International Journal of Advances in Engineering and Management (IJAEM)
Volume 7, Issue 06 June 2025, pp: 137-141 www.ijaem.net ISSN: 2395-5252

Real-Time Facial Expression Detection System

[1]Avani Fatesinhbhai Chaudhari, [2]Priyanka Sharma
Bhagwan Mahavir College of Engineering and Technology, Surat, Gujrat

---

Date of Submission: 28-05-2025 Date of Acceptance: 08-06-2025

ABSTRACT: Real-time Facial Expression Detection system using Artificial Intelligence and Deep Learning. The project combined theoretical concepts with practical implementation, focusing on deep learning. A Convolutional Neural Network (CNN) was trained on the FER2013 dataset to classify seven emotions: Happy, Sad, Angry, Disgusted, Fearful, Surprised, and Neutral. Both a custom CNN and VGG16 (via transfer learning) were used to compare performance and training efficiency.[9] Technologies like Python, TensorFlow, Keras, and OpenCV were used. Key features included real-time webcam integration, face detection with Haar Cascades, and emotion prediction with visual overlays. The model training involved data augmentation and validation accuracy evaluation. The internship helped build key skills in model design, image pre-processing, real-time integration, and performance analysis, offering valuable insights into deploying AI in real-world applications.

KEYWORDS: CNN, OPENCV, FER2013, Dataset, Image processing.

I. INTRODUCTION

This project goes beyond basic image classification by implementing a real-time facial expression detection system using deep learning. Leveraging CNN models trained on the FER2013 dataset, along with OpenCV for face detection and live webcam input, the system efficiently identifies and classifies human emotions.[9] It demonstrates not only technical proficiency in AI/ML but also the ability to build responsive and intelligent solutions applicable in real-world environments.[1]

Objectives:

- To develop a deep learning model capable of accurately classifying facial expressions using the FER2013 dataset.
- To integrate computer vision techniques (OpenCV) for real-time face detection from live webcam input[2].
- To design a responsive and user-friendly interface for real-time emotion monitoring.
- To evaluate the performance of the CNN model using metrics such as accuracy, precision, recall, and F1-score on validation data.
- To ensure real-time processing capabilities by optimizing the system for low-latency performance.
- To demonstrate the practical applications of emotion detection, such as in human-computer interaction, mental health monitoring, or customer experience analysis.
- To explore and implement data preprocessing and augmentation techniques to improve model generalization and robustness.
- To validate the system in different lighting and background conditions, ensuring reliability in varied real-world environments.
- To document the development process and provide reproducible code for further research and improvement.
- To investigate ethical considerations and privacy implications associated with real-time facial analysis technologies.

Motivation:

Understanding human emotions is a fundamental aspect of effective human-computer interaction. The motivation behind this project stems from the growing demand for systems that can interpret and respond to human emotions in real time. Whether in mental health monitoring, customer service, or interactive entertainment, the ability to detect facial expressions enhances the responsiveness and empathy of AI systems. By leveraging deep learning and computer vision, this project aims to bridge the gap between machine perception and human emotional intelligence. The integration of CNNs with real-time video processing highlights a practical and scalable approach to emotion recognition, moving from theoretical models to applications with tangible impact.[3]

1.1 SYSTEM FUNCTIONALITY

Deep learning is a key component in Machine Learning and Artificial Intelligence, excelling in tasks such as image recognition, object detection, and emotion classification. For our project, which focuses on facial expression detection, the goal is to identify and classify emotions from facial expressions in real-time. Unlike the use of emojis in communication, our system displays the textual description of the detected emotion.[4]

The system employs Convolutional Neural Networks (CNNs) to perform feature extraction[3]. The architecture consists of convolutional layers, ReLU layers, and max-pooling layers for efficient learningand recognition. Data pre-processing steps include data augmentation and morphological operations to enhance the model's performance and accuracy.

In the Fig 3.1.1 facial expressions are defined as the configuration of facial muscles used to convey specific emotional states to an observer. Emotions can be classified into six broad categories: Anger, Disgust, Fear, Happy, Sad, Surprise, and Neutral. The goal is to train a model that can accurately differentiate between these emotions using a Convolutional Neural Network (CNN) with the FER2013 dataset.[1][5]

[1]The model design begins with initializing the CNN architecture, where the input can be either a static or dynamic image. This model consists of several layers, including convolution layers for feature extraction, pooling layers for dimensionality reduction, flatten layers to convert data into a 1D array, and dense layers for classification. The convolution layers are crucial for achieving higher accuracy, especially with large datasets. The dataset, provided in CSV format with pixel values, is preprocessed by converting it into images. These images are then used to classify emotions based on the corresponding facial expressions. Additionally, hyperparameters are fine-tuned to optimize model performance and accuracy.

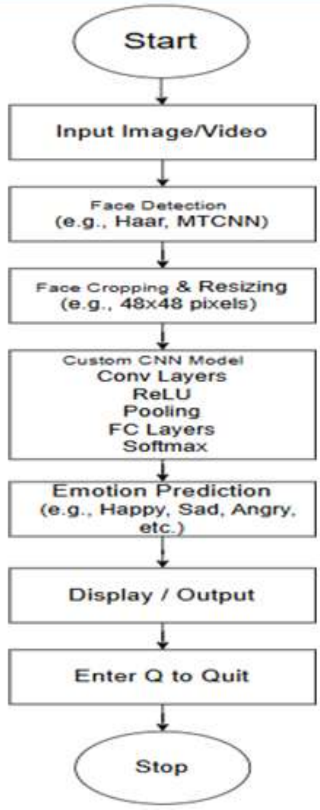


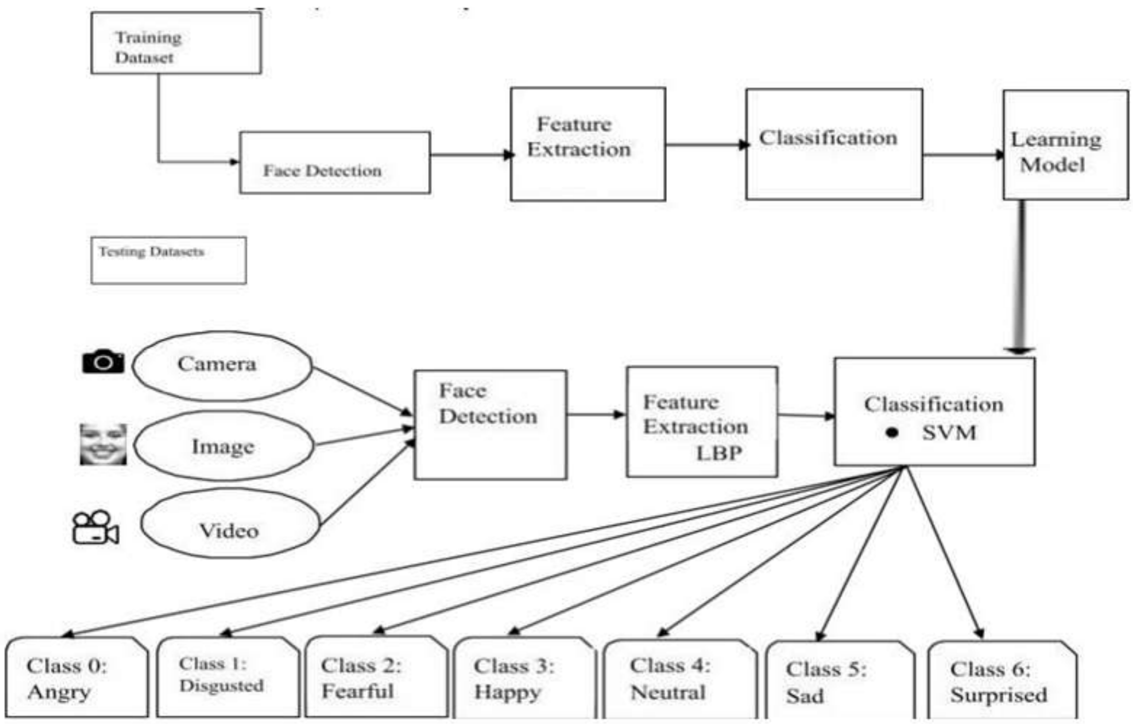
Fig 1.1.1 Block diagram of System

II. EXPERIMENTATION

The training phase, the system receives grayscale images of faces along with their corresponding emotion labels. It learns a set of weights for the CNN model.[5] The input to the system consists of facial images, and intensity normalization is applied to these images. The normalized images are then used to train the Convolutional Neural Network. To avoid performance bias due to the order in which the examples are presented, a validation dataset is used to select the best set of weights after performing training with samples in different orders. The output of the training phase is a set of weights that yield the best performance on the training data.

During the testing phase, the system receives a grayscale image of a face from the test dataset and outputs the predicted emotion using the network weights learned during training. The output is a single label corresponding to one of the seven basic emotions: Happy, Sad, Angry, Surprise, Neutral, Disgust, or Fear.

International Journal of Advances in Engineering and Management (IJAEM)
Volume 7, Issue 06 June 2025, pp: 137-141 www.ijaem.net ISSN: 2395-5252



SYSTEM ARCHITECTURE

SOURCE CODE FOR MODEL TRAINING:

```
train_dir=
"C:\\Users\\avani\\Downloads\\archive\\train"
val_dir=
"C:\\Users\\avani\\Downloads\\archive\\test"
train_datagen=
ImageDataGenerator(rescale=1./255)
val_datagen = ImageDataGenerator(rescale=1./255)

train_generator=
train_datagen.flow_from_directory(
train_dir,
target_size=(48,48),
batch_size=64,
color_mode="grayscale",
class_mode='categorical')

validation_generator=
val_datagen.flow_from_directory(
val_dir,
target_size=(48,48),
batch_size=64,
color_mode="grayscale",
class_mode='categorical')

emotion_model = Sequential()

emotion_model.add(Conv2D(32, kernel_size=(3, 3),
activation='relu', input_shape=(48,48,1)))
emotion_model.add(Conv2D(64, kernel_size=(3, 3),
activation='relu'))
emotion_model.add(MaxPooling2D(pool_size=(2,
2)))
emotion_model.add(Dropout(0.25))

emotion_model.add(Conv2D(128, kernel_size=(3,
3), activation='relu'))
emotion_model.add(MaxPooling2D(pool_size=(2,
2)))
emotion_model.add(Conv2D(128, kernel_size=(3,
3), activation='relu'))
emotion_model.add(MaxPooling2D(pool_size=(2,
2)))
emotion_model.add(Dropout(0.25))

emotion_model.add(Flatten())
emotion_model.add(Dense(1024, activation='relu'))
emotion_model.add(Dropout(0.5))
emotion_model.add(Dense(7, activation='softmax'))

cv2.ocl.setUseOpenCL(False)

emotion_dict = {0: "Angry ", 1: "Disgusted", 2:
"Fearful", 3: "Happy ", 4: "Neutral", 5: "Sad", 6:
"Surprised"}

emotion_model.compile(loss='categorical_crossentr
opy', optimizer=Adam(learning_rate=0.0001),
metrics=['accuracy'])
emotion_model_info = emotion_model.fit(
train_generator,
```

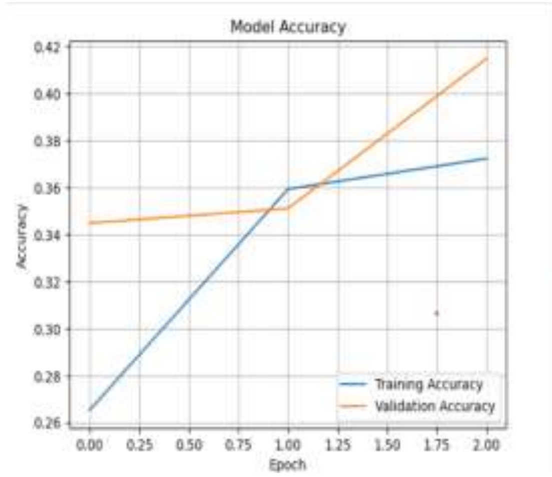
International Journal of Advances in Engineering and Management (IJAEM)
Volume 7, Issue 06 June 2025, pp: 137-141 www.ijaem.net ISSN: 2395-5252

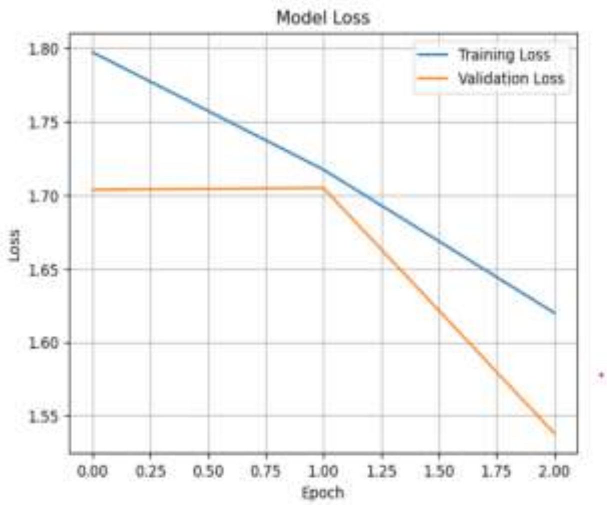
```
steps_per_epoch=28709 // 64,
      epochs=3,
validation_data=validation_generator,
validation_steps=7178 // 64)
emotion_model.save_weights('emotion_model.weights.h5')
```

III. MODEL TESTING AND EVALUTION

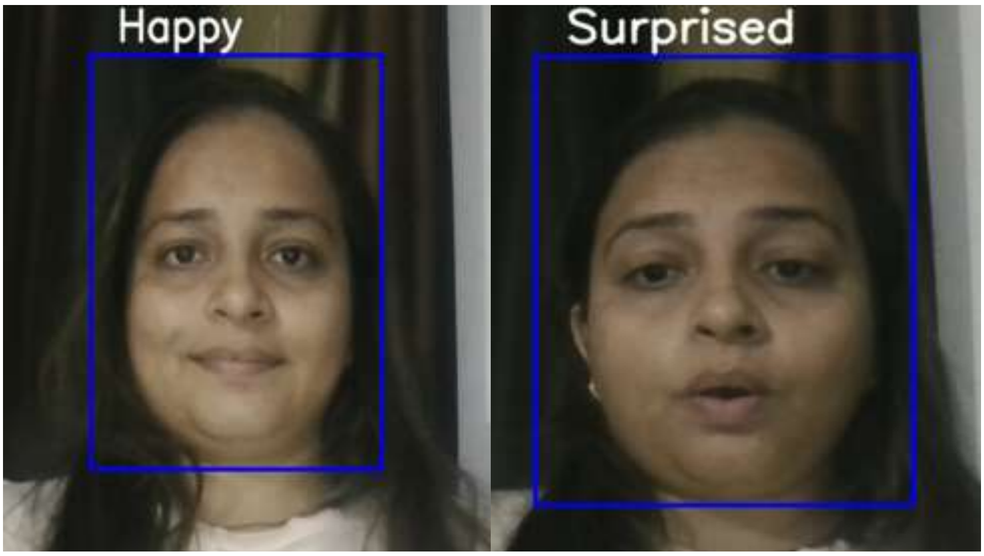
After training the model for 50 epochs, the testing results showed that the CNN model was able to learn and improve over time. In the last 10 epochs (41 to 50), the training accuracy ranged from 65% to 75%, and the validation accuracy stayed between 59% to 60%.[3][5]

These results show that the model performs well, but the validation accuracy can be improved further by tuning the model or using more data.





TESTING RESULT:



IV. CONCLUSION

The Real-time Facial Expression Detection system successfully demonstrated the integration of deep learning with computer vision to recognize human emotions through facial expressions. By leveraging both a custom-built CNN and the pre-trained VGG16 model via transfer learning, the project enabled a comparative analysis of model performance and efficiency. The use of the FER2013 dataset, along with data augmentation techniques, improved the model's ability to generalize across diverse facial expressions.The system's real-time capabilities—achieved through webcam integration, face detection via Haar Cascades, and visual emotion overlays—highlight its practical applicability in areas like human-computer interaction, surveillance, and user experience enhancement.

Technologies such as Python, TensorFlow, Keras, and OpenCV played a crucial role in building and deploying this solution.

This project not only deepened understanding of core concepts like CNN architecture, image preprocessing, and transfer learning but also provided hands-on experience in integrating AI models into real-world systems. Overall, the internship experience.

REFERENCES

[1]. Majeed, Aseel Wadood, Shaimaa Hameed Shaker, and Ali Adel Saeid. "Understanding the techniques and methods for real-time face recognition and tracking: A comprehensive survey." AIP Conference Proceedings. Vol. 3169. No. 1. AIP Publishing, 2025.

[2]. Aishwarya, M., Neelima, N. (2021). The Analogy of Haar Cascade and HOG Approaches for Facial Emotion Recognition. In: Kaiser, M.S., Xie, J., Rathore, V.S. (eds) Information and Communication Technology for Competitive Strategies (ICTCS 2020). Lecture Notes in Networks and Systems, vol 190. Springer, Singapore. https://doi.org/10.1007/978-981-16-0882-7_62Gould, L; Richeson, W; and Erickson, F., 1991, “Performance Evaluation of a Camless Engine Using Valve Actuation with Programmable Timing,” SAE Paper No. 910450.

[3]. Farfade, Sachin Sudhakar, Mohammad J. Saberian, and Li-Jia Li. "Multi-view face detection using deep convolutional neural networks." Proceedings of the 5th ACM on International Conference on Multimedia Retrieval. 2015.

[4]. Sharma, Manik, et al. "Facial detection using deep learning." IOP Conference Series: Materials Science and Engineering. Vol. 263. No. 4. IOP Publishing, 2017.

[5]. Hu, Guosheng, et al. "When face recognition meets with deep learning: an evaluation of convolutional neural networks for face recognition." Proceedings of the IEEE international conference on computer vision workshops. 2015.

[6]. Arshed, Muhammad Asad, et al. "Multiclass ai-generated deepfake face detection using patch-wise deep learning model." Computers 13.1 (2024): 31.

[7]. Rostami, Mohsen, AmirhamzehFarajollahi, and Hashem Parvin. "Deep learning-based face detection and recognition on drones." Journal of Ambient Intelligence and Humanized Computing 15.1 (2024): 373-387.

[8]. Kashyap, Gautam Siddharth, et al. "Detection of a facemask in realtime using deep learning methods: Prevention of COVID-19." Research Advances in Network Technologies. CRC Press, 2024. 224-240.

[9]. Haq, H. B. U., et al. "Enhanced real-time facial expression recognition using deep learning." Acadlore Transactions on AI and Machine Learning 3.1 (2024): 24-35.

[10]. Pordoy, Jamie, et al. "Multi-Frame Transfer Learning Framework for Facial Emotion Recognition in e-Learning Contexts." IEEE Access (2024).

[11]. Win, Khin Cho, Zahid Akhtar, and C. Krishna Mohan. "Does Hard-Negative Contrastive Learning Improve Facial Emotion Recognition?." Proceedings of the 2024 7th International Conference on Machine Vision and Applications. 2024.

[12]. Tomar, Pragya Singh, Kirti Mathur, and Ugrasen Suman. "Fusing facial and speech cues for enhanced multimodal emotion recognition." International Journal of Information Technology 16.3 (2024): 1397-1405.

[13]. Kopalidis, Thomas, et al. "Advances in facial expression recognition: A survey of methods, benchmarks, models, and datasets." Information 15.3 (2024): 135.