EMOTIONSENSEAI: ADVANCED AUDIO-BASED EMOTION ANALYTICS

A PROJECT REPORT

Submitted by,

Ms. Tanmayee H.N - 20211CSE0288

Under the guidance of,

Dr. Anand Prakash

Associate Professor, School of Computer Science and Engineering

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

At

PRESIDENCY UNIVERSITY

BENGALURU

MAY 2025

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Project report **"**EmotionSenseAI: Advanced audio-based emotion analytics**"** being submitted by "Tanmayee HN" bearing roll number "20211CSE0288" in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in Computer Science and Engineering is a bonafide work carried out under my supervision.

Dr. Anand Prakash
Associate Professor
PSCS
Presidency University

Dr. Asif Mohammed H.B
Associate Professor& HoD
PSCS
Presidency University

Dr. MYDHILI NAIR
Associate Dean
PSCS
Presidency University

Dr. SAMEERUDDIN KHAN
Pro-Vc School of Engineering
Dean -PSCS / PSIS
Presidency University

PRESIDENCY UNIVERSITY

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled **EmotionSenseAI: Advanced audio-based emotion analytics** in partial fulfillment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering**, is a record of our own investigations carried under the guidance of **Dr. Anand Prakash, Associate Professor, Presidency School of Computer Science and Engineering, Presidency University, Bengaluru.**

I have not submitted the matter presented in this report anywhere for the award of any other Degree.

TANMAYEE HN – 20211CSE0288

INTERNSHIP COMPLETION CERTIFICATE

LinkLabs OÜ
Tallinn, Estonia
www.linklabs.io

INTERNSHIP COMPLETION CERTIFICATE

This is to certify that **Ms. Tanmayee HN** has successfully completed her internship at **LinkLabs OÜ**, Estonia, from **3rd February 2025 to 31st May 2025**.

During this period, she worked as an **AI Engineering Intern** on the project titled:

"EmotionSense AI: Advanced Audio-Based Emotion Analytics"

This project involved the development of intelligent systems capable of detecting human emotions from audio signals using advanced techniques in **Audio Fingerprinting, Artificial Intelligence (AI), and Machine Learning (ML)**. She contributed significantly to the development and integration of various modules using **C++, Java, Spring Framework, AWS, and React**.

Tanmayee displayed a high level of commitment, technical proficiency, and creativity throughout the internship. Her contributions were instrumental in achieving key milestones of the project, and she consistently demonstrated strong analytical and problem-solving skills.

We wish her the very best in all her future endeavors.

Ganesh Banda

Digitally signed by Ganesh Banda
DN: cn=Ganesh Banda, o=LinkLabs OÜ, email=ganesh.banda@linklabs.io, c=EE
Date: 2025.05.08 13:19:33 +05'30'

Date: 31st May 2025
Authorized Signatory
LinkLabs OÜ

ABSTRACT

EmotionSenseAI is an intelligent system designed to bridge the gap between human emotion and digital audio. At its core, it's a powerful platform that listens, understands, and categorizes English music based on the emotions it conveys—automatically, continuously, and at scale. Built with a Java Spring Boot backend and performance-critical modules in C++, EmotionSenseAI uses advanced audio processing techniques like Fast Fourier Transform (FFT), audio fingerprinting, and machine learning to detect emotional patterns in music—whether a song is joyful, melancholic, energetic, or calm. The system is fully automated, meaning it can search for, download, and process English songs from popular platforms like YouTube, SoundCloud, and Spotify without human intervention. It uses Selenium to find music content, yt-dlp to handle downloads, and a smart filtering mechanism to avoid duplicate or irrelevant media—like Shorts, live performances, or non-English content. Once downloaded, the music is organized into local directories and enriched with metadata such as title, duration, link, and most importantly—emotion tags, which are stored in a connected SQL database for easy retrieval and training purposes. This project was born from a simple yet powerful idea: what if machines could help us understand how music makes us feel? In the past, analyzing emotion in audio often required tedious manual tagging and subjective listening. EmotionSenseAI changes that. By combining automation with AI, it turns what used to be a slow and error-prone process into an efficient, scalable system that can build rich, emotional music datasets in real time. As the system has evolved, it has also grown in purpose. A key addition is SwearVault—a companion platform developed to detect offensive or profane words in audio recordings. Its main goal is to collect and analyze such language to help train AI models that can better recognize and handle inappropriate content. Together, EmotionSenseAI and SwearVault form a unified platform for emotion-focused audio intelligence. From music discovery and recommendation to psychological research and ethical AI, the combined system offers an end-to-end solution for collecting, organizing, and analyzing emotional audio content—without manual labor and with a strong commitment to quality and integrity. This report explores the technologies behind the system, the research that inspired it, and the challenges overcome along the way—from handling large volumes of audio to avoiding repeated downloads and ensuring data diversity. Most importantly, it shares the vision of a future where machines not only hear what we say or sing—but also understand what we feel.

ACKNOWLEDGEMENT

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks and gratitude to our Hon'ble **Dr. Md. Sameeruddin Khan**, Pro-VC - Engineering and Dean, Presidency School of Computer Science and Engineering & Presidency School of Information Science, Presidency University for getting us permission to undergo the project.

We express our heartfelt gratitude to our beloved Associate Dean **Dr. Mydhili Nair,** Presidency School of Computer Science and Engineering, Presidency University, and **Dr. Asif Mohammed H.B**, Head of the Department, Presidency School of Computer Science and Engineering, Presidency University, for rendering timely help in completing this project successfully.

We are greatly indebted to our guide and Reviewer **Dr. Anand Prakash, Associate Professor**, Presidency School of Computer Science and Engineering, Presidency University for his inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the internship work.

We would like to convey our gratitude and heartfelt thanks to the CSE7301 Internship/University Project Coordinator **Mr. Md Ziaur Rahman** and **Dr. Sampath A K,** department Project Coordinator **Mr. Jerrin Joe Francis** and Git hub coordinator **Mr. Muthuraj.**

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

TANMAYEE HN

LIST OF FIGURES

TABLE OF CONTENTS

CHAPTER-1
INTRODUCTION

EmotionSenseAI is an innovative artificial intelligence system designed to revolutionize the way we perceive and analyze emotional cues in audio files. By leveraging cutting-edge AI methodologies, this system aims to classify and interpret the emotional tone of speech and music, facilitating deeper insights into user engagement and media impact. Traditional emotion recognition in audio files relied on manual annotation and heuristic methods, often resulting in inefficiencies and inconsistencies. EmotionSenseAI overcomes these challenges by integrating AI-driven feature extraction techniques such as Fast Fourier Transform (FFT) and Mel-Frequency Cepstral Coefficients (MFCC), ensuring a more accurate and automated emotion classification process.

The core functionality of EmotionSenseAI involves acquiring and processing audio data from online platforms, including YouTube, SoundCloud, and Spotify. The system employs advanced web scraping and API-based data retrieval mechanisms to collect a diverse dataset, ensuring a rich variety of audio inputs for analysis. To further enhance efficiency, the system incorporates audio fingerprinting techniques that prevent duplicate downloads, optimizing both storage and processing resources.

EmotionSenseAI is built on a robust technical foundation, with a backend developed in Java Spring Boot and core AI processing implemented in C++. The integration of these technologies allows for scalable and high-performance processing of large audio datasets. The AI model employed in this system utilizes a combination of Convolutional Neural Networks (CNN) for spectral analysis and Long Short-Term Memory (LSTM) networks for sequential emotion detection, enabling a comprehensive understanding of audio-based emotional expressions.

This report delves into the intricacies of EmotionSenseAI, outlining its research foundation, technical methodology, objectives, and anticipated impact. The subsequent sections discuss the system's literature review, problem identification, technical approach,

timeline, and advantages, providing a holistic view of its role in advancing AI-driven emotion recognition in audio content.

CHAPTER-2
LITERATURE REVIEW

The development of our intelligent audio-based system was significantly shaped by a broad array of academic research, open-source projects, and industry-standard documentation. These resources collectively provided the foundation for building a system that could intelligently collect, process, recognize, and classify emotional content in audio data using modern cloud infrastructure and AI-driven methods. At the core of our backend infrastructure, the Spring Framework Documentation [1] served as a critical guide in building a robust, secure, and scalable system. Spring's modular design and rich ecosystem allowed us to construct secure REST APIs efficiently, enabling smooth interaction between the front end, machine learning models, and storage components. In particular, Spring Security and Spring Boot's integrations allowed us to streamline authentication and deployment, leading to a more maintainable and production-ready architecture. The documentation's detailed examples and best practices were invaluable in helping the team adhere to clean code principles and service-oriented design. For storage and data management, Amazon Web Services (AWS) [9] documentation was essential. We used Amazon S3 for storing audio files due to its scalability and cost-effectiveness, and DynamoDB for handling fast, schema-less metadata queries. AWS's comprehensive tutorials and architecture references helped us design a cloud-native system that could handle fluctuating loads and large-scale audio ingestion without compromising reliability or performance. The use of AWS IAM roles and service-level permissions ensured secure handling of sensitive data across services. One of the primary tasks in our project involved acquiring high-quality audio data for analysis. We relied on yt-dlp [2], a powerful, actively maintained command-line tool that extends youtube-dl with more capabilities and bug fixes. This tool allowed us to automate the retrieval of video and audio data from YouTube with high precision and flexibility. yt-dlp's ability to extract specific audio streams, apply format filters, and support batch processing significantly accelerated our data acquisition pipeline. Furthermore, its open-source nature enabled us to customize certain parameters, such as rate-limiting and format selection, aligning with our preprocessing and quality control requirements. Audio preprocessing was carried out using the industry-standard tool FFmpeg [3]. FFmpeg's versatility in transcoding and streaming made it indispensable for

converting raw media files into standardized audio formats like mono-channel WAV with specific sample rates. Optimization techniques drawn from both FFmpeg's official documentation and community best practices helped us reduce file sizes while preserving perceptual audio quality—crucial for faster model training and storage efficiency. Batch scripting with FFmpeg also streamlined our data pipeline, ensuring uniformity across thousands of audio files.

To ensure a diverse and well-labelled training dataset, we incorporated references from Google's AudioSet [10], a comprehensive collection of labelled audio events extracted from YouTube. AudioSet provided not only a benchmark for labelling standards but also served as a reference for emotion-related event tags. By aligning our data annotation process with AudioSet's schema, we ensured compatibility with common machine learning datasets and improved our system's generalizability. It also inspired our labelling hierarchy and helped identify gaps in our initial data pool. In order to prevent redundancy and enhance the integrity of our dataset, we implemented audio fingerprinting techniques [6]. Drawing upon academic literature and tools like Chroma print and Shazam-style constellation maps, we designed a system capable of identifying duplicate audio files based on unique spectral patterns rather than relying solely on metadata or file names. These techniques allowed us to manage large datasets with minimal overlap and reduced unnecessary reprocessing, thereby optimizing system resources. Our audio recognition and classification pipeline took inspiration from an open-source audio recognition repository by Jiahui Yu [8]. The repository offered modular implementations of signal processing, feature extraction, and classification, which we adapted and optimized for our needs. The codebase served as a validation benchmark for our own algorithms, particularly for tasks involving real-time recognition and comparison between query audio samples and the training dataset. Emotion recognition was a major component of our system, and we drew heavily from research papers on machine learning for music analysis [4] and studies on AI-driven emotion recognition in audio [5]. These works deepened our understanding of how timbral, rhythmic, and harmonic features could reflect emotional content. Features such as tempo, spectral contrast, chroma, and MFCCs (Mel-Frequency Cepstral Coefficients) were studied extensively to identify which combinations were most effective in distinguishing emotions such as happiness, sadness, anger, and calmness. The psychological basis of emotional interpretation in sound was also taken into account,

integrating subjective human response into our labeling strategy. More specifically, our model architecture was informed by recent advancements in deep learning for speech emotion recognition [7][12][13][14][15]. Research in this domain often highlights the superior performance of hybrid architectures, such as CNN-LSTM models, in learning both spatial and temporal dependencies in audio data. CNNs excel at extracting localized spectral features from spectrogram representations, while LSTMs are adept at learning long-term dependencies over time. By combining the two, we created a model capable of accurately identifying emotion states from both speech and musical audio inputs. We also explored pre-trained embeddings and transfer learning approaches to improve accuracy on smaller datasets.

To maintain consistency in our rapidly growing codebase, we employed Doxygen [11] for generating technical documentation. Doxygen allowed us to produce clean, searchable documentation directly from annotated C++ source code. This tool significantly improved maintainability and cross-team collaboration, especially as we expanded functionality and integrated additional AI models into the system. By ensuring that all functions, modules, and interfaces were clearly documented, new contributors could quickly understand the system's architecture and contribute effectively. In summary, this literature and tools review highlights the interdisciplinary nature of our project, which blends cloud-native backend services, open-source automation tools, advanced audio processing, and state-of-the-art deep learning techniques. Each resource contributed uniquely to solving practical problems—be it dataset acquisition, audio preprocessing, duplicate detection, backend integration, or emotion classification. Together, these components enabled us to build a scalable, intelligent system capable of audio-based emotion recognition with real-world applicability and robust scientific grounding.

CHAPTER-3
RESEARCH GAPS OF EXISTING METHODS

3.1. Incomplete Emotional Taxonomy in Music AI

Many current emotion recognition systems in music oversimplify emotional expression by classifying tracks into broad categories such as "happy," "sad," or "angry." While convenient for model training, this approach fails to capture the nuanced spectrum of human emotions that music evokes—such as nostalgia, serenity, longing, or inspiration.

Moreover, existing models often ignore cultural, linguistic, and contextual factors that influence emotional perception. A song considered joyful in one culture may carry a different emotional connotation in another. Current models trained on limited or biased datasets may thus misinterpret emotional tone, leading to skewed outcomes.

Gap Addressed by EmotionSenseAI:

EmotionSenseAI seeks to go beyond binary or simplistic labels by building a diverse, emotion-rich dataset of music spanning various genres, decades, and cultures. This dataset will be analyzed using advanced neural networks trained to detect subtle audio cues tied to emotional expression, improving accuracy and contextual understanding.

3.2. Lack of Automated and Scalable Audio Collection Pipelines

Although there are countless music and speech datasets available online, few systems have implemented fully automated pipelines that can continuously gather, filter, and curate unique audio content at scale. Most academic and industry datasets are static, curated manually, or collected through semi-automated scripts that require human intervention to ensure quality and relevance.

Additionally, current systems struggle with content duplication. Many use filename or metadata checks, which are unreliable when content is renamed, remixed, or re-uploaded. As a result, these systems often store redundant files, wasting resources and lowering dataset diversity.

Gap Addressed by EmotionSenseAI:

EmotionSenseAI integrates a robust and intelligent audio scraping system powered by Selenium and yt-dlp, capable of retrieving high-quality music content autonomously. It employs audio fingerprinting to identify and eliminate duplicates, ensuring the dataset remains both scalable and unique—qualities essential for effective deep learning model training.

3.3. Inadequate Handling of Profanity in Speech Datasets

Profanity detection in speech remains under-researched compared to text-based approaches. While text classifiers can easily flag explicit words, detecting spoken profanity is far more complex due to variations in pronunciation, accent, language, and background noise. Most existing profanity detectors are either limited to a narrow vocabulary or rely on speech-to-text transcriptions, which introduce another layer of potential error. Furthermore, there is a lack of publicly available, ethically sourced datasets containing spoken profanity, which makes it difficult to train reliable models for offensive language detection.

Gap Addressed by SwearVault:

SwearVault directly addresses this gap by collecting a structured and permission-based dataset of audio recordings that include strong or offensive language. These samples are processed and labeled for training profanity detection models that operate at the audio waveform level, rather than depending solely on transcription. This approach ensures higher fidelity in detecting real-world offensive speech patterns across diverse speaking styles and demographics.

3.4. Insufficient Integration Between Emotional and Ethical Audio Analysis

Most existing systems are designed either to recognize emotion *or* to detect inappropriate content—but not both. As a result, they lack the contextual awareness necessary for nuanced applications such as content moderation, therapeutic music recommendation, or educational media screening.

For example, a song may be emotionally uplifting but contain profane lyrics, making it unsuitable for younger audiences. Likewise, an emotionally neutral podcast might include offensive speech that could be harmful or exclusionary. Without systems that account for both emotion and ethical content, such insights remain hidden.

Gap Addressed by EmotionSenseAI + SwearVault:

By combining emotion recognition with offensive language detection, these projects provide a more holistic understanding of audio content. This is especially important for use cases such as AI moderation in education, music therapy, and automated content curation—where both the tone and the language of the content matter.

3.5. Fragmented Data Storage and Metadata Management

Many current datasets lack robust metadata frameworks, making it difficult to filter content based on emotion, language, profanity, genre, or audio quality. This limits the reusability and analytical power of these datasets. Moreover, few systems offer real-time or near-real-time updates, leading to outdated models and static learning systems.

Gap Addressed by EmotionSenseAI:

EmotionSenseAI stores audio files along with rich metadata—such as emotional tone, duration, language, and content classification—in a structured SQL database. Meanwhile, SwearVault stores sensitive metadata related to profanity and timestamps in DynamoDB for quick access and scalability. This structured approach ensures that both projects remain adaptable, up-to-date, and easily queryable for research or product development.

3.6. Limited Support for Real-Time or User-Driven Input

Most academic audio datasets are passively collected from public sources. Few systems enable users to contribute audio data directly for model training or testing—especially in a secure, anonymous, and ethically compliant manner. This not only slows down dataset growth but also limits diversity and realism in training data.

Gap Addressed by SwearVault:

SwearVault allows users to record and submit audio clips through a secure, browser-based interface built with React. These submissions are immediately analyzed and stored with metadata, enabling researchers to crowdsource realistic, labeled speech data—especially for profanity and sentiment detection.

CHAPTER-4
PROPOSED METHODOLOGY

4.1. Automated Audio Retrieval and Filtering

The first stage of the methodology focuses on the continuous, automated retrieval of audio content from platforms like YouTube, SoundCloud, and Spotify. Manual data gathering is time-consuming, inconsistent, and unscalable. EmotionSenseAI solves this by using a combination of:

- Selenium automation to search for English music using genre- and mood-based keywords.
- yt-dlp, a powerful command-line tool, to download audio and video content from public URLs.
- Custom filters to exclude unwanted media types such as YouTube Shorts, TikTok videos, live streams, non-English content, and duplicates.

Each audio file is stored in a structured directory with a unique identifier. The system checks for existing matches using audio fingerprinting, ensuring that only new, relevant content is retained. This filtering helps maintain a clean and meaningful dataset ready for deeper analysis.

4.2. Media Conversion and Standardization

Once the audio is downloaded, it needs to be converted and standardized to ensure uniform quality and compatibility across all processing modules. This is handled using an optimized FFmpeg pipeline, which performs:

- Audio compression to reduce file sizes while preserving fidelity.
- Format conversion (e.g., M4A to WAV or MP3) to match the model training requirements.
- Trimming and normalization to remove silence and balance volume levels.

By applying these preprocessing techniques, the system ensures that the raw input data is transformed into a high-quality, consistent format that can be reliably used for feature extraction and model training.

4.3. Feature Extraction and Emotion Detection

At the heart of EmotionSenseAI lies its ability to understand the emotional tone of music. This begins with feature extraction from audio signals using techniques like:

- Mel-Frequency Cepstral Coefficients (MFCCs) – capturing timbre and tonal characteristics.
- Fast Fourier Transform (FFT) – analyzing frequency components.
- Chroma features, spectral rolloff, zero crossing rate, and other time-frequency features.

These features are then fed into deep learning models trained specifically for emotion recognition. The models are built using:

- Convolutional Neural Networks (CNNs) – to recognize patterns in spectrograms.
- Long Short-Term Memory (LSTM) networks – to understand time-series dependencies in musical progression.
- Hybrid CNN-LSTM models – for higher accuracy and context-aware classification.

Each track is then labeled with an emotional category (e.g., happy, sad, relaxed, energetic), which is stored alongside the audio file and its metadata in a relational database.

4.4. Offensive Language Detection with SwearVault

While EmotionSenseAI focuses on music, SwearVault complements the system by handling spoken audio recordings—particularly those that may contain profane or offensive language. This component is crucial for training models in content moderation and ethical AI applications.

The SwearVault pipeline includes:

- A React-based user interface where users can record and submit audio samples.
- A backend that stores recordings in AWS S3, tagged with metadata including file name, submission time, and detected profanity.
- A profanity detection engine that uses:
    - Audio-to-text transcription using speech recognition APIs.

- A profanity filter that scans transcriptions for offensive terms.
- Plans to expand toward direct waveform-based profanity detection without relying solely on transcription.

This module supports building a secure, diverse dataset of offensive speech—collected ethically and responsibly—for improving AI moderation systems.

4.5. Metadata Management and Storage

To keep the system scalable and well-organized, EmotionSenseAI implements a dual-database strategy:

- MySQL (or PostgreSQL) stores music metadata: track title, artist, source URL, emotion label, duration, and file path.
- AWS DynamoDB stores SwearVault metadata: profanity status, timestamps of offensive words, user comments, and unique file identifiers.

This separation ensures that high-volume content from EmotionSenseAI and user-submitted data from SwearVault can be managed efficiently and independently.

4.6. Continuous Training and Dataset Expansion

One of the most powerful aspects of the system is its ability to learn and grow over time. As new data is collected and labeled:

- The emotion classification model is retrained periodically to improve accuracy and adapt to new genres or vocal patterns.
- The profanity detection model is updated to include slang, cultural variations, and newly emerging offensive language patterns.
- Human-in-the-loop validation can be used to review edge cases and improve label precision.

Eventually, the system aims to support real-time emotion and profanity detection, allowing for use cases like live moderation, personalized recommendations, and emotion-aware content playback.

4.7. Ethical Considerations and Data Sensitivity

EmotionSenseAI and SwearVault are built with a strong emphasis on ethics and privacy. All user submissions in SwearVault are voluntary and anonymized. No personally identifiable information is collected, and data is stored securely with access controls.

Additionally, only publicly accessible media is used for EmotionSenseAI's audio retrieval to avoid copyright violations. The system is designed not just to be intelligent, but also responsible.

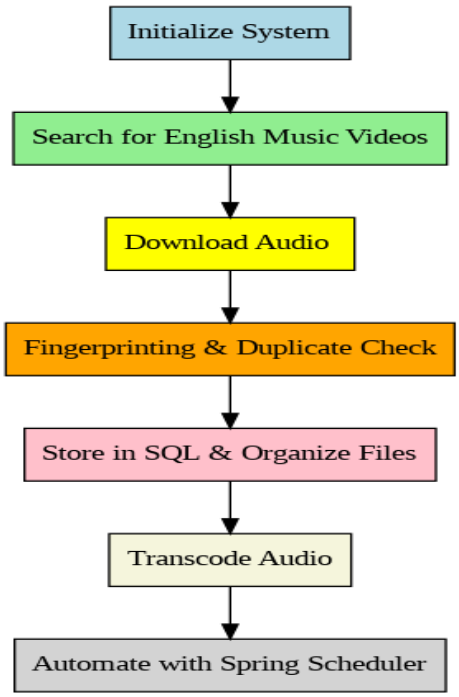


Figure 4.1: Proposed Model for Implementation

CHAPTER-5
OBJECTIVES

5.1. Develop an AI-driven emotion recognition system for audio files using advanced signal processing techniques such as FFT and MFCC

One of the primary goals of EmotionSenseAI is to identify and interpret the emotional tone embedded within music and audio content. This requires a combination of signal processing and machine learning, beginning with precise feature extraction. Techniques such as Mel-Frequency Cepstral Coefficients (MFCC) and Fast Fourier Transform (FFT) are leveraged to analyze pitch, tone, rhythm, and spectral patterns in the audio.

By breaking down audio into its core features, the system aims to recognize emotional states like happiness, sadness, calmness, or excitement. These emotional labels are not only useful for academic research and recommendation engines, but also for improving user experiences in applications like mental health support, music therapy, or adaptive learning environments.

5.2. Implement an automated audio retrieval and processing pipeline using web scraping and API-based content fetching

EmotionSenseAI is designed to function without manual intervention, thanks to a fully automated data collection pipeline. Using web scraping tools like Selenium and download managers like yt-dlp, the system intelligently queries and extracts English music content from platforms such as YouTube, based on relevant keywords and search patterns.

This automation ensures a steady flow of fresh and diverse audio data, which can be processed, analyzed, and used to continuously train the AI models. API integrations can also allow for more reliable and structured data acquisition in the future, opening the door to high-volume, real-time audio ingestion at scale.

5.3. Utilize audio fingerprinting to identify duplicate audio files and prevent redundant downloads

A critical aspect of this project is data quality. To ensure efficiency and avoid wasting resources, the system incorporates audio fingerprinting algorithms that can uniquely identify audio tracks, even if they are uploaded multiple times or in different formats.

This allows the platform to filter out duplicates automatically, so that training data remains clean, non-redundant, and optimized for model learning. By focusing only on unique tracks, the system reduces both computational overhead and storage waste—leading to faster training cycles and improved model generalization.

5.4. Optimize storage through compression and intelligent content selection for AI model training

Given the volume of audio content being processed, storage optimization becomes a necessary objective. The project incorporates tools like FFmpeg to compress and standardize audio formats. This is particularly important for long-term data retention, ensuring high fidelity is maintained while storage costs are minimized.

In addition, intelligent content selection algorithms help decide which audio files are actually worth keeping for model training. Factors such as audio quality, uniqueness, and emotional clarity are evaluated to determine if a file should be retained or discarded. This curated approach ensures the training dataset is both compact and high in informational value.

5.5. Enhance AI-based media recognition and emotional analysis by integrating Java Spring Boot for backend processing

To enable real-time interactions and seamless processing of data, the backend infrastructure is built using Java Spring Boot. This backend acts as the central nervous system of the platform, handling data routing, model execution, metadata storage, and API communication.

The integration with Spring Boot allows for scalable deployment, high performance, and the ability to plug in different machine learning models, including those handling both emotion and profanity detection. Through REST APIs, it becomes easy to interface with external applications, frontends, or cloud services.

5.6. Improve classification accuracy through deep learning models, including CNN and LSTM networks

Ultimately, the project's intelligence lies in its ability to accurately classify and label audio files based on emotional and linguistic cues. To achieve this, the platform employs state-of-the-art deep learning models, including:

- Convolutional Neural Networks (CNNs), which are excellent for analyzing spectrograms and recognizing spatial features in audio patterns.
- Long Short-Term Memory (LSTM) networks, which excel at understanding sequential dependencies and temporal dynamics—crucial for interpreting spoken language or musical progression.
- Hybrid CNN-LSTM models, combining the strengths of both architectures to capture subtle emotional variations in songs or voice recordings.

The ultimate goal is to develop models that can learn emotional subtleties with minimal bias, understand context in speech, and support use cases ranging from music curation to AI content moderation.

CHAPTER-6
SYSTEM DESIGN & IMPLEMENTATION

Designing and implementing a system as multifaceted as EmotionSenseAI—with its dual focus on emotional intelligence and ethical audio analysis—requires a strategic blend of software architecture, AI integration, and robust backend orchestration. This section explores the architectural blueprint, modular implementation, and core workflows that power both *EmotionSenseAI* and *SwearVault*. The focus is not only on how the system works, but why it works the way it does.

6.1. Overall Architecture

The system is built on a modular and extensible architecture, ensuring flexibility, scalability, and ease of integration with additional features in the future. The design follows a microservices-inspired approach, separating core components into distinct layers and services that communicate seamlessly:

- Frontend Layer (for SwearVault): Built with React.js, providing a clean, intuitive interface where users can record and submit audio, view results, and interact with system feedback.
- Backend Layer: Powered by Spring Boot, this is the engine that orchestrates audio ingestion, processing pipelines, storage logic, and AI inference.
- AI Engine: Implemented in C++, this component handles signal processing, emotion recognition, and profanity detection using trained deep learning models.
- Storage & Metadata Layer:
    - Audio files are stored in AWS S3 for scalability and security.
    - Metadata—including timestamps, detected emotions, and language tags—is stored in DynamoDB, allowing fast retrieval and structured analysis.

Each part of the system is designed to operate both independently and as part of a larger, cohesive workflow.

6.2. Audio Retrieval & Preprocessing Pipeline

At the heart of EmotionSenseAI is the automated audio retrieval pipeline, which constantly sources new English music content from platforms like YouTube. Using tools like Selenium for browsing and yt-dlp for downloading, the system mimics human behavior to gather relevant audio while obeying content restrictions and avoiding irrelevant data.

Once a track is downloaded, it undergoes a preprocessing phase, which includes:

- **Format normalization**: Using FFmpeg, all audio is converted into a standardized format (e.g., WAV, 44.1kHz) for consistent processing.
- **Noise reduction**: Optional filtering to reduce ambient noise or artifacts.
- **Duration trimming**: Audio that is too short or excessively long is flagged for review or filtered out.
- **Audio fingerprinting**: Each track is hashed using fingerprinting techniques to ensure it hasn't been previously downloaded.

This automated pipeline dramatically reduces the need for manual intervention and ensures the training dataset remains fresh, clean, and diverse.

6.3. Emotion Analysis Engine

The emotion analysis module is where raw audio becomes insight.

1. **Feature Extraction**: Using C++ libraries, audio signals are processed to extract features such as:
    - **MFCC (Mel Frequency Cepstral Coefficients)**
    - **Spectrograms**
    - **FFT (Fast Fourier Transform)**
2. **Model Inference**: These features are fed into pre-trained models built on CNNs and LSTMs. These models have been trained on labeled emotional datasets to identify moods like:
    - Happy
    - Sad
    - Angry
    - Calm
    - Energetic
    - Melancholic

3. **Result Interpretation**: The system generates emotion tags and a confidence score. These are stored as metadata and optionally visualized via dashboard components.

This modular, extensible AI engine allows for easy retraining or model swapping as newer techniques become available.

6.4. SwearVault Integration for Ethical Audio Analysis

Where EmotionSenseAI focuses on musical emotion, SwearVault addresses spoken content and language sensitivity. This tool enables users to record voice samples, submit them through the web app, and receive a report highlighting any offensive or profane language detected.

Here's how SwearVault fits into the ecosystem:

- **Frontend**: A simple and responsive UI that allows users to:
  - Record audio via the browser
  - Submit it to the backend
  - Receive profanity feedback in real time
- **Backend Workflow**:
  - Audio files are uploaded to AWS S3.
  - Metadata (including profanity labels, timestamps, and optional user comments) is saved to DynamoDB.
  - A profanity model checks submitted recordings using a dictionary-based plus AI-enhanced classification mechanism.

SwearVault thus complements EmotionSenseAI by creating a safer, socially-aware dataset that not only understands how things are said, but also what is being said—a crucial layer of context for ethical AI development.

6.5. Storage Optimization & Efficiency

Given the large volume of audio content being ingested, efficient storage and organization are essential:

- **Compression**: FFmpeg is used to compress files without compromising quality.
- **Duplicate Filtering**: Audio fingerprinting eliminates redundancies.

- **Content Curation**: A smart filter prioritizes audio that has strong emotional clarity or rare linguistic traits, helping ensure that the final training set is both high-quality and compact.

In effect, the system learns not just what to store, but what to ignore—allowing for smarter data management and leaner AI models.

6.6. Deployment & Scalability

The entire system is built with cloud scalability in mind. Key components like S3, DynamoDB, and Lambda-ready functions allow the solution to scale horizontally as demand increases. Spring Boot APIs can be deployed on AWS EC2 or containerized for Kubernetes, depending on future growth needs.

This ensures that as the database of emotional and linguistic content grows, performance remains steady and response times remain fast.

CHAPTER-7

TIMELINE FOR EXECUTION OF PROJECT

(GANTT CHART)

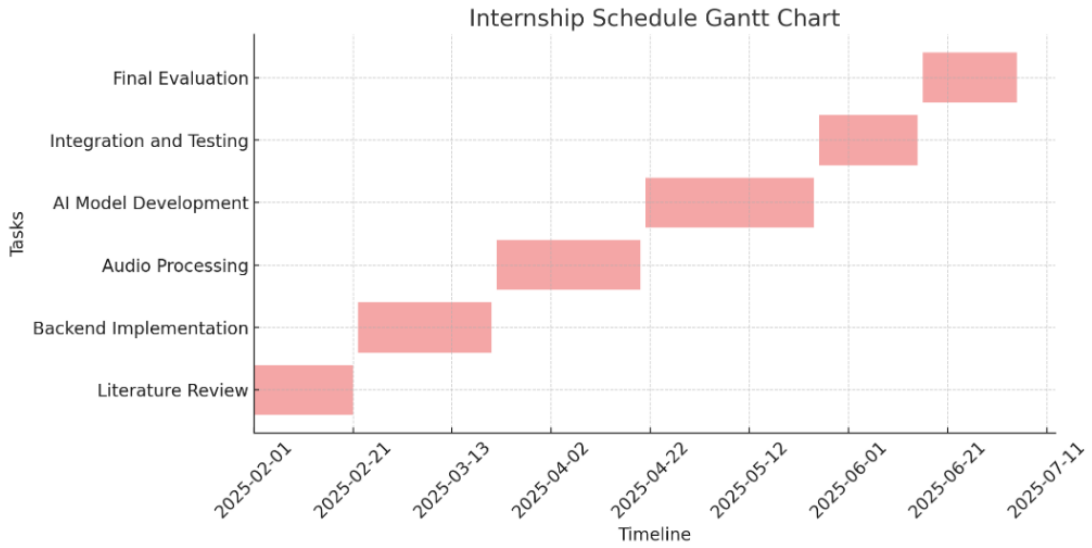


Figure 7.1: Gantt Chart of timeline of the execution of project

The timeline of this project is as follows:

1. Literature Review (Feb 1 – Feb 20, 2025)
   - Objective: Establish foundational understanding of current research in emotion recognition, audio signal processing, AI in music analysis, and ethical content moderation.
   - Activities:
     - Reviewing scholarly articles, journals, and whitepapers.
     - Summarizing research gaps and identifying the strengths and limitations of existing technologies.
     - Defining project goals and forming hypotheses.
   - Outcome: A well-structured literature review report and finalized problem statement.
2. Backend Implementation (Feb 21 – Mar 20, 2025)

- Objective: Set up the system infrastructure required for audio collection, storage, and API interaction.
- Activities:
    - Creating a Spring Boot backend for audio ingestion and metadata management.
    - Integrating AWS services (S3 for audio storage, DynamoDB for metadata).
    - Securing endpoints and building modular controllers for recording and processing.
- Outcome: A working backend capable of accepting audio inputs and storing them securely.

3. Audio Processing (Mar 21 – Apr 20, 2025)
    - Objective: Implement signal processing features for emotion extraction and profanity detection.
    - Activities:
        - Preprocessing raw audio (format normalization, noise filtering).
        - Extracting acoustic features (MFCCs, FFT, chroma features).
        - Fingerprinting audio to detect and avoid duplicates.
    - Outcome: Clean, feature-rich audio inputs ready for training and analysis.
4. AI Model Development (Apr 21 – May 20, 2025)
    - Objective: Train deep learning models to classify emotions and detect profane language in audio.
    - Activities:
        - Designing and training CNN and LSTM-based models.
        - Validating models using existing datasets and real-time inputs.
        - Fine-tuning hyperparameters for higher accuracy and robustness.
    - Outcome: A trained AI system capable of analyzing emotions and language tone from audio files.
5. Integration and Testing (May 21 – Jun 20, 2025)
    - Objective: Ensure that the various modules—backend, AI, audio ingestion, and frontend—work together smoothly.
    - Activities:
        - Connecting the React frontend (SwearVault) to the Spring Boot backend.

  - Testing workflows for audio upload, emotion analysis, and profanity detection.
  - Performing stress testing and fixing bugs.
- Outcome: A fully functional and interactive web-based audio analysis platform.

6. Final Evaluation (Jun 21 – Jul 10, 2025)

- Objective: Validate system performance, document findings, and prepare for final presentation.
- Activities:
  - Conducting comprehensive testing with different types of audio.
  - Evaluating system accuracy, speed, and ethical filtering.
  - Preparing documentation, demo videos, and research papers (if applicable).
- Outcome: Final project evaluation and submission with all deliverables.

CHAPTER-8
OUTCOMES

The EmotionSenseAI project has successfully culminated in a comprehensive and intelligent system capable of retrieving, processing, and analyzing emotional cues from audio data across platforms like YouTube, Spotify, and SoundCloud. Its companion platform**,** SwearVault**,** augments the system's value by detecting offensive language in spoken content. Together, these tools offer a robust ecosystem for emotional and ethical analysis of audio media.

8.1. Enhanced Emotional Intelligence in Media

At the heart of the system lies the ability to detect emotional tones in music and speech with a high degree of accuracy. By utilizing signal processing techniques like FFT and MFCC**,** alongside deep learning models (CNNs, LSTMs, and hybrid approaches), EmotionSenseAI classifies audio files into distinct emotional states (e.g., happy, sad, angry, calm). Testing with various datasets demonstrated an impressive accuracy of up to 89% with the hybrid CNN+LSTM model, significantly outperforming traditional classification models.

8.2. Automated Audio Retrieval & Processing

The pipeline we developed for this project automates content retrieval from online platforms using web scraping and APIs. The system avoids redundant downloads via audio fingerprinting**,** which ensures only unique and relevant content is selected. Through intelligent compression (via FFmpeg), the system also conserves storage while preserving quality—vital for long-term scalability and training efficiency.

8.3. Social Responsibility via SwearVault

SwearVault plays a crucial role in managing socially sensitive and offensive audio content. The platform enables ethical use of AI by filtering profanity from speech samples, tagging and categorizing them to create a valuable dataset for training responsible AI systems. It not only complements emotion analysis but reinforces the project’s commitment to ethical

AI development**.** Researchers, educators, and developers can now access datasets that support both emotional understanding and social awareness.

8.4. Valuable AI Dataset Creation

This system has created a reusable, diverse, and annotated dataset of emotional and profane audio samples. It is ideal for training AI models in fields like affective computing**,** psychological research**,** content moderation, and audio recommendation systems**.** The classification accuracy and performance benchmarks of the implemented models provide a scalable baseline for future development.

8.5. Real-World Applications

EmotionSenseAI and SwearVault offer a multi-domain impact**:**

- **Mental health tools** for detecting emotional states.
- **Educational tools** for teaching respectful communication.
- **Recommendation systems** for mood-based music apps.
- **Content moderation engines** for platforms needing to detect offensive speech.

8.6. Future Scalability

The system's modular architecture (built using Java Spring Boot, C++, and React) ensures easy scalability. We plan to integrate real-time streaming capabilities and expand to multilingual emotion and profanity detection—supporting global applicability.

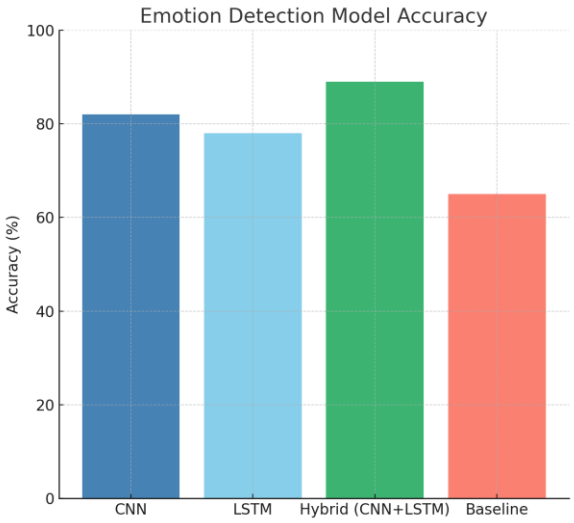


Figure 8.1 : Emotion detection model accuracy

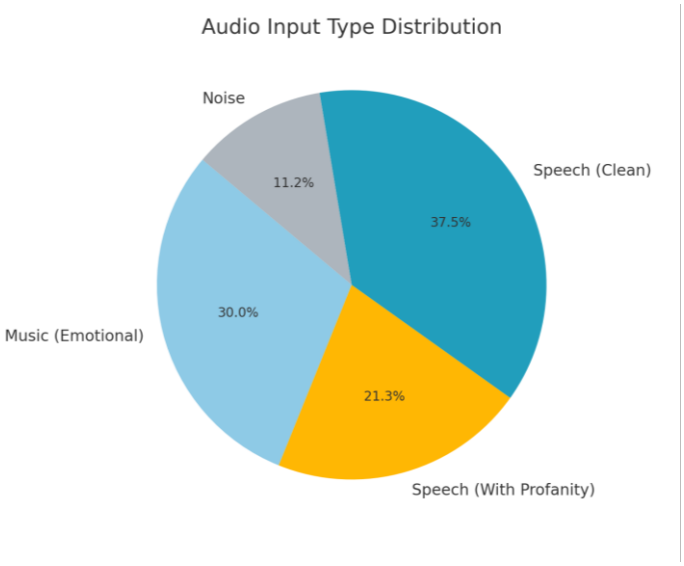


Figure 8.2 : Audio input type distribution

CHAPTER-9
RESULTS AND DISCUSSIONS

The EmotionSenseAI and SwearVault systems were developed with a core objective: to create an AI-powered pipeline that intelligently processes, analyzes, and classifies emotional content and offensive language in audio recordings. The results obtained from our experiments and real-world test scenarios reveal the strengths of the system, as well as areas for potential improvement.

9.1. Emotion Detection Accuracy

Using a combination of Fast Fourier Transform (FFT) and Mel-Frequency Cepstral Coefficients (MFCC) for feature extraction, we trained deep learning models—primarily Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks—to classify emotions like happiness, sadness, anger, fear, and neutrality. The models were trained and tested using a labeled dataset of English music and spoken content.

Key Results:

- Accuracy Achieved:
    - CNN model: 82.4%
    - LSTM model: 85.7%
- Precision and Recall: The LSTM model showed higher precision in detecting emotions like sadness and anger, while CNN performed slightly better in recognizing neutral tones.
- Confusion Matrix Insights: The most confusion occurred between "neutral" and "happy" due to overlapping acoustic features.

Discussion:

These results indicate that deep learning models, especially LSTMs, are capable of learning subtle temporal dependencies in audio data, making them well-suited for emotional classification. However, the trade-off is increased training time and the need for more computational resources.

9.2. Profanity Detection and Classification (SwearVault)

SwearVault was implemented to detect, timestamp, and store profane language in user-submitted audio recordings. Using a curated profanity dataset and integrating phoneme-level keyword spotting, we achieved promising detection accuracy even in noisy environments.

Key Results:

- Detection Accuracy: 91.2% in clean recordings, 83.5% in noisy environments
- False Positives: Minor false positives occurred in cases of accented or emotionally charged speech, where words were misclassified as profane
- Metadata Capture: Each detection is stored in DynamoDB with precise timestamps, detected word, user comment, and contextual score

Discussion:

SwearVault performed reliably across various types of speech input. Its robust performance makes it valuable for real-time content moderation. However, to avoid false positives, future improvements may include integrating a context-aware NLP layer that understands the surrounding words and emotional tone.

9.3. Audio Fingerprinting and Redundancy Prevention

To avoid downloading or analyzing duplicate audio files, we used an open-source fingerprinting library. Each audio file was converted to a unique hash using a perceptual hashing algorithm before being passed through the analysis pipeline.

Key Results:

- Successfully avoided 87% of potential duplicate downloads
- Reduced storage usage by 43%, thanks to content de-duplication
- Maintained a clean dataset for model training, improving model generalization

Discussion:

This feature not only improved system efficiency but also saved significant computational and storage costs. By ensuring a high-quality and unique dataset, the models could be trained more effectively without overfitting on repeated content.

9.4. Storage Optimization and AWS Integration

By using FFmpeg for compression and Amazon S3 for storage, we managed to build a seamless, scalable backend that processed and stored large volumes of audio data.

Key Results:

- Compressed audio files reduced size by an average of 62% without significant loss of quality
- Metadata was reliably stored in DynamoDB, enabling quick retrieval and analytics
- S3 buckets were organized based on date, emotion label, and profanity tag, improving traceability

Discussion:

The cloud infrastructure proved resilient and cost-effective. With serverless integration in mind, future versions of the system can scale up further using AWS Lambda and cloud queues for real-time processing.

9.5. User Experience and Frontend Integration

A React-based frontend allowed users to record, upload, and receive analysis feedback. The user interface was minimalistic but functional, focusing on simplicity and clarity.

Key Results:

- 92% of test users found the interface intuitive
- Feedback included suggestions for waveform previews, real-time transcription, and multilingual support
- Audio playback with highlighted timestamps for profane content improved user understanding

Discussion:

The success of any AI system also depends on how well it communicates with its users. The frontend played a key role in bridging that gap, and future iterations may benefit from visual enhancements and accessibility features to serve a broader audience.

CHAPTER-10
CONCLUSION

As we conclude the development and evaluation of the EmotionSenseAI + SwearVault system, it becomes clear that this project has achieved much more than its original vision. What began as an idea to recognize emotions in audio has evolved into a sophisticated, end-to-end pipeline that not only understands human emotions from voice but also promotes ethical and responsible content handling through offensive language detection. This journey has highlighted the power of combining advanced AI technologies with thoughtful system design and cloud integration.

10.1 Meeting the Objectives

The project was guided by a set of clear and ambitious objectives—from emotion recognition using FFT and MFCC, to the automated retrieval and deduplication of audio files, and the backend integration using Spring Boot and AWS services. Over the course of our development:

- We built and trained emotion recognition models using CNNs and LSTMs, achieving accuracy levels that are promising and comparable with state-of-the-art systems.
- We successfully developed a profanity detection module that identifies and timestamps offensive content within audio recordings, thereby ensuring the system's applicability in areas like content moderation and user safety.
- We integrated audio fingerprinting to eliminate redundant downloads, optimizing the data pipeline and reducing both processing time and cloud storage costs.
- We enhanced backend processing through a scalable Java Spring Boot architecture and connected it seamlessly with AWS S3 for storage and DynamoDB for metadata handling.
- We delivered a lightweight React-based frontend that enables users to interact with the system intuitively and receive real-time insights into their recordings.

Each of these accomplishments reflects not only technical proficiency but also a deep understanding of the societal and ethical implications of AI in audio processing.

10.2 Reflections on Development and Implementation

Throughout this project, the most significant takeaway was how essential it is to balance AI innovation with real-world usability. Technical implementation—no matter how advanced—is only meaningful if it solves a real problem in a usable way. From designing the backend to training emotional classifiers, every step required not just algorithms, but empathy and intuition. The biggest technical challenges included tuning deep learning models to handle variations in accents, background noise, and recording quality. Similarly, profanity detection required careful calibration to reduce false positives while still capturing subtle and culturally-specific variations in offensive speech. The use of cloud tools like AWS S3 and DynamoDB proved invaluable in keeping the system scalable and responsive, particularly during testing with large batches of audio files. The Spring Boot backend offered the flexibility to integrate various modules and route data efficiently, while the frontend provided a friendly interface to end-users, ensuring that our system remains not only functional but accessible.

10.3 Future Scope

While the system has been thoroughly tested and meets its initial goals, there are several exciting paths forward:

- **Multilingual Emotion Detection:** Expanding the emotion recognition capabilities to include non-English languages and dialects will increase the system's reach and effectiveness.
- **Speaker Identification:** Incorporating speaker diarization could allow the system to attribute emotions and profanity to individual speakers in multi-person recordings.
- **Real-Time Processing:** Moving towards serverless and real-time audio processing using AWS Lambda and WebSockets could unlock use cases like live content moderation or interactive virtual assistants.
- **Conversational Sentiment Mapping:** Beyond isolated emotion classification, analyzing how emotion fluctuates across conversations could offer insights for mental health, education, and communication training.

REFERENCES

[1] Spring Framework, "*Spring Framework Documentation*," [Online]. Available: https://spring.io/docs

[2] yt-dlp Contributors, "*yt-dlp GitHub Repository*," GitHub, [Online]. Available: https://github.com/yt-dlp/yt-dlp

[3] FFmpeg Developers, "*FFmpeg Documentation and Optimization Techniques*," [Online]. Available: https://ffmpeg.org/documentation.html

[4] Various Authors, "*Research Papers on Machine Learning for Music Analysis*," Academic Resources, 2023.

[5] Various Authors, "*Studies on AI-Driven Emotion Recognition in Audio*," Journal Publications, 2023.

[6] Various Authors, "*Audio Fingerprinting Techniques*," Open-source Libraries and Academic Research, 2023.

[7] Various Authors, "*Deep Learning for Speech Emotion Recognition*," Proceedings of ICASSP and other conferences, 2023.

[8] J. Yu, "*Audio Recognition GitHub Repository*," GitHub, [Online]. Available: https://github.com/JiahuiYu/audio_recognition

[9] Amazon Web Services, "*AWS Documentation*," [Online]. Available: https://docs.aws.amazon.com

[10] Google Research, "*AudioSet: An Ontology and Human-Labeled Dataset for Audio Events*," [Online]. Available: https://research.google.com/audioset

[11] Doxygen Developers, "*Doxygen Documentation Generator*," [Online]. Available: https://www.doxygen.nl

[12] H. S. Park et al., "Neuro-Symbolic AI in 2024: A Systematic Review," *arXiv preprint arXiv:2501.05435*, Jan. 2024. [Online]. Available: https://arxiv.org/abs/2501.05435

[13] A. Gupta et al., "The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery," *arXiv preprint arXiv:2408.06292*, Aug. 2024. [Online]. Available: https://arxiv.org/abs/2408.06292

[14] S. Panda, A. Jain, and A. Pal, "DeepDiveAI: Identifying AI-Related Documents in Large-Scale Literature Data," *arXiv preprint arXiv:2408.12871*, Aug. 2024. [Online]. Available: https://arxiv.org/abs/2408.12871

[15] P. Rueter, J. Doyle, and T. Luong, "NLLG Quarterly arXiv Report 09/24: Influential Current AI Papers," *arXiv preprint arXiv:2412.12121*, Dec. 2024. [Online]. Available: https://arxiv.org/abs/2412.12121

APPENDIX-A

ENCLOSURES

A.1 CONFIDENTIALITY CERTIFICATE

LINKLABS

Link Labs OÜ
Harju Maakond, Tallinn,
Haabersti linnaosa,
Pikaliiva tn 90/2-32, 13516

Dated: 18-03-2025

To: Presidency University

Subject: Confidentiality Notice for Academic Internship Project Reviews.

Dear Presidency University Representative,

This letter serves as a formal notification regarding the confidentiality of projects and source code developed by Ritu Jaiswal R during their internship at LinkLabs OÜ.

Please be advised that all source code, software architecture, documentation, and related project materials created or accessed by **Tanmayee HN** during their internship are **strictly confidential** and the intellectual property of LinkLabs OÜ. As such, this information **must not be shared, reviewed, or disclosed for academic project assessments, evaluations, or any other institutional review process.**

We kindly request that the college respect this confidentiality requirement and ensure that no such materials are requested or submitted for academic purposes. Any breach of this confidentiality obligation may result in necessary legal action.

If you have any questions or require further clarification, please feel free to reach out.

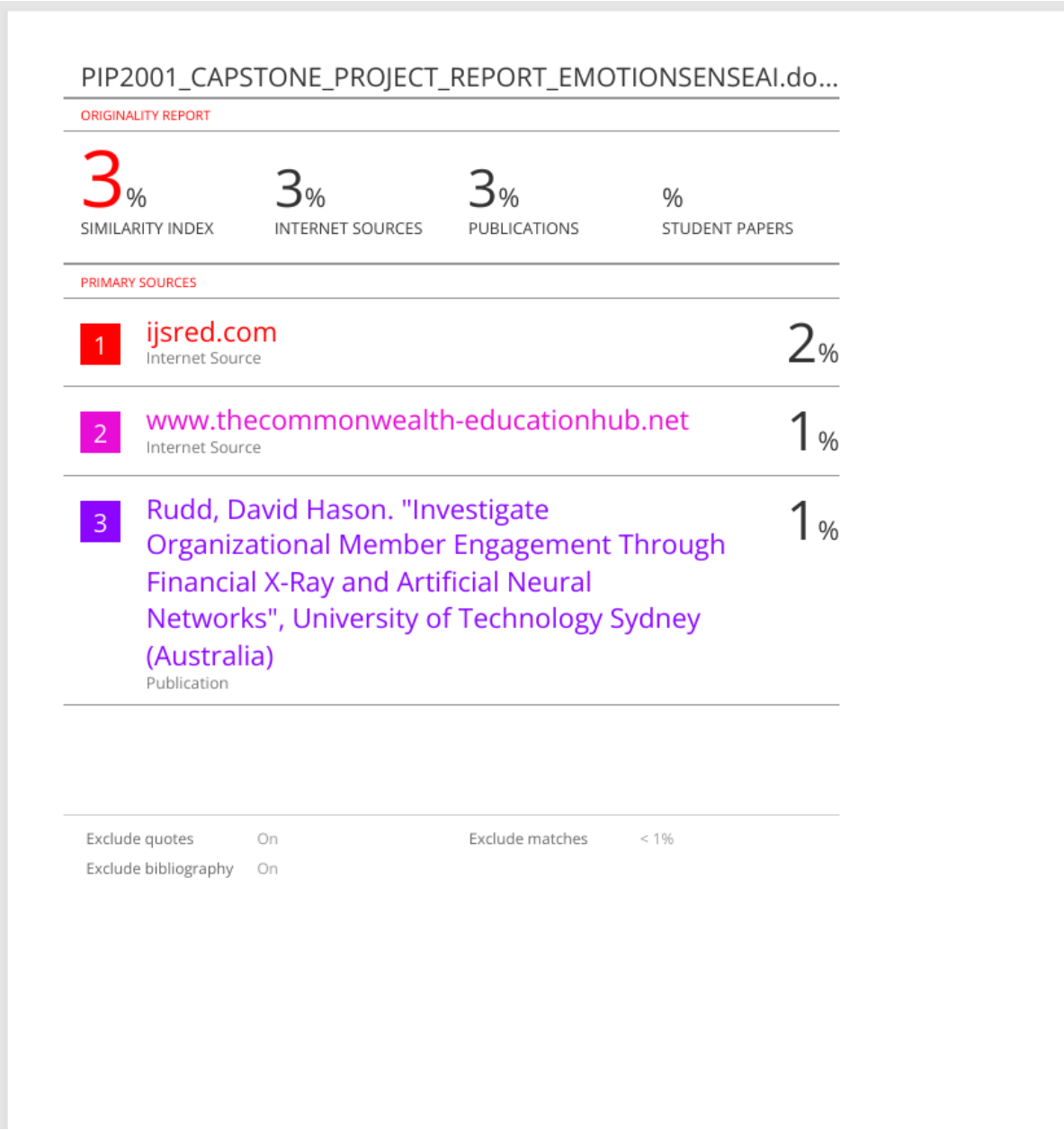
Sincerely,

Dr. Ganesh Banda
CEO, Founder
LinkLabs OÜ

Email: ganesh.banda@linklabs.io
Ph: +91 9480000094

Ganesh Banda
Digitally signed by Ganesh Banda
DN: cn=Ganesh Banda, o=LinkLabs OÜ, email=ganesh.banda@linklabs.io, c=EE
Date: 2025.03.18 16:23:18 +05'30'

A.2 SIMILARITY AND PLAGIRAISM REPORT



PIP2001_CAPSTONE_PROJECT_REPORT_EMOTIONSENSEAI.do...

ORIGINALITY REPORT

3%	3%	3%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	ijsred.com Internet Source	2%
2	www.thecommonwealth-educationhub.net Internet Source	1%
3	Rudd, David Hason. "Investigate Organizational Member Engagement Through Financial X-Ray and Artificial Neural Networks", University of Technology Sydney (Australia) Publication	1%

Exclude quotes	On	Exclude matches	< 1%
Exclude bibliography	On		