“Emotion Recognition using CNN”

A Project Report Submitted to

Amrita School of Engineering

in partial fulfilment of Requirements for the Degree of

Bachelor of Technology

in Computer Science and Engineering (Artificial Intelligence):

22AIE457

Full Stack Development

Submitted by Team-10

BODE VAMSI KRUSHNA – (CH.EN.U4AIE20008)

SAKILAM ABHIMAN - (CH.EN.U4AIE20055)

SANGEETH AJITH – (CH.EN.U4AIE20056)

SARTHAK YADAV - (CH.EN.U4AIE20058)

SHAIK HUZAIFA FAZIL - (CH.EN.U4AIE20060)

Under the Guidance of

Dr. K Venkatraman

Amrita School of Computing

Amrita Vishwa Vidyapeetham University

Chennai-601 103, Tamil Nadu, India.

May 2023.

TABLE OF CONTENTS

List of Figures

List of Tables

ABSTRACT

Facial emotion recognition is a widely studied and difficult task in computer vision. Many researchers have proposed different techniques, either standalone or ensemble-based, to improve the accuracy of emotion classification. However, this research focuses on enhancing accuracy and processing efficiency by using a single standalone neural network to correctly classify human emotions based on facial expressions. In this project, we used transfer learning using EfficientNetB3 as our base model. The project makes use of the FER2013 dataset which consists of 28709 training and 7178 testing images. We use oversampling techniques on the training data as it is very unbalanced due to large differences between different classes of emotions. This unbalance results in poor performance of our model. After everything, our proposed model yields an accuracy of 0.9330 and a loss of 0.3046.

Keywords*: Facial emotion recognition, FER2013, EfficientNetB3, Oversampling*

INTRODUCTION

An emotion-based music recommender system is an artificial intelligence (AI) application that analyzes a user's emotions and recommends music based on those emotions. It uses machine learning algorithms to recognize patterns in user data, such as listening history and social media activity, to determine the user's current emotional state. Once the user's emotional state is identified, the system suggests songs or playlists that are likely to resonate with the user's current mood.

Here are some key components of an emotion-based music recommender system:

1. Data Collection: The system collects data from various sources, including the user's listening history, social media activity, and sensor data (such as heart rate or facial expressions).
2. Emotion Detection: The system uses machine learning algorithms to analyze the data and detect the user's emotional state. Some systems use facial recognition technology to analyze the user's facial expressions, while others use natural language processing (NLP) to analyze the user's social media activity.
3. Music Recommendation: Once the system has identified the user's emotional state, it recommends music that is likely to match that emotional state. This can involve analyzing the user's listening history and suggesting similar songs or artists, or it can involve recommending songs or playlists that are known to be associated with the user's emotional state.
4. Feedback Loop: As the user interacts with the system, the system learns more about the user's preferences and emotional states. This feedback loop allows the system to continually refine its recommendations and provide more personalized and accurate suggestions over time.

There are several benefits of using an emotion-based music recommender system. Firstly, it can help users discover new music that they may not have otherwise found. Secondly, it can enhance the user's listening experience by providing music that matches their current emotional state. Finally, it can improve user engagement and satisfaction with music streaming platforms by providing a more personalized and intuitive experience.

However, there are also some challenges associated with emotion-based music recommender systems. One of the main challenges is accurately detecting the user's emotional state, which can be difficult given the complexity and subjectivity of human emotions. Additionally, there is a risk of perpetuating stereotypes or limiting the user's music choices by associating certain emotions with specific genres or artists.

PROBLEM STATEMENT

Despite the availability of music streaming services, users still struggle to find music that matches their current emotional state, leading to a less satisfying listening experience. Therefore, there is a need for an effective emotion-based music recommender system that can accurately detect and recommend music based on the user's emotional state.

PROJECT SCOPE

The project scope for detecting facial emotions can vary depending on the specific goals and objectives of the project. Here are some possible components that could be included in the project scope:

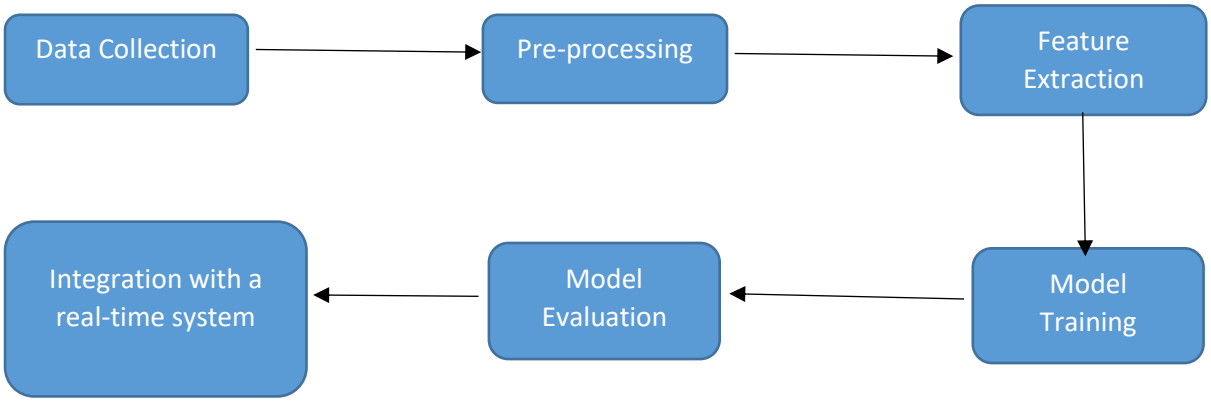

Fig1: Overall architecture of proposed model

RELATED WORKS

The paper by Isha Talegaonkar et al. presents the importance of Human-Computer Interaction (HCI) in a digital world and highlights the role of facial expressions in non-verbal communication. The proposed system aims to develop a Facial Expression Recognition (FER) system using Convolutional Neural Networks (CNN) to classify facial expressions in real-time. The system can be utilized for emotion analysis during activities like watching movie trailers or video lectures. The conclusion states that a CNN model was created and experimented with, achieving satisfactory train and test accuracies, which are, 0.7989 and 0.6012 respectively. The model is capable of real-time emotion classification using a webcam. Overall, the paper introduces a relevant and promising approach for FER using CNN, highlighting its potential applications and presenting positive results.[1]

Facial expressions are a fundamental means of human communication, and deep learning methods in artificial intelligence have been employed to enhance human-computer interactions. However, accurately recognizing and understanding facial expressions can be challenging. This research by Lutfiah Zahara et al. proposes a real-time facial emotion prediction and classification system using the Convolutional Neural Network (CNN) algorithm and the OpenCV library, specifically TensorFlow and Keras, implemented on a Raspberry Pi. The system involves three main processes: face detection, facial feature extraction, and facial emotion classification. The experimental results using the FER-2013 dataset and CNN method achieved a facial expression prediction accuracy of 65.97%.[2]

While many researchers focus on improving accuracy, the study by Gede Putra Kusuma, Jonathan, and Andreas Pangestu Lim aims to enhance the processing efficiency of emotion classification by utilizing a single standalone neural network. The proposed approach involves a modified Convolutional Neural Network (CNN) based on the VGG-16 classification model, which was pre-trained on the ImageNet dataset and fine-tuned for emotion classification. The classification process is carried out on the FER-2013 dataset, consisting of over 35,000 face images with in-the-wild settings, representing 7 distinct emotions, and divided into 80% training, 10% validation, and 10% testing data. The proposed approach achieves an accuracy of 69.40%, outperforming many standalone-based models.[3]

In this study, Yousif Khaireddin and Zhuofa Chen try to enhance the accuracy of FER2013 using CNN. They have achieved the highest classification accuracy using a single network on the FER2013 dataset. They have employed the VGGNet architecture, carefully fine-tuned its hyperparameters, and experimented with various optimization methods. Notably, their model achieves a state-of-the-art single-network accuracy of 73.28% on the FER2013 dataset without the need for additional training data.[4]

Benyoussef Abdellaoui et al. proposed a custom CNN model. They applied the Keras Tuner model optimizer for the FER2013 dataset. Keras Tuner is an open-source library within the TensorFlow ecosystem that provides an easy-to-use API for automating the process of hyperparameter tuning in Keras deep learning models, enabling users to find the optimal configuration for their models. They got a training accuracy of 0.8313 and a validation accuracy of 0.53.[5]

Despite the increasing interest in real-time facial emotion recognition for human-computer interaction, existing datasets in this field suffer from various issues such as unrelated photos, imbalanced class distributions, and misleading images that can adversely impact accurate classification. To address these problems, Abou Zafra et al. make use of new dataset called 3RL in this project, consisting of approximately 24,000 labeled images representing five basic emotions: happiness, fear, sadness, disgust, and anger. In comparison to other well-known datasets like FER and CK+, experiments conducted using commonly used algorithms like SVM and CNN demonstrate significant improvements in generalization on the 3RL dataset, achieving an accuracy of up to 91.4%, while results on FER2013 and CK+ datasets range from approximately 60% to 85%.[6]

The project by Mengyu Rao, Ruyi Bao, and Liangshun Dong is a comparative study between CK+ and FER2013 datasets. They used four different models for the comparison. These models are VGG19, ResNet18, ResNet50, and Xception. They also apply Hybrid Data Augmentation by applying horizontal flips and adding Gaussian noise. The final results show that the CK+ dataset gives better accuracies in all four models than FER2013. [7]

The research by Ozioma Collins Oguine et al. proposes a Hybrid Architecture that combines the Haar Cascade Face Detection algorithm with a CNN Model. The CNN architecture processes input images of size 48x48x1 from the FER 2013 dataset. With the suggested modifications outlined in the paper, the proposed model achieved an average predictive accuracy of 70%. The weighted average accuracy on the test dataset was also 70%.[8]

DATASET

FER-2013:

The dataset comprises a collection of images depicting faces, with each image consisting of grayscale pixels arranged in a 48x48 grid. These images have undergone an automated registration process, ensuring that the face within each image is approximately centered and occupies a consistent amount of space.

The main objective of this task is to assign an emotion category to each face based on the displayed facial expression. There are seven distinct emotion categories used for classification: Angry, Disgust, Fear, Happy, Sad, Surprise, and Neutral. The purpose is to develop a model that can accurately predict the emotion category for any given face image.

To facilitate model development and evaluation, the dataset has been divided into two sets: a training set and a public test set. The training set comprises 28,709 examples, which will be used to train and fine-tune the emotion classification model. The public test set consists of 3,589 examples, which will be used to assess the performance of the trained model and determine its ability to generalize to unseen data.

By analyzing and learning from this dataset, the goal is to create a model capable of accurately categorizing facial expressions into one of the seven specified emotions, thereby enabling the recognition and understanding of emotions from facial images.

REQUIREMENTS

Functional Requirements: Functional requirements are a type of software requirement that specify the functions or capabilities that a software system must possess to meet the needs of its users. These requirements define what the system should do, how it should behave, and what its inputs and outputs are.

- The system should be able to detect facial emotions accurately.
- The system should be able to recognize different facial emotions such as happy, sad, angry, etc.
- The system should be able to provide real-time feedback on detected facial emotions.
- The system should be able to work under different lighting conditions.
- The system should be able to work with different skin tones.

Non-functional Requirements: Non-functional requirements are a type of software requirement that describe the qualities or attributes that a software system should have, rather than its specific functions or capabilities. These requirements define how the system should perform and behave, and they are often related to system performance, security, usability, maintainability, and reliability.

- The system should be reliable and accurate in detecting facial emotions.
- The system should have a fast response time to provide real-time feedback.
- The system should have a high level of security to protect the privacy of the user.
- The system should be easy to use and intuitive for the user.
- The system should be compatible with different operating systems and hardware.
- The system should have good performance even under heavy usage.
- The system should be scalable and able to handle a large number of users.
- The system should have good accessibility for users with disabilities.

METHODOLOGY

1) Data Analysis and Visualization: Data analysis involves exploring and examining datasets to understand their structure, patterns, and relationships. It often includes tasks such as data cleaning, data manipulation, and statistical analysis. Python provides powerful libraries such as Pandas, NumPy, and SciPy that facilitate these tasks. Pandas, in particular, are widely used for data manipulation, transformation, and analysis, offering flexible data structures and data analysis tools.

   Visualization, on the other hand, involves representing data visually using graphs, charts, and plots. Python provides several libraries for data visualization, the most popular being Matplotlib and Seaborn. Matplotlib offers a wide range of customizable plots, while Seaborn provides higher-level functions for creating aesthetically pleasing statistical visualizations. Additionally, libraries like Plotly and Bokeh enable interactive and web-based visualizations.

   We have plotted Bar charts and Pie charts for visualizing and better understanding of the dataset we are going to work on.

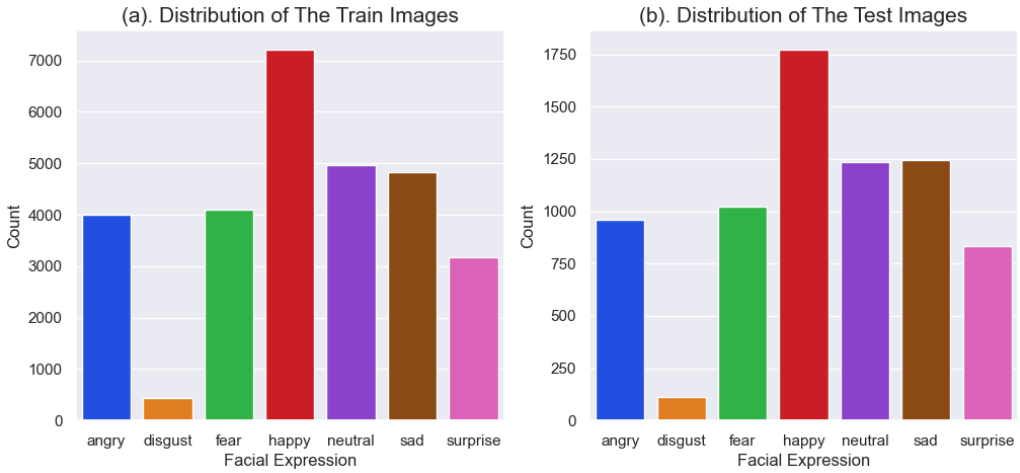

Fig2: Bar charts for Data Visualization

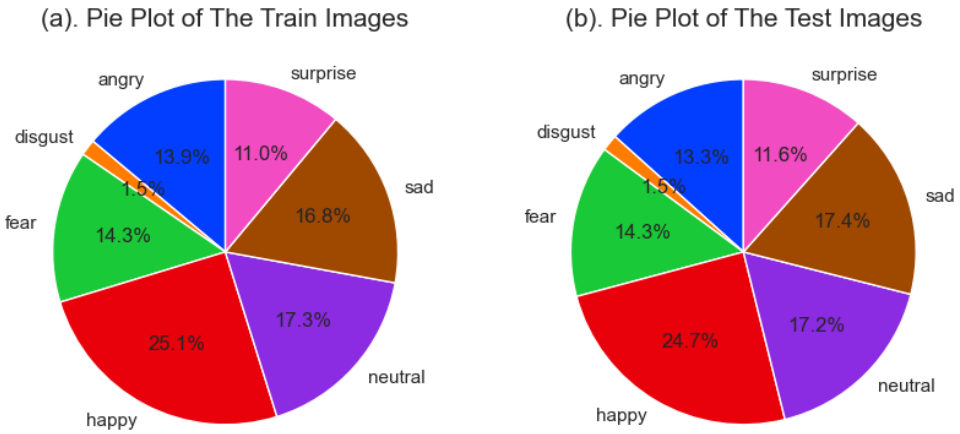

Fig3: Pie charts for Data Visualization

2) Data Pre-processing: Data pre-processing is a crucial step in the data analysis pipeline. It involves preparing and transforming raw data into a format that is suitable for analysis and modeling. Data pre-processing helps in improving the quality and reliability of the data, removing inconsistencies, handling missing values, and ensuring that the data is in a consistent and usable form. Oversampling the minority class helps to provide more training examples for the model to learn from and can improve the model's ability to accurately classify the minority class. However, it is essential to be cautious with oversampling, as blindly applying it can lead to overfitting or artificially inflating the importance of the minority class.
Careful evaluation and consideration of different oversampling techniques are necessary to ensure that the model benefits from the increased representation of the minority class without introducing biases or degrading the performance of the majority class.
This project oversamples the minority classes. The goal is to increase the representation of the minority class in the training dataset by creating additional synthetic samples.
Overall, our code performs oversampling by duplicating or creating synthetic samples to increase the representation of the minority class in the training dataset.

The data after oversampling:

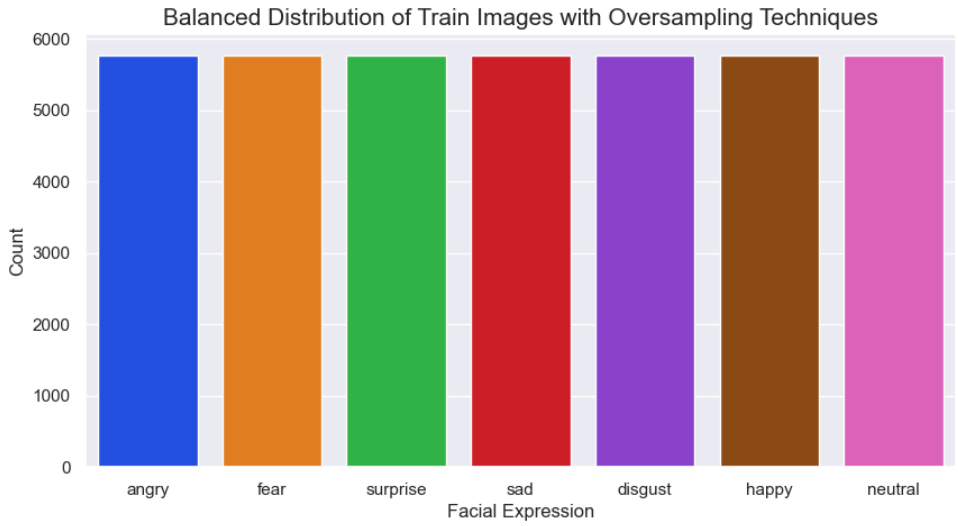

Fig4: Bar chart of Train Data after Oversampling

3) Model Development: We have applied transfer learning in this project, that is, we have taken a pre-trained model as the base model and upgraded the same. This project makes use of EfficientNetB3. EfficientNetB3 is a specific variant of the EfficientNet architecture. It refers to the third model in the EfficientNet series, which is known for its balance between model size and performance.

EfficientNetB3 is designed by scaling the base EfficientNet architecture using the compound scaling method. It has more parameters and is deeper compared to EfficientNetB0 and EfficientNetB1, but it is still more efficient than larger models like EfficientNetB4 or EfficientNetB7.

The scaling of EfficientNetB3 involves increasing the depth, width, and resolution of the network. The depth is increased by adding more layers, while the width is increased by widening the channels in each layer. The resolution is increased by using larger input image sizes.

EfficientNetB3 has been pre-trained on a large-scale dataset such as ImageNet, allowing it to learn general features from a diverse range of images. It can be fine-tuned or used as a feature extractor for various computer vision tasks, including image classification, object detection, and image segmentation.

Compared to earlier versions of EfficientNet, EfficientNetB3 typically offers improved performance on image classification benchmarks while maintaining a relatively efficient model size and computational cost.

The model description is as below:

- We provide EfficientNetB3 as our base model.
- After the base model, we have added the BatchNormalizatoin layer which normalizes the activations of the previous layer. It helps in stabilizing and accelerating the training process.
- After this, we add a Dense layer.
- Now we apply Dropout. This layer randomly sets a fraction of the input units to 0 at each update during training to prevent overfitting.
- Finally, we add one more Dense layer which will be the model's Output layer.

4) Model Evaluation: We measure the Accuracy and the Loss of our model.
Accuracy is a metric that measures the proportion of correctly classified samples out of the total number of samples. It provides an overall assessment of how well the model is performing in terms of correct predictions. Accuracy is often used in classification tasks where the goal is to assign the correct label or class to each input.
Loss, also referred to as the loss function or objective function, quantifies the difference between the predicted outputs of the model and the actual ground truth labels. It represents the error or discrepancy between the predicted values and the true values.

5) Saving the model: We save this model as an HDF5 (.h5) file. These files are often used to save and load trained models. When training a deep learning model, the weights, architecture, and other necessary parameters of the model can be saved in a .h5 file format. This allows you to save the model's state and use it later for prediction, fine-tuning, or sharing with others.

6) WebApp: The application we described consists of multiple modules that work together to create a web-based system for real-time face detection, track recommendation, and streaming video. Here's a combined explanation of each module's purpose:

   i. Spotipy:
      - Spotipy is a module used to establish a connection to Spotify and retrieve tracks using the Spotipy wrapper.
      - It provides functionality to authenticate with Spotify, access user playlists, search for tracks, and retrieve track information.
   ii. haarcascade:
      - The haarcascade module is used for face detection.
      - It includes pre-trained classifiers (haarcascade XML files) that can detect faces in images or video frames.
      - These classifiers utilize Haar-like features and machine learning techniques to identify facial features.
   iii. camera.py:
      - The camera.py module is responsible for video streaming, frame capturing, prediction, and track recommendation.
      - It utilizes the webcam or camera input to capture video frames.
      - It performs real-time face detection using the haarcascade module.
      - For each detected face, it makes predictions and recommends tracks based on the detected emotion or facial expression.
   iv. main.py:
      - main.py is the main Flask application file.
      - It defines routes and handles HTTP requests from the web page.
      - It interacts with the camera.py module to initiate video streaming, capture frames, and receive predictions and track recommendations.
      - It sends the processed data to the web page for display and interaction.
   v. index.html:
      - index.html is an HTML file located in the 'templates' directory.

- It serves as the web page for the application.
- It provides the user interface and displays the video stream, detected faces, predicted emotions, and recommended tracks.
- It includes basic HTML and CSS code for structuring and styling the web page.

vi. utils.py:

- utils.py is a utility module used for video streaming from the web camera.
- It employs threads to enable real-time video capture and processing.
- It assists in achieving smooth and responsive streaming by utilizing concurrent execution.

Together, these modules work in tandem to create a web-based application that streams video from a webcam, detects faces in real-time, predicts emotions or facial expressions, and recommends tracks based on the detected emotion. Users can interact with the application through the web page and view the video stream along with the corresponding predictions and recommendations.

RESULTS

In this project, we used transfer learning by taking EfficientNetB3 as the base model. After training the model we calculate the Accuracy and Loss of our model.

Table2: Accuracy and Loss values proposed model

Accuracy	**0.9330**
Loss	**0.3046**

We have plotted the Accuracy and Loss graphs for our model:

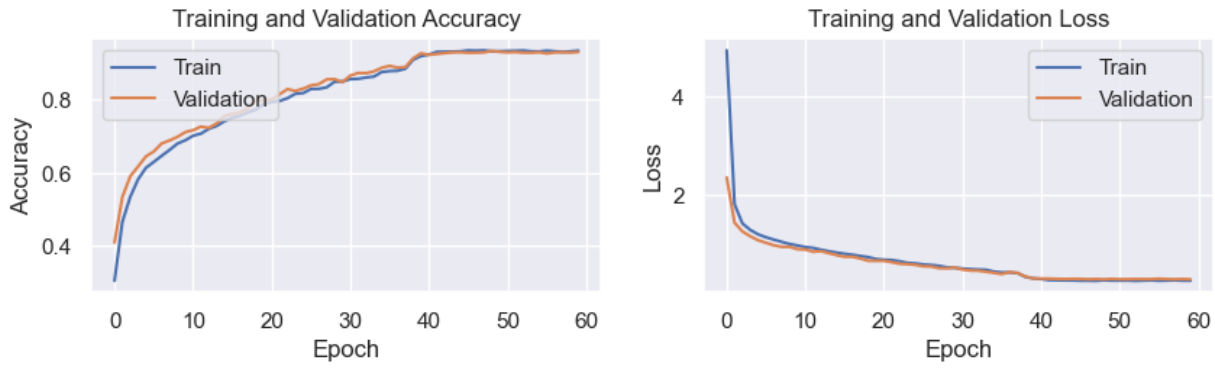

Fig5: Accuracy and Loss Plots

We have also plotted Confusion matrix:

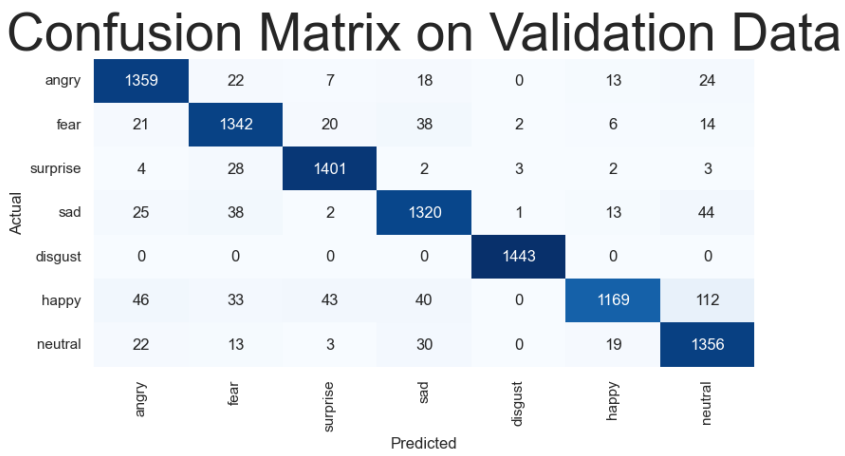

Fig6: Confusion Matrix

CONCLUSION

In conclusion, the Emotion-Based Music Recommender incorporating real-time face recognition, a web app, and Spotipy integration offers an innovative and personalized music recommendation experience. By analyzing facial expressions in real-time, the system can detect and interpret users' emotions, providing them with music that aligns with their current mood. The web app interface serves as a user-friendly platform where individuals can access the recommender system and interact with its features.

The integration of Spotipy, a popular music streaming and recommendation service, enhances the functionality of the Emotion-Based Music Recommender. Spotipy's extensive music library and recommendation algorithms enable the system to provide accurate and diverse song suggestions based on the detected emotions. Users can enjoy a seamless experience of discovering music that resonates with their feelings at any given moment.

This technology has the potential to revolutionize the way we engage with music, bridging the gap between our emotions and the songs we listen to. It adds a layer of emotional intelligence to music

recommendations, allowing users to explore a wide range of genres, artists, and tracks that align with their emotional state. By combining real-time face recognition, web app accessibility, and Spotipy integration, the Emotion-Based Music Recommender offers a unique and personalized music discovery experience tailored to each individual's emotions.

REFERENCES

[1] Talegaonkar, Isha and Joshi, Kalyani and Valunj, Shreya and Kohok, Rucha and Kulkarni, Anagha, Real Time Facial Expression Recognition using Deep Learning (May 18, 2019). Proceedings of International Conference on Communication and Information Processing (ICCIP) 2019, http://dx.doi.org/10.2139/ssrn.3421486

[2] L. Zahara, P. Musa, E. Prasetyo Wibowo, I. Karim and S. Bahri Musa, "The Facial Emotion Recognition (FER-2013) Dataset for Prediction System of Micro-Expressions Face Using the Convolutional Neural Network (CNN) Algorithm based Raspberry Pi," *2020 Fifth International Conference on Informatics and Computing (ICIC)*, Gorontalo, Indonesia, 2020, pp. 1-9, doi: 10.1109/ICIC50835.2020.9288560.

[3] Kusuma Negara, I Gede Putra & Jonathan, Jonathan & Lim, Andreas. (2020). Emotion Recognition on FER-2013 Face Images Using Fine-Tuned VGG-16. Advances in Science, Technology and Engineering Systems Journal. 5. 315-322. 10.25046/aj050638.

[4] Yousif Khaireddin and Zhoufa Chen. (2021). Facial Emotion Recognition: State of the Art Performance on FER2013. https://doi.org/10.48550/arXiv.2105.03588

[5] Abdellaoui, Benyoussef & Moumen, Aniss & Idrissi, Younes & Remaida, Ahmed. (2021). Training the Fer2013 Dataset with Keras Tuner. 409-412. 10.5220/0010735600003101.

[6] Rahmeh Abou Zafra, Lana Ahmad Abdullah, Rouaa Alaraj, Rasha Albezreh, Tarek Barhoum, Khloud Al Jallad. (2022). An experimental study in Real-time Facial Emotion Recognition on new 3RL dataset. https://doi.org/10.33140/JCTCSR

[7] by Mengyu Rao, Ruyi Bao, and Liangshun Dong. (2022). Face Emotion Recognization Using Dataset Augmentation Based on Neural Network. https://doi.org/10.1145/3561518.3561519

[8] Ozioma Collins Oguine, Kanyifeechukwu Jane Oguine, Hashim Ibrahim Bisallah, Daniel Ofuani. (2022). Hybrid Facial Expression Recognition (FER2013) Model for Real-Time Emotion Classification and Prediction. https://doi.org/10.48550/arXiv.2206.09509