CNN Based Real-time Customer Emotion Detection System

Jul Jalal Al-Mamur Sayor*, Nishat Tasnim Shishir†
*†Department of Internet of Things and Robotics Engineering, Bangabandhu Sheikh Mujibur Rahman Digital University, Bangladesh
Email: 1901029@iot.bdu.ac.bd*, 1901030@iot.bdu.ac.bd†

***Abstract*—In the age of digital communication and customer-centric business models, understanding the emotions of our customers is crucial. This is an innovative machine learning project that leverages Flask to create a web-based platform for real-time customer emotion detection. This project aims to provide businesses with a powerful tool to gauge customer sentiment and enhance the overall customer experience. This project utilizes state-of-the-art machine learning techniques to accurately detect and categorize a wide range of emotions, from happy to disgust. It processes visual customer interactions with valuable insights to improve their products and services.**

***Index Terms*—Machine Learning Model, Flask, Customer-centric business models, Emotion Detection.**

I. Introduction

In today's digital era, businesses face an ever-increasing challenge in understanding and responding to customer emotions.Most of the times (roughly in 55 percent cases) [1], the facial expression is a nonverbal way of emotional expression, and it can be considered as concrete evidence to uncover whether an individual is speaking the truth or not [2]. In an age defined by rapid technological advancements and the integration of artificial intelligence, the once-fanciful notion of human-robot interaction has become a pivotal element across a spectrum of industries, with the service sector reaping substantial benefits.The integration of robots and artificial intelligence into businesses has led to numerous advancements in efficiency, productivity, and customer service. However, a critical element remains to be addressed the ability of robots to understand and respond to human emotions effectively. On the otherhand with the growth of online platforms, the expression of customer sentiments has shifted significantly towards visual forms, such as images and videos. With the growth of online platforms, the expression of customer sentiments has shifted significantly towards visual forms, such as images and videos. Customers frequently convey their feelings, reactions, and experiences through images posted on social media, product reviews, and various digital platforms. To remain customer-centric, businesses must gain a deep understanding of these visual expressions and promptly respond to customer emotions. The emergence of image-based communication has reshaped the landscape of customer feedback and engagement. Therefore, it is vital for businesses to possess the capability to analyze and interpret customer emotions in real time using image data. This research project delves into the dynamic realm where customer emotions, as expressed through images, meet state-of-the-art technology.The objectives of the project are given below:

- Develop a robust emotion detection system that can recognize and interpret human emotions.
- Implement real-time capabilities, allowing the system to process and respond to customer emotions as they occur, facilitating immediate interactions.
- Enhance human-robot interactions by integrating the system with robots to understand and respond empathetically to customers' emotional states, thereby improving the quality of service in industries like retail and hospitality.
- Create a versatile emotion detection model that can be integrated seamlessly with existing robots and online platforms, ensuring compatibility and ease of adoption.
- Develop a user-friendly and intuitive interface for businesses to implement the emotion detection system effectively, making it accessible for a wide range of applications.
- Establish a framework for continuous improvement and adaptation, enabling the system to evolve alongside changes in customer preferences and emotional expressions.

The rest of the paper is structured as follows. Section II describes recent related works. Section III provides an in-depth analysis of the methodology and system design. Section IV describes the performance measure of the model.section V shows the result and outcome of the system. Section VI provides a look of model interpretability. Section VII demonstrates the discussion of the paper where we contrast our work with some recent state-of-the-art works. And finally, section VIII includes the conclusion of the work.

II. Related Works

Human emotions are classified into seven basic categories: neutral, surprise, disgust, fear, anger, sadness, and happiness. Facial muscles are intricate structures that are used to transmit emotions in humans [3].Many papers using deep learning for facial expression processing have been published in recent years [4], [5]. In the context of human robot interaction a robot must be able to read the facial expressions of the people it is dealing with in order to accomplish accurate human-robot communication. In [6] the paper's goal is to use computer vision based on deep convolutional neural networks to create an end-to-end pipeline for human-NAO robot interaction. The goal of the paper is to develop a simple pipeline by optimising various types of convolutional neural networks (CNNs) for better accuracy, generalisation, and inference speed. This is achieved through the use of several optimisation techniques, such as the state-of-the-art rectified Adam, FER2013 database augmentation with images from other databases, and asynchronous threading at inference time using the Neural Compute Stick 2 preprocessor. In [7] the project works to enhance human-robot interaction (HRI), robots must be able to recognise human emotions. This paper suggests a humanoid robot emotion identification system. Through the use of a camera that records individuals' faces, the robot is able to identify their emotions and react accordingly. Six fundamental emotions are taught to the emotion identification system, which is built on a deep neural network: happiness, rage, disgust, fear, sorrow, and surprise. Initially, a convolutional neural network (CNN) is trained on a large number of static images in order to extract visual information. Secondly, a long short-term memory (LSTM) recurrent neural network is employed to ascertain the correlation between the six fundamental emotions and the alteration of facial expressions in picture sequences. In order to better detect human behaviours in e-business, the work [8] provide a face expression recognition model. In this work that uses fuzzy approaches. This technique proposes to categorise images using a fuzzy clustering model once the characteristics that are used as inputs into a classification system have been extracted. One of the predetermined emotion categories is the model's output.

III. Methodology and System Design

In this section, we discuss the technologies, working procedure and the methodology of the project.

A. Technologies and Materials

1) Google Colab: Google Colab is used as an online environment for running the model.

2) Python: Python is used programming language for implementing various components.

3) TensorFlow and Keras: TensorFlow and Keras are using for building and training the emotion detection model.

4) Flask: A lightweight Python web framework is used for building the web application.

5) HTML and CSS: HTML and CSS are used for designing the web application's user interface.

6) Dataset: In this project the "Face expression recognition dataset" is used where there is total 35.9K files and two directories named "train" and "validation". Each directories contains seven other directories. The directory "train" contains angry(3993 files), disgust(436 files), fear(4103 files), happy (7164 files), neutral(4982 files), sad(4938 files), surprise(3205 files). The directory validation contains angry(960 files), disgust(111 files), fear(1018 files), happy(1825 files), neutral(1216 files), sad(1139 files), surprise(797 files)

B. Working Procedure

The work commences by collecting data from the "Face Expression Recognition" dataset on Kaggle, which includes images representing seven types of emotions. Python serves as the foundational technology for the project, from data collection to model deployment. The collected data undergoes preprocessing techniques.For the core of the project, a Convolutional Neural Network (CNN) model is developed, leveraging TensorFlow and Keras.The model is trained using the mentioned Kaggle dataset. With the trained CNN model in place, the project transitions to real-time emotion detection. To achieve this, we create a web application using Flask, HTML and CSS. The interplay between the model and the web application facilitates this seamless real-time emotion analysis.Users receive feedback on their emotional expressions via the web application's interface. The project ensures continuous monitoring to assess real-time performance and user satisfaction. The basic workflow of the project is given in the figure 1

C. Methodology

1) Exploratory Data Analysis and visualization: The "Face expression recognition dataset" is used in this project where there is total 35.9K files and two directories named "train" and "validation". Each directories contains seven other directories named angry, happy, neutral, sad, disgust, surprise and fear. The complete dataset, obtained from the 'Face Expression Recognition' dataset on Kaggle, is extensive. Due to practical constraints, a representative portion of the data was chosen to ensure a smooth and manageable project execution. For the training we use 2525 images from the seven files (figure 2) and for the training we use 2495 images from the seven files (figure 3).

2) Preprocessing and Feature Engineering:

- Dataset Balancing The original dataset has a significant class imbalance. To balance the data for model training we extract a limited number of images (300) from each label. The balanced data is shown in figure 4
- Normalization We also normalize the pixel values of the images by dividing them by 255. Normalization is a common preprocessing step that scales the pixel values to a range between 0 and 1, making it easier for the neural network to learn.

3) Feature Extraction:

- Label Encoding The emotion labels from the dataset are encoded into numerical values using the LabelEncoder from scikit-learn. Emotions like 'angry', 'disgust', 'fear', etc., are converted to numerical labels (0, 1, 2, etc.). This encoding is necessary for model training.
- One-Hot Encoding One hot encoding is one method of converting data to prepare it for an algorithm and get a better prediction.After the label encoding, we use a function from Keras which is applied to the numerical labels. This function converts the numerical labels into one-hot encoded vectors, where each class (emotion) is

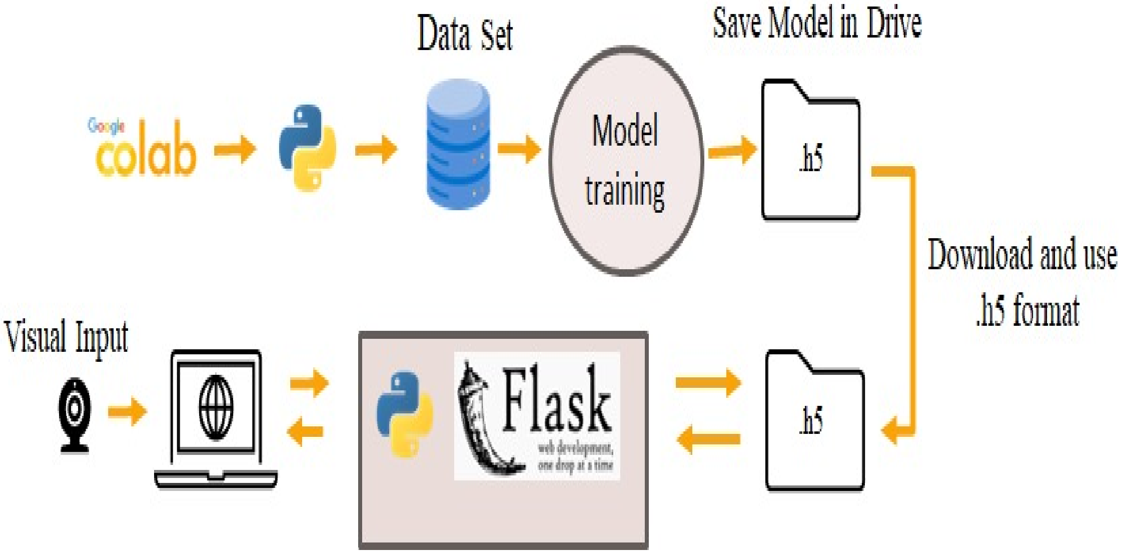


Fig. 1: Basic Workflow of the project

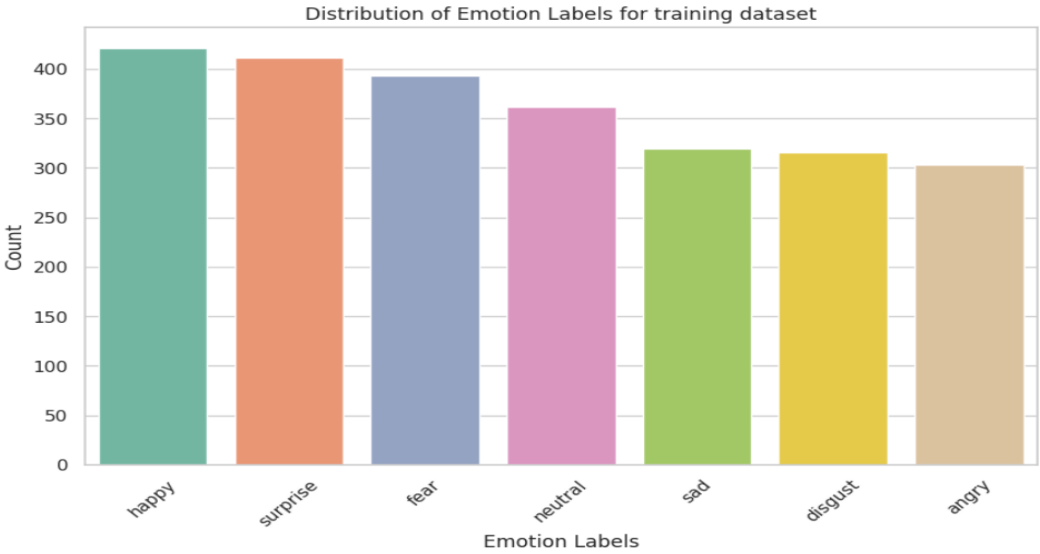


Fig. 2: Distribution of Emotion Labels for training dataset

represented as a binary vector with a ’1’ at the index corresponding to the class and ’0’s elsewhere.

4) Model Building: We use Convolutional Neural Network (CNN) architecture for building the model.The model architecture consists of convolutional layers, max-pooling layers, and fully connected layers. CNNs are particularly well-suited for this task due to their ability to automatically learn and extract intricate patterns and features from images. These deep learning networks consist of multiple convolutional layers that apply filters to the input images, progressively capturing hierarchical features. Subsequently, max-pooling layers down-sample the feature maps, retaining the most relevant information. The extracted features are then passed through fully connected layers, enabling the model to make emotion predictions.

- Convolutional Layers We use four convolutional layers (Conv2D) that are responsible for detecting various features in the input images. Convolutional layers, such as Conv2D, are fundamental for feature extraction in image data. They apply filters to detect patterns, edges, and textures in the input images. These layers use the ReLU activation function for non-linearity.ReLU is computationally efficient and introduces sparsity. It only activates when the input is positive, effectively ”switching off” negative values. This sparsity can lead to a more efficient learning process as many neurons remain inactive during

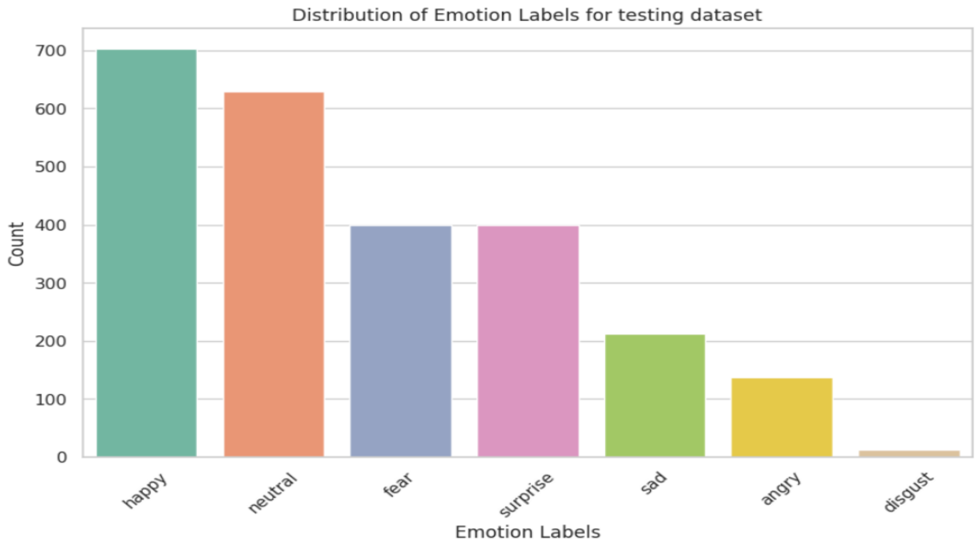


Fig. 3: Distribution of Emotion Labels for testing dataset

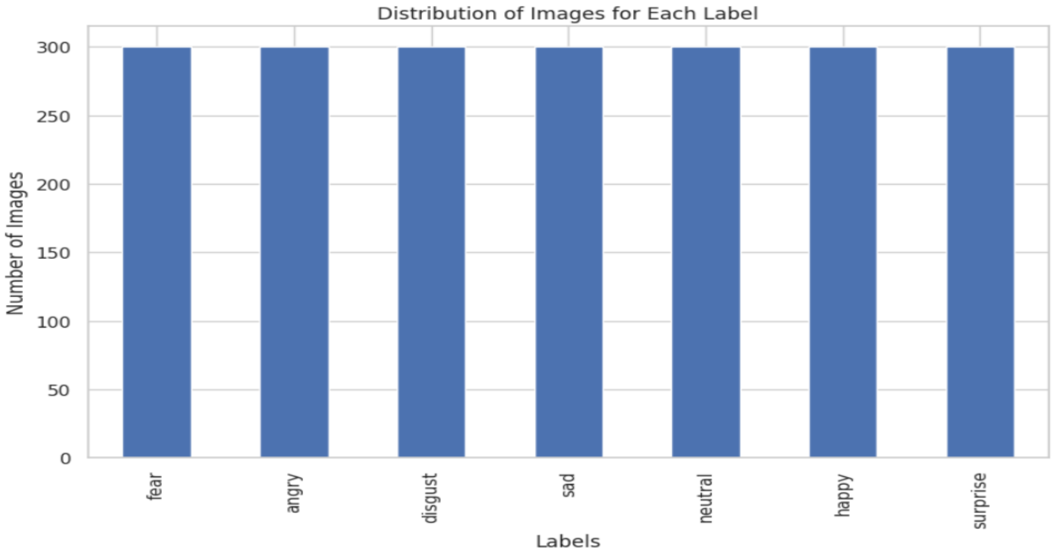


Fig. 4: Dataset Balancing

training. In this layer the kernel size is (3,3) and the input shape is (48,48,1).

- Max-Pooling Layers Max-pooling layers (MaxPooling2D) follow convolutional layers to reduce the spatial dimensions of the feature maps and retain the most important information.MaxPooling layers play a crucial part in down-sampling the feature maps generated by the convolutional layers. By selecting the maximum value in each local region, MaxPooling helps retain the most significant features while reducing computational complexity. This process is particularly important for efficiently capturing salient details in facial expressions. The pool size is (2,2).
- Dropout Layers Dropout layers (Dropout) are included to prevent overfitting by randomly setting a fraction of input units to zero during training. Overfitting occurs when a model becomes too specialized in learning the training data, which may lead to poor generalization to new, unseen data. Dropout layers work by randomly deactivating a fraction of neurons during training, effectively preventing the network from relying too heavily on specific neurons. This encourages the network to learn more robust and generalized features, ultimately improving the model's ability to accurately recognize emotions in a wide range of facial expressions.

- Flatten Layer After the convolutional and max-pooling layers, a Flatten layer is used to transform the 2D feature maps into a 1D vector.
- Fully Connected Layers The model contains fully connected (Dense) layers for making predictions. These layers have ReLU activation functions as well. Fully connected layers are responsible for learning complex patterns and relationships in the data. They can recognize higher-level features that are not directly evident in the raw input.
- Output Layer The final Dense layer has 7 units (one for each emotion class) with a softmax activation function, which is typical for multi-class classification tasks. The softmax function serves the critical role of producing a probability distribution over the possible classes. This is a crucial step in the CNN's decision-making process, as the last layer is responsible for making the ultimate prediction regarding the class of an input image. By applying the softmax function, the CNN ensures that the output of the model is a set of class probabilities, indicating the likelihood of the input belonging to each class. This probability distribution facilitates not only class prediction but also a measure of confidence in the model's decision, making it a fundamental component of the classification process.
  The model of the project is shown in figure 5

5) Model Deployment: In our project, we employed Flask, HTML, and CSS to deploy our emotion recognition model. Flask, a Python web framework, served as the backend, providing an API for user interactions. HTML is used to structure the web interface. CSS is used to enhance the visual appeal and user-friendliness of the web interface. This deployment allows our trained model to make real-time predictions, providing a user-friendly platform for emotion recognition.

IV. Performance Measure

A. ROC

The ROC curve is a graphical representation of the classifier's ability to distinguish between different classes by varying the decision threshold. Specifically, we use the ROC curve to evaluate our model's ability to differentiate between seven distinct emotional states, such as happiness, sadness, anger, and more. The ROC curve is created by plotting the True Positive Rate (Sensitivity) against the False Positive Rate at various threshold values.The area under the ROC curve (AUC) serves as a quantitative measure of our model's performance. A higher AUC indicates better discriminative power, with a maximum value of 1 representing a perfect classifier. By analyzing the ROC curve and AUC, we can assess the model's ability to correctly identify emotions, helping us make informed decisions about the model's suitability for real-world emotion recognition applications. Our model's ROC plotting is shown in 6

B. Confusion matrix

It provides a detailed breakdown of how well the model classifies instances into different emotional categories. The confusion matrix is particularly valuable in the context of our project, where we aim to accurately identify emotions, such as happiness, sadness, anger, and more, from facial expressions.The confusion matrix consists of rows and columns, with each row representing the actual (true) class and each column representing the predicted class. It quantifies various performance metrics, including true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). These metrics allow us to evaluate the model's ability to correctly classify emotions and identify any potential misclassifications. Our model's confusion matrix is shown in 7a

C. Classification Report

It provides detailed insights into the model's precision, recall, F1-score, and support for each emotional class, enabling a thorough evaluation of its accuracy and reliability.

- Precision: Precision measures the proportion of true positive predictions (correctly identified emotions) out of all positive predictions. In our project, precision helps us understand how well the model avoids false positives when classifying emotions.
- Recall: Recall, also known as sensitivity or true positive rate, quantifies the proportion of true positive predictions out of all actual positive instances. It indicates the model's ability to capture and correctly classify emotions.
- F1-Score: The F1-score is the harmonic mean of precision and recall, providing a balanced measure of a model's accuracy. It is particularly useful when dealing with imbalanced datasets, ensuring a fair assessment of the model's performance.
- Support: Support represents the number of instances in each class, offering insights into the distribution of emotions in the dataset. This information helps us understand the model's ability to recognize less frequent emotions.

Our model's classification Report is shown in 7b

D. Accuracy

The training accuracy, test accuracy and the f1 score of our model is respectively 0.9990476190476191 (99 percents), 0.4090909090909091(40.9%), 0.4205398804939781(42%).

V. Results and Outcomes

The developed emotion recognition model is effective in recognizing and categorizing emotions accurately from facial expressions. It can detect the facial expression in real-time. The predictions of the model is shown in the figure 8.

VI. Model Interpretability

In our project, we utilized Grad-CAM for visualizing the regions in images that influenced the emotion recognition model's predictions. By generating heatmaps, Grad-CAM highlighted the facial features crucial for the model's decision-making, offering insights into its reasoning. This technique

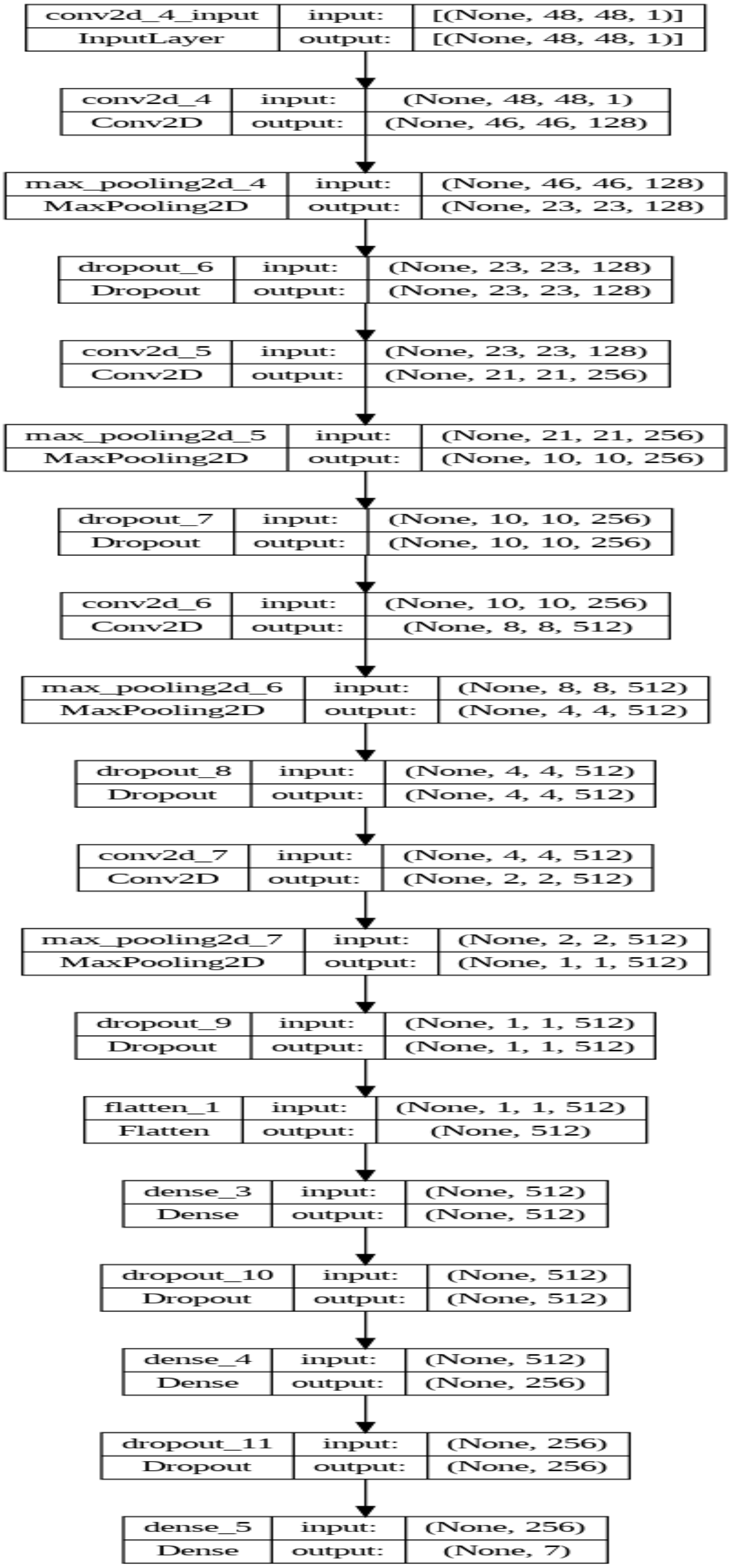

conv2d_4_input | input: | [(None, 48, 48, 1)]
InputLayer | output: | [(None, 48, 48, 1)]
conv2d_4 | input: | (None, 48, 48, 1)
Conv2D | output: | (None, 46, 46, 128)
max_pooling2d_4 | input: | (None, 46, 46, 128)
MaxPooling2D | output: | (None, 23, 23, 128)
dropout_6 | input: | (None, 23, 23, 128)
Dropout | output: | (None, 23, 23, 128)
conv2d_5 | input: | (None, 23, 23, 128)
Conv2D | output: | (None, 21, 21, 256)
max_pooling2d_5 | input: | (None, 21, 21, 256)
MaxPooling2D | output: | (None, 10, 10, 256)
dropout_7 | input: | (None, 10, 10, 256)
Dropout | output: | (None, 10, 10, 256)
conv2d_6 | input: | (None, 10, 10, 256)
Conv2D | output: | (None, 8, 8, 512)
max_pooling2d_6 | input: | (None, 8, 8, 512)
MaxPooling2D | output: | (None, 4, 4, 512)
dropout_8 | input: | (None, 4, 4, 512)
Dropout | output: | (None, 4, 4, 512)
conv2d_7 | input: | (None, 4, 4, 512)
Conv2D | output: | (None, 2, 2, 512)
max_pooling2d_7 | input: | (None, 2, 2, 512)
MaxPooling2D | output: | (None, 1, 1, 512)
dropout_9 | input: | (None, 1, 1, 512)
Dropout | output: | (None, 1, 1, 512)
flatten_1 | input: | (None, 1, 1, 512)
Flatten | output: | (None, 512)
dense_3 | input: | (None, 512)
Dense | output: | (None, 512)
dropout_10 | input: | (None, 512)
Dropout | output: | (None, 512)
dense_4 | input: | (None, 512)
Dense | output: | (None, 256)
dropout_11 | input: | (None, 256)
Dropout | output: | (None, 256)
dense_5 | input: | (None, 256)
Dense | output: | (None, 7)

Fig. 5: Model Plot

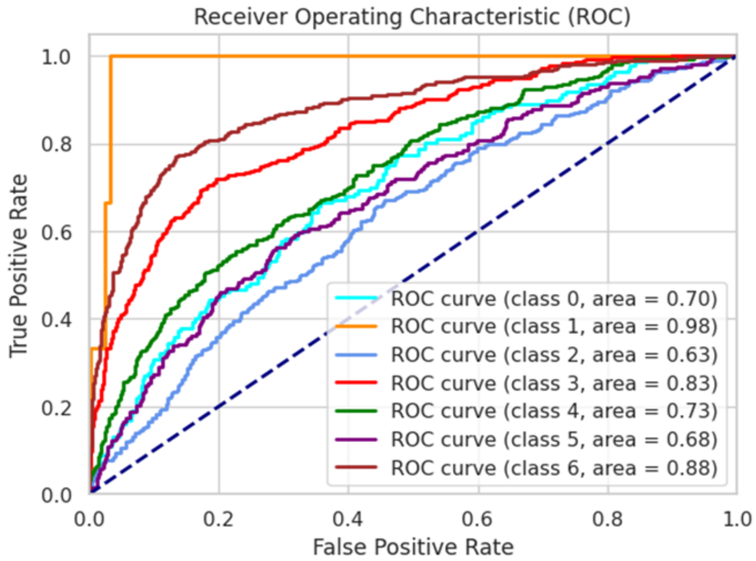
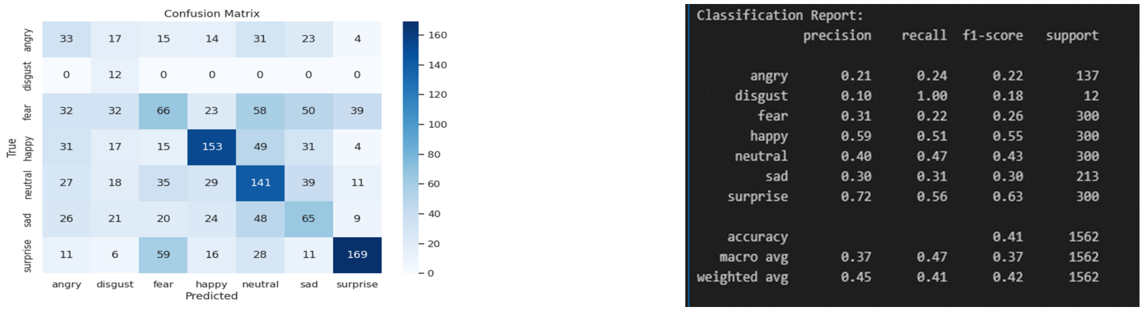


Fig. 6: ROC plot



	angry	disgust	fear	happy	neutral	sad	surprise
angry	33	17	15	14	31	23	4
disgust	0	12	0	0	0	0	0
fear	32	32	66	23	58	50	39
happy	31	17	15	153	49	31	4
neutral	27	18	35	29	141	39	11
sad	26	21	20	24	48	65	9
surprise	11	6	59	16	28	11	169

(a) Confusion matrix

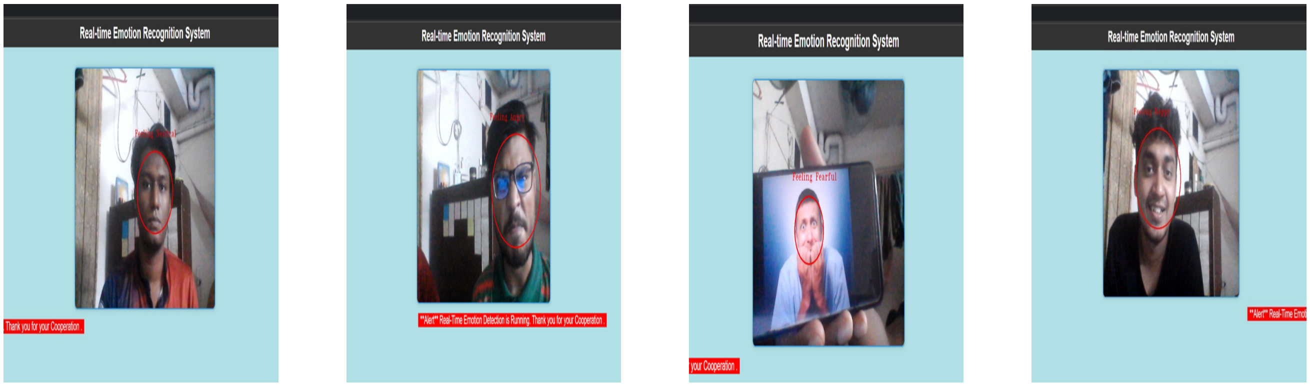
```
Classification Report:
              precision    recall  f1-score   support

       angry       0.21      0.24      0.22       137
     disgust       0.10      1.00      0.18        12
        fear       0.31      0.22      0.26       300
       happy       0.59      0.51      0.55       300
     neutral       0.40      0.47      0.43       300
         sad       0.30      0.31      0.30       213
    surprise       0.72      0.56      0.63       300

    accuracy                           0.41      1562
   macro avg       0.37      0.47      0.37      1562
weighted avg       0.45      0.41      0.42      1562
```

(b) Classification Report

Fig. 7: Performance Measure


Real-time Emotion Recognition System
Thank you for your Cooperation
Real-time Emotion Recognition System
Alert Real-Time Emotion Detection is Running. Thank you for your Cooperation
Real-time Emotion Recognition System
Feeling Fearful
your Cooperation
Real-time Emotion Recognition System
Alert Real-Time Emo

(a) Emotion(Neutral)

(b) Emotion(Angry)

(c) Emotion(Fear)

(d) Emotion(Happy)

Fig. 8: Predictions of the model

provided transparency into the CNN's decision process, enabling a better understanding of how it interprets emotions from facial expressions. Such interpretability is crucial for enhancing the trustworthiness and explainability of AI systems, making Grad-CAM a valuable tool in our emotion recognition project.Our model's interpretability is shown through the figures 9 , 10.

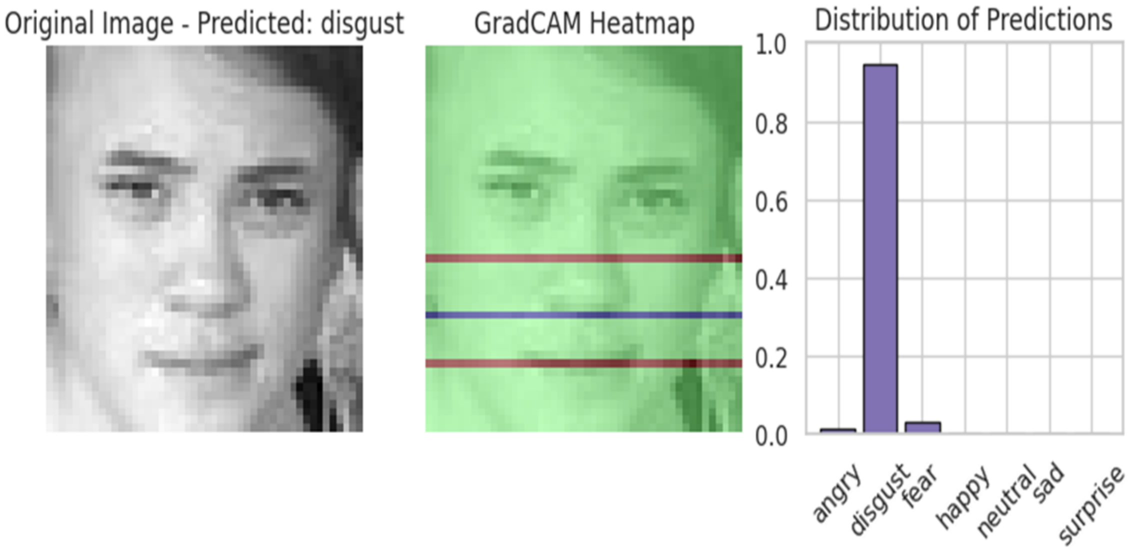


Fig. 9: GradCAM(Prediction:Disgust)

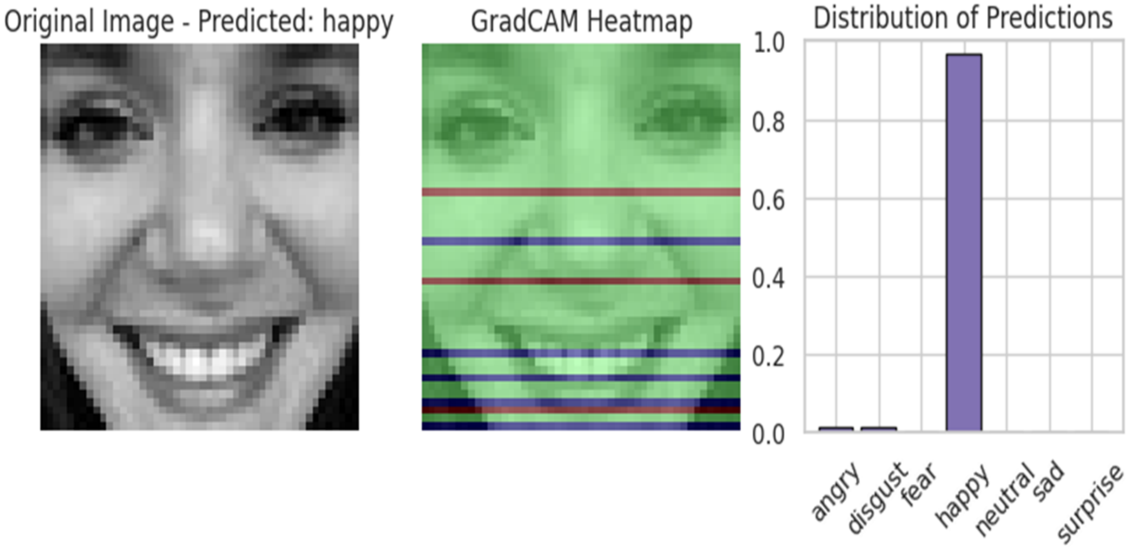


Fig. 10: GradCAM(Prediction:Happy)

VII. Discussion

A. Limitations

- Accuracy: The achieved test accuracy of 40.9% is below the average for the Face expression recognition dataset, indicating room for improvement.
- Data Quality: The model's performance may be affected by variations in data quality, such as viewing angles and image rotations.
- Dataset Size: The Face expression recognition dataset's size and diversity may limit the model's ability to generalize to a broader range of facial expressions.
- Real-time Recognition: The system's real-time facial expression recognition may have latency issues that need to be addressed for practical applications.
- Limited Emotions: The model predicts 7 basic emotions, and its applicability to more nuanced emotional states may be limited.

B. Future Scope

There are several avenues for future research and development that can build upon the work presented in this project. Some potential directions for future work include:

- Further Research: Conducting more in-depth investigations into our CNN model to gain a deeper understanding and uncover additional insights.
- Enhancements: Implementing improvements to overcome the limitations mentioned earlier.

- Integration: Exploring opportunities to integrate this project with other related initiatives or technologies in the field to create a more comprehensive solution.
- Scaling: Evaluating the feasibility of scaling up the project to address larger or more complex scenarios, which can have a broader impact.
- New Applications: Investigating how the findings and techniques from this project can be applied to other domains or industries, potentially leading to innovative solutions.
- User Feedback: Collecting feedback from users or stakeholders and incorporating their suggestions for enhancing the project's usability and effectiveness.

VIII. Conclusion

In conclusion, this project has successfully utilized the Face expression recognition dataset to develop a Convolutional Neural Network (CNN) model for the prediction of 7 human facial expressions. The achieved test accuracy of 40.9% is notable, particularly considering that the average accuracies on the in dataset typically fall within the range of 60 % ± 5 %. This indicates that the CNN model is approaching a level of accuracy that can be considered satisfactory for its intended purpose. The project's key findings and outcomes are significant for the field of facial emotion recognition, as they demonstrate the potential of deep learning techniques in accurately identifying human facial expressions from real-time customaer's visual input. The ability to predict emotions from facial expressions has wide-ranging practical applications, including affective computing, human-computer interaction, and emotion recognition in various domains.

To further improve this project and its outcomes, several recommendations can be considered:

- Parameter Tuning: Exploring the addition of new parameters to the CNN model and optimizing the existing ones to enhance accuracy and robustness.
- Learning Rate Optimization: Adjusting the learning rate and implementing adaptive learning strategies to improve the model's convergence speed and accuracy.
- Noise and Illumination Handling: Develop techniques to adapt the system to low-light conditions and mitigate noise in facial images, thereby improving the model's performance in real-world scenarios.
- Model Complexity: Experimenting with the inclusion of additional layers in the CNN architecture and extending the number of training epochs to achieve higher accuracy, while monitoring for signs of overfitting.
- : Training and testing the CNN model on other available datasets to assess its generalization capabilities and applicability across different facial expression recognition tasks.

References

[1] A. Mehrabian, *Nonverbal communication*. Routledge, 2017.

[2] M. Bartlett, G. Littlewort, E. Vural, K. Lee, M. Cetin, A. Ercil, and J. Movellan, "Data mining spontaneous facial behavior with automatic expression coding," in *Verbal and Nonverbal Features of Human-Human and Human-Machine Interaction: COST Action 2102 International Conference, Patras, Greece, October 29-31, 2007. Revised Papers*. Springer, 2008, pp. 1–20.

[3] S. Modi and M. H. Bohara, "Facial emotion recognition using convolution neural network," in *2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE, 2021, pp. 1339–1344.

[4] A. Raghuvanshi and V. Choksi, "Facial expression recognition with convolutional neural networks," *CS231n Course Projects*, vol. 362, 2016.

[5] Z. Yu and C. Zhang, "Image based static facial expression recognition with multiple deep network learning," in *Proceedings of the 2015 ACM on international conference on multimodal interaction*, 2015, pp. 435–442.

[6] D. O. Melinte and L. Vladareanu, "Facial expressions recognition for human–robot interaction using deep convolutional neural networks with rectified adam optimizer," *Sensors*, vol. 20, no. 8, p. 2393, 2020.

[7] T.-H. S. Li, P.-H. Kuo, T.-N. Tsai, and P.-C. Luan, "Cnn and lstm based facial expression analysis model for a humanoid robot," *IEEE Access*, vol. 7, pp. 93 998–94 011, 2019.

[8] A. Jamshidnejad and A. Jamshidined, "Facial emotion recognition for human computer interaction using a fuzzy model in the e-business," in *2009 Innovative Technologies in Intelligent Systems and Industrial Applications*. IEEE, 2009, pp. 202–204.