# LIP READING - AN EFFICIENT CROSS AUDIO-VIDEO RECOGNITION USING 3D CONVOLUTIONAL NEURAL NETWORKS

Submitted in partial fulfillment of the requirements of

**University of Mumbai**

For the Degree of

## Bachelor of Engineering in CSE (AIM/IOT)

Submitted by

## MR. KRUTIK KHANDARE [ROLL NO.: 29] MISS PRATIKSHA KURKUTE [ROLL NO.: 32]

**MISS SONAL MHATRE [ROLL NO.: 40]**

Under the guidance of

## PROF. SARITA KHEDIKAR



**DEPARTMENT OF CSE (AIML/IOT)**

**SMT. INDIRA GANDHI COLLEGE OF ENGINEERING**

Ghansoli, Navi Mumbai - 400701

## Academic year : 2022-2023

**Project Report Approval for T.E.**

This project report entitled “**Lip Reading - An efficient Cross Audio-Video Recognition using 3D Convolutional Neural Networks”**

By

**Mr. Krutik Khandare** (Roll No.: 29) **Miss Pratiksha Kurkute** (Roll No.: 32) **Miss Sonal Mhatre** (Roll No.: 40)

are approved for the degree of Bachelor of Engineering in Computer Engineering, Semester VII, University of Mumbai.

Examiner 1 Examiner 2 Prof. Sarita Khedikar Internal Guide

Prof. Sonali.Deshpande Dr. Sunil Chavan

Head of Department Principal

Date: 30/10/2021

Place: Ghansoli, Navi Mumbai.

## Declaration

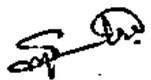
We declare that this written submission represents our own ideas in our own words and where others ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any act/data/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.



Mr. Krutik Khandare [Roll No.: 29]



Miss Pratiksha Kurkute [Roll No.: 32]



Miss Sonal Mhatre [Roll No.: 40]

Date: 30/10/2021

Place: Ghansoli, Navi Mumbai.

## Abstract

Title: Lip Reading - An efficient Cross Audio-Video Recognition using 3D Convolutional Neural Networks

Audio-video recognition (AVR) has been considered as a solution for speech recognition tasks when the audio is corrupted, as well as a visual recognition method used for speaker verification in multi speaker scenarios. The approach of AVR systems is to leverage the extracted information from one modality to improve the recognition ability of the other modality by complementing the missing information By using a relatively small network architecture and much smaller data set for training, our proposed method surpasses the performance of the existing similar methods for audio-visual matching, which use 3D CNNs for feature representation. We also demonstrate that an effective pair selection method can significantly increase the performance.

This system will be helpful for or can be used in applications related to improved hearing aids, Video conferencing in silent environments, High-quality speech recovery from background noise, Generating a voice for people who cannot produce voiced sounds (aphonia). It will be also useful in applications related to biometric authentication.

## List of Abbreviations

* B.E.: Bachelor of Engineering
* DFD: Data Flow Diagram
* CNN: Convolutional Neural Network
* 3D CNN: 3 Dimensional Convolutional Neural Network
* NLP: Natural Language Processing
* VS Code: Visual Studio Code
* GUI: Graphical User Interface

## List of Figures

* Fig.2.2.1 VS Code
* Fig.2.2.2 PyCharm
* Fig. 2.2.3 Natural Language Processing
* Fig.3.1.1 Agile Model
* Fig.4.5.1 Data Flow Diagram (Level 0)
* Fig.4.5.2 Data Flow Diagram (Level 1)
* Fig.4.5.3 Data Flow Diagram (Level 2)
* Fig.5.1 Flow Chart
* Fig.5.2.1 Use Case Diagram
* Fig.5.2.2 Sequence Diagram
* Fig. 5.2.3 Class Diagram
* Fig. 6.1.1 3D CNN
* Fig. 6.2.1 Home Page
* Fig. 6.2.2 Real Time Video Capture Screen

## List of Tables

* Table 3.1 Timeline Chart

**Index**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Chapter No.** | **Content** | | | **Page No.** |
| **I** | **Abstract** | | | i |
| **II** | **List of Abbreviations** | | | ii |
| **III** | **List of Figures & Tables** | | | iii |
| **1** | **Introduction** | | | 1 |
|  | 1.1 | Problem Statement | | 2 |
|  | 1.2 | Objectives | | 2 |
|  | 1.3 | Scope | | 2 |
|  | 1.4 | Report Organization | | 2 |
| **2** | **Review of Literature** | | | 4 |
|  | 2.1 | Research Paper Analysis, literature survey | | 5 |
|  | 2.2 | Methodology | | 6 |
|  |  | 2.2.1 | Visual Studio Code (VS Code) | 6 |
|  |  | 2.2.2 | PyCharm | 7 |
|  |  | 2.2.3 | Natural Language Processing (NLP) | 8 |
| **3** | **Planning and Formulation** | | | 9 |
|  | 3.1 | Project Development Model | | 10 |
|  |  | 3.1.1 | Project Model: (any process model) e.g Agile Model | 10 |
|  |  | 3.1.2 | Phases of Agile Model | 10 |
|  |  | 3.1.3 | Advantages of Agile Model | 11 |
|  |  | 3.1.4 | Disadvantages of Agile Model | 11 |
|  | 3.2 | Timeline Chart | | 11 |
|  | 3.3 | Feasibility Analysis | | 12 |
|  |  | 3.3.1 | Technical Feasibility | 12 |
|  |  | 3.3.2 | Economic Feasibility | 12 |
|  |  | 3.3.3 | Operational Feasibility | 12 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **4** | **Requirement Analysis** | | | 13 |
|  | 4.1 | Hardware Requirements | | 14 |
|  | 4.2 | Software Requirements | | 14 |
|  | 4.3 | Functional Requirements | | 14 |
|  | 4.4 | Non-functional Requirements | | 14 |
|  | 4.5 | Data Flow Diagrams | | 15 |
|  |  | 4.5.1 | DFD Level 0 | 15 |
|  |  | 4.5.2 | DFD Level 1 | 15 |
|  |  | 4.5.3 | DFD Level 2 | 16 |
| **5** | **System Design** | | | 17 |
|  | 5.1 | Flow Chart | | 18 |
|  | 5.2 | UML Diagrams | | 19 |
|  |  | 5.2.1 | Use Case Diagram | 19 |
|  |  | 5.2.2 | Sequence Diagram | 20 |
|  |  | 5.2.3 | Class Diagram | 21 |
|  |  | 5.2.4 | Activity Diagram | 22 |
| **6** | **Algorithm Development** | | |  |
|  | 6.1 | Algorithm Development | | 23 |
|  |  | 6.1.1 | 3D CNN | 23 |
|  | 6.2 | Model Development | | 24 |
| **7** | **Conclusion** | | | 27 |
|  | 7..1 | Conclusion | | 28 |
|  | 7..2 | Future Scope | | 28 |
|  | **Acknowledgement** | | | 29 |
|  | **References** | | | 30 |

# CHAPTER 1 INTRODUCTION

## Introduction

The most natural methodology of communication among people, in general, is “talking” which we call “speech.” But sadly, this natural kind of communication for the individuals those are dumb and hearing lessened cannot use. The strategy of word recognition offered to assist hearing lessened or dumb individuals talk to the others during a normal technique. It’s a visible methodology of talking within which solely lip movements are applied to established vocalized words. Visual speech recognition may be a technique that acknowledges the words by movement of the lip. Visual speech recognition is the process of reading the lip. The deaf person and the hearing-impaired person can easily recognize the speech by lip movements. Since earlier times, people have been apprehended that the movement of the lip has had some speech knowledge. In various application areas like speech in the changing region, spaces where you are not ought to be compelled to talk and catastrophe situations volcanic activity visual speech is significant.

### Problem Statement

Audio-video recognition (AVR) has been considered as a solution for speech recognition tasks when the audio is corrupted, as well as a visual recognition method used for speaker verification in multi speaker scenarios. The approach of AVR systems is to leverage the extracted information from one modality to improve the recognition ability of the other modality by complementing the missing information By using a relatively small network architecture and much smaller data set for training, our proposed method surpasses the performance of the existing similar methods for audio-visual matching, which use 3D CNNs for feature representation. We also demonstrate that an effective pair selection method can significantly increase the performance. The act or process of determining the intended meaning of the speaker by utilizing all visual clues accompanying speech attempts, such as lip movements, facial expressions, and bodily gestures, used especially by people with impaired hearing.

### Objectives

* + 1. Recognise phrases and sentences being spoken by a talking face, with or without the audio.
    2. Support multiple languages (English, Hindi).
    3. Real time interaction result
    4. Create a user-friendly interface that could help to have a better conversation in the absence of audio.
    5. Refining and retuning parameters for better efficiency using algorithms like Adaboost as compared to the existing system.

### Scope

* + 1. Speech to text conversion becomes a link between deaf and normal people. It will help the person with hearing disability by providing subtitles to any video available on the internet.
    2. The person who doesn't have earphones or cannot hear it properly, can use this application to get the Subtitles.
    3. It will let the user having hearing disability answer the call. The app may help them to know what the other end caller is saying by reading the subtitles which will get converted by the end caller’s speech.

### Report Organization

* **In Chapter 2,** we will see the literature survey which will tell us more about the background of the project including the work that has already been done in this field.
* **In Chapter 3,** Planning and Formulation of the project is given. Usage of Agile model and how we integrated and worked around the model.
* **In Chapter 4,** shines light upon the Requirements that are needed and analysis of the system to uncover the additional requirements of the project.
* **In Chapter 5,** the system proposed is introduced which will tell the deep specification of the project and will tell how the different modules of the system will work, the flow of the project regarding data flow, control flow and other flow of the system.
* **In Chapter 6,** we see the implementation of the algorithm of the project and process of model building.
* **In Chapter 7,** Conclusion and Future Scope of this project is mentioned.

# CHAPTER 2 REVIEW OF LITERATURE

## Review of Literature

### Research Paper Analysis

The basic theory of lipreading in 1954 was proposed by Sumby and Pollack [1], and it was rst proposed that the features of lip motion could be used to identify the speaker's speech content. In 1984, Petajan [2] successfully extracted features from lip movement and combined them with speech recognition to form an Audio-Visual Automatic Speech Recognition (AV-ASR) system. The results show that the system is more robust than ordinary speech recognition systems.

Over the past years, as deep learning technology has obtained extraordinary achievements in various fields, the focus of lipreading has also changed. Instead of trying to design some feature extraction algorithms manually to extract features, researchers used the deep network's powerful representation learning ability to automatically learn good features according to the task objectives. These features often have good generalization ability and can achieve good performance in a variety of scenarios. In 2011, Ngiam et al. [3] proposed an AV-ASR system based on depth autoencoder and Restricted Boltzmann Machines (RBMs) [4]. The visual feature extraction method based on the deep learning method is introduced into multimodal speech recognition for the rst time. In 2014, Noda et al. [5] of Waseda University used CNN as a feature extraction tool for lip image. The experimental results show that the visual features obtained by Convolutional Neural Network (CNN) are significantly better than the traditional methods including principal component analysis. In 2016, Wand et al. [6] used Long Short-Term Memory (LSTM) for lipreading and achieved a recognition rate of 79.6% on GRID. In 2016, Chung and Zisserman [7] established the first large-scale English lipreading database LRW under natural conditions according to the BBC program. Assael et al. [8] proposed LipNet based on the spatial-temporal convolution network and recurrent neural network in 2017 and used CTC as a network loss function in the LipNet network. The WLAS network proposed by Chung et al. in 2017, which is composed of CNN and Recurrent Neural Networks (RNN), obtains a 46.8% sentence accuracy rate on the LRS database with 10000 sample sentences.

At present, lipreading methods are divided into two categories according to different feature extraction methods: 1) Lipreading based on traditional manual feature extraction method; 2) Lipreading based on deep learning feature extraction method. For the traditional manual feature extraction method, the lip region should be extracted rstly; then, the feature extraction algorithm designed by the researchers extracting the bottom moving features of the lip region; and then through some linear functions such as Principal Component Analysis (PCA) and Discrete Cosine Transform (DCT) is used to process the extracted features and encode them into equal length feature vectors. Finally, suitable classes such as Artificial Neural Network (ANN), HMM are used for classification. In the deep learning method, it can be used iterative learning method to automatically extract more features than traditional methods from the video or image sequence; Then obtain the scores of each category through the deep model, and then adjust the network model parameters by way of backpropagation according to the labels of the training data, and nally achieve a good classification effect. In this paper, we develop a system for audio-video speech recognition called ‘Lip Reading’. In this, the system will recognise phrases and sentences being spoken by a talking face, with or without the audio. The system will also support multiple languages using Natural Language Processing (NLP).It will give results in real time.

To address the problem, we propose to use the 3D Convolutional Neural Networks models that have recently been employed for action recognition, scene understanding, and speaker verification and demonstrated promising results [9]–[6][7]. 3D CNNs concurrently extract features from both spatial and temporal dimensions, so the motion information is captured and concatenated in adjacent frames. We use 3D CNNs to generate separate channels of information from the input frames. The combination of all channels of correlated information creates the final feature representation.

The focus of the research effort described in this paper is to implement two non-identical 3D CNNs for audio-visual matching. The goal is to design nonlinear mappings that learn a non-linear embedding space between the corresponding audio-video streams using a simple distance metric. This architecture can be learned by evaluating pairs of audio-video data and later used for distinguishing between pairs of matched and non-matched audio-visual streams. One of the main advantages of our audio-visual model is the noise-robust audio features, which are

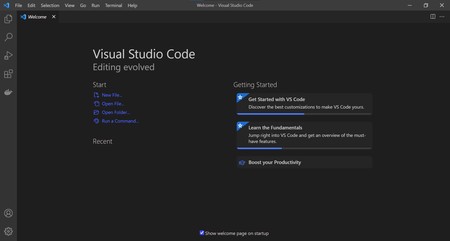
extracted from speech features with locality characteristics, and the visual features, which are extracted from spatial

and temporal information of lip motions. Both audio-visual features are extracted using 3D CNNs, allowing the temporal information to be treated separately for better decision making.

### Methodology

* + 1. **Visual Studio Code (VS Code)**

Visual Studio Code is a streamlined code editor with support for development operations like debugging, task running, and version control. It aims to provide just the tools a developer needs for a quick code-build-debug cycle and leaves more complex workflows to fuller featured IDEs, such as Visual Studio IDE. At its heart, Visual Studio Code features a lightning fast source code editor, perfect for day-to-day use. With support for hundreds of languages, VS Code helps you be instantly productive with syntax highlighting, bracket-matching, auto-indentation, box-selection, snippets, and more[10].



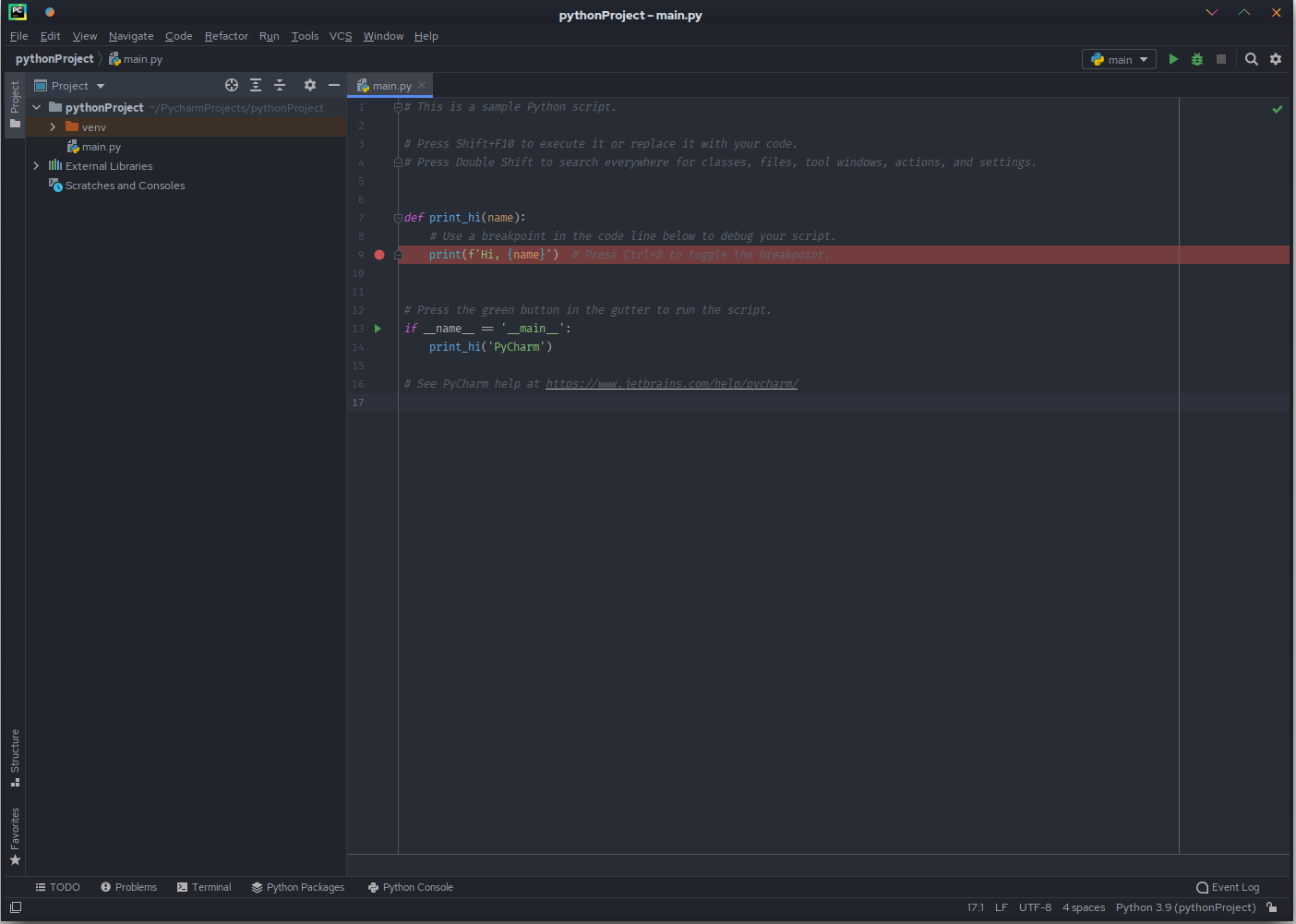
### Fig 2.2.1 Visual Studio

* + 1. **PyCharm**

PyCharm is an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) (IDE) used in [computer programming](https://en.wikipedia.org/wiki/Computer_programming), specifically for the [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) language. It is developed by the [Czech](https://en.wikipedia.org/wiki/Czech_Republic) company [JetBrains](https://en.wikipedia.org/wiki/JetBrains) (formerly known as IntelliJ). It provides code analysis, a graphical debugger, an integrated unit tester, integration with [version control systems](https://en.wikipedia.org/wiki/Revision_control) (VCSes), and supports web development with [Django](https://en.wikipedia.org/wiki/Django_(web_framework)) as well as [data science](https://en.wikipedia.org/wiki/Data_science) with [Anaconda](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)).

PyCharm is [cross-platform](https://en.wikipedia.org/wiki/Cross-platform), with [Windows](https://en.wikipedia.org/wiki/Windows), [macOS](https://en.wikipedia.org/wiki/MacOS) and [Linux](https://en.wikipedia.org/wiki/Linux) versions. The Community Edition is released under the [Apache License](https://en.wikipedia.org/wiki/Apache_License), and there is also Professional Edition with extra features – released under a [proprietary license](https://en.wikipedia.org/wiki/Proprietary_software)

[11].

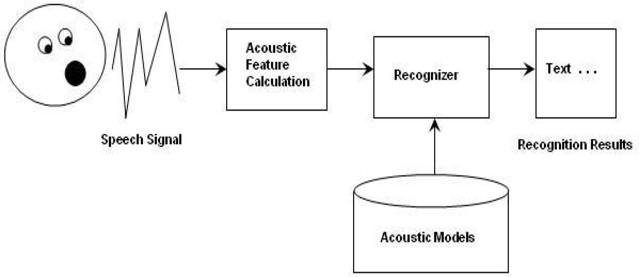


### Fig. 2.2.1 PyCharm

* + 1. **Natural Language Processing (NLP)**

Natural language processing (NLP) is a subfield of [linguistics](https://en.wikipedia.org/wiki/Linguistics), [computer science](https://en.wikipedia.org/wiki/Computer_science), and [artificial intelligence](https://en.wikipedia.org/wiki/Artificial_intelligence) concerned with the interactions between computers and human language, in particular how to program computers to process and analyze large amounts of [natural language](https://en.wikipedia.org/wiki/Natural_language) data. The goal is a computer capable of "understanding" the contents of documents, including the contextual nuances of the language within them. The technology can then accurately extract information and insights contained in the documents as well as categorize and organize the documents themselves.

Challenges in natural language processing frequently involve [speech recognition](https://en.wikipedia.org/wiki/Speech_recognition), [natural language understanding](https://en.wikipedia.org/wiki/Natural-language_understanding), and [natural language generation](https://en.wikipedia.org/wiki/Natural-language_generation)[12].



**Fig. 2.2.3 Natural language processing**

# CHAPTER 3 PLANNING AND FORMULATION

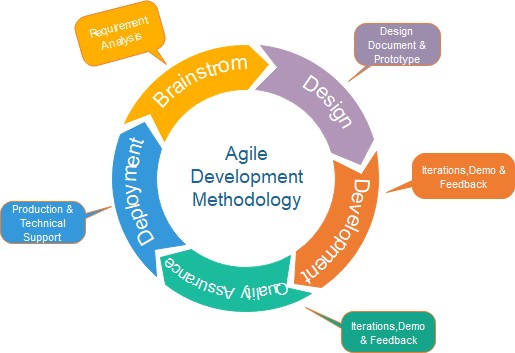
## Planning and Formulation

### Project Development Model

* + 1. **Project Model: Agile Model**

The meaning of Agile is swift or versatile."Agile process model" refers to a software development approach based on iterative development. Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning. The project scope and requirements are laid down at the beginning of the development process. Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance.

Each iteration is considered as a short time "frame" in the Agile process model, which typically lasts from one to four weeks. The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements. Each iteration involves a team working through a full software development life cycle including planning, requirements analysis, design, coding, and testing before a working product is demonstrated to the client.[13]



### Fig. 3.1.1 Agile Mode

* + 1. **Phases of Agile Model:**
       1. **Requirements gathering:** In this phase, you must define the requirements. You should explain business opportunities and plan the time and effort needed to build the project. Based on this information, you can evaluate technical and economic feasibility.
       2. **Design the requirements**: When you have identified the project, work with stakeholders to define requirements. You can use the user flow diagram or the high-level UML diagram to show the work of new features and show how it will apply to your existing system.
       3. **Construction/ Iteration:** When the team defines the requirements, the work begins. Designers and developers start working on their project, which aims to deploy a working product. The product will undergo various stages of improvement, so it includes simple, minimal functionality.
       4. **Testing:** In this phase, the Quality Assurance team examines the product's performance and looks for the bug.
       5. **Deployment:** In this phase, the team issues a product for the user's work environment.
       6. **Feedback:** After releasing the product, the last step is feedback. In this, the team receives feedback about the product and works through the feedback.

### Advantage of Agile Method:

* + - 1. Frequent Delivery
      2. Face-to-Face Communication with clients.
      3. Efficient design and fulfils the business requirement.
      4. Anytime changes are acceptable.
      5. It reduces total development time

### Disadvantage of Agile Method:

* + - 1. It is not useful for small development projects.
      2. There is a lack of intensity on necessary designing and documentation.
      3. It requires an expert project member to take crucial decisions in the meeting.
      4. Cost of Agile development methodology is slightly more as compared to other development methodology.
      5. The project can quickly go off track if the project manager is not clear about requirements and what outcome he/she wants.

### Timeline Chart

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Timeline Chart for the project Lip Reading- An Efficient Cross Audio-Video Recognition Using 3D Convolutional Neural Network** | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| **Months** | **August** | | | | **September** | | | | **October** | | | | **November** | | | | **December** | | | | **January** | | | | **February** | | | | **March** | | | |
| **Phases** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** | **1** | **2** | **3** | **4** |
| Requirements Analysis |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Feasibility Study |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Design and architecture |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Product Development |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Implementati on |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Testing |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Report Organization |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Table 3.1 Timeline Chart**

### Feasibility Analysis

Next step in analysis is feasibility study. By performing a feasibility study the scope of the system will be defined completely. Most computer systems are developed to satisfy a known user requirement. This means that the first event in the life cycle of a system is usually the task of studying whether it is feasible to computerize a system under consideration or not. Once the decision is made, a report is forwarded and it is known as Feasibility Report. The feasibility is studied under the following contexts:

### Technical Feasibility

It involves determining whether or not a system can actually be constructed to solve the problem at hand. The technical issues raised during the feasibility stage of investigation are related to achievability of project’s goal and possibility of completion of project.

### Economical Feasibility:

This feasibility deals with the cost/benefit analysis. A number of intangible benefits like user friendliness, robustness and security were pointed out. The cost that will be incurred upon the implementation of this project would be quite nominal.

### Operational Feasibility:

The developed system will be very reliable and user friendly. All the features and operations that we will implement in our project are possible to implement and thus feasible. This will facilitate easy use and adoptability of the system. With the use of menus, and proper validation required it becomes fully understandable to the common user and operational with the user.

# CHAPTER 4 REQUIREMENTS ANALYSIS

## Requirements Analysis

### Hardware Requirement

* + 1. Processor: Intel i7 (9th Gen)
    2. Hard Disk: 1 TB or more
    3. Computer / Laptop
    4. Webcam and Microphone
    5. RAM: 8 GB or above

### Software Requirement

* + 1. Operating System : Windows 10/11
    2. Visual Studio
    3. PyCharm
    4. Python
    5. Shell

### Functional Requirement

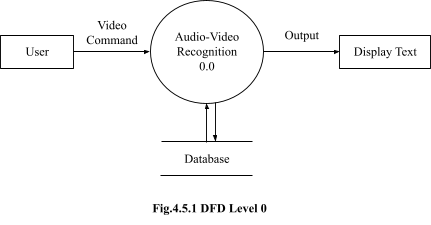
* + 1. In software engineering, a functional requirement defines a function of a software system or its component. A function is described as a set of inputs, the behavior, and outputs. Functional requirements may be calculations, technical details, data manipulation and processing and other specific functionality that define what a system is supposed to accomplish.
    2. Python:Backend of our project is based on python as it provides good functionalities for the working of the project.We had used python tkinter to make the GUI.
    3. Natural language processing (NLP)**:** The NLP helps us to work with the user inputs to process the voice commands and regional languages into computer language so that the computer can understand the user input and give the output accordingly and effectively.

### Non-Functional Requirements

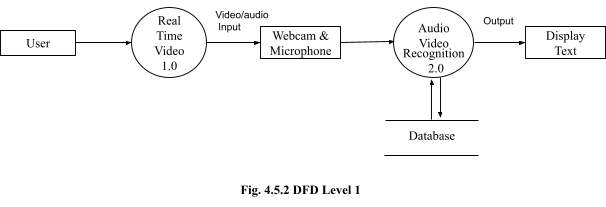
* + 1. In system engineering and requirements engineering, a non-functional requirement is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors.
    2. This should be contrasted with functional requirements so that our design can be more effective and more beneficial for the user to use our application.
    3. There is a section of dialog flow that we want to use to make our application more efficient, but that functionality is charging a cost of 300$ which is very high. If we had that functionality our application can be more handy then now.

### Data Flow Diagram

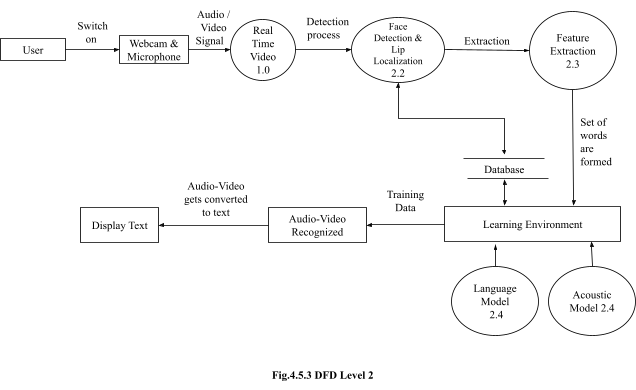
* + 1. **DFD LEVEL 0:-**



### DFD LEVEL 1:



* + 1. **DFD LEVEL 2:**

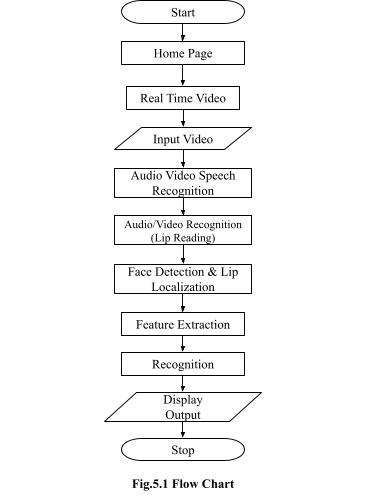


## CHAPTER 5 SYSTEM DESIGN

* 1. **Flow Chart**

## System Design

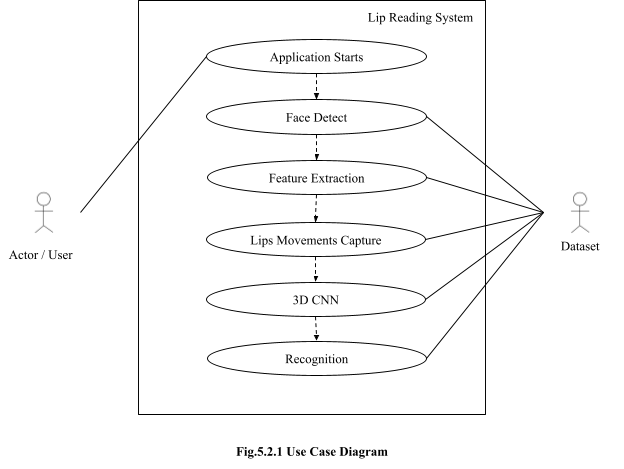
A flowchart is a type of [diagram](https://en.wikipedia.org/wiki/Diagram) that represents a [workflow](https://en.wikipedia.org/wiki/Workflow) or [process](https://en.wikipedia.org/wiki/Process). A flowchart can also be defined as a diagrammatic representation of an [algorithm](https://en.wikipedia.org/wiki/Algorithm), a step-by-step approach to solving a task.



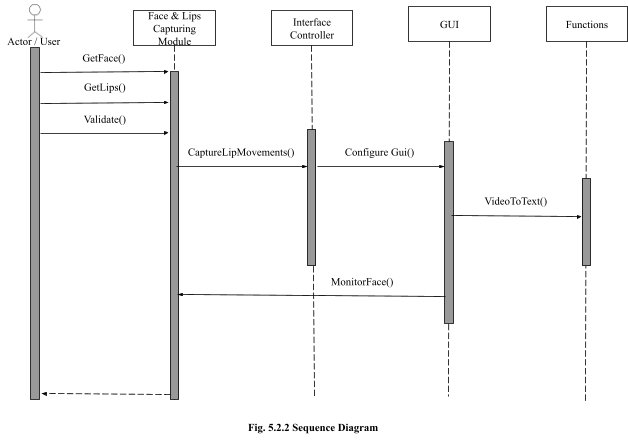
This flow chart represents flow of data or say the users flow while accessing the software.

### UML Diagrams

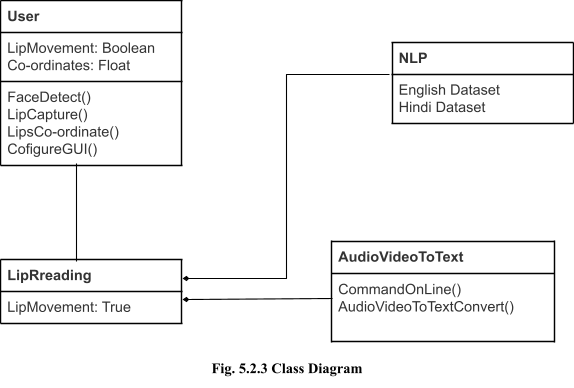
* + 1. **Use Case Diagram**



**5.2.4 Sequence Diagram**



**5.2.3 Class Diagram**



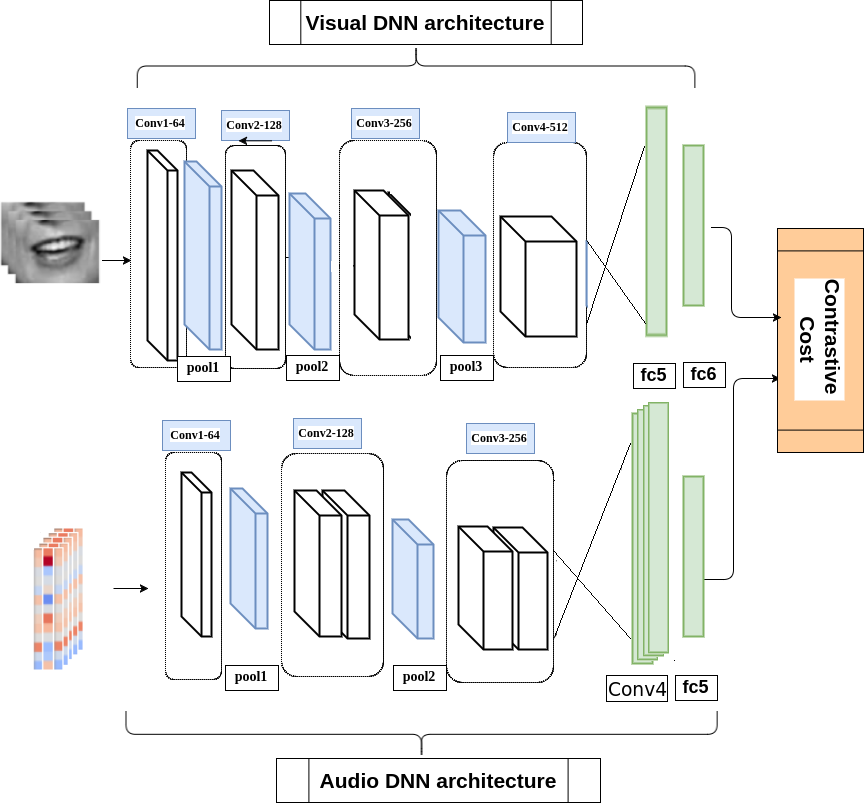
**CHAPTER 6 ALGORITHM DEVELOPMENT**

## Algorithm Development

### 1 Algorithm Development

**6.1.1 3D CNN**

The 3D convolutional neural network in which two different networks with different sets of weights must be trained. For the visual network, the lip motions spatial information alongside the temporal information are incorporated jointly and will be fused for exploiting the temporal correlation. For the audio network, the extracted energy features are considered as a spatial dimension, and the stacked audio frames form the temporal dimension. In the 3D CNN , the convolutional operations are performed on successive temporal frames for both audio-visual streams.



### Fig 6.1.1 3D CNN

**6.2 Model Development**

### # Implementation of GUI of the system

import sys

from tkinter import TOP import cv2

import threading import tkinter as tk import tkinter.ttk as ttk

from queue import Queue from PIL import Image from PIL import ImageTk

class App(tk.Frame):

def init (self, parent, title):

tk.Frame. init (self, parent) self.is\_running = False self.thread = None

self.queue = Queue()

self.photo = ImageTk.PhotoImage(Image.new("RGB", (800, 600), "white")) parent.wm\_withdraw()

parent.wm\_title(title) self.create\_ui() self.grid(sticky=tk.NSEW)

self.bind('<<MessageGenerated>>', self.on\_next\_frame) parent.wm\_protocol("WM\_DELETE\_WINDOW", self.on\_destroy) parent.grid\_rowconfigure(0, weight = 1) parent.grid\_columnconfigure(0, weight = 1)

parent.wm\_deiconify()

def create\_ui(self):

self.button\_frame = ttk.Frame(self)

self.start\_button = ttk.Button(self.button\_frame, text="Real Time", command=self.start) self.start\_button.pack(side=tk.LEFT)

self.stop\_button = ttk.Button(self.button\_frame, text="Pause", command=self.stop) self.stop\_button.pack(side=tk.LEFT)

self.stop\_button = ttk.Button(self.button\_frame, text="Close", command=self.on\_destroy) self.stop\_button.pack(side=tk.LEFT)

inputtxt = tk.Text(self,

height=8, width=60)

inputtxt.pack(side=tk.BOTTOM)

self.view = ttk.Label(self, image=self.photo) self.view.pack(side=tk.TOP, fill=tk.BOTH, expand=True) self.button\_frame.pack(side=tk.BOTTOM, fill=tk.X, expand=True)

def on\_destroy(self): self.stop() self.after(20)

if self.thread is not None: self.thread.join(0.2)

self.winfo\_toplevel().destroy()

def start(self):

self.is\_running = True

self.thread = threading.Thread(target=self.videoLoop, args=()) self.thread.daemon = True

self.thread.start()

def stop(self):

self.is\_running = False

def videoLoop(self, mirror=False):

No=0

cap = cv2.VideoCapture(No) cap.set(cv2.CAP\_PROP\_FRAME\_WIDTH, 800)

cap.set(cv2.CAP\_PROP\_FRAME\_HEIGHT, 600)

while self.is\_running:

ret, to\_draw = cap.read() if mirror is True:

to\_draw = to\_draw[:,::-1]

image = cv2.cvtColor(to\_draw, cv2.COLOR\_BGR2RGB) self.queue.put(image) self.event\_generate('<<MessageGenerated>>')

def on\_next\_frame(self, eventargs): if not self.queue.empty():

image = self.queue.get()

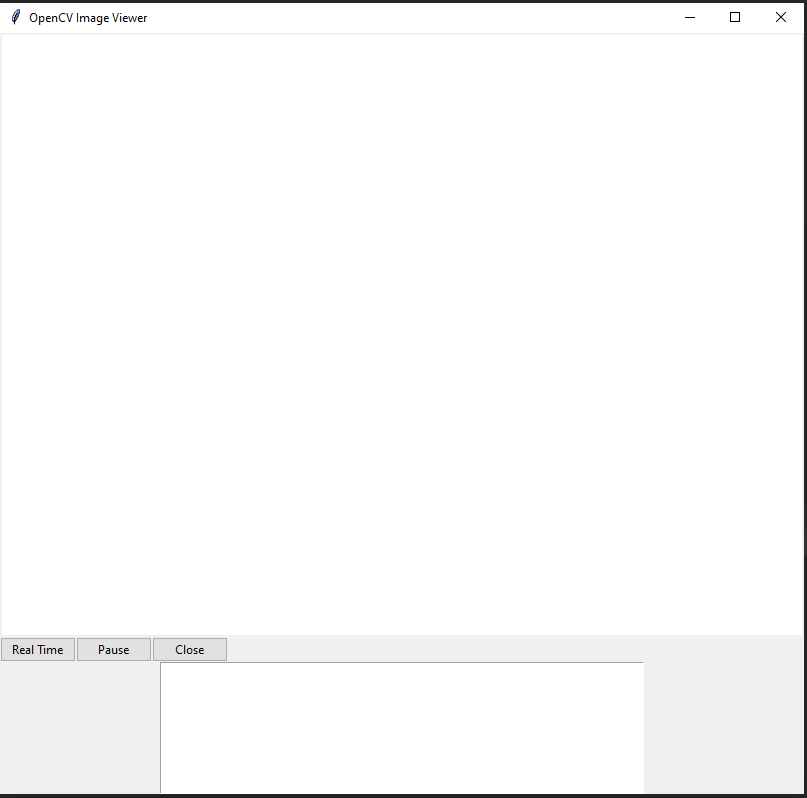
image = Image.fromarray(image) self.photo = ImageTk.PhotoImage(image) self.view.configure(image=self.photo)

def main(args):

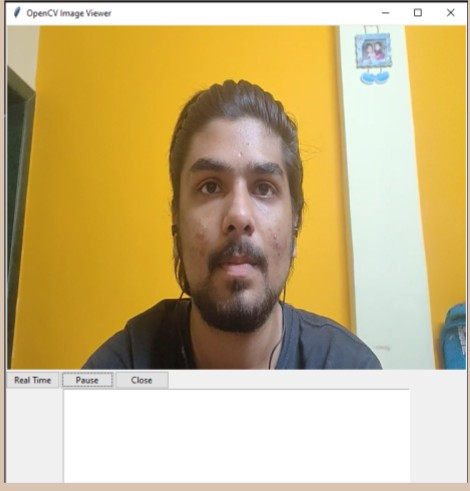
root = tk.Tk()

app = App(root, "OpenCV Image Viewer") root.mainloop()

if name == ' main ': sys.exit(main(sys.argv))



### Fig. 6.2.1 Home Page



**Fig. 6.2.2 Real Time Video Capture Screen**

## CHAPTER 7

**CONCLUSION AND FUTURE SCOPE**

* 1. **Conclusion**

## Conclusion and Future Scope

In this study, we proposed an AVSR system based on deep learning architectures for audio and visual feature extraction. In this project we develop a system that will allow input voice to see into text format. Communication among human beings is dominated by spoken language, therefore it is natural for people to expect voice interfaces with computers. This can be accomplished by developing a voice recognition system: Audio/Video-to-text which allows computers to translate voice requests and dictation into text. This project made a clear and simple overview of working of Audio/Video to text systems.

### Future Scope

* + 1. It will be useful in applications related to improved hearing aids.
    2. Video conferencing in silent environments.
    3. High-quality speech recovery from background noise .
    4. Generating a voice for people who cannot produce voiced sounds (aphonia).
    5. It will be also useful in applications related to biometric authentication.

## Acknowledgement

As every project is ever complete with the guidance of experts. So we would like to take this opportunity to thank all those individuals who have contributed in visualizing this project.

We express our deepest gratitude to our project guide Prof. ------------(CSE(AIML/IOT) Department, Smt. Indira Gandhi College of Engineering, University of Mumbai) for her valuable guidance, moral support and devotion bestowed on us throughout our work.

We would also take this opportunity to thank our project coordinator Prof. Sarita Khedikar for her guidance in selecting this project and also for providing us all the details on proper presentation of this project.

We extend our sincere appreciation to our entire professors from Smt. Indira Gandhi College of Engineering for their valuable inside and tip during the designing the project. Their contributions have been valuable in many ways that we find it difficult to acknowledge them individually.

We are also grateful to our HOD Prof. Sonali Deshpande for extending his help directly and indirectly through various channels in our project.

If I can say in words I must at the outset my intimacy for receipt of affectionate care to Smt. Indira Gandhi College of Engineering for providing such a simulating atmosphere and wonderful work environment.

## References

1. W. H. Sumby and I. Pollack, ``Visual contribution to speech intelligibility in noise,'' J. Acoust. Soc. Amer., vol. 26, no. 2, pp. 212215, Mar. 1954.
2. E. D. Petajan, ``Automatic lipreading to enhance speech recognition,'' in Proc. Global Telecommun. Conf., 1984, pp. 265272.
3. J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, ``Multimodal deep learning,'' in Proc. 28th Int. Conf. Mach. Learn. (ICML), 2011, pp. 689696.
4. H. Lee, C. Ekanadham, and A. Y. Ng, ``Sparse deep belief net model for visual area V2,'' in Proc. Adv. Neural Inf. Process. Syst., 2008, pp. 873880.
5. K. Noda, Y. Yamaguchi, K. Nakadai, H. G. Okuno, and T. Ogata, ``Lipreading using convolutional neural network,'' in Proc. Conf. Int. speech Commun. Assoc., 2014, pp. 11491153.
6. S. Suwajanakorn, S. M. Seitz, and I. Kemelmacher-Shlizerman, ``Synthesizing Obama: Learning lip sync from audio,'' ACM Trans. Graph., vol. 36, no. 4, p. 95, Jul. 2017.
7. J. S. Chung and A. Zisserman, ``Lip reading in the wild,'' in Proc. Asian Conf. Comput. Vis. Cham, Switzerland: Springer, 2016, pp. 87103.
8. Y. M. Assael, B. Shillingford, S. Whiteson, and N. de Freitas, ``Lip-Net: End-to-end sentence-level lipreading,'' 2016, arXiv:1611.01599. [Online]. Available: <http://arxiv.org/abs/1611.01599>
9. K. Noda, Y. Yamaguchi, K. Nakadai, H. G. Okuno, and T. Ogata, ``Lipreading using convolutional neural network,'' in Proc. Conf. Int. speech Commun. Assoc., 2014, pp. 11491153.
10. <https://en.wikipedia.org/wiki/Visual_Studio_Code>
11. <https://en.wikipedia.org/wiki/PyCharm>
12. <https://en.wikipedia.org/wiki/Natural_language_processing>
13. [<https://www.javatpoint.com/software-engineering-agile-model>