KokoroBot: An Empathetic Chatbot

Contents

1 Introduction **2**

2 Workflow of the project **3**

3 Materials & Methods **3**
- 3.1 Dataset . . . 3
- 3.2 Data Preprocessing . . . 3
  - 3.2.1 Converting the Dataset . . . 3
  - 3.2.2 Loading the Dataset . . . 4
  - 3.2.3 Preprocessing Images . . . 4
- 3.3 Model Building . . . 4
  - 3.3.1 Splitting the Data . . . 4
  - 3.3.2 Defining the CNN Model . . . 4
  - 3.3.3 Compiling the Model . . . 4
  - 3.3.4 Training the Model . . . 4
  - 3.3.5 Evaluating the Model . . . 4
- 3.4 Integrating with LLM . . . 4
  - 3.4.1 Loading the API Key . . . 4
  - 3.4.2 Defining Functions . . . 5
  - 3.4.3 Chatbot Loop . . . 5
- 3.5 Integrating with Streamlit . . . 5

4 Results **5**
- 4.1 Previous attempts: . . . 5
- 4.2 Pre-trained Models . . . 6
  - 4.2.1 VGG16 . . . 6
  - 4.2.2 MobileNetV2 . . . 6
  - 4.2.3 Custom Convolutional Neural Networks (CNNs) . . . 6
  - 4.2.4 Model C1 . . . 6
  - 4.2.5 Model C2 . . . 6
  - 4.2.6 Model C3 . . . 7
  - 4.2.7 Challenges in Emotion Classification . . . 7
- 4.3 Improved Approach . . . 8
  - 4.3.1 Comparison and Outcome . . . 8

5 Conclusion **9**

1 Introduction

Emotions are a fundamental aspect of human experience, influencing our decisions, interactions, and overall well-being. They provide valuable insights into our mental state, guiding our responses and behaviors in various situations. Central to human communication, emotions help us navigate complex social interactions and understand others' feelings, influencing our decisions both in everyday life and in critical situations. Understanding emotions allows us to empathize with others, anticipate their needs, and respond appropriately, thereby enhancing the quality of our interactions and relationships. Advancements in AI and machine learning have led to the development of sophisticated models capable of recognizing emotions from facial expressions with remarkable accuracy. This technology not only aids in understanding human sentiment but also offers numerous practical applications that can significantly improve lives.

In response to the growing need for empathetic interactions, we have developed KokoroBot, an advanced AI-driven application designed to engage users in personalized conversations based on their emotional states. The name "**Kokoro**" is derived from the Japanese word meaning "**heart**" or "**mind**," reflecting the bot's aim to connect with users on a deeper emotional level.

KokoroBot: An Empathetic Chatbot

KokoroBot is an advanced AI-driven application designed to engage users in personalized conversations based on their emotional states. Built using Streamlit for the frontend, it seamlessly integrates Google's Gemini-Pro AI model for response generation and TensorFlow for real-time emotion detection from webcam frames. The chatbot's interface is meticulously styled using Streamlit with custom CSS, ensuring a visually appealing and user-friendly experience that distinguishes between user, bot, and system messages. The chat history is managed within the session state to maintain context across interactions. On the backend, the chatbot leverages OpenCV for capturing and processing webcam frames, employing image processing techniques to extract meaningful features. A TensorFlow model, built on a convolutional neural network (CNN), is used for real-time emotion detection. This model analyzes the preprocessed webcam frame to classify the user's emotion into categories such as happiness, sadness, anger, or fear. This CNN-based approach ensures accurate and fast emotion recognition, enhancing the chatbot's ability to respond appropriately to the user's emotional state. The chatbot's interaction flow allows users to input messages through a text input field. Upon submission, the chatbot captures a frame from the webcam, processes it using OpenCV, and feeds it into the TensorFlow emotion detection model. Based on the detected emotion and the user's input message, the chatbot generates an empathetic response using the Gemini-Pro AI model. This response is crafted to be supportive and positive, celebrating joy, offering comfort for sadness, reassuring for fear, and providing advice to manage anger.

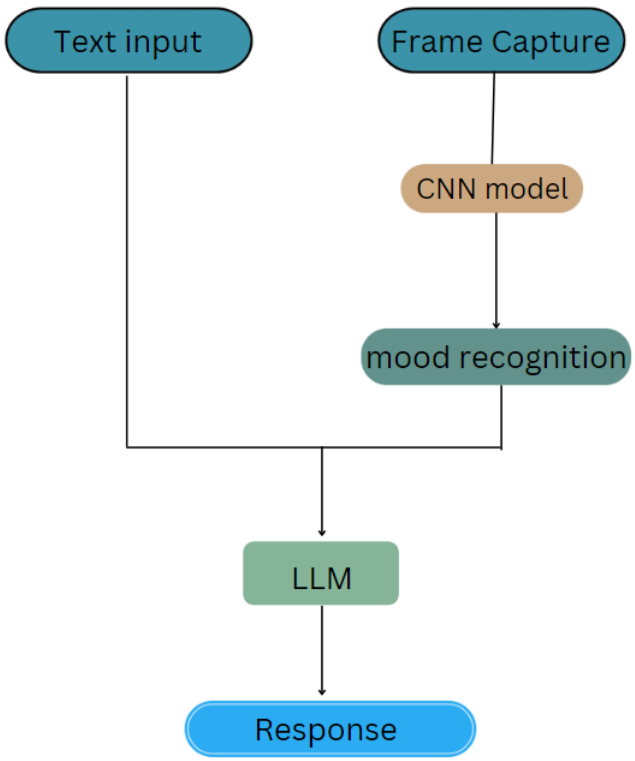
In summary, KokoroBot combines cutting-edge AI technologies, including CNNs for image processing and emotion detection, with a user-friendly Streamlit frontend and Google's powerful Gemini-Pro AI for response generation. This integration ensures a seamless and engaging user experience, where the chatbot not only responds to textual input but also adapts its responses based on the user's detected emotional state, creating a supportive and empathetic interaction environment.

Report Overview

In this report, we present the workflow, materials, methods, results, and discussion related to the development and evaluation of the KokoroBot chatbot. We discuss the dataset used, the architecture of the emotion detection model, training and evaluation processes, and the implications of our

findings. The goal is to demonstrate the effectiveness and potential applications of KokoroBot in providing empathetic and supportive interactions based on real-time emotional analysis.

2 Workflow of the project



3 Materials & Methods

3.1 Dataset

The AffectNet dataset, sourced from Kaggle, is a comprehensive collection of over 1 million facial images annotated with seven emotions: anger, contempt, disgust, fear, happiness, sadness, and surprise, along with neutral expressions. This dataset is invaluable for emotion recognition research due to its diversity, featuring images collected from the internet with variations in lighting, background, facial orientation, ethnicity, age, and gender. For this project, we focused on four primary emotions: anger, sadness, happiness, and fear, chosen for their distinct and recognizable facial features, which are crucial for real-world applications. Each image in AffectNet is meticulously labeled by human annotators, ensuring high-quality annotations. The images are 96x96 pixels, providing a balanced resolution for detailed facial feature extraction without excessive computational load.

3.2 Data Preprocessing

3.2.1 Converting the Dataset

The original dataset containing images of seven different emotions is filtered to include only four emotions: anger, sad, happy, and fear. This reduction is aimed at improving the model's accuracy.

3.2.2 Loading the Dataset

The filtered images are loaded into the program, and the number of images per emotion is counted to ensure a balanced dataset. This step is crucial for ensuring that the model is trained on a well-distributed dataset, preventing bias toward any particular emotion.

3.2.3 Preprocessing Images

Images are standardized by resizing and normalizing them. This ensures that all images fed into the model have consistent dimensions and pixel value distributions, which is critical for effective training.

3.3 Model Building

3.3.1 Splitting the Data

The dataset is divided into training and testing sets, typically with an 80/20 split. This allows the model to be trained on one portion of the data and validated on another to evaluate its performance.

3.3.2 Defining the CNN Model

A Convolutional Neural Network (CNN) is constructed to classify the images into one of the four emotions. The model includes several convolutional layers to extract features from the images, pooling layers to reduce the spatial dimensions, and fully connected layers to perform the final classification.

3.3.3 Compiling the Model

The CNN is compiled with an optimizer (Adam), a loss function (categorical cross-entropy), and accuracy as the evaluation metric. These choices help in optimizing the model's weights during training to minimize the loss and improve accuracy.

3.3.4 Training the Model

The model is trained using the training dataset, with validation on the testing set. Callbacks like EarlyStopping, ReduceLROnPlateau, and ModelCheckpoint are used to monitor training progress, adjust the learning rate when necessary, and save the best model.

3.3.5 Evaluating the Model

The trained model's performance is assessed using various metrics, and its predictions are visualized to ensure it correctly identifies emotions in the test images.

3.4 Integrating with LLM

3.4.1 Loading the API Key

The API key for the Google Gemini-Pro AI model is loaded from environment variables, allowing the program to authenticate and use the model for generating empathetic responses.

3.4.2 Defining Functions

Utility functions are created to handle tasks such as capturing images from a webcam, preprocessing these images, loading the emotion detection model, predicting emotions, and generating responses using the Gemini-Pro model. These functions streamline the process of integrating emotion detection with the chatbot's conversational capabilities.

3.4.3 Chatbot Loop

The chatbot continuously interacts with the user by capturing webcam images to detect their emotions and generating appropriate responses. This loop ensures real-time interaction, making the chatbot capable of responding empathetically based on the user's detected emotional state.

This code effectively combines image processing, neural network training, and language model integration to create a robust system capable of detecting emotions and providing empathetic responses in real-time, showcasing an advanced application of artificial intelligence in human-computer interaction.

3.5 Integrating with Streamlit

`app.py` is a Streamlit application that replicates the functionalities of the `final_code.ipynb` notebook, offering a user-friendly interface. This code integrates an emotion detection model with a language model to create an empathetic chatbot that interacts based on detected emotions and user input. It captures real-time webcam frames to analyze emotions, predicts them using a pre-trained model, and utilizes the Google Gemini-Pro AI model to generate appropriate responses. Session state management in Streamlit, such as st.session_state, is crucial for maintaining conversation history and context. User interactions are handled through a Streamlit form, validating input before processing. Responses are generated dynamically: the webcam captures frames, which are preprocessed and analyzed for emotion prediction, updating session_state.emotion. The Gemini-Pro AI model then generates responses tailored to the user's emotions and input, stored in session_state.chat_history and displayed through Streamlit's chat interface. Custom CSS styling ensures a visually appealing and organized display of chat messages. Overall, this application provides a seamless and engaging user experience, supported by environment variable management, model loading, image preprocessing, and real-time interaction through Streamlit's intuitive interface.

4 Results

4.1 Previous attempts:

Previous attempts involved using seven emotions for mood detection using a pre-trained model and two custom-defined models. The results showed classification reports and plots, indicating the model's performance. However, accuracies and other metrics were observed to be very low. The complexity of distinguishing between eight emotions might have contributed to this, as the models struggled to generalize across a broad range of emotional states.

4.2 Pre-trained Models

4.2.1 VGG16

- **Architecture:**VGG16 is a deep convolutional neural network architecture introduced by the Visual Geometry Group from the University of Oxford. It consists of 16 layers (13 convolutional layers, 3 fully connected layers) and is known for its simplicity and effectiveness in image classification tasks. It uses small 3x3 convolution filters and a uniform architecture with deep layers, which enhances its ability to capture detailed features.
- **Accuracy:** 60.40%

4.2.2 MobileNetV2

- **Architecture:**MobileNetV2 is a lightweight deep neural network architecture developed by Google. It is designed for mobile and embedded vision applications. It improves upon the original MobileNet by introducing inverted residuals and linear bottlenecks, which enhance performance while maintaining computational efficiency. This architecture balances high accuracy with low resource consumption.
- **Accuracy:** 47.59%

4.2.3 Custom Convolutional Neural Networks (CNNs)

4.2.4 Model C1

This custom CNN model is built with a series of convolutional, batch normalization, max pooling, dropout, and dense layers to extract features and perform classification.

Layers Used	**Number**
Convolutional layers	6
Batch normalization layers	2
Max pooling layers	3
Dropout layers	3
Dense layers	2
Flatten layer	1
Accuracy Achieved	60.07%

Table 1: Model C1 Details

4.2.5 Model C2

This custom CNN incorporates a deeper architecture with more layers, including batch normalization and dropout layers, to improve regularization and prevent overfitting.

Layers Used	Number
Convolutional layers	8
Batch normalization layers	6
Max pooling layers	4
Dropout layers	4
Dense layers	2
Flatten layer	1
Accuracy Achieved	69.25%

Table 2: Model C2 Details

4.2.6 Model C3

This custom CNN model is very similar to Model C2, but slight adjustments in the architecture or hyperparameters result in a marginally higher accuracy.

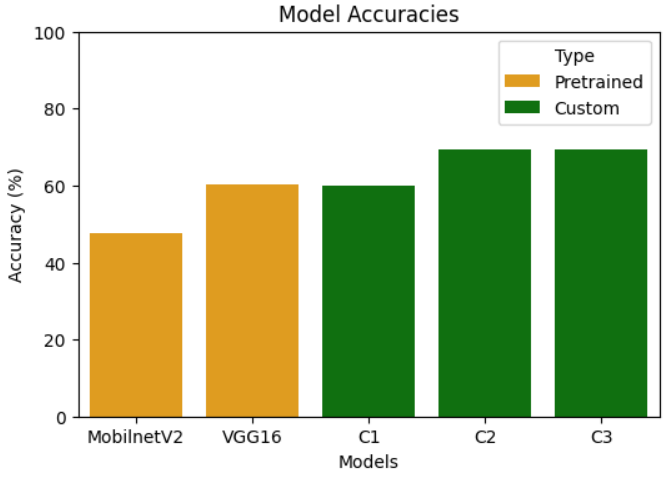
Layers Used	Number
Convolutional layers	8
Batch normalization layers	6
Max pooling layers	4
Dropout layers	4
Dense layers	2
Flatten layer	1
Accuracy Achieved	69.41%

Table 3: Model C3 Details

4.2.7 Challenges in Emotion Classification

Distinguishing between similar emotional states, such as anger and disgust, poses a significant challenge for mood detection models. These emotions often share overlapping facial expressions and features, making it difficult for models to generalize well across such nuanced differences. The complexity of the task is reflected in the varied performance of the models, with custom CNNs (C2 and C3) demonstrating higher accuracy due to their deeper architectures and sophisticated regularization techniques.

These are accuacies of diferent models:



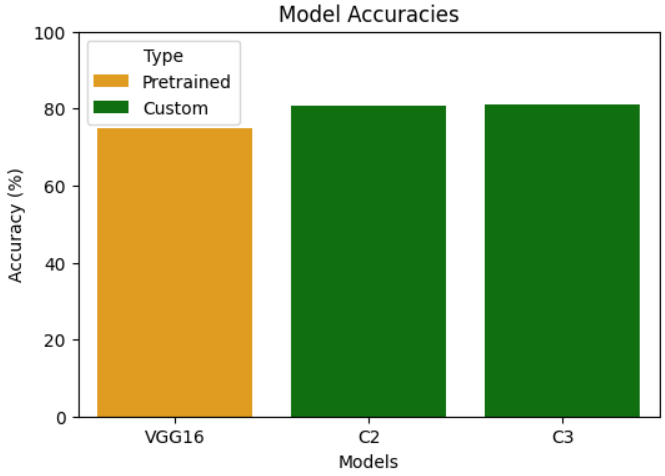
4.3 Improved Approach

To address the low accuracy, we reduced the number of emotions to four: anger, sad, happy, and fear. These emotions were chosen because they can be distinguished easily, allowing the models to be trained more effectively. Anger, sadness, happiness, and fear are fundamental emotions with distinct characteristics that can be recognized through facial expressions and other physiological cues, making them suitable for effective emotion detection.

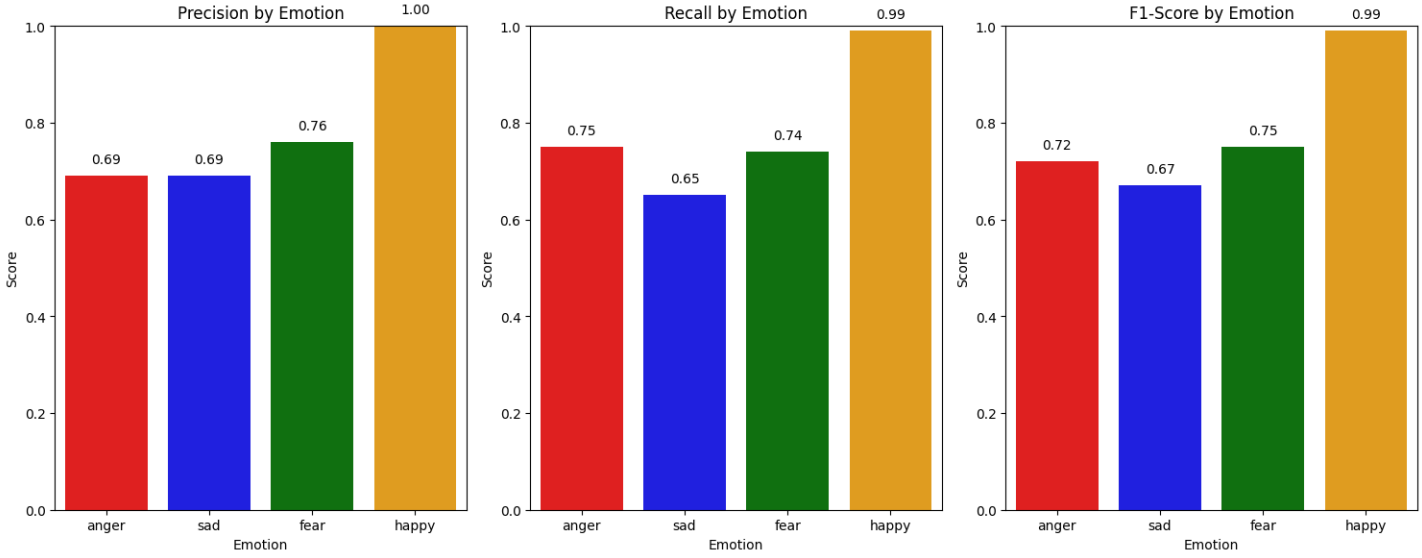
4.3.1 Comparison and Outcome

The simplified model focusing on four emotions significantly improved accuracy and other metrics compared to the previous attempts. Specifically, the reduced emotional range allowed the models to focus more effectively on distinguishing between the four primary emotional states. This approach led to a clearer differentiation between emotions and improved model performance.

These are accuacies of diferent models:



So C3 model is choosen as the final model and metrics of the choosen model are as follows:



5 Conclusion

In conclusion, the development of KokoroBot represents a significant advancement in the field of human-computer interaction, leveraging AI to create empathetic and supportive chatbot interactions based on real-time emotional analysis. Key highlights and findings from the project include:

- **Emotional Recognition and Response:** KokoroBot integrates a Convolutional Neural Network (CNN) for real-time emotion detection from webcam frames. This model effectively identifies user emotions such as happiness, sadness, anger, and fear, enabling the chatbot to respond empathetically and appropriately.
- **Model Selection and Performance:** The project explored several pre-trained and custom CNN models for emotion classification. Ultimately, Model C3, a custom CNN architecture, was selected due to its superior accuracy and performance in distinguishing between the chosen emotional states.
- **Integration of Technologies:** The chatbot was built using Streamlit for the frontend, providing a user-friendly interface for interaction. It integrates Google's Gemini-Pro AI model for generating empathetic responses, enhancing the quality of user-bot interactions.
- **Dataset and Data Preprocessing:** The AffectNet dataset was used, consisting of facial images annotated with emotions. Data preprocessing included filtering and standardizing images, ensuring a balanced dataset and optimal training conditions.
- **Real-time Interaction:** The chatbot captures webcam frames, processes them using OpenCV for feature extraction, and analyzes them using the emotion detection model. It then generates appropriate responses based on both the user's input and detected emotional state.
- **User Experience and Interface:** KokoroBot's interface was designed with a focus on user experience, employing custom CSS for visual appeal and Streamlit's session state management for maintaining conversation context.

Overall, KokoroBot demonstrates the successful integration of AI technologies to provide empathetic and responsive interactions based on real-time emotional analysis. This project not only contributes to the field of AI-driven emotional recognition and chatbot development but also showcases practical applications in enhancing user experience through empathetic human-computer interactions.

References

1. https://deepmind.google/technologies/gemini/
2. https://docs.streamlit.io/
3. https://docs.opencv.org/4.x/dd/d43/tutorial_py_video_display.html
4. https://github.com/YasminSalehi/emotion-aware-chatbot
5. www.kaggle.com
6. https://www.mathworks.com/help/deeplearning/ref/vgg16.html
7. https://keras.io/api/applications/mobilenet/