INDIAN INSTITUTE OF INFORMATION TECHNOLOGY
ALLAHABAD

COURSE PROJECT THESIS

Facial Emotion Recognition

Author:
Banoth Naveen IIT2021045

Supervisor:
Dr. Muneendra Ojha

A thesis submitted in partial fulfillment of the requirements for Course completion

Researcher:Banoth Naveen
Department of Information Technology

June 7, 2024

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY ALLAHABAD

Abstract

Dr.Muneendra Ojha
Department of Information Technology

Facial Emotion Recognition

by Banoth Naveen IIT2021045

This project uses machine learning algorithms, specifically Convolutional Neural Networks (CNNs), to recognize facial emotions. The goal is to create a real-time application that can recognize emotions in facial expressions with accuracy. The project tackles the difficulties of automated facial recognition systems while highlighting the importance of facial expression recognition in mental health diagnosis, artificial human expression, and human-machine interaction. The project uses CNNs to identify emotions, such as anger, disgust, fear, happiness, sadness, surprise, and neutral, by leveraging the FER2013 dataset from Kaggle, which consists of grayscale images classified into seven emotions. Improving efficiency and accuracy in foretelling human emotions in real-time situations is the aim.

Contents

Chapter 1

Problem Statement

1.1 Objective

Facial Emotion Recognition is a project that uses machine learning algorithms to detect emotions from facial expressions. This project aims to develop an application that can detect emotions from real-time images. It uses Machine and Deep Learning Techniques like Convolutional Neural Networks to train the model.

1.2 Brief Intro and Background

The Human facial expressions are important for visually expressing a lot more information. Facial expression recognition is essential in the field of human-machine interaction. Automated facial recognition systems have many applications, including understanding of human behavior, diagnosing mental disorders, and synthetic human expression. Identifying facial expressions through computers with high detection rates is still a challenging task.

Emotion recognition plays a crucial role in the era of Artificial intelligence and Internet of things.It has enormous potential for robotics, healthcare, human-computer interaction,behavioral modeling and biometric security. Feelingsalgorithms identify emotions based on facialexpressions, textual information, voice, bodily language, brain, or heartsignals. Basic feelings, attitude, and control overEmotions and the capacity to elicit them may also beinvestigated in order to analyze feelings. This project identifies several forms of machine learning, both supervised and unsupervisedmethods for classifying emotions and extracting features.A comparative examination of the different machine learning methods employed in the cited articles has also been conducted. It reveals the extent and uses of automated emotion detectionsystems throughout a range of industries. This report also addresses a number ofcharacteristics to improve the precision, safety, and effectiveness of the framework.

In this project, we have used a variety of intensive deep learning techniques (convolutional neural networks) to identify the main seven human emotions: ANGER, DISGUST, FEAR, HAPPY, NEUTRAL, SAD, SURPRISE.

1.3 Data Source

1.3.1 Data Set

The dataset used in this project is the FER2013 (Facial Expression Recognition 2013 Dataset) dataset from Kaggle. The dataset contains 35,685 examples of 48x48 pixel grayscale images of faces divided into train and test datasets. Images are categorized based on the emotion shown in the facial expressions (happiness, neutral, sadness, anger, surprise, disgust, fear). The faces have been automatically registered so that the face is more or less centred and occupies about the same amount of space in each image. The task is to categorize each face based on the emotion shown in the fa- cial expression into one of seven categories (0-Angry, 1-Disgust, 2=Fear, 3-Happy, 4-Sad, 5-Surprise, 6-Neutral).

1.3.2 Problem

Emotions are often context-dependent, and recognizing emotions accurately requires an understanding of the surrounding context. A facial expression may have different meanings based on the situation, making it challenging to develop a one-size-fits-all model.

1.3.3 Link

Kaggle Dataset Link

1.4 Project Overview

1.4.1 Objective

Our objective is to predict the expression of the human face in real-time as fast and as accurately as possible.

Our goal is to predict the expression of a face in the image as accurately as possible. The higher the test accuracy, the better our model will perform in the real world.

1.4.2 Approach

We will utilize Convolutional Neural Networks (CNNs) to train the model for emotion classification. CNNs are particularly well-suited for image data, making them suitable for tasks involving facial emotion recognition or any application where spatial relationships within the data are crucial.

Chapter 2

Literature Review

2.1 Paper 1

S L Happy;Anjith George;Aurobinda Routray, “ A Real Time Facial Expression Classification System Using Local Binary Patterns.,”2012 IEEE.

This research paper proposes a facial expression classification algorithm that classifies different facial expressions using Principal Component Analysis (PCA), employs the Haar classifier for face detection, and uses the Local Binary Patterns (LBP) histogram of varying block sizes of a face image as feature vectors. Given its low computer complexity, the expression classification technique is implemented in real time. Facial expressions differ from person to person in terms of their strength and variety, hence a customized method to facial expression analysis is suggested. The method categorizes six fundamental emotions—happiness, sorrow, disgust, fear, surprise, and anger—using grayscale frontal face photos of individuals.

2.1.1 Conclusion

For the categorization of facial expressions, a real-time bespoke algorithm has been created. Higher recognition accuracy is achieved by the experimental selection of the block size for LBP feature extraction. The testing findings show that facial expressions may be recognized with an more accuracy than when LBP characteristics are used. Higher accuracy is achieved by extracting both local and global features from the face picture using the block LBP histogram features. Accurate classification may be achieved by combining additional cascaded block LBP features with high quality images. However, because of the related temporal complexity, it is realistically challenging to execute in real time.

2.2 Paper 2

Facial Emotion Recognition: A multi-task approach using deep learning research done by group of people from Department of Computer Engineering K. J. Somaiya College of Engineering Mumbai,Maharashtra In this work, a single model produces labels for emotion, gender, age, and race simultaneously using a Convolutional Neural Network (CNN) architecture for multi-task learning. Each classification job has its own output layer in the CNN’s architecture, which uses different softmax functions for each task. Seven fundamental human emotions are recognized; gender is identified as either male, female, or uncertain; race is characterized as either Asian, African-American, or Caucasian; and age estimation is broken down into five age groups. Pose normalization in pre-processing includes locating eye centers and rotating pictures to match, with a maximum rotation of 10 degrees to avoid distortion. The CNN architecture achieves a 67% cross-validation accuracy on the FER dataset by using decreasing dropout rates, which were derived from an earlier model.To maximize performance, the model uses certain weights for each task's loss function. Callbacks are used to improve training efficiency and avoid overfitting. Examples include early halting and reduction on plateau.

2.2.1 Conclusion

When training on all labels with the RAF-Db dataset, the validation accuracy in the context of multi-task learning reaches 79%. However, the accuracy drops to 53% when the same model, which was pre-trained on RAF-Db, is retrained using single-task learning on FER (which only includes emotion labels). As a result, the best multi-task learning model (which produces better outcomes) is trained only on RAF-Db. In contrast to traditional single-task learning with the same CNN architecture, this paper presents a novel approach to multi-task learning that includes gender, age, race, and emotion prediction. The results are noticeably better. Improving Convolutional Neural Networks' (CNNs') accuracy in facial emotion recognition is the main goal of this work.

2.3 Paper 3

Emotion recognition from facial expression using deep convolutional neural network by D Y Liliana State Polytechnic of Jakarta, Indonesia.According to Ekman and Friesen's definition, the eight classes of emotion that are the focus of this study on facial expression and emotion recognition are happy, sad, surprised, fear, disgust, angry, contempt, and neutral. The Extended Cohn Kanade database (CK+), which contains 10,708 photos from 123 subjects that have been resized to 100x100 pixels in preparation for input into a CNN system, is the dataset that was used. Two convolutional layers, two subsampling layers, and a fully connected layer for classification make up the suggested CNN architecture.

Different quantities of training and testing data from the CK+ database were used in the experiments. Findings indicate that mean square error (MSE) decreases noticeably with increasing training data size, while MSE is linearly correlated with testing data size. An MSE of 0.3729 was obtained from the experiment using 10,000 training data and 708 testing data. Furthermore, accuracy rates ranging from 87.73% to 98.09% were recorded for each emotion class, with an overall testing set average of 92.81%. Facial expression recognition using the CNN architecture works well and achieves good accuracy for a variety of emotion classes.

2.3.1 Conclusion

The experimental scenario yields an average accuracy rate of 92.81%, which offers a noteworthy window into the system's performance. The surprise class has the highest accuracy (98.09%), while the anger class has the lowest accuracy (87.73%). Notwithstanding these achievements, misclassifications in every class point to the need for additional system enhancement, which motivates future research projects to take a closer look at a revised architecture. In conclusion, the CK+ database is successfully used by the suggested Convolutional Neural Network (CNN) architecture to identify eight classes of facial expressions. The experiments confirm that the system performs better with larger training datasets by showing a decline in mean square error as the training data size increases. The system achieves a 92.81% overall accuracy rate, laying the groundwork for further attempts to improve CNN architecture and achieve even better results.

2.4 Paper 4

Facial emotion recognition using deep learning: review and insights by Wafa Mellouka and Wahida Handouzia published in The 2nd International Workshop on the Future of Internet of Everything (FIoE) August 9-12, 2020, Leuven, Belgium. This paper investigates in detail the increasing interest in deep learning-based facial emotion recognition (FER) among researchers. Important steps in the FER task include data processing, proposing a model architecture, and recognizing emotions. Preprocessing, a standard procedure in all cited papers, involves methods to improve training efficiency and deal with overfitting, such as resizing, cropping, normalization, and data augmentation. Many of the techniques covered in this review exhibit high accuracy—some reaching over 90%. The use of inception layers, consideration of occlusion images, residual block addition, and the benefit of using iconized faces in input are among the noteworthy contributions.

Several deep learning structures are investigated for the extraction of spatiotemporal features, such as CNN-LSTM, 3DCNN, and Deep CNN combinations. The findings indicate that the techniques put forth by Yu and Liang et al. yield better precision than those of Kim et al. Emphasizing CNN as a foundational network for FER, researchers frequently apply CNN networks for spatial data and combine CNN-RNN, especially LSTM networks, for sequential data. For CNN parameterized faces in input, the Softmax function and Adam optimization algorithm are frequently utilized.

2.4.1 Conclusion

This work presents a thorough survey of current FER research, highlighting multiple CNN and CNN-LSTM architectures. The talk focuses on the high accuracy rates that researchers have attained, indicating that machines are becoming more and more capable of deciphering human emotions and improving interactions between humans and machines. Even with its success, FER can only learn simple emotions; therefore, future research should concentrate on building more robust architectures and bigger databases that can identify a wider variety of emotions, including complex ones.

Modern research clearly shows a shift toward multimodal analysis, highlighting the significance of integrating various modalities for optimal emotion detection. Powerful multimodal deep learning architectures and databases, like the combination of audio and visual modalities investigated .

Chapter 3

Proposed Methodology

3.1 Problem Statement

Our primary objective is to predict the expression of the human face in real-time as fast and as accurately as possible. Our basic goal is to predict the expression of a face in the image as accurately as possible. The higher the test accuracy, the better our model will perform in the real world.

3.2 Pre-Processing

3.2.1 splitting dataset into train,validation and test

1. **Training:** 28709
2. **PublicTest:** 3589
3. **PrivateTest:** 3589

3.2.2 Viewing Dataset:

Emotion labels : surprise, angry, fear, disgust, happy, neutral, sad.

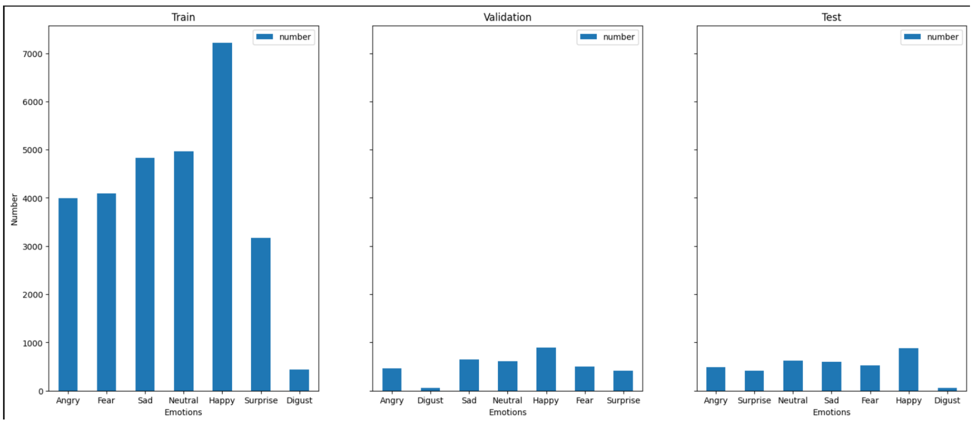

FIGURE 3.1: Dataset View

3.2.3 Data augmentation and image processing

1. **Removing Dodgy Images:**
2. **Normalization:**

- `rescale = 1./255` ensures that pixel values are scaled between 0 and 1, a common practice to normalize the input data.

3. **Rotation Range:**

   - `rotation_range = 30` specifies a range of 30 degrees for random rotation of images. This aids in making the model robust to variations in pose.

4. **Shear Range:**

   - `shear_range = 0.3` introduces shearing transformations with a range of 0.3, which can be beneficial for handling deformations.

5. **Zoom Range:**

   - `zoom_range = 0.3` allows random zooming within a range of 0.3, providing the model with variations in scale.

6. **Width and Height Shift Range:**

   - `width_shift_range` and `height_shift_range`, both set to 0.4, enable random horizontal and vertical shifts, respectively. This helps the model generalize to different spatial locations.

7. **Horizontal Flip:**

   - `horizontal_flip = True` randomly flips images horizontally. This is useful for augmenting data and improving model robustness.

8. **One-Hot Encoding:**

   - The emotion labels are one-hot encoded using the `to_categorical` function from Keras. It converts the integer labels (presumably stored in the 'emotion' column of the DataFrame) into one-hot encoded vectors. The number of classes is inferred from the length of the vector.

3.2.4 Purpose and impact of image preprocessing

Data augmentation, as implemented by the training data generator, introduces diversity into the training set by applying various transformations to images. This helps the model become more invariant to different orientations, scales, and deformations. Normalization ensures consistency in pixel values.

On the other hand, the validation data generator is used for preprocessing the validation dataset without introducing random transformations. It ensures that the validation set is processed consistently for model evaluation.

A function named CRNO (presumably standing for Convert, Reshape, Normalize, and One-hot encode) and utilizes it to preprocess image data. The function takes a DataFrame (df), representing a dataset with pixel sequences and corresponding emotion labels, and a string (dataName) indicating the type of data (e.g., train, val, test)

3.3 Models

3.3.1 Convolution Neural Networks

CNN in Machine Learning:

The extended form of artificial neural networks, known as convolutional neural networks (CNNs), is primarily used to extract features from grid-like matrix datasets. For instance, visual datasets with a significant emphasis on data patterns, such as pictures or videos

CNN architecture:

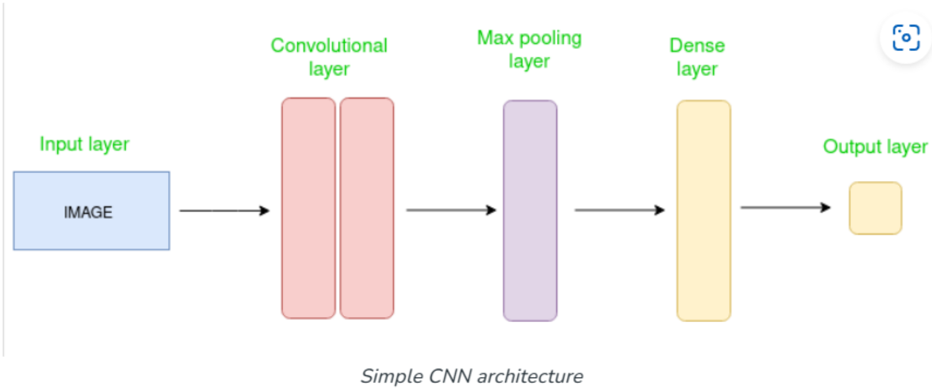

FIGURE 3.2: CNN Architecture

1. **Input layer**
   - This layer is where we feed data into our model. An image or series of images will typically be the input for CNN. This layer, which has width 32, height 32, and depth 3, contains the image's raw input.
2. **Convolutional Layers:**
   - This layer is where the feature from the input dataset is extracted. It applies to the input images a collection of learnable filters called kernels. The filters/kernels are typically 2x2 or 3x3 or 5x5 matrices in size. It computes the dot product between the kernel weight and the matching input image patch as it moves over the input image data. Ad feature maps are the result of this layer. Assuming we apply 12 filters in total to this layer, the result will be an output volume with 32 x 32 x 12 dimensions.
3. **Activation Layer :**
   - By adding an activation function to the output of the preceding layer, activation layers add nonlinearity to the network. it will apply an element-wise activation function to the output of the convolution layer. Some common activation functions are RELU: max(0, x), Tanh, Leaky RELU, etc.
4. **Pooling Layer**

- This layer is periodically inserted in the covnets and its main function is to reduce the size of volume which makes the computation fast reduces memory and also prevents overfitting. Two common types of pooling layers are max pooling and average pooling.

5. **Flattening Layer**

   - The resulting feature maps are flattened into a one-dimensional vector after the convolution and pooling layers so they can be passed into a completely linked layer for categorization or regression.

6. **Fully Connected Layer**

   - It takes the input from the previous layer and computes the final classification or regression task.

7. **Output Layer**

   - The output from the fully connected layers is then fed into a logistic function for classification tasks like sigmoid or softmax which converts the output of each class into the probability score of each class.

Advantages and Use Cases:

- Good at detecting patterns and features in images, videos, and audio signals.
- Robust to translation, rotation, and scaling invariance.
- End-to-end training, no need for manual feature extraction.
- Can handle large amounts of data and achieve high accuracy..

3.3.2 Model Using 2 Convolution Layer and 2 Dense Layers

Model Architecture Module 1:

- Convolutional Layer 1:
  - Filters: $2 \times 2 \times$ num_features
  - Kernel Size: $(3, 3)$
  - Input Shape: $(\text{width}, \text{height}, 1)$
  - Batch Normalization
  - ReLU Activation
- Convolutional Layer 2:
  - Filters: $2 \times 2 \times$ num_features
  - Kernel Size: $(3, 3)$
  - Padding: 'same'
  - Batch Normalization
  - ReLU Activation
- Max Pooling Layer:

  - Pool Size: $(2, 2)$
  - Strides: $(2, 2)$

Module 2:

- Convolutional Layer 3:
  - Filters: $2 \times$ num_features
  - Kernel Size: $(3, 3)$
  - Padding: 'same'
  - Batch Normalization
  - ReLU Activation
- Convolutional Layer 4:
  - Filters: $2 \times$ num_features
  - Kernel Size: $(3, 3)$
  - Padding: 'same'
  - Batch Normalization
  - ReLU Activation
- Max Pooling Layer:
  - Pool Size: $(2, 2)$
  - Strides: $(2, 2)$
- Flatten:
  - Flattening the output to prepare for the dense layers.

Dense Layers:

- Dense Layer 1:
  - Units: $2 \times$ num_features
  - ReLU Activation
  - Batch Normalization
- Dense Layer 2:
  - Units: num_features
  - ReLU Activation
  - Batch Normalization

Output Layer:

- Dense Layer with num_classes units and softmax activation.

Compilation:

- Categorical Crossentropy Loss
- Adam Optimizer with specified learning rate and parameters
- Accuracy metric used for evaluation

3.3.3 Model using 3 Covolution and 3 Dense Layers

Model Architecture

3.3.4 Model using 3 Convolution and 3 Dense Layers

Model Architecture

Module 1

- **Convolutional Layer 1:**
  - Filters: $2 \times 2 \times \text{num\_features}$
  - Kernel Size: $(3, 3)$
  - Input Shape: $(\text{width}, \text{height}, 1)$
  - Batch Normalization
  - ReLU Activation
- **Convolutional Layer 2:**
  - Filters: $2 \times 2 \times \text{num\_features}$
  - Kernel Size: $(3, 3)$
  - Padding: 'same'
  - Batch Normalization
  - ReLU Activation
- **Max Pooling Layer:**
  - Pool Size: $(2, 2)$
  - Strides: $(2, 2)$

Module 2

- **Convolutional Layer 3:**
  - Filters: $2 \times \text{num\_features}$
  - Kernel Size: $(3, 3)$
  - Padding: 'same'
  - Batch Normalization
  - ReLU Activation
- **Convolutional Layer 4:**
  - Filters: $2 \times \text{num\_features}$
  - Kernel Size: $(3, 3)$
  - Padding: 'same'
  - Batch Normalization
  - ReLU Activation
- **Max Pooling Layer:**
  - Pool Size: $(2, 2)$
  - Strides: $(2, 2)$

Module 3

- **Convolutional Layer 5:**
  - Filters: num_features
  - Kernel Size: $(3, 3)$
  - Padding: ’same’
  - Batch Normalization
  - ReLU Activation
- **Convolutional Layer 6:**
  - Filters: num_features
  - Kernel Size: $(3, 3)$
  - Padding: ’same’
  - Batch Normalization
  - ReLU Activation
- **Max Pooling Layer:**
  - Pool Size: $(2, 2)$
  - Strides: $(2, 2)$

Transition to Dense Layers

- **Flatten Layer:**
  - Flattens the output to prepare for the dense layers.

Dense Layers

- **Dense Layer 1:**
  - Units: $2 \times 2 \times 2 \times$ num_features
  - Batch Normalization
  - ReLU Activation
- **Dense Layer 2:**
  - Units: $2 \times 2 \times$ num_features
  - Batch Normalization
  - ReLU Activation
- **Dense Layer 3:**
  - Units: $2 \times$ num_features
  - Batch Normalization
  - ReLU Activation

Output Layer

- **Output Dense Layer:**
  - Units: num_classes
  - Activation: Softmax

Model Compilation

- **Loss Function:**
  - Categorical Crossentropy
- **Optimizer:**
  - Adam optimizer with specified learning rate, beta parameters, and epsilon.

Live Emotion Detection

Face and Emotion Models:

Haar Cascade Classifier: It uses a Haar Cascade Classifier to detect faces in real-time from the webcam feed.

Emotion Recognition Model: Loads a pre-trained Keras model for emotion recognition. The model predicts emotions such as 'angry,' 'disgusted,' 'fearful,' 'happy,' 'neutral,' 'sad,' and 'surprised.'

Video Capture:

OpenCV VideoCapture: Utilizes the OpenCV `VideoCapture` object to capture video frames from the default camera (camera index 0).

Real-Time Processing:

Inside a continuous loop, the code reads frames from the video capture.

Grayscale Conversion: Converts each frame to grayscale for face detection using the Haar Cascade Classifier.

Face Detection and Emotion Prediction:

Face Detection: Detects faces in the grayscale frame and draws rectangles around them.

Region of Interest (ROI): Extracts the region of interest (ROI) for each detected face, resizes it, and preprocesses it for the emotion recognition model.

Emotion Prediction: The loaded deep learning model predicts the emotion based on the preprocessed face image.

Displaying Results:

If a face is detected, the code overlays the predicted emotion label on the frame. If no face is found, it displays a "No Face Found!" message on the frame.

User Interaction:

The real-time video feed and emotion predictions are displayed in a window.
Exit Interaction: The user can exit the application by pressing the 'q' key.

Chapter 4

Analysis of Proposed Model Performance and Comparison

4.1 Results of the Models

4.1.1 Using 2 Dense and 2 Modules of CNN layers

Visualizing Training Performance

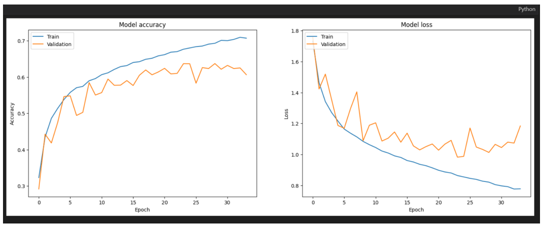

FIGURE 4.1: Visualizing Training performance

Evaluate Test Performance CNN Model Accuracy on test set: 0.6400

Confusion Matrix

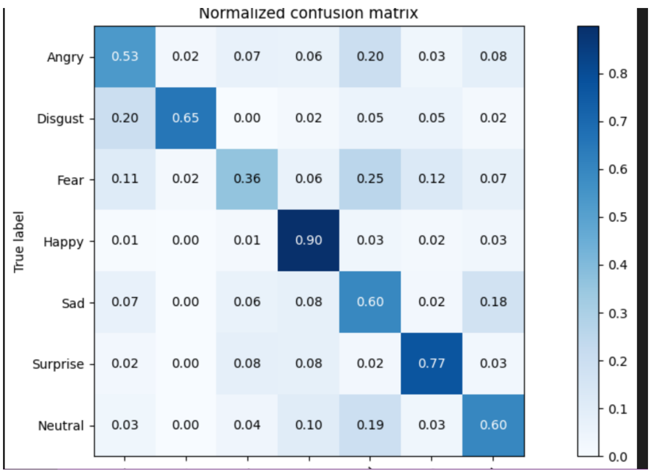

FIGURE 4.2: Confusion Matrix

4.1.2 Using 3 Dense and 3 Modules of CNN layers

Visualizing Training Performance

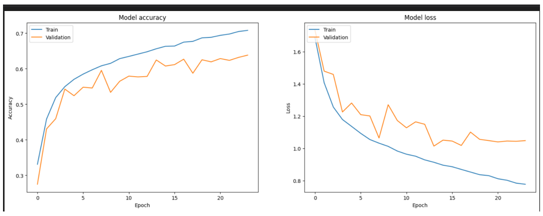

FIGURE 4.3: Visualizing Training performance

Evaluate Test Performance

- CNN Model Accuracy on test set: 0.6484
- Precision: 0.6466
- Recall: 0.6484
- F1 Score: 0.6409

Confusion Matrix

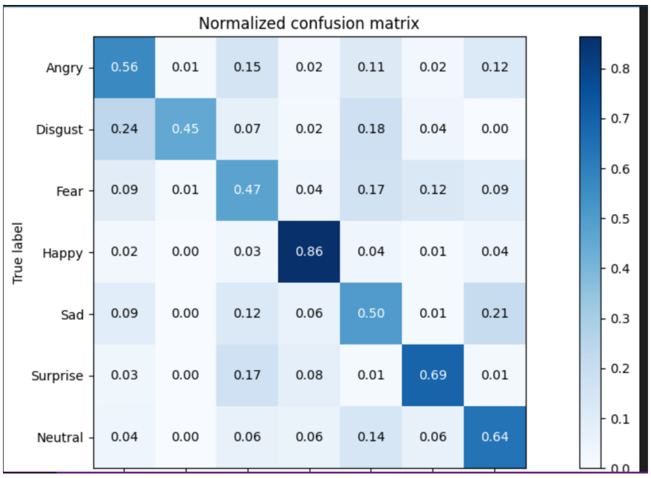

FIGURE 4.4: Confusion Matrix

4.1.3 Comparative Analysis: CNN Models with Different Convolution Layers

Using 3 Convolution Layers

The CNN model with 3 convolution layers achieved an accuracy of 64.00% on the test set.

Using 2 Convolution Layers

The CNN model with 2 convolution layers demonstrated an accuracy of 64.84% on the test set. Additionally, the model's performance was further evaluated using precision, recall, and F1 score:

- **Precision:** 64.66%
- **Recall:** 64.84%
- **F1 Score:** 64.09%

4.1.4 Analysis and Comparison

Both models show competitive performance, with the model using 2 convolution layers slightly outperforming the one with 3 convolution layers in terms of accuracy (64.84% vs. 64.00%). The additional evaluation metrics for the model with 2 convolution layers provide a more detailed understanding of its performance.

The precision of 64.66% indicates the ratio of correctly predicted positive observations to the total predicted positives. The recall of 64.84% represents the ratio of correctly predicted positive observations to the actual positives. The F1 score, which considers both precision and recall, is 64.09%.

In summary, while the model with 2 convolution layers has a slightly higher accuracy, the choice between the two models may depend on the specific goals and requirements of the task. The inclusion of precision, recall, and F1 score adds valuable insights into the model's performance beyond accuracy alone.

4.1.5 Result

The primary output is the classification of the emotional state being expressed by the person in the image or video frame

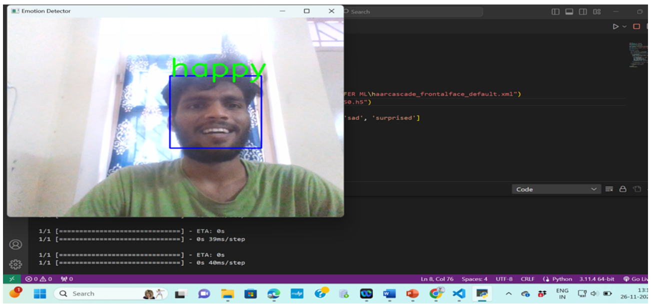

FIGURE 4.5: Live emotion Prdiction

Chapter 5

Conclusion and Future Work

5.1 Conclusion

The implemented CNN model demonstrated promising results, achieving an accuracy of 64.84% on the test set. Precision, recall, and F1 score metrics further indicated the effectiveness of the model in classifying emotions. The project's significance lies in its potential applications, including human-machine interaction, understanding human behavior, and aiding in the diagnosis of mental disorders.

The exploration of various deep learning techniques and the incorporation of a well-prepared dataset contributed to the project's success. The comprehensive evaluation, including the analysis using confusion matrices, provided valuable insights into the model's performance.deep-learning

5.2 Feature Work

To further fine tuning model using gridsearch, specifically:

a. Different optimizer such as RMSprop, Adagrad.

b. experimenting dropout with batch-normalization.

c. experimenting different dropout rat

To collect more data and train the model with balance dataset.

Emotion recognition is significant for machine learning and artificial intelligence. The future innovation in emotion recognition will allow machines to understand how people feel, which is the first step for them to fulfil our needs........by visionify .

(Wafa Mellouka*, 2020; S L Happy, 2012)

Bibliography

S L Happy Anjith George, Aurobinda Routray (2012). "A Real Time Facial Expression Classification System Using Local Binary Patterns". In: *IEEE*. URL: https://www.researchgate.net/publication/261525128_A_Real_Time_Facial_Expression_Classification_System_Using_Local_Binary_Patterns.

Wafa Mellouka*, Wahida Handouzia (2020). "Facial emotion recognition using deep learning: review and insights". In: *The 2nd interernational Workshop on the Future of Internet of Everything (FIoE)*. URL: https://www.sciencedirect.com/science/article/pii/S1877050920318019.

1. Facial emotion recognition using deep learning: review and insights The 2nd intererational Workshop on the Future of Internet of Everything (FIoE), (August 2020)
2. Laith S., Tahir F. S., Abdulrahman A. A. Effectiveness of new algorithms for facial recognition based on deep neural networks. International Journal of Nonlinear Analysis and Applications . 2022;13(1):2171–2178.
3. V. Ramya, Shaik Reshma, Afrah Samreen, and U. Chandrasekhar, "Facial Emotion Recognition Using ML Algorithms," in Proceedings of the IEEE International Conference on Sustainable Expert Systems, 2022, pp. 389-402. DOI: 10.1007/978-981-16-7657-4_32. ISBN: 978-981-16-7656-7.
4. T. U. Ahmed, S. Hossain, M. S. Hossain, R. ul Islam and K. Andersson, "Facial Expression Recognition using Convolutional Neural Network with Data Augmentation," 2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR), Spokane, WA, USA, 2019, pp. 336-341, doi: 10.1109/ICIEV.2019.8858529.
5. Convolutional Neural Networks, Explained Link
6. Pandas Library https://pandas.pydata.org/pandas-docs/stable/
7. OpenCV Tutorial: A Guide to Learn OpenCV https://pyimagesearch.com/2018/07/19/opencv-tutorial-a-guide-to-learn-opencv/
8. D. Lekhak, "Facial Expression Recognition System using Convolutional Neural Network," December 2017. [Online]. Available: Link
9. André Teixeira Lopes, Edilson de Aguiar, Alberto F. De Souza, Thiago Oliveira-Santos, "Facial Expression Recognition with Convolutional Neural Networks: Coping with Few Data and the Training Sample Order," January 2017.Link
10. Z. Zeng, M. Pantic, G. Roisman, and T. Huang, "A survey of affect recognition methods: audio, visual and spontaneous expressions," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 1, pp. 39-58, 2009.
11. Christou, N., Kanojiya, N. (2019). Human Facial Expression Recognition with Convolution Neural Networks. In: Yang, XS., Sherratt, S., Dey, N., Joshi, A. (eds) Third International Congress on Information and Communication Technology. Advances in Intelligent Systems and Computing, vol 797. Springer, Singapore.https://doi.org/10.1007/978-981-13-1165-9$_4$9