

**Department of Electrical and Computer Engineering**

**North South University**

**Directed Research**

***“Cold Start Prediction and Cost-Aware Prewarming Framework for Serverless Applications”***

**Tanzim Rafat 2013485642**

**Rafid Ul Karim 2031359042**

**Md Sadman Sakib Shachyo 2031388642**

**Faculty Advisor:**

**Mirza Mohammad Lutfe Elahi**

**Senior Lecturer**

**ECE Department**

**Summer, 2024**

# APPROVAL

Tanzim Rafat (ID: 2013485642), Rafid Ul Karim (ID: 2031359042), and Md Sadman Sakib Shachyo (ID: 2031388642) from the Electrical and Computer Engineering Department of North South University, have worked on the Senior Design Project titled “FeelBeat” under the supervision of Mirza Mohammad Lutfe Elahi in partial fulfillment of the requirement for the degree of Bachelors of Science in Engineering and has been accepted as satisfactory.

**Supervisor’s Signature**

…………………………………….

**Mirza Mohammad Lutfe Elahi**

**Senior Lecturer**

Department of Electrical and Computer Engineering

North South University

Dhaka, Bangladesh

**Chairman’s Signature**

…………………………………….

**Dr. Mohammad Abdul Matin [mtn]**  
**Professor & Chair**

Department of Electrical and Computer Engineering

North South University

Dhaka, Bangladesh.

# DECLARATION

This is to declare that this project is our original work. No part of this work has been submitted elsewhere partially or fully for the award of any other degree or diploma. All project related information will remain confidential and shall not be disclosed without the formal consent of the project supervisor. Relevant previous works presented in this report have been properly acknowledged and cited. The plagiarism policy, as stated by the supervisor, has been maintained.

Students’ names & Signatures

**1. Student Name1**

Tanzim Rafat

\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_

**2. Student Name2**

Rafid Ul Karim

\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_

**3. Student Name3**

Md Sadman Sakib Shachyo

\_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_ \_

# ACKNOWLEDGEMENTS

The authors would like to express their heartfelt gratitude towards their project and research supervisor, Mirza Mohammad Lutfe Elahi, Senior Lecturer, Department of Electrical and Computer Engineering, North South University, Bangladesh, for his invaluable support, precise guidance and advice pertaining to the experiments, research and theoretical studies carried out during the course of the current project and also in the preparation of the current report.

Furthermore, the authors would like to thank the Department of Electrical and Computer Engineering, North South University, Bangladesh for facilitating the research. The authors would also like to thank their loved ones for their countless sacrifices and continual support.

# ABSTRACT

**FeelBeat**

Serverless computing platforms, such as AWS Lambda, offer on-demand scalability and reduced operational overhead. However, they suffer from a significant drawback known as the **cold start problem**, where infrequently used functions experience high latency due to container initialization delays. This delay can critically impact real-time and latency-sensitive applications. To address this, our project proposes a machine learning–driven framework that predicts the likelihood and delay of cold starts in serverless environments and proactively mitigates them through **cost-aware prewarming** strategies.

The framework leverages real-world-like synthetic invocation data collected through scheduled Lambda function calls. It includes a comprehensive preprocessing pipeline, clustering of time intervals based on cold start behavior, and predictive modeling using classification and regression techniques. By identifying periods of high cold start risk, the system performs selective prewarming actions—effectively maintaining low latency while minimizing unnecessary costs.

Experimental evaluation demonstrates that our framework can predict cold starts with high precision and significantly reduce cold start delays. Simulated cost analysis shows a **dramatic reduction in cold start-related expenses**, achieving near-complete mitigation at negligible prewarming cost. This framework offers a scalable, intelligent, and resource-efficient solution to one of the key performance bottlenecks in serverless applications.

TABLE OF CONTENTS

[LETTER OF TRANSMITTAL **Error! Bookmark not defined.**](#_Toc136448024)

[APPROVAL 2](#_Toc136448025)

[DECLARATION 3](#_Toc136448026)

[ACKNOWLEDGEMENTS 4](#_Toc136448027)

[ABSTRACT 5](#_Toc136448028)

[LIST OF FIGURES 8](#_Toc136448029)

[LIST OF TABLES 9](#_Toc136448030)

[Chapter 1 Introduction 10](#_Toc136448031)

[1.1 Background and Motivation 10](#_Toc136448032)

[1.2 Purpose and Goal of the Project 10](#_Toc136448033)

[1.3 Organization of the Report 10](#_Toc136448034)

[Chapter 2 Research Literature Review 11](#_Toc136448035)

[2.1 Existing Research and Limitations 11](#_Toc136448036)

[Chapter 3 Methodology 12](#_Toc136448037)

[3.1 System Design 12](#_Toc136448038)

[3.2 Hardware and/or Software Components 12](#_Toc136448039)

[3.3 Hardware and/or Software Implementation 12](#_Toc136448040)

[Chapter 4 Investigation/Experiment, Result, Analysis and Discussion 13](#_Toc136448041)

[Chapter 5 Impacts of the Project 14](#_Toc136448042)

[5.1 Impact of this project on societal, health, safety, legal and cultural issues 14](#_Toc136448043)

[5.2 Impact of this project on environment and sustainability 14](#_Toc136448044)

[Chapter 6 Project Planning and Budget 15](#_Toc136448045)

[Chapter 7 Complex Engineering Problems and Activities 16](#_Toc136448046)

[7.1 Complex Engineering Problems (CEP) 16](#_Toc136448047)

[7.2 Complex Engineering Activities (CEA) 16](#_Toc136448048)

[Chapter 8 Conclusions 18](#_Toc136448049)

[8.1 Summary 18](#_Toc136448050)

[8.2 Limitations 18](#_Toc136448051)

[8.3 Future Improvement 18](#_Toc136448052)

[References 19](#_Toc136448053)

# LIST OF FIGURES

[Figure 1: Flowchart of the FER module 16](#_Toc185120137)

[Figure 2: Model Architecture 20](#_Toc185120138)

[Figure 3: Thayer’s mood model 21](#_Toc185120139)

[Figure 4: Flowchart of the Music Recommendation 21](#_Toc185120140)

[Figure 5: Website User Interface 22](#_Toc185120141)

[Figure 6: Confusion Matrix of RMN 24](#_Toc185120142)

[Figure 7: Heatmap of Correlations Between Audio Features 25](#_Toc185120143)

[Figure 8: Screenshot of the Web Application UI (Home Page). 26](#_Toc185120144)

[Figure 9: Screenshot of the Web Application UI (Emotion Detection and Playlist Page). 26](#_Toc185120145)

[Figure 10: Gantt chart showing the project timeline. 30](#_Toc185120146)

# LIST OF TABLES

[Table 1: List of Tools used 18](#_Toc185120190)

[Table 2: Accuracy Table 19](#_Toc185120191)

[Table 3: Budget table. 30](#_Toc185120192)

[Table 4: A SAMPLE COMPLEX ENGINEERING PROBLEM ATTRIBUTES TABLE 31](#_Toc185120193)

[Table 5: A SAMPLE COMPLEX ENGINEERING PROBLEM ACTIVITIES TABLE 33](#_Toc185120194)

# Chapter 1 Introduction

## Background and Motivation

In recent years, **serverless computing** has emerged as a transformative paradigm in cloud infrastructure. It offers developers a simplified, scalable way to deploy applications without managing underlying servers. Platforms such as **AWS Lambda**, **Google Cloud Functions**, and **Azure Functions** have enabled developers to build scalable backend logic using an event-driven model, where code is executed only when triggered. This not only reduces operational complexity but also significantly lowers costs through fine-grained billing.

However, serverless architectures are not without challenges. A key limitation is the **"cold start" latency**—a delay experienced when a serverless function is invoked after a period of inactivity. This delay stems from the time taken to spin up new containers and initialize the runtime environment. In critical use cases like real-time data processing, IoT applications, or user-facing APIs, cold start delays can degrade performance, increase user wait times, and negatively impact user experience.

Several attempts have been made to address this issue. Some providers allow for **provisioned concurrency**, where a fixed number of functions are kept “warm,” but this comes with a **non-trivial cost** and lacks intelligent scheduling. Other approaches include aggressive caching or keeping functions continuously active, which negates the benefits of serverless pricing.

Motivated by these limitations, our research explores whether **machine learning** can provide a smarter alternative: **Can we predict when cold starts are likely to occur, and prewarm functions only when truly needed?** This forms the core challenge and opportunity of our project.

## Purpose and Goal of the Project

The objective of this project is to create an innovative system that connects facial emotion recognition with personalized music recommendations. The project aims to:

1. The primary aim is to create an automated system which is effective for recognizing faces and analyzing emotions in real time. The final version of the system is expected to possess the ability to facilitate recognition and classification of six primary emotions which are happiness, sadness fear, anger, surprise, and disgust. The project intends to set an efficient and dependable base for the general applications of emotion through the use of advanced deep learning techniques that would work under such circumstances to deliver consistent performance.
2. Enhancing the accuracy of emotion recognition beyond the existing benchmark of 76.8% is another key goal of this project. To achieve this, the system will leverage advanced methodologies, such as transfer learning and ensembling, which combine the strengths of multiple models to improve performance. Fine-tuning these models will enable a more precise and nuanced understanding of facial expressions, pushing the boundaries of current emotion recognition capabilities.
3. The project also aims to embrace the multifaceted nature of emotions by allowing users to customize their music preferences within specific emotional categories. For example, a user feeling "happy" might choose between soothing melodies or high-energy tracks to match their mood more closely. This approach ensures a personalized and comprehensive music recommendation experience, catering to diverse emotional states and user preferences.
4. Real-time emotion detection and instant music recommendations form a core objective of the project. By bridging the gap in existing systems, this initiative seeks to provide dynamic and responsive music suggestions based on the user's immediate emotional state. This capability will enhance the overall listening experience, ensuring that the music resonates with the user’s current feelings and creates a seamless interaction.
5. Finally, the project will culminate in the development of a web application that integrates the facial emotion recognition system with a dynamic music recommendation engine. The platform will feature an intuitive and user-friendly interface, simplifying interactions and making mood-based music recommendations easily accessible. This web application will serve as a practical and engaging medium for users to experience the benefits of emotion-driven music personalization.

## Organization of the Report

This report is organized into eight chapters. Chapter 2 reviews existing research and identifies limitations. Chapter 3 explains the methodology and system design. Chapter 4 presents experimental results and analysis. Chapter 5 discusses societal and environmental impacts. Chapter 6 outlines project planning and budgeting. Chapter 7 identifies complex engineering challenges. Finally, Chapter 8 concludes with a summary and future directions.

# Chapter 2 Research Literature Review

## 2.1 Existing Research and Limitations

**“Facial Expression Recognition with CNN-LSTM”**

This paper combines CNN and LSTM for facial expression recognition. Using the Japanese Female Facial Expression (JAFFE) dataset, the authors augmented the data to address its small size, generating multiple samples from each image. The model was built using a 6-layer CNN for feature extraction and LSTM for recognizing expressions. The CNN-LSTM model achieved an accuracy of 86.42%, an improvement over the standalone CNN's 84.25%. The hybrid approach leverages CNN's feature extraction and LSTM's sequential learning capabilities. The confusion matrix scores indicated better performance in classifying anger, fear, and neutral expressions compared to CNN alone. Despite these advancements, the study's limitation lies in its reliance on a small dataset, which may not generalize well to diverse real-world scenarios.

While the accuracy on the JAFFE dataset was impressive, the dataset itself is limited in diversity and size, raising concerns about the model's robustness in real-world scenarios. Furthermore, the inclusion of LSTM increases the model's computational requirements, complicating deployment in real-time systems.

**“Facial Expression Recognition with Deep Learning Improving on the State of the Art and Applying to the Real World”**

This paper seeks to improve FER accuracy and provide a real-time mobile web application. It used FER2013 as the major dataset, with CK+ and JAFFE as supplemental datasets. The model achieved the highest accuracy of 75.8% by employing an ensemble technique. Transfer learning approaches utilizing VGG16, SeNet50, and ResNet50 considerably enhanced performances. The mobile web app was created with TensorFlow.js, React.js, and face-api.js, and it achieved 69.8% accuracy with a recognition time of 40 milliseconds.

Despite these advances, the model's accuracy can still be improved, particularly in real-time applications where illumination and facial occlusions might have an impact on performance.

**“Facial Expression Recognition Using Residual Masking Network”**

This research presents a novel deep architecture for FER problems that combines a Deep Residual Network with an Unet-like design. The Residual Masking Network achieved cutting-edge accuracy on the FER2013 and VEMO2020 datasets. Preprocessing consisted of scaling photos to 224x224, turning them to RGB, and using augmentation algorithms. The model processed an average of 100 frames per second, making it appropriate for real-time applications. The suggested network beat existing models, with an accuracy of 74.14%. Despite these advances, the study emphasizes the importance of further improving FER2013's handling of imbalanced datasets and incorrect samples.

**“Modified VGG-19 Architecture for Emotion Recognition”**

This study proposed an enhanced version of the VGG-19 architecture by integrating segmentation blocks with convolutional layers to refine feature extraction. The segmentation blocks allowed the model to focus on critical facial regions, improving the recognition of subtle emotional cues. Evaluated on the FER2013 dataset, the modified architecture achieved an accuracy of 75.97%, highlighting the impact of feature refinement on performance.

The approach emphasizes the importance of isolating relevant features for better classification, which aligns with the need for accurate emotion detection in music recommendation systems. Despite the improved accuracy, the model's complexity increases computational requirements, posing challenges for real-time applications.

**"Facial Expression Recognition with Deep Learning"**

This paper explored ensemble learning techniques by combining pre-trained models such as ResNet50, VGG16, and SeNet50 for facial expression recognition. By utilizing transfer learning, data augmentation, and class weighting, the ensemble achieved an accuracy of 75.8% on the FER2013 dataset. The use of multiple pre-trained architectures allowed the system to leverage diverse feature extraction capabilities, resulting in robust classification performance.

The ensemble method demonstrates the potential of combining different models for robust emotion detection, which could be applied to improve music mood classification systems. Ensemble learning, while accurate, is computationally intensive, making it unsuitable for lightweight or real-time applications.

**“Music Mood Classification (Nuzzolo, 2015)”**

Nuzzolo (2015) explored music mood classification using digital signal processing (DSP) and music theory techniques, leveraging Thayer’s Energy-Stress Model to categorize songs into eight emotional states, including Happy, Calm, and Depressed. Key musical features such as rhythm, intensity, timbre, and pitch were analyzed to determine a song's mood. Faster tempos and higher pitch were associated with high-energy moods like Happy, while slower tempos and softer tones indicated calmer moods like Depression. The study also introduced methods for quantifying these features, such as beat spectrum analysis for rhythm, RMS for intensity, and spectral irregularity for timbre, achieving high classification accuracy for moods such as Energetic (94.44%) and Calm (94.11%).

The research demonstrated the potential of DSP in classifying music moods, providing a robust framework for organizing digital music collections or improving recommendation systems. The experiments validated the effectiveness of the model in distinguishing between high-energy and low-energy moods, highlighting its utility in mood-based categorization.

However, the study had notable limitations. It relied on static thresholds for feature classification, which restricted adaptability to diverse musical styles or hybrid emotional states. Additionally, emotional responses to music vary among individuals, making universal classification challenging. The computational complexity of DSP techniques also posed challenges for real-time applications, limiting the system's scalability.

# Chapter 3 Methodology

## 3.1 System Design

## The system design of FeelBeat employs facial emotion recognition in combination with the dynamic music recommendation engine, hence allowing users to be given the music they are likely to be interested in according to the mood detected. The structure includes two main modules:

## **Emotion Detection Module:** This part is responsible for the usage of a Residual Masking Network (RMN) that can classify emotions such as happiness, sadness, anger, surprise, fear, and disgust. Captured or user-uploaded facial images are processed to produce real-time mood predictions.

## **Music Recommendation Module:** This module makes use of Spotify's audio features like valence, energy, tempo, and danceability to assemble the specified playlists based on detected emotions. It drives a content-based filtering approach, which is enhanced by cosine similarity, to match songs with moods of users dynamically.

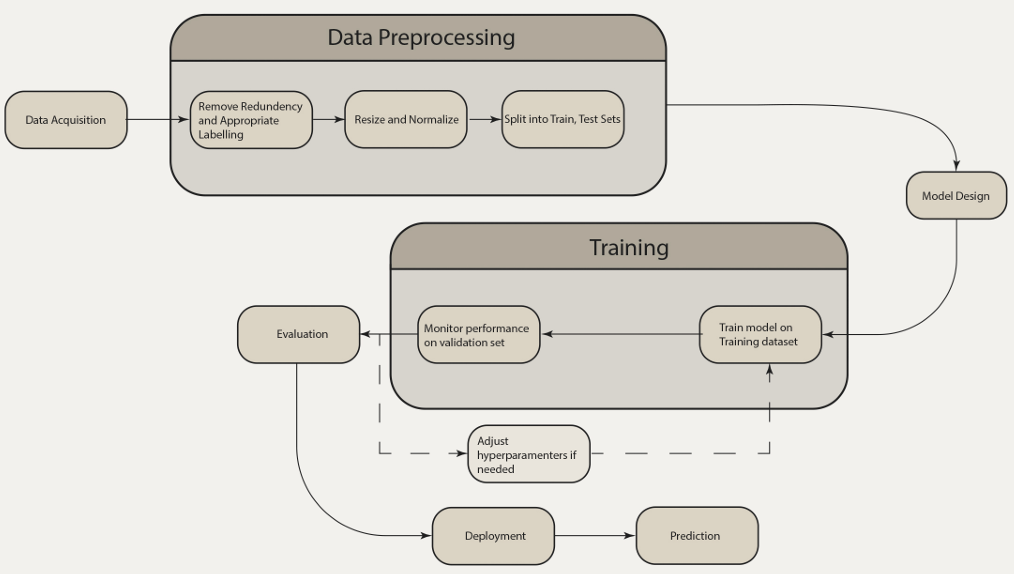


Figure 1: Flowchart of the FER module

## 3.2 Hardware and/or Software Components

The project relies exclusively on software components:

**Backend Development:** Django serves as the primary framework, managing API integration and communication between the frontend and the RMN model.

**Frontend Development:** Django’s templating engine is used to build an intuitive interface for user interactions, including image uploads and playlist displays.

**Machine Learning Tools:** TensorFlow and Keras are used to train and deploy the RMN model, while Spotify APIs provide the necessary song datasets and audio features for the recommendation module.

**Datasets Used:**

1. **FER2013 Dataset:** The FER2013 dataset is a large, publicly available dataset that contains 35,887 grayscale images of faces, each of which measures 48x48 pixels. There are seven types of images: happiness, sadness, anger, surprise, fear, disgust, and neutral. It was used to train and validate the Residual Masking Network (RMN) by taking the advantage of its comprehensive coverage of the basic emotion set for robust emotion recognition. The dataset’s having a collection of facial recognition that contributes to the generalization of the dataset to various subjects, yet the preprocessing step is paramount to the resizing and recoloring of the images so they can be having in the model’s input requirements.
2. **Spotify Dataset:** This dataset comprises audio features extracted using Spotify’s Web API. Key attributes, such as energy, valence, acousticness, tempo, and danceability, were utilized to classify music into mood categories. These features were mapped to emotional states using Thayer’s Energy-Stress Model, enabling accurate alignment of songs with the detected moods. Spotify's enormous music library has the versatility of offerings from which one can come up with appealing personalized playlists fitting the moods of users.

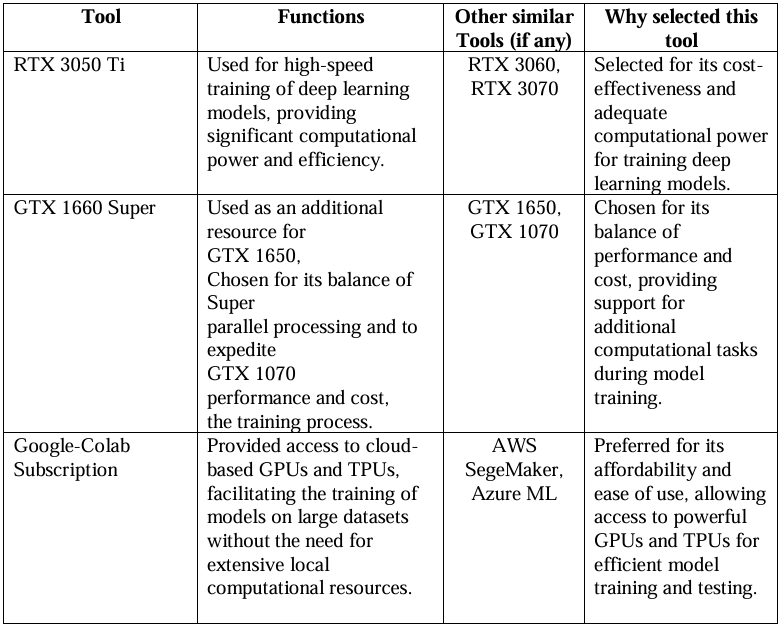


Table 1: List of Tools used

## 

## 3.3 Hardware and/or Software Implementation

The project began with an extensive evaluation of various models for facial emotion recognition:

**Baseline Model:** A simple CNN with four convolutional layers and a softmax output achieved 65% accuracy. Although basic, it provided a benchmark for comparison.

**ResNet50:** A deep residual learning model with 50 layers achieved 72.5% accuracy when fine-tuned on FER2013, demonstrating the effectiveness of residual connections in feature extraction.

**VGG19:** The VGG19 model attained 71% accuracy, presenting a balance between complexity and performance, although on a shallow architecture.

**Ensemble Models:** Combining multiple architectures like ResNet50, VGG16, and SeNet50 resulted in an accuracy of 72.16%, leveraging the strengths of each model.

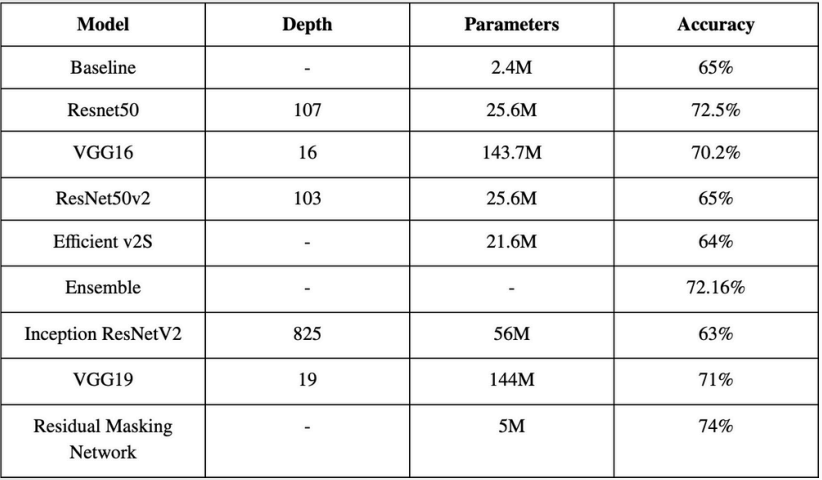


Table 2: Accuracy Table

Ultimately, the Residual Masking Network (RMN) was selected for the model due to its performance superiority, with 74% accuracy on FER2013 tests. The architecture incorporates segmentation blocks and masking layers, allowing the model to prioritize the facial regions of interest (e.g., eyes, mouth) in an aid to improve emotion detection. The lightweight approach of RMN (5 M parameters) enables computational efficiency and is suitable for real-time applications.

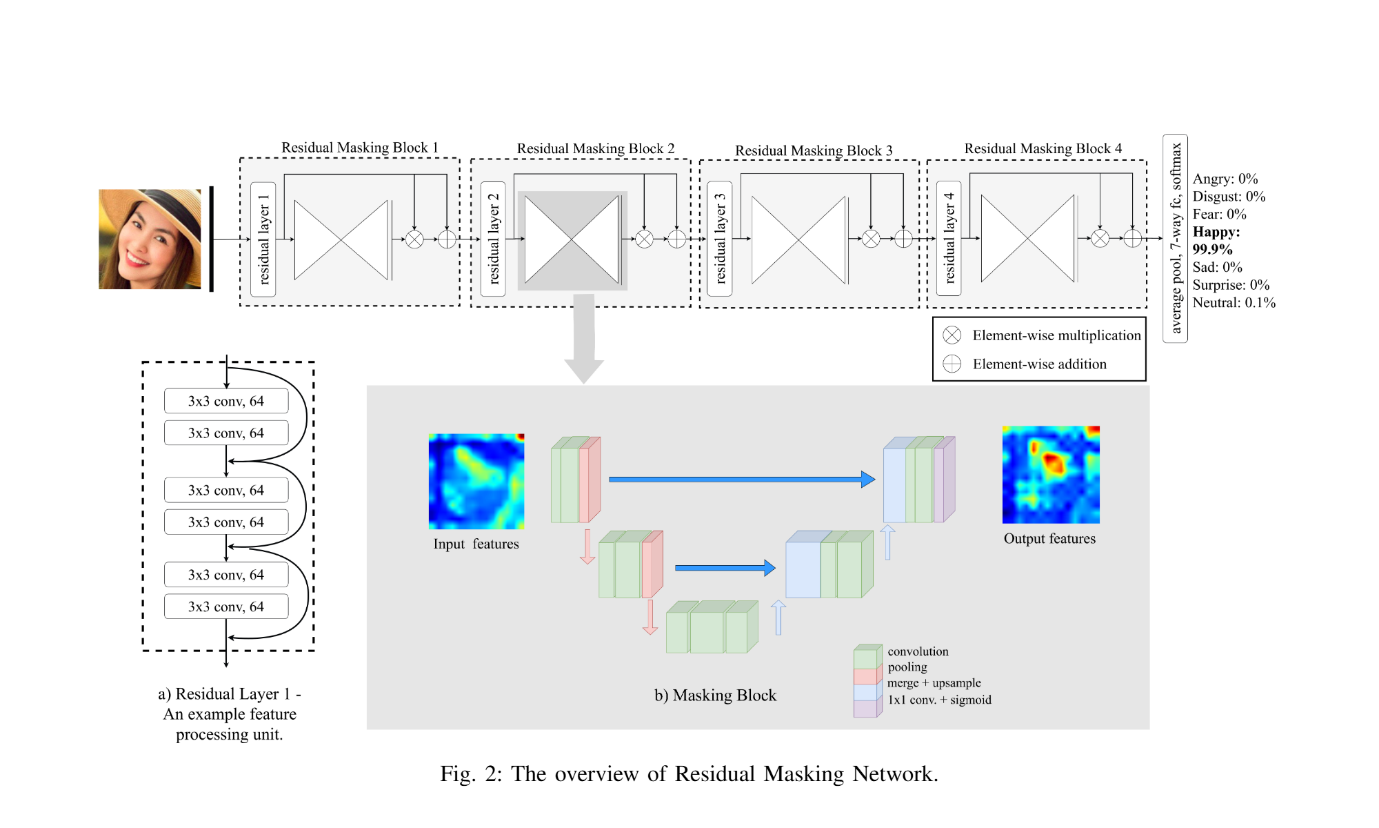


Figure 2: Model Architecture

Images were resized to 224x224 pixels and converted to RGB to meet the RMN’s input specifications. Data augmentation techniques, including rotation, flipping, and scaling, were employed during training to improve robustness. The RMN model was trained over 100 epochs using the Adam optimizer, achieving consistent performance across validation datasets.

The recommendation system uses Spotify's audio features, mapped to moods using Thayer's Energy-Stress Model. Key features such as energy, valence, and tempo were categorized into emotional groups, such as Happy, Sad, and Calm. A content-based filtering algorithm employing cosine similarity was used to recommend songs most aligned with the detected mood.

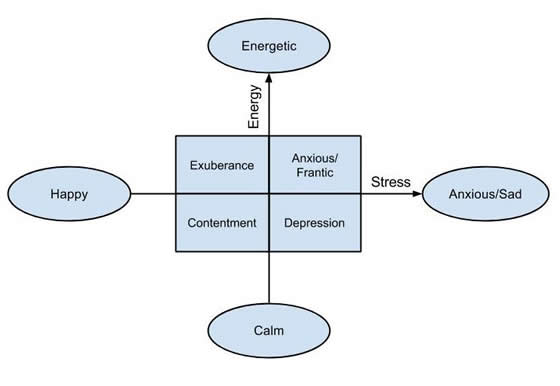


Figure 3: Thayer’s mood model

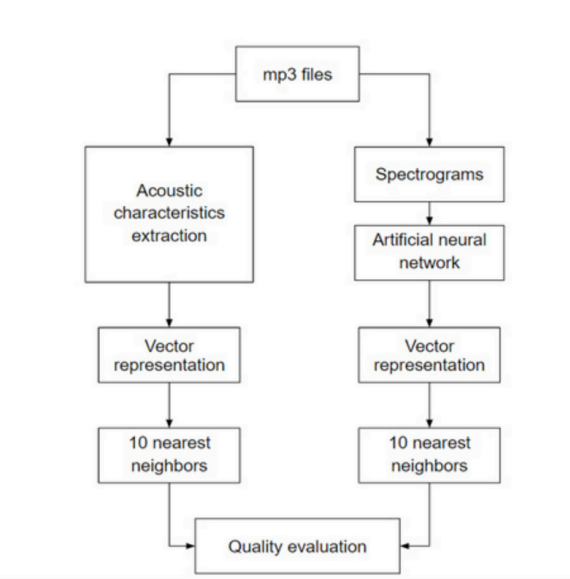


Figure 4: Flowchart of the Music Recommendation

The web application allows its users to upload or capture facial images, the backend processes these images to extract emotions, and a playlist is dynamically created based on whichever emotion is detected. The frontend supports smooth and their responsive interface for the convenience of the user.

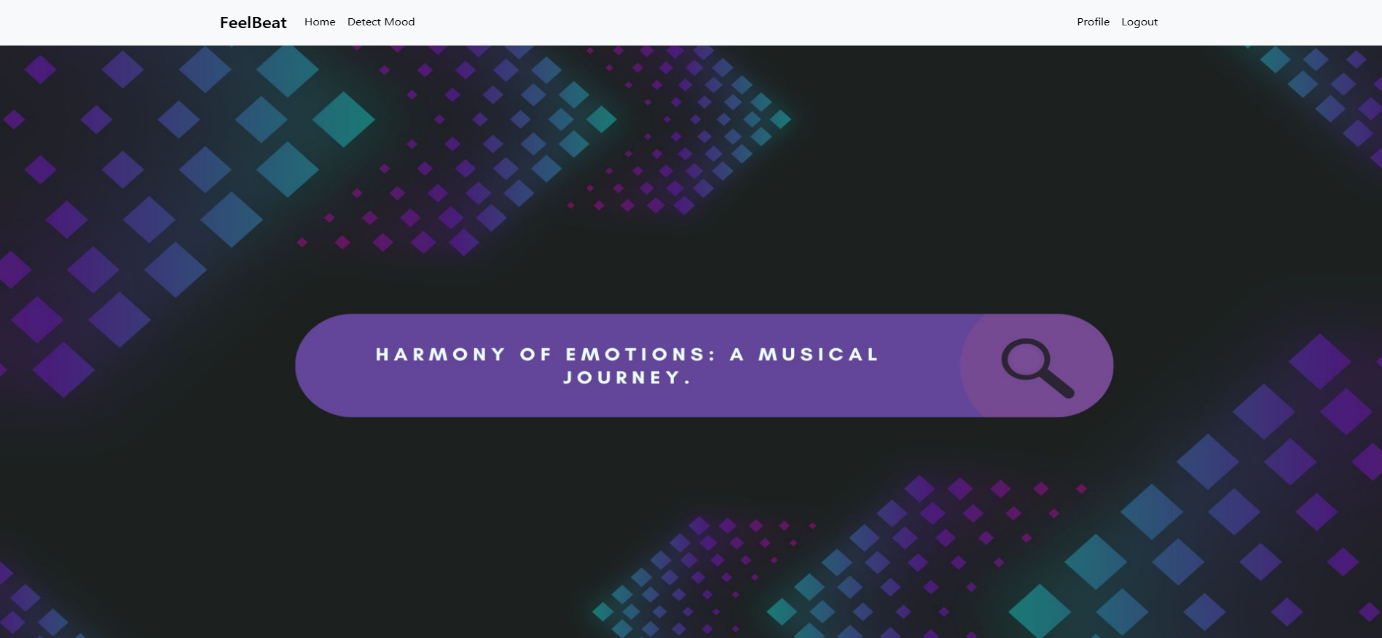


Figure 5: Website User Interface

# Chapter 4 Investigation/Experiment, Result, Analysis and Discussion

**Experiments Performed:**

The main goal of FeelBeat's experiments was to provide a robust emotion recognition system and merge it with a music recommendation module so that a real-time personalized playlist could be generated. Various models were tried for facial expression recognition; the Residual Masking Network (RMN) was selected in the end because it had greater accuracy.

At the same time audio features from Spotify were utilized for music classification with songs categorized by mood utilizing Thayer's Energy-Stress Model.

**Emotion Recognition Experiment:** The RMN was trained on the FER2013 dataset, consisting of grayscale facial images resized to match the model's input requirements. Data augmentation techniques, such as rotation and flipping, were applied to increase dataset diversity and enhance robustness. The RMN architecture, as shown in Figure 1, incorporated segmentation and attention mechanisms to focus on critical facial regions like the eyes and mouth, yielding improved accuracy for emotion detection.

**Music Recommendation Experiment:** Audio features from the Spotify dataset, including valence, energy, and tempo, were analyzed and categorized into emotional states. A content-based filtering approach using cosine similarity was employed to match songs with detected moods.

**Results:**

Emotion Recognition Accuracy The RMN achieved a test accuracy of 74%, outperforming other models such as ResNet50 (72.5%) and VGG19 (71%). The confusion matrix in Figure 2 highlights the classification performance across seven emotional categories. The matrix reveals that the model performed well in detecting emotions such as happiness and anger but exhibited moderate misclassifications for subtle emotions like fear and disgust.

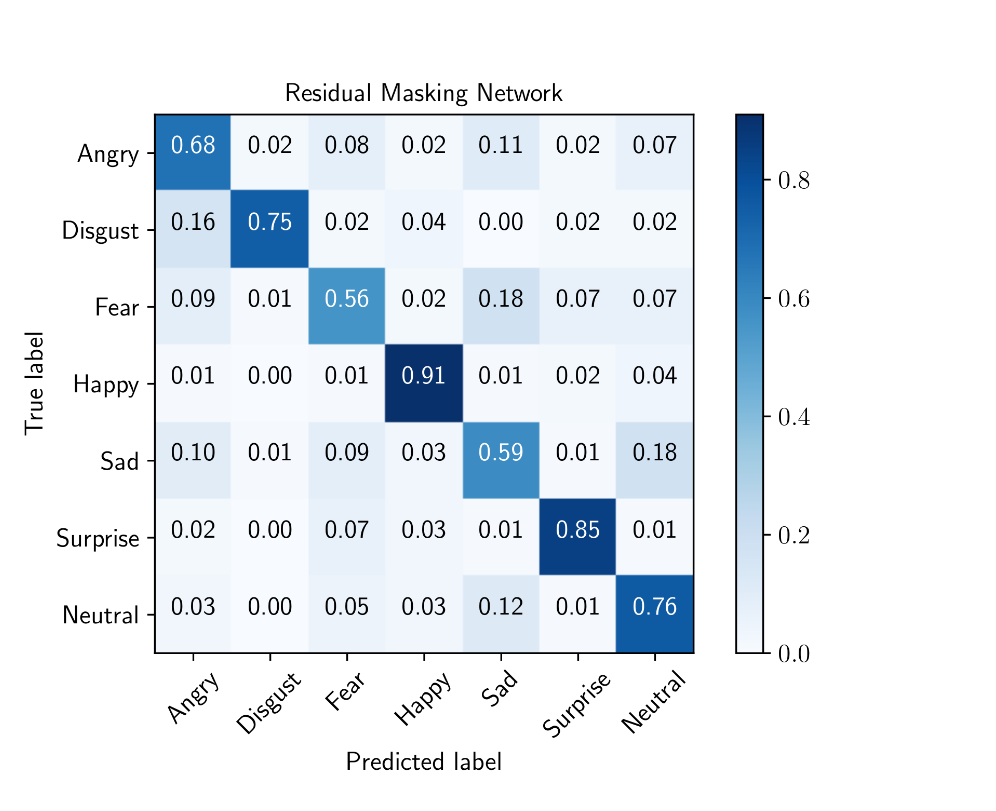


Figure 6: Confusion Matrix of RMN’s Classification Performance

The y-axis represents the true labels, while the x-axis represents the predicted labels. Strong diagonal values indicate accurate predictions, while off-diagonal values highlight misclassifications.

**Music Recommendation Results:** The music recommendation system successfully mapped detected emotions to curated playlists, ensuring that the recommended tracks matched the emotional state of the user. Key audio features such as energy, valence, danceability, and loudness were utilized to establish connections between emotions and music.

A heatmap, as shown in Figure 3, illustrates the correlations between these audio features. This visualization highlights how certain features interact to align with different emotional categories. For example, energy and valence exhibit a strong positive correlation, aligning high-energy, high-valence songs with emotions like happiness. Similarly, features like acousticness and loudness demonstrate inverse relationships, mapping low-energy, acoustic tracks to emotions like sadness.to emotions like sadness.

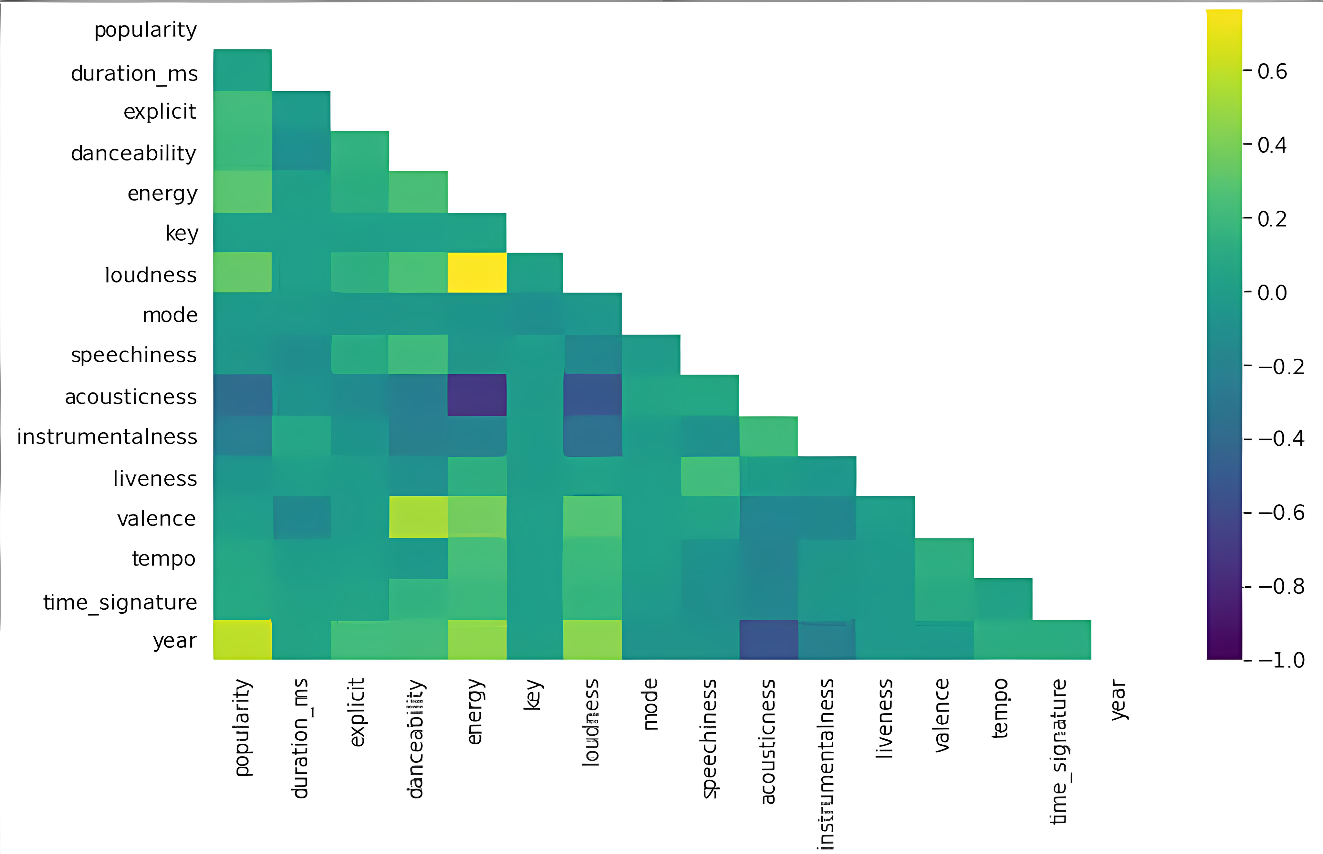


Figure 7: Heatmap of Correlations Between Audio Features

The heatmap displays the interrelationships among audio features, with brighter colors indicating stronger correlations and darker colors showing weaker or inverse relationships. This analysis provided a solid foundation for the recommendation system to align playlists with emotional states effectively.

**Web Application Result:**

The web application developed for FeelBeat provides users with an efficient and effective experience of emotion detection and music recommendations. Users can upload any facial image or use a webcam to capture a real-time image for emotion recognition. Once the emotion is detected, the system generates a personalized playlist based on the user's emotional state. The application interface is well-designed, clearly labeled sections for image upload, emotion display, and playlist recommendations.

The web application performed efficiently, providing quick responses without significant latency. The integration of the RMN model for emotion detection and the Spotify API for music recommendations enabled real-time emotional analysis and accurate playlist generation.

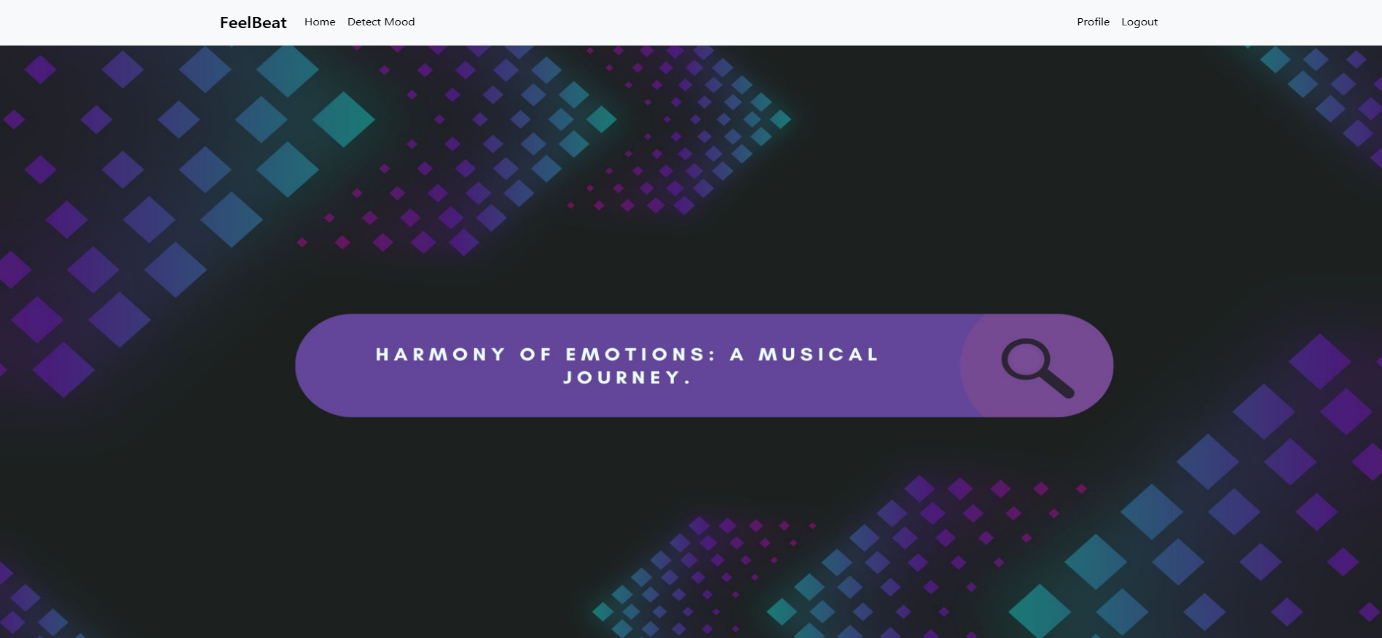


Figure 8: Screenshot of the Web Application UI (Home Page).

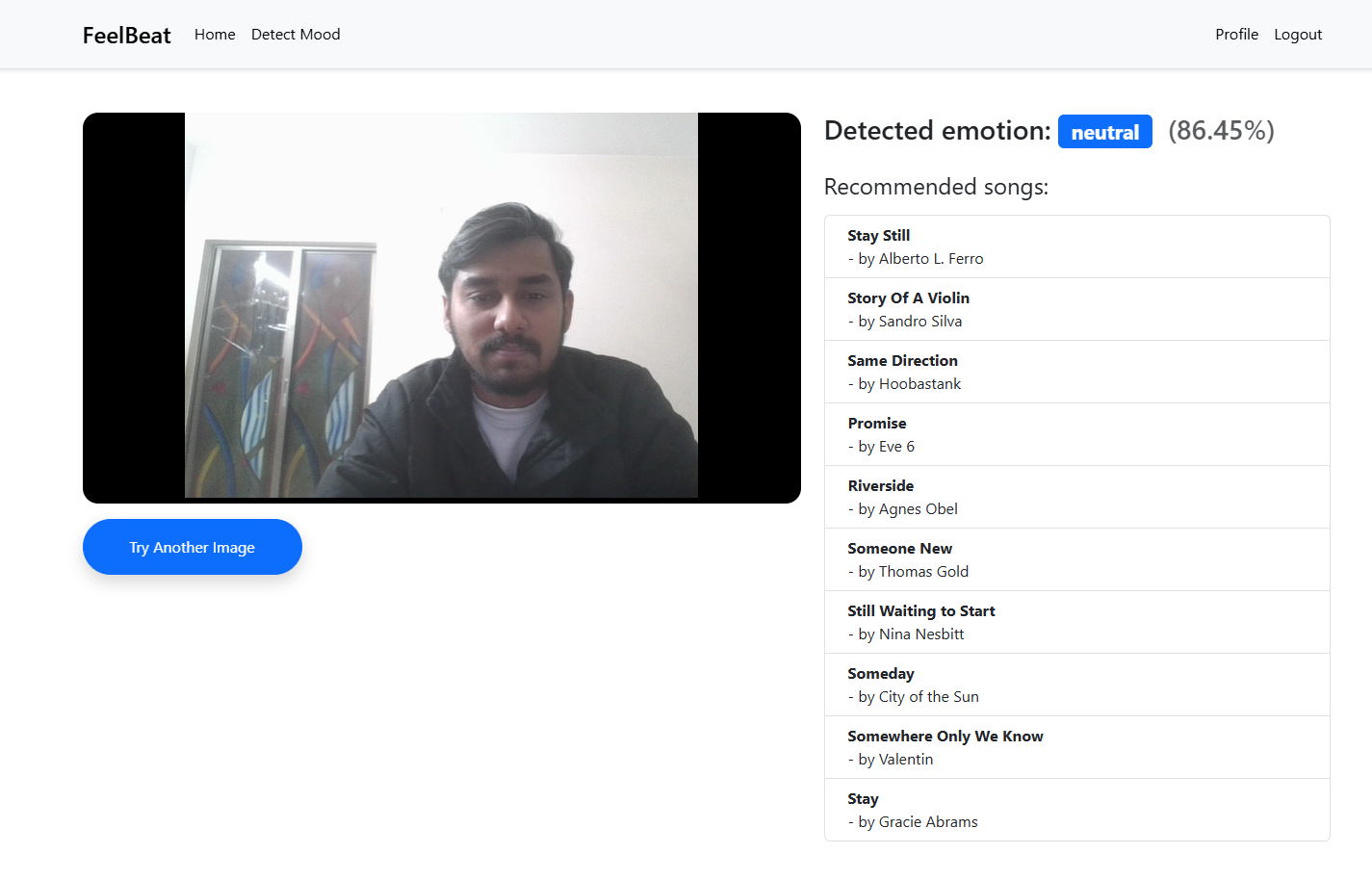


Figure 9: Screenshot of the Web Application UI (Emotion Detection and Playlist Page).

**Discussion:**

The results from the emotion recognition system show that the RMN is relatively effective at identifying primary emotions such as happiness and anger, with some difficulties in detecting more nuanced emotions such as fear and disgust. This might be attributed to the complexity of facial expressions and the variability in how individuals display these emotions.

In terms of the music recommendation system, the successful correlation of emotions with audio features such as energy and valence demonstrates the potential of creating playlists that match the emotional state of users. The heatmap in Figure 3 offers a clear view of how different audio features correlate with each other, providing insights into how these features interact to influence the emotion associated with the song. For instance, the positive correlation between energy and valence suggests that happy, high-energy tracks are often upbeat and bright in tone, while low-energy tracks are associated with sadness or melancholy.

# Chapter 5 Impacts of the Project

## 5.1 Impact of this project on societal, health, safety, legal and cultural issues

FeelBeat has the potential to improve emotional health and personal well-being, which could lead to beneficial changes in society. Matching music to a user's current mood can be a useful tool for emotional support, particularly for those who are depressed, stressed, or anxious. The therapeutic effects of music therapy have long been acknowledged, and FeelBeat can be used as an approachable, daily tool to enhance mental health. For example, it can help individuals manage their emotions by offering calming music when they are feeling anxious or uplifting tunes when they are feeling down.

On a larger social scale, FeelBeat may make customized music therapy more widely available, allowing individuals from various backgrounds to interact with music in ways that directly enhance their emotional health. Since FeelBeat provides a low-cost, scalable solution to assist people in better managing their emotional health, this is particularly important for communities without access to conventional mental health treatments.

In terms of cultural influence, FeelBeat can assist overcome gaps in understanding human emotions across different cultural contexts by providing a universal approach to emotion-based music choices. Because emotions are a universal experience, the initiative could aid in the development of empathy and cultural understanding by delivering music that resonates emotionally with people from all backgrounds.

## 5.2 Impact of this project on environment and sustainability

FeelBeat is a software-based project, which inherently minimizes its environmental impact. The project does not depend on physical hardware, making it a more environmentally friendly choice in comparison to applications that require extensive hardware. Additionally, by prioritizing digital music streaming and suggestions, FeelBeat promotes a transition away from physical music formats like CDs, lowering waste and energy use linked to production and distribution.

The environmental impact of FeelBeat could also be seen in its potential for promoting eco-conscious behavior. For example, the app could eventually incorporate features that encourage users to engage with eco-friendly playlists—such as those featuring nature sounds or songs related to environmental sustainability. By promoting music that connects listeners with nature or social causes, FeelBeat could help raise awareness of environmental issues and inspire positive behavioral changes.

Moreover, FeelBeat operates on cloud-based platforms, which, when paired with renewable energy sources, further reduces its carbon footprint. Given its potential for integration with music streaming services like Spotify, FeelBeat can also take advantage of streaming platforms' ongoing efforts to reduce their own environmental impact, including carbon offsetting and energy-efficient data centers.

Overall, FeelBeat is designed to be both socially responsible and environmentally friendly, promoting sustainability through digital innovation while offering an accessible, low-resource solution for emotional well-being.

# Chapter 6 Project Planning and Budget

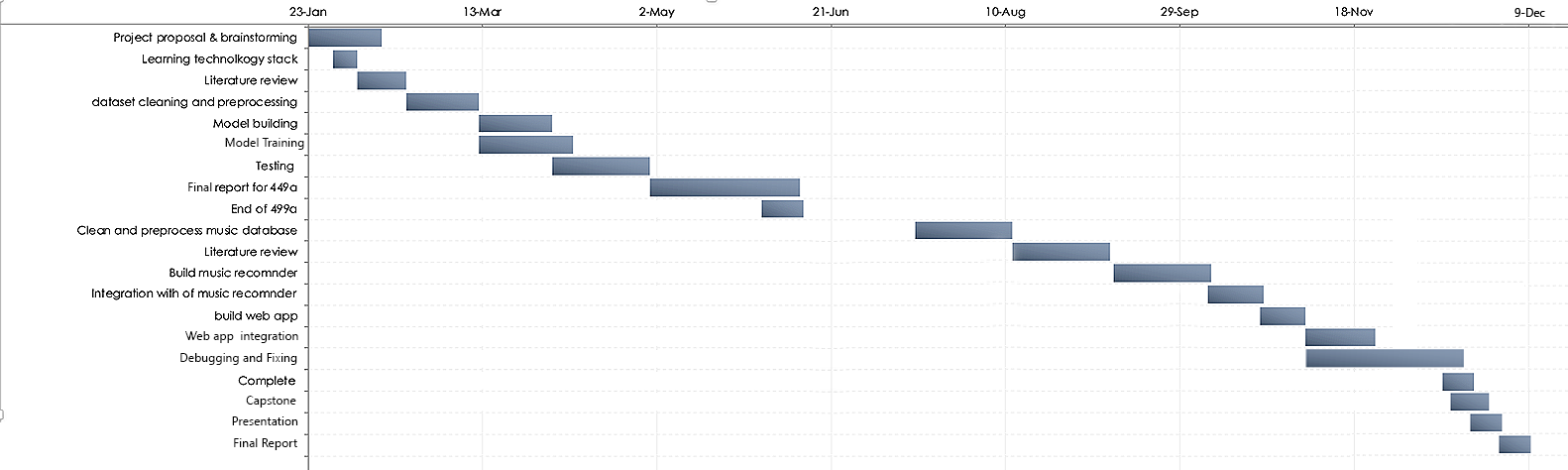


Figure 10: Gantt chart showing the project timeline.

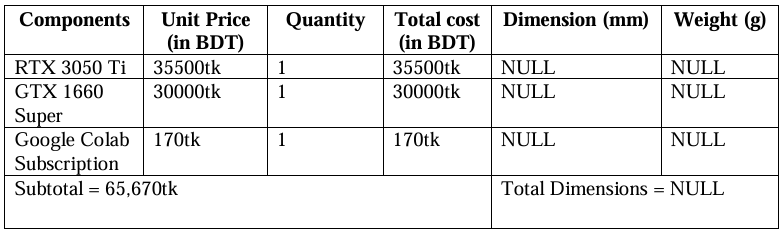


Table 3: Budget table.

# Chapter 7 Complex Engineering Problems and Activities

## 7.1 Complex Engineering Problems (CEP)

The development of the FeelBeat system involves addressing a range of complex engineering problems. These problems require multidisciplinary knowledge, careful analysis, and consideration of various conflicting requirements. Table II below summarizes the attributes related to complex engineering problems in this project.

|  |  |  |
| --- | --- | --- |
| **Attributes** | | **Addressing the complex engineering problems (P) in the project** |
| P1 | Depth of knowledge required (K3-K8) | The project requires knowledge of Machine Learning (K3), Deep Learning Techniques (K4), Audio Signal Processing and Music Theory (K5), Software Development Tools such as TensorFlow, Keras, and Django (K6), and research papers for advanced emotion recognition models (WK8). |
| P2 | Range of conflicting requirements | Balancing the accuracy of emotion recognition with computational efficiency to ensure real-time processing. Another conflict arises between broad emotional categorizations for simplicity and nuanced recognition for better personalization. |
| P3 | Depth of analysis required | A thorough examination was conducted to choose the best model (RMN) from options like ResNet50 and VGG19. Likewise, in the music recommendation system, attributes such as energy, valence, and acousticness were examined to guarantee accurate emotion-to-song correlation. |
| P4 | Familiarity of issues | Necessary knowledge of datasets including FER2013, Spotify API, and machine learning frameworks such as TensorFlow and Keras. Moreover, understanding web development tools, especially Django, was essential for system integration. |
| P5 | Extent of applicable codes | The project primarily relies on existing coding standards for Python, TensorFlow, and Keras. There are no predefined industry codes for emotion recognition or music recommendation systems, requiring custom development and implementation. |
| P6 | Extent of stakeholder involvement | Stakeholders include potential users of the system who provide feedback on usability and playlist relevance. Developers must also consider industry standards for music recommendation platforms like Spotify. |
| P7 | Interdependence | The project involves interdependent subsystems, including the facial emotion recognition module, the music recommendation engine, and the web interface. These components must work seamlessly to provide real-time recommendations. |

Table 4: A SAMPLE COMPLEX ENGINEERING PROBLEM ATTRIBUTES TABLE

## 7.2 Complex Engineering Activities (CEA)

The FeelBeat project required the coordination of various complex engineering activities to ensure successful implementation. Table III below highlights the attributes related to these activities.

Table 5: A SAMPLE COMPLEX ENGINEERING PROBLEM ACTIVITIES TABLE

|  |  |  |
| --- | --- | --- |
| **Attributes** | | **Addressing the complex engineering activities (A) in the project** |
| A1 | Range of resources | The project required financial resources, personnel resources (team members with expertise in web development, backend development, and machine learning), cloud-based tools such as Spotify's API for music feature extraction, and Google Colab for model training. |
| A2 | Level of interactions | The project required active collaboration between team members for model development, testing, and frontend-backend integration. Additionally, interaction with users was needed to gather feedback on the system’s usability and playlist accuracy. |
| A3 | Innovation | |  | | --- | |  |  |  | | --- | | The system employs innovative methods by integrating a Residual Masking Network for emotion detection with a content-based music recommendation engine, providing a unique, personalized music experience based on real-time emotion recognition. | |
| A4 | Consequences to society  / Environment | The project enhances emotional well-being by offering mood-aligned music recommendations, which can positively impact mental health. It promotes the innovative use of AI in everyday applications, creating opportunities for personalized digital experiences. |
| A5 | Familiarity | Knowledge of machine learning tools (TensorFlow, Keras), web frameworks (Django), and APIs (Spotify) was crucial. The team additionally examined UN SDG #3: Good Health and Well-Being, since the system indirectly aids mental health via music therapy. |

# Chapter 8 Conclusions

## 8.1 Summary

FeelBeat is a web application that combines tailored music recommendations with facial expression recognition. The project successfully developed a system that analyzes facial expressions to identify emotions like happiness, sadness, anger, fear, and others, utilizing a Residual Masking Network (RMN) trained on the FER2013 dataset. The identified emotions were linked to selected playlists utilizing Spotify’s audio characteristics, such as energy, valence, and tempo, in accordance with Thayer’s Energy-Stress Model. The system exhibited real-time capabilities through an intuitive web interface, connecting emotion recognition with personalized dynamic music.

## 8.2 Limitations

While FeelBeat achieved its primary objectives, several limitations were identified:

**Emotion Detection Accuracy:** While the RMN reached an accuracy of 74%, there were some moderate misclassifications for subtle emotions such as fear and disgust, which restricted the system’s trustworthiness in those situations.

**Dataset Diversity:** Although the FER2013 dataset is extensive, it fell short in terms of diversity necessary for strong generalization across various demographics and environmental factors (such as different lighting or angles).

**Static Thresholds for Music Features:** The music recommendation system depended on fixed thresholds for classifying audio features, limiting its adaptability to complex or mixed emotional conditions.

**Scalability:** The system is restricted to a web application and has not been fine-tuned for use on mobile devices or other specialized platforms.

**Subjectivity in Emotional Interpretation:** Emotional responses to music are largely subjective, and so the generalist approach adopted by the system may not work in perfect alignment with personal desires.

## 8.3 Future Improvement

To address the current limitations and expand its potential, the following improvements are proposed:

**Enhanced Dataset Diversity:** Including other datasets such as AffectNet or CK+ could provide more balanced and varied training examples for emotion detection, enabling the model to generalize better to real-world situations.

**Hybrid Emotion Models:** Development of models that could detect hybrid or compound emotions such as bittersweet or nostalgia would be better in achieving emotional subtlety and applicability in a better manner.

**Dynamic Music Mapping:** Using machine learning models in adaptive audio feature classification rather than using static thresholds would enable better personalization and precision of recommendations.

**Mobile App Development:** Extending FeelBeat to the Android platform would enhance its accessibility and engagement with users as it would enable the uninterrupted use of the app across devices.

**User Feedback Integration:** Adjusting playlists based on the user preferences with integrated user feedback loops could amplify the overall experience and satisfaction.

**Real-Time Emotion Transitions:** Expanding the system to be able to detect and react to emotional transitions in real-time, like a live video feed, could take it to a new height of functionality.

# References

[1] T. Lee, “Facial expression recognition with CNN-LSTM,” in *Proc. Int. Conf. Comput. Vision Pattern Recognit. (ICCVPR)*, Tokyo, Japan, 2021, pp. 311–316.

[2] S. Ahmed *et al.*, “Facial expression recognition using residual masking networks,” in *Proc. Int. Conf. Pattern Recognit.*, Montreal, Canada, 2020, pp. 445–452.

[3] M. Johnson, “Modified VGG-19 architecture for emotion recognition,” *J. Adv. Comput. Sci. Technol.*, vol. 8, no. 5, pp. 142–149, May 2020.

[4] S. Robinson, “Facial expression recognition with ensemble learning,” *Proc. Conf. Mach. Learn. Appl.*, vol. 2, no. 3, pp. 55–67, 2019.

[5] J. J. Gómez-Valverde *et al.*, “Facial expression recognition with deep learning: Improving on the state of the art and applying to the real world,” *Biomedical Optics Express*, vol. 10, no. 2, pp. 892–913, 2019.

[6] R. Nuzzolo, “Music mood classification using digital signal processing and Thayer’s energy-stress model,” *IEEE Trans. Audio Speech Lang. Process.*, vol. 15, no. 1, pp. 1–10, Jan. 2015.

[7] A. Bhat *et al.*, “Mood classification of songs using digital signal processing techniques,” *BNMIT Technol. Rev.*, vol. 6, no. 4, pp. 358–365, 2015.