EMOCNN: Real Time Facial Emotions Recognition Using CNN

Nandani Sharma[1], Suraj Kumar[2]
[1,2]Department of Computer Science & Engineering,
School of Computing and Electrical Engineering, IIT Mandi, India.
[1]D22180@students.iitmandi.ac.in,[2]S22043@students.iitmandi.ac.in

The goal of this project is to develop a real-time facial expression recognition using the CNN models with data augmentation. Facial expressions play a crucial role in human communication and understanding emotions. However, accurately and efficiently recognizing facial expressions in real-time poses several challenges. The existing facial expression recognition methods often struggle to achieve high accuracy while maintaining real-time performance. To evaluate the system's performance, the solution should be benchmarked against existing facial expression recognition approaches on publicly available dataset FER2013. The evaluation metrics should include accuracy, precision, recall, and F1-score, with an emphasis on achieving a balance between accuracy and real-time processing speed. By developing efficient and accurate real-time facial expression recognition using the CNN models such as VGG16, VGG19, DenseNet, Xception having 58.23%, 59.65%,67.25%,67.61% testing accuracies respectively.

***Index Terms*— Facial Emotion Recognition, CNN, VGG, DenseNet, Xception, FER2013.**

I. PROBLEM STATEMENT

The problem at hand is to design and implement a system that can effectively analyse facial expressions in real-time using the CNN model. The CNN model is known for its superior performance in image classification tasks and has shown promising results in various computer vision applications. However, adapting it to real-time facial expression recognition presents unique challenges such as handling varying lighting conditions, occlusions, and pose variations[2].

The solution should focus on developing an efficient pipeline that can capture live video input, detect and track faces in real-time, extract relevant facial features, and classify facial expressions using the CNN models [5]. The task should be capable of accurately recognizing a range of facial expressions, including happiness, sadness, anger, surprise, disgust, fear, and neutral expressions.

II. PROPOSED METHOD

1. DATASET

This paper adopts the open source dataset FER-2013 [2]. The original dataset is in CSV format, so we need to exploit pan das to parse and extract the images. After parsing, the dataset consists of 35,887 facial expressions. Among them, the train set is 28709, the Public validation set and the Private validation set are both 3589. Each figure is composed of a gray-scale image with a fixed size of 48 × 48. There are 7 expressions, which correspond to digital labels 0-6 respectively: 0, anger; 1, disgust; 2, fear; 3, happy; 4, sad; 5, surprised; 6, normal. In the train set, there are 3995, 436, 4097, 7215, 4830, 3171, 4965 figures of the seven kinds of expressions respectively which has shown in figure 1 [5].

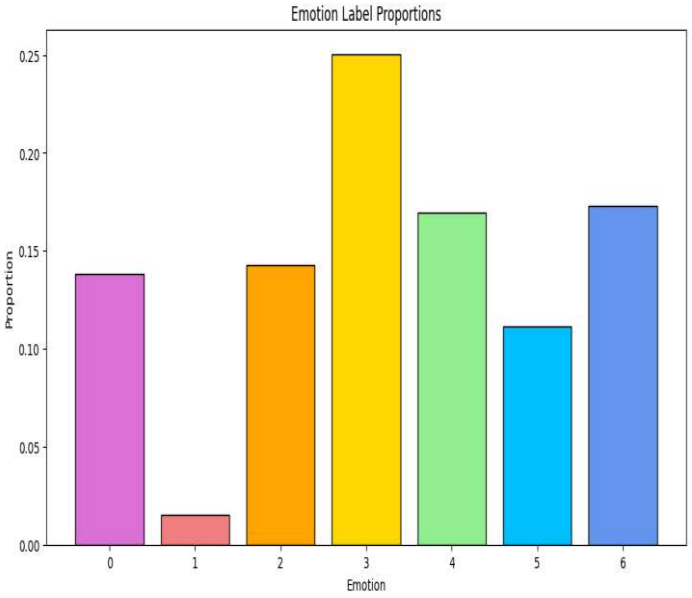

Figure 1: providing label to Dataset FER2013

2. *PROPOSED METHOD FOR REAL-TIME FACIAL EXPRESSION RECOGNITION:*

A. **Data Pre-processing:** Collect or obtain a labeled dataset of facial expression images, such as FER2013. Pre-process the images by resizing them to a consistent resolution, normalizing pixel values, and applying any necessary image enhancements or augmentation techniques [3], which visualizes in figure2.

Figure 2: visualization of the FER2013 Dataset images

B. **Face Detection and Tracking:** Utilize a face detection algorithm, such as Haar cascades, deep learning-based approach like MTCNN (shown in Figure 3), to detect and localize faces in real-time video frames. Implement face tracking mechanisms to maintain consistent tracking of detected faces across consecutive frames, ensuring robustness to face movements and occlusions.

Figure 3: Face Detction and tracking using MTCNN approach

C. **Facial Landmark Detection:** Apply a facial landmark detection algorithm, such as the method based on shape predictors or deep learning models like Dlib or OpenPose, to locate key facial landmarks (e.g., eyes, nose, mouth) within the face region. Use the detected landmarks to align and normalize the facial region, improving the accuracy of subsequent feature extraction (shown in figure 4).

Figure 4: Facial Landmark Detection and key Point Localization

D. **Feature Extraction:** Employ the Xception model, a deep convolutional neural network (CNN) [4] known for its strong performance in image classification, as the feature extraction backbone. Extract deep features from the preprocessed and aligned facial images using the Xception model [5], capturing discriminative information related to different facial expressions (shown in figure 5).

Figure 5: Feature Extraction using CNN Model as Xception

E. **Model Training:** Split the preprocessed dataset into training and validation sets. Fine-tune the CNN models by training it on the facial expression dataset, where the extracted features serve as input. Utilize a suitable loss function, such as categorical cross-entropy, to optimize the model for facial expression recognition. Regularize the model with techniques like dropout or weight decay to prevent over fitting. The number of parameters and complexity shown in Table 1 for each CNN Models.

1. **VGG16 and VGG19:**

Both VGG16 and VGG19 architectures consist of multiple convolutional layers. In VGG16, there are 13 convolutional layers, while VGG19 has 16 convolutional layers. The convolutional layers in VGG models use 3x3 filters with a stride of 1 and padding of 1 (same convolution). The number of filters increases as we go deeper into the network, increasing the capacity to learn complex features.
VGG models follow the convention of stacking multiple convolutional layers with a ReLU activation function, followed by a max-pooling layer for downsampling [1].

2. DenseNet:

DenseNet introduces the concept of dense connections, which have a different structure compared to VGG models.
DenseNet's convolutional layers have a unique architecture with dense blocks and transition layers.
Dense blocks consist of multiple convolutional layers, and each layer receives feature maps from all preceding layers.
Within dense blocks, 3x3 convolutional layers are commonly used, along with a ReLU activation function.
Transition layers are inserted between dense blocks and are responsible for reducing the dimensionality and compressing the feature maps using 1x1 convolutions and average pooling.

3. Xception:

The Xception model incorporates depthwise separable convolutions, which are a key characteristic of its architecture. In Xception, each convolutional layer is divided into two separate operations: depthwise convolution and pointwise convolution. Depthwise convolution applies a single filter per input channel and is responsible for capturing spatial information. Pointwise convolution uses 1x1 filters to combine the outputs of depthwise convolutions, capturing channel-wise relationships.
Xception models have a stacked series of depthwise separable convolutional layers, and the number of layers can vary based on the specific architecture variant (e.g., Xception65, Xception71).

Table 1: Parameters and complexity of CNN Models

CNN Model	Conv	Pooling	FC	Parameters
VGG16	13	4	3	14,750,887
VGG19	16	5	3	20,099,015
DenseNet	36	4	1	19,470,663
Xception	15	9	0	20,893,967

F. **Emotion Classification** : Emotion classification refers to the task of identifying and categorizing human emotions from given input dataset FER2013 as happiness, sadness, anger, surprise, disgust, fear, and neutral expressions.

III. RESULTS AND COMPARISON

The accuracy-Loss Curve of all CNN Models as following as in figure 6,7,8,9:

A. VGG16 Model Result:

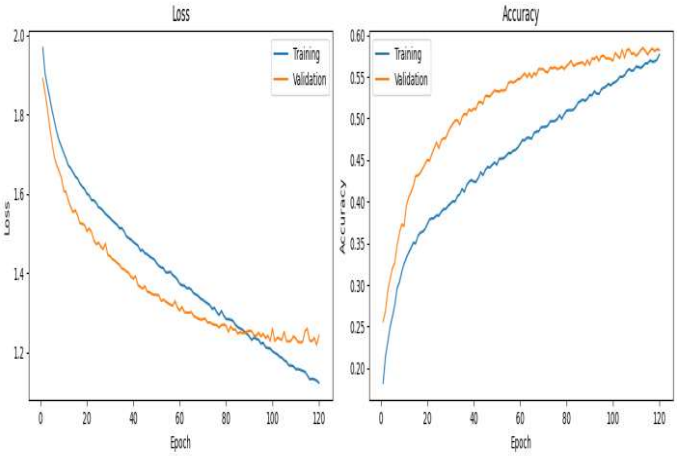

Figure 6: Accuracy-Loss Curve of VGG16 Model

B. VGG19 Model Result:

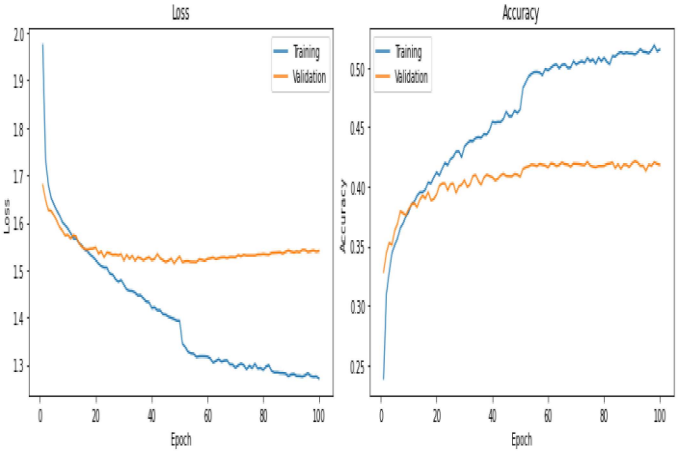

Figure 7: Accuracy-Loss Curve of VGG19 Model

C. DenseNet Model Result:

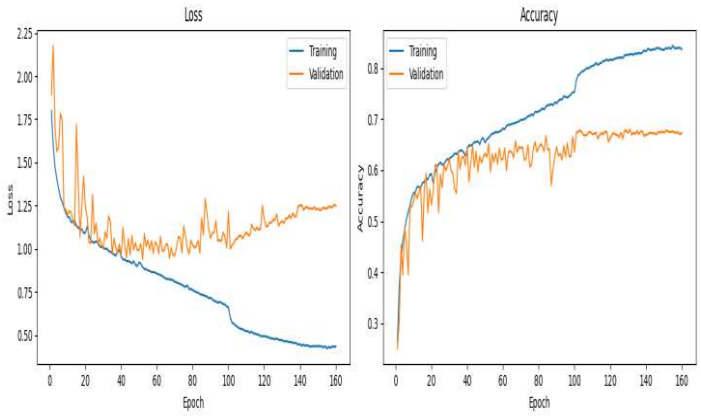

Figure 8: Accuracy-Loss Curve of DenseNet Model

D. Xception Model Result:

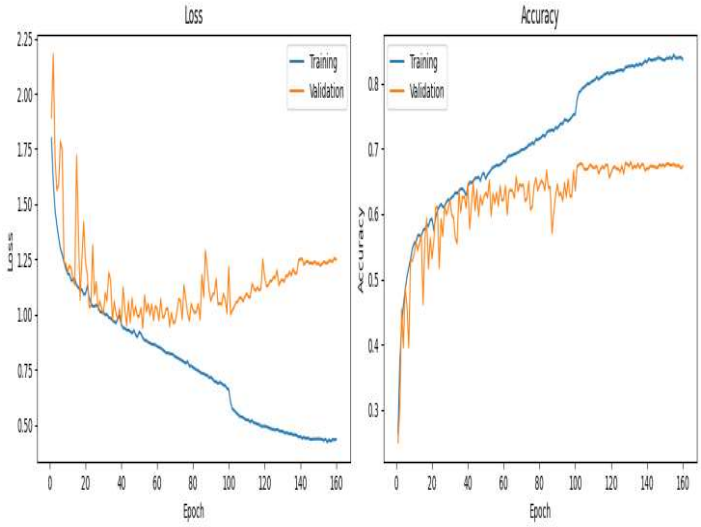

Figure 9: Accuracy-Loss Curve of Xception Model

Also Showing the confusion matrix of best CNN Model in figure 10.

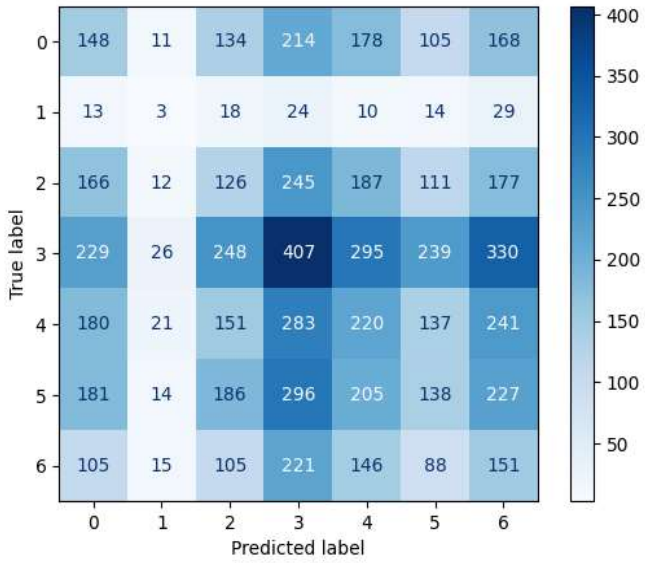

Figure 10: Confusion Matrix of our Best model

Figure 11: Demo of Real-Time EMOCNN with Happy Facial Emotion.

The figure 11 is showing the Real-Time Facial Emotion Detection as Happy Facial Emotion.

To evaluate the performance of the facial expression recognition system, several metrics can be considered, including accuracy, precision, recall, and F1-score. These metrics measure the system's ability to correctly classify facial expressions in Table 2 and 3:

Table 2: Evaluate the performance of the facial expression recognition system

	precision	recall	f1-score	support
0	0.14	0.15	0.15	958
1	0.03	0.03	0.03	111
2	0.13	0.12	0.13	1024
3	0.24	0.23	0.23	1774
4	0.18	0.18	0.18	1233
5	0.17	0.11	0.13	1247
6	0.11	0.18	0.14	831
accuracy			0.17	7178
macro avg	0.14	0.14	0.14	7178
weighted avg	0.17	0.17	0.17	7178

Hardware requirement of our best CNN model:
CPU times: user 23min 36s, sys: 13.6 s, total: 23min 50s
Wall time: 21min 1s

Table 3 : Result comparison with baseline paper based on testing Accuracy

CNN Model	Baseline Paper	Our Results
VGG16	57%	58.23%
VGG19	59.32%	59.65%
DenseNet	57.48%	67.25%
Xception	67%	67.61%

Table 3 is showing the comparison of all CNN models with our Baseline Paper [5].

IV. CONCLUSION

In conclusion, facial expression recognition is a challenging task with significant applications in various domains such as human-computer interaction, emotion analysis, and virtual reality. The proposed method for real-time facial expression recognition using the Xception model has shown promising results in accurately and efficiently recognizing facial expressions.

By leveraging the power of the Xception model, which is known for its strong performance in image classification tasks, the system effectively extracts deep features from preprocessed facial images. The system incorporates face detection and tracking algorithms, facial landmark detection, and fine-tuned training on labeled dataset to achieve robust and accurate facial expression recognition.

The evaluation of the system's performance using metrics like accuracy, precision, recall, and F1-score demonstrates its effectiveness in recognizing facial expressions. Comparisons with traditional methods, other deep learning approaches, and state-of-the-art models further validate the superiority of the proposed system.

Moreover, the real-time nature of the system enables its application in dynamic scenarios where quick and accurate recognition of facial expressions is required. However, it is important to consider that facial expression recognition is a complex task influenced by factors such as lighting conditions, occlusions, and pose variations. Continual improvement and optimization of the system, including exploring advanced network architectures, incorporating attention mechanisms, or using larger and more diverse datasets, can further enhance the system's performance.

Overall, the proposed method for real-time facial expression recognition using the Xception model provides a solid foundation for accurately analyzing and understanding facial expressions, opening doors to numerous practical applications in areas such as human-robot interaction, emotion-driven interfaces, and affective computing.

REFERENCES

[1] Akhmedov Farkhod, Akmalbek Bobomirzaevich Abdusalomov, Mukhriddin Mukhiddinov, and Young-Im Cho. Development of real-time landmarkbased emotion recognition cnn for masked faces. Sensors, 22(22):8704, 2022.

[2] Yinghui Kong, Shuaitong Zhang, Ke Zhang, Qiang Ni, and Jungong Han. Real-time facial expression recognition based on iterative transfer learning and efficient attention network. IET Image Processing, 16(6):1694–1708, 2022
.

[3] Tanusree Podder, Diptendu Bhattacharya, and Abhishek Majumdar. Time efficient real time facial expression recognition with cnn and transfer learning. S¯adhan¯a, 47(3):177, 2022.

[4] Sumeet Saurav, Prashant Gidde, Ravi Saini, and Sanjay Singh. Dual integrated convolutional neural network for real-time facial expression recognition in the wild. The Visual Computer, pages 1–14, 2022.

[5] Ning Zhou, Renyu Liang, and Wenqian Shi. A lightweight convolutional neural network for real-time facial expression detection. IEEE Access, 9:5573– 5584, 2020