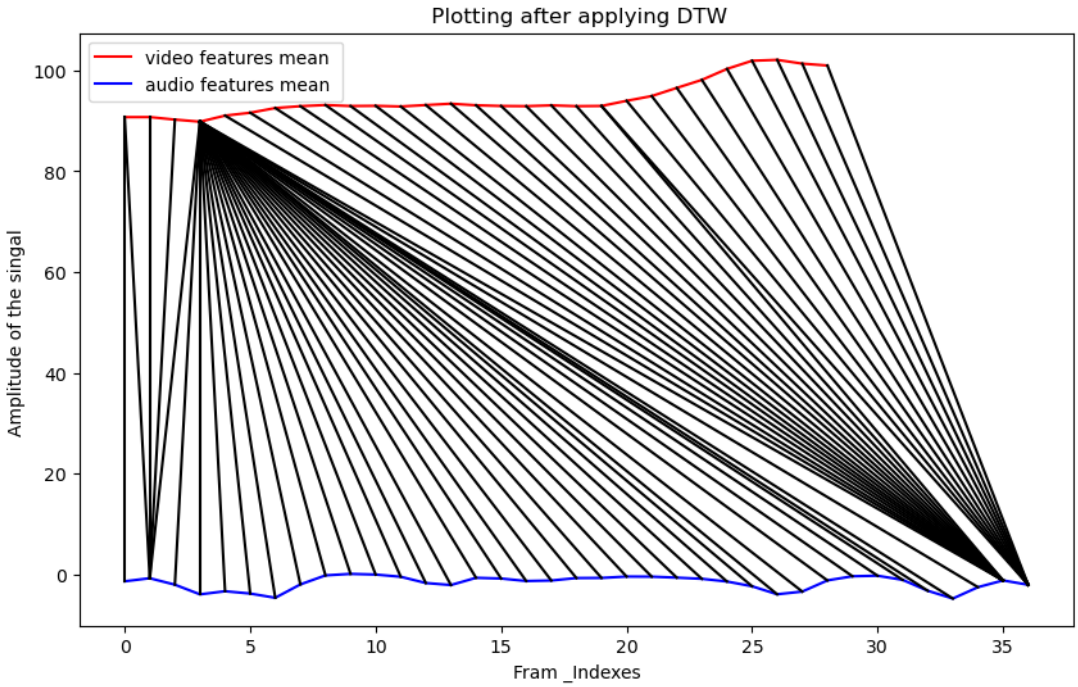

Figure 2.3: Applying DTW over audio frames and visual frames

Thereby illustrating the methods: in this code Google Colab code Example for time series over audio and video singal The fig 2.2 displays the number

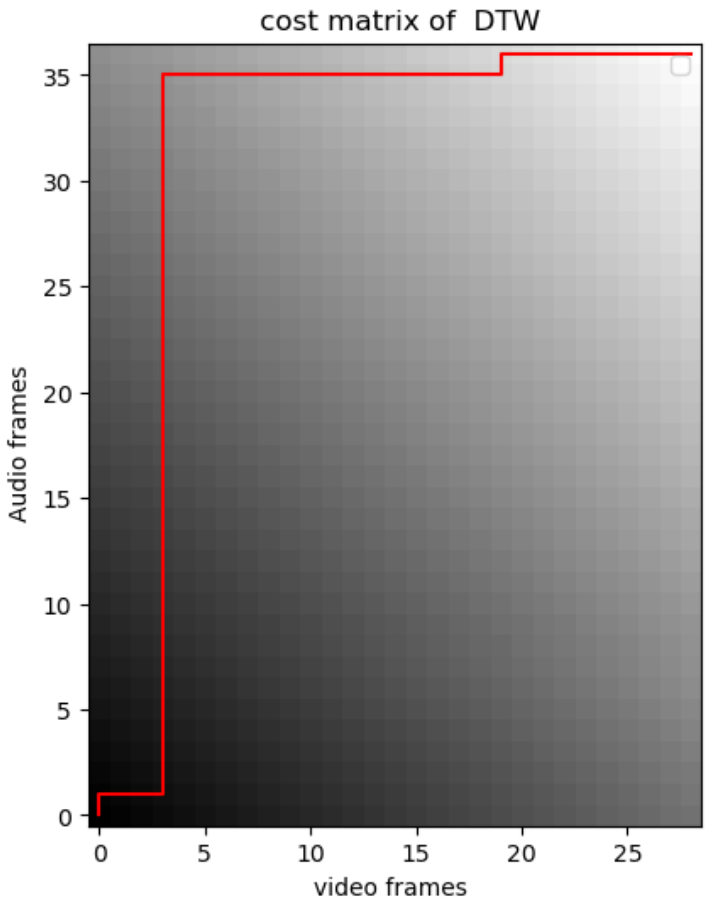

Figure 2.4: Cost Matrix

of frames and amplitude for both audio indicating with a blue outline, while video with a red outline. The amplitude for MCFF is lower then the visual due to its nature of applying logarithm to Mel flter bank [6]. while the visual amplitude is based on the gray scale intensities.

Furthermore, the DTW is mapped for each audio frame for that each video frame as a fact audio is much quicker than the lip movement. Even though the sequence of frames is different in length, DTW is able to locate each point through the use of the cost matrix provided in the next figure [66].

* Through my thinking, from the blog i refereed this formula [53],

Figure 2.5: Lip edge detection

$$\begin{aligned}\text{Cost-function-matrix(Aud, fra)} = \text{distances(Audio, Frames)} \\ + \min\{\text{Cost-function-matrix(Audio-1, frame-1)}, \\ \text{Cost-function-matrix(Audio-1, frame)}, \\ \text{Cost-function-matrix(Audio, frame-1)}\}\end{aligned} \tag{2.1}$$

Aud is audio, and fra is frames; there are several distance matrixes, for example, euclidian and Manhattan distances.

– An informative study was conducted to identify image processing. In contrast, specific methods, such as Gaussian blurriness and the median of that blueness, were implemented. The problem is that implementing median blueness restricts the edges of the image and hence produces fewer edges. It was essential to know how the model extracts such information, for instance, if it works over the pixels and how these edges affect the model's learning. While working over images it was hard to find which of the specific methods must be applied for successive information and which method must be conducted first and last in sequence.

  Thereby, a blog provided some detail information about the methodology of Blurness in an image processing [58], where the author states that it's a techniques used to spread the pixels of an image. Also this technique is mentioned in the research paper of speakingface [3].

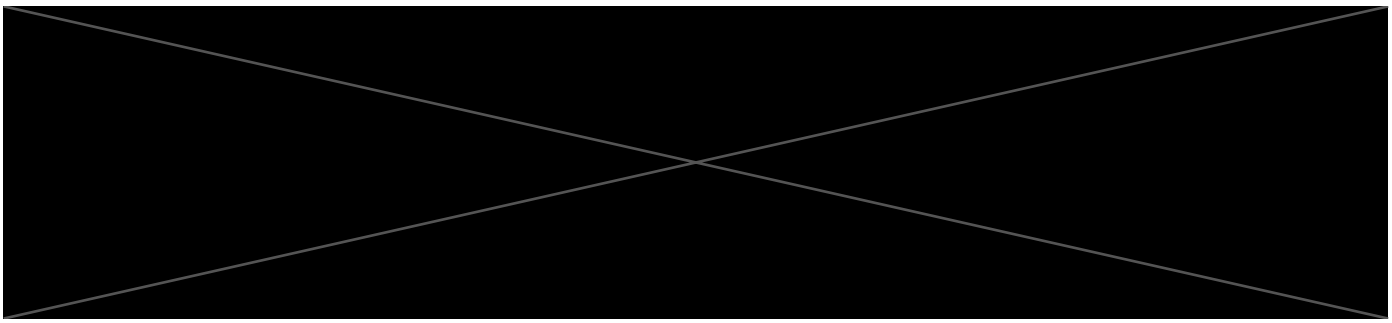

Figure 2.6: The alteration of different methods used onver an image

Thereby median blurriness used in open CV wasn't important as fact that it used to reduce the edges of an image. While also to produce a hard edged image of its lips, the algorithm of edge detection was not able to identity the lips rather detects extra edges when zoomed since in fig 2.5 with the code Edge detection of the lip structure .

The Fig 2.6 displays the conductive study implemented over frames of video, wherein the first figure displays the original image through which a Gaussian blur method is applied to produce a blurred type of image. The application of Gaussian filter are blur, low pass filter, noise suppression, construction of Gaussian pyramids for scaling [10]. The third image is the edge detection there are several methods used for edge detection they are [27]:

* Discrete Gradient Operator considered to be the first derivative. Indeed there are variety of gradient operators.
* Laplacian Operator which is the second derivative consist of zero crossing.
* Canny Edge detection, which is widely used and it uses the best parts of Laplacian operator and gradient operator.
* Corner detection

It is important also to know the types of edge detection, that would be namely Surface Normal Discontinuity, Depth discontinuity, Surface reluctance discontinuity and illumination discontinuity. The code for the above image transitions is present over here. google colab code for preprocessing methods which can be used The figure 2.7 display the Laplacian edge detection from the normal image signal with the pixel index and the intensity of the image

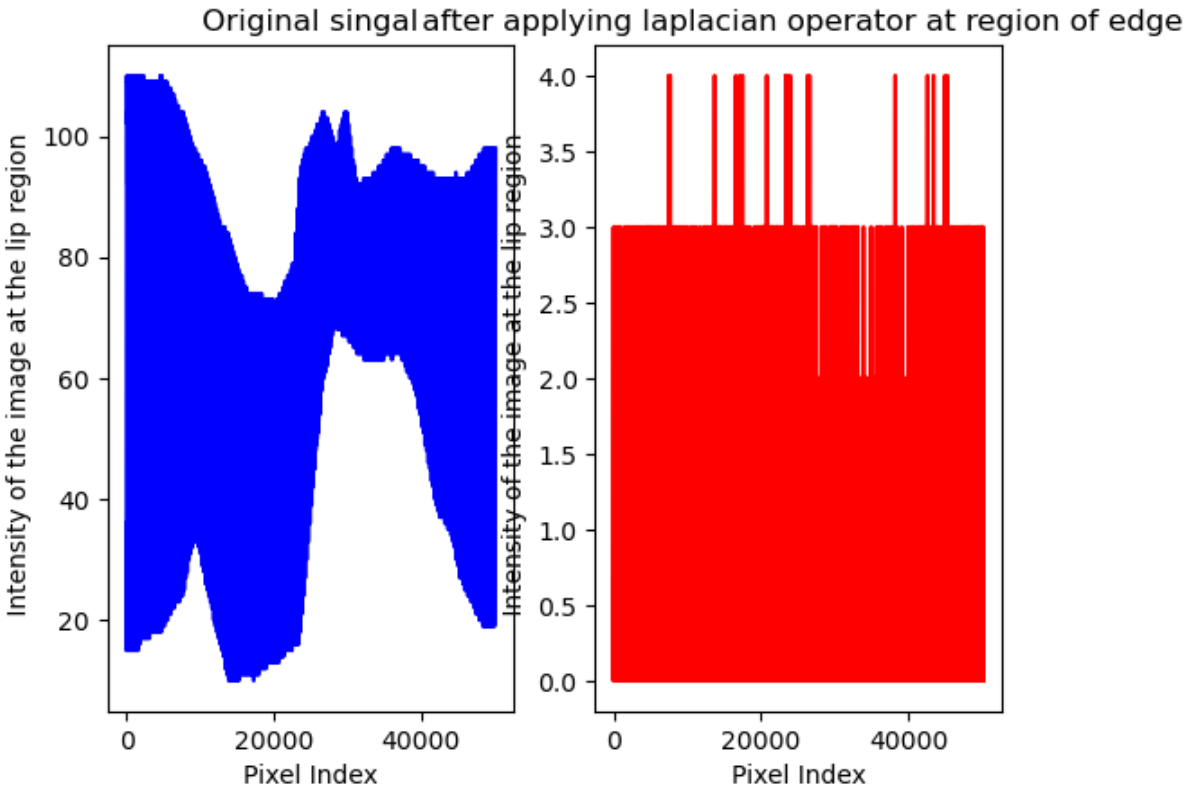

Figure 2.7: Waveform of laplacian edges

at the lip region and this is the code for it Laplkacian waveform. The sharped image is performed by applying the addative weights of original image and Morphed image, the image is not distinct at present but there can be several variation for performing different times of methods and estimating them with additive weights. The last image display the morphological methods. There are several morphological methods , while over here two methods are applied morphed close and gradient. [47]. As fact, the library used is Opencv, wherein there are also other image preprocessing libraries like for example [39] Scikit-image, sciPy, Pillow/Pil, NumPy, Mahotas, SimpleITK and Pgmagick. The reason that in this study Opencv is used because of the use of reading video data.

Another impact scenario i have came across is face recognition, for example a figure 2.8 consist of an image behind the person, actually it was from a television portraying a person's image at some lesser intensity. I have used several algorithm for detecting the second person over there, for example with the use of dlib library denoting that if a person has 3 eyes neglect the

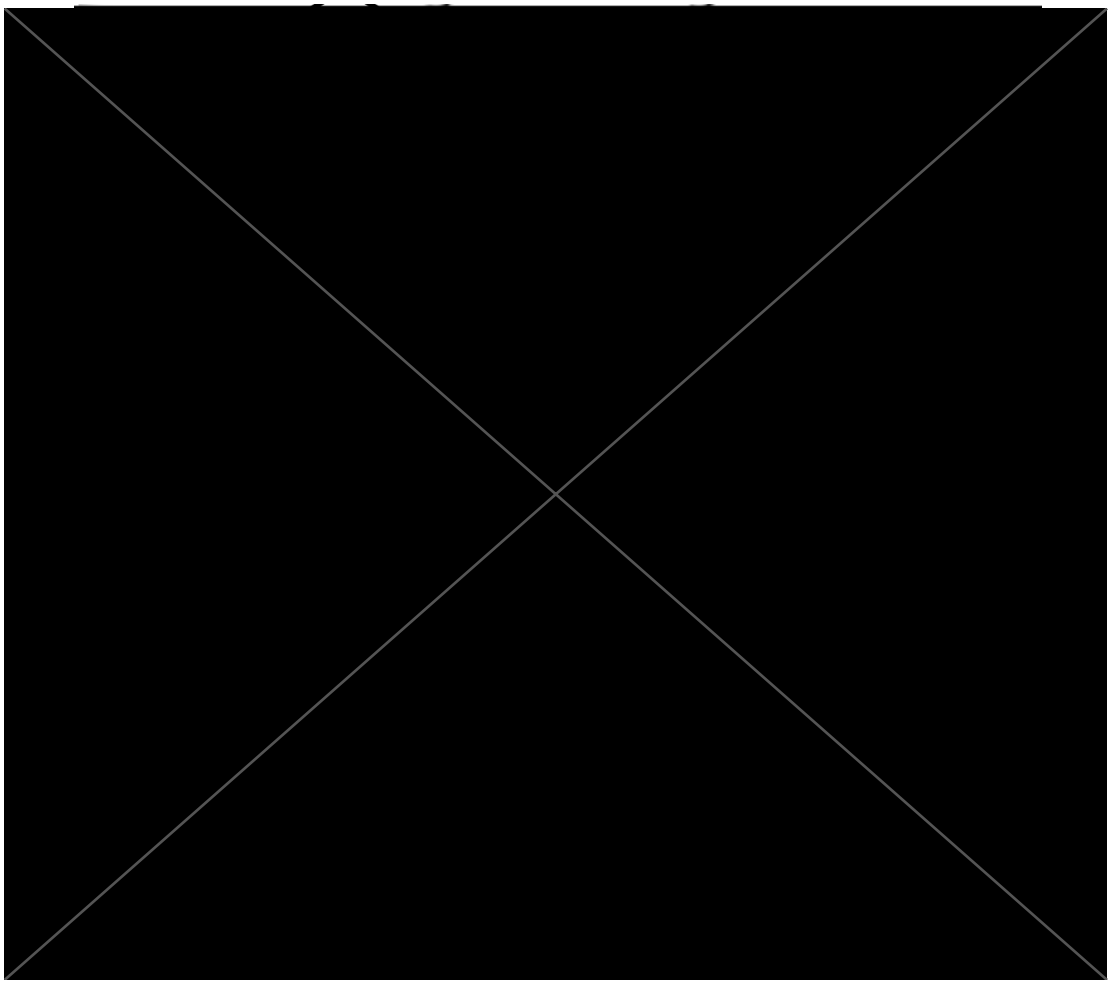

Figure 2.8: An unknow image with lesser intencity from a tv seen

image since for this case study the requirement is only one person in an image. The problem was then the background image had lesser intensity and would not even detect the eyes of the person. Through some cognitive hypothesis multiple people in an image will cause some losses.

For this purpose a pre-trained get frontal face detector with the help of histogram oriented gradient was implemented, while also a heavier method called CNN face detection model v1 for finding accurate faces in the image. In

```
Detected 1 using dlib cnn in frame 0
Detected 1 using dlib cnn in frame 1
Detected 1 using dlib cnn in frame 2
detected this number of 2 in the perticular frame
Detected 2 faces present in POLITICS_00001.mp4
 Skippinmg the video output for video named by POLITICS_00001.mp4 due to multiple
faqce detection
```

Figure 2.9: Neglecting the video with 2 faces

the face detection pipeline Open CV DNN Face Detection is enrolled to detect the faces present in the frame, while the blob operator initiates the images to the neural and net output detects through the use of bounding boxes if there is a face present in an image, with a threshold of 0.5, if the threshold exist the image is present in that axis. However it was hard to identify what method to used, specially the code, so i have used some reference to understand the syntax used for this process [41]. The second pipeline, extracts the person Through the use of Dlib's CNN detector where if an image is identified then it's appended or if more then 2 faces are detected it returns None. The lip extraction pipeline is used only for a person's lip if if detected. In Brief the code does if more then two faces are detected then the video is skipped. The code for face detection is over here Google colab example

CHAPTER 3

Methodologies

3.1 Method-1

Method one was specifically initialised after receiving the dataset on the 4TH of July 2024, and the structure performed over here is basically from the experiments performed from the research paper, for example, "Joon Son Chung and Andrew Zisserman " [11], " Data-Efficient visual speech recognition with end to end crossmodal audio token synchronisation" [5] and two likedin learning video on face recognition and deep learning image recognition(2018), addition to gain influence with dlib library and deep learning algorithms to be performed on images [35] [34].

3.1.1 Pre-processing mythologies

Pre-processing is generally an essential task before preparing a model; it took a while to install dlib into the system, as a fact, during the working. The problem was that the lipreading study could not be conducted on university sources for the requirement of admit credentials for installing a visual studio for dlib. Since the deadline for the dissertation at the initial was on 4TH September 2024, this caused me to hasten the progress without gaining more ideas about computer vision. Therefore, the methods display a brief working with not many basic steps of computer vision over pre-processing.

- One of the most essential and complex tasks was creating pipelines for several purposes; in contrast, these pipelines were specifically used while creating frames

from current mp4 video files using the cv2 library and using librosa to achieve audio mp3 data from the videos. To perform this specific operation, libraries like "os" using Linux commands were implemented to access files via pipeline from an input directory and produce an output of frames, audio and CSV files to a particular folder for further analysis.

Additionally, the design pipelines were created in such a way that it extracts files using commands like 'listdir' that request for the directory and files present, the 'os.path.join' request for joining paths, 'dir' to know if the specific directory is present. In contrast, Through the fundamentals of face recognition algorithms, the library dlib is is used to capture facial structures, such as the eyes, nose, lips and other facial features. So, have extracted the lip structure from the defined coordinates by initialising from the dlib library a pre-trained face shape detector that's '.getfrontalfacedetector' [14] and The coordinates for the lip structure are from 48 to 68 coordinates points. The openCV is used to open and read the video files. Thereby iterating all sequences for different words present in the lrw dataset.

The challenges faced during this preprocessing process were specifically during the iterations phase. For example, the primary approach was to achieve a distinct lip image; in return, it achieved a non-actual lip region like the lower lip with a chin region. As a matter of fact, In the primary stage, the dlib method wasn't used; thereby an alternate method was implemented. Even if the person used to tilt during the video, the cv2 would produce an image of the tilted face where the lip region would not detected. Therefore, Dlib became an essential tool for the lipreading study, which helped to gain a distinct lip region by using predefined coordinates of the lips and thereby iterating over the cv2 methods to extract the region of interest [28]. If the region of interest was not found, the pipeline algorithm would produce feedback saying that the lip region was not generated; this loss was due to the face tilting.

During the reading process on the literature survey, Some papers emphasised two methods of solving lip reading: the statical method using pixels while the other using the geometric coordinates for the face and using time movements while the person is speaking. The reference for this idea is mentioned in this research

paper, "Another point of view on visual speech recognition " [45] The resulting problem over graphical points is displayed in the image below, Google Colab code grphical points on face problem. The image detected in figure 3.1 and 3.2 are the

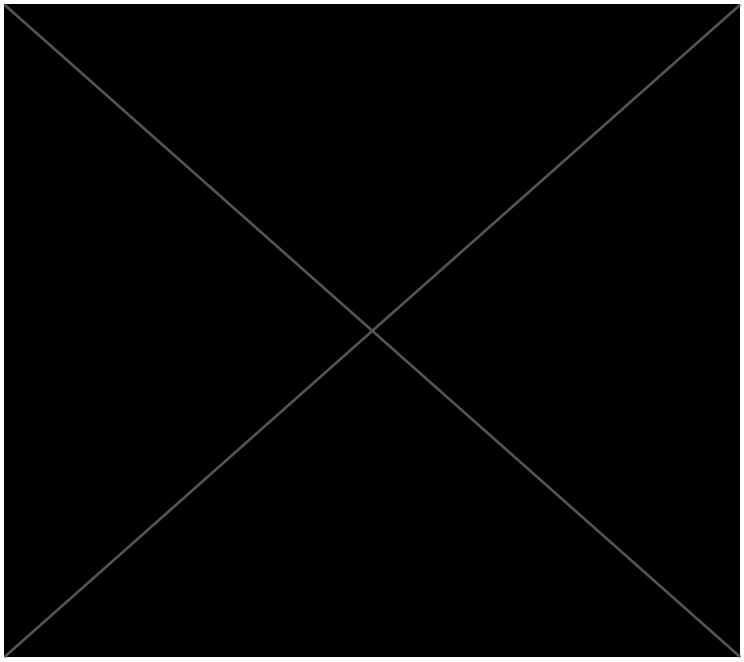

Figure 3.1: Problem associated with creating coordinates points for a moving face

Figure 3.2: Lip coordinates

coordinate markings through the use of the function from Opencv called cv2.circle for landmarking the face and a for loop to produce a line for each points present in the circle, while cv2.line method used for connectivity and so figure 3.1 is the face detection and the 3.2 is the lip detection from the same code where the coordinates

are not mentioned for the output of the second image [16]. The research paper suggest to apply GCN for the coordinates present in the figure for later approach.

While performing analysis on the region of interest for the lip region, the output images display some blueness. As a result it was essential to investigate the problems, based on noises and understand the chronology of de-noising methods.

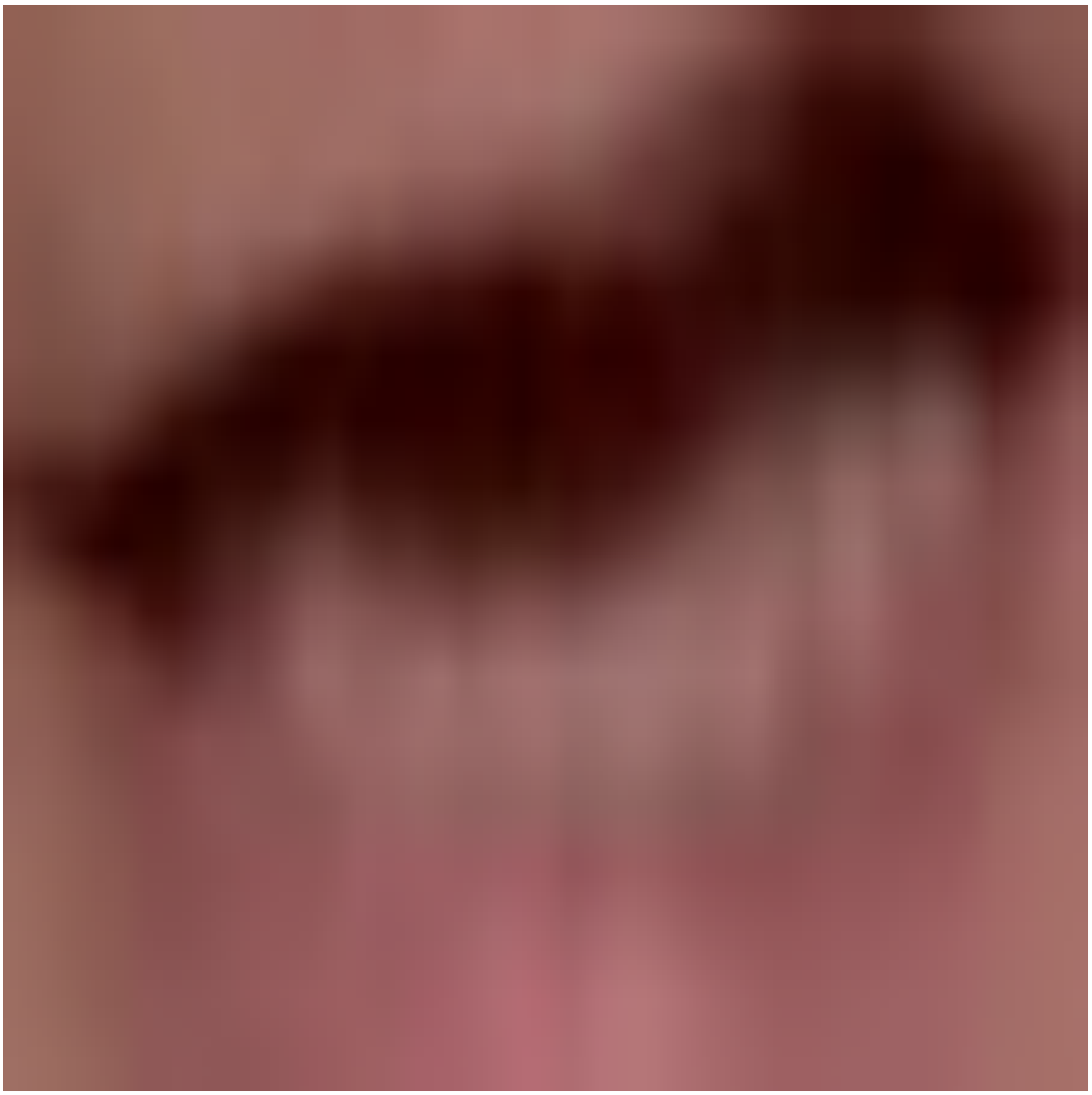

Figure 3.3: Displaying the blurrness during the lip movement

Several methods exist to nullify and detect noises where i have refereed a research paper to understand the properties of noises "Image noises reduction using linear and non linear filtering" [75], but due to early submission it was hard to provide this changes to this method to enhance the accuracy. The image in fig 3.3 displays some blurriness present at the edges while also it is hard to identify the upper lip as a fact that it diminishes.

- It was important to create a CSV file to provide systematic sequencing. This CSV file will contribute all the system directory addresses for video frames and audio data. Additionally, it was essential to create labels within the CSV file. Therefore, a logic was implemented to populate this label to determine if the directory path consisted of a suitable word or not by implementiong an if loop. If the word in the directory consisted of politics or workers, it would enable that particular word to be in the label column or else Unknown. The table below displays the number of frames present for the training, test, and validation sets. The minimum number of frames present in the train, test and validation were due to the tilting of the face and loss of lip extraction through the use of logic where if the mouth region was not detected and if the mouth region is detected but mouth size should not be 0.

Folder	Tot No. of Frames	Min No. of Frames	Max No. of Frames
Train	57835	8	29
Test	2898	28	29
Validation	2897	28	29

Table 3.1: Number of frames

To perform audio extraction, a required library used was MoviePy, where a method called VideoFileClip was used to open the video file, later extract and thereby write to the directory by iterating through all the video files and saving these mp3 files with the name of the video [13]. While converting images to a NumPy array that is during the process of normalization dividing frame by 255 and converting to float32, there were specific needs as larger images would tend to run the memory out while running the code even after creating batches. Therefore, a specific image size was required to analyse this extensive data; an image size of (32 * 32) is considered the best since it was fast in processing. Therefore, resizing images from a higher dimension to a lower dimension was essential.

Two CSV files were produced for specific operations in order of modelling. The first one was used for performing the 2D CNN operation, while the other was used for performing the 3D CNN operation. The only difference between them was that for the 3D CNN, it was essential to have all the frames confined in a list. This can help further in mapping with audio data.

3.1.2 2D-CNN mythologies

- While creating the 2D CNN model, in the first phase, the model tended to overfit; thereby, methods like Data augmentation and callbacks were used to reduce the overfitting of the 2D CNN model. For augmentation, a unique library from tensorflow keras is used to preprocess the frame using the ImageDataGenerater method [20]. The constraints are tuned at the rotation of the image to be 20, the width shift range to be 0.2, and the height shift range to be 0.2, and the horizontal flip should be accurate [63]. From tensorflow keras callbacks, importing early stopping was essential while running over a high epoch rate. During the learning phase, the process will be halted if there is no improvement in the validation loss and accuracy, which prevents overfitting. In this function, parameters monitor the validation loss; the following parameter is patience, determining the number of epochs it must wait before it approaches an improvement [67] [17].

  ReduceLROPlateau was used for fine-tuning, with minimum learning rate and factor [18]. The Parameters used for the 2D CNN model are The input shapes required here (32, 32, 3), where 32 and 32 are height, width, and RGB, which are the essential measures for the input of the 2D CNC model. An activation function called "relu" is used to know which layers should be fed to other layers [4]. So, the total number of parameters for this model is 4,289 090. There are three layers of Convolutions with 32, 64, and 128 filters and size (3*3). There are two dense layers within it. 3 Maxpolling helps detect the image location, which drops to 0.5 [52]. The model was compiled with optimiser Adam [29] and "categorical crossentropy". The model was fitted using 55 epochs and at a batch size of 40. The model was saved at some file location for further analysis, and a classification report was produced there.

3.1.3 3D-CNN mythologies

- In regards of 2DD CNN, the same procedure was used, with a slight improvement over the code. The first step was to upload the CSV dataset containing all the path addresses of the frames in the form of list for the current video. At the first phase

while performing the analysis of the model there were several misconceptions about using this model.

Firstly, It was important to know the model's shape and depth. The 3d CNN model's shape is (depth, height, width, and Colour). The complication was that each video should have an equal number of frames as the depth displays the number of frames [56]. So, the training model's depth varied since some frames weren't captured, leaving behind a gap or a trail of voids. To solve this issue, during some searches encountered some strategical functions called padding and truncating from the research paper LipType [43], which helped in interpreting the use of these functions to fill up the voids with zero for similar lengths. Additionally, the maximum length of the video sequence was 29 frames; therefore, the essential step required here is padding. However, during the phase of creating the model, decided not to implement padding because the focus was on getting the lip gesture, and if these voids are introduced, so does it introduce noise, which may result in over-fitting [22].

Working on the code, encountered challenges in 2D during normalisation, such as out-of-memory issues. Additionally, managing all the frames, primarily all the video frame addresses, was in a list within a string. For example, this format (["frame1", "frame2", "frame3"]) is converted to (["frame1"], ["frame2"], ["frame3"]) through the use of the logic strip, replace and split methods and converting it to an array. Reflecting on the journey, the final configuration of the 3D CNN model was a significant achievement. The number of videos that were filtered was around 1980 for the training set. The code below displays the number of frames and the construction of the 3D CNN. Perhaps the accuracy, in the beginning, was low as 52 per cent enhanced this methods lip data augmentation was implemented to improve the accuracy; this method was random flip left-right, random brightness, random contrast, random hue and random saturation [55].

3.1.4 Multi-Model mythologies

- This multi-model was initiated by extracting both audio and video in the form of frames and audio components; the video components were the same as the method used in the 3D Conv model, consisting of frame size (29, 32, 32, 3). The

only exception was not adding dropouts. The audio component extraction was processed through the library called librosa [15]. Moreover, the extraction process was through a method (librosa.feature.mfcc) called from librosa to extract the Mel-frequency cepstral coefficient (MFCCs) from the audio [36]. This library also tracks chroma features, spectral contrast, and more. While modelling LSTM, There was a need for truncating and padding since the shape of the audio component (37, 13), where 37 was the depth of audio tokens [54], initially did not match the video component with a depth of 29. The sample rate is kept at default at 16000 Hz. So, to achieve (29, 13). If the number of frames in the MFCC component is less

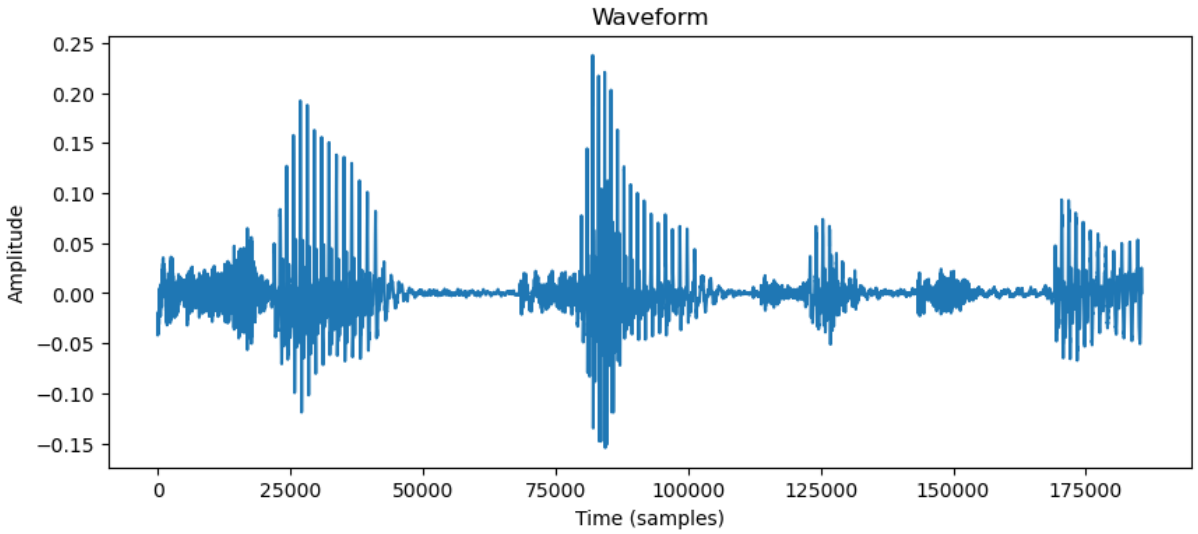

Figure 3.4: Ploting waveform of the word politics

than the maximum frame, it will pad the MFCC's component with several zeros, while if it is more significant, then it will truncate. A basic extraction of audio is displayed over here Google Colab code Example for audio wave spectrum

By getting a suitable audio MCFF described above, the next step is to concatenate both the audio tokens and the frames for each video. For frames, a convolution model is fed with a kernel size of (3, 3, 3), which extracts the space-time features of the frames, producing a feature map, which is three consecutive frames at a time, known as temporal dimensions. At the same time, the spatial dimensions of the kernel are (3*3) pixel patches, such as the height and width of the pixel. These dimensions are reduced by max-pooling, a kernel function of (2, 2, 2) to a flattened 1D vector, and later a dense layer for achieving compact features. In

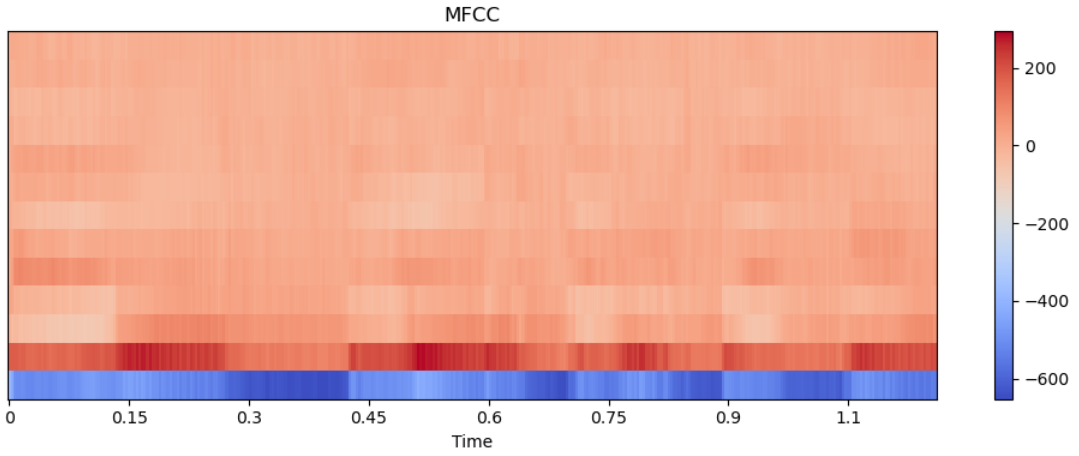

Figure 3.5: A spectogram-like representation of the MCFF ciefficent of the word politics

addition to the LSTM model, the input shape is a sequence of 29-time steps which should associate with the number of frames and 13 audio features per step and thereby process step by step by retaining the memory of past information and later the fusion model concatenates with the frame data and audio data for further processing them in a dense layer to analyse the interaction between the frames and audio and later produces the class probabilities.

3.1.5 Machine learning Model mythologies

- The neural network does consist of some disadvantage due to its transparency and the black box problem which make it hard to understand how neural network came to a decision to solve a particular problem [59]. Hence to over such limitation it would be reasonable to use machine learning algorithm. The machine learning algorithms used analyses the lipreading are SVM and Decision tree.

  The steps include reading the images through the path addresses using method 'imread' from OpenCV class [26], converting the frame path to list form for each image, and the HOG extracts, hog features from the grey scale images or frames that are provided through csv file [69] [2]. A label encoder from the sklearn preprocessing class was implemented for converting the labels to binary format [32]. The next step was training the model using svm [33] and testing through classification reports and confusion matrix [31].

3.1.6 LSTM Model for audio classification mythologies

- LSTM model for audio classification is a sequential deep learning model [21], over here is implemented with the exact shape of (29, 13) for classification [19] with dropouts at 0.5 [52] and a dense layer with softmax activation function that determines the probability of classes and used for multi-class to overcome the problem over sigmoid function [9]. The total parameters are 204,546, epoch set at 20 and a batch size of 32 without early calling. However, there was a need to investigate the original shape of the audio data (37, 13), which narrowed while working on truncating and padding (29, 13) for the multi-model. Therefore, an analysis was implemented using the same formula but with the original shape of the audio data (37, 13).

3.2 Results for method 1

This section will recall the above section, describing the results and comparison of the models, specifically the 2d and 3D CNN models and the visual speech model.

3.2.1 2D CNN Model Analysis report

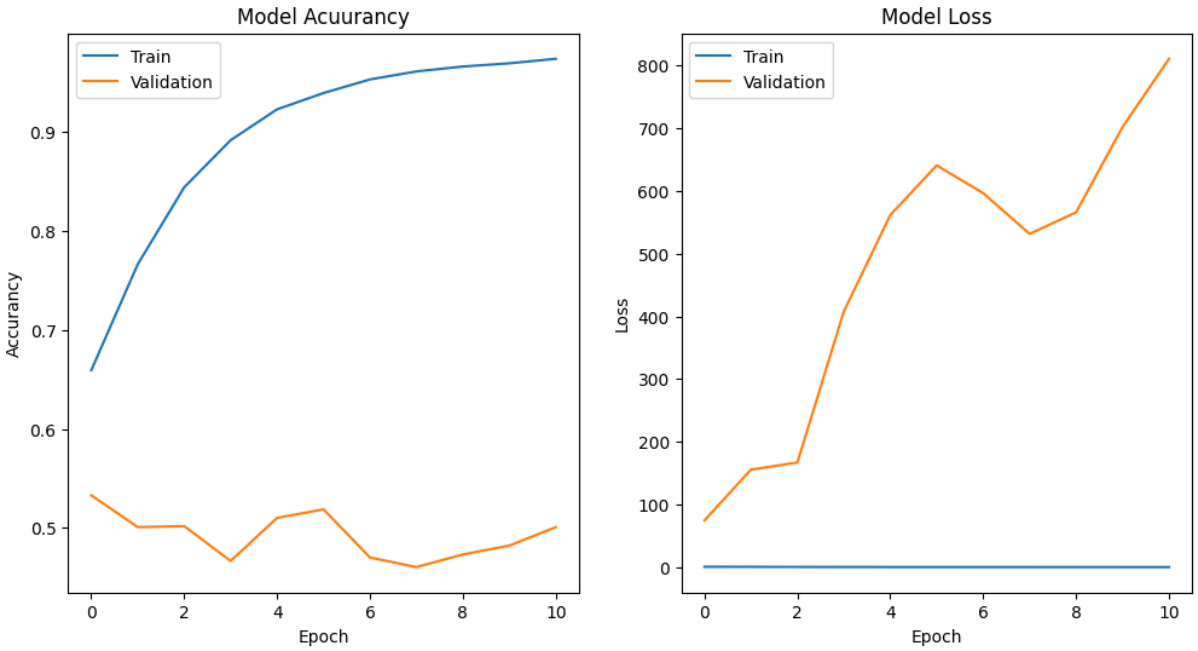

Figure 3.6: Model accuracy and Model loss for 2D cnn Model

- The above graph represents the training progress of a 2D CNN model. The X-axis represents the epoch rate; the model learned for 11 out of 55 epochs. At the same time, the y-axis determines the model accuracy and loss for the training and validation set.

```
Classification Report:
              precision    recall  f1-score   support

           0       0.64      0.69      0.66      1448
           1       0.66      0.61      0.64      1450

    accuracy                           0.65      2898
   macro avg       0.65      0.65      0.65      2898
weighted avg       0.65      0.65      0.65      2898


Confusion Matrix:
[[995 560]
 [453 890]]
```

Figure 3.7: Classification report for 2D CNN model

The training loss increases, wherein the model learns well from the training set while the validation set fluctuates. The training loss keeps decreasing and shows good performance over the training data, while the validation skyrockets and indicates an inferior generalization, resulting in overfitting.

However, there was no proper result, due to inadequate time and in hast forget using methods like shuffling the data, and find the best parameter for the batch size and the other hyper-parameters.

The obtained results compared for the 2D CNN with other authors is tricky, there is evidence from a survey displaying the results and techniques used to obtain better results by Adriana Fernandez Lopez and Federico M.Sukno [23]. Through which display 97 percent accuracy using CNN model, to obtain such results higher resolution were used, indeed this cannot be confined over this device as a fact this procedure was applied over a smaller data set and there are many more methods that can improve the interpret-ability.

The interpretation of the Classification report for the 2d CNN is an accuracy of 65 per cent with a confusion matrix classifying the class of the word used. 995 out

of 1448 frames prove politics, and 890 out of 1450 frames display that workers. While 453 and 560 are wrongly classified as politics and workers [31].

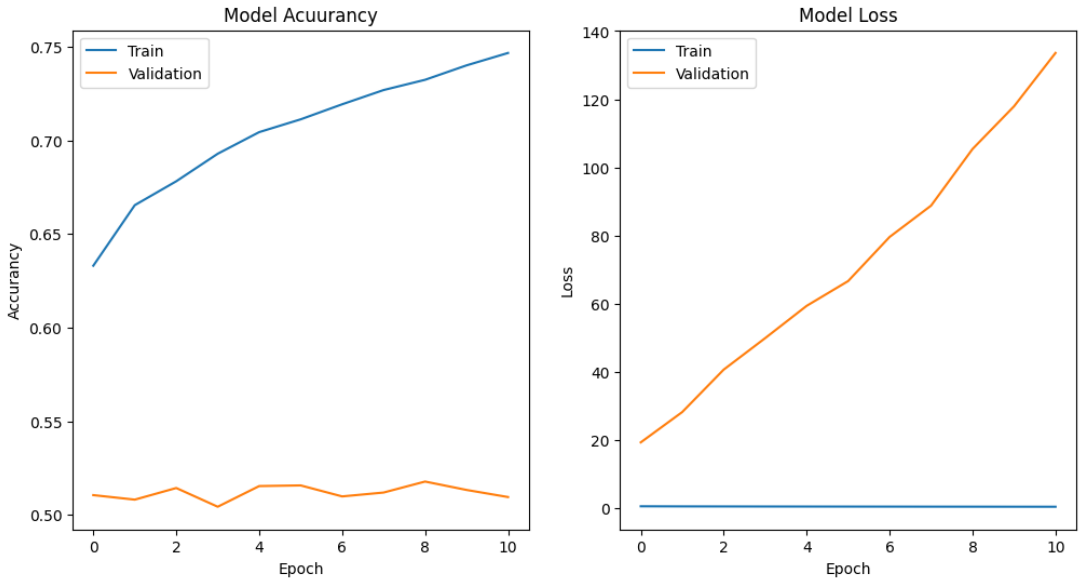

Figure 3.8: Model's Loss and Accuracy for VGG16

While also tried using different models like VGG16 and ResNet. But the accuracy keeps decreasing and shows deep over fitting over the learning models. VGG16 displays the same phenomena as of the 2D CNN. The interpretation of the classifi-

```
Classification Report:
              precision    recall  f1-score   support

           0       0.60      0.74      0.66      1448
           1       0.66      0.50      0.57      1450

    accuracy                           0.62      2898
   macro avg       0.63      0.62      0.61      2898
weighted avg       0.63      0.62      0.61      2898


Confusion Matrix:
[[1078  729]
 [ 370  721]]
```

Figure 3.9: Classification report for VGG16

cation report of VGG16 displays an accuracy of 62 per cent with a confusion matrix determining that 1078 out of 1448 frames are correctly predicted as politics and 721 out of 1450 are predicted as workers. While 370 and 729 frames are wrongly predicted for politics and workers, which displays some difference with the 2D CNN model [31].

3.2.2 Machine learning Analysis report

- Machine learning models test accuracy at 64 per cent while training accuracy at 82.60 per cent, with a confusion matrix denoting 915 out of 1448 frames classified as politics and 929 out of 1450 frames predicted as workers. While 915 and 929 are incorrectly predicted. Interestingly, the confusion matrix closely aligns with 2D CNN's confusion matrix.

```
                precision    recall  f1-score   support

             0       0.64      0.63      0.63      1448
             1       0.64      0.64      0.64      1450

      accuracy                           0.64      2898
     macro avg       0.64      0.64      0.64      2898
  weighted avg       0.64      0.64      0.64      2898

[101]: array([[915, 533],
              [521, 929]])
```

Figure 3.10: Classification report for SVM model

For decision tree the model suffers from overfitting with train accuracy at 100 percent and test accuracy at 52 percent, with a confusion matrix displaying that 797 out of 1462 are correctly predicted for politics and 785 are correctly predicted for workers. While 651 and 785 aren't correctly predicted for both the classes. Comparing result from the survey displays that most of the machine learn models

```
                precision    recall  f1-score   support

             0       0.55      0.55      0.55      1462
             1       0.54      0.55      0.54      1436

      accuracy                           0.55      2898
     macro avg       0.55      0.55      0.55      2898
  weighted avg       0.55      0.55      0.55      2898

20]: array([[797, 665],
            [651, 785]])
```

Figure 3.11: Classification report for decision tree model

performed on lip reading are SVM for feature extraction and later used HMMs model for synchronising frames and audio. This method gave a better accuracy in the survey [23].

3.2.3 3D CNN Analysis report

- The graphical image provided at figure 3.11 displays the accuracy and loss for training and validation set during training the model. The image shows some over fitting from the other models that had a larger over fitting. For seeking the best parameter a hyper parameter tuning was perform for batch size 64 or 32 and learning rate at 0.001 or 0.0001 resultant a best score of 0.54639 with parameters 32 batch size and 0.0001 learning rate. Previously had imputed series of batch sizes, different epoch rates and learning rate, but the process consumed alot of time hence had to turn over with few parameters.

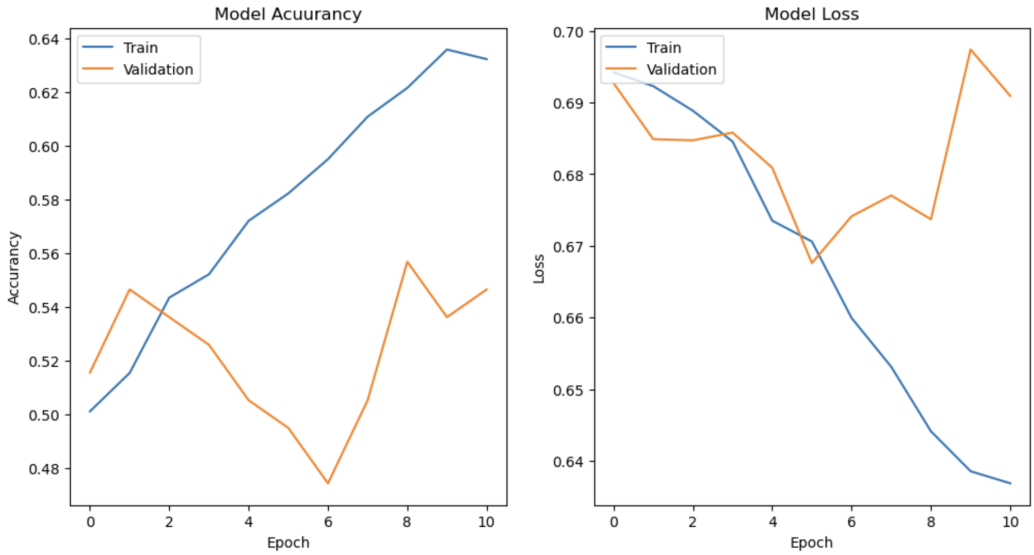

Figure 3.12: Model's Loss and accuracy for 3D CNN

With respect to the classification report for 3D cnn displays the accuracy of the model to be 59 percent with the confusion matrix build through the test set of 98 videos determines that 30 out of 48 videos were classified to be politics and 28 out of 50 are considered as workers. The misclassified examples are at 22 and 18, perhaps this results do align close to 2D CNN by not with the accuracy as the accuracy of the 2D CNN is better then the 3D CNN.

3.2.4 Multi model Analysis report

- The graph of the multi-model using both LSTM and 3D CNN displays good accuracy and slight convergence with both training and validation data in the

```
              precision    recall  f1-score   support

    POLITICS       0.58      0.62      0.60        48
     WORKERS       0.61      0.56      0.58        50

    accuracy                           0.59        98
   macro avg       0.59      0.59      0.59        98
weighted avg       0.59      0.59      0.59        98


Confusion Matrix:
[[30 18]
 [22 28]]
```

Figure 3.13: Classification report of 3D CNN

graph, while the losses also slightly converge. This results in the model performing well using audio information since the lip frame information is inadequate without audio information. As a fact, this statement also aligns with the McGurk effect [37]; therefore, both have to be compressed to find an essential relationship for better performance of the model.

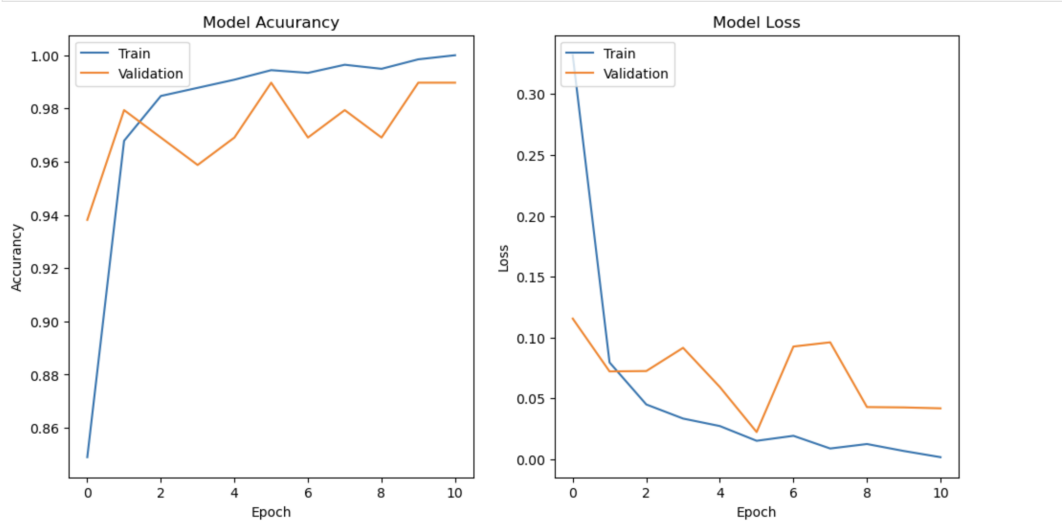

Figure 3.14: Model's loss and accuracy graph for 3D CNN model

The classification report displays an accuracy of 99 percent and with a confusion matrix displaying that 48 out 48 videos are correctly predicted for politics and 49 out of 50 videos are predicted to be workers from a test set of 98 videos. The misclassified are about 1 and 0. While the precision and recall are 99 percent.

	precision	recall	f1-score	support
POLITICS	0.98	1.00	0.99	48
WORKERS	1.00	0.98	0.99	50
accuracy			0.99	98
macro avg	0.99	0.99	0.99	98
weighted avg	0.99	0.99	0.99	98

```
Confusion Matrix:
[[48  0]
 [ 1 49]]
```

Figure 3.15: Classification report for multimodel

3.2.5 LSTM audio Model Analysis report

- For the LSTM model using audio, it was interesting to see the model converging at certain point, with a defined accuracy. The graphical figure displays the trends over the epoch and accuracy, even Model doesn't suffer from over fitting. There are several fluctuation for validation across the epochs and slighter fluctuation for the train data.

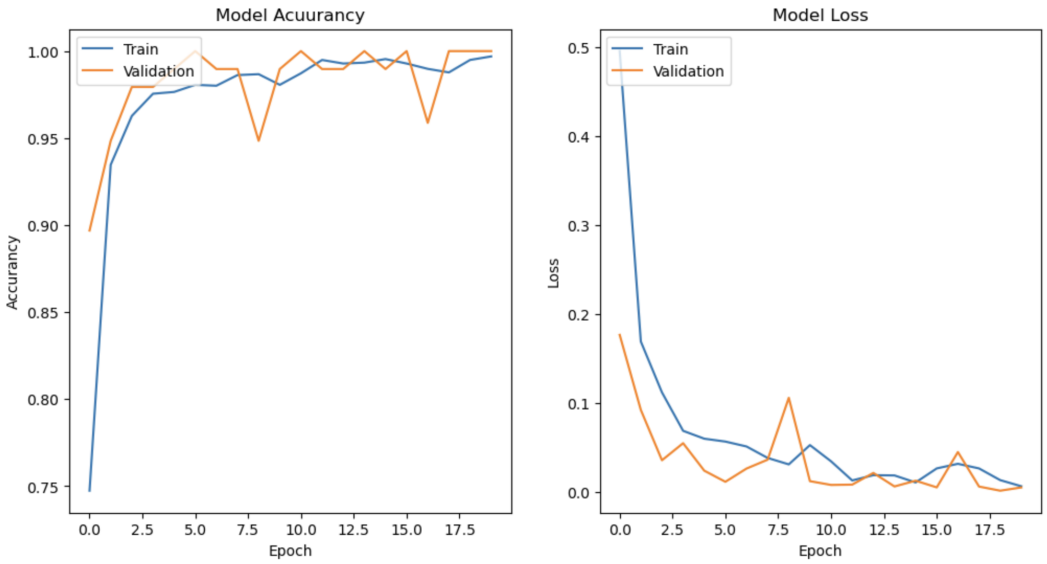

Figure 3.16: Model's loss and accuracy for LSTM model using audio data

The Classification shows 100 per cent accuracy, with a confusion matrix determining that 48 out of 48 soundtracks were correctly predicted for politics and 50 out of 50 soundtracks were correctly predicted for workers. While none were misclassified from

```
              precision    recall  f1-score   support

    POLITICS       1.00      1.00      1.00        48
     WORKERS       1.00      1.00      1.00        50

    accuracy                           1.00        98
   macro avg       1.00      1.00      1.00        98
weighted avg       1.00      1.00      1.00        98


Confusion Matrix:
[[48  0]
 [ 0 50]]
```

Figure 3.17: Classification report for LSTM audio model

the test set of 98 soundtracks. This suggests that different tones or phonetics, like politics, sounded different from workers.

3.3 Code implementation for method 1

For preprocessing steps:Colab

For 2D cnn, vggNet, Resnet model:colab

3D CNN, Multi model, LSTM on audio data:- Colab

Machine learning : Colab

3.4 Method 2

The same strategy is applied of method-1 in method-2, The only difference is noise removal using different filtering techniques which are already briefly mention in hands on practice, the code over here targets the mouth extraction and the removal of noise and a sharped image is an addition of weights of morphed method and Gaussian blur method producing a sharp image. The code has some updates over the pipeline, where it can be applied over a larger dataset.

3.5 Results for method 2

3.5.1 3D CNN Method 2

Considering the results can be ambiguous for the reason of systems halts and memory disruption, The figure below displays the training phase of the model. The model ran for 41 out of 100 epoch, and the validation weakens after the 10th epoch displaying overfitting. For classification report displays an accuracy of 65 percent, But while

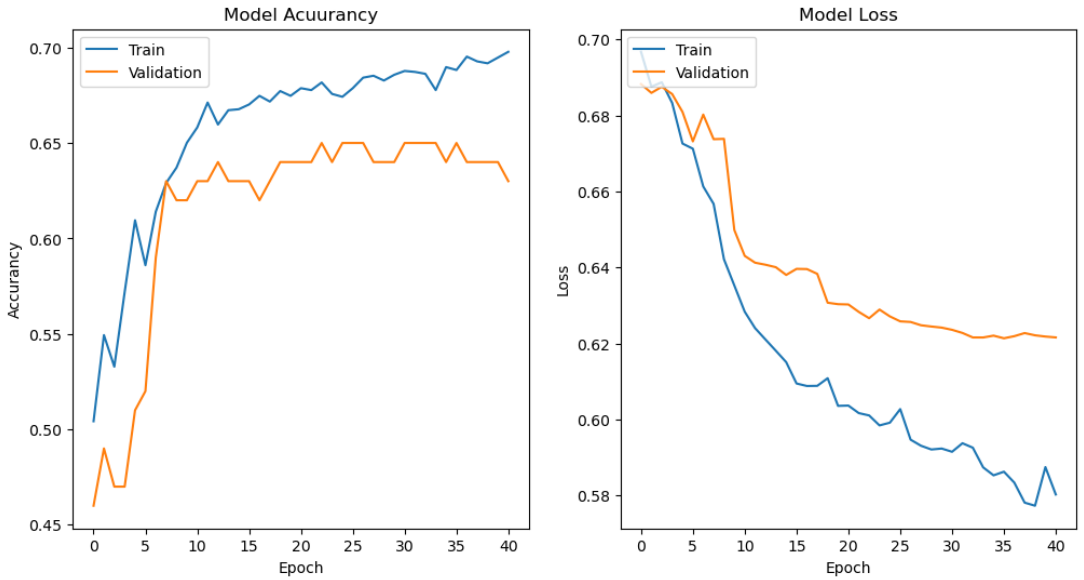

Figure 3.18: Model's accuracy and loss pattern

	precision	recall	f1-score	support
POLITICS	0.67	0.58	0.62	50
WORKERS	0.63	0.72	0.67	50
accuracy			0.65	100
macro avg	0.65	0.65	0.65	100
weighted avg	0.65	0.65	0.65	100

Confusion Matrix:
[[29 21]
[14 36]]

Figure 3.19: Classification Report

ruining the method, there were different accuracy's ranging from 67 percent at the first phase and later 65 percent. The confusion matrix display that 29 videos out of 50 were successfully classified as politics and 36 were classified as workers the remaining

were misclassified. The precision and recall for Politics are 67 percent and 58 percent, Recall are 63 percent and 72 percent. Comparing method 1 and 2, it displays that noise reduction can enhance the interpret ability of the images and hence produces a better accuracy.

3.5.2 Results for Multi Model uysing Method 2

The graph displays the models training and validation evaluation, which displays that, the model is learning well with audio and frame. The classification report displays an

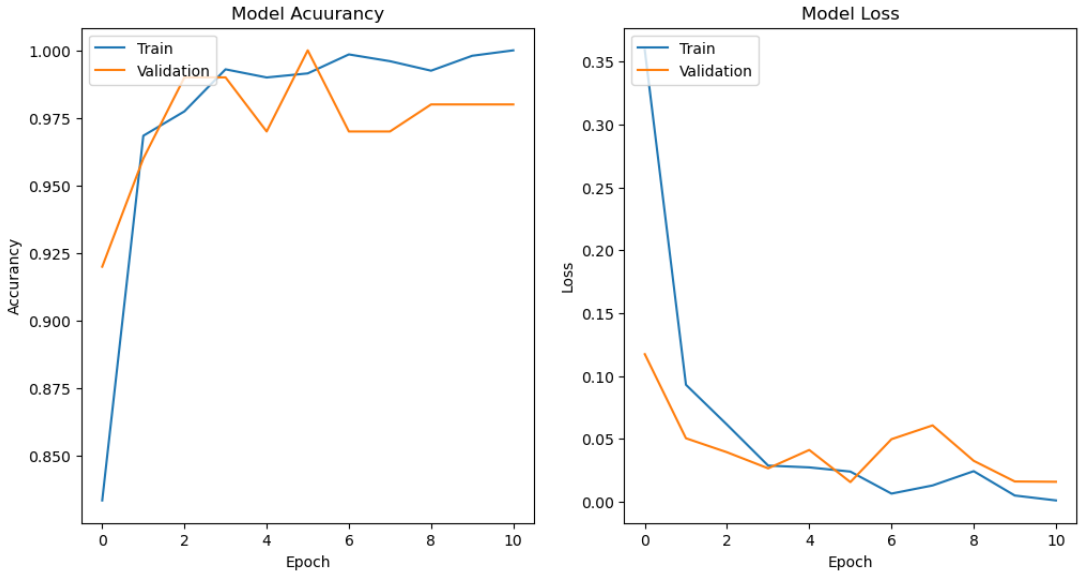

Figure 3.20: Models accuracy and loss pattern

	precision	recall	f1-score	support
POLITICS	0.96	1.00	0.98	50
WORKERS	1.00	0.96	0.98	50
accuracy			0.98	100
macro avg	0.98	0.98	0.98	100
weighted avg	0.98	0.98	0.98	100

```
Confusion Matrix:
[[50  0]
 [ 2 48]]
```

Figure 3.21: Classification report for method 2 using multi model

accuracy of 98 percent which is one percent less with respect to method one, However The confusion matrices display that 50 out of 50 are correctly classified to be politics and 48 out 50 are classified as workers. Only two videos are misclassified.

3.5.3 Results for 2D CNN, ResNet and VggNet Method 2

All the three methods display a deep over fitting of the model for 50 epoch, The

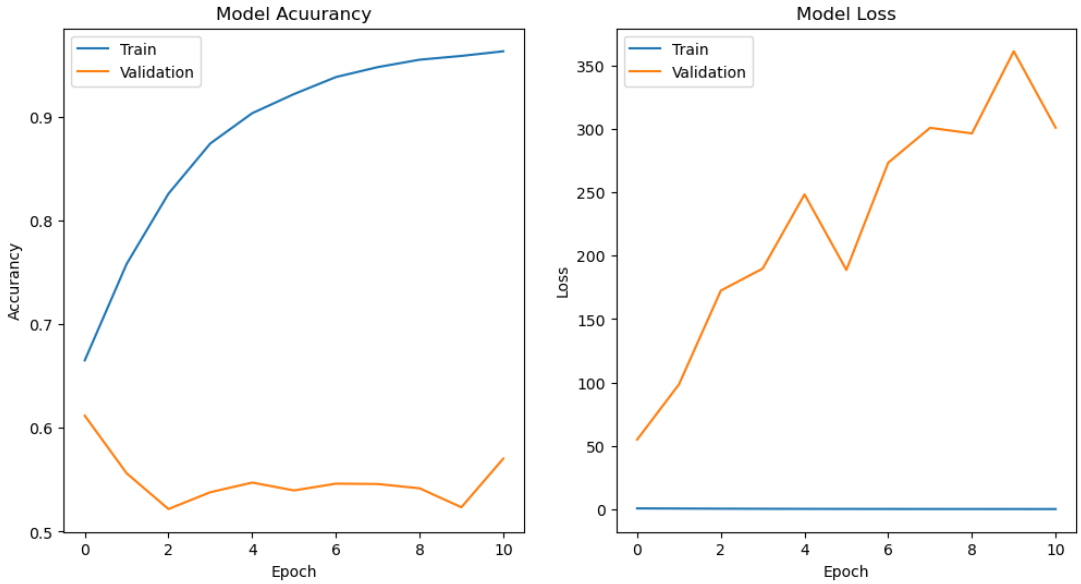

Figure 3.22: 2D CNN model's performance on training and validation accuracy and losses

```
Classification Report:
              precision    recall  f1-score   support

           0       0.59      0.68      0.63      1450
           1       0.63      0.54      0.58      1450

    accuracy                           0.61      2900
   macro avg       0.61      0.61      0.61      2900
weighted avg       0.61      0.61      0.61      2900


Confusion Matrix:
[[983 670]
 [467 780]]
```

Figure 3.23: Classification report for 2D CNN model

classification report for 2D CNN model displays a models accuracy of 61 percent, which

is comparioble lesser then from the method one, While the confusion matrix displays that 983 out of 1450 fames where rightly classfied to be politics and 780 fames out of 1450 frames where classified to be workers. The remain frames where missclassified.

3.5.4 Results for VggNet Method 2

The VggNet was fitted with a batch size of 64 and an enoch that ran for 11 out of 55 epoch. The classification report displays an accuracy of 61 percent and the confusion

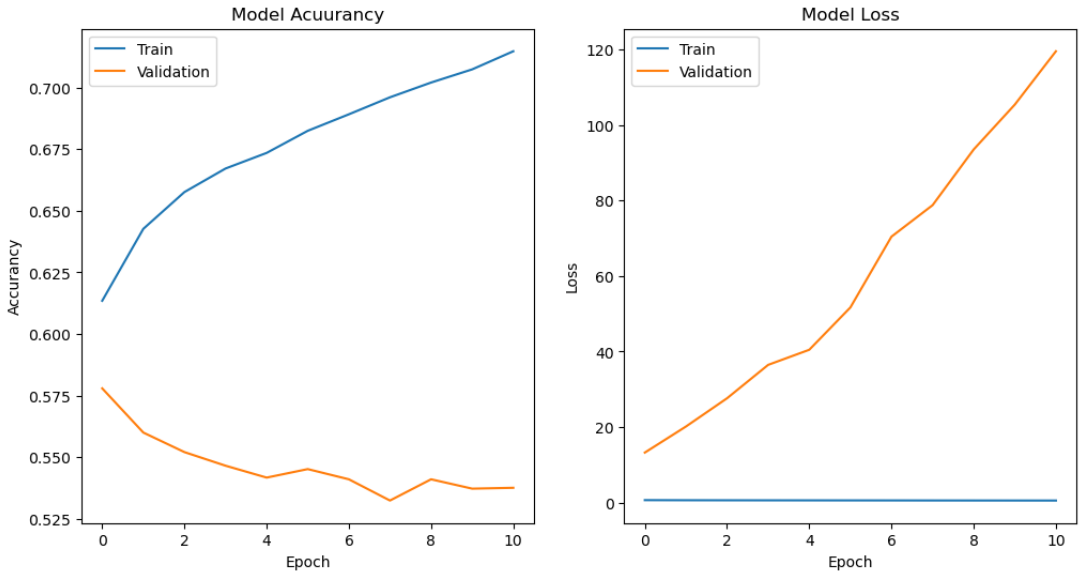

Figure 3.24: VggNet model's performance based on method 2

```
Classification Report:
              precision    recall  f1-score   support

           0       0.59      0.68      0.63      1450
           1       0.63      0.54      0.58      1450

    accuracy                           0.61      2900
   macro avg       0.61      0.61      0.61      2900
weighted avg       0.61      0.61      0.61      2900


Confusion Matrix:
[[983 670]
 [467 780]]
```

Figure 3.25: Classification report for VggNet using method 2

matrix display that 983 out of 1450 where classified tpo be politics and 780 out of 1450

frames where classified to be workers. Other where miss classification.

3.5.5 Results for ResNet Method 2

The ResNet was fitted using the same parameters that of the VGGNet AND THE 2d cNN model.

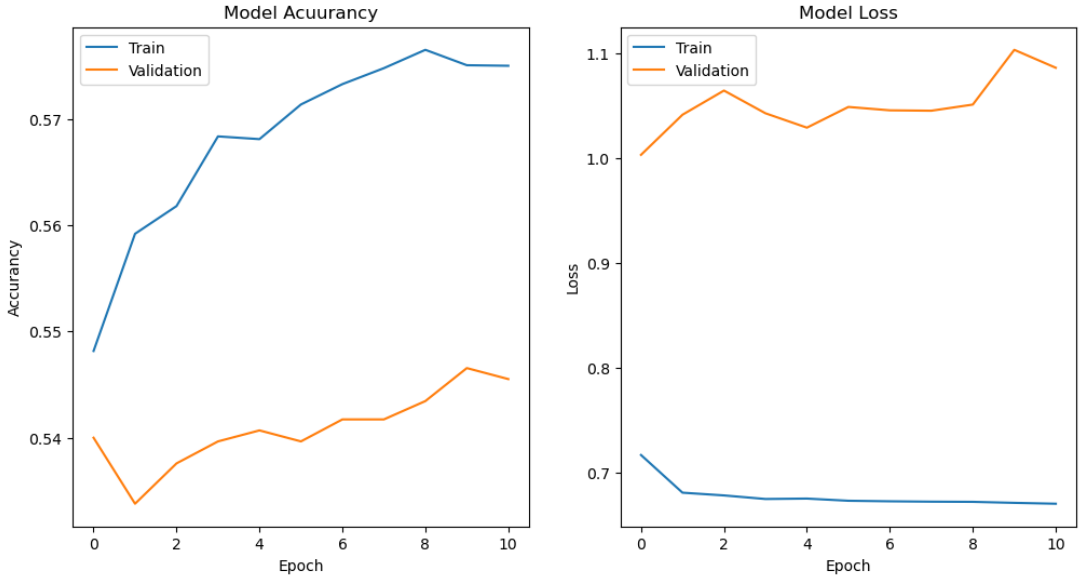

Figure 3.26: ResNet Models Performance based on Method 2

```
Classification Report:
              precision    recall  f1-score   support

           0       0.66      0.32      0.43      1450
           1       0.55      0.83      0.66      1450

    accuracy                           0.58      2900
   macro avg       0.60      0.58      0.55      2900
weighted avg       0.60      0.58      0.55      2900


Confusion Matrix:
[[ 465  243]
 [ 985 1207]]
```

Figure 3.27: Classification report for

The classification report display a accuracy of the model to be 58 percent and the confusion matrix display that 465 out of 1450 frames were classfied to be politics and

1207 out of 1450 frames to be workers, It seems interesting that resNet tends to work well for the second class. The other values present on the confusion matrices are misclassification.

3.5.6 Results for Machine learning models (SVM, Decision Tree) using Method 2

	precision	recall	f1-score	support
0	0.62	0.61	0.61	1450
1	0.62	0.62	0.62	1450
accuracy			0.62	2900
macro avg	0.62	0.62	0.62	2900
weighted avg	0.62	0.62	0.62	2900

```
: array([[884, 566],
        [544, 906]])
```

Figure 3.28: Classification report for SVM

the accuracy of the model is 62 percent and the classification report display that 884 out 1450 frames were correctly classified to be politics and 906 out of 1450 were correctly classified to be workers, while others are misclassified.

	precision	recall	f1-score	support
0	0.54	0.54	0.54	1450
1	0.54	0.54	0.54	1450
accuracy			0.54	2900
macro avg	0.54	0.54	0.54	2900
weighted avg	0.54	0.54	0.54	2900

```
: array([[783, 667],
        [667, 783]])
```

Figure 3.29: Classification report for decision tree using method 2

For the decision tree the accuracy of the model is 54 percent with the confusion matrics with 783 frames correctly classify politics while 783 frames correctly classify to be workers out of 1450 frames.

3.6 Code implementation for method 2

For preprocessing steps:Colab

For 2D CNN Model Data from colab

For 3D CNN and others : Data From Google Colab Link

For machine learning Google colab link

3.7 Some More methods applied

There are also more methods that aren't proceeded due to the time phase, this are some of the displayed method for example the kolman's method. The Kolmans method was initiated on the second last week there wasn't enough time to test the model several times as a fact of running out of application memory. The application of Kalman's filter is that, it's able to tract the frames flow in a video, or the optical flow. Kalmans filtering method enables it by estiminating the probabilities of the state of a system at each time step, even in the presence of noise. The Kalman's filtering method is called from the openCV2 class, for example (cv2.kalmanfilter), the syntax used over here are analysed through a blog [57], detailed the information to use this method and syntaxes. Also tried performing affine transformer, the syntax applied over here is from [25]. The result for Kalmans transform remains the same as that of the method 2 applied over 3D CNN.

Code for Kalman's data link

3.8 Test on homophones

3.8.1 3D CNN on Homophomnes

The classification algorithm is used over the word "other" and "others", having a bit similar sound, the training model accuracy displays several fluctuations in the model, while also in model loss.

The classification displays an accuracy of 56 percent with different precision and recall score, the recall for others is 18 percent, while for other it's 94 percent. Moreover, the confusion matrix displays about 47 out of 50 videos are considered to be others

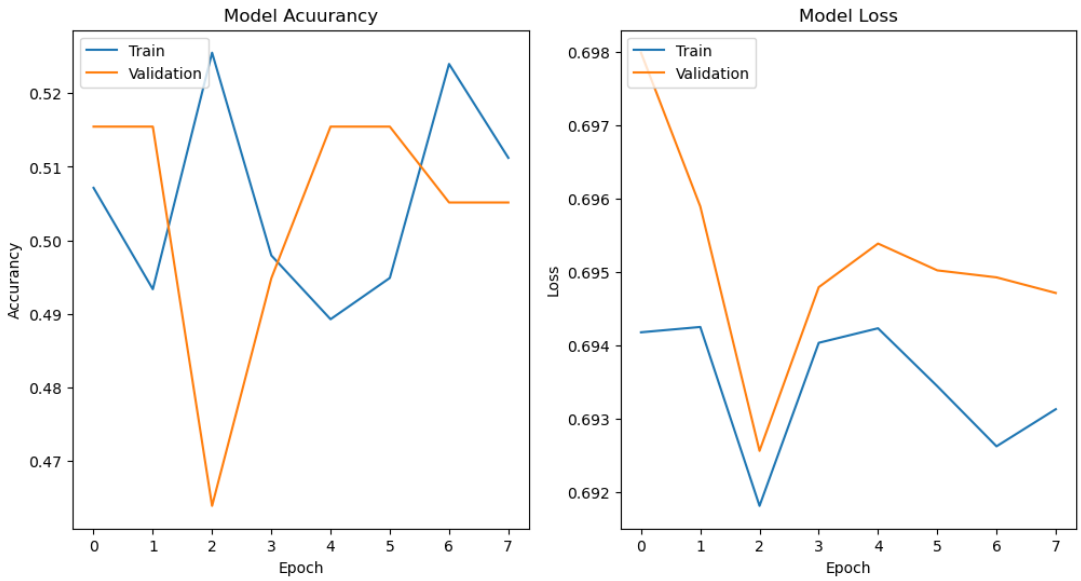

Figure 3.30: Training observation for homophones

	precision	recall	f1-score	support
OTHER	0.53	0.94	0.68	50
OTHERS	0.75	0.18	0.29	50
accuracy			0.56	100
macro avg	0.64	0.56	0.49	100
weighted avg	0.64	0.56	0.49	100

Confusion Matrix:
[[47 3]
[41 9]]

Figure 3.31: Classification report on homphones

while 9 out of 50 videos are considered to be other. This shows an hard impact of homophones. Code for 3D CNN Model Link

3.8.2 Multi Model on Homophones

The model displays the validation accuracy gradually increases with several fluctuation, while the training accuracy performs well. The classification displays an improved

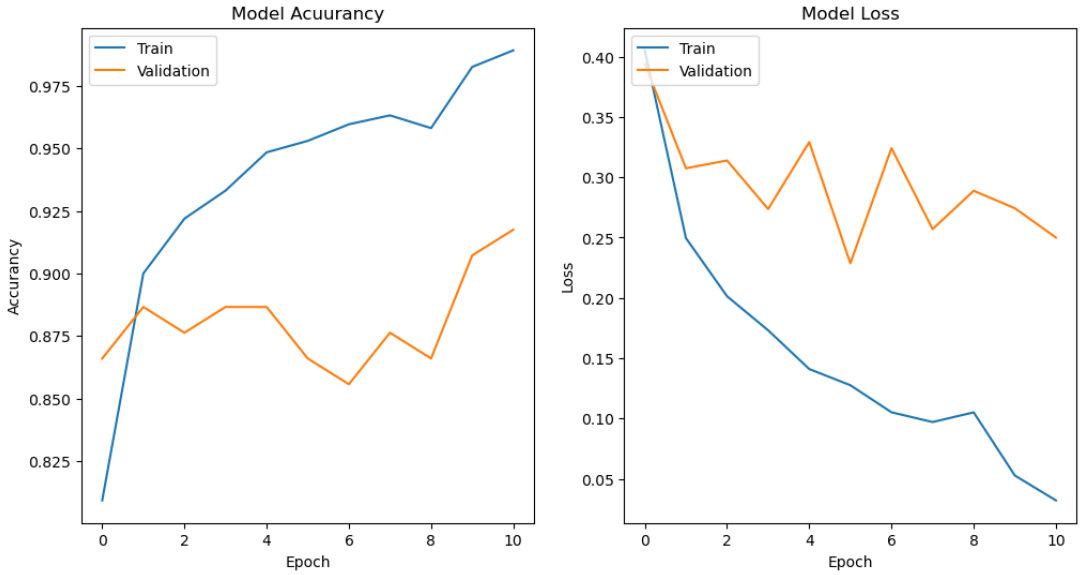

Figure 3.32: Multi Model training process on homophones

	precision	recall	f1-score	support
OTHER	0.88	0.98	0.92	50
OTHERS	0.98	0.86	0.91	50
accuracy			0.92	100
macro avg	0.93	0.92	0.92	100
weighted avg	0.93	0.92	0.92	100

```
Confusion Matrix:
[[49  1]
 [ 7 43]]
```

Figure 3.33: Classification report on homophones

model over the audio based model, with an accuracy of 92 percent and a confusion matrix of 49 out 50 videos correctly classified as others and 43 out of 50 videos classified to be other and rest misclassified. Code for MultiModel is : Link

For the pre-processing section for others and other link

CHAPTER 4

Conclusions

From the methods performed and the results obtained, infer that visual speech recognition alone cannot perform optimum unless combining it with audio, as a fact the multi model provides better classification through the use of confusion matrix. Perhaps, Several methods could be implemented in the pre-processing section for example the affinity transformers, Kalman filtering with more noise removal filters thereby passing all the frames further for feature extraction using spatiotemporal features by CNN and by using timedistributed with Bidirectional LSTM for converging Audio with frames for a perfect model. Indeed also DTW and HMMs can be used to combines frames and phonetics. There was also a need to perform different types noise reduction test, and as a point of consideration, different images had different light intensities therefore noise reduction parameters would vary across different frames, for example the types of edges present in the images. The lrw and lrs are challenging dataset cause its not a lab produced dataset over which it can be further analysed using landmark plots over the face, as an approach if the person moves either direction the eigan values can me measured and thereby using graphics for tilted faces. Also there was a need to perform different loss detection test, like the CTC test and audio loss detection.

Bibliography

[1] Use live speech and personal voice on your iphone, ipad, or mac. https://support.apple.com/en-gb/111869, 2023. Accessed: 2024-09-05.

[2] Self-Driving 5. scikit-image hog, 2024. Accessed: 2024-09-09.

[3] Madina Abdrakhmanova, Askat Kuzdeuov, Sheikh Jarju, Yerbolat Khassanov, Michael Lewis, and Huseyin Atakan Varol. Speakingfaces: A large-scale multimodal dataset of voice commands with visual and thermal video streams. *Sensors*, 21(10), 2021.

[4] Abien Fred Agarap. Deep learning using rectified linear units (relu), 2019.

[5] Young Jin Ahn, Jungwoo Park, Sangha Park, Jonghyun Choi, and Kee-Eung Kim. Syncvsr: Data-efficient visual speech recognition with end-to-end crossmodal audio token synchronization, 2024.

[6] Bagher Baba Ali, Waldemar Wojcik, Orken Mamyrbayev, Mussa Turdalyuly, and Nurbapa Mekebayev. Speech recognizer-based non-uniform spectral compression for robust mfcc feature extraction. *Przeglad Elektrotechniczny*, 94(6):90–93, 2018.

[7] Timur R. Almaev and Michel F. Valstar. Local gabor binary patterns from three orthogonal planes for automatic facial expression recognition. In *2013 Humaine Association Conference on Affective Computing and Intelligent Interaction*, pages 356–361, 2013.

[8] Yannis M Assael, Brendan Shillingford, Shimon Whiteson, and Nando De Freitas. Lipnet: End-to-end sentence-level lipreading. *arXiv preprint arXiv:1611.01599*, 2016.

[9] Jason Brownlee. Softmax activation function with python. https://machinelearningmastery.com/

softmax-activation-function-with-python/, 2020. Accessed: 2024-09-09.

[10] Hanni Bunny. Gaussian filter and derivatives of gaussian. https://hannibunny.github.io/orbook/preprocessing/04gaussianDerivatives.html, 2024. Accessed: 2024-09-17.

[11] Joon Son Chung and Andrew Zisserman. Lip reading in the wild. In *Asian Conference on Computer Vision*, 2016.

[12] Joon Son Chung and Andrew Zisserman. Lip reading in the wild. In Shang-Hong Lai, Vincent Lepetit, Ko Nishino, and Yoichi Sato, editors, *Computer Vision – ACCV 2016*, pages 87–103, Cham, 2017. Springer International Publishing.

[13] The Python Code. How to extract audio from video in python, n.d. Accessed: 2024-09-07.

[14] Dlib Developers. dlib_pybind11.get_frontal_face_detector. http://dlib.net/python/index.html#dlib_pybind11.get_frontal_face_detector, 2024. Accessed: 2024-09-09.

[15] Librosa Developers. Librosa documentation. https://librosa.org/doc/latest/index.html, 2023. Accessed: 2024-09-08.

[16] OpenCV Developers. Drawing functions in opencv. https://docs.opencv.org/4.x/dc/da5/tutorial_py_drawing_functions.html, 2024. Accessed: 2024-09-09.

[17] TensorFlow developers. tf.keras.callbacks.earlystopping, 2024. Accessed: 2024-09-09.

[18] TensorFlow developers. tf.keras.callbacks.reducelronplateau, 2024. Accessed: 2024-09-09.

[19] TensorFlow Developers. tf.keras.layers.lstm. https://www.tensorflow.org/api_docs/python/tf/keras/layers/LSTM, 2024. Accessed: 2024-09-09.

[20] TensorFlow Developers. tf.keras.preprocessing.image.imagedatagenerator. https://www.tensorflow.org/api_docs/python/tf/keras/

preprocessing/image/ImageDataGenerator, 2024. Accessed: 2024-09-09.

[21] TensorFlow Developers. tf.keras.sequential. https://www.tensorflow.org/api_docs/python/tf/keras/Sequential, 2024. Accessed: 2024-09-09.

[22] Mahidhar Dwarampudi and N V Subba Reddy. Effects of padding on lstms and cnns, 2019.

[23] Adriana Fernandez-Lopez and Federico M. Sukno. Survey on automatic lip-reading in the era of deep learning. *Image and Vision Computing*, 78:53–72, 2018.

[24] Yun Fu, Xi Zhou, Ming Liu, Mark Hasegawa-Johnson, and Thomas S. Huang. Lipreading by locality discriminant graph. In *2007 IEEE International Conference on Image Processing*, volume 3, pages III – 325–III – 328, 2007.

[25] GeeksforGeeks. Python opencv | affine transformation, 2023. Accessed: 2024-09-11.

[26] GeeksforGeeks. Python opencv | cv2.imread() method, 2024. Accessed: 2024-09-09.

[27] Berthold K. P. Horn. Edge detection. Technical report, Columbia University, Department of Computer Science, 1986. Accessed: 2024-09-17.

[28] Davis E King. Dlib-ml: A machine learning toolkit. *The Journal of Machine Learning Research*, 10:1755–1758, 2009.

[29] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[30] L Ashok Kumar, D Karthika Renuka, S Lovelyn Rose, I Made Wartana, et al. Deep learning based assistive technology on audio visual speech recognition for hearing impaired. *International Journal of Cognitive Computing in Engineering*, 3:24–30, 2022.

[31] Scikit learn developers. Model evaluation, 2024. Accessed: 2024-09-09.

[32] Scikit learn developers. sklearn.preprocessing.labelencoder, 2024. Accessed: 2024-09-09.

[33] Scikit learn developers. Support vector machines, 2024. Accessed: 2024-09-09.

[34] LinkedIn Learning. Build cutting-edge image recognition systems, 2018. Accessed: 2024-09-07.

[35] LinkedIn Learning. Build cutting-edge facial recognition systems, 2024. Accessed: 2024-09-07.

[36] Librosa. librosa.feature.mfcc. https://librosa.org/doc/main/generated/librosa.feature.mfcc.html, 2023. Accessed: 2024-09-08.

[37] Harry McGurk and John MacDonald. Hearing lips and seeing voices. *Nature*, 264(5588):746–748, 1976.

[38] M Musalia, Shondipon Laha, J Cazalilla-Chica, J Allan, L Roach, J Twamley, S Nanda, M Verlander, A Williams, I Kempe, et al. A user evaluation of speech/phrase recognition software in critically ill patients: a decide-ai feasibility study. *Critical Care*, 27(1):277, 2023.

[39] Neptune.ai. 10 image processing libraries for machine learning in python, 2023. Accessed: 2024-09-05.

[40] Hea Choon Ngo, Ummi Rabaâah Hashim, Raja Rina Raja Ikram, Lizawati Salahuddin, and Mok Lee Teoh. A pipeline to data preprocessing for lipreading and audio-visual speech recognition. *International Journal*, 9(4), 2020.

[41] Victor Olufemi. Face detection, 2021. Accessed: 2024-09-05.

[42] Şaban Öztürk and Bayram Akdemir. Effects of histopathological image preprocessing on convolutional neural networks. *Procedia computer science*, 132:396–403, 2018.

[43] Laxmi Pandey and Ahmed Sabbir Arif. Liptype: A silent speech recognizer augmented with an independent repair model. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–19, 2021.

[44] E. D. Petajan. *Automatic Lipreading To Enhance Speech Recognition (speech Reading)*. PhD thesis, ProQuest Dissertations & Theses Global, 1984. Order No. 8502266.

[45] Baptiste Pouthier, Laurent Pilati, Giacomo Valenti, Charles Bouveyron, and Frédéric Precioso. Another Point of View on Visual Speech Recognition. In *24th*

INTERSPEECH Conference, volume 2023, pages 4089–4093, Dublin, Ireland, August 2023. ISCA.

[46] Mengyang Pu, Yaping Huang, Qingji Guan, and Haibin Ling. Rindnet: Edge detection for discontinuity in reflectance, illumination, normal and depth. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6879–6888, October 2021.

[47] PyImageSearch. Opencv morphological operations, 2021. Accessed: 2024-09-05.

[48] Preeth Raguraman, Mohan R., and Midhula Vijayan. Librosa based assessment tool for music information retrieval systems. In *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 109–114, 2019.

[49] Changchong Sheng, Xinzhong Zhu, Huiying Xu, Matti PietikÃ¤inen, and Li Liu. Adaptive semantic-spatio-temporal graph convolutional network for lip reading. *IEEE Transactions on Multimedia*, 24:3545–3557, 2022.

[50] Sami Sieranoja, Md Sahidullah, Tomi Kinnunen, Jukka Komulainen, and Abdenour Hadid. Audiovisual synchrony detection with optimized audio features. In *2018 IEEE 3rd International Conference on Signal and Image Processing (ICSIP)*, pages 377–381, July 2018.

[51] Joon Son Chung, Andrew Senior, Oriol Vinyals, and Andrew Zisserman. Lip reading sentences in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[52] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[53] Walmart Global Tech. Time series similarity using dynamic time warping explained. *Medium*, 2020. Accessed: 2024-09-13.

[54] TensorFlow. Understanding masking and padding. https://www.tensorflow.org/guide/keras/understanding_masking_and_padding, 2023. Accessed: 2024-09-08.

[55] TensorFlow Authors. Data augmentation. https://www.tensorflow.org/tutorials/images/data_augmentation, 2023. Accessed: 2024-09-08.

[56] TensorFlow Authors. Video classification with a 3d convolutional neural network. https://www.tensorflow.org/tutorials/video/video_classification, 2023. Accessed: 2024-09-08.

[57] Pierian Training. Kalman filter opencv python example, 2023. Accessed: 2024-09-11.

[58] Henrique Vedoveli. Blur in image processing: An introductory guide, 2023. Accessed: 2024-09-05.

[59] Warren J Von Eschenbach. Transparency and the black box problem: Why we do not trust ai. *Philosophy & Technology*, 34(4):1607–1622, 2021.

[60] Zhou Wang and Alan C. Bovik. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE Signal Processing Magazine*, 26(1):98–117, 2009.

[61] Wikipedia. Mcgurk effect — wikipedia, the free encyclopedia, 2024. [Online; accessed 13-August-2024].

[62] Wikipedia contributors. Consonance and dissonance — Wikipedia, the free encyclopedia, 2024. [Online; accessed 15-August-2024].

[63] Wikipedia contributors. Data augmentation — Wikipedia, the free encyclopedia, 2024. [Online; accessed 27-August-2024].

[64] Wikipedia contributors. Discrete laplace operator — Wikipedia, the free encyclopedia, 2024. [Online; accessed 15-August-2024].

[65] Wikipedia contributors. Dlib — Wikipedia, the free encyclopedia, 2024. [Online; accessed 14-August-2024].

[66] Wikipedia contributors. Dynamic time warping — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Dynamic_time_warping&oldid=1227533339, 2024. [Online; accessed 17-August-2024].

[67] Wikipedia contributors. Early stopping — Wikipedia, the free encyclopedia, 2024. [Online; accessed 28-August-2024].

[68] Wikipedia contributors. Grayscale — Wikipedia, the free encyclopedia, 2024. [Online; accessed 14-August-2024].

[69] Wikipedia contributors. Histogram of oriented gradients — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Histogram_of_oriented_gradients&oldid=1239612892, 2024. [Online; accessed 14-August-2024].

[70] Wikipedia contributors. Lip reading — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Lip_reading&oldid=1230026729, 2024. [Online; accessed 12-August-2024].

[71] Wikipedia contributors. Nonlinear dimensionality reduction — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Nonlinear_dimensionality_reduction&oldid=1238822556, 2024. [Online; accessed 18-August-2024].

[72] Wikipedia contributors. Optical flow — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Optical_flow&oldid=1226182231, 2024. [Online; accessed 18-August-2024].

[73] Wikipedia contributors. Point cloud — Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Point_cloud&oldid=1234699797, 2024. [Online; accessed 17-August-2024].

[74] Wikipedia contributors. Wiener filter — Wikipedia, the free encyclopedia, 2024. [Online; accessed 3-September-2024].

[75] Ni Ni Win, Khin Kyaw, Thu Win, and Phyo Aung. Image noise reduction using linear and nonlinear filtering techniques. *Int J Sci Res Publ*, 9(8):816–821, 2019.

[76] Dujuan Zhang, Jie Li, and Zhenfang Shan. Implementation of dlib deep learning face recognition technology. In *2020 International Conference on Robots Intelligent System (ICRIS)*, pages 88–91, 2020.