Virtual Interview Simulator

Major Project Report

submitted in partial fulfilment of the requirements
for award of the degree of

BACHELOR OF TECHNOLOGY
by

Aditya Gattani: (21BEC010)
Tusshar Paul: (21BEC082)
Anand Mohan Arya: (21BEC091)
Vineet Mittal: (21BEC002)

Under the guidance of
Dr. Manoranjan Rai Bharti
Associate Professor
And
Dr. Anshu Thakur
Assistant Professor

Department of Electronics and Communication Engineering
National Institute of Technology Hamirpur
Hamirpur, Himachal Pradesh - 177005, India
December 2024

राष्ट्रीय प्रौद्योगिकी संस्थान हमीरपुर
हमीरपुर (हि॰प्र॰) – 177 005 (भारत)
NATIONAL INSTITUTE OF TECHNOLOGY HAMIRPUR
HAMIRPUR (H.P.) - 177 005 (INDIA)
(An Institute of National Importance under Ministry of Education (Shiksha Mantralaya)

Candidates' Declaration

We hereby declare that the work which is being presented in the project report titled **Virtual Interview Simulator** in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology and submitted in the **Department of Electronics and Communication Engineering**, National Institute of Technology Hamirpur, is an authentic record of our own work carried out during a period from **July 2024 – December 2024** under the guidance of **Dr. Manoranjan Rai Bharti, Associate Professor and Dr. Anshu Thakur**, Assistant professor of Electronics and Communication Engineering, National Institute of Technology Hamirpur.

The matter presented in this project report has not been submitted by us for the award of any other degree of this or any other Institute/University.

Aditya Gattani: 21BEC010
Tusshar Paul: 21BEC082
Anand Mohan Arya: 21BEC091
Vineet Mittal: 21BEC002

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Dr. Anshu Thakur
Project Mentor
Dept. of ECE
NIT Hamirpur

Signature .....

Date : .....

.....
Signature of Head of Department

Acknowledgement

We would like to express our sincere gratitude to Dr. Manoranjan Rai Bharti, Assistant Professor at the Department of Electronics and Communications Engineering, National Institute of Technology (NIT) Hamirpur, for his invaluable guidance and unwavering support throughout the course of our major project. His deep expertise, coupled with his dedication to mentoring, provided us with the direction and clarity needed to successfully complete this project. Dr. Bharti's constant encouragement, insightful feedback, and constructive criticism pushed us to explore new dimensions and improve upon our initial designs. His timely suggestions and patience throughout the research and development phases played a pivotal role in shaping our project and elevating its quality. We are extremely fortunate to have had the opportunity to work under his guidance, and we are deeply grateful for his contribution to our academic growth.

We owe a deep sense of gratitude to the Department of Electronics and Communication Engineering and the National Institute of Technology, Hamirpur which provided us with access to the necessary equipment for the fulfillment of the major project within due course.

Lastly, this project would not have been possible without the work and assistance of our group members. The commitment of each team member was essential for the success of this endeavor. We would like to thank everyone who provided advice, recommendations, or assistance during the duration of this project and thereafter.

Aditya Gattani : 21BEC010
Tusshar Paul : 21BEC082
Anand Mohan Arya : 21BEC091
Vineet Mittal : 21BEC002

Abstract

Interviewing can be a nerve-wracking experience, especially when faced with unpredictable questions and the dynamic back-and-forth of a real interview. To help individuals prepare effectively, our platform introduces a **phone screen agent** that creates an interactive and realistic practice environment tailored specifically to the role the user is pursuing.

By simply providing a resume and job details, users can engage with an AI-powered interviewer who role plays based on their background. This AI interviewer not only asks personalized questions but also listens to responses and offers intelligent follow-ups, replicating the flow of a genuine interview.

What sets this tool apart is its ability to adopt multiple distinct personas, such as a fast-paced Wall Street professional or a creative director. This versatility allows users to prepare for a wide range of interview styles and scenarios, helping them handle unexpected curveballs with confidence. Unlike scripted practice sessions, this dynamic interaction forces users to think critically, articulate their thoughts clearly, and adapt in real-time.

By practicing in a low pressure environment, users can refine their skills, improve their fluency, and walk into actual interviews feeling calm, prepared, and ready to make a positive impression.

In addition to communication skills, success in securing positions at top companies requires a solid foundation in coding. However, many aspirants with strong technical skills struggle to articulate their solutions effectively, which can be a significant barrier to achieving their goals. Our platform addresses this challenge by combining coding and communication practice into a seamless experience.

The platform allows users to solve coding problems from a diverse set of frequently asked questions. Once the user submits their solution, the AI model generates follow-up questions related to the code they've written, challenging their understanding and reasoning. In addition, it asks staple questions commonly encountered in interviews, ensuring comprehensive preparation.

At the end of the session, users receive a detailed interview profile that highlights their strengths and areas for improvement. This includes insights into their verbal fluency, ability to handle questions under pressure, and overall performance. The platform also tracks progress over time, providing a growth curve that helps users evaluate their improvement and build confidence as they prepare for real interviews.

By offering a realistic, adaptive, and multifaceted preparation tool, our platform empowers aspirants to excel in both the technical and behavioral aspects of interviews. Whether practicing for technical roles or honing soft skills, users can gain a significant edge, ensuring they are ready to succeed in any professional environment.

Contents

List of Figures

VIRTUAL INTERVIEW SIMULATOR

Department of Electronics and Communication Engineering
National Institute of Technology Hamirpur

1 Introduction

1.1 Project background

In today's highly competitive job market, effective preparation is paramount to securing desirable roles in top organizations. The interview process, a critical step in recruitment, is often fraught with challenges such as performance anxiety, lack of familiarity with diverse interview formats, and an inability to effectively articulate thoughts under pressure. To address these challenges, the Virtual Interview Simulator emerges as a cutting-edge solution that combines the power of technology, artificial intelligence, and educational psychology to create a comprehensive interview preparation platform.

The Virtual Interview Simulator is designed to provide an immersive and interactive experience for aspirants, enabling them to practice and refine their skills in a controlled, risk free environment. By simulating real-world interview scenarios, the tool bridges the gap between theoretical knowledge and practical execution, ensuring candidates are well-prepared to face even the most demanding interview panels. This platform goes beyond generic preparation methods by incorporating domain-specific questions, role-specific challenges, and personalized feedback, catering to the unique needs of each user.

1.2 The Need for Advanced Interview Preparation Tools

Modern interviews have evolved to assess not only technical proficiency but also soft skills, critical thinking, and cultural fit. Companies employ diverse formats, including behavioral interviews, technical problem-solving sessions, case studies, and stress interviews, making it essential for candidates to be adaptable and versatile. Traditional preparation methods, such as studying commonly asked questions or attending mock interviews, often fall short of equipping candidates with the skills to navigate such dynamic scenarios effectively.

Furthermore, the increasing reliance on virtual and remote interviews necessitates familiarity with digital communication tools and an ability to present oneself confidently over video calls. Many candidates face challenges in adapting to these formats, which can result in subpar performances despite possessing the required skills.

The Virtual Interview Simulator addresses these gaps by offering targeted, realistic simulations tailored to modern interview processes. The Virtual Interview Simulator is not just a preparation tool—it is a transformative platform designed to empower users to excel in interviews and unlock their full potential[1].

By leveraging state-of-the-art technology and focusing on the holistic development of candidates, the simulator addresses the evolving demands of the job market. Whether for individual users, educational institutions, or corporate training programs, this innovative solution redefines how we approach interview readiness in the modern era.

1.3 The Evolving Landscape of Interview Preparation

Interviews have transformed significantly over the past decade, reflecting the changing needs of employers and industries. Modern interview processes often consist of multiple stages, including behavioral assessments, technical problem-solving, situational role-plays, and stress interviews. In addition, the rise of remote work and virtual hiring processes has introduced new challenges, such as presenting oneself confidently on video platforms and navigating technical issues.

These changes have made traditional preparation methods increasingly insufficient. For instance:

1. Lack of Real-World Exposure Mock interviews, while useful, often lack the depth and unpredictability of actual interviews. Candidates may memorize responses to standard questions but struggle when faced with dynamic or unexpected queries[4].
2. Inadequate Soft Skills Training.
3. The Virtual Interview Simulator addresses these challenges by providing a versatile and accessible solution that prepares users for the complexities of modern interviews.
4. With the increasing use of virtual interviews and role-specific assessments, candidates must now be proficient with digital communication tools and possess the ability to adapt to various interviewer styles.
5. Changing Formats and Expectations
6. Access to high-quality preparation tools, such as professional coaching or mock interviews with industry experts, is often limited to those with significant financial resources.

1.4 Benefits

The Virtual Interview Simulator provides significant advantages for individuals, educational institutions, and corporate training programs

1. For Individual Users
    - Confidence Building: By practicing in a safe and supportive environment, users gain the confidence to perform effectively in real interviews.
    - Targeted Preparation: Personalized feedback ensures users focus on their unique weaknesses, maximizing their preparation efficiency.
    - Adaptability: Exposure to diverse scenarios and interviewer styles equips candidates to handle a wide range of challenges.

1. For Educational Institutions
    - Enhanced Career Services: Colleges and universities can integrate the simulator into their career development programs, helping students transition seamlessly from academia to industry.
    - Scalable Solution: The simulator allows institutions to prepare large cohorts of students without the need for extensive resources or personnel

1. For Corporate Training Programs

- Talent Development: Companies can use the simulator to train employees for internal promotions or cross-functional roles.
- Recruitment Tool: Organizations can employ the platform to conduct mock interview drives and evaluate potential hires in a structured manner.

1.5 Key Features

The Virtual Interview Simulator is built to replicate the nuances of real interviews while providing a supportive framework for learning and growth. Some of its standout features include:

1. Realistic Role-Specific Simulations :
   The simulator recreates scenarios tailored to specific roles, industries, and levels of expertise, ensuring users practice within a relevant context. From coding challenges for software developers to behavioral questions for managerial positions, the simulator spans a wide range of professional domains.
2. AI-Driven Interactions:
   Advanced AI algorithms mimic diverse interviewer personalities, including friendly, neutral, and challenging styles, enabling users to adapt to varied approaches. Dynamic question generation ensures that sessions remain unpredictable and engaging, closely resembling real-world interactions. Comprehensive Feedback and Analytics.
3. Post-interview feedback:
   highlights areas of improvement, such as verbal articulation, non-verbal cues, and content clarity. Detailed analytics track progress over time, helping users identify trends and focus on specific weaknesses[6].
4. Soft Skills Enhancement :
   Modules dedicated to improving communication, confidence, and stress management help users build a holistic skill set. Scenarios like handling difficult questions or navigating unexpected interruptions prepare candidates for real-time challenges. Accessibility and Convenience.

1.6 Advantages

Unlike traditional preparation methods, the Virtual Interview Simulator offers a scalable and cost-effective solution that caters to a wide range of users, from students and job seekers to professionals transitioning into new roles.

It democratizes access to high-quality interview preparation resources, ensuring that users from diverse backgrounds can compete on an equal footing. By providing a safe space to fail and learn, the simulator instills confidence and reduces the fear of judgment often associated with real interviews.

For organizations, the simulator can serve as a valuable tool for pre-hire assessments and internal training programs, fostering a more prepared and skilled workforce. Educational institutions can integrate it into their curricula to help students transition seamlessly from academia to industry.

2 Literature Review

2.1 Introduction

Preparing for job interviews is a critical step for candidates aiming to secure their desired roles. Despite the availability of various preparation methods, many candidates feel underprepared, especially when it comes to real-world scenarios. Traditional approaches often lack the depth and personalization required to effectively prepare candidates for the challenges of modern interviews.

Virtual Interview Simulators (VIS) address these shortcomings by providing a simulated interview environment that is both practical and interactive. These tools focus on the dual aspects of interviews-verbal and nonverbal communication. Verbal skills are critical for delivering clear and coherent answers, while nonverbal cues, such as facial expressions and body language, play a significant role in creating a positive impression.

The importance of role-specific preparation cannot be overstated. VIS uses advanced AI algorithms to generate tailored questions based on the candidate's chosen role, skills, and industry. This ensures that preparation is not generic but focused, providing candidates with the confidence to face job-specific challenges.

Moreover, nonverbal analysis, which is often overlooked in traditional methods, is integrated into the simulator. This allows candidates to receive feedback on their posture, eye contact, and gestures, helping them to refine their overall presentation. By providing dynamic, real-time feedback, VIS ensures that candidates are well-prepared for diverse interview scenarios.

With their accessibility and advanced features, Virtual Interview Simulators have transformed interview preparation from a rigid process into an engaging and highly effective experience. They empower candidates to practice anytime, anywhere, fostering both confidence and competence.

2.2 Related Work

Traditional Methods: Mock Interviews and Static Question Banks

The conventional approach to interview preparation typically involves mock interviews and static question banks. Mock interviews are conducted with peers or mentors to simulate the interview experience. However, these sessions are limited by the availability of mentors and often lack consistent, actionable feedback, particularly on nonverbal communication.

Static question banks, while useful, fall short of providing a comprehensive preparation experience. They offer a predefined set of questions that may not align with the candidate's specific job role or industry. Additionally, these methods lack interactivity, which is essential for adapting to the evolving nature of modern interviews

Advanced Techniques: Machine Learning in Interview Preparation

Some advanced methods have introduced machine learning (ML) tools to analyze historical interview data. These tools can identify patterns and common pitfalls, offering general rec-

ommendations to candidates. However, they remain impersonal and are unable to provide real-time interaction or focus on individual needs, which are crucial for effective preparation.

2.3 Proposed Approach: AI-Powered Virtual Interview Simulator

The Virtual Interview Simulator (VIS) represents a significant leap forward in interview preparation technology. By leveraging Artificial Intelligence (AI), this system provides a highly personalized and immersive experience. The simulator begins by gathering input from the candidate regarding their job role, skills, and industry. Using this data, it generates a tailored set of technical and behavioral questions.

The VIS employs Natural Language Processing (NLP) to evaluate verbal responses[7]. It focuses on key aspects such as clarity, coherence, and relevance. Simultaneously, computer vision analyzes nonverbal cues, including facial expressions, posture, and gestures, offering detailed feedback to improve the candidate's overall presentation.

To enhance learning, the simulator incorporates adaptive algorithms that adjust the difficulty level and question types based on the candidate's performance. This ensures steady improvement over time. Data augmentation techniques are also applied to expand the dataset, improving the accuracy and robustness of the system.

This innovative approach not only prepares candidates for the technical and behavioral aspects of interviews but also builds their confidence by simulating real-world scenarios. By providing real-time feedback and personalized recommendations, the Virtual Interview Simulator bridges the gap between traditional methods and the demands of modern interviews.

Additionally, the Virtual Interview Simulator integrates speech-to-text technology for accurate transcription of responses, enabling candidates to review their answers in detail. The system also supports multiple languages, catering to a diverse user base and ensuring accessibility across different regions.

Furthermore, the platform maintains a detailed progress tracker, allowing candidates to monitor their improvement over time and identify areas requiring additional focus. With a secure and user-friendly interface, the VIS ensures a seamless experience for candidates, from setup to feedback delivery.

By combining advanced AI techniques such as neural networks, deep learning, and reinforcement learning, the simulator continues to evolve, staying aligned with the latest industry trends and requirements. Its scalability also makes it suitable for both individual users and organizations looking to enhance their recruitment processes.

3 Technologies and Frameworks Used

3.1 HTML

HTML stands for Hyper Text Mark-up Language. It is used to design web pages using markup language. HTML is the combination of Hypertext and Mark-up language. Hypertext defines the link between the web pages. Mark-up language is used to define the text document within the tag which defines the structure of web pages. HTML5 is the fifth and current version of HTML. It has improved the markup available for documents and has introduced application programming interfaces (API) and Document Object Model (DOM).

3.2 CSS

Cascading Style Sheets, fondly referred to as CSS, is a simply designed language intended to simplify the process of making web pages presentable. CSS allows you to apply styles to web pages. More importantly, CSS enables you to do this independent of the HTML that makes up each web page.
CSS is essential for creating visually appealing and maintainable web pages. It enhances the website look and feel and user experience by allowing precise control over the presentation of HTML elements. Mastering CSS is crucial for effective web design and development.

3.3 JavaScript

JavaScript is a programming language used to create dynamic content for websites. It is a lightweight, cross-platform, and single-threaded programming language. JavaScript is an interpreted language that executes code line by line providing more flexibility. HTML adds Structure to a web page, CSS styles it, and JavaScript brings it to life by allowing users to interact with elements on the page, such as actions on clicking buttons, filling out forms, and showing animations. JavaScript is also used on the Server side to do operations like accessing databases, file handling, and security features to send responses to browsers

3.4 Python

Python is a widely used high-level, general-purpose, interpreted, dynamic programming language. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than would be possible in languages such as C++ or Java. The language provides constructs intended to enable clear programs on both a small and large scale. Python supports multiple programming paradigms, including object-oriented, imperative, and functional programming or procedural styles. It features a dynamic type system and automatic memory management and has a large and comprehensive standard library. Python interpreters are available for installation on many operating systems, allowing Python code execution on a wide variety of systems.

3.5 Flask

Flask is a lightweight and flexible web framework for Python. It's designed to make getting started with web development quick and easy, while still being powerful enough to build complex web applications. Flask is an API of Python that allows us to build web applications. It was developed by Armin Ronacher. Flask's framework is more explicit than Django's framework and is also easier to learn because it has less base code to implement a simple web application. Flask Python is based on the WSGI(Web Server Gateway Interface) toolkit and Jinja2 template engine

3.6 TensorFlow

TensorFlow is a popular open-source library developed by Google for machine learning and artificial intelligence applications. It provides a flexible framework for building and training a wide range of models, from simple linear regression to complex deep neural networks[5].

3.7 Keras

Keras is a Python wrapper library that provides wrappers to other DL libraries such as TensorFlow, CNTK, Theano, MXNet, and Deeplearning4.27 It was developed with the goal of rapid experimentation and released under the MIT license. Keras runs under Python 2.7 to 3.6 and provides GPUs and CPU support. Keras is Python's high-level deep learning platform that can operate on top of TensorFlow.

The most important advantage of using Keras, created by Francois Chollet, is the time saved by its easy-to-use but efficient high-level APIs, allowing quick prototyping for a concept. Keras helps us use TensorFlow's principles in a much more straightforward and user-friendly way without writing unnecessary boilerplate software to create deep learning models. The principal reason for success of Keras is its ease of elasticity and flexibility. In addition to providing easy access to specialized libraries, Keras assures that we can still utilize the benefits provided by the TensorFlow package.[8] Using the common pip or conda install command, Keras can be installed easily. We must presume that we have TensorFlow installed because it needs to be used as a backend for the creation of the Keras model (Sarkar et al., 2018)

3.8 Pandas

Pandas is a Python package that provides fast, flexible, and expressive data structures designed to make working with "relational" or "labeled" data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python. Additionally, it has the broader goal of becoming the most powerful and flexible open source data analysis / manipulation tool available in any language. It is already well on its way towards this goal[10].

3.9 numPy

NumPy: The Foundation of Scientific Computing in Python NumPy, short for Numerical Python, is a fundamental library for numerical computing in Python. It provides a powerful N-dimensional array object, along with a collection of tools for working with these arrays efficiently.

3.10 matPlotLib

Matplotlib is a powerful plotting library in Python used for creating static, animated, and interactive visualizations. Matplotlib's primary purpose is to provide users with the tools and functionality to represent data graphically, making it easier to analyze and understand. It was originally developed by John D. Hunter in 2003 and is now maintained by a large community of developers.

3.11 jupyter notebook

Jupyter Notebook (formerly IPython Notebook) is a web-based interactive computational environment for creating notebook documents. Jupyter Notebook is built using several open-source libraries, including IPython, ZeroMQ, Tornado, jQuery, Bootstrap, and MathJax. A Jupyter Notebook application is a browser-based REPL containing an ordered list of input/output cells which can contain code, text (using Github Flavored Markdown), mathematics, plots and rich media. Jupyter Notebook is similar to the notebook interface of other programs such as Maple, Mathematica, and SageMath, a computational interface style that originated with Mathematica in the 1980s. Jupyter interest overtook the popularity of the Mathematica notebook interface in early 2018.[15]

JupyterLab is a newer user interface for Project Jupyter, offering a flexible user interface and more features than the classic notebook UI. The first stable release was announced on February 20, 2018.[17][16] In 2015, a joint $6 million grant from The Leona M. and Harry B. Helmsley Charitable Trust, The Gordon and Betty Moore Foundation, and The Alfred P. Sloan Foundation funded work that led to expanded capabilities of the core Jupyter tools, as well as to the creation of JupyterLab.[15]

GitHub announced in November 2022 that JupyterLab would be available in its online Coding platform called Codespace.[2] In August 2023, Jupyter AI, a Jupyter extension, was released. This extension incorporates generative artificial intelligence into Jupyter notebooks, enabling users to explain and generate code, rectify errors, summarize content, inquire about their local files, and generate complete notebooks based on natural language prompts. [12] JupyterHub is a multi-user server for Jupyter Notebooks. It is designed to support many users by spawning, managing, and proxying many singular Jupyter Notebook servers

3.12 openCV

Opencv is a huge open-source library for computer vision, machine learning, and image processing. Now, it plays a major role in real-time operation which is very important in today's systems. By using it, one can process images and videos to identify objects, faces, or even the handwriting of a human. When it is integrated with various libraries, such as NumPy, python is capable of processing the opencv array structure for analysis.

To Identify an image pattern and its various features we use vector space and perform mathematical operations on these features. The first OpenCV version was 1.0. OpenCV is released under a BSD license and hence it's free for both academic and commercial use. It has C++, C, Python, and Java interfaces and supports Windows, Linux, Mac OS, iOS and Android. When opencv was designed the main focus was real-time applications for computational efficiency. All things are written in optimized C/C++ to take advantage of multi core processing[4].

3.13 sckit-learn

Scikit-learn has emerged as a powerful and user-friendly Python library. Its simplicity and versatility make it a better choice for both beginners and seasoned data scientists to build and implement machine learning models. In this article, we will explore about Sklearn. Scikit-learn is an open-source Python library that implements a range of machine learning, pre-processing, cross-validation, and visualization algorithms using a unified interface. It is an open-source machine-learning library that provides a plethora of tools for various machine-learning tasks such as Classification, Clustering, and many more.

3.14 SciPy

SciPy is an open-source Python library which is used to solve scientific and mathematical problems. It is built on the NumPy extension and allows the user to manipulate and visualize data with a wide range of high-level commands. As mentioned earlier, SciPy builds on NumPy and therefore if you import SciPy, there is no need to import NumPy.

Key Features of SciPy

- Mathematical Functions:Specialized functions like Bessel, Gamma, and error functions and polynomial fitting and manipulation.
-
- Optimization: Tools for finding minima, maxima, or roots of functions and methods like gradient descent, linear programming, and curve fitting.
- Linear Algebra:Advanced operations such as eigenvalues, matrix decomposition (e.g., QR, SVD), and solving linear systems.
-

3.15 librosa

Librosa is valuable Python music and sound investigation library that helps programming designers to fabricate applications for working with sound and music document designs utilizing Python. This Python bundle for music and sound examination is essentially utilized when we work with sound information, like in the music age (utilizing Lstm's), Automatic Speech Recognition. The library is not difficult to utilize and can deal with fundamental as well as cutting edge errands connected with sound and music handling. It is open source and uninhibitedly accessible under the ISC License.

The library upholds a few elements connected with sound records handling and extraction like burden sound from a circle, register of different spectrogram portrayals, symphonious percussive source detachment, conventional spectrogram decay, stacks and translates the sound, Time-space sound handling, successive demonstrating, coordinating consonant percussive partition, beat-simultaneous and some more. Librosa assists with picturing the sound signs and furthermore does the component extractions in it utilizing different sign handling methods[13].

3.16 pyAudio

PyAudio provides Python bindings for PortAudio v19, the cross platform audio I/O library. With PyAudio, you can easily use Python to play and record audio on a variety of platforms, such as GNU/Linux, Microsoft Windows, and Apple macOS

PyAudio is a Python library that provides bindings for PortAudio, a cross-platform audio library. It is widely used for audio processing tasks like recording, playback, and real-time audio stream handling.

Key Features of PyAudio:

- Audio Playback: Play audio files or generated audio streams in real-time.
- Audio Recording: Record audio from input devices like microphones.
- Stream Management: Provides support for audio streaming (input/output) with adjustable parameters like sample rate and chunk size.
- Cross-Platform: Compatible with major operating systems, including Windows, macOS, and Linux.
- Custom Audio Processing: Allows developers to process audio data on the fly, enabling tasks like real-time speech recognition, audio synthesis, or filtering

3.17 wave

The wave module in Python's standard library provides a simple interface for reading and writing WAV files. It's particularly useful for basic audio processing tasks like: Reading WAV file metadata: Get information about the file's parameters, such as sample rate, number of channels, and bit depth. Reading raw audio data: Extract the raw audio data from a WAV file as a byte string. Writing WAV files: Create new WAV files with specified parameters and write audio data to them

3.18 Anaconda

Anaconda is an open-source distribution of the Python and R programming languages for data science that aims to simplify package management and deployment. Package versions in Anaconda are managed by the package management system, conda, which analyzes the current environment before executing an installation to avoid disrupting other frameworks and packages.

The Anaconda distribution comes with over 250 packages automatically installed. Over 7500 additional open-source packages can be installed from PyPI as well as the conda package and virtual environment manager.

It also includes a GUI (graphical user interface), Anaconda Navigator, as a graphical alternative to the command line interface. Anaconda Navigator is included in the Anaconda distribution, and allows users to launch applications and manage conda packages, environments and channels without using command-line commands. Navigator can search for packages, install them in an environment, run the packages and update them[15].

4 Methodology

4.1 CNN

A convolutional neural network is a feed-forward neural network that is generally used to analyze visual images by processing data with grid-like topology. It's also known as a ConvNet. A convolutional neural network is used to detect and classify objects in an image Artificial Intelligence has come a long way and has been seamlessly bridging the gap between the potential of humans and machines. And data enthusiasts all around the globe work on numerous aspects of AI and turn visions into reality and one such amazing area is the domain of Computer Vision. This field aims to enable and configure machines to view the world as humans do, and use the knowledge for several tasks and processes (such as Image Recognition, Image Analysis and Classification, and so on). And the advancements in Computer Vision with Deep Learning have been a considerable success, particularly with the Convolutional Neural Network algorithm[10].

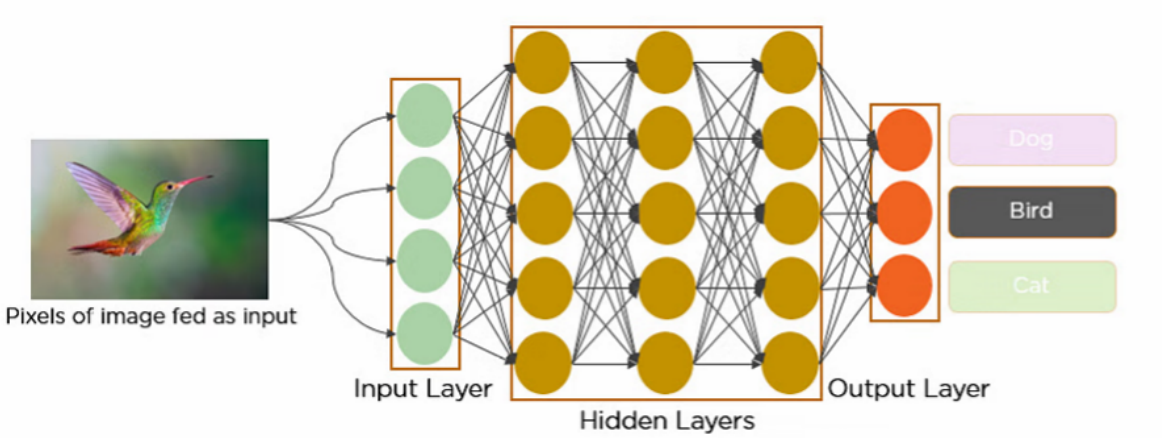

Figure 1: Convolution Neural Network to identify the image of bird[9]

4.2 Layers in a Convolutional Neural Network

A convolution neural network has multiple hidden layers that help in extracting information from an image. The four important layers in CNN are:

1. Convolution layer
2. ReLU layer
3. Pooling layer
4. Fully connected layer
5. ReLU layer/ Activation Layer
6. Flattening

7. Output Layer

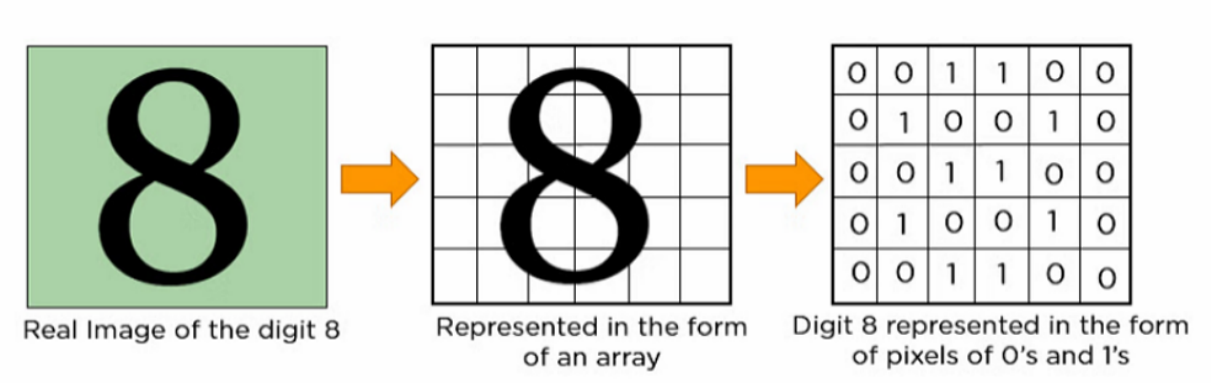

Figure 2: Feature Extraction[3]

4.3 Convolutional Neural Network Training

Training a Convolutional Neural Network (CNN) involves guiding the model to recognize patterns in data through a step-by-step learning process. This is typically done using supervised learning, where the CNN is fed a bunch of images with their correct labels, and it gradually learns how to associate images with the right labels. Here's how the process works:

- Data Preparation

First things first, the images need to be prepared before training can start. This means making sure all the images are uniform in terms of format and size. By preprocessing the data in this way, you ensure that the CNN gets consistent input, which is crucial for its learning process.

- Loss Function

Once the images are ready, the next step is to figure out how well the CNN is doing. This is where the loss function comes into play. Think of it as a scorecard that measures the difference between what the model predicted and the actual label of the image. The smaller the difference, the better the model is performing, so the goal is to reduce this gap as much as possible

- Optimizer

Now that we know how well (or poorly) the CNN is performing, it's time to improve it. The optimizer is like a coach that adjusts the network's weights to help it do better. It tweaks the model's parameters to minimize the loss function, ultimately leading to more accurate predictions over time

- Backpropagation

Backpropagation is the magic behind the scenes that makes everything work. It's the process of figuring out how much each weight in the network contributed to the errors and then adjusting those weights accordingly. The optimizer uses this information to make smarter updates, helping the model get better with each round of training

4.4 LSTM

LSTM excels in sequence prediction tasks, capturing long-term dependencies. Ideal for time series, machine translation, and speech recognition due to order dependence.The article provides an in-depth introduction to LSTM, covering the LSTM model,architecture, working principles, and the critical role they play in various applications.

The LSTM architectures involves the memory cell which is controlled by three gates: The input gate, the forget gate, and the output gate. These gates decide what information to add to, remove from, and output from the memory cell.

- The input gate controls what information is added to the memory cell.
- The forget gate controls what information is removed from the memory cell.
- The output gate controls what information is output from the memory cell.

This allows LSTM networks to selectively retain or discard information as it flows through the network, which allows them to learn long-term dependencies. The LSTM maintains a hidden state, which acts as the short-term memory of the network. The hidden state is updated based on the input, the previous hidden state, and the memory cell's current state.

Bidirectional LSTM (Bi LSTM/ BLSTM)

This is a recurrent neural network (RNN) that is able to process sequential data in both forward and backward directions. This allows Bi LSTM to learn longer-range dependencies in sequential data than traditional LSTMs[8], which can only process sequential data in one direction. Bi LSTMs are made up of two LSTM networks, one that processes the input sequence in the forward direction and one that processes the input sequence in the backward direction. The outputs of the two LSTM networks are then combined to produce the final output.

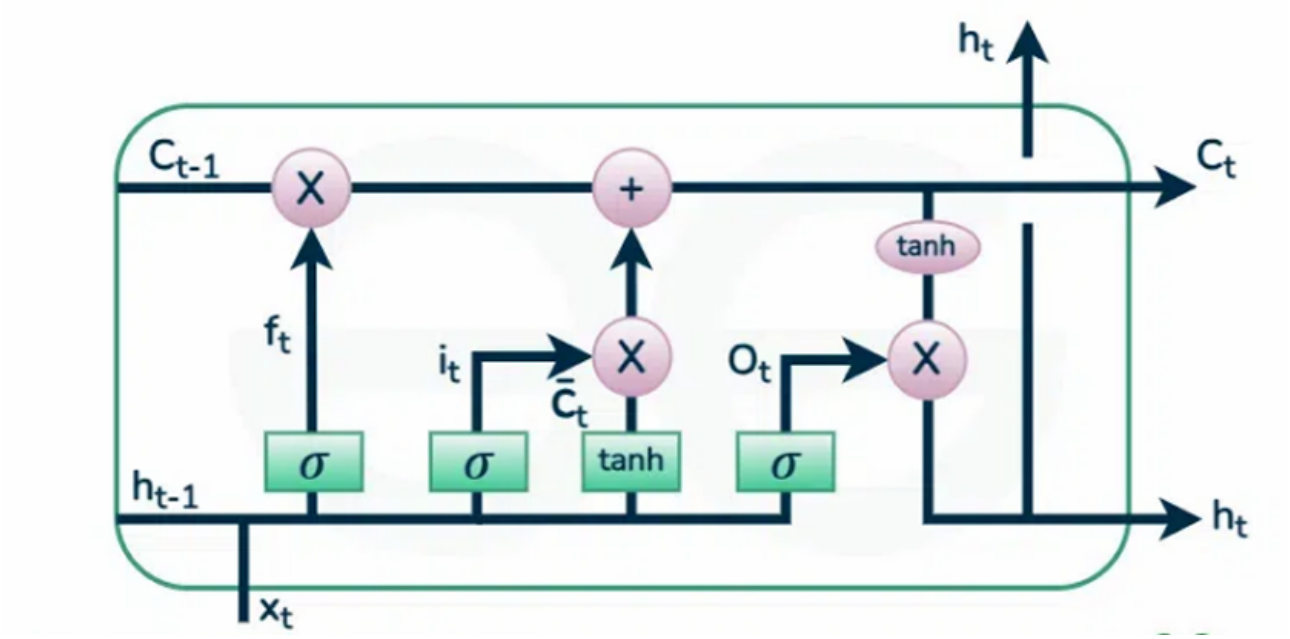

Figure 3: LSTM Model[11]

4.4.1 Application of LSTM

Some of the famous applications of LSTM includes:

- Language Modeling: LSTMs have been used for natural language processing tasks such as language modeling, machine translation, and text summarization. They can be trained to generate coherent and grammatically correct sentences by learning the dependencies between words in a sentence.
- Speech Recognition: LSTMs have been used for speech recognition tasks such as transcribing speech to text and recognizing spoken commands. They can be trained to recognize patterns in speech and match them to the corresponding text.
- Time Series Forecasting: LSTMs have been used for time series forecasting tasks such as predicting stock prices, weather, and energy consumption. They can learn patterns in time series data and use them to make predictions about future events.
- Anomaly Detection: LSTMs have been used for anomaly detection tasks such as detecting fraud and network intrusion. They can be trained to identify patterns in data that deviate from the norm and flag them as potential anomalies.
- Video Analysis: LSTMs have been used for video analysis tasks such as object detection, activity recognition, and action classification. They can be used in combination with other neural network architectures, such as Convolutional Neural Networks (CNNs), to analyze video data and extract useful information.

5 Analysis

5.1 Audio Analysis

Audio information plays a rather important role in the increasing digital content that is available today, resulting in a need for methodologies that automatically analyse such content: audio event recognition for home automations and surveillance systems, speech recognition, music information retrieval, multimodal analysis (e.g. audio-visual analysis of online videos for the content-based recommendation), etc.

Here we present the theoretical background behind the wide range of the implemented methodologies, along with evaluation metrics for some of the methods. pyAudioAnalysis has been already used in several audio analysis research applications: smart-home functionalities through audio event detection, speech emotion recognition, depression classification based on audio-visual features, music segmentation, multimodalcontent-based movie recommendation and health applications (e.g. monitoring eating habits).

The feedback provided from all these particular audio applications has led to practical enhancement of the library.The emotional information embedded in speech plays a crucial role in human communication, as it provides feedback without altering linguistic content. Spoken communication operates through two channels: the primary channel, which conveys linguistic information, and the secondary channel, which communicates paralinguistic cues like tone, emotional state, and gestures. Recognizing and processing secondary channel information enhances communication by aiding convergence, clarifying intent, avoiding misunderstandings, and providing additional speaker details like origin, gender, or age. Emotion recognition systems simplify

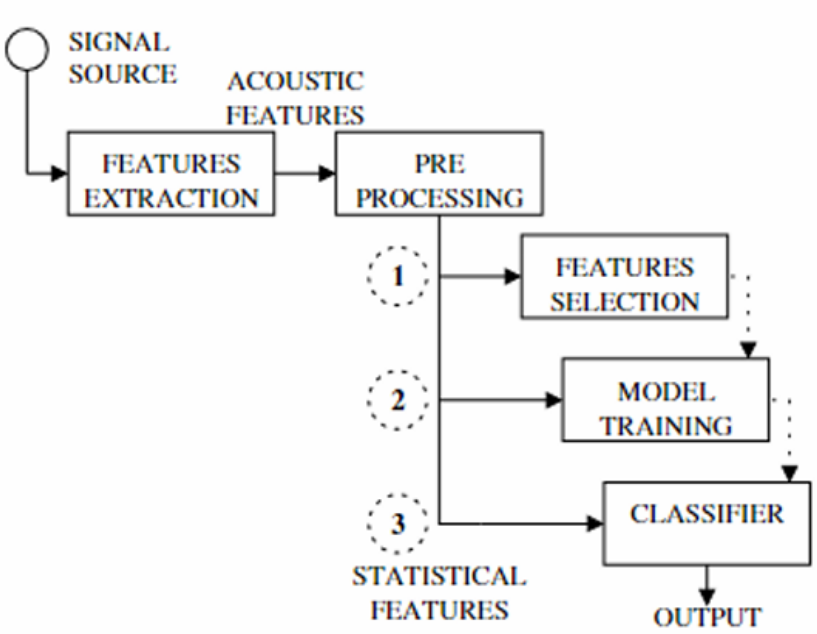

Figure 4: Audio Analysis Model

complex emotions into 2D or 3D spaces, using dimensions like valence (positive/negative emotion), activation (intensity), and dominance (submissiveness/strength). Such systems have applications in psychiatry, CRM, speech synthesis, alarms, and more[12].

5.2 Video Analysis

This image highlights two distinct approaches to detecting and analyzing human emotions: vision-based techniques and physiological (bio-signal) methods. Both approaches leverage deep learning models to process input data and extract meaningful insights about human emotions. The first approach, vision-based techniques, involves using a camera to capture images or video of a person's face.

These visuals serve as the input to a deep learning model, which analyzes facial features such as expressions, gestures, and other visual cues. The model classifies the observed facial expressions into predefined categories such as anger, happiness, sadness, or surprise. This method addresses emotion detection as a classification problem, where the primary goal is to predict a specific emotion based on the visual data. Vision-based techniques are widely used due to their non-invasive nature and the accessibility of cameras in devices like smartphones and computers.

5.3 Eye Detection

A "haar cascade classifier" is an effective machine learning based approach for object detection. To train a haar cascade classifier for eye detection, the algorithm initially needs a lot of positive images (images of eyes) and negative images (images without eyes). Then the classifier is trained from these positive and negative images. It is then used to detect eyes in other images. We can use already trained haar cascades for eye detection.

For eye detection in the input image, we need two haar cascades one for face one that processes

the input sequence in the backward direction. The outputs of the two LSTM networks are then combined to produce the final output

To detect eyes in an image and draw bounding boxes around them, you can follow the steps given below:

- Import the required library. In all the following examples, the required Python library is OpenCV. Make sure you have already installed it.
- Read the input image using cv2.imread() in a grayscale. Specify the full image path.
- Initiate the Haar cascade classifier objects faceCascade = cv2.CascadeClassifier() for face detection and eyeCascade = cv2.CascadeClassifier for eyes detection. Pass the full path of the haar cascade xml files. You can use the haar cascade file haarcascadeFrontalfacealt.xml to detect faces in the image and haarcascadeEyetreeEyeglasses.xml to detect eyes[15].
- Detect faces in the input image using faceCascadeDetectMultiScale(). It returns the coordinates of detected faces in (x,y,w,h) format.
- Define roi as image[y:y+h, x:x+w] for the detected face. Now detect eyes within the detected face area (roi). Use eyeCascade.detectMultiScale(). It also returns the coordinate of the bounding rectangle of eyes in (ex,ey,ew,eh) format.
- Draw the bounding rectangles around the detected eyes in the original image using cv2.rectangle().
- Display the image with the drawn bounding rectangles around the eyes.

5.4 Head Detection

- First, we detect the face in the webcam feed/video using the above-mentioned haarcascade classifier for the face and make a green color bounding box around[15]. it.
- Next, we detect the eyes using a similar haarcascade classifier trained on eyes and make a red color bounding box around each eye. In addition to making a box around each eye, we also identify and store the center of each box. Here, we are assuming that the center of the bounding box is the same as the center of the eye.
- For computing the angle of tilt we will assume that the line joining the centers of two eyes is perpendicular to the face. We have the coordinates of two centers in terms of (x,y) coordinates. The x-axis is the horizontal axis and y-axis is the vertical axis.
- When two points are given and the angle which the line joining the two points makes with the x-axis can be obtained from geometry using the following expression: In our case, the angle made by the line joining the centers of two eyes with the horizontal is computed.
- The positive angle indicates the right tilt and the negative angle indicates the left tilt. Provided a margin of error of 10 degrees (i.e, if the face tilts more than 10 degrees on either side the program will classify as right or left tilt)[17].

5.5 Pose Estimation

Pose estimation is a computer vision technique that is used to predict the configuration of the body(POSE) from an image. The reason for its importance is the abundance of applications that can benefit from technology.

Human pose estimation localizes body key points to accurately recognize the postures of individuals given an image. These estimations are performed in either 3D or 2D[6].

The main process of human pose estimation includes two basic steps:

- localizing human body joints/key points
- grouping those joints into valid human pose configuration In the first step, the main focus is on finding the location of each key points of human beings. E.g. Head, shoulder, arm, hand, knee, ankle. The second step is grouping those joints into valid human pose configuration which determines the pairwise terms between body parts.

5.6 Hand Detection

We will use mediapipe and OpenCV libraries in python to detect the Right Hand andLeft Hand. We will be using the Hands model from mediapipe solutions to detect hands, it is a palm detection model that operates on the full image and returns an oriented hand bounding box.

Mediapipe is Google's open-source framework, used for media processing. It is cross-platform or we can say it is platform friendly. It can run on Android, iOS, and the web that's what Cross-platform means, to run everywhere.

OpenCV is a Python library that is designed to solve computer vision problems. OpenCV supports a wide variety of programming languages such as C++,Python, Java etc. Support for multiple platforms including Windows, Linux, and MacOS.

5.7 Emotion Detection

The emotion detection system for interview preparation utilizes advanced computer vision and machine learning techniques to analyze facial expressions in real-time and identify emotions. The process begins with data acquisition, where a live video stream is captured using a camera device. Each frame of the video is processed to detect faces using OpenCV's pre-trained Haar Cascade Classifier, which identifies facial regions by analyzing predefined patterns in grayscale images. Once detected, the face is extracted as a Region of Interest (ROI) for further analysis. The ROI is then preprocessed by converting it to RGB format, resizing it, and normalizing pixel values to prepare it for emotion detection[4].

The preprocessed facial data is analyzed using the DeepFace library, which employs state-of-the-art deep learning models to classify emotions such as happy, sad, angry, neutral, and surprised. The system identifies the dominant emotion and displays it in real-time, providing instant feedback to the user. This output is integrated with other metrics like speech analysis, hand gesture recognition, and posture evaluation to create a comprehensive assessment of the interview performance. Privacy and security are key considerations in this system, ensuring that all video and emotion data are processed locally or securely without unauthorized storage of sensitive information. By combining cutting-edge tools like OpenCV, DeepFace, and Python,

this methodology offers a reliable and efficient way to help candidates improve their emotional stability and non-verbal communication for interview preparation.

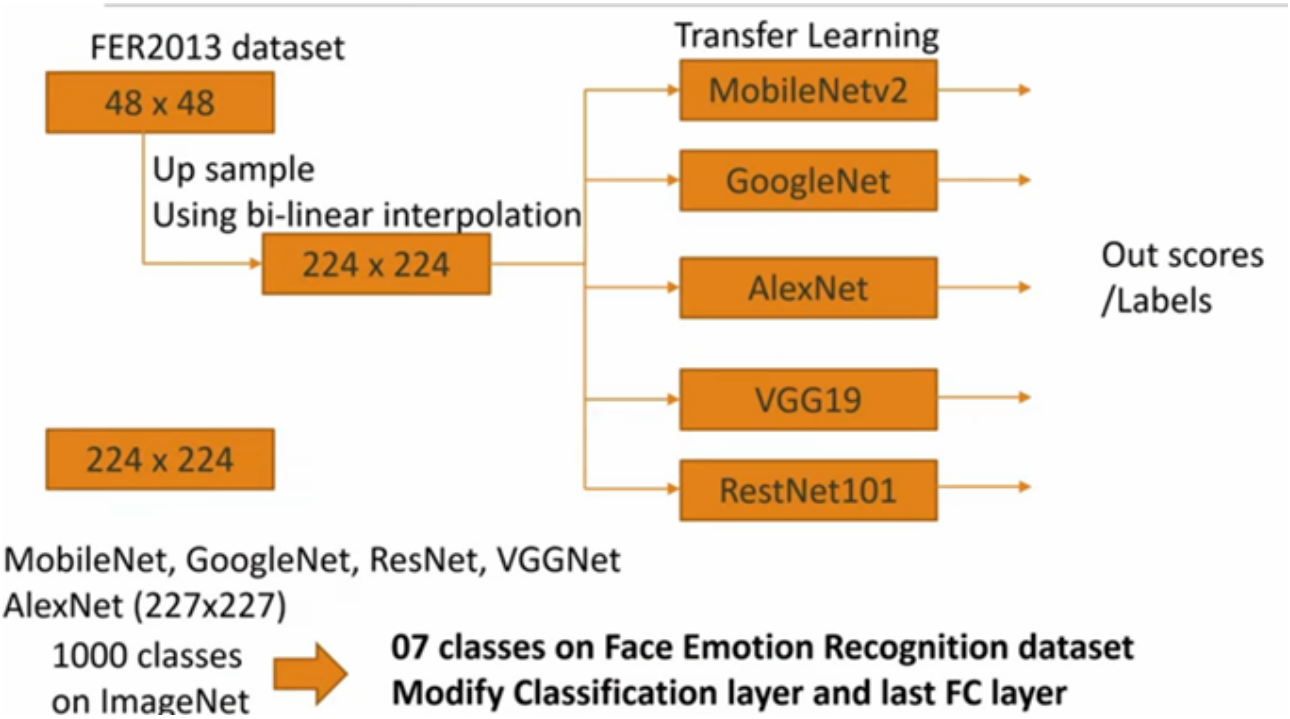

Figure 5: Emotion Detection Model[3]

```
new_model.compile(loss="sparse_categorical_crossentropy", optimizer="adam", metrics=["accuracy"])
training_Data=[] ##data array
img_size=224
Datadirectory ="C:/Users/adity/Downloads/archive/images/validation/"
classes=["0","1","2","3","4","5","6"] ## list of classes
def create_training_Data():
    for category in classes:
        path=os.path.join(Datadirectory ,category)
        class_num=classes.index(category) ## 0,1, label
        for img in os.listdir(path):
            try:
                img_array=cv2.imread(os.path.join(path,img))
                new_array=cv2.resize(img_array,(img_size,img_size))
                training_Data.append([new_array,class_num])
            except Exception as e:
                pass
create_training_Data()
X=[] #data/feature
Y=[] #Label

for features,label in training_Data:
    X.append(features)
    Y.append(label)

X=np.array(X).reshape(-1,img_size,img_size,3) ##converting it into 4 dimmensional
Y=np.array(Y)
X=X/255.0
X.shape
```

Figure 6: Code for Emotion Detection

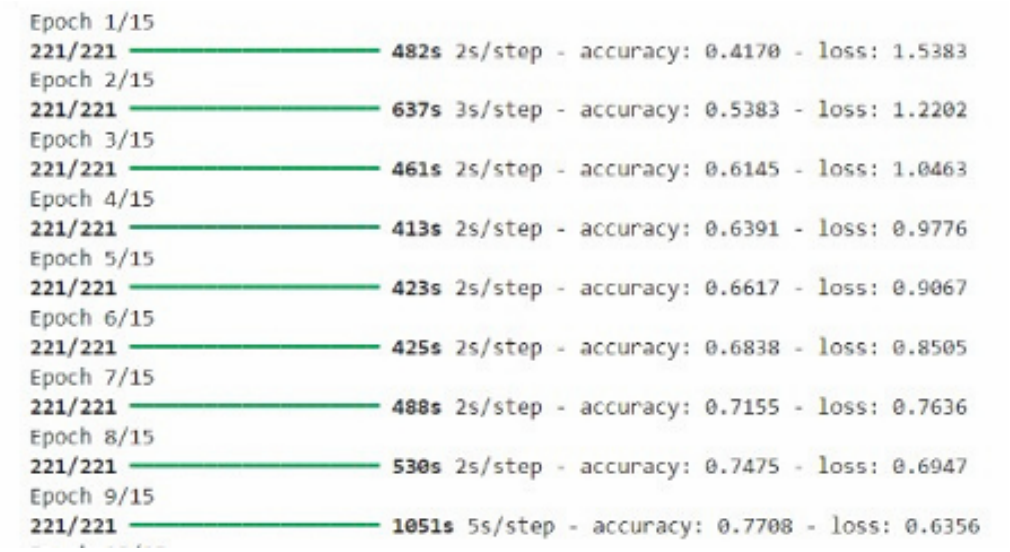

Figure 7: Training of Model

5.8 Question Generation

Pretrained model on English language using a masked language modeling (MLM) objective. It was introduced in this paper and first released in this repository. This model is uncased: it does not make a difference between english and English.

Differently to other BERT models, this model was trained with a new technique: Whole Word Masking. In this case, all of the tokens corresponding to a word are masked at once.

The overall masking rate remains the same. The training is identical – each masked WordPiece token is predicted independently. After pre-training, this model was fine-tuned on the SQuAD dataset with one of our fine-tuning scripts. See below for more information regarding this fine-tuning. Disclaimer: The team releasing BERT did not write a model card for this model so this model card has been written by the Hugging Face team

5.8.1 Description

BERT is a transformers model pretrained on a large corpus of English data in a selfsupervised fashion. This means it was pretrained on the raw texts only, with no humans labelling them in any way (which is why it can use lots of publicly available data) with an automatic process to generate inputs and labels from those texts[11].

More precisely, it was pretrained with two objectives:

- Masked language modeling (MLM): taking a sentence, the model randomly masks 15% of the words in the input then run the entire masked sentence through the model and has to predict the masked words.

  This is different from traditional recurrent neural networks (RNNs) that usually see the words one after the other, or from autoregressive models like GPT which internally mask the future tokens. It allows the model to learn a bidirectional representation of the sentence.

- Next sentence prediction (NSP): the models concatenates two masked sentences as inputs during pretraining. Sometimes they correspond to sentences that were next to each other in the original text, sometimes not. The model then has to predict if the two sentences were following each other or not.

This way, the model learns an inner representation of the English language that can then be used to extract features useful for downstream tasks: if you have a dataset of labelled sentences for instance, you can train a standard classifier using the features produced by the BERT model as inputs. This model has the following configuration:

- 24-layer
- 1024 hidden dimension
- 16 attention heads
- 336M parameters

5.8.2 Limitation

This model should be used as a question-answering model. You may use it in a question answering pipeline, or use it to output raw results given a query and a context. You may see other use cases in the task summary of the transformers documentation.

The BERT model was pretrained on BookCorpus, a dataset consisting of 11,038 unpublished books and English Wikipedia (excluding lists, tables and headers)

5.9 FeedBack

When someone mentions "Question Answering" as an application of BERT, what they are really referring to is applying BERT to the Stanford Question Answering Dataset (SQuAD).

The task posed by the SQuAD benchmark is a little different than you might think.

Given a question, and a passage of text containing the answer, BERT needs to highlight the "span" of text corresponding to the correct answer. The SQuAD homepage has a fantastic tool for exploring the questions and reference text for this dataset, and even shows the predictions made by top-performing models.

The two pieces of text are separated by the special [SEP] token. BERT also uses "Segment Embeddings" to differentiate the question from the reference text. These are simply two embeddings (for segments "A" and "B") that BERT learned, and which it adds to the token embeddings before feeding them into the input layer. For every token in the text, we feed its final embedding into the start token classifier.

The start token classifier only has a single set of weights (represented by the blue "start" rectangle in the above illustration) which it applies to every word. After taking the dot product between the output embeddings and the 'start' weights, we apply the softmax activation to produce a probability distribution over all of the words. Whichever word has the highest probability of being the start token is the one that we pick. We repeat this process for the end token–we have a separate weight vector this.

For every token in the text, we feed its final embedding into the start token classifier. The start token classifier only has a single set of weights (represented by the blue "start" rectangle in the above illustration) which it applies to every word. After taking the dot product between the output embeddings and the 'start' weights, we apply the softmax activation to produce a probability distribution over all of the words.

Whichever word has the highest probability of being the start token is the one that we pick. We repeat this process for the end token–we have a separate weight vector this. For Question Answering we use the BertForQuestionAnswering class from the transformers library.

This class supports fine-tuning, but for this example we will keep things simpler and load a BERT model that has already been fine-tuned for the SQuAD benchmark. The transformers library has a large collection of pre-trained models which you can reference by name and load easily. The full list is in their documentation here.

For Question Answering, they have a version of BERT-large that has already been fineTuned for the SQuAD benchmark. BERT-large is really big... it has 24-layers and an embedding size of 1,024, for a total of 340M parameters! Altogether it is 1.34GB, so expect it to take a couple minutes to download to your Colab instance.

We've concatenated the question and answerText together, but BERT still needs a way to distinguish them. BERT has two special "Segment" embeddings, one for segment "A" and one for segment "B".

Before the word embeddings go into the BERT layers, the segment A embedding needs to be added to the question tokens, and the segment B embedding needs to be added to each of the answerText tokens. These additions are handled for us by the transformer library, and all we need to do is specify a '0' or '1' for each token

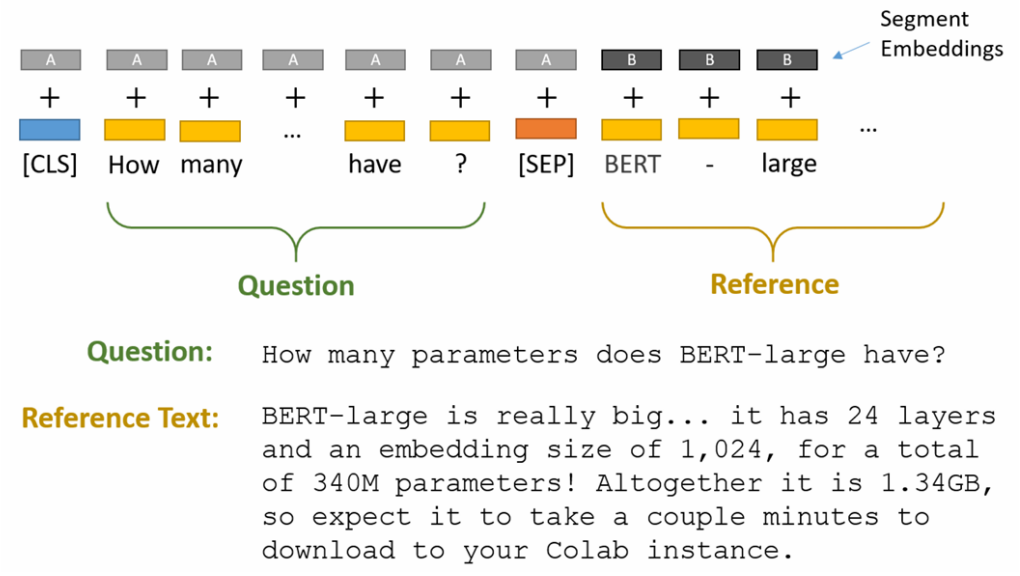

Figure 8: Feedback Model[13]

6 Results And Analysis

We tested two ways to recognize emotions: a) subject-dependent - for each user separately and b) subject-independent - for all users together.

In both cases, for 3-NN classifier, data were randomly divided into the teaching part (70%) and the testing part (30%), and for MLP into three groups: teaching (70%), testing (15%), and validation (15%). Neural network was trained using backpropagation algorithm with conjugate gradient method [13]. The results of the classification for subject-dependent case are shown:

Subject	MLP	3-NN
1	0.94	0.97
2	0.96	0.96
3	0.90	0.98
4	0.74	0.90
5	0.96	0.96
6	0.93	0.97
Average	**0.90**	**0.96**

Figure 9: Accuracy Table[12]

Recognition of emotions based on facial expressions for all users (subject-independent) is much more useful and versatile than for an individual user (subject-dependent). A subject-independent system is designed to generalize across a wide variety of users, making it adaptable for real-world applications where training on specific individuals is not feasible.

This approach ensures inclusivity and robustness, enabling the system to handle diverse facial structures, expressions, and variations due to cultural or demographic factors. In the subject-independent approach, the classifier accuracies (CA) for 3-NN and MLP algorithms were respectively 95.5 and 75.9.

The significantly higher accuracy of the 3-NN classifier demonstrates its ability to effectively recognize and categorize emotions with minimal error, making it a preferred choice for tasks demanding high precision. The MLP algorithm, though less accurate, still provides a baseline for performance and could be improved with further optimization of its architecture or training process.

The results are very good, especially for the 3-NN classifier, which showcases its capacity to differentiate between emotions even when dealing with complex and overlapping datasets. For

that case, in order to determine which emotions are the easiest and which are the most difficult to distinguish, it is necessary to calculate the confusion matrices.

Confusion matrices for 3-NN and MLP classifiers offer a granular view of the system's performance, highlighting areas of strength and weakness in emotion recognition.

These matrices not only identify which emotions are frequently confused but also provide insights into how the classifiers can be further refined to minimize misclassifications. Understanding these distinctions is critical for improving real-world usability, as certain applications, such as mental health monitoring or human-computer interaction, require accurate identification of subtle emotional nuances.

Emotions	neutral	joy	surprise	anger	sadness	fear	disgust
neutral	425	2	1	3	10	0	1
joy	9	421	0	2	6	0	5
surprise	1	1	429	0	0	11	0
anger	7	1	0	428	1	0	6
sadness	20	4	0	2	416	1	0
fear	5	0	19	0	6	412	1
disgust	2	2	1	9	2	0	427

Figure 10: Confusion Matrix[14]

Emotions	neutral	joy	surprise	anger	sadness	fear	disgust
neutral	1130	65	1	40	505	4	45
joy	61	1102	0	68	149	0	124
surprise	0	0	1056	4	0	194	13
anger	47	157	0	1317	30	0	90
sadness	193	41	15	4	726	77	21
fear	4	2	404	0	55	1201	3
disgust	41	109	0	43	11	0	1180

Figure 11: Confusion Matrix[14]

7 Conclusion And Future Scope

7.1 Conclusion

The Virtual Interview Simulator is a groundbreaking tool that bridges the gap between technical proficiency and effective communication, providing aspirants with a comprehensive platform to prepare for real-world interviews. By simulating dynamic, role-specific interactions and delivering personalized feedback, it helps users build confidence, improve verbal articulation, and refine their problem-solving skills under pressure. The simulator equips individuals to handle diverse interviewer styles and unpredictable scenarios, making them more prepared and competitive in professional settings. This solution is not only beneficial for individual aspirants but also holds significant value for educational institutions and corporate training programs. It offers a structured and accessible approach to interview preparation, addressing a critical need for holistic readiness in competitive job markets. With its innovative features, the Virtual Interview Simulator has the potential to empower users to achieve success in their careers.

7.2 Future Scope

The future of the Virtual Interview Simulator is brimming with possibilities for innovation and growth. One area of focus is enhancing AI capabilities to make interactions even more realistic and context-aware. Advanced natural language processing (NLP) models can be integrated to improve emotional intelligence and provide more nuanced, adaptive follow-ups. Additionally, the simulator could generate domain-specific questions tailored to emerging industries, ensuring relevance and adaptability to market trends. Personalization will also play a key role in the evolution of the platform. By developing detailed user profiles that adapt over time, the simulator can create customized training plans that target individual strengths and weaknesses. Users could choose specific areas to focus on, such as technical problem-solving or behavioral questions, and adjust difficulty levels to match their progress. Integration with platforms like LinkedIn, LeetCode, or HackerRank would further enhance the user experience by tailoring simulations to job roles or skills already demonstrated. Another promising avenue is the development of industry-specific modules to prepare users for niche roles in fields like finance, healthcare, and technology. These modules could simulate complex scenarios, including panel interviews and case-study discussions. For added engagement, gamification elements such as leaderboards, achievement badges, and progress tracking could be introduced, motivating users while providing actionable insights into their performance trends. The platform's reach can be expanded globally by introducing multilingual support and mobile friendly or offline versions. This would cater to non-English speakers and make the simulator accessible to users in remote areas. Collaborative features, such as peer-to-peer simulations and group discussions, could also be added to mimic workplace dynamics and team-based scenarios. Finally, the Virtual Interview Simulator could serve as a valuable recruitment tool for companies by enabling mock interview drives and direct talent evaluation. With continuous innovation and expansion, this platform has the potential to revolutionize interview preparation and ensure users are ready to meet the challenges of the professional world with confidence and success.

References

[1] Giannakopoulos T, Pikrakis A. Introduction to Audio Analysis: A MATLAB Approach. Academic Press; 2014.

[2] Theodoridis S, Koutroumbas K. Pattern Recognition, Fourth Edition. Academic Press, Inc.; 2008.

[3] Hyoung-Gook K, Nicolas M, Sikora T. MPEG-7 Audio and Beyond: Audio Content Indexing and Retrieval. John Wiley & Sons; 2005.

[4] Gouyon F, Klapuri A, Dixon S, Alonso M, Tzanetakis G, Uhle C, et al. An experimental comparison of audio tempo induction algorithms. Audio, Speech, and Language Processing, IEEE Transactions on. 2006; 14(5):1832–1844. doi: 10.1109/TSA.2005.858509

[5] Pikrakis A, Antonopoulos I, Theodoridis S. Music meter and tempo tracking from raw polyphonic audio. Proceedings of the International Conference on Music Information Retrieval (ISMIR); 2004. [Link] (https://github.com/tyiannak/pyAudioAnalysis/wiki)

[6] Plumpe M, Acero A, Hon H, Huang X. HMM-based smoothing for concatenative speech synthesis. Proc. ICSLP 1998

[7] Pikrakis A, Giannakopoulos T, Theodoridis S. A Speech/Music Discriminator of Radio Recordings Based on Dynamic Programming and Bayesian Networks. IEEE Transactions on Multimedia; 2008; 5(10):846–855.

[8] Anguera Miro X, Bozonnet S, Evans N, Fredouille C, Friedland G, Vinyals O. Speaker diarization: A review of recent research.Audio, Speech, and Language Processing, IEEE Transactions on. 2012; 20(2):356–370.

[9] Tranter SE, Reynolds DA. An overview of automatic speaker diarization systems.Audio, Speech, and Language Processing, IEEE Transactions on. 2006; 14(5):1557–1565. doi: 10.1109/TASL.2006.878256

[10] Giannakopoulos T, Petridis S. Fisher linear semi-discriminant analysis for speaker diarization .Audio, Speech, and Language Processing, IEEE Transactions on. 2012; 20(7):1913–1922. doi: 10.1109/TASL.2012.2191285

[11] Vendramin L, Campello RJ, Hruschka ER. On the Comparison of Relative Clustering Validity Criterin. SDM 2009 (pp. 733–744).

[12] Vinciarelli A, Dielmann A, Favre S, Salamin H. Canal9: A database of political debates for analysis of social interactions. Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009. 3rd International Conference on(pp. 1–4).

[13] Bartsch MA, Wakefield GH. Audio thumbnailing of popular music using chroma-based representations. IEEE Transactions on Multimedia; 2005; 7(1):96–104. doi: 10.1109/TMM.2004.840597

[14] Lehinevych T, Kokkinis-Ntrenis N, Siantikos G, Dogruoz AS, Giannakopoulos T, Konstantopoulos S. Discovering Similarities for Content-Based Recommendation and Browsing in Multimedia Collections. 2014 Tenth International Conference on Signal-Image Technology and Internet-Based Systems (SITIS).

[15] Giannakopoulos T, Smailis C, Perantonis S, Spyropoulos C. Realtime depression estimation using mid-term audio features. International Workshop on Artificial Intelligence and Assistive Medicine; 2014.

[16] Tsiakas K, Watts L, Lutterodt C, Giannakopoulos T, Papangelis A, Gatchel R, et al. A Multimodal Adaptive Dialogue Manager for Depressive and Anxiety Disorder Screening: A Wizard-of-Oz Experiment. 8th PErvasive Technologies Related to Assistive Environments (PETRA2015) conference.

[17] Giannakopoulos T, Siantikos G, Perantonis S, Votsi NE, Pantis J. Automatic soundscape quality estimation using audio analysis. 8th PErvasive Technologies Related to Assistive Environments (PETRA2015) conference.