MOCK INTERVIEW PREPARATION

A PROJECT REPORT

Submitted by

N SWARNA KARTHIKA 2021115114

fABHARNA SHREE M **2022115071**

HARSHAVARDHINI S 2022115131

A report for the Summer Project - May 2024
submitted to the faculty of

INFORMATION SCIENCE AND TECHNOLOGY

in partial fulfillment

for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

INFORMATION TECHNOLOGY

DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY

COLLEGE OF ENGINEERING GUINDY

ANNA UNIVERSITY

CHENNAI 600 025

MAY 2024

ANNA UNIVERSITY

CHENNAI - 600 025

BONAFIDE CERTIFICATE

Certified that this project report titled **“MOCK INTERVIEW PREPARATION”** is the bonafide work of **N SWARNA KARTHIKA (2021115114)**, **ABHARNA SHREE M (2022115071)**, **HARSHAVARDHINI s (2022115131)**, who carried out project work under my supervision. Certified further that to the best of my knowledge and belief, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion on this or any other candidate.

PLACE:CHENNAI
DATE:

Dr. K. Vidya
ASSOCIATE PROFESSOR
PROJECT GUIDE
DEPARTMENT OF IST, CEG
ANNA UNIVERSITY
CHENNAI 600025

COUNTERSIGNED

Dr. S. SWAMYNATHAN
HEAD OF THE DEPARTMENT
DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY
COLLEGE OF ENGINEERING GUINDY
ANNA UNIVERSITY
CHENNAI 600025

ABSTRACT

Preparing for interviews can be challenging, especially without access to feedback on both verbal and non-verbal cues. This project aims to create a smart mock interview app that enhances interview readiness by using advanced video and audio analysis. The app utilizes eye gaze tracking and emotion recognition for real-time video feedback, assessing how well users maintain eye contact and identifying emotions.

The app also leverages a large language model (LLM) to dynamically generate interview questions based on the user's resume. It compares user responses with model-generated answers, delivering similarity scores to help users understand how well they addressed the question. Additionally, the app can scrape subject-specific questions from web resources, allowing users to practice on varied topics, and it uses the LLM to evaluate the accuracy and relevance of answers, ensuring that users receive targeted, constructive feedback.

By combining advanced AI techniques with real-time feedback, this project offers a comprehensive mock interview solution designed to improve user confidence and interview performance.

ABSTRACT TAMIL

ACKNOWLEDGEMENT

It is my privilege to express my deepest sense of gratitude and sincere thanks to **Dr. K. VIDYA,** Associate Professor, Project Guide, Department of Information Science and Technology, College of Engineering, Guindy, Anna University, for her constant supervision, encouragement, and support in my project work. I greatly appreciate the constructive advice and motivation that was given to help me advance my project in the right direction.

I am grateful to **Dr. S. SWAMYNATHAN,** Professor and Head, Department of Information Science and Technology, College of Engineering Guindy, Anna University for providing us with the opportunity and necessary resources to do this project.

I would also wish to express my deepest sense of gratitude to the Members of the Project Review Committee: **Dr. M. VIJAYALAKSHMI,** Professor, **Dr. H. RIASUDHEEN,** Teaching Fellow of Information Science and Technology,College of Engineering Guindy, Anna University, for their guidance and useful suggestions that were beneficial in helping me improve my project.

I also thank the faculty member and non teaching staff members of the Department of Information Science and Technology, Anna University, Chennai for their valuable support throughout the course of our project work.

ABHARNA SHREE M
(2022115071)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

API	Application Programming Interface
BART	Bidirectional and Auto-Regressive Transformers
CSRF	Cross-Site Request Forgery
DNN	Deep Neural Network
FER	Facial Expression Recognition
LLM	Large Language Model
MVC	Model-View-Controller
NLP	Natural Language Processing
OCR	Optical Character Recognition
PDF	Portable Document Format
PyPDF2	Python PDF Toolkit
STT	Speech-to-Text
UI	User Interface
UX	User Experience
VAD	Voice Activity Detection
WER	Word Error Rate

CHAPTER 1

INTRODUCTION

Mock interview preparation plays a critical role in helping candidates improve their chances of success during job interviews. Despite the rise of various interview preparation tools, they often fail to provide personalized feedback or realistic simulation of interview scenarios. Existing platforms mainly focus on static question sets, with limited support for assessing non-verbal cues such as body language, facial expressions.

This project aims to address these gaps by developing an intelligent mock interview preparation app. The app integrates AI-based video and audio analysis to offer real-time, dynamic feedback. Using eye gaze tracking (via dlib), the app evaluates the user's level of eye contact, while emotion recognition (FER) analyzes facial expressions to assess emotional responses during the interview. Additionally, the system processes audio to evaluate speech aspects like tone, fluency, and clarity, giving users insights into how they can improve their verbal communication.

Furthermore, the app leverages Large Language Models (LLMs) to generate tailored interview questions based on the user's resume. The app then compares the user's responses to ideal answers and generates similarity scores, helping users identify areas of strength and those requiring improvement. To enhance the breadth of the interview preparation, the system scrapes web data to pull relevant questions, allowing for dynamic and up-to-date practice.

The goal is to create a comprehensive and customizable interview preparation tool that improves the overall preparedness of candidates by offering feedback that extends beyond mere question answering, focusing on delivery, presentation, and emotional intelligence.

1.1 PROBLEM STATEMENT

Despite the availability of interview preparation tools, most platforms fail to offer personalized, dynamic feedback that addresses both the verbal and non-verbal aspects of interviews. This leads to candidates missing opportunities to improve their body language, speech delivery, and emotional control during interviews.

1.2 RESEARCH OBJECTIVES

- Develop an AI-driven mock interview app that provides real-time, comprehensive feedback on verbal and non-verbal responses during interviews.
- Incorporate technologies like eye gaze tracking, emotion recognition, and audio processing to analyze and improve interview performance.
- Use Large Language Models (LLMs) to generate personalized interview questions based on the user's resume and assess the quality of responses using similarity scores.
- Integrate web scraping to gather diverse, up-to-date interview questions, offering a wider range of practice scenarios.

1.3
OVERVIEW OF THE PROPOSED SYSTEM

This work presents a comprehensive mock interview preparation app that combines video, audio, and AI analysis to simulate a real-world interview experience. The app's video component uses dlib for eye gaze tracking, alongside emotion recognition to analyze facial expressions, providing insights into the candidate's body language. The audio analysis evaluates aspects such as tone, clarity, and pacing of the speech. Additionally, the app uses an LLM to generate customized interview questions based on the user's resume and compares their responses with ideal answers. The system also features a dynamic question-generation module that pulls relevant interview questions from online sources, ensuring diverse and up-to-date practice.

1.4
TESTING AND VALIDATION

To validate the system's performance, it is essential to test it in both simulated and real-world environments. Various testing methodologies will be explored, including:

Simulation Testing: The mock interview app will be tested in a controlled simulation environment to assess its ability to generate realistic questions and evaluate responses. Real-World Testing: After initial simulation testing, real-world testing will be performed to ensure the app's functionality and usability are effective in practical scenarios.

1.5
RESEARCH CHALLENGES

- Lack of datasets for training emotion recognition and gaze tracking models specifically tailored for interview scenarios.
- Developing accurate LLM-driven question generation that maintains relevance to the candidate's resume.
- Ensuring real-time analysis of video and audio for dynamic feedback without performance bottlenecks.

1.6
RESEARCH CONTRIBUTION

The contribution of this work lies in creating a fully integrated system for interview preparation that blends AI-based video and audio feedback with LLM-generated interview questions. The system will provide a comprehensive evaluation, improving the candidate's verbal and non-verbal communication and overall confidence in real interview settings.

1.7
ORGANIZATION OF THE REPORT

This report is structured as follows:

CHAPTER 2: Literature Review – This chapter provides a review of existing research on interview preparation technologies, AI in interview systems, and the methodologies employed in similar systems.

CHAPTER 3: System Design – This chapter discusses the architecture and design of the system, detailing the components, algorithms, and technologies used in building the app.

CHAPTER 4: Implementation – This chapter explains the process of implementing the app, including the programming languages, tools, and frameworks employed.

CHAPTER 5: Results and Analysis – This chapter presents the outcomes of the system, comparing the results with expectations and discussing any challenges encountered.

CHAPTER 6: Conclusion and Future Work – This chapter summarizes the findings, discusses the implications of the work, and outlines areas for future research and development.

CHAPTER 2

LITERATURE SURVEY

2.1
PDF Resume Parsing and Summarization

PDF parsing and summarization are key steps in processing text-heavy documents like resumes. Parsing extracts raw text from a PDF, making it accessible for automated analysis and manipulation. Summarization then condenses the parsed text to its core points, providing a manageable summary that captures essential information without too much detail.

2.1.1 PyPDF2

'PyPDF2' is a Python library used to extract text from PDF documents. It allows access to the text content on each page of the PDF, converting it into a usable format for further processing. PyPDF2 helps retrieve and organize resume data, for summarization and question generation.

2.1.2 BART

BART (Bidirectional and Auto-Regressive Transformers) is a transformer-based model designed for text summarization. Using the model 'facebook/bart-large-cnn' model, BART generates a concise summary of the extracted resume text. This summarization step allows the system to focus on the most relevant points within the resume, making the subsequent question generation more targeted and efficient.

2.2 Question and Answer Generation

2.2.1 Language Model Integration

For generating context-specific interview questions and expected answers, the system uses 'Mistral-7B-Instruct-v0.3'. This model is optimized for instruction-following tasks, making it suitable for creating interview questions based on resume content.

2.2.2 Chain Mechanism

The model is accessed through langchain 'LLMChain', utilizing a prompt template that includes the summarized resume content to guide the model in generating questions and expected answers.

2.2.3 Data Extraction and Formatting

The response, consisting of questions and answers, is parsed to separate questions from their expected answers using regular expressions, making them ready for user interaction.

2.3 Video and Audio Input Processing

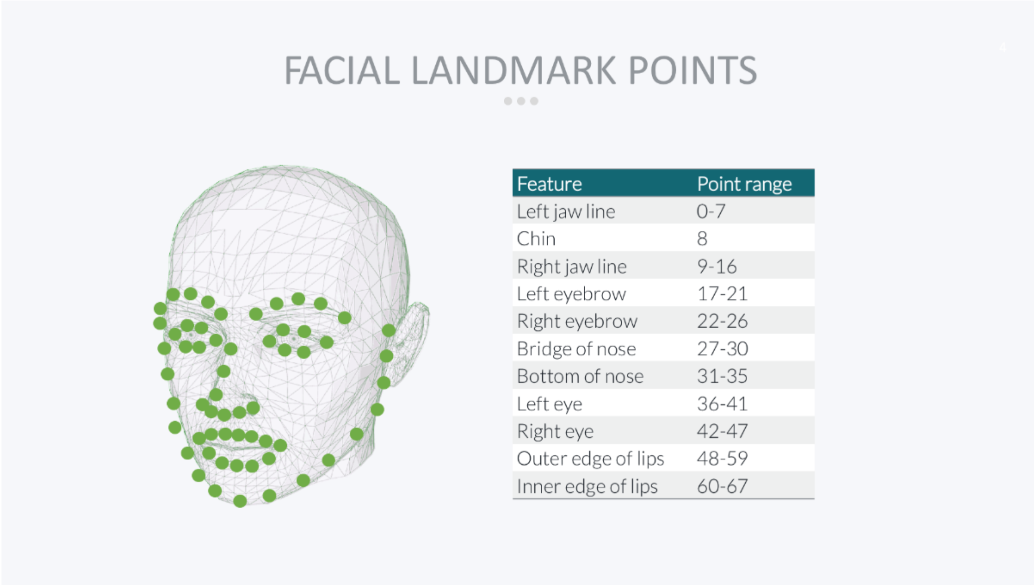


Figure 2.1: Facial Recognition Points Detected by Dlib

2.3.1 Video Capture and Analysis

The system captures live video of the user using OpenCV, which is then resized, converted to grayscale, and streamed in real-time. Facial landmarks are identified with ‘dlib’, focusing particularly on eye regions for behavioral analysis.

2.3.2 Emotion Detection

The FER library is employed to detect emotions from the video frames, using MTCNN to identify facial features before categorizing emotions (e.g., happiness, sadness) based on facial expressions.

2.3.3 Eye Movement Detection

Eye positions are monitored through landmarks detected by ‘dlib’, and movements are tracked to measure possible nervousness or

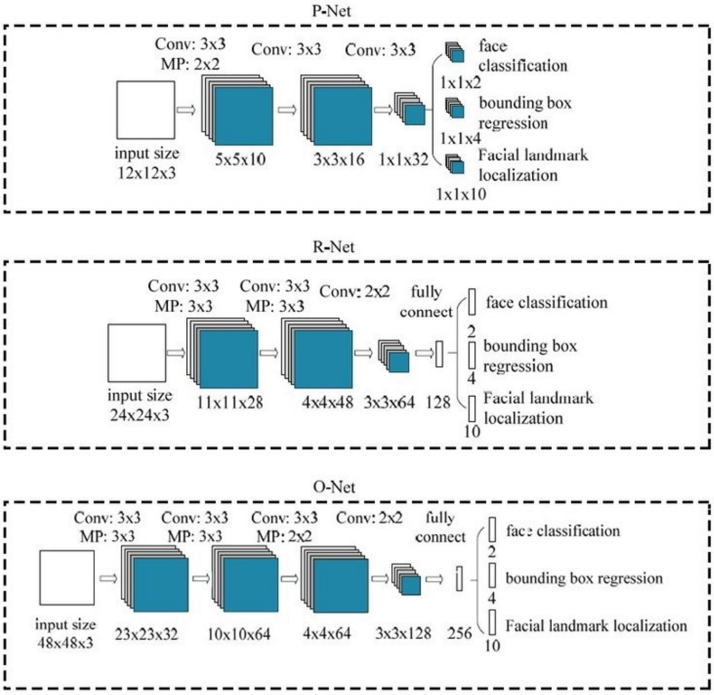


Figure 2.2: Face Detection Using MTCNN for Emotion Recognition

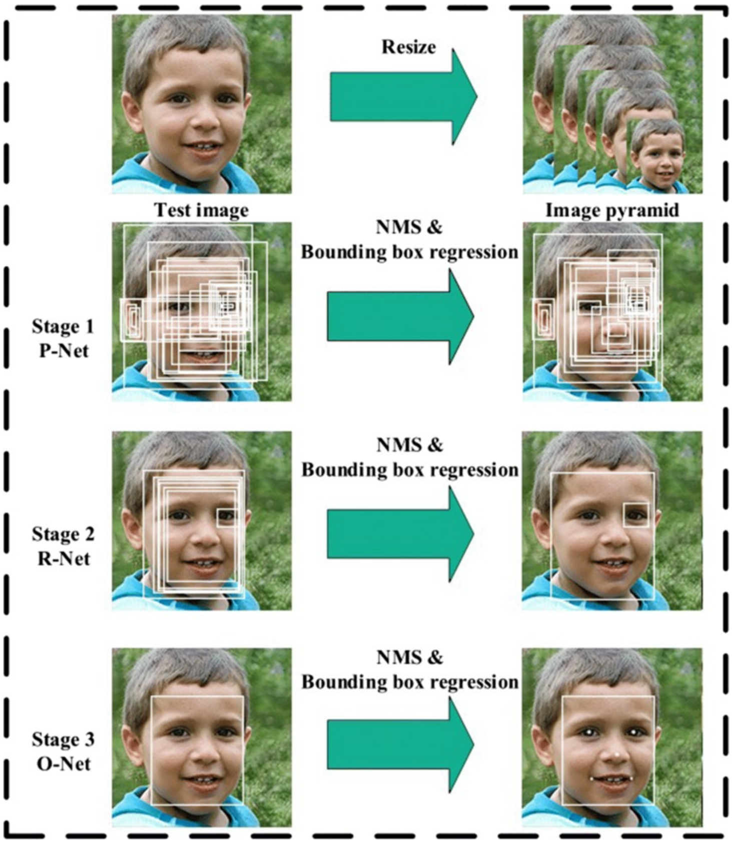


Figure 2.3: MTCNN Facial Landmark Detection

attentiveness based on thresholded movement frequency in left and right directions.

2.4
Speech-to-Text Conversion and Answer Accuracy Assessment

2.4.1 Speech Recognition

Google's Speech-to-Text API is used to transcribe spoken answers. This transcription allows the comparison of spoken responses to generated answers.

2.4.2 Similarity Checking

The 'paraphrase-MiniLM-L6-v2' model is used to compare the accuracy of spoken answers against the expected answers. The model computes sentence similarity, converting it to a percentage score to gauge how closely the user's spoken answer aligns with the ideal response.

2.5
Evaluation and Feedback System

2.5.1 Confidence and Nervousness Detection

Combining metrics from audio and video, the system computes overall performance feedback. Eye movements and detected emotions are analysed to evaluate confidence and nervousness.

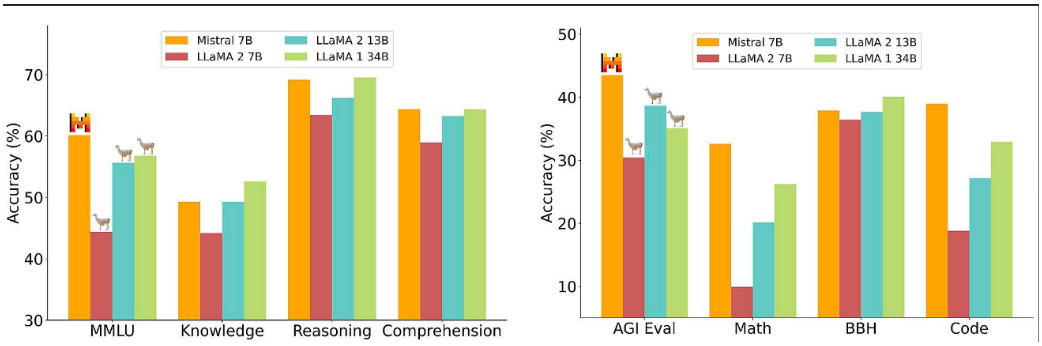


Figure 2.4: Performance Comparison of Mistral 7B with Other Language Models

2.5.2 Technical Score Calculation

Based on the similarity score between spoken and generated answers, the system calculates a technical accuracy percentage, giving an overall comparison of the candidate's alignment with expected answers.

2.5.3 Feedback Presentation

The system outputs a comprehensive assessment, displaying the predominant emotion, eye movement statistics, and the technical accuracy score, summarizing the user's performance.

CHAPTER 3

SYSTEM DESIGN

The application is a real-time interview preparation system that analyzes a user's video feed, emotions, and eye movements, extracts resume content, generates personalized interview questions, and evaluates spoken answers in comparison to expected responses.

3.1
System Architecture

3.1.1 Frontend

The frontend is built using HTML, CSS, and JavaScript for video streaming, form submissions, and dynamic updates. Video streaming is handled by `WebRTC` for low-latency performance, with JavaScript for real-time feedback on facial expressions and emotion analysis.

3.1.2 Backend (Django)

The backend uses Django to handle REST API requests, video streaming, resume extraction, question generation, and response processing. Django's `REST Framework (DRF)` is used to create API endpoints. Django Channels enable asynchronous handling for real-time video processing and speech analysis tasks.

3.1.3 External APIs

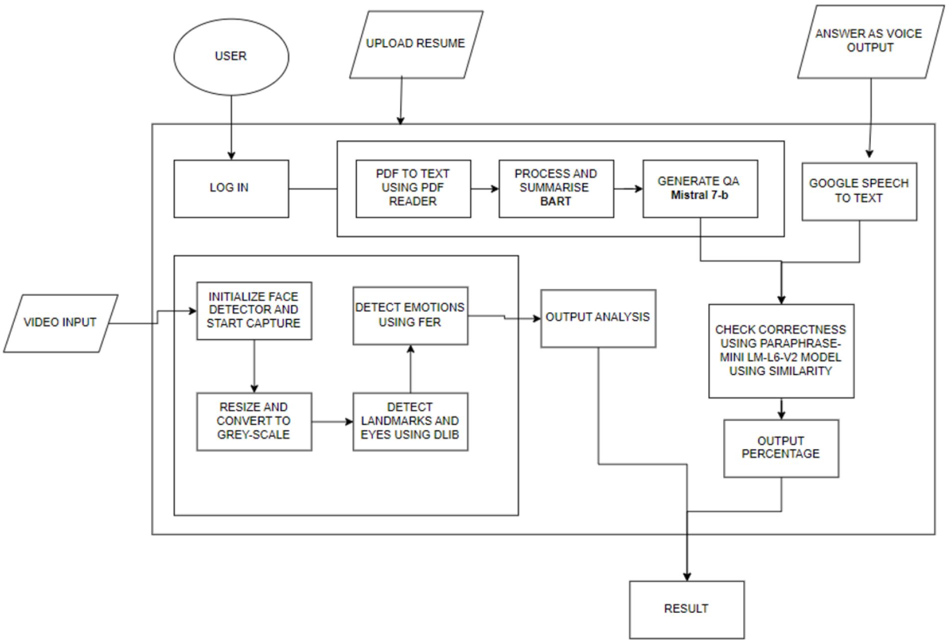


Figure 3.1: Overall Architecture of the Mock Interview System

External APIs are employed for question generation and text summarization:

- **Hugging Face Transformers API**: Provides NLP models for question generation and summarization, specifically leveraging the `bart-large-cnn` and `Mistral-7B` models.
- **Google Speech Recognition API**: Converts captured audio to text for real-time speech processing.

3.2
Technical Specifications and Calculations

3.2.1 Eye Movement and Emotion Analysis

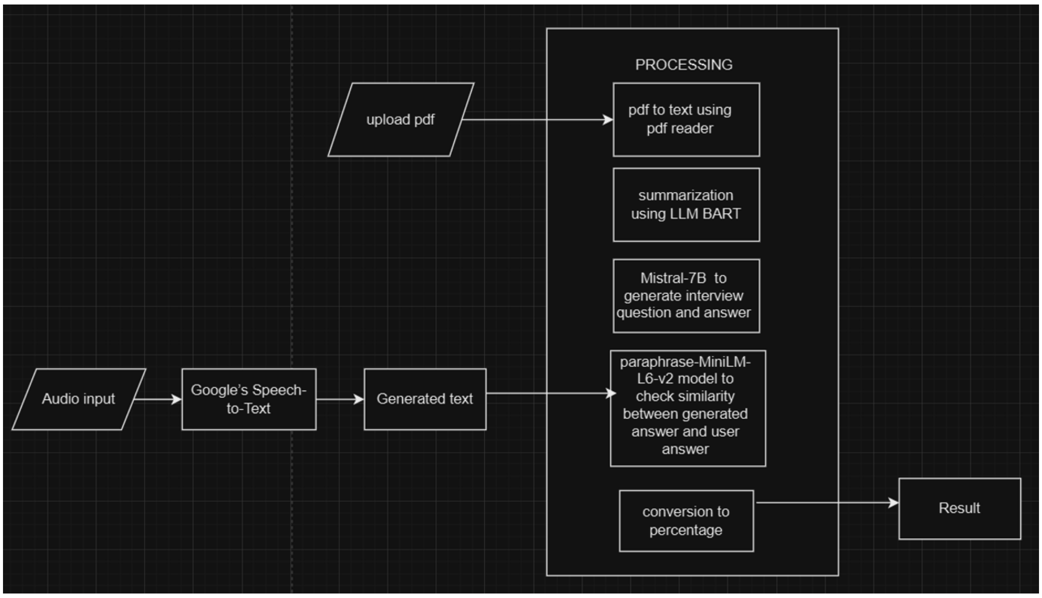


Figure 3.2: Architecture for Question Generation and Audio Analysis

- **Landmark Detection**: Dlib's model tracks 68 facial points, focusing on eye region points for movement analysis.
- **Movement Calculation**: Eye movement is detected by computing the distance between frames' center coordinates of the eyes. A threshold (`MOVEMENT_THRESHOLD`) is set to ignore small, involuntary movements.
- **Emotion Detection**: FER's output is classified into categories; majority voting over recent frames determines the most prevalent emotion.

3.2.2 Resume Parsing and Summarization

- **Text Extraction**: Text from each PDF page is concatenated to form a complete resume.
- **Summarization Model**: The `bart-large-cnn` model condenses the resume content, limiting output to 100 words.

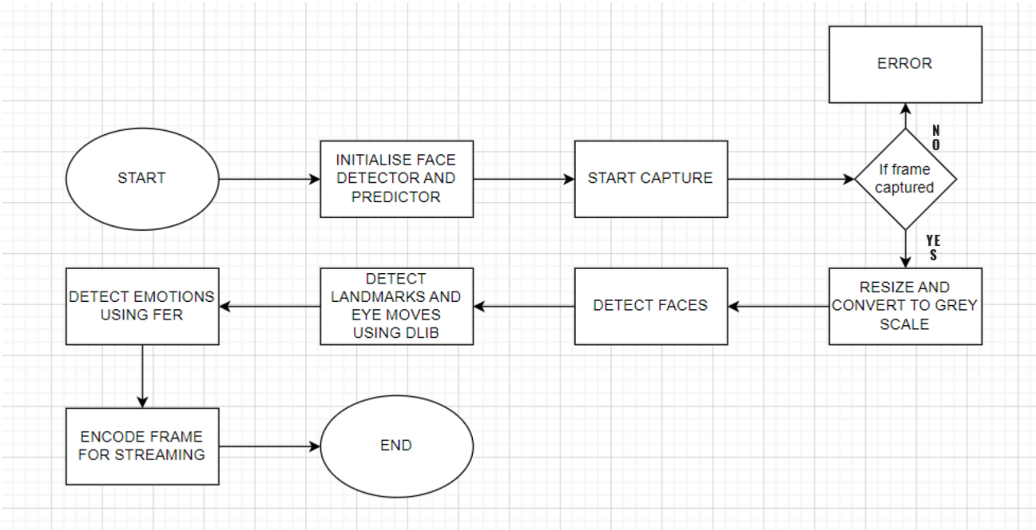


Figure 3.3: Architecture for Video Analysis Module

3.2.3 Interview Question Generation

- **Prompting Method**: The summarized resume serves as input prompt for LLM-generated questions.
- **Regex Parsing**: Splits questions and expected answers for structured frontend display.

3.2.4 Answer Similarity Calculation

- **Cosine Similarity Score**: Calculated using Sentence Transformers API; scores above 0.7 are considered acceptable matches.

3.2.5 Mathematical/Statistical Considerations

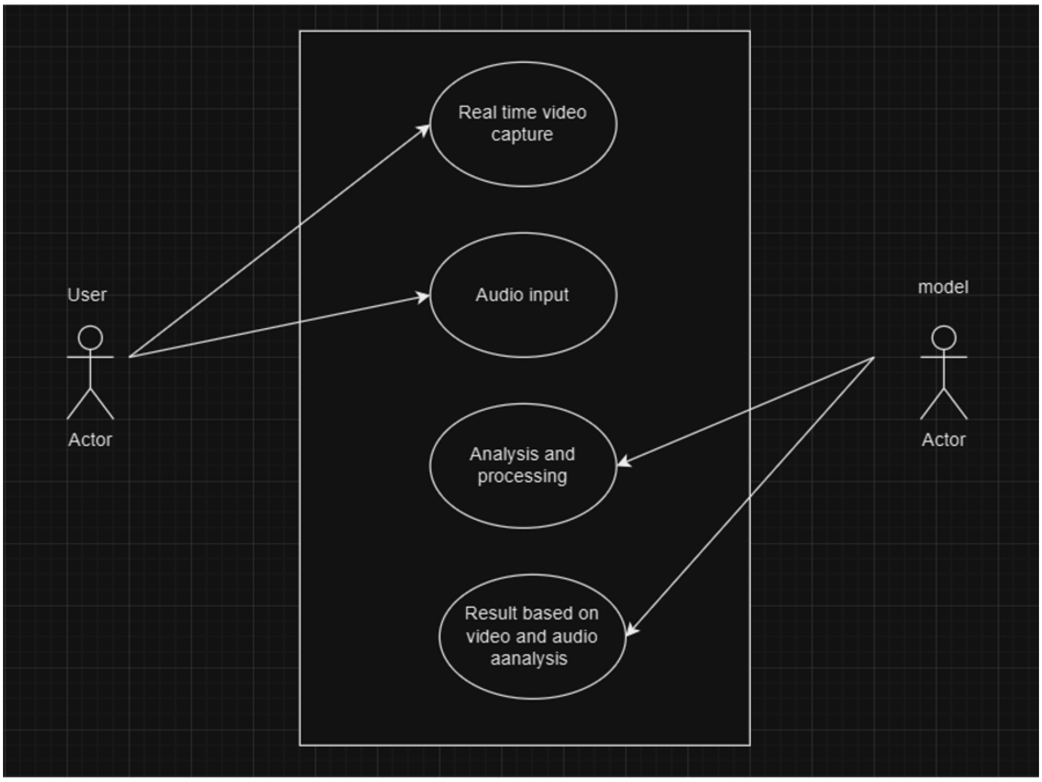


Figure 3.4: Use Flow of the Mock Interview System

- **Moving Average for Emotions**: Calculating a moving average stabilizes fluctuations in momentary emotions.

- **Dynamic Movement Threshold**: Eye movement thresholds are dynamically set according to video resolution, allowing for consistent results across devices.

CHAPTER 4

IMPLEMENTATION

This chapter details the implementation of the Mock Interview Application.

4.1 Resume Parsing

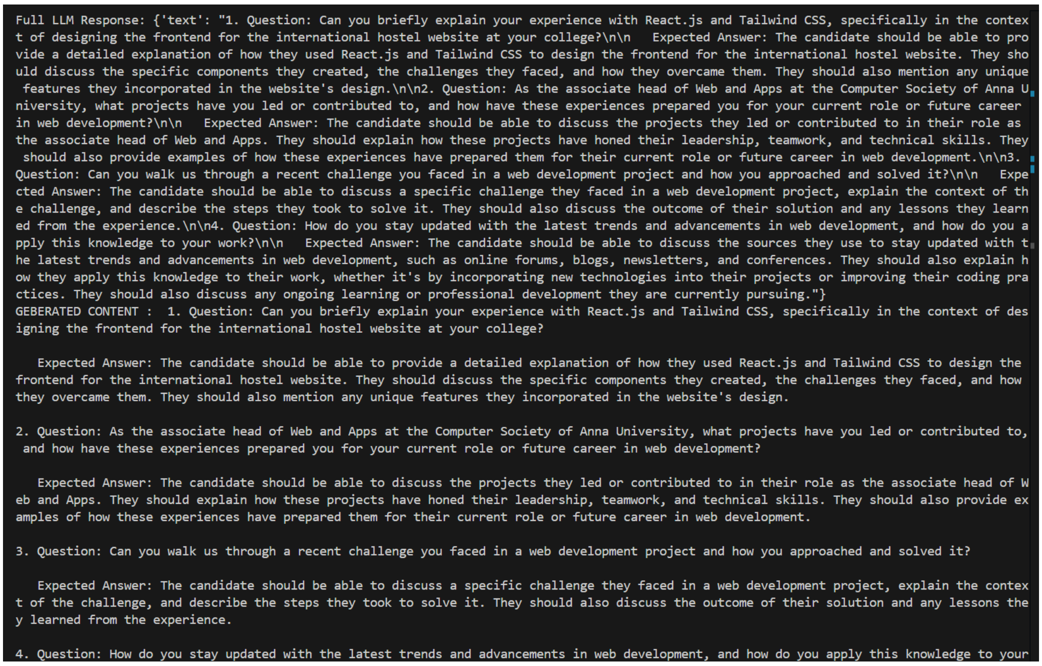
The application begins by processing the candidate's resume to extract essential details, which are then used to generate personalized interview questions.

4.1.1 Extracting Text

The uploaded PDF resume is parsed using the PyPDF library to extract text content. The text is organized and cleaned, preparing it for further processing.

4.1.2 Summarization

The Hugging Face model `facebook/bart-large-cnn` is employed to summarize the extracted text, highlighting key points such as projects, skills, and achievements. This summary, focusing on relevant details, is the basis for generating customized interview questions.



```
Full LLM Response: {'text': "1. Question: Can you briefly explain your experience with React.js and Tailwind CSS, specifically in the contex
t of designing the frontend for the international hostel website at your college?\n\n   Expected Answer: The candidate should be able to pro
vide a detailed explanation of how they used React.js and Tailwind CSS to design the frontend for the international hostel website. They sho
uld discuss the specific components they created, the challenges they faced, and how they overcame them. They should also mention any unique
 features they incorporated in the website's design.\n\n2. Question: As the associate head of Web and Apps at the Computer Society of Anna U
niversity, what projects have you led or contributed to, and how have these experiences prepared you for your current role or future career
in web development?\n\n   Expected Answer: The candidate should be able to discuss the projects they led or contributed to in their role as
the associate head of Web and Apps. They should explain how these projects have honed their leadership, teamwork, and technical skills. They
 should also provide examples of how these experiences have prepared them for their current role or future career in web development.\n\n3.
Question: Can you walk us through a recent challenge you faced in a web development project and how you approached and solved it?\n\n   Expe
cted Answer: The candidate should be able to discuss a specific challenge they faced in a web development project, explain the context of th
e challenge, and describe the steps they took to solve it. They should also discuss the outcome of their solution and any lessons they learn
ed from the experience.\n\n4. Question: How do you stay updated with the latest trends and advancements in web development, and how do you a
pply this knowledge to your work?\n\n   Expected Answer: The candidate should be able to discuss the sources they use to stay updated with t
he latest trends and advancements in web development, such as online forums, blogs, newsletters, and conferences. They should also explain h
ow they apply this knowledge to their work, whether it's by incorporating new technologies into their projects or improving their coding pra
ctices. They should also discuss any ongoing learning or professional development they are currently pursuing."}
GEBERATED CONTENT :  1. Question: Can you briefly explain your experience with React.js and Tailwind CSS, specifically in the context of des
igning the frontend for the international hostel website at your college?

   Expected Answer: The candidate should be able to provide a detailed explanation of how they used React.js and Tailwind CSS to design the
frontend for the international hostel website. They should discuss the specific components they created, the challenges they faced, and how
they overcame them. They should also mention any unique features they incorporated in the website's design.

2. Question: As the associate head of Web and Apps at the Computer Society of Anna University, what projects have you led or contributed to,
 and how have these experiences prepared you for your current role or future career in web development?

   Expected Answer: The candidate should be able to discuss the projects they led or contributed to in their role as the associate head of W
eb and Apps. They should explain how these projects have honed their leadership, teamwork, and technical skills. They should also provide ex
amples of how these experiences have prepared them for their current role or future career in web development.

3. Question: Can you walk us through a recent challenge you faced in a web development project and how you approached and solved it?

   Expected Answer: The candidate should be able to discuss a specific challenge they faced in a web development project, explain the contex
t of the challenge, and describe the steps they took to solve it. They should also discuss the outcome of their solution and any lessons the
y learned from the experience.

4. Question: How do you stay updated with the latest trends and advancements in web development, and how do you apply this knowledge to your
```

Figure 4.1: LLM Response Generation and Similarity Scoring

4.2
Question Generation

Based on the summarized resume content, relevant questions are generated, offering a customized question set tailored to the candidate's background.

4.2.1 Utilizing Summarized Content

Using the summarized text, the Hugging Face model `mistralai/Mistral-7B-Instruct-v0.3` generates interview questions based on the candidate's past projects, skills, and achievements.

4.2.2 Question Streams

- **Resume-based Questions:** These questions focus on the

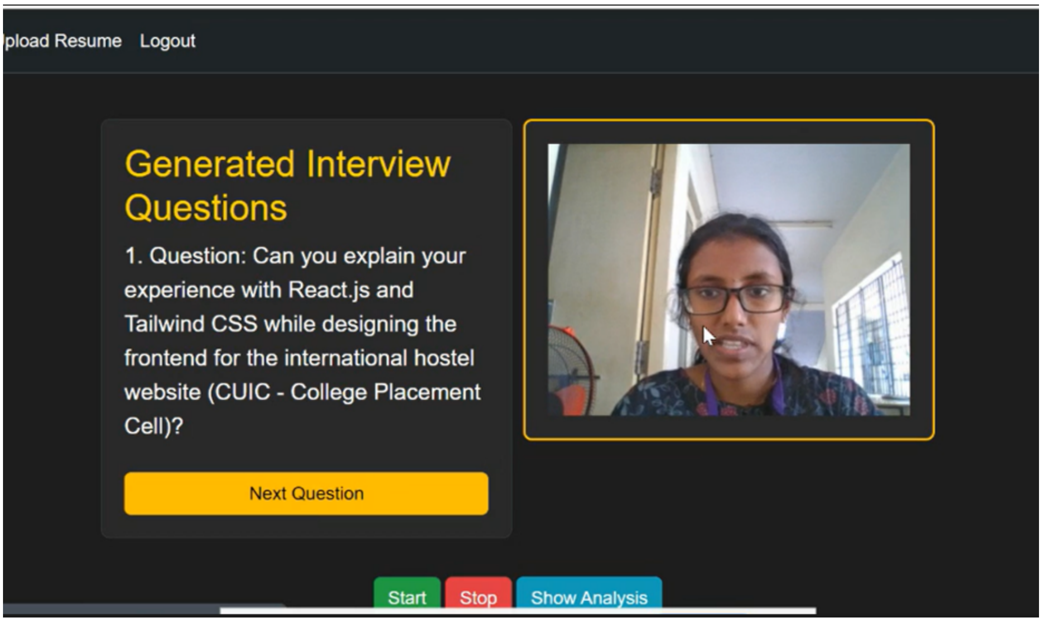


Figure 4.2: Question Generation and OpenCV Face Detection

projects, achievements, and skills mentioned in the resume, allowing assessment of the candidate's familiarity and depth in these areas.

- **Subject-based Questions:** Questions are generated to evaluate foundational knowledge in relevant technical fields.

subsubsection **4.2.2.0 Web Scraping**

Leveraging Beautiful Soup, the application scrapes common technical topics (e.g., OOP, Java) from online sources. This scraped content is used to generate subject-specific questions that assess the candidate's knowledge of core technical concepts.

4.3
Video Analysis

The video analysis module evaluates the candidate's non-verbal

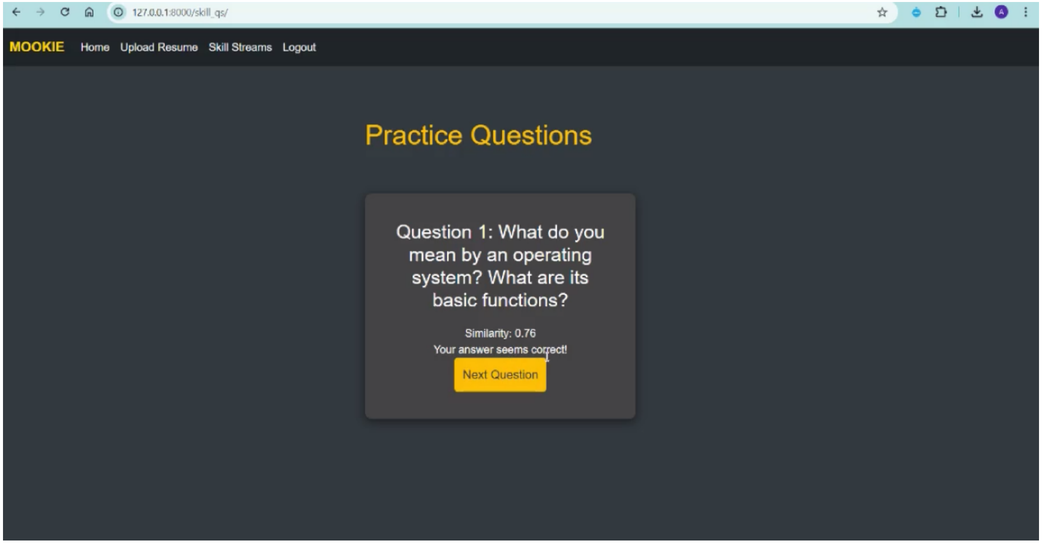


Figure 4.3: Sample output showing evaluation of a correct answer.

cues during the interview, focusing on eye movements and emotional expressions to assess the candidate's confidence.

4.3.1 Eye Movement Detection

Video frames are captured using OpenCV, with the dlib library used to identify and track eye regions. Dlib's facial landmark detection locates and analyzes eye movements. By tracking the pupil positions and counting shifts, the system detects whether the candidate is focused or distracted. Fewer eye movements indicate higher focus and confidence, while excessive movement suggests nervousness.This is detailed in algorithm 4.1

4.3.2 Emotion Detection

The `fer` (Facial Expression Recognition) library is used to detect a range of emotions such as happiness, sadness, anger, surprise, and

Figure 4.4: Eye Movement Detection using Dlib

Algorithm 4.1 Eye Movement Detection Algorithm

Input: Frame sequences with detected eye regions
Output: Left and Right Eye Movement Counts
Algorithm
Initialize: Set `prev_left_eye_center`, `prev_right_eye_center` to `None`
`current_time` ← Current time in seconds
if `prev_left_eye_center` and `prev_right_eye_center` are not `None` **then**
    `left_eye_movement` ← Absolute difference in X-coordinates of `left_eye_center` and `prev_left_eye_center`
    `right_eye_movement` ← Absolute difference in X-coordinates of `right_eye_center` and `prev_right_eye_center`
    if `left_eye_movement` > `MOVEMENT_THRESHOLD` and `current_time` - `last_left_move_time` > `TIME_THRESHOLD` **then**
        if `left_eye_center[0]` < `prev_left_eye_center[0]` **then**
            Increment `left_eye_moves`
        else
            Increment `right_eye_moves`
        end if
        Update `last_left_move_time` to `current_time`
    end if
    if `right_eye_movement` > `MOVEMENT_THRESHOLD` and `current_time` - `last_right_move_time` > `TIME_THRESHOLD` **then**
        if `right_eye_center[0]` < `prev_right_eye_center[0]` **then**
            Increment `left_eye_moves`
        else
            Increment `right_eye_moves`
        end if
        Update `last_right_move_time` to `current_time`
    end if
end if

```
Eye moves left: 0, Eye moves right: 0
Eye moves left: 0, Eye moves right: 0
Eye moves left: 0, Eye moves right: 0
Eye moves left: 0, Eye moves right: 0
Eye moves left: 0, Eye moves right: 0
Eye moves left: 0, Eye moves right: 0
Eye moves left: 0, Eye moves right: 0
Eye moves left: 0, Eye moves right: 0
Eye moves left: 0, Eye moves right: 0
Eye moves left: 0, Eye moves right: 0
Eye moves left: 0, Eye moves right: 0
Eye moves left: 0, Eye moves right: 0
Eye moves left: 0, Eye moves right: 0
Eye moves left: 2, Eye moves right: 0
Eye moves left: 2, Eye moves right: 0
Eye moves left: 2, Eye moves right: 0
Eye moves left: 2, Eye moves right: 0
Eye moves left: 2, Eye moves right: 0
Eye moves left: 2, Eye moves right: 0
Eye moves left: 2, Eye moves right: 0
Eye moves left: 2, Eye moves right: 0
Eye moves left: 2, Eye moves right: 0
Eye moves left: 2, Eye moves right: 0
Eye moves left: 2, Eye moves right: 0
Eye moves left: 2, Eye moves right: 0
Eye moves left: 2, Eye moves right: 0
Eye moves left: 2, Eye moves right: 0
```

Figure 4.5: Eye Movement tracking

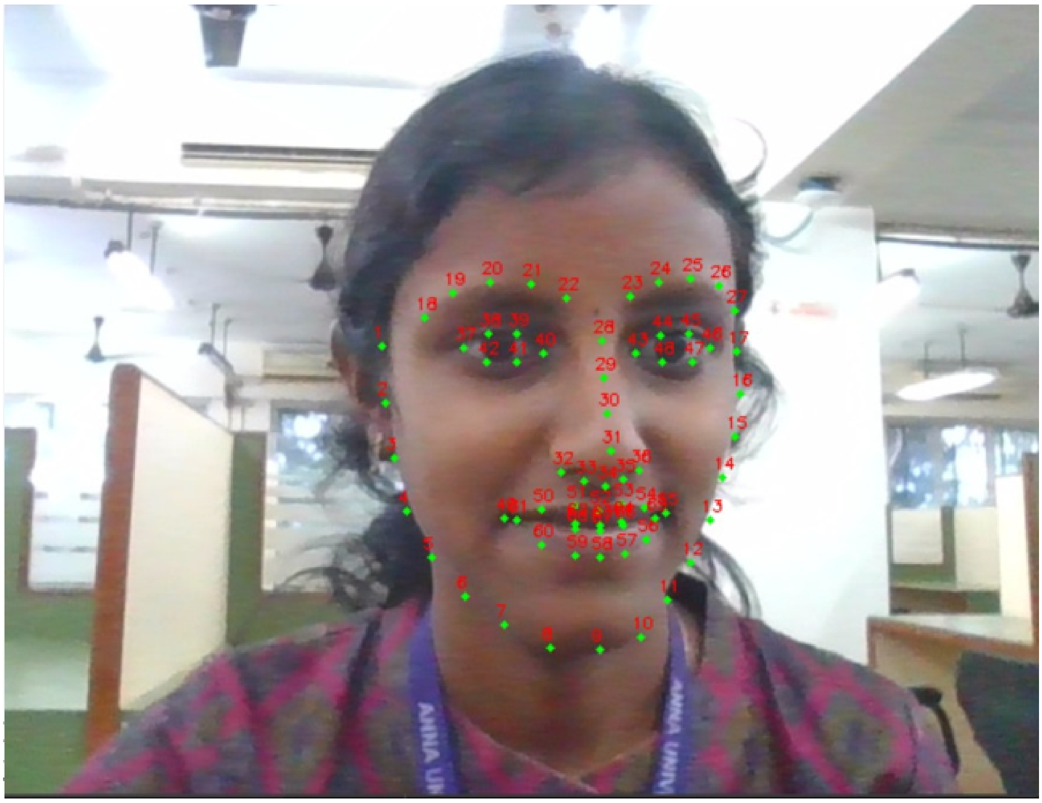


Figure 4.6: Face Detection using Dlib for Emotion Analysis

neutrality. A rolling count of detected emotions is maintained to determine the primary emotion expressed by the candidate during different parts of the interview.

4.4
Audio Analysis

The audio analysis module transcribes and evaluates the candidate's spoken answers.

4.4.1 Speech Recognition

The `SpeechRecognition` library captures and transcribes audio input, converting the candidate's spoken answers into text in real time for further analysis.

4.4.2 Answer Similarity Scoring

The Hugging Face model `sentence-transformers/paraphrase-MiniLM-L6-v2` evaluates the similarity between the candidate's answers and expected correct responses. A similarity score is calculated based on semantic content; higher scores indicate accurate and relevant responses, while lower scores suggest a lack of depth or correctness in the answer.

4.5
Flow Summary

- **Resume Processing:** The resume is parsed and summarized to generate customized questions.
- **Video and Audio Capture:** The system captures and analyzes video for non-verbal cues and audio for spoken content.
- **Answer Evaluation:** Answers are compared to ideal responses for semantic accuracy.
- **Final Assessment:** Insights from text, video, and audio analyses are compiled, providing a comprehensive performance evaluation of the candidate.

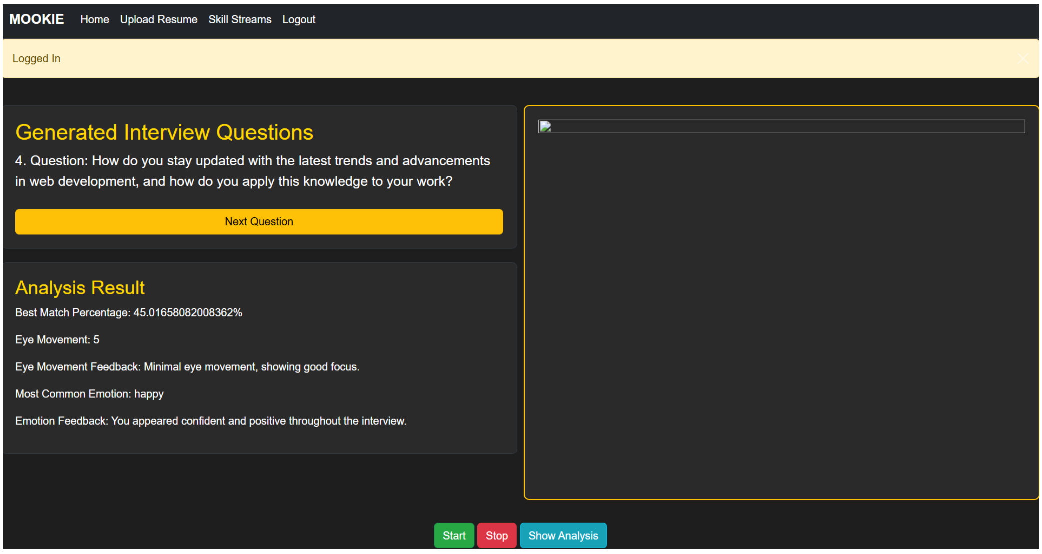


Figure 4.7: Video Analysis Result - Detected Emotion: Happy

CHAPTER 5

RESULTS AND ANALYSIS

The mock interview system successfully integrates a variety of AI and computer vision technologies to deliver a comprehensive interview assessment, providing targeted insights into both technical knowledge and behavioural attributes. Each stage of the system's workflow, from resume parsing and summarization to question generation, live video analysis, and speech recognition, operated effectively under test conditions, delivering detailed and accurate performance feedback for users.

5.1
Video Analysis

The video analysis component successfully captured real-time visual data of user interactions, focusing on eye movement and emotional expressions. Using ‘OpenCV‘ for video capture and ‘dlib‘ for facial landmark detection, the system was able to accurately identify key facial features, with particular emphasis on the eyes, to analyze attentiveness and engagement levels. Additionally, the FER (Facial Expression Recognition) library detected emotions such as happiness, sadness, and nervousness by categorizing facial expressions in real-time. This allowed for a nuanced view of the user's emotional state, providing valuable insights into non-verbal cues during the mock interview. The integration of these visual metrics contributed to a comprehensive assessment of user confidence and emotional consistency, supporting a behaviour-oriented evaluation in the feedback system.

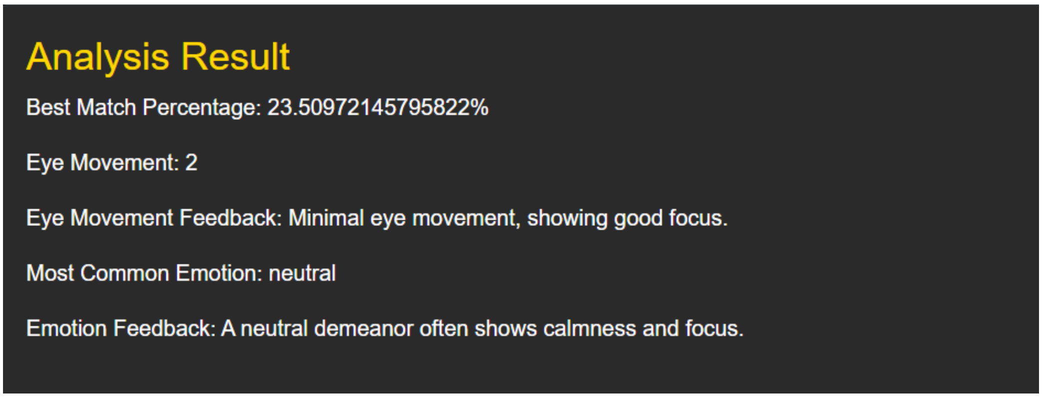


Figure 5.1: Real-time video analysis output showing neutral facial expression detection.

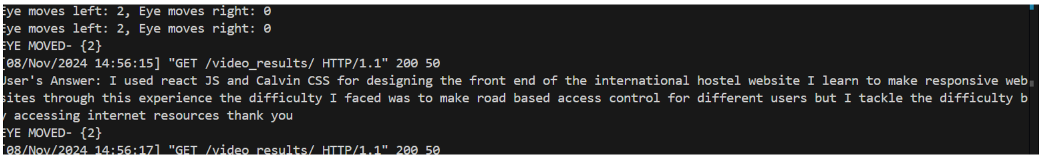


Figure 5.2: Console output showing speech-to-text detection and transcription results.

5.2 Audio Analysis

The audio analysis component performed effectively in converting speech to text and evaluating the accuracy of spoken answers. Google's Speech-to-Text API reliably transcribed users' verbal responses, maintaining high accuracy across various accents and speech patterns. The 'paraphrase-MiniLM-L6-v2' model compared the transcribed answers to the ideal responses generated by the question and answer component, yielding a similarity score that represented how closely the user's answers aligned with expected answers. This similarity score provided a quantitative basis for evaluating technical accuracy, while the transcriptions themselves enabled detailed feedback on content clarity and relevance. Overall, the audio analysis not only captured the user's verbal responses with precision but also facilitated a deeper understanding of their alignment with technical expectations.

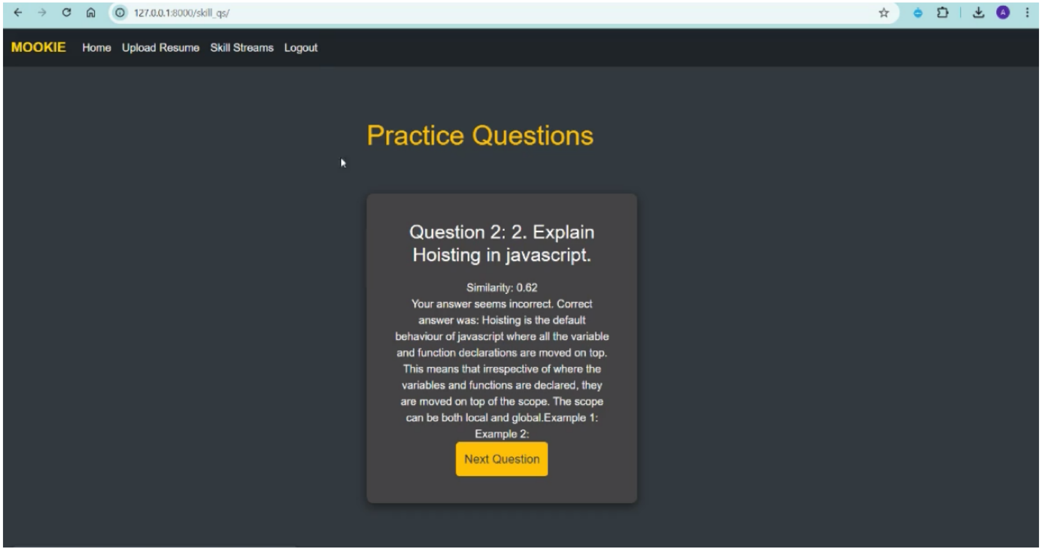


Figure 5.3: Sample output showing evaluation of an incorrect answer.

To illustrate correct and incorrect answer assessments, the following images show system feedback based on user responses:

5.3
Comparison of similarity Models

We compared two models for sentence similarity: BERT for resume-based questions and MiniLM-L6-v2 for subject-based questions. Each model was evaluated based on accuracy, speed, and resource usage, with the results summarized in Figure red5.4.

BERT: Achieved high accuracy, making it suitable for nuanced resume-based answers. However, it showed higher resource usage and slower speed, highlighting a trade-off between precision and efficiency. BERT is computationally intensive, requiring more memory and processing power, which affected its response time.

MiniLM-L6-v2: While slightly less accurate than BERT, MiniLM-L6-v2 performed well for general subject-based questions, which

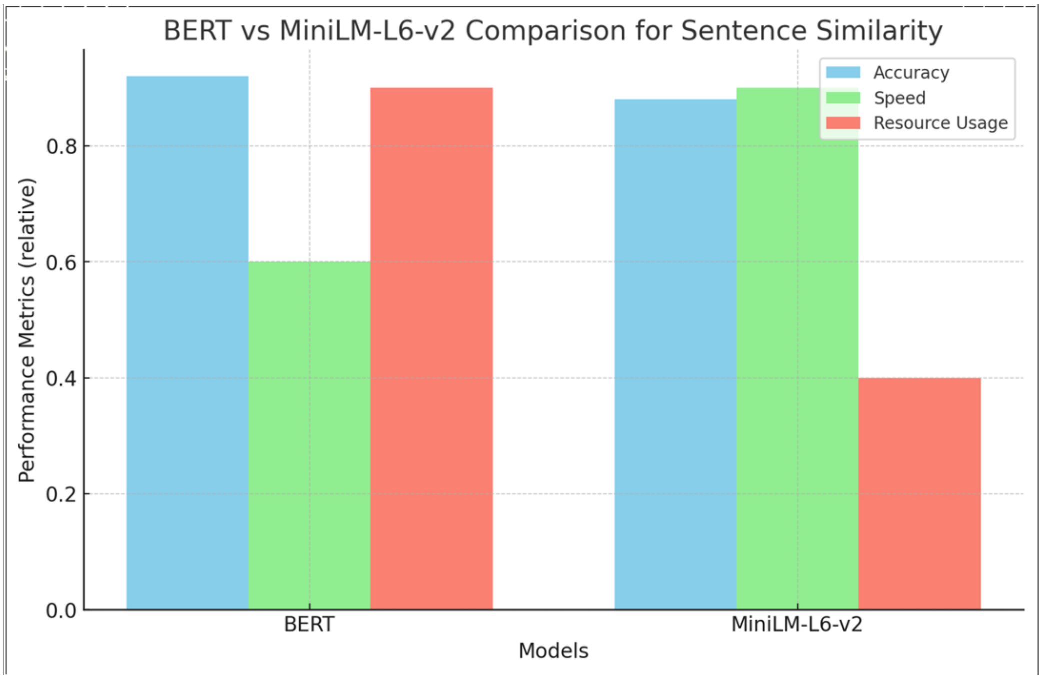


Figure 5.4: Comparison of BERT and MiniLM-L6-v2 models in terms of accuracy, speed, and resource usage for sentence similarity tasks.

required less nuanced understanding. It demonstrated faster response times and lower resource usage, making it ideal for real-time applications. This lightweight model balanced efficiency with acceptable accuracy for straightforward questions.

In summary, the combined use of BERT and MiniLM-L6-v2 allowed the system to leverage both detailed analysis and efficient response, depending on the complexity and requirements of each question type.

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 CONCLUSION

The creation of an AI-powered mock interview preparation system represents an exciting step forward in how we can assist candidates in preparing for real-world interviews. By combining advanced technologies like emotion recognition, eye gaze tracking, and Large Language Models (LLMs), this system offers a more comprehensive approach to interview preparation. It not only helps candidates improve their answers but also fine-tunes their delivery, helping them become more aware of their body language and emotional tone—elements that can often make or break an interview.

One of the core strengths of this project is its practicality. The system makes it easy for anyone to practice interviews, whether they are experienced professionals or those new to the job market. By using AI to generate questions tailored to each individual's resume, the system adapts to different job roles, ensuring users are always practicing with the right content. This level of personalization makes the system valuable for users across various industries.

In the end, the project highlights the potential of AI to transform traditional interview preparation. It shows that, with the right tools, interview practice can be more interactive and effective. By integrating real-time video analysis and AI-driven feedback, candidates can identify areas for improvement and approach interviews with greater confidence. With further advancements in technology, this system can become an even more powerful tool, helping candidates everywhere refine their skills and stand out in the job market.

6.2
FUTURE WORK

Looking ahead, there are several exciting opportunities to improve and expand this system.

First, the emotion recognition and gaze tracking models can be enhanced by adding more diverse datasets. This will help the system recognize a wider range of emotional expressions, improve accuracy in different settings, and adapt to cultural differences in body language. More varied data will help the system provide more detailed and nuanced feedback.

The question generation process can also benefit from integrating newer, more advanced language models. As language models continue to improve, they will become better at understanding the context of the user's resume and generating more specific, relevant questions. This will further personalize the experience, making it more useful for users in various job sectors.

In addition, improving the real-time feedback functionality will make the system even more effective. For example, integrating edge computing could allow the system to process and deliver feedback faster, ensuring that the candidate receives immediate suggestions on how to adjust their answers or body language during the mock interview.

Another promising direction is to make the system more accessible. Currently, it's available for desktop users, but expanding it to mobile and web platforms would give users more flexibility. By creating a web-based version or mobile app, candidates could practice interviews anytime, anywhere, making the system even more valuable for people on the go or those in locations without easy access to in-person interview coaching.

Finally, building a progress tracking system would be a great addition. This feature would allow users to monitor their improvement over time, receive targeted feedback based on past performance, and adapt their practice sessions to focus on areas that need the most attention. By keeping track of user progress, the system can evolve into a continuous learning tool that supports candidates throughout their entire job search journey.

In conclusion, this project has the potential to make a real impact on how people prepare for interviews. By incorporating AI in meaningful ways, the system is poised to help candidates improve their interview skills more efficiently and effectively than traditional methods. As AI technology advances, the system will continue to evolve, offering even more personalized, real-time feedback, and ultimately helping users become more confident and successful in their job interviews.

REFERENCES

[1]. King, D. E., ”Dlib-ml: A Machine Learning Toolkit.” Journal of Machine Learning Research.

[2]. S. Zheng et al. (2021): Use of MT-CNN for facial detection in real-time scenarios.

[3]. Reimers, N., and Gurevych, I. ”Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks.” EMNLP-IJCNLP.

[4]. Li et al. (2020): Research on contextual question generation from resumes and unstructured text.

[5]. Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K., ”BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding,” In Proceedings of NAACL-HLT, 2019.

[6]. Lewis, M., Liu, Y., Goyal, N., et al., ”BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension,” In Proceedings of ACL, 2020.

[7]. Raffel, C., Shazeer, N., Roberts, A., et al., ”Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer,” Journal of Machine Learning Research, 2020.

[8]. Cao, Q., Shen, L., Xie, W., Parkhi, O. M., Zisserman, A., ”VGGFace2: A Dataset for Recognising Faces Across Pose and Age,” In Proceedings of the International Conference on Automatic Face and Gesture Recognition, 2018.

[9]. Eyben, F., Wöllmer, M., and Schuller, B., ”Opensmile: The Munich Versatile and Fast Open-source Audio Feature Extractor,” In Proceedings of the ACM International Conference on Multimedia, 2010.

[10]. Park, J. H., Shin, J. (2021). ”A Study on Interview Feedback and Assessment Using Artificial Intelligence,” IEEE Access.

[11]. Deng, J., Guo, J., Xue, N., and Zafeiriou, S., ”ArcFace: Additive Angular Margin Loss for Deep Face Recognition,” In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[12]. Hinton, G., Vinyals, O., Dean, J., ”Distilling the Knowledge in a Neural Network,” In Proceedings of NeurIPS Deep Learning and Representation Learning Workshop, 2015.

[13]. Elizalde, B., Pellegrino, E., and Shriberg, E., ”Predicting Speech and Facial Expression Based Emotion in Job Interviews,” In Proceedings of the IEEE Spoken Language Technology Workshop, 2016.

[15]. He, K., Zhang, X., Ren, S., Sun, J., ”Deep Residual Learning for Image Recognition,” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016.

REFERENCES

10

[1] Yi-Ting Chen Ashish Tawari John Canny Jinkyu Kim, Teruhisa Misu. Grounding human-to-vehicle advice for self-driving vehicles. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[2] Trevor Darrell John Canny Jinkyu Kim, Anna Rohrbach and Zeynep Akata. Textual explanations for self-driving vehicles. *European Conference on Computer Vision(ECCV)*, 2018.

[3] Daocheng Fu, Xin Li, Licheng Wen, Min Dou, Pinlong Cai, Botian Shi, and Yu Qiao. Drive like a human: Rethinking autonomous driving with large language models. *arXiv preprint arXiv:2307.07162*, 2023.

[4] Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kenneth KY Wong, Zhenguo Li, and Hengshuang Zhao. Drivegpt4: Interpretable end-to-end autonomous driving via large language model. *arXiv preprint arXiv:2310.01412*, 2023.

[5] Licheng Wen, Daocheng Fu, Xin Li, Xinyu Cai, Tao Ma, Pinlong Cai, Min Dou, Botian Shi, Liang He, and Yu Qiao. Dilu: A knowledge-driven approach to autonomous driving with large language models. *arXiv preprint arXiv:2309.16292*, 2023.

[6] Long Chen, Oleg Sinavski, Jan Hünermann, Alice Karnsund, Andrew James Willmott, Danny Birch, Daniel Maund, and Jamie Shotton. Driving with llms: Fusing object-level vector modality for explainable autonomous driving. *arXiv preprint arXiv:2310.01957*, 2023.

[7] Long Chen, Yuchen Li, Chao Huang, Yang Xing, Daxin Tian, Li Li, Zhongxu Hu, Siyu Teng, Chen Lv, Jinjun Wang, et al. Milestones in autonomous driving and intelligent vehicles—part 1: Control, computing system design, communication, hd map, testing, and human behaviors. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2023.

[8] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[9] Urs Muller, Jan Ben, Eric Cosatto, Beat Flepp, and Yann Cun. Off-road obstacle avoidance through end-to-end learning. *Advances in neural information processing systems*, 18, 2005.

[10] Bustanul Arifin, Bhakti Yudho Suprapto, Sri Arttini Dwi Prasetyowati, and Zainuddin Nawawi. Steering control in electric power steering autonomous vehicle using type-2 fuzzy logic control and pi control. *World electric vehicle journal*, 13(3):53, 2022.

[11] Mingxuan Hu, Yajun Pang, and Long Gao. Advances in silicon-based integrated lidar. *Sensors*, 23(13):5920, 2023.

[12] Marcelo Pereira, David Silva, Vitor Santos, and Paulo Dias. Self calibration of multiple lidars and cameras on autonomous vehicles. *Robotics and Autonomous Systems*, 83:326–337, 2016.

[13] Igal Bilik, Oren Longman, Shahar Villeval, and Joseph Tabrikian. The rise of radar for autonomous vehicles: Signal processing solutions and future research directions. *IEEE signal processing Magazine*, 36(5):20–31, 2019.

[14] Niels Joubert, Tyler GR Reid, and Fergus Noble. Developments in modern gnss and its impact on autonomous vehicle architectures. In *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 2029–2036. IEEE, 2020.

[15] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[16] Aram Bahrini, Mohammadsadra Khamoshifar, Hossein Abbasimehr, Robert J Riggs, Maryam Esmaeili, Rastin Mastali Majdabadkohne, and Morteza Pasehvar. Chatgpt: Applications, opportunities, and threats. In *2023 Systems and Information Engineering Design Symposium (SIEDS)*, pages 274–279. IEEE, 2023.

[17] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[18] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama:

Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[19] Bu Jin, Xinyu Liu, Yupeng Zheng, Pengfei Li, Hao Zhao, Tong Zhang, Yuhang Zheng, Guyue Zhou, and Jingjing Liu. Adapt: Action-aware driving caption transformer. *arXiv preprint arXiv:2302.00673*, 2023.

[20] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17853–17862, 2023.