Emotion Detection Using Deep Learning

Public link to proposal: https://drive.google.com/file/d/1-OgLo8QCRcrziVlRIKjVNrwJPgi97mli/view?usp=sharing

Team members

Sravani Medarametla

sravanimedarametla@my.unt.edu

Chandrika Chandra

chandrikachandra@my.unt.edu

Akhila Sannayila

akhilasannayila@my.unt.edu

Nitish Reddy Gangannagari

nithishreddygangannagari@my.unt.edu

Ram Prasad Jampani

ramprasadjampani@my.unt.edu

Roles

Sravani Medarametla: Documentation

Duties: - Establishing a centralized and organized document repository where all project-relateddocuments, files, and records can be stored. It can be accessed in google drive. And also helps with model development and testing.

Chandrika Chandra: Developer

Duties: Designing, implementing, and maintaining models from scratch using programming languages like python. Gathering dataset for training and testing the model.

Akhila Sannayila: Deployment

Duties: create a deployment plan, Prepare and configure the production and staging environments, Execute the deployment plan, including deploying application code, databasechanges, configuration updates, and other required components.

Nitish Reddy Gangannagari: Developer

Duties: writing the code, Identify and fix bugs, errors, and issues in the codebase through testing,debugging, and problem-solving.

Ram Prasad Jampani: Lead

Duties: Allocation of tasks, Helding the meeting, Managing the timelines, taking updates fromdifferent roles, Helping team members.

Collaborative Work Plan and Meeting Schedule

Collaborative Work Plan and Meeting Schedule:

We will be having face-to-face weekly meetings to discuss the individual progress.

Meeting Format:

Face-to-Face meeting

Agenda:

- Examine the progress made since the last meeting.
- Share any difficulties or impediments encountered.
- Discuss your data analysis and modeling thoughts and discoveries.
- Define precise tasks and objectives for the following week.
- Confirm next steps and responsibilities.

Tools:

- To keep track of everything, use the Teamwork Scheduler.
- Google Docs for collaborative document sharing.
- We'll have a Google Drive folder dedicated to this project, where we'll save datasets, code scripts, and project-related papers.
- GitHub repository
- Heroku for model deployment.

Shared Responsibilities:

Team members must arrive on time and prepare for meetings to maximize collaboration. If unable to attend, members should inform the group in advance and propose a new time for rescheduling.

ABSTRACT

Deep learning techniques are used in this research to create an emotion recognition system. The photos are divided into seven emotional content categories: happy, neutral, sorrow, anger, surprise, disgust, and horror. The goal is to develop a deep learning model, especially one that uses convolutional neural networks (CNNs), that is capable of reliably categorizing these emotional expressions, even when they are subtle or complicated. The research intends to gather complete understanding of emotional states while solving the limits of cross-cultural emotion recognition by merging multiple data sources.

This project's practical relevance stems from its prospective applications, which include sentiment analysis for customer feedback, mental health evaluation, and human-computer interaction. Finally, these applications will help to design more empathic and responsive AI systems. This project's major goal is to enhance emotion detection technology, which has the potential to improve human-AI interactions through a variety of applications such as human-computer interaction, tailored user experiences, mental health screening, and sentiment analysis. Emotion detection using several data sources.

Motivation:

Emotions are important in human communication and relationships. Machines can communicate and interact with humans more effectively if they recognize emotions. This is especially crucial in the development of artificial intelligence (AI) systems that will interact with people. Deep learning is an effective method for extracting complicated patterns from data. Convolutional neural networks (CNNs) are a sort of deep learning model that excels at emotion detection. CNNs can learn to extract relevant information from facial photos for emotion recognition. Deep learning emotion detection has a wide range of possible applications. It has the potential to enhance customer service, education, healthcare, security, and entertainment.

Challenges in emotion detection using CNN:

- One of the difficulties in using CNN to detect emotions is that facial expressions are typically subtle and can be influenced by a range of factors such as lighting, position, and cultural background. Furthermore, settings, people may suppress or exaggerate their feelings.
- Another challenge is that there is no "correct" method to label emotions. The same facial expression might be interpreted differently by various people. This can make it challenging to compile an appropriate tagged dataset of facial photographs.

Method of Approach:

Data collection and preprocessing:

The first stage is to gather a dataset of facial photos or speech signals labeled with the emotions they represent. The dataset should be sufficiently broad and diverse to allow the model to learn to generalize to new data.

The dataset must be preprocessed after it has been collected. This could include scaling and normalizing photos, as well as extracting information from audio signals. Preprocessing the data improves model performance and reduces the danger of overfitting.

Dataset: https://www.kaggle.com/datasets/jonathanoheix/face-expression-recognition-dataset

Model architecture design:

The following phase will be to create a CNN architecture that is well-suited for emotion recognition. CNNs are a sort of deep learning model that excels at image recognition.

The following layers comprise a typical CNN architecture for emotion detection:

- **Convolutional layers** are used to extract features from input photos.
- **Pooling layers**: These layers lower the size of the convolutional layers' feature maps.
- **Fully connected layers**: These layers categorize the features according to their emotions.

CNN's design will be determined by the quantity and complexity of the dataset.

Model training:

After designing the CNN architecture, the model must be trained using preprocessed data. The model will learn to extract data features useful for emotion recognition. The model is trained by feeding it preprocessed data as well as emotion labels. The model's parameters are

then updated to minimize the loss function. This procedure is repeated until the model achieves convergence.

Model evaluation:

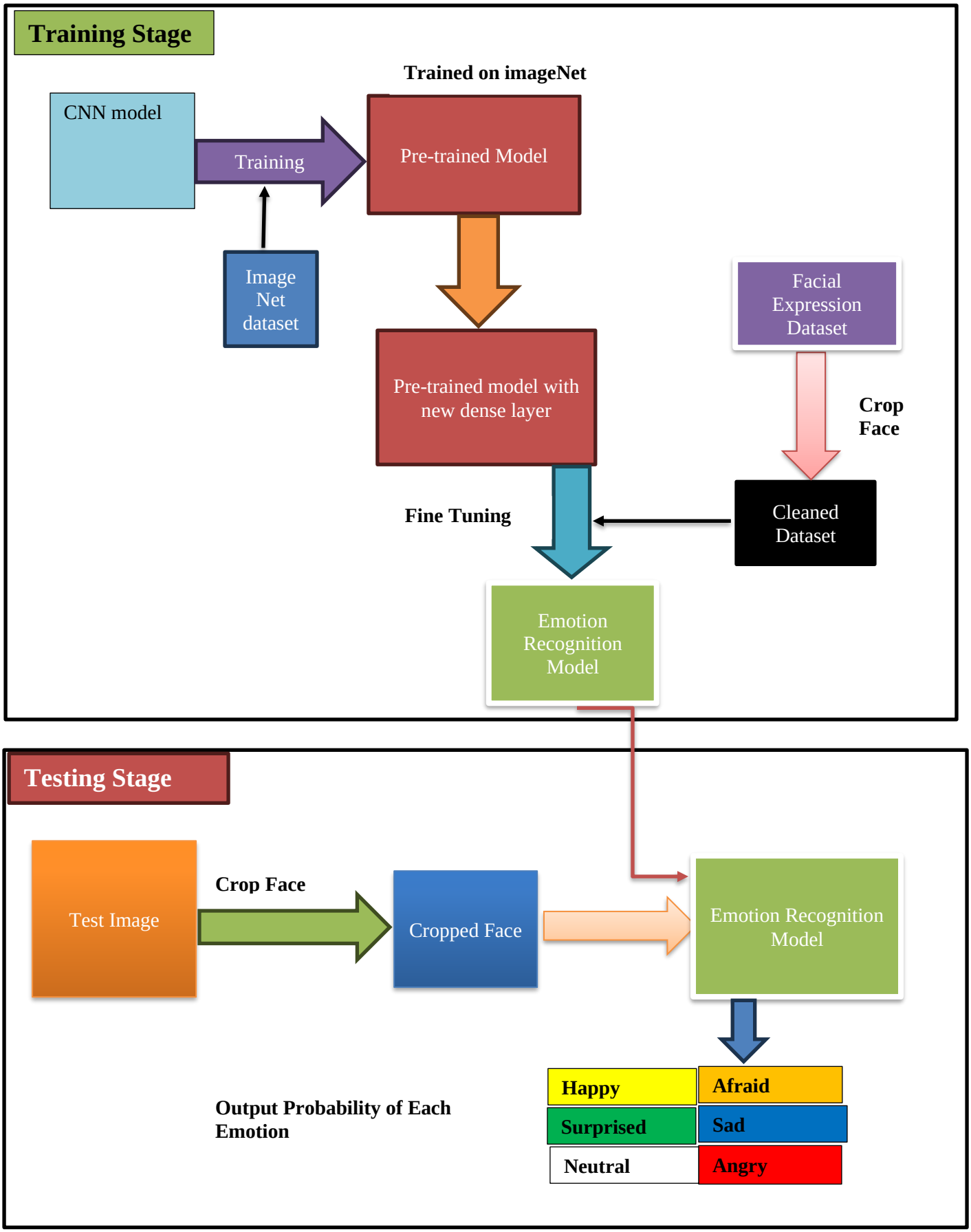
After training, the model must be assessed on a hold-out test set. This will aid in determining the model's ability to generalize to new data. The model is evaluated by feeding it test data and comparing its predictions to the ground truth labels. The model's accuracy is then calculated.

Model deployment:

The model can be deployed to production when it has been trained and evaluated. This may entail integrating the model with a web application utilizing the Flask framework and Heroku.

Once deployed, the model can be used to anticipate people's moods in real time. This data can then be used to improve areas such as customer service, education, healthcare, security, and entertainment.

Illustration of proposed model



Technologies And Platform

Programing Language: Python

Deployment platform : heuroku ,Flask (API)

Cloud platform : Google Collab

Dataset : Kaggle dataset

Milestones

Week 1:

- Collect or pick a facial imagine collection for emotion recognition.
- Begin gathering or organizing the information in the data set.
- Build the development environment, incorporating any crucial libraries and tools.

Week 2:

- Continue collecting data and guaranteeing a varied assortment of face expressions.
- Resize, normalize, and enrich gathered photos.
- Identify the set of data with emotion labels (for example, joyful, sad).
- Distribute the dataset into three parts: training, validation, and testing.
- Build an initial CNN model for emotion recognizing.

Week 3:

- Compile the model using suitable functions for loss and optimizers.
- On the training dataset, train the model.
- On the verification set, monitor and assess model performance.
- Based on the test's findings, fine-tune the model.
- Create a module for detecting emotions in real time.
- Capture video input (for example, with a camera) and perform preprocessing processes (grayscale, face recognition).

Week 4:

- To evaluate the model's performance, run it on the test dataset.
- Specify the most significant metrics (accuracy, precision, recall, and F1-score).
- Confusion matrices and performance graphs should be created.
- Integrate the learned model to predict emotions in real time.
- Select a platform for distribution (web, mobile, or desktop).
- Create a simple user interface for engaging with the emotion recognition system.
- Prepare detailed project documentation, including dataset specifics, model design, and deployment instructions. Create a simple project report that summarizes the results, issues, and improvements.

Incremental Features

- Creating a personalized user interface for presenting emotion forecasts, offering explanations, and improving the user experience.
- Collect and use user input on the model's predictions to improve model accuracy and satisfaction with users.
- Create customized applications for scenarios, such as measuring mental health, evaluating feedback from users, or offering amusement.
- Implement security and confidentiality protections, especially if the work offers gathering and analysis of data from users. Ensure that data privacy regulations are followed, and that proper user consent has been granted.

High Level Architecture