MVJ COLLEGE OF ENGINEERING
Since 1982

An Autonomous Institute
(Affiliated to Visvesvaraya Technological University, Belagavi
Approved By AICTE, New Delhi,
Recognized by UGC under 2(f) & 12(B)
Accredited by NBA and NAAC)

PROJECT PHASE-2 REPORT

ON

"SPEECH RECOGNITION USING MACHINE LEARNING"

Submitted in partial fulfillment of requirements for the award of degree of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE & ENGINEERING

Submitted By:

GUNDA NIKITHA SRINIVAS	**1MJ20CS074**
ISWARYA V	**1MJ20CS086**
KRITHI NAGA SAI VANGALA	**1MJ20CS101**
NISHA ELIZABETH RENJI	**1MJ20CS137**

Under the Guidance of
Ms. Navya K S
Assistant Professor, Department of CSE.

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
MVJ COLLEGE OF ENGINEERING
BANGALORE-67
ACADEMIC YEAR 2023-24

MVJ COLLEGE OF ENGINEERING
Since 1982
Engineering A Better Tomorrow

An Autonomous Institute

MVJ COLLEGE OF ENGINEERING

Whitefield, Near ITPB, Bangalore-67

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that phase-II of the project work, entitled **“SPEECH RECONITION USING MACHINE LEARNING”** is a bonafide work carried out

GUNDA NIKITHA SRINIVAS	**1MJ20CS074**
ISWARYA V	**1MJ20CS086**
KRITHI NAGA SAI VANGALA	**1MJ20CS101**
NISHA ELIZABETH RENJI	**1MJ20CS137**

in partial fulfillment for the award of degree of Bachelor of Engineering in Computer Science & Engineering of the Visvesvaraya Technological University, Belagavi during the academic year 2023-24. It is certified that all the corrections/suggestions indicated for internal assessment have been incorporated in the report. The project report has been approved as it satisfies the academic requirements

Signature of the Guide	**Signature of the HOD**	**Signature of the Principal**
(Ms. Navya K S)	**(Dr. Kiran Babu T.S)**	**(Dr. Suresh Babu V)**

Name of the Examiners: **Signature with Date**

1.
2.

MVJ COLLEGE OF ENGINEERING Since 1982
Engineering A Better Tomorrow

An Autonomous Institute

MVJ COLLEGE OF ENGINEERING

Whitefield, Near ITPB, Bangalore-67

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We,

GUNDA NIKITHA SRINIVAS	**1MJ20CS074**
ISWARYA V	**1MJ20CS086**
KRITHI NAGA SAI VANGALA	**1MJ20CS101**
NISHA ELIZABETH RENJI	**1MJ20CS137**

hereby declare that the entire Phase-II work of the project titled **“SPEECH RECOGNITION USING MACHINE LEARNING”** embodied in this project report has been carried out by us during the 8th semester of B.E. degree at MVJCE, Bangalore under the esteemed guidance of **Ms. Navya K S, Assistant Professor, Dept. of CSE,** MVJCE affiliated to Visvesvaraya Technological University, Belagavi. The workembodied in this dissertation work is original and it has not been submitted in part or full for any other degree in any University.

Place: MVJCE, Bangalore

Date:

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany a successful completion of any task would be incomplete without the mention of people who made it possible, success is the epitome of hard work and perseverance, but steadfast of all is encouraging guidance.

So, with gratitude we acknowledge all those whose guidance and encouragement served as beacon of light and crowned our effort with success.

We are thankful to our Principal **Dr. Suresh Babu V,** for his encouragement and support throughout the project work.

We are thankful to our Vice Principal **Dr. Brindha M,** for her encouragement and support throughout the project work.

We are thankful to our COE **Dr. M A Lourdu Anthony Raj**, for his encouragement and support throughout the project work.

We are also thankful to our HOD, **Dr. Kiran Babu T.S, Dept. of CSE** for his incessant encouragement & all the help during the project work.

We consider it a privilege and honour to express our sincere gratitude to our guide

Ms. Navya K S, Assistant Professor, Dept. of CSE for her valuable guidance throughout the tenure of this project work, and whose support and encouragement made this work possible.

It is also an immense pleasure to express our deepest gratitude to all faculty members of our department for their cooperation and constructive criticism offered, which helped us a lot during our project work.

Finally, we would like to thank all our family members and friends whose encouragement and support was invaluable.

Thanking you

ABSTRACT

Machine Learning is widely used to detect the movement of lips. It has been observed that the data generated through visual motion of mouth and corresponding audio are highly correlated. This fact has been exploited for lip reading and for improving speech recognition. We propose a system that uses CNN (Convolutional Neural Network) which is trained and used to detect the movement of lips and predict the words being spoken. This trained CNN will be able to detect the words that are spoken within the video and display it in a text format. The CNN may also rely on additional information provided by the context, knowledge of the language, we hope to learn whether the utilization of machine learning, more specifically the DNN (Deep Neural Network), could also be an appropriate candidate for solving the problem of lip reading. The main aim of our project is to accurately recognize the phrases being spoken through automated lip reading

TABLE OF CONTENTS

CHAPTERS **PAGE NO.**

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Visual lip-reading plays an important role in human-computer interaction in noisy environments where audio speech recognition may be difficult. The art of lip reading has various applications, for example it can be used to help people with hearing disabilities, or possibly by security forces in situations where it is necessary to identify a person’s speech when the audio records are not available. However, like audio speech recognition, lip-reading systems also face several challenges due to variances in the inputs, such as with facial features, skin colors, speaking speeds, and intensities. It is almost impossible to manually create a computer algorithm that will be reading completely accurately from the lips. Even human professionals in this field can correctly estimate nearly every other word and can do so only under ideal conditions. Therefore, the complex task of lip reading is suitable candidate for extensive research in the field of deep learning. Lip reading is also an extremely difficult task because several different words can be spoken with almost indistinguishable lip movements. Therefore, the problem of lip reading provides unique challenges. This has led to numerous advancements in the field of automated speech recognitions systems using machine learning. Several models have been developed to improve hearing aids, for silent dictation in noisy public environments, identification for security purposes etc. However not until the use of Deep Learning did the accuracy of these models increase. The use of Deep Learning and deep neural networks has revolutionized the quality of automated lip reading systems due to the large amounts of data sets that can be used. There are mainly four stages involved in the technique used to perform automated lip reading. Namely, face detection, cropping module, feature extraction and text decoding. The primary aim of face detection algorithms is to determine whether there is any face in an image or not. The cropping module is used to crop out the region of interest (in this case, the lips) and feature extraction helps in extracting the required features. The task of decoding text based on the movement of lips is complex. It requires complete and extensive training of the model to be able to recognize lip movements.

CHAPTER 2

LITERATURE SURVEY

2.1 CONCATENATIVE SPEECH RECOGNITION USING MORPHEMES

AUTHORS: Afshan Jafri

This paper proposes a new approach to automatic speech recognition (ASR) that leverages morphemes, the building blocks of words. Traditional ASR systems rely on whole words, which can limit vocabulary size and struggle with uncommon phrases. This method, called Concatenative ASR, breaks down words into prefixes, suffixes, and stems (morphemes) and builds a dictionary of pronunciations for these smaller units. The system then uses grammatical rules to combine these morphemes into recognized words. This constrained approach reduces errors by preventing nonsensical combinations while still allowing for a vast vocabulary. The paper details the development of this ASR system using Arabic as an example. The researchers designed the system parameters specifically for Arabic and analyzed its performance compared to a standard word-based ASR. Their findings suggest that Concatenative ASR is most effective for large vocabularies (up to half a million words) while the standard model works best for smaller ones (up to five thousand words). The paper concludes that this method is applicable to any language that builds words by concatenating morphemes, offering a powerful tool for ASR with extensive vocabularies

2.2 AUTOMATIC SPEECH RECOGNITION SYSTEM

AUTHORS: Ishrat Sultana, Nirmaljeet Kaur Pannu

The paper on "Automatic Speech Recognition System" introduces a novel approach aimed at overcoming the limitations of existing technology. It proposes a hybrid model architecture that integrates deep learning techniques, including recurrent neural networks (RNNs), convolutional neural networks (CNNs), and transformer architectures, to enhance transcription accuracy. Additionally, the system incorporates multimodal input processing by integrating visual cues such as lip movement and facial expressions, improving recognition performance in challenging acoustic environments and for speakers with accents or speech impediments. Furthermore, advanced language models with contextual embeddings and attention mechanisms enable the system to better understand the semantic context of spoken utterances, leading to more accurate transcriptions. The paper emphasizes adaptability through transfer learning and continuous learning mechanisms, ensuring quick adaptation to new domains, languages, and user contexts. Moreover, the privacy-preserving design safeguards sensitive voice data through anonymization, encryption, and strict access controls, while real-time performance optimization ensures low latency and high throughput capabilities, making the proposed ASR system suitable for a wide range of applications and use cases.

2.3 A MODEL FOR THE APPLICATION OD AUTOMATIC SPEECH RECOGNITION FOR GENERATING LESSON SUMMARIES

AUTHORS: Phillip Blunt, Bertram Haskins

The paper proposes a system that leverages Automatic Speech Recognition (ASR) technology to automatically generate summaries of classroom lessons. This could be beneficial for students by providing a secondary resource to reinforce understanding of the material. The authors discuss the technical underpinnings of ASR, including feature extraction from speech recordings and different machine learning models for recognizing continuous speech. They acknowledge the challenges of background noise in classrooms and how to manage it for better accuracy. The core of the paper proposes a model for an ASR system that specifically targets educational environments. This system would transcribe lectures and then use additional techniques to generate summaries that focus on key topics. The envisioned benefits include providing students with a transcript of the lesson and highlighting the important keywords and their order of presentation. The authors mention that this model was used to develop a working prototype system, although the details of the prototype aren't included in the paper itself. Their research suggests this technology has promise for enhancing the learning experience.

2.4 HARRIS HAWKE’S SPARSE AUTO ENCODER NETWORKS FOR AUTOMATIC SPEECH RECOGNITION SYSTEM

AUTHORS: Chuan-Chi Lai, Chuan-Ming Liu, Mohammed Hasan Ali, Mustafa Musa, Jaber Sura, Khalid Abd, Ajed Rehman, Mazhar Javed Awan, Adzgauskiene Saeed Ali Bahaj

The paper proposes a method to improve Automatic Speech Recognition (ASR) systems, which convert spoken language into text. Traditional ASR systems struggle with background noise and variations in speech patterns (accents). The proposed system, HHSAE-ASR, tackles these challenges in two ways. First, it uses a machine learning technique called a Sparse Auto-Encoder (SAE) to analyze speech features extracted from audio. The SAE helps the system identify important characteristics of speech that distinguish it from noise. Second, HHSAE-ASR incorporates a method called Harris Hawks Optimization (HHO) to fine-tune the SAE. HHO is inspired by the hunting behavior of Harris hawks, and it helps the system learn from speech data more effectively. This fine-tuning improves the accuracy of ASR, especially in noisy environments. Overall, HHSAE-ASR aims to create a more robust ASR system by combining feature extraction with optimized learning techniques. This combination allows the system to handle the complexities of real-world speech patterns.

2.5DEVELOPING A SPEECH RECOGINITION SYSTEM FOR RECOGNIZING TONAL SPEECH SIGNALS USING A CONVOLUTIONAL NEURAL NETWORK

AUTHORS: Sakshi Dua, Sethuraman Sambath Kumar, Yasser Albagory, Rajakumar Ramalingam, Ankur Dumka, Rajesh Singh, Mamoon Rashid, Anita Gehlot, Sultan ,S. Alshamrani and Ahmed Saeed AlGhamdi

The paper proposes a new approach to speech recognition for languages that rely on pitch, or tonal languages. Traditional methods struggle with these languages because they focus on the order of sounds, not pitch variations. This research introduces a speech recognition system built on a convolutional neural network (CNN). CNNs are powerful for image recognition, and here, the authors adapt them to analyze tonal speech. The system takes speech audio as input and converts it into a format suitable for the CNN. The CNN then extracts features from the speech data that capture both the sequence of sounds and the pitch variations. Finally, the system classifies these features to recognize the spoken words. The authors developed a tonal speech dataset for training and testing the CNN. Their system achieved promising results, demonstrating that CNNs are a viable approach for tonal speech recognition. This paves the way for more accurate speech recognition technology for tonal languages.

CHAPTER 3

PROBLEM IDENTIFICATION AND PROPOSED SOLUTION

3.1 Existing System

Existing automatic speech recognition (ASR) systems utilize sophisticated algorithms to convert spoken language into text with high accuracy. These systems rely on deep learning models, like recurrent neural networks (RNNs) and transformer architectures, trained on vast amounts of speech data. Initially, audio input undergoes preprocessing, including noise reduction and feature extraction. The processed audio is then fed into the ASR model, which employs techniques like convolutional neural networks (CNNs) and attention mechanisms to decipher linguistic patterns and context. Despite advancements, challenges persist, particularly in handling accents, background noise, and contextual ambiguity. Continuous refinement through data augmentation and model optimization drives the evolution of ASR technology.

3.2 Demerits of Existing System

Automatic speech recognition (ASR) systems have revolutionized the way we interact with technology. From voice assistants like Siri and Alexa to dictation software and automated captioning, ASR has transformed the landscape of human-computer interaction. However, despite their undeniable advancements, these systems are not without their limitations. One of the most pressing challenges facing ASR is its susceptibility to errors. In noisy environments, the background chatter can easily overwhelm the speech signal, leading to misinterpretations and inaccuracies in transcriptions. This is particularly problematic for applications like voice-activated searches or dictation software, where precise understanding is crucial. Furthermore, ASR systems can struggle with speakers who have strong accents or speech impediments. The variations in pronunciation patterns introduced by accents or speech disorders can trip up the system's algorithms, resulting in garbled or nonsensical outputs. This highlights the need for ASR systems to become more adaptable and robust in recognizing diverse speech patterns. Another hurdle lies in the realm of context comprehension. ASR systems primarily focus on the acoustic features of speech, often neglecting the nuances of language such as sarcasm, humor, or double meanings. This can lead to misinterpretations, especially when dealing with ambiguous or colloquial language. For instance, the system might not differentiate between a sarcastic "great job" and a genuine one, potentially creating confusion in the communication. Scalability and adaptation to diverse languages and dialects pose another challenge. While ASR systems can be trained for specific languages, handling a multitude of languages and their regional variations requires extensive training data and fine-tuning. This can be a costly and time-consuming process, hindering the widespread adoption of ASR across different cultures and regions. Finally, privacy concerns loom large with ASR systems. The ability to capture and process voice data raises questions about data security and user consent. ASR systems may store voice recordings for training purposes or use them to personalize user experiences. It is crucial to ensure that user data is handled responsibly, with robust security measures in place and clear user consent obtained before processing any sensitive voice information. Addressing these challenges is paramount for the continued advancement of ASR technology. Researchers are exploring various avenues to improve accuracy and robustness. One promising approach involves leveraging deep learning techniques, which allow ASR systems to learn complex patterns from vast amounts of data. This can enhance the system's ability to handle noise variations and recognize diverse speech patterns. Furthermore, incorporating natural language processing (NLP) capabilities can equip ASR systems with a better understanding of context and language nuances. By analyzing the surrounding text and situation, NLP can help the system interpret ambiguous phrases and provide a more accurate understanding of the speaker's intent .In terms of scalability, advancements in data augmentation techniques can help create more diverse training datasets for different languages and dialects. This can allow ASR systems to adapt to a wider range of speech patterns without requiring massive amounts of real-world data collection. Finally, addressing privacy concerns requires robust regulations and user-centric design principles. Implementing clear opt-in mechanisms for data collection and storage, coupled with strong data encryption practices, can help build trust and ensure user privacy.In conclusion, while ASR technology has made significant strides, overcoming the limitations discussed above is essential for its continued success. By focusing on improving accuracy, robustness, and user privacy, we can pave the way for a future where ASR systems seamlessly bridge the gap between spoken language and machine comprehension.

3.3 Proposed System

We propose a system that will take in a video input from the user. This video is to be pre-processed and divided into frames of images. This is done to have non inclined values and to help recognize the face in a better manner. The next step is to detect the region of interest that is the mouth and crop it out. This cropped ROI is to be passed to the convoluted neural network (CNN) for further processing. Here the visual features are extracted, and the model is trained, based on which the spoken words are decoded.Pre-processing: Initially, the video is to be divided into frames of images. These frames of images obtained will most likely be in the RGB format. These images should then be converted to grayscale from RGB to avoid additional count of parameters present in an RGB image which is just an overhead to the system. The obtained set of frames from the video is then passed onto processing ,Face Detection and Cropping: Once the frames have been obtained from the video, proposed system will detect the face in the frame if it exists and for the simplicity of our project, we are assuming that our system will be able to detect faces with full frontal view only discarding the possibility of having partial or side views of a human. We plan to make use of the DLib face detector and landmark predictor with 68 landmarks making use of the Haar features to be able to detect a face in the frame. After the detection of the face, the frames with no face will be discarded. The next step will be to be able to identify our Region of Interest (ROI) which is the lips and the mouth region in this case. It is to be identified with help of the haar cascade classifier itself. Once the mouth region has been identified we will need to crop out the mouth region to be able to detect the mouth and the lip moment and for further processing and training of our system. The RGB channels need to be standardised to have zero mean and unit variance. After the process is done the images will be saved as a NumPy array with the cropped region images as values, it might look like the representation shown in Figure 4. The whole process of face detection and cropping is represented in Figure 3Text Classification and Decoding: Once the normalization is done, the data will be fed into the CNN for training and text decoding. The CNN learns on its own by having many epochs and passing the information learnt among the multiple hidden layers. The decoding will be done by matching the lip movement with the image data and the given dataset used for training, the word spoken will be predicted . The words spoken will then be embedded together for the whole video. The words predicted need to be put together to form the original sentence which was spoken by the individual in the dataset.

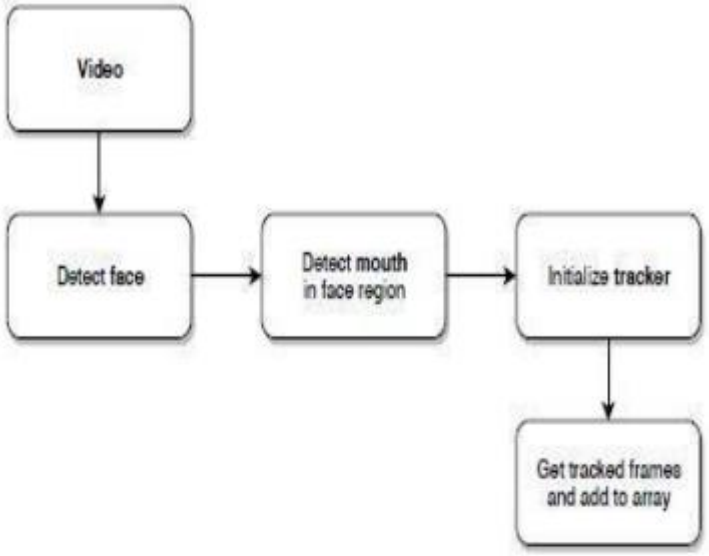

Video
Detect face
Detect mouth in face region
Initialize tracker
Get tracked frames and add to array

Fig : FACE DETECTION AND CROPPING PROCESS

Fig : Example of Numpy array sample

Feature extraction and normalization: After the images are stored as an array, the features from the ROI need to be extracted. The spatio temporal features need to be extracted and fed into the CNN as an input for training of the model [2]. Normalization of the image frames is necessary to avoid any irregularities in the dataset. For example, a person might take one second to pronounce a word, while another individual may take two seconds to pronounce the same word. Leaving such irregularities unattended may cause discrepancies in training and the results. So, we make use of normalization to be able to have an even Training data. Text Classification and Decoding: Once the normalization is done, the data will be fed into the CNN for training and text decoding. The CNN learns on its own by having many epochs and passing the information learnt among the multiple hidden layers. The decoding will be done by matching the lip movement with the image data and the given dataset used for training, the word spoken will be predicted [4]. The words spoken will then be embedded together for the whole video. The words predicted need to be put together to form the original sentence which was spoken by the individual in the dataset. The proposed system architecture is designed based on working of a Convoluted Neural Network (CNN). A Convolutional Neural Network is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in CNN is much lower as compared to other classification algorithms. It is designed with an input layer, three hidden layers and an output layer. A SoftMax layer can also be used as a probability classifier and max pooling to reduce the number of parameters for the consecutive layers. The system will be tested using both 3 hidden layer architecture as well as the 5 hidden layer architecture, but the 3-layer architecture will be given more priority due to the computation problems for 5-layer architecture.

3.4 Advantages of the proposed system

The proposed automatic speech recognition (ASR) system offers several advantages over existing technology, contributing to enhanced performance, adaptability, and user satisfaction:

- Improved Accuracy: By utilizing a hybrid model architecture that combines multiple deep learning techniques and incorporates multimodal input processing, the proposed system achieves higher accuracy in transcribing speech, even in challenging acoustic environments or for speakers with accents or speech impediments. The integration of visual cues such as lip movement and facial expressions further enhances accuracy, reducing transcription errors and improving overall recognition performance.
- Enhanced Adaptability: Through the use of transfer learning and continuous learning mechanisms, the proposed system demonstrates improved adaptability to new domains, languages, and user contexts. By fine-tuning pre-trained models on domain-specific or language-specific data and incorporating user feedback for model refinement, the system quickly adapts to evolving requirements without requiring extensive retraining, leading to

faster deployment and better performance in diverse environments.

- Context-aware Understanding: The incorporation of advanced language models with contextual embeddings and attention mechanisms enables the proposed system to better understand the semantic context of spoken utterances. By considering surrounding words and phrases, the system generates more accurate transcriptions, particularly in cases of homophones or ambiguous speech, resulting in improved comprehension and higher-quality output.
- Privacy-preserving Design: With a focus on data security and user confidentiality, the proposed ASR system adopts a privacy-preserving design that ensures the protection of sensitive voice data. Through anonymization, encryption, and strict access controls, the system mitigates privacy concerns and builds trust among users, fostering widespread adoption and usage in sensitive applications such as healthcare or finance.
- Real-time Performance Optimization: Optimized for real-time performance, the proposed ASR system delivers low latency and high throughput capabilities, ensuring seamless operation in time-sensitive applications such as live transcription or interactive voice interfaces

3.5 System Architecture

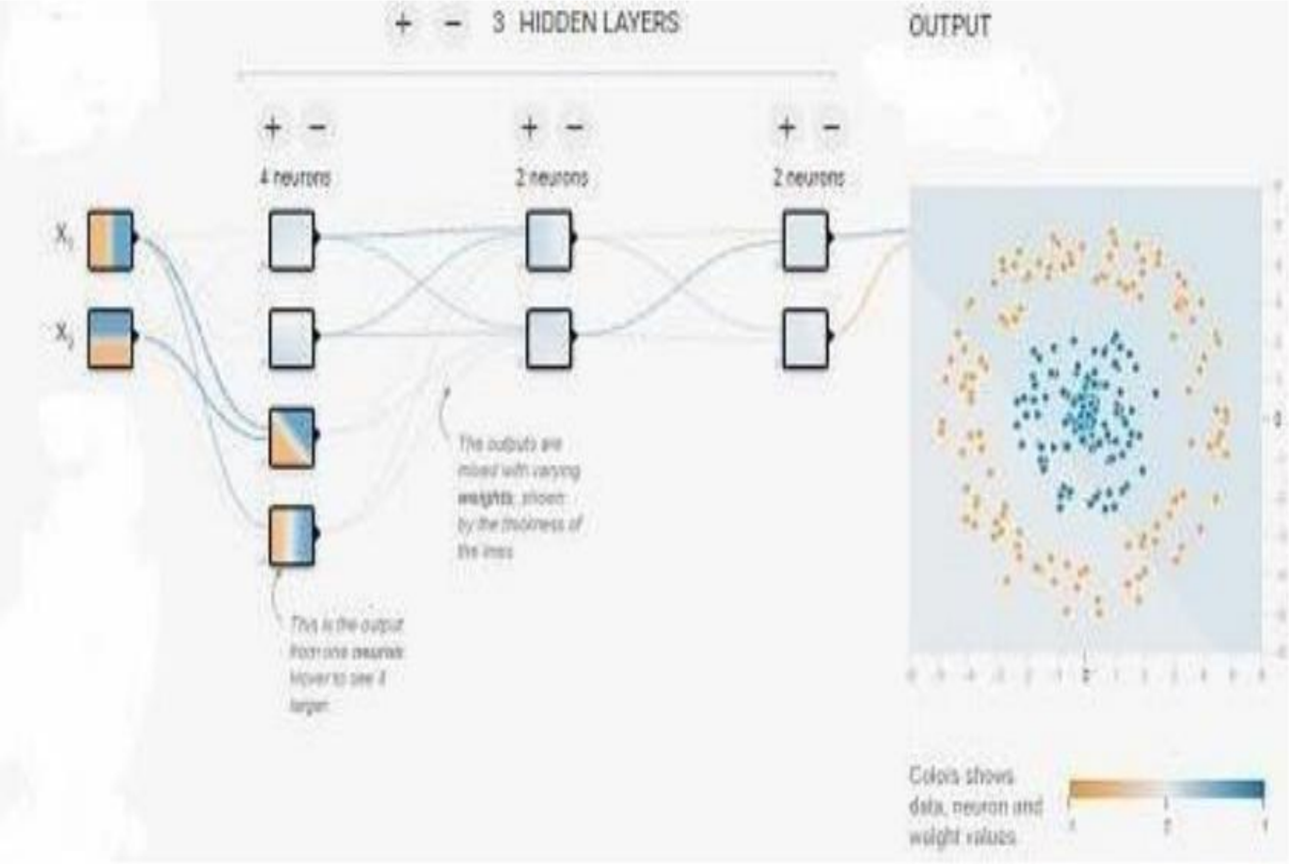

+ − 3 HIDDEN LAYERS
OUTPUT
+ −
4 neurons
+ −
2 neurons
+ −
2 neurons
X_1
X_2
The outputs are mixed with varying weights, shown by the thickness of the lines
This is the output from one neuron. Hover to see it larger.
Colors shows data, neuron and weight values

Figure 3.5: Neuron Architecture

3.6 Use Case Diagram

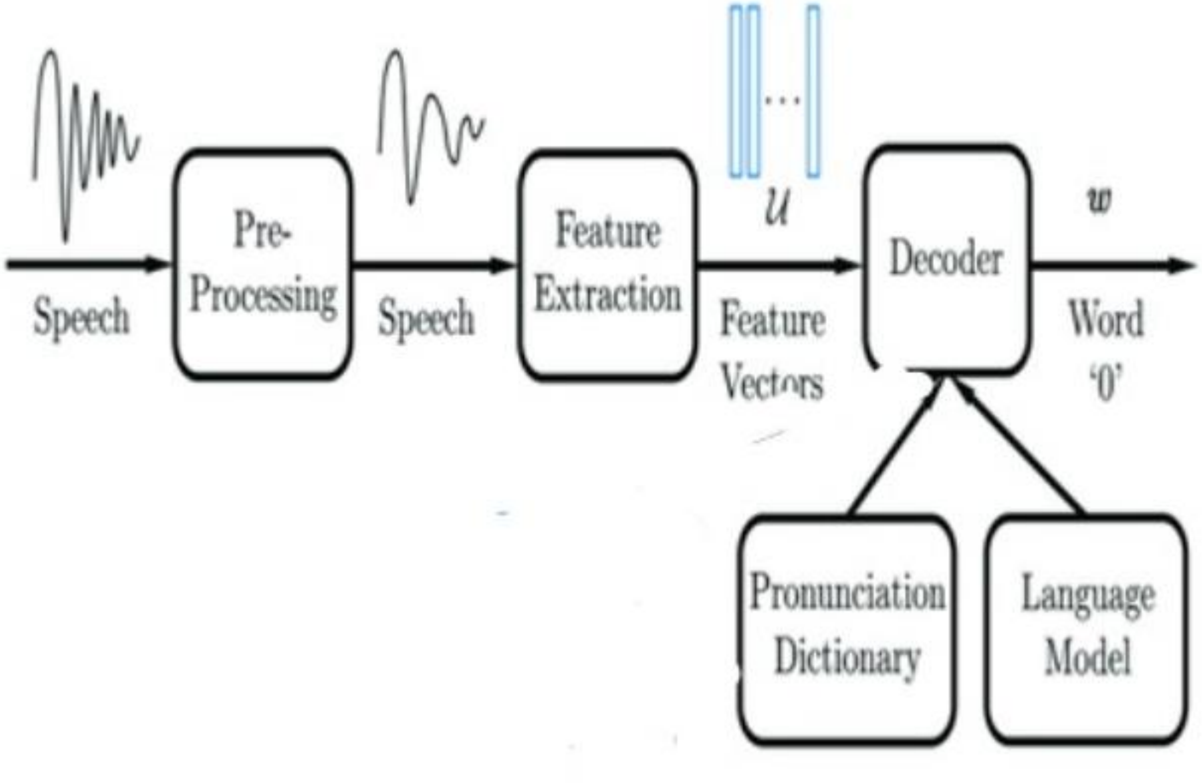


Fig : USE CASE DIAGRAM

CHAPTER 4

OBJECTIVES & EXPECTED OUTCOME

4.1 Objective

- Fast High Accuracy
- Robustness to Noise and Variations
- Real-time Processing
- Speaker Independence
- Language Support
- Computational Efficiency

4.2 Scope

The Scope of this project is to give a better, more accurate model which tries to overcome the limitations of Existing Systems. The following are the methods used for the same:

- **Secure Protocol Transmission:**

Through the use of encrypted protocols, we minimize data security breaches and ensure data safety. In this project, we use Secure WebSockets, a secure and encrypted protocol to forward packets to multiple peers. We minimize data security breaches and ensure data safety by using encrypted protocols. In this project, we forward packet to multiple peers using Secure WebSockets, an encrypted and secure protocol.

- **Image-to-Speech Conversion for Accessibility (Enhanced):**

This technology goes beyond simply describing objects. It can provide contextually relevant descriptions, including actions, emotions, and relationships within the image. Imagine an AI assistant for the visually impaired describing a busy street scene, not just listing objects but also conveying the flow of traffic, pedestrian activity, and potential hazards. Accurately capturing the nuances of human interaction and emotions depicted in images requires advanced machine learning models trained on diverse datasets.

- **Multimodal Learning for Enhanced Recognition (Deeper Dive):**

The model analyzes both the visual and audio information simultaneously. This allows it to leverage visual cues to disambiguate words with similar sounds, especially in noisy environments. Imagine a conversation happening in a crowded restaurant. The model can use visual cues like lip movements and speaker positions to identify individual voices and transcribe their speech accurately. This approach can significantly improve speech recognition accuracy in challenging scenarios, leading to more robust and reliable applications.

- **Automated Caption Generation for Videos (Further Exploration):**

This technology can generate captions that not only transcribe spoken words but also include speaker identification, emotional tone, and contextual information. Imagine educational videos where captions highlight key points, identify speakers in discussions, and translate content into multiple languages. This technology can revolutionize accessibility for multimedia content, making it more inclusive and engaging for diverse audiences.

- **Content Creation and Storytelling (Creative Frontiers):**

AI models can analyze the visual content and generate text descriptions that go beyond simple object recognition. They can create poems, scripts, or narratives inspired by the image. Imagine an AI tool that analyzes a picture of a bustling marketplace and generates a fictional story based on the characters, interactions, and overall atmosphere. This technology has the potential to unlock new avenues for creative expression and storytelling, blurring the lines between human and machine-generated content.

4.3 Methodology

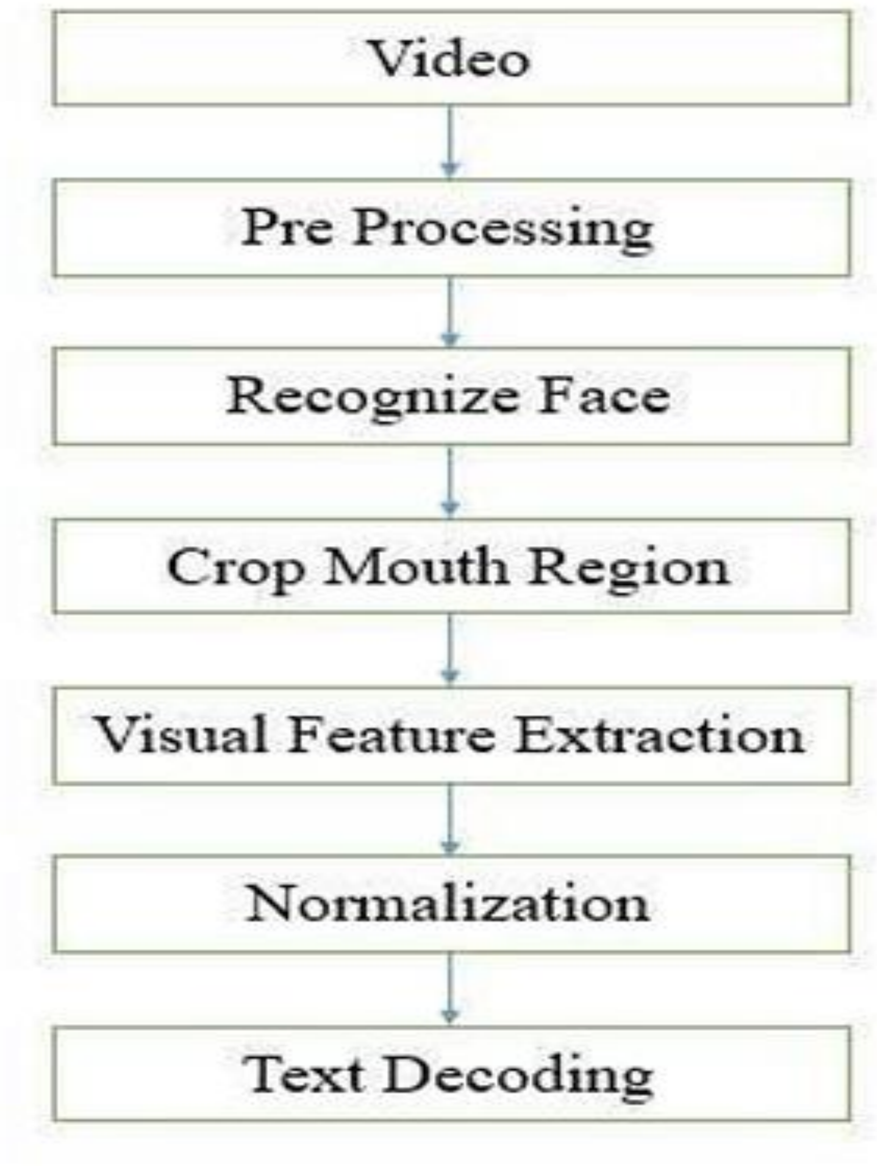


Figure 4 : Methodology

4.3.1 Streamlit

Streamlit streamlines the creation of interactive data apps directly within the Python environment. It allows developers to rapidly prototype and deploy data dashboards without needing separate frontend languages like HTML, CSS, or JavaScript. With Streamlit, you can write your entire app in Python, utilizing built-in widgets like charts, tables, and text inputs for user interaction and data visualization. Customization options and themes are available, and you can even integrate custom CSS for advanced styling. Sharing your apps is simple through a link, making them accessible from any web browser. Streamlit offers deployment flexibility, allowing you to share them locally, embed them within existing applications, or deploy them to the cloud. As an open-source and community-driven project, Streamlit fosters collaboration and provides support for developers. This focus on Python-centric development empowers data scientists and machine learning engineers to concentrate on core data analysis and logic, leaving the complexities of frontend development behind.

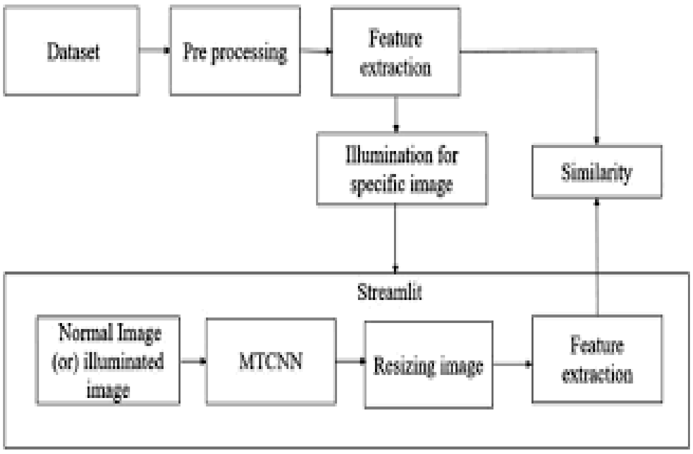


Figure 4.3.1: Streamlit Architecture

4.3.2 TensorFlow

TensorFlow stands out as a powerful open-source library for numerical computation and machine learning tasks. Its core strength lies in efficiently handling large datasets through tensors, multidimensional arrays. This makes it ideal for building and training machine learning models, particularly deep neural networks. TensorFlow provides a flexible development environment, supporting various programming languages like Python, Java, and C++, offering adaptability to different project needs. Additionally, it runs seamlessly on diverse platforms, including CPUs, GPUs, and TPUs, enabling efficient hardware utilization for faster computations. Beyond its technical prowess, TensorFlow boasts a large and active community, offering extensive resources, tutorials, and support. Furthermore, it comes equipped with visualization tools like TensorBoard, allowing users to effectively analyze and understand model training and performance. These features, combined with its wide applicability in image recognition, natural language processing, and scientific computing, solidify TensorFlow as a leading choice for various machine learning endeavors.

4.3.3 Python

Python stands out as a high-level, general-purpose programming language renowned for its readability and beginner-friendly approach. Its clear and concise syntax, emphasizing indentation over curly braces, makes it intuitive to learn and write. Python boasts a rich standard library, offering modules for a diverse range of tasks, from data analysis and web development to scientific computing. Its object-oriented capabilities enable the creation of complex and modular applications. This, coupled with its extensive community offering vast resources, libraries, and frameworks, makes Python a versatile tool for various projects. Its cross-platform compatibility ensures seamless operation across different operating systems. Python's popularity stems from its wide applicability in web development, data science, machine learning, automation, and more, making it a favorite among both beginners and seasoned programmers.Python boasts several strengths that solidify its position as a versatile and powerful tool. Its extensive standard library and vast ecosystem of third-party libraries empower developers to tackle diverse tasks, from web development and data analysis to automation and scientific computing. The open-source nature fosters a vibrant community, continuously contributing libraries, frameworks, and resources, ensuring ongoing support and development. Additionally, Python's interpreted nature allows for faster development cycles and efficient debugging. This cross-platform compatibility makes it a portable choice, seamlessly running on various operating systems. Frameworks like Django and Flask power some of the largest web applications, while libraries like NumPy, Pandas, and Scikit-learn make Python a go-to choice for data science and machine learning. Python's strengths extend to scientific computing, automation, and even education, making it a valuable tool across various fields and a favorite among both beginners and experienced programmers.

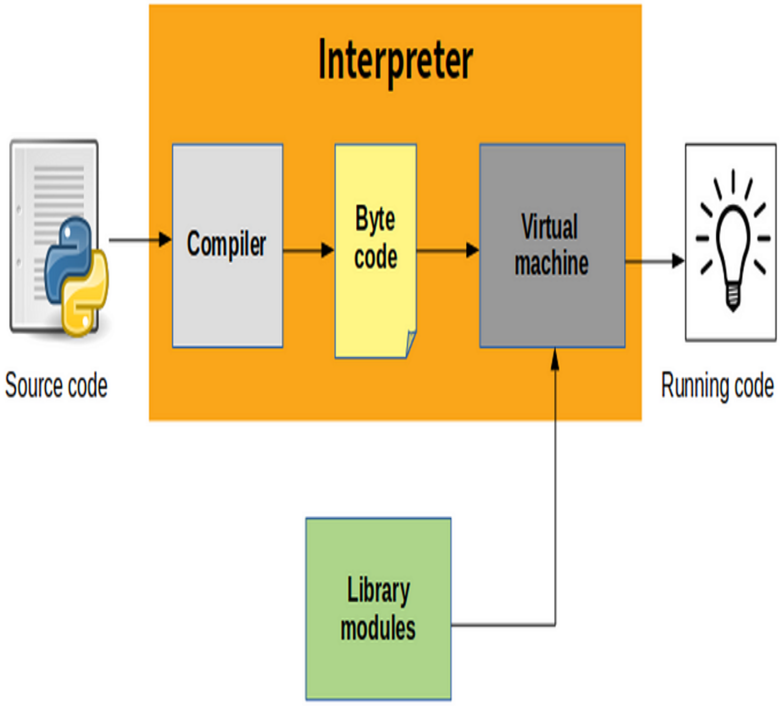


Figure 4.3.2: Python Architecture

4.4 Expected Outcome

The implementation of the work was carried out.

- **Accurate Lip Detection:**

The system should reliably identify and track the lips within a video frame, even in challenging situations like varying lighting, head movements, or partial occlusions.

- **Lip Movement Recognition:**

The system should accurately recognize the specific movements of the lips associated with different phonemes (sounds) in spoken language. This involves distinguishing subtle changes in lip shapes and movements.

- **Word Segmentation:**

The system should be able to segment the lip movements into individual words based on pauses, changes in lip patterns, or other visual cues.

- **Text Generation:**

Based on the recognized lip movements and word segmentation, the system should generate the corresponding text in a clear and accurate format. This may involve handling variations in pronunciation and accents.

- **Real-time Performance**:

Ideally, the system should strive for real-time or near real-time processing, minimizing the delay between lip movements and the generated text output. This is crucial for practical applications like speech-to-text translation for hearing-impaired individuals.

CHAPTER 5

ABOUT THE ALGORITHM

Preprocessing:

- The system begins by receiving a video as input.
- Face detection algorithms locate the face within each frame, narrowing the focus to the relevant area.
- Specific lip region extraction techniques, like landmark detection or facial feature extraction, isolate the lips from the face.
- Finally, normalization ensures consistent processing by standardizing the extracted lip region in terms of size and intensity.

Feature Extraction with CNN:

- Each normalized lip region frame is fed into a pre-trained CNN model.
- This powerful network extracts high-level features from the lip image, capturing crucial information about lip shapes, movements, and textures.
- These features essentially represent the visual characteristics associated with the spoken phonemes.

Greedy Decoding with RNN:

- The extracted features from each frame are sequentially fed into an RNN, such as LSTM.
- At each time step (frame), the RNN employs a greedy approach to predict the most likely phoneme based on the current frame's features and the context provided by previously predicted phonemes.
- This temporal context is crucial, as the RNN's internal state retains information from past frames, allowing it to capture the dynamic evolution of lip movements and the sequential nature of spoken language.
- As each phoneme is predicted, it's progressively added to a text sequence, ultimately forming the final text output.

Step-by-Step Greedy Decoding:

- The RNN starts with an empty text sequence and an initial internal state.
- It processes the first frame's features, predicting the most likely phoneme based on that information.
- This predicted phoneme is appended to the text sequence, and the RNN's internal state is updated to incorporate this new information.
- This process repeats for each subsequent frame, with the RNN continuously predicting the next most likely phoneme based on the current frame and the evolving context.
- Once all frames are processed, the final text sequence containing the predicted phonemes is generated.
- While the greedy algorithm offers advantages like simplicity and computational efficiency, it also has limitations. Errors in earlier predictions can propagate through the sequence, potentially leading to inaccurate text output. Additionally, the greedy approach might not always find the most accurate sequence of phonemes, especially in cases with ambiguity or complex lip movements.

CNN Feature Extraction:

- The CNN takes a normalized lip region frame as input, represented as a matrix X of size (h x w x 3), where h and w are the height and width of the image, and 3 represents the RGB color channels.
- The CNN processes X through multiple convolutional layers, each applying a filter F of size (k x k x 3) and producing a feature map. The output of the ith convolutional layer is calculated as:
- Y_i = f(W_i * X + b_i) where W_i is the weight matrix of the ith layer, b_i is the bias vector, and f is a non-linear activation function like ReLU.
- This process extracts high-level features from the lip image, capturing information about its shape, texture, and movement. The resulting feature vector, denoted as Z, summarizes these characteristics.

RNN Decoding:

- The RNN, typically an LSTM, receives the extracted feature vector Z from each frame as input.
- At each time step (t), the RNN's internal state, denoted as h_t, and cell state, denoted as c_t, are updated based on the current feature vector Z_t and the previous state information:
- h_t = f_h(W_hh_t-1 + W_xz_t + b_h)
- c_t = f_c(W_ch_t-1 + W_cz_t + b_c)
- These equations involve weight matrices (W_h, W_x, W_c) and bias vectors (b_h, b_c) specific to the RNN architecture.
- The activation functions f_h and f_c control the flow of information within the RNN.
- The RNN then predicts the most likely phoneme p_t based on the current state and feature vector:
- p_t = softmax(W_p * h_t + b_p)where W_p is the weight matrix connecting the hidden state to the output layer, and b_p is the bias vector.
- The softmax function normalizes the output into probabilities for each phoneme.

Greedy Choice:

- The greedy algorithm selects the phoneme with the highest probability (p_t) as the predicted phoneme at that time step.
- This phoneme is appended to the text sequence, and the RNN's internal state is updated with this information, influencing the prediction for the next frame.
- This process iterates for each frame in the video, with the RNN continuously updating its state and predicting the next most likely phoneme based on the greedy approach. Ultimately, the generated sequence of phonemes forms the final text output.
- While the greedy algorithm offers a computationally efficient solution, it may not always find the globally optimal sequence due to potential error propagation and suboptimal choices at each step.

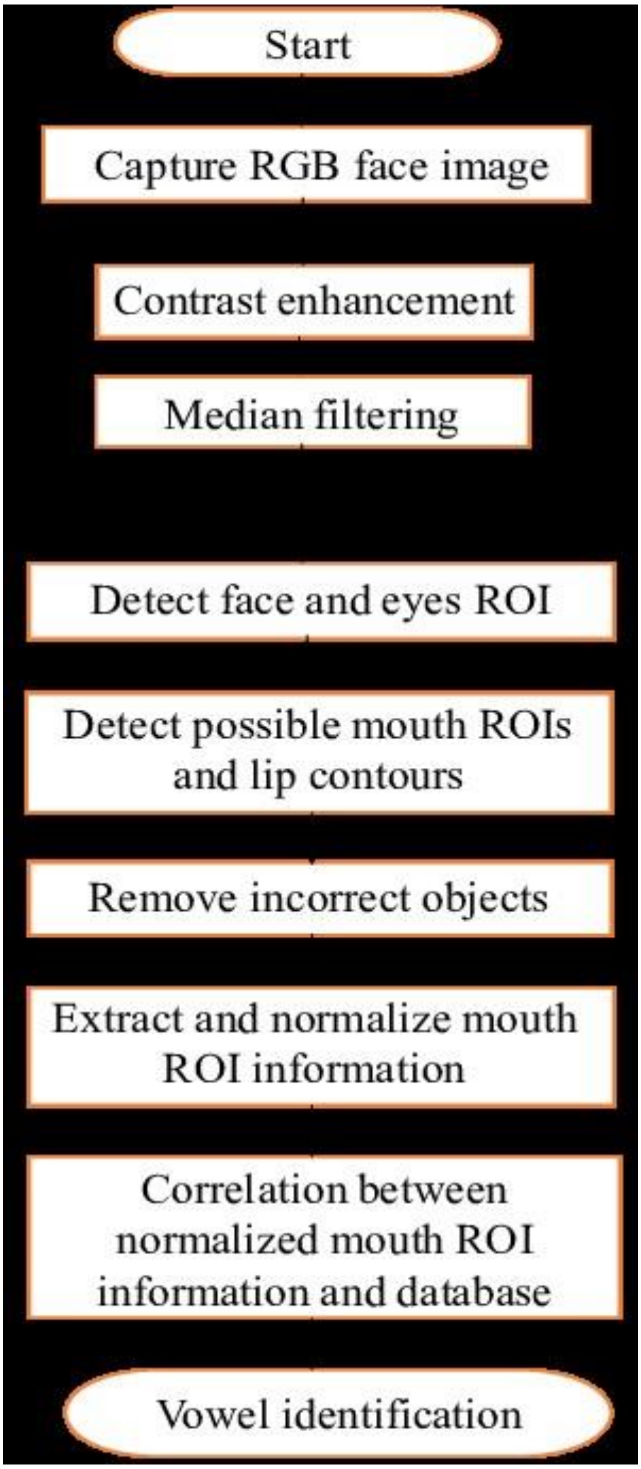

Start
Capture RGB face image
Contrast enhancement
Median filtering
Detect face and eyes ROI
Detect possible mouth ROIs and lip contours
Remove incorrect objects
Extract and normalize mouth ROI information
Correlation between normalized mouth ROI information and database
Vowel identification

Figure 5.1: Image Recognition process

Mathematical Underpinnings:

- Convolutional layers within the CNN apply filters (F) of size (k x k) to the input image (X) to extract features.
- The output (Y) at each layer is calculated using mathematical equations like:
- Y_i = f(W_i * X + b_i)
- where W_i is the weight matrix of the ith layer, b_i is the bias vector, and f is a non-linear activation function like ReLU.
- RNNs, particularly LSTMs, rely on equations like:
- h_t = f_h(W_hh_t-1 + W_xz_t + b_h)
- c_t = f_c(W_ch_t-1 + W_cz_t + b_c)where W_h, W_x, W_c are weight matrices, b_h, b_c are bias vectors, and f_h, f_c are activation functions controlling information flow within the network.Phoneme prediction probabilities (p_t) are calculated using the softmax function:
- p_t = softmax(W_p * h_t + b_p)where W_p is the weight matrix connecting the hidden state to the output layer, and b_p is the bias vector.
- This intricate interplay of machine learning algorithms and mathematical operations allows the system to analyze lip movements in video frames, recognize the corresponding phonemes, and ultimately generate the spoken text, bridging the gap between visual information and language understanding.

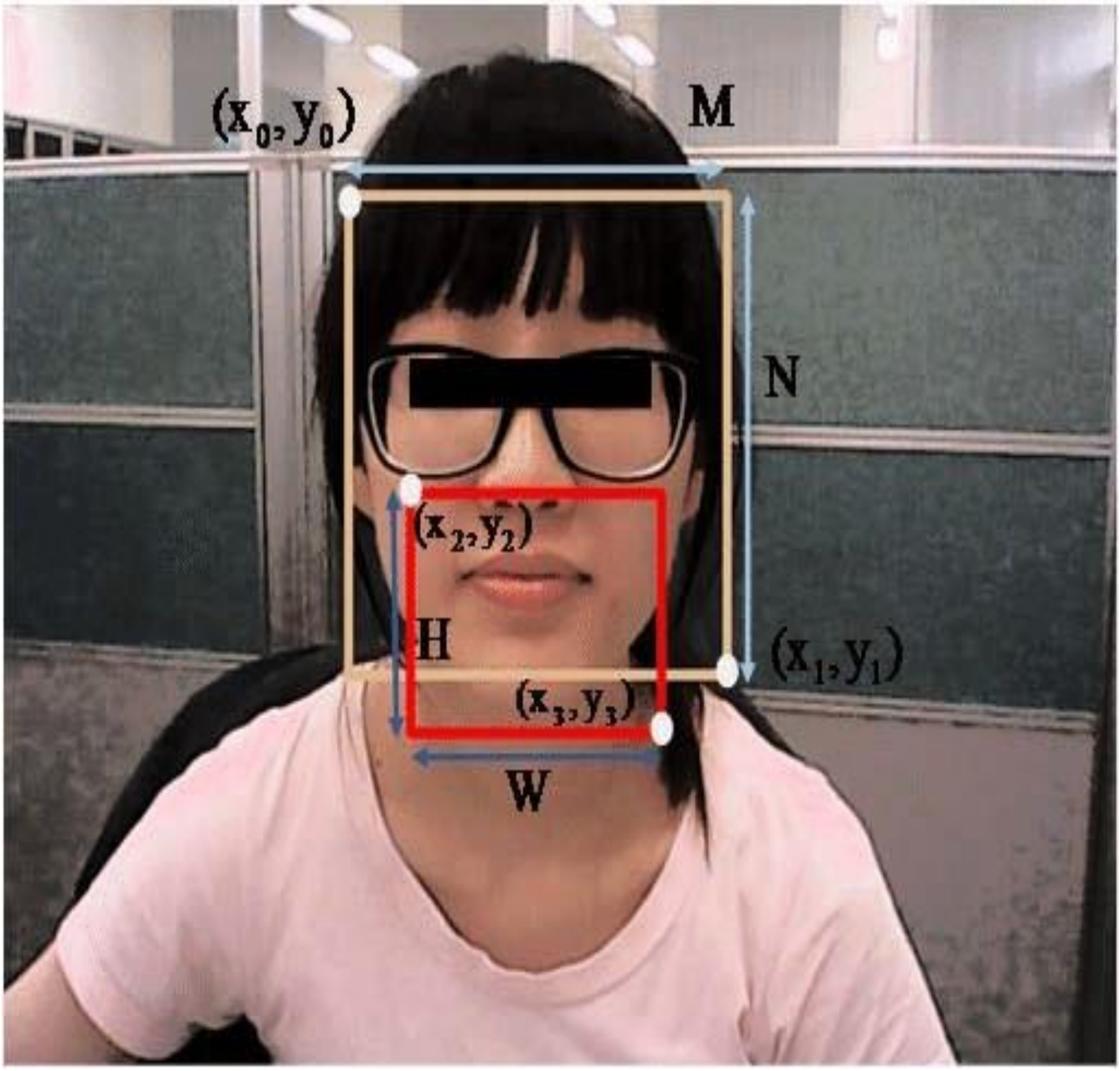


Figure 5.2: Lip Recognition

CHAPTER 6

RESOURCE REQUIREMENTS

6.1 Hardware Requirements

Processors	:	AMD Ryzen 3600X
RAM	:	8 GB (min)
Storage	:	128 GB
Standard Devices	:	Keyboard, Monitor and Mouse

6.2 Software Requirements

For developing the application, the following are the Software Requirements:

- Operating Systems supported: Windows 11, 64-bit operating system
- Language :PYTHON
- Streamlit
- Tensorflow

6.3 Functional Requirements

Lip reading systems aim to interpret speech visually by analyzing lip, face, and tongue movements. They have the potential to revolutionize communication for individuals with hearing impairments or in noisy environments. To achieve effective lip reading, systems must meet several functional requirements:

Accurate and real-time lip detection and tracking, relevant visual feature extraction, robust viseme/phoneme classification, word recognition and speech output, and real-time performance, user-friendliness, robustness, and privacy.

6.3 Non Functional Requirements

Alongside functional requirements, lip reading systems must adhere to various non-functional requirements for effective operation and widespread adoption. These requirements encompass aspects like performance, scalability, usability, security, and reliability. Lip reading systems should operate in real-time, handling lip movements and generating speech output without noticeable delays. The systems should be able to handle increasing usage and data volumes without compromising performance or accuracy. They should be easy to use and learn, with intuitive interfaces and minimal training requirements for both speakers and users. Robust security measures should be employed to protect user privacy and ensure that lip data is handled securely. The systems should operate consistently and reliably, minimizing errors and downtime to maintain user confidence. Addressing these non-functional requirements can enhance communication accessibility and broaden the acceptance of lip reading systems.

CHAPTER 7

TESTING

Testing in a project refers to the crucial process of evaluating a system, software, or product to ensure it meets specified requirements and functions correctly. It involves verifying and validating the product against defined standards, identifying defects or errors, and mitigating risks. Testing activities encompass creating test plans, designing test cases, executing tests, analysing results, and reporting issues. By conducting various types of testing throughout the project lifecycle, such as unit testing, integration testing, and acceptance testing, developers can ensure the product's quality, reliability, and performance. Ultimately, testing plays a vital role in delivering a high-quality, user-friendly, and dependable product to customers.

7.1 Unit Testing

Unit testing plays a crucial role in ensuring the reliability and functionality of the Automatic Speech Recognition System. Unit testing constitutes the foundation of ensuring each individual component of your lip reading system functions correctly in isolation. In this context, it involves meticulously examining the functionality of discrete modules such as face detection, cropping, feature extraction, and CNN prediction. For instance, unit tests for the face detection module would involve feeding it various images with and without faces to ascertain its ability to accurately identify facial features. Similarly, tests for the cropping module would verify its capability to precisely isolate the mouth region from the detected face. Moreover, unit tests for the CNN model would scrutinize its performance in extracting relevant visual features and predicting spoken words accurately. These tests encompass a spectrum of scenarios, including edge cases and unexpected inputs, ensuring the robustness and reliability of each module before integration.

7.2 Integration Testing

Integration testing focuses on assessing how effectively the various modules of your lip reading system collaborate and function together as a cohesive unit. This entails examining the interactions and data flow between modules to ensure seamless integration and interoperability. For instance, integration tests would validate that the output from the face detection module serves as the input for the cropping module without loss or corruption of information. Similarly, they would confirm that the cropped mouth regions are accurately fed into the CNN model for feature extraction and word prediction. These tests encompass scenarios simulating real-world usage, evaluating the system's behavior under different conditions such as varying video qualities, lighting conditions, and speaking styles. By scrutinizing the system's performance in an integrated environment, integration testing aims to identify and rectify any inconsistencies or compatibility issues between modules

7.3 Validation Testing

Validation testing assesses whether the system meets the requirements and expectations of the end-users. In the case of your lip reading system, validation testing would involve evaluating its performance against real-world scenarios and data sets. This could include testing the system with various videos containing different speakers, accents, lighting conditions, and speaking speeds to validate its accuracy and reliability in practical usage. Validation tests would focus on measuring the system's ability to correctly recognize spoken words from lip movements and produce accurate textual transcriptions. Additionally, user feedback and usability testing could be conducted to ensure that the system meets the needs and expectations of its intended users, such as individuals with hearing disabilities or security personnel.

CHAPTER 8

PROJECT OUTCOMES

This paper implemented using this Streamlit for front end application and provides a user friendly interface for loading videos from the testing dataset

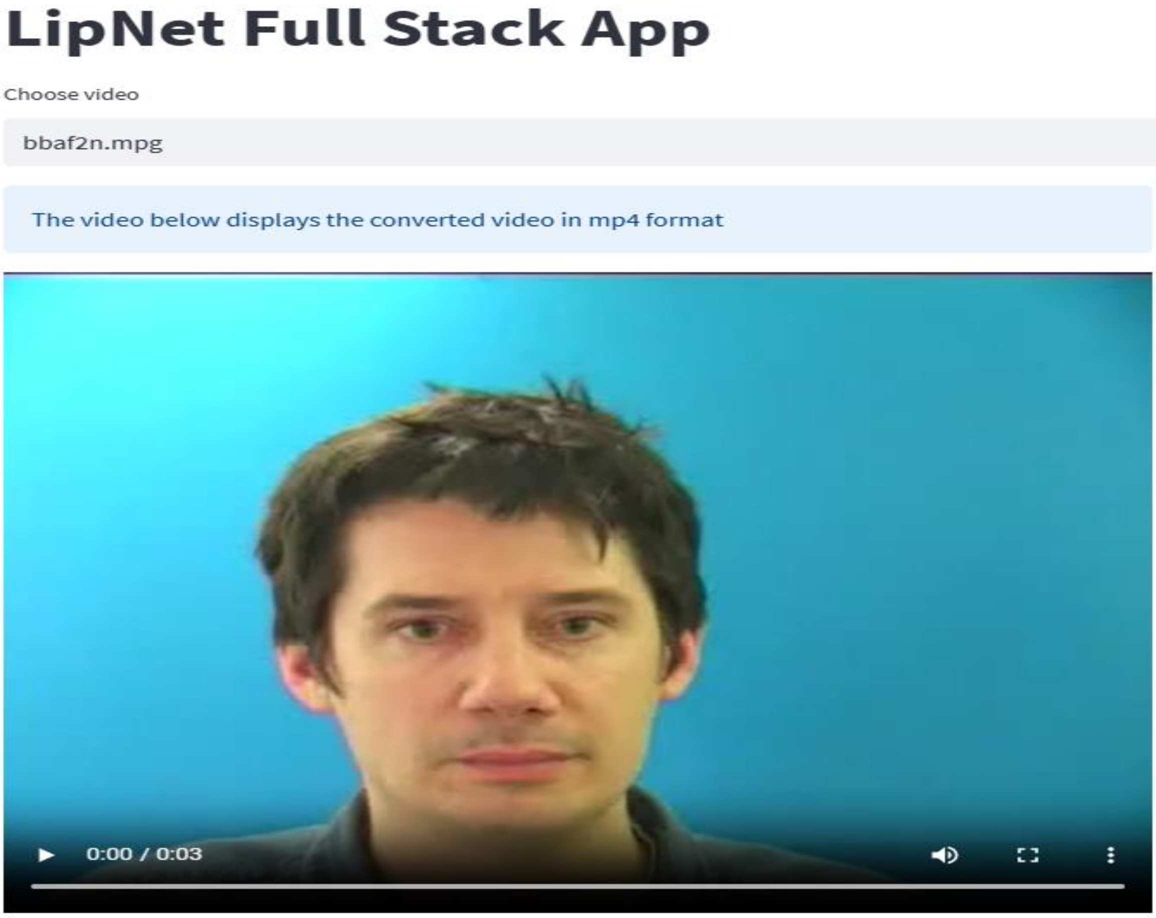
Web Application Implementation

Imagine a clean and user-friendly webpage (like) with a prominent "Upload Video" button in the center. Below the button, there could be a text area displaying instructions like "Upload a video in mpg format, maximum size 50MB".

LipBuddy

This application is originally developed from the LipNet deep learning model.

Figure 8.1: Application Name



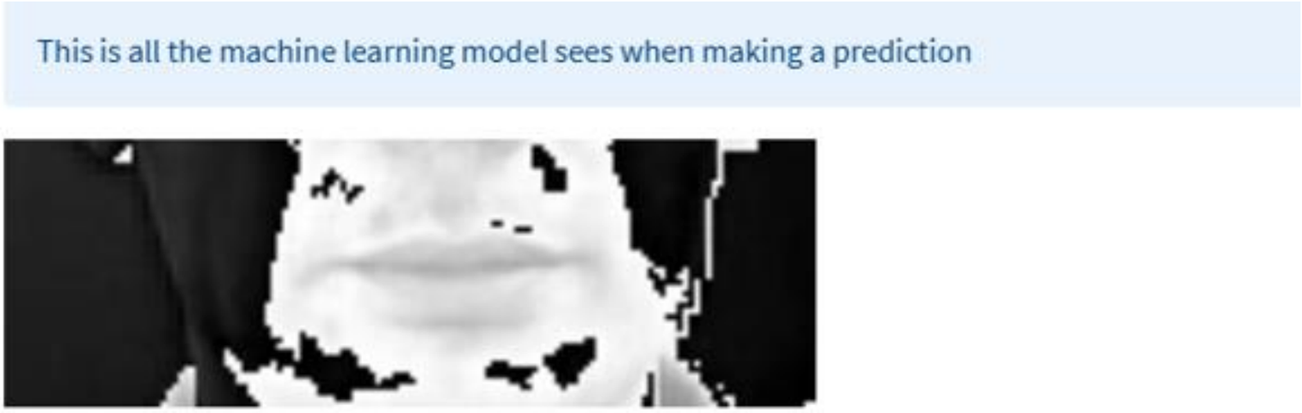
LipNet Full Stack App

Choose video

bbaf2n.mpg

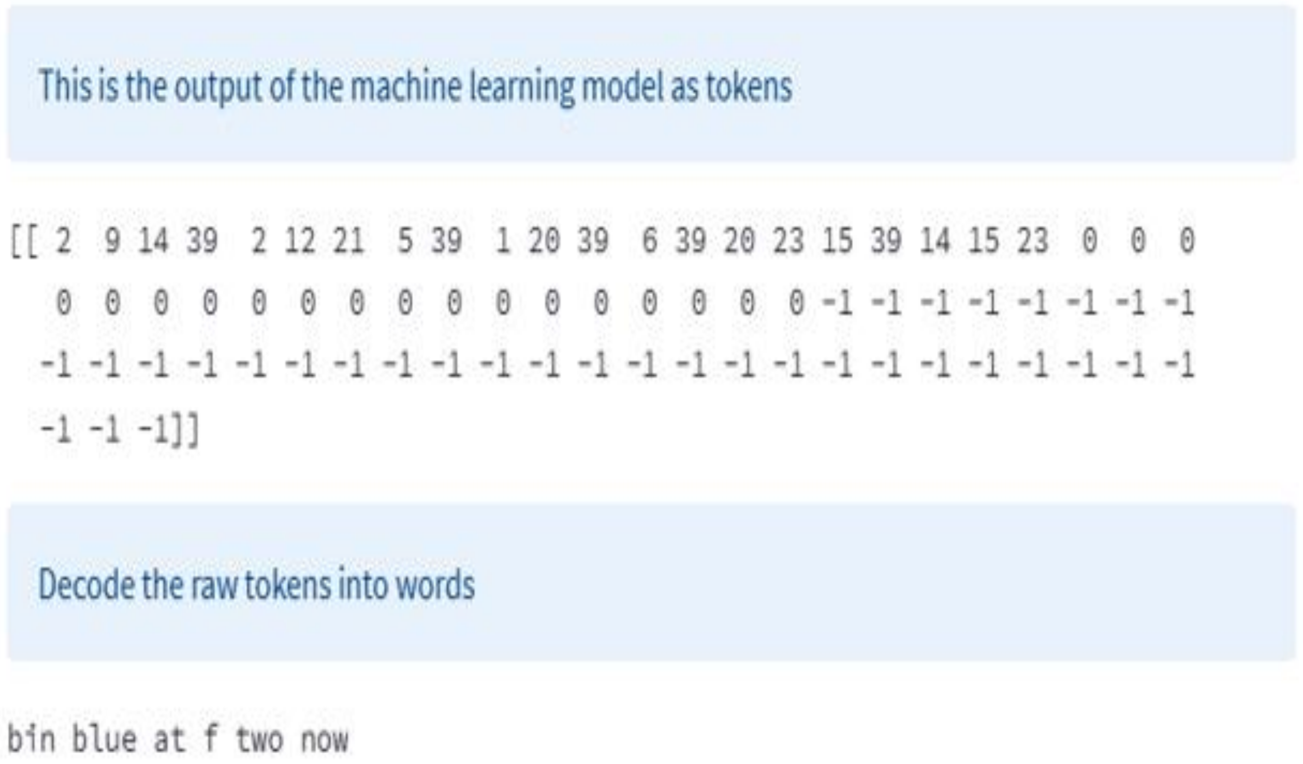
The video below displays the converted video in mp4 format

Figure 8.2: Video uploading



This is all the machine learning model sees when making a prediction

Figure 8.3: Gif image of the cropped mouth region



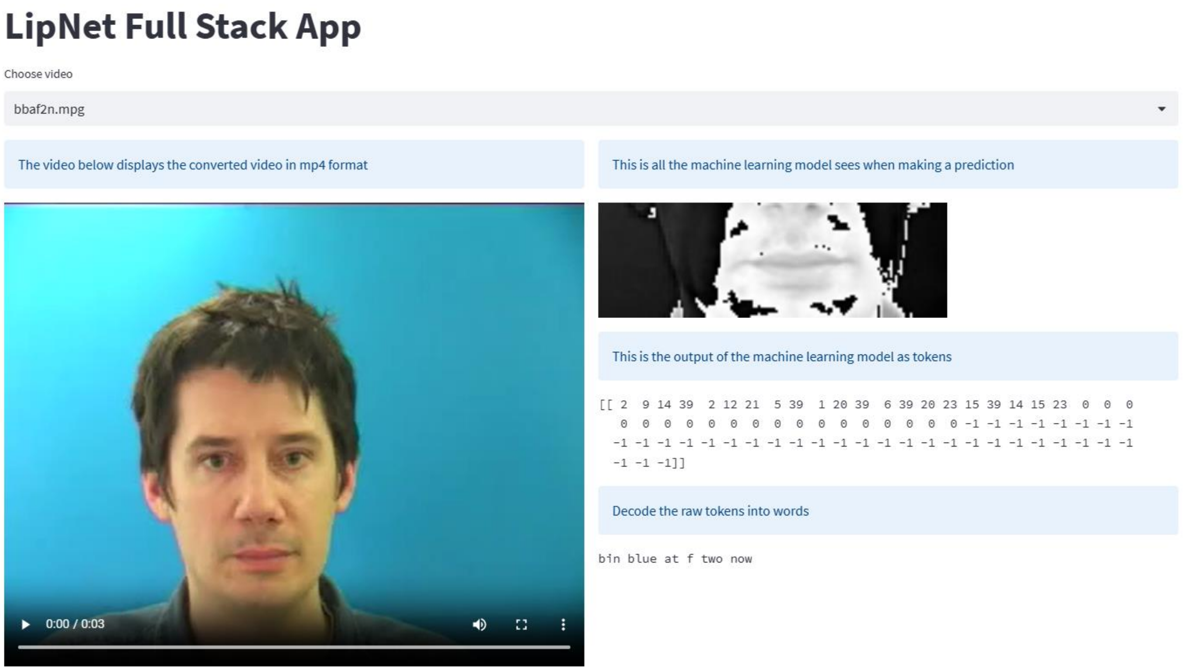
This is the output of the machine learning model as tokens

```
[[ 2  9 14 39  2 12 21  5 39  1 20 39  6 39 20 23 15 39 14 15 23  0  0  0
   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 -1 -1 -1 -1 -1 -1 -1 -1
  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
  -1 -1 -1]]
```

Decode the raw tokens into words

```
bin blue at f two now
```

Figure 8.4: Output generated in letters and numerical values



LipNet Full Stack App

Choose video

bbaf2n.mpg

The video below displays the converted video in mp4 format

This is all the machine learning model sees when making a prediction

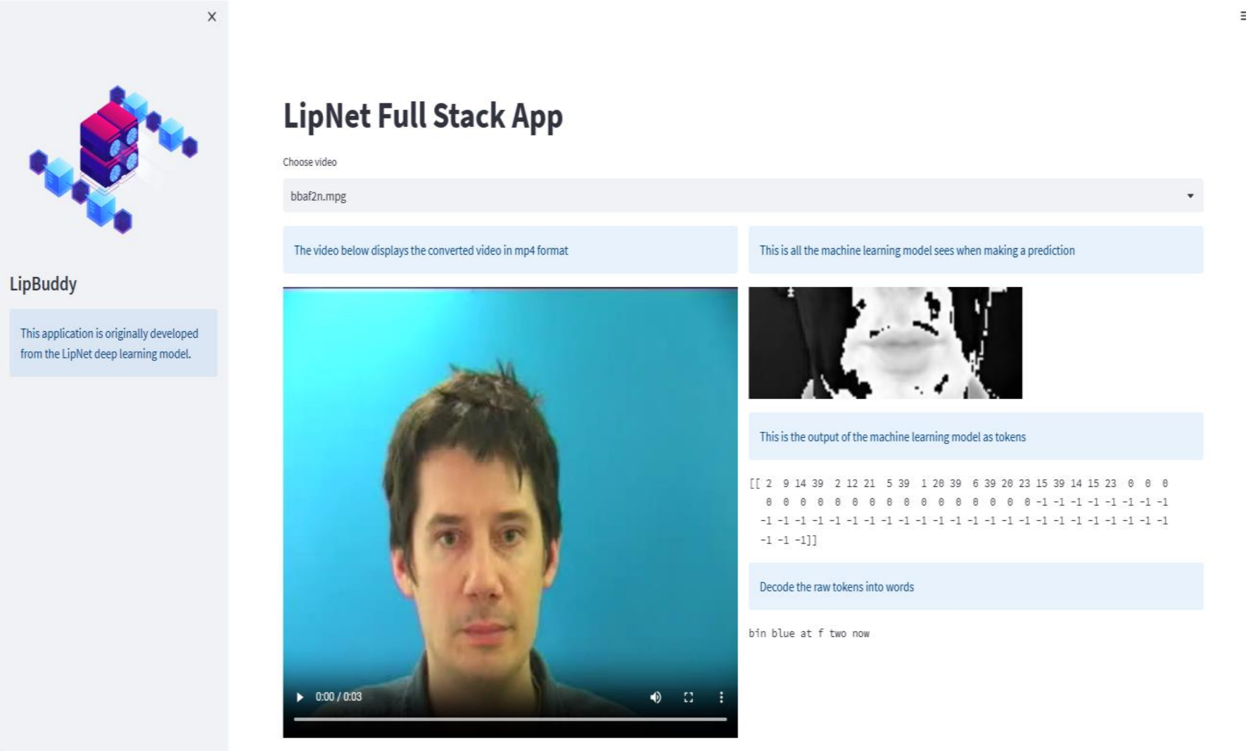
This is the output of the machine learning model as tokens

```
[[ 2  9 14 39  2 12 21  5 39  1 20 39  6 39 20 23 15 39 14 15 23  0  0  0
   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 -1 -1 -1 -1 -1 -1 -1 -1
  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
  -1 -1 -1]]
```

Decode the raw tokens into words

```
bin blue at f two now
```

Fig 8.5 : video upload and output generation



LipBuddy

This application is originally developed from the LipNet deep learning model.

LipNet Full Stack App

Choose video

bbaf2n.mpg

The video below displays the converted video in mp4 format

This is all the machine learning model sees when making a prediction

This is the output of the machine learning model as tokens

```
[[ 2  9 14 39  2 12 21  5 39  1 20 39  6 39 20 23 15 39 14 15 23  0  0  0
   0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0 -1 -1 -1 -1 -1 -1 -1 -1
  -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1 -1
  -1 -1 -1]]
```

Decode the raw tokens into words

```
bin blue at f two now
```

Fig 8.6 : FRONT END APPLICATION

- Once the user uploads a video, it's sent to the server for processing. The server-side script would likely use Python with libraries like OpenCV for video processing.A background process would handle video pre-processing steps: Splitting the video into individual frames (imagine a series of still images extracted from the video Converting frames from RGB to grayscale format (grayscale removes color information, reducing processing

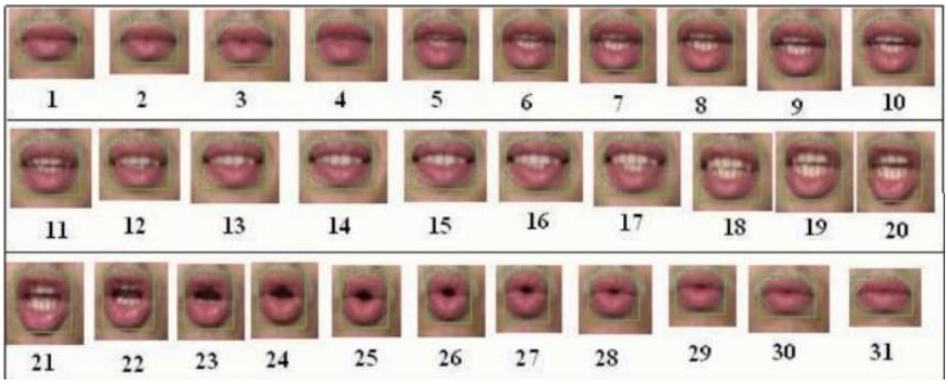


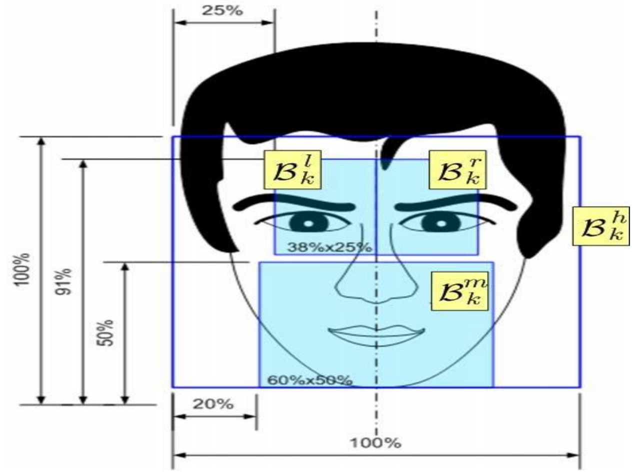
Fig:8.7: Cropped mouth images

Fig:8.8 Cropped mouth region for a word being pronounced

Lip Detection and Normalization:

- Each grayscale frame undergoes lip detection using pre-trained models like Haar cascades or facial landmark detectors.
- Imagine a frame with a bounding box drawn around the detected mouth region .
- The detected mouth region is cropped and resized to a standard size for consistency.

Normalization techniques might be applied to adjust brightness and contrast across frames.



Lip Reading with CNN:

- The pre-processed frames containing only the mouth region are fed into the trained CNN model.
- The CNN architecture, as described in the project, might have multiple convolutional layers for feature extraction and fully connected layers for classification.
- Each frame goes through the CNN, and the network outputs a probability distribution for each word in the vocabulary.

Text Decoding and Subtitle Generation:

- The CNN's output (probability distribution) is analyzed to identify the most likely word for each frame.
- A sequence of these predicted words is then assembled into sentences.
- Techniques like language models or beam search could be used to refine sentence construction and improve accuracy.
- The generated sentences become the subtitles for the video.

CONCLUSION

In conclusion, the development of our lip reading system represents a significant advancement in the field of visual speech recognition, with implications ranging from enhancing accessibility for individuals with hearing disabilities to improving security and forensics applications. Through the integration of cutting-edge machine learning techniques, including convolutional neural networks and deep learning algorithms, we have created a system capable of accurately transcribing spoken words from lip movements with a high degree of accuracy. Our project underscores the transformative potential of technology in bridging communication barriers and expanding the capabilities of human-computer interaction. Moving forward, we envision further research, refinement, and deployment of our system across diverse domains, with the ultimate goal of fostering inclusivity, efficiency, and innovation in our society. We are optimistic that the outcomes of this project will not only advance academic knowledge but also yield tangible benefits for individuals, industries, and Communities worldwide.

REFERENCES

[1] Assael, Y.M., Shillingford, B., Whiteson, S., de Freitas, N. (2017). "Lipnet: Sentence-level lipreading." Under submission to ICLR 2017. arXiv:1611.01599.

[2] Almajai, S., Cox, R., Harvey, R., & Lan, Y. (2016). "Improved speaker independent lip-reading using speaker adaptive training and deep neural networks." In IEEE International Conference on Acoustics, Speech and Signal Processing

[3] Michael Wand and Jurgen Schmidhuber. (2017). "Improving Speaker-Independent Lipreading with Domain Adversarial Training." The Swiss AI Lab IDSIA, USI & SUPSI, Manno-Lugano, Switzerland. arXiv:1708.01565v1

[4] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., et al. (2016). "TensorFlow: Large-scale machine learning on heterogeneous distributed systems."

[5] Chung, J. S., & Zisserman, A. (2016). "Lip Reading in the Wild." In Asian Conference on Computer Vision

[6] Viola, P., & Jones, M. (2001). "Rapid object detection using a boosted cascade of simple features." In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, volume 1

[7] Brendan Shillingford, Yannis Assael, Matthew W. Hoffman, Thomas Paine, Cían Hughes, Utsav Prabhu, Hank Liao, Hasim Sak, Kanishka Rao, Lorrayne Bennett, Marie Mulville, Ben Coppin, Ben Laurie, Andrew Senior and Nando de Freitas. (Year). "LARGE-SCALE VISUAL SPEECH RECOGNITION." DeepMind & Google.

[8] Lele Chen, Zhiheng Li, Ross K. Maddox, Zhiyao Duan, & Chenliang Xu. (Year). "Lip Movements Generation at a Glance." Wuhan university and University of Rochester

[9] Joon Son Chung, Andrew Senior, Oriol Vinyals, & Andrew Zisserman. (Year). "Lip Reading Sentences in the Wild." Department of Engineering Science, University of Oxford 2Google DeepMind.

[10] Bradski, G.,& Kaehler, A.(2008). “learning opencv : Computer Vision with opencv Library”,o’Reilly media