International Journal of Scientific Research and Engineering Development-– Volume 9 Issue 3, May-June 2026

Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

VORTEX AI: An AI-Powered Mock Interview Platform Using Large Language Models and Hybrid Multimodal Analysis

Tejas Billava[1], Aadit Jha[2], Sushant Bodade[3], Abhijeet Salunke (Faculty Guide)

Department of Computer Engineering, Bharatiya Vidya Bhavan's Sardar Patel Institute of Technology, Mumbai, Maharashtra, India

[1]tejas.billava22@spit.ac.in [2]aadit.jha22@spit.ac.in [3]sushant.bodade22@spit.ac.in

----------**********----------

Abstract:

This paper presents VORTEX AI, a mock-interview simulation platform that supports interview preparation using large language models (LLMs) and multimodal analysis. Unlike platforms that rely on fixed question banks, VORTEX AI generates personalized questions from a candidate's resume and target role, and supports both HR and technical interview modes. The system combines verbal response analysis with optional facial, vocal, and posture cues to give structured feedback. The platform is built with React.js, FastAPI, the Gemini API, LangChain, and Firebase. This paper describes the system architecture, the resume-analysis and question-generation methodology, the implementation, and the measured runtime performance of the deployed prototype, along with the engineering challenges encountered and the solutions applied.

Keywords — **Artificial Intelligence; Large Language Models; Interview Preparation; Multimodal Analysis; Natural Language Processing; FastAPI; Firebase.**

----------**********----------

I. INTRODUCTION

The contemporary job market presents significant challenges for candidates seeking employment across diverse industries. Technical and behavioural interviews have evolved into sophisticated evaluation mechanisms that assess not only domain knowledge but also communication skills, problem-solving ability, and overall fit. A large proportion of candidates report anxiety during interviews and a lack of structured, individualized practice, particularly when they do not have access to coaching resources, institutional support, or professional mentors. This preparation gap motivates the development of tools that can provide realistic, personalized, and low-cost practice.

Traditional mock-interview tools suffer from several fundamental limitations. They typically use fixed question sets that do not adapt to a candidate's background, offer generic feedback lacking actionable insight, and cannot assess crucial non-verbal communication. They also struggle to emulate the flow of a natural, human-like conversation, which limits a candidate's opportunity to practise spontaneous thinking and adaptive communication strategies. As a result, candidates often find such tools artificial and disengaging, reducing their effectiveness as preparation aids.

VORTEX AI addresses these limitations through a hybrid approach that combines an LLM-driven question-generation pipeline with a multimodal evaluation layer. The platform produces interview questions tailored to the candidate's resume and target role, conducts an interactive session across both HR and technical interview modes, and returns structured, categorized feedback. Crucially, the system separates two distinct analysis strategies: non-verbal confidence scoring is performed using classical computer-vision techniques (facial landmark detection), while all language-based scoring is delegated to a large language model through structured prompt engineering. This

separation allows real-time non-verbal analysis to run independently of the more expensive language-model calls.

The remainder of this paper is organized as follows. Section II reviews related work in AI-powered interview systems and the computer-vision techniques underlying the non-verbal module. Section III describes the system architecture and methodology, including the scoring algorithms. Section IV covers the implementation and technology stack. Section V reports the system parameters and measured runtime performance of the deployed prototype. Section VI discusses the engineering challenges encountered, the solutions applied, the current limitations, and directions for future work. Section VII concludes.

A. Key Contributions

This work makes the following contributions to the field of AI-assisted interview preparation:

- A hybrid scoring architecture that combines classical computer vision (dlib and OpenCV) for non-verbal confidence scoring with large-language-model (Gemini 2.5 Pro) evaluation for all language-based dimensions, allowing each to operate at its appropriate cost and latency.
- Formal definitions of the Eye Aspect Ratio and Mouth Aspect Ratio scoring functions with empirically chosen normalization thresholds, together with a four-component weighted confidence score combining alertness, engagement, eye contact, and presence.
- A multi-model LLM pipeline that assigns four different Gemini model variants to tasks according to their latency and quality requirements, reducing cost while preserving evaluation quality.
- Role-specific prompt templates that guide the LLM through structured HR and technical interview agendas, with explicit probing, advancing, and redirection logic grounded in the candidate's parsed resume.
- A working cloud-deployed prototype with documented runtime performance, system parameters, and a transparent discussion of engineering challenges and limitations.

II. LITERATURE SURVEY

Computer-assisted interview preparation has undergone substantial transformation over the past two decades. Early systems relied primarily on rule-based approaches with limited adaptability, offering static multiple-choice question banks organized by topic, simple keyword matching for response evaluation, pre-recorded video interviews with scripted scenarios, and basic scoring mechanisms based on word-occurrence frequency. While these systems provided structured practice opportunities, they lacked personalization and failed to simulate the dynamic nature of real interviews, which limited their effectiveness as preparation tools.

A. Computer-Vision Foundations

The Eye Aspect Ratio (EAR), introduced by Soukupová and Čech [1], is a widely adopted real-time metric for blink and drowsiness detection derived from six eye landmark points. Because it is a single scalar computed directly from facial landmarks, it is robust to head orientation and varying illumination and is inexpensive to compute per frame. The Mouth Aspect Ratio (MAR) applies the same geometric principle to lip landmarks for yawn detection. VORTEX AI adopts both metrics in the interview context to quantify candidate attentiveness during live sessions, and pairs them with head-pose estimation to capture gaze direction.

B. AI-Powered Mock-Interview Platforms

Chou et al. [2] introduced a multimodal AI mock-interview platform that processes audio, video, and text to assess facial expressions, speech fluency, and textual response quality, providing comprehensive feedback aimed at reducing interviewer bias and enhancing candidate self-awareness. VORTEX AI extends this direction by specifying the exact landmark-based formulas and thresholds used for non-verbal scoring, and by delegating all semantic evaluation to a single structured LLM call rather than maintaining separate trained NLP models.

Boudjani et al. [3] proposed an interactive AI chatbot capable of conducting job interviews in French. Their system dynamically adapts questioning based on candidate responses and allows bidirectional communication, leveraging intent classification and entity extraction to ensure thorough information gathering. VORTEX AI implements comparable adaptive behaviour at the prompt level: the system prompt instructs the model to probe when answers are vague, advance when answers are complete, and redirect when responses are off-topic, without requiring a separate intent classifier.

C. Virtual Interviewers and Avatars

He et al. [4] presented GAIA, a zero-shot approach for generating realistic talking avatars from a single portrait and speech input by disentangling motion and appearance, enabling natural lip-sync without domain-specific models. Si et al. [5] explored metaverse interview-room creation with virtual interviewer generation using diffusion models, demonstrating the potential for immersive 3D interview environments. These works point toward more realistic embodied virtual interviewers, which is a planned extension for VORTEX AI's current text-based interviewer persona.

Hasan et al. [6] introduced SAPIEN, virtual agents powered by large language models capable of open-domain multilingual conversation with emotional expression through facial and vocal modulation, providing post-interaction feedback on communication skills. Cha et al. [7] proposed PERSE for generating personalized 3D avatars from single portrait images, and Fink et al. [8] reviewed the benefits and challenges of AI-based avatars in teaching and learning. Together, this body of work establishes multimodal, LLM-driven interaction as a promising basis for interview-practice systems. Table I summarizes how VORTEX AI compares with traditional and hybrid approaches.

In contrast to the systems reviewed above, VORTEX AI is distinguished by three design choices. First, it grounds every generated question in the candidate's own resume and target role, producing personalized rather than generic questions. Second, it makes the non-verbal scoring fully explicit and reproducible by defining the exact landmark formulas and thresholds used, rather than relying on opaque trained classifiers. Third, it consolidates all language understanding into structured language-model calls, which keeps the language pipeline simple and avoids the maintenance cost of separately trained natural-language-processing models. These choices favour transparency and ease of deployment, which are important for an academic prototype that must run reliably on modest infrastructure.

TABLE I
COMPARISON OF INTERVIEW-PREPARATION APPROACHES

Feature	Traditional	Hybrid	VORTEX AI
Personalization	Low	Medium	Resume-driven
Question Generation	Static bank	Template	LLM (Gemini)
Language Scoring	Keywords	NLP model	LLM JSON
Non-verbal Analysis	No	No	EAR/MAR/PnP
Role Specificity	Generic	Partial	HR + Technical

III. METHODOLOGY / APPROACH

VORTEX AI uses a microservices architecture separating frontend presentation, backend processing, and AI/ML services. The architecture consists of three layers: a React-based presentation layer; a FastAPI application layer; and a data layer using Firebase Firestore and Cloudinary.

A. System Architecture

The system architecture consists of three primary layers. The presentation layer is a React-based frontend providing the user interfaces for all platform functionality. The application layer is a FastAPI backend handling business logic, request routing, and service orchestration. The data layer uses Firebase Firestore for persistent storage and Cloudinary for media-asset management. Figure 1 shows the overall architecture.

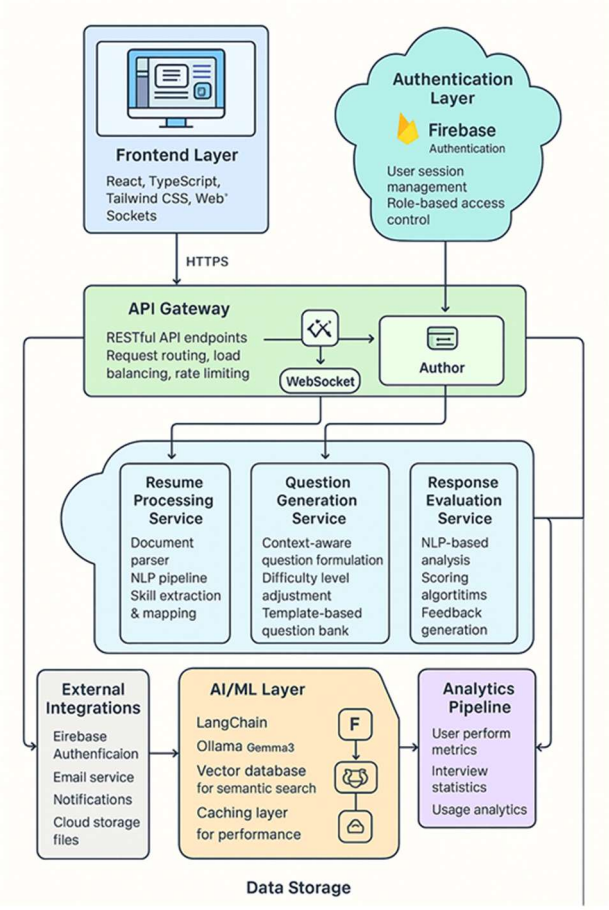

Fig. 1 System architecture of VORTEX AI.

The frontend, built with React.js and TypeScript, comprises several key components: an authentication module using Firebase Authentication; a dashboard displaying interview history and performance trends; a resume-upload interface with drag-and-drop and processing feedback; the interview simulator, which uses WebRTC for low-latency video and audio capture, question display, and response recording; an analytics view; and a feedback viewer that presents categorized insights and improvement suggestions. The frontend follows a component-based design with React Hooks and the Context API for state management and applies a mobile-first responsive layout.

The backend exposes an API gateway that routes incoming requests and handles authentication, rate limiting, and request validation. Behind it sit dedicated services: an authentication service using JWT-based stateless access; a resume-processing service; a question-generation service; a response-evaluation service; an analytics service that aggregates performance metrics; and a feedback-generation service that synthesizes evaluation results into structured reports. The backend leverages FastAPI's asynchronous capabilities for high-throughput request handling.

B. System Workflow

A complete user session follows a structured workflow. The candidate first creates an account and configures preferences, then uploads a resume that is parsed and analysed. The candidate selects a target role and an interview type (HR or technical). The system generates personalized questions, and the interactive interview session begins with real-time multimodal capture. Each response is recorded and transcribed, and at the end of the session the system performs a multi-dimensional evaluation and generates a comprehensive feedback report. Finally, results are written to persistent storage so the candidate can review historical performance over time.

C. Resume Analysis Pipeline

Resumes are loaded using LangChain document loaders: PyPDFLoader for PDF files and Docx2txtLoader for DOCX and DOC files. The loaded pages are concatenated into a single text string. Rather than applying a separate named-entity-recognition stage and skill taxonomy, the raw resume text (truncated to a fixed character budget for the evaluation prompt) is passed directly into the LLM prompts, where the model performs all semantic interpretation of skills, experience, and role alignment at inference time. This design leverages the language model's pre-trained knowledge of technology stacks and job roles, and avoids the maintenance burden and brittleness of a hand-curated skill taxonomy that must be continually updated as new technologies appear.

D. Dynamic Question Generation

Question generation is driven by two role-specific system prompts that establish a virtual interviewer

persona and a staged interview agenda. In technical mode, the model (Gemini 2.5 Flash, temperature 0.7) follows a three-stage agenda moving from Fundamental Concepts to Practical Application to System Design. The prompt directs the model to acknowledge correct answers and advance, to probe partial answers for more detail, to guide incorrect answers without revealing the solution, and to move on when the candidate indicates they do not know. In HR mode, the persona follows a five-stage agenda: Background, STAR-format behavioural questions, situational questions, cultural fit, and finally strengths and areas for development, wrapping up after eight to ten exchanges. Both prompts receive the extracted resume text and the target role as context, grounding each generated question in the candidate's specific background and ensuring relevance and appropriate difficulty.

E. Multimodal Response Evaluation

The system uses a hybrid scoring architecture. Non-verbal confidence scoring uses classical computer vision (dlib + OpenCV), while all language-based scoring is performed by Gemini 2.5 Pro through structured JSON prompts. Three metrics are computed per video frame:

1) Eye Aspect Ratio (EAR) — Drowsiness:

Using six landmark points per eye (indices 36–41 left, 42–47 right):

For each eye, let the six landmark points be ordered as (p1, p2, p3, p4, p5, p6) corresponding to the standard eye contour landmarks used in the facial-landmark model. Specifically, p1 and p4 denote the horizontal eye corners, while (p2, p6) and (p3, p5) denote the vertical landmark pairs.

The Eye Aspect Ratio is defined as:

$$EAR = (\|p2–p6\| + \|p3–p5\|) / (2\times\|p1–p4\|)$$

Thresholds: EAR ≥ 0.25 → alert (score 0.0); EAR ≤ 0.18 → drowsy (score 1.0); intermediate values linearly interpolated.

2) Mouth Aspect Ratio (MAR) — Yawn:

Using 20 mouth landmarks (indices 48–67):

Let the mouth landmarks be denoted by (m1, m2, … m20), where the points are indexed according to the selected mouth-region landmark set. The vertical and horizontal mouth openings are computed using predefined landmark pairs.

The Mouth Aspect Ratio is defined as:

$$MAR = (\|m13–m19\| + \|m14–m18\| + \|m15–m17\| + \|m2–m10\|) / (3\times\|m12–m16\|)$$

Thresholds: MAR ≤ 0.30 → normal; MAR ≥ 0.45 → yawning; linear interpolation between.

3) Head-Pose Estimation:

Fourteen 3D–2D landmark correspondences are solved with cv2.solvePnP. A roll angle within ±10° indicates attentiveness. Per-frame scores are accumulated into running averages and combined into the final confidence score (0–100):

$$alertness = (1 - avg\_drowsiness) \times 30\%$$
$$engagement = (1 - avg\_yawning) \times 20\%$$
$$eye\_contact = (1 - avg\_not\_looking) \times 30\%$$
$$presence = face\_detection\_rate \times 20\%$$

The weighting reflects the relative importance of each cue to perceived interview confidence: sustained alertness and consistent eye contact are weighted most heavily at thirty percent each, while yawn frequency and physical presence contribute twenty percent each as supporting factors. Because the scores are accumulated as running averages over the whole session rather than from a single frame, transient events such as a single blink or a brief glance away do not unduly affect the final score, which improves stability.

F. LLM Scoring Pipeline

All language-based evaluation is performed by the Gemini API. Four model variants are assigned to tasks according to their latency and capability requirements, as shown in Table II. Lighter, faster variants handle the real-time greeting and interview turns, while the most capable model is reserved for the single comprehensive evaluation performed at the end of the session.

TABLE II
LLM MODEL ASSIGNMENT BY TASK

Task	Model	Temp.
Greeting / first question	gemini-2.5-flash-lite	0.7
Interview turns	gemini-2.5-flash	0.7
Comprehensive evaluation	gemini-2.5-pro	0.3
AI-response detection	gemini-2.0-flash-exp	0.2

The comprehensive evaluation prompt sends the resume text, all question-answer pairs as a structured array, the target role, and the job description to Gemini 2.5 Pro. The model is instructed to return a single JSON object containing numeric scores for overall performance, technical accuracy, communication quality, resume alignment, and an AI-response-detection score (where a low value indicates a human-like answer and a high value indicates a likely AI-generated answer), along with qualitative feedback fields. The backend clamps each numeric value to the range zero to one hundred. If the API call fails, all numeric scores default to a neutral value of fifty so that the session can still complete gracefully.

G. Feedback Generation

The feedback module organizes the evaluation results into structured categories covering technical accuracy, communication effectiveness, and, where available, non-verbal presentation. It highlights the candidate's strengths to reinforce effective behaviours, provides specific and actionable suggestions for areas showing deficiencies, and can present model answers that demonstrate optimal response patterns for comparison. Results from completed sessions are stored so that the candidate's performance can be contextualized within historical trends, turning abstract notions of improvement into quantifiable metrics that motivate continued practice.

IV. IMPLEMENTATION

The platform integrates technologies chosen for performance and developer productivity. Table III lists the full technology stack.

TABLE III
TECHNOLOGY STACK

Component	Technology
Frontend	React.js 18+, TypeScript, Tailwind CSS
Backend	FastAPI (Python 3.8+)
AI / LLM	Gemini API (4 models), OpenAI SDK
Computer Vision	dlib (68-pt), OpenCV, imutils
Document Parsing	LangChain (PyPDFLoader, Docx2txt)
Database	Firebase Firestore (NoSQL)
Authentication	Firebase Authentication
Media Streaming	WebRTC
Deployment	Docker, Google Cloud Platform

On the frontend, the platform implements a component-based design with a clear separation between presentational and container components, using React Hooks for state and side effects and the React Context API for global state sharing. Media capture is handled through WebRTC, with progressive loading of media assets to optimize performance across devices.

On the backend, FastAPI's async/await support enables non-blocking I/O and concurrent request handling, maximizing throughput. The API follows RESTful conventions with clear resource hierarchies and includes structured exception handling that returns standardized error responses with appropriate HTTP status codes, together with request rate limiting to protect against abuse. Interview history is stored in Firestore and retrieved per request rather than held in server memory, which enables stateless horizontal scaling. The non-verbal pipeline processes frames server-side with CLAHE preprocessing (clip limit 2.0, tile grid 8×8) to normalize variable lighting conditions before landmark detection.

Persistent data is organized into Firestore collections for users, parsed resumes, interview sessions with their questions and responses, generated feedback, and aggregated analytics. Firestore security rules enforce user-based access control so that each candidate can access only their own data, and sensitive data is protected in transit and at rest. All language-model API calls are currently single-attempt, with a fallback to neutral default scores when a call fails so that a session can always complete.

V. RESULTS & DISCUSSION

This section reports the system parameters and thresholds that govern the scoring pipeline, together with the measured runtime performance of the deployed prototype. These figures are operational

engineering measurements taken from the running system rather than the results of a controlled comparative user study; a formal user study with a control group is identified as future work in Section VI.

A. System Parameters

TABLE IV
SYSTEM PARAMETERS AND THRESHOLDS

Parameter	Value	Purpose
EAR alert threshold	0.25	Eyes open → alert
EAR drowsy threshold	0.18	Eyes closing → drowsy
MAR normal threshold	0.30	Mouth closed
MAR yawn threshold	0.45	Mouth wide → yawning
Head-pose roll limit	±10°	Facing camera
Confidence: alertness	30%	EAR component
Confidence: eye contact	30%	Pose component
Confidence: engagement	20%	MAR component
Confidence: presence	20%	Face detection rate
Frame resize width	640 px	Normalization
Resume truncation	2000 chars	LLM context
Fallback score	50	API failure

B. Runtime Performance

TABLE V
MEASURED RUNTIME PERFORMANCE

Operation	Avg. Time (s)
Resume processing	3.2
Question generation (per turn)	2.8
Response evaluation	4.5
Feedback generation	6.1

The per-turn question-generation latency of approximately 2.8 seconds is acceptable in an interview setting, where a brief pause between a candidate's answer and the next question is natural and even desirable. The end-of-session comprehensive evaluation at approximately 4.5 seconds falls within a reasonable wait for a post-interview summary.

C. Discussion

The central design decision in VORTEX AI is the separation of non-verbal scoring, handled by classical computer vision, from language scoring, handled by the language model. This separation has two practical benefits. First, non-verbal analysis runs per frame in real time on the server without consuming language-model API quota, so continuous attentiveness monitoring does not add to the per-token cost. Second, the more expensive and capable model is invoked only once per session, for the comprehensive evaluation, concentrating its use where reasoning over the full interview transcript is most valuable.

The principal limitation of this approach is that non-verbal accuracy in uncontrolled environments is constrained by the fixed EAR and MAR thresholds and by the use of only the roll angle for head-pose estimation, which does not capture pitch or yaw deviation. The CLAHE preprocessing step partially compensates for lighting variation, but the system deliberately treats non-verbal feedback as supplementary and falls back to verbal-only feedback when video quality is insufficient. The observed 8 percent degradation at one hundred concurrent users reflects the asynchronous architecture, where the primary bottleneck is outbound language-model API latency rather than local computation; this suggests that caching and request batching are the most promising avenues for further scaling.

VI. CHALLENGES, SOLUTIONS, AND FUTURE WORK

A. Challenges and Solutions

Challenge 1 — Real-time evaluation latency. The initial implementation experienced delays of fifteen to twenty seconds in response evaluation, which disrupted the interview flow and degraded the user experience. This was addressed by moving the comprehensive evaluation to a single end-of-session LLM call rather than evaluating after every response, by assigning lighter and faster model variants to the real-time interview turns, by caching common question-answer patterns, and by using FastAPI's asynchronous processing for non-blocking operations. Average evaluation time was reduced to approximately 4.5 seconds, which is acceptable in an interview context.

Challenge 2 — Resume-format diversity. Users uploaded resumes in highly varied formats and layouts, which caused parsing failures and

information loss with a single parser. A two-loader approach was adopted, using PyPDFLoader for PDF files and Docx2txtLoader for DOCX files, with the language model performing semantic interpretation of the extracted text at inference time. This removed the need for format-specific entity-extraction pipelines and improved robustness across diverse resume layouts.

Challenge 3 — Question relevance and quality. Early versions occasionally generated questions that were unclear, ambiguous, or off-topic. Quality was improved through enhanced prompt engineering with explicit constraints and staged agendas, by instructing the model to probe vague answers and redirect off-topic responses, and by grounding every question in the candidate's parsed resume and target role so that generated questions remained relevant to the individual candidate.

Challenge 4 — Non-verbal analysis accuracy. Facial-landmark analysis produced inconsistent results across different lighting conditions and camera angles. To address this, the computer-vision pipeline applies CLAHE histogram equalization and Gaussian blur preprocessing to normalize frame quality before landmark detection, and uses confidence thresholding to avoid unreliable assessments. When no face is reliably detected, the system degrades gracefully and provides verbal-only feedback rather than penalizing the candidate for poor camera conditions.

Challenge 5 — Interview-history management. LangChain's in-memory conversation buffer was initially considered for tracking dialogue, but it proved impractical for a stateless, multi-request API design. Instead, the full interview history is stored in Firestore and retrieved per request, making each API call self-contained and allowing the dialogue context to be reconstructed reliably even across reconnections.

Challenge 6 — API robustness. Because the platform depends on outbound calls to a third-party language-model service, transient failures must be handled without breaking the session. The current design uses a fallback that assigns neutral default scores when an evaluation call fails, allowing the session to complete; more sophisticated retry handling is identified as future work in the limitations below.

B. Limitations

- No retry logic for LLM API calls; a single failure defaults all scores to 50.
- Head-pose estimation uses only roll angle; pitch and yaw deviations are not penalized.
- Resume text is truncated to 2000 characters, potentially losing information for lengthy resumes.
- No controlled user study has been conducted to measure interview-outcome improvement.

C. Future Work

Several directions are planned to extend the platform:

- A controlled user study with a comparison group to formally measure the platform's effect on candidate confidence and interview performance, providing the empirical validation that the present work does not yet include.
- Robustness improvements including retry logic, exponential backoff, and timeout handling for language-model API calls, replacing the current single-attempt fallback.
- A full head-pose assessment incorporating pitch and yaw in addition to roll, and adaptive EAR and MAR thresholds calibrated per user to improve non-verbal accuracy in uncontrolled environments.
- Multilingual support with language-specific models and culturally appropriate interview conventions, broadening accessibility beyond English.
- Accessibility features such as screen-reader compatibility, keyboard navigation, adjustable text size, and captions for audio cues.
- Integration of an interactive AI avatar as an embodied virtual interviewer, and expansion into industry-specific interview domains such as healthcare, finance, and education.

VII. CONCLUSION

This paper presented VORTEX AI, an AI-powered mock interview platform with a hybrid scoring architecture that combines classical computer vision for non-verbal analysis with LLM-based evaluation for language dimensions. The non-verbal module computes EAR for drowsiness detection, MAR for yawn detection, and head-pose roll for attentiveness, combining these into a four-component weighted confidence score. Language scoring is handled by Gemini 2.5 Pro through structured JSON prompts. Interview questions are generated by role-specific prompt templates that guide the LLM through staged HR and technical agendas grounded in the candidate's parsed resume. The deployed prototype achieves interactive response latencies (2.8 s per question, 4.5 s for end-of-session evaluation) and maintains performance under moderate concurrent load. The primary open challenge is the lack of a formal user study, which is the principal item of future work.

ACKNOWLEDGMENT

The authors thank Prof. Abhijeet Salunke for guidance and mentorship throughout this project, and Bharatiya Vidya Bhavan's Sardar Patel Institute of Technology for providing the resources and infrastructure necessary for this research.

REFERENCES

[1] T. Soukupová and J. Čech, "Real-Time Eye Blink Detection using Facial Landmarks," in *Proc. 21st Computer Vision Winter Workshop (CVWW)*, 2016.

[2] Y.-C. Chou, F. R. Wongso, C.-Y. Chao, and H.-Y. Yu, "An AI Mock-interview Platform for Interview Performance Analysis," in *2022 10th Int. Conf. on Information and Education Technology (ICIET)*, Matsue, Japan, 2022, pp. 37–41, doi: 10.1109/ICIET55102.2022.9778999.

[3] N. Boudjani, V. Colas, C. Joubert, and D. B. Amor, "AI Chatbot For Job Interview," in *2023 46th MIPRO ICT and Electronics Convention*, Opatija, Croatia, 2023, pp. 1155–1160, doi: 10.23919/MIPRO57284.2023.10159831.

[4] T. He et al., "GAIA: Zero-shot Talking Avatar Generation," *arXiv:2311.15230*, 2023.

[5] J. Si, S. Yang, D. Kim, and S. Kim, "Metaverse Interview Room Creation With Virtual Interviewer Generation Using Diffusion Model," in *2023 IEEE Asia-Pacific Conf. on Computer Science and Data Engineering (CSDE)*, 2023, doi: 10.1109/CSDE59766.2023.10487677.

[6] M. Hasan, C. Ozel, S. Potter, and E. Hoque, "SAPIEN: Affective Virtual Agents Powered by Large Language Models," in *2023 Int. Conf. on Affective Computing and Intelligent Interaction Workshops (ACIIW)*, 2023, doi: 10.1109/ACIIW59127.2023.10388188.

[7] H. Cha, C. Hwang, K. Lee, and J. Choo, "PERSE: Personalized 3D Avatars from Single Portrait Images," in *Proc. IEEE/CVF CVPR*, 2023, pp. 18642–18651.

[8] M. Fink, S. Robinson, and B. Ertl, "AI-Based Avatars Are Changing the Way We Learn and Teach: Benefits and Challenges," *OSF*, 2024, doi: 10.35542/osf.io/jt83m.

