

**A CONVOLUTIONAL NEURAL NETWORK BASED FRAMEWORK FOR FACIAL EMOTION RECOGNITION**

**A MINI PROJECT REPORT**

***Submitted by***

**BALAJI P (9517202109010)**

**JAYARAM N (9517202109022)**

**THILLAI PRABAKAR T (9517202109054)**

***in partial fulfillment for the award of the degree***

***of***

**BACHELOR OF TECHNOLOGY**

**IN**

**ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**

**MEPCO SCHLENK ENGINEERING COLLEGE, SIVAKASI**

**ANNA UNIVERSITY : CHENNAI 600 025**

**OCTOBER 2023**

**MEPCO SCHLENK ENGINEERING COLLEGE, SIVAKASI**

**AUTONOMOUS**

**DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE**

## https://lh5.googleusercontent.com/zNka7zD4qVeBnvWlplpLRs0AhIfyJjynl0Y2sPhYAkYB_kgqY56FgxjF1WVSLdzKUNV1mSRE1xG8FJnbkZZ529aoZw5spPvW0R-MyUTvZzc5MNA6XyrD5FI-pmidFP5Fo-JuhJaom5SYt9aa2d89-bhUVc1HPBPTS704MiLgcRAmaFCTlQGTRmeON30JVQC6VpjQ25gmNw

## BONAFIDE CERTIFICATE

Certified that this project report **“A CONVOLUTIONAL NEURAL NETWORK BASED FRAMEWORK FOR FACIAL EMOTION RECOGNITION”** is thebonafide work of **“BALAJI P(9517202109010) JAYARAM N (9517202109022) THILLAI PRABAKAR T (9517202109054)** whocarried out the mini project in 19AD552- Machine Learning Techniques Laboratory during the fifth semester July 2023 – October 2023 under my supervissssion.

**SIGNATURE SIGNATURE**

**Mrs.L.Prasika, M.E.,(Ph.D) Dr.J.Angela Jennifa Sujana, M.E.,Ph.D**

**Assistant Professor(SG) Professor & Head**

Artificial Intelligence and Data Science Artificial Intelligence and Data Science Mepco Schlenk Engineering College,Mepco Schlenk Engineering College Sivakasi- 626 005,Virudhunagar Sivakasi- 626 005,Virudhunagar

The project report submitted for the viva voce held on ………….

**INTERNAL EXAMINER EXTERNAL EXAMINER**

**TABLE OF CONTENTS**

|  |  |  |  |
| --- | --- | --- | --- |
| **CHAPTER NO** | **TITLE** | | **PAGE NO.** |
|  | ABSTRACT | | 4 |
| 1 | INTRODUCTION | |  |
|  | 1.1 | Overview of Project | 6 |
|  | 1.2 | Problem Statement | 7 |
|  | 1.3 | Modules Description | 8 |
|  | 1.4 | Models Description | 9 |
|  | 1.5 | Model Architecture | 9 |
| 2 | PROPOSED SYSTEM | |  |
|  | 2.1 | System Design | 10 |
| 3 | SYSTEM REQUIREMENTS | | 11 |
| 4 | IMPLEMENTATION | |  |
|  | 4.1 | Dataset | 12 |
|  | 4.2 | Source Code | 13 |
|  | 4.3 | Results | 17 |
| 5 | CONCLUSION | | 19 |
| 6 | BIBLIOGRAPHY | | 20 |

**ABSTRACT**

Facial emotion recognition is a key component in human-computer interaction, offering a non-intrusive means of understanding and responding to user emotions. This project presents a real-time facial emotion recognition system using deep learning techniques. The system employs a convolutional neural network (CNN) trained on a dataset of facial expressions to accurately classify emotions such as anger, disgust, fear, happiness, sadness, surprise, and neutrality.

The training process involves preprocessing facial images, utilizing grayscale images to enhance model efficiency. The trained model is then integrated into a real-time video processing pipeline, capturing and analyzing facial expressions from a live webcam feed. The system incorporates the OpenCV library for face detection and localization.

The application of this system extends to various domains, including human-computer interaction, virtual reality, and emotion-aware computing. By providing instantaneous feedback on user emotions, the system enhances user experience and enables adaptive system responses. The project contributes to the growing field of affective computing, emphasizing the practical implementation of deep learning models in real-world scenarios.

**ACKNOWLEDGEMENT**

First and foremost we **praise and thank “The Almighty”,** the lord of all creations, who by his abundant grace has sustained us and helped us to work on this project successfully.

We really find unique pleasure and immense gratitude in thanking our respected management members**,** who is the backbone of our college.

A deep bouquet of thanks to respected Principal **Dr.S.Arivazhagan M.E.,Ph.D.,** for having provided the facilities required for our mini project.

We sincerely thank our Head of the Department **Dr. J. Angela Jennifa Sujana M.E.,Ph.D.,** Professor & Head, Department of Artificial Intelligence and Data Science, for her guidance and support throughout the mini project .

We extremely thank our project coordinator **Mrs.L.Prasika M.E.,(Ph.D),** Assistant Professor,who inspired us and supported us throughout the mini project.

We extend our heartfelt thanks and profound gratitude to all the faculty members of Artificial Intelligence and Data Science department for their kind help during our mini project work.

We also thank our parents and our friends who had been providing us with constant support during the course of the mini project work.

**INTRODUCTION**

* 1. **OVERVIEW**

Facial recognition technology has become a prominent area of research and application in computer vision. Real-time facial emotion recognition adds a dynamic layer to this technology by providing insights into the emotional states of individuals as they express themselves.

The implemented system leverages deep learning, specifically a convolutional neural network (CNN), to analyze facial features and predict emotions in real-time. The CNN model has been trained on a dataset containing images of faces labeled with corresponding emotions. This training enables the model to learn patterns and features indicative of different emotional states.

The real-time aspect of the system is achieved through continuous video capture from a webcam. Each video frame is processed to detect faces using a Haar cascade classifier, a popular method in computer vision for object detection. Once a face is detected, the region of interest (ROI) is extracted and fed into the pre-trained CNN model for emotion prediction.

The emotional predictions are then overlaid onto the video frames, providing a live display of the recognized emotions associated with each detected face. This application finds use in various domains, including human-computer interaction, user experience research, and even in fields like market research for gauging consumer reactions.

**1.2 PROBLEM STATEMENT**

**Background:**

Facial emotion recognition is a significant aspect of human-computer interaction and has applications in various fields, including user experience design, mental health monitoring, and human-robot interaction. Real-time analysis of facial expressions enables immediate and dynamic responses in different scenarios.

**Objective:**

The objective of this project is to develop a real-time facial emotion recognition system using a convolutional neural network (CNN). The system should be able to accurately detect and classify facial expressions in live video feeds, providing instantaneous feedback on the emotional states of individuals.

**Challenges:**

- Achieving real-time performance in both face detection and emotion prediction.

- Handling variations in lighting conditions, facial expressions, and face orientations.

- Ensuring the model's accuracy and generalization to different individuals and demographics.

**Evaluation Metrics**:

The system's performance will be evaluated based on:

- Accuracy of emotion recognition in real-time scenarios.

- Processing speed and efficiency in capturing, detecting faces, and predicting emotions.

- Robustness to variations in environmental conditions.

**1.3 MODULES DESCRIPTION**

**KERAS MODULE**

Keras is an open-source deep learning framework written in Python. It provides a user-friendly, high-level API for building and training deep neural networks. Keras provides pre-defined models for common deep learning architectures, including convolutional neural networks (CNNs), recurrent neural networks (RNNs), and deep feedforward networks. These models can be easily used for various tasks such as image classification, natural language processing, and more.

**TENSORFLOW MODULE**

TensorFlow is a leading open-source deep learning framework developed by Google, well-suited for various machine learning tasks. TensorFlow boasts a thriving community, extensive documentation, and tools for distributed computing and model deployment. TensorFlow 2.0 emphasizes ease of use and integrates the Keras high-level API. Its compatibility, rich ecosystem, visualization tools, and open-source nature make it a top choice for developing, training, and deploying machine learning and deep learning models across a range of applications.

**NUMPY MODULE**

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O and much more.

**MATPLOTLIB**

Matplotlib is a library in Python that enables users to generate visualizations like histograms, scatter plots, bar charts, pie charts and much more. Seaborn is a visualization library that is built on top of Matplotlib. It provides data visualizations that are typically more aesthetic and statistically sophisticated.

**OpenCV (Open Source Computer Vision):**

OpenCV simplifies image processing tasks and transforming pixel values. This is fundamental for various computer vision applications. OpenCV provides a collection of algorithms for computer vision tasks, such as feature detection, object recognition, and image stitching. These algorithms form the foundation for building intelligent applications.Due to its efficiency, OpenCV is suitable for real-time applications, including real-time object detection, facial recognition, and augmented reality.

**1.4 MODELS DESCRIPTION**

**SEQUENTIAL**

The `Sequential` model in TensorFlow/Keras is a straightforward and linearly stacked neural network architecture. It is a powerful and easy-to-use tool for constructing simple feed forward neural networks where each layer flows sequentially from the previous one. This model is particularly well-suited for tasks like image classification or regression, where the data can be transformed through a series of layers, progressively extracting higher-level features.

In a `Sequential` model, layers are added one at a time, and each subsequent layer takes the output of the previous layer as its input.

**1.5 MODEL ARCHITECTURE**

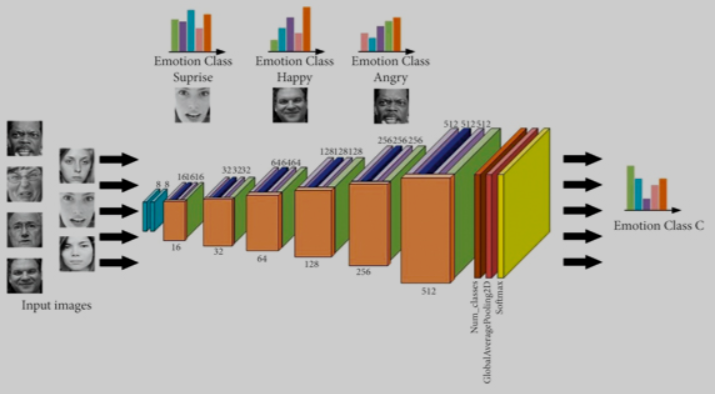
****

Figure 1.5.1 Convolution Architecture

**PROPOSED SYSTEM**

**2.1 SYSTEM DESIGN**

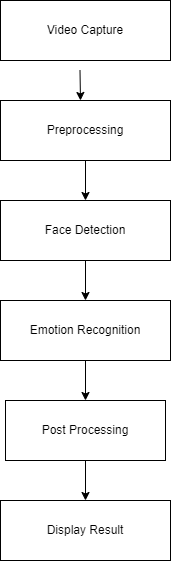
****

Figure 2.1.1 Block Diagram

The Facial recognition dataset that is used in the prediction of the real time emotion is used in building various deep learning models. Firstly, the dataset is preprocessed by resizing all the images to uniform pixel range, augmentation of the images to avoid overfitting the images. This is done to provide accurate predictions by the deep learning algorithms. Then we also use various pretrained deep learning models compared with various parameters and accuracy of each model is found out.

**SYSTEM REQUIREMENT**

HARDWARE REQUIREMENTS

Computer with Windows 10 /Linux Operating System

SOFTWARE REQUIREMENTS

Jupyter Notebook Version 6.5.4

PACKAGES : NumPy 1.20.3 , Matplotlib 3.8.0 , Keras 2.14.0 , Tensorflow 2.12.0 ,OpenCv 2.4.9

**IMPLEMENTATION**

The Project is implemented by using Python OpenCv Module as the Back-end for making predictions.

**4.1 DATASET**

The dataset is organized into 2 folders (train, test, ) and contains sub folders for each image category (Happy/Sad/Fear/Angry/Surprise/Disgusted/Neutral). There are 35,887 X-Ray images (PNG) and 7 categories (Happy/Sad/Fear/Angry/Surprise/Disgusted/Neutral).

Each data sample is labeled with the emotion or emotions . Common emotion labels include happiness, sadness, anger, surprise, fear, disgust, and neutral. The dataset should encompass a wide range of emotions and scenarios to ensure the model's robustness in real-world applications. It may include posed expressions as well as spontaneous ones.

The dataset designed to capture real-time scenarios, meaning that the emotions depicted in the data are dynamic and can change over time. The dataset include samples from individuals of different ages, genders, and cultural backgrounds to ensure the model's generalization across diverse populations. The dataset is typically divided into training, validation, and test sets to facilitate model training and evaluation.

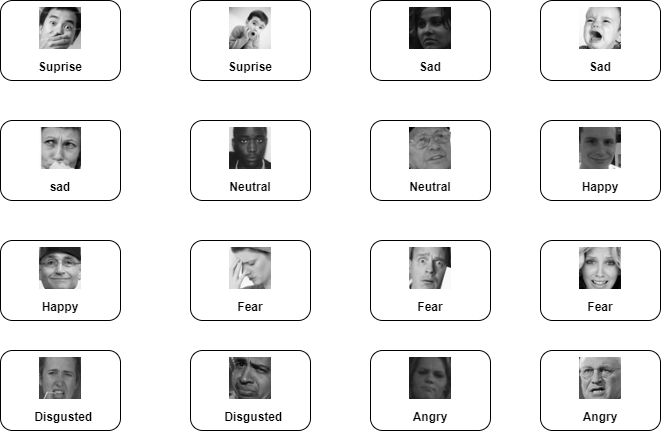


Figure 2.1.2 Sample Datasets

**4.2 SOURCE CODE**

import tensorflow as tf

from tensorflow.keras import layers, models

from tensorflow.keras.preprocessing.image import ImageDataGenerator

train\_data\_dir = 'C:\\Users\\Jeyaram\\Downloads\\archive (9)\\train'

test\_data\_dir = 'C:\\Users\\Jeyaram\\Downloads\\archive (9)\\test'

train\_datagen = ImageDataGenerator(rescale=1./255)

test\_datagen = ImageDataGenerator(rescale=1./255)

batch\_size = 32

img\_height = 48

img\_width = 48

train\_generator = train\_datagen.flow\_from\_directory(

train\_data\_dir,

target\_size=(img\_height, img\_width),

batch\_size=batch\_size,

class\_mode='categorical',

color\_mode='grayscale'

)

test\_generator = test\_datagen.flow\_from\_directory(

test\_data\_dir,

target\_size=(img\_height, img\_width),

batch\_size=batch\_size,

class\_mode='categorical',

color\_mode='grayscale'

)

num\_classes = 7

model = models.Sequential()

model.add(layers.Conv2D(32, (3, 3), activation='relu', input\_shape=(img\_height, img\_width, 1)))

model.add(layers.Flatten())

model.add(layers.Dense(64, activation='relu'))

model.add(layers.Dense(num\_classes, activation='softmax')) # Assuming 'num\_classes' is the number of classes

model.compile(optimizer='adam', loss='categorical\_crossentropy', metrics=['accuracy'])

model.save('my\_model.keras')

import cv2

import numpy as np

from tensorflow.keras.models import load\_model

emotion\_model = load\_model('my\_model.keras')

emotion\_labels = ['Angry', 'Disgust', 'Fear', 'happy', 'Sad', 'Surprise', 'neutral']

face\_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade\_frontalface\_default.xml')

def predict\_emotion(face\_roi):

face\_roi\_gray = cv2.cvtColor(face\_roi, cv2.COLOR\_BGR2GRAY)

face\_roi\_gray = cv2.resize(face\_roi\_gray, (48, 48))

face\_roi\_gray = face\_roi\_gray / 255.0

face\_roi\_gray = np.expand\_dims(face\_roi\_gray, axis=0)

predicted\_emotion = np.argmax(emotion\_model.predict(face\_roi\_gray))

return emotion\_labels[predicted\_emotion]

cap = cv2.VideoCapture(0)

while True:

ret, frame = cap.read()

gray = cv2.cvtColor(frame, cv2.COLOR\_BGR2GRAY)

faces = face\_cascade.detectMultiScale(gray, scaleFactor=1.3, minNeighbors=5)

for (x, y, w, h) in faces:

face\_roi = frame[y:y+h, x:x+w]

emotion = predict\_emotion(face\_roi)

cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 0, 0), 2)

cv2.putText(frame, emotion, (x, y-10), cv2.FONT\_HERSHEY\_SIMPLEX, 0.9, (255, 0, 0), 2, cv2.LINE\_AA)

cv2.imshow('Emotion Recognition', frame)

if cv2.waitKey(1) & 0xFF == ord('q'):

break

cap.release()

cv2.destroyAllWindows()

import tensorflow as tf

from tensorflow.keras import layers, models

from tensorflow.keras.preprocessing.image import ImageDataGenerator

import matplotlib.pyplot as plt

model.compile(optimizer='adam', loss='categorical\_crossentropy', metrics=['accuracy'])

history = model.fit(

train\_generator,

epochs=5,

validation\_data=test\_generator

)

plt.plot(history.history['loss'], label='Training Loss')

plt.plot(history.history['val\_loss'], label='Validation Loss')

plt.title('Loss Percentage Plot')

plt.xlabel('Epoch')

plt.ylabel('Loss')

plt.legend()

plt.show()

**4.3 RESULT**

****

Figure 4.3.1 Real time emotion detection

****

Figure 4.3.2 Real time emotion detection

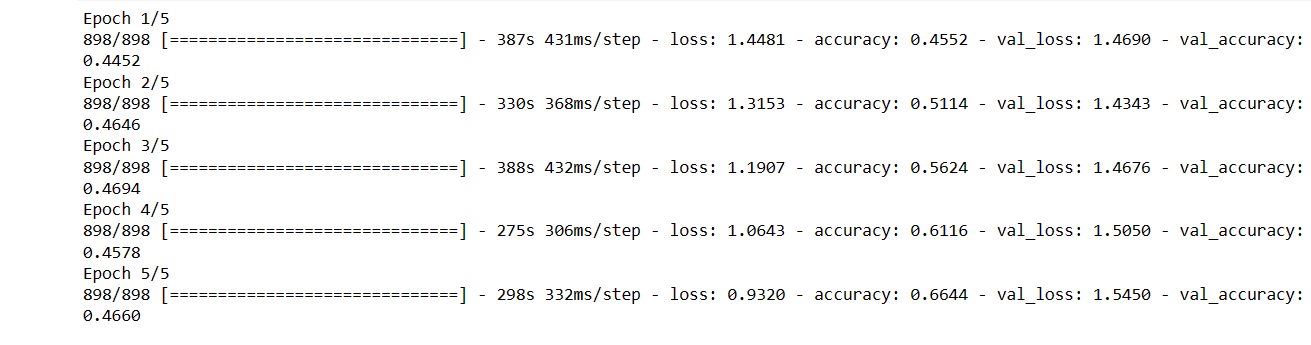
****

Figure 4.3.3 Accuracy metrics

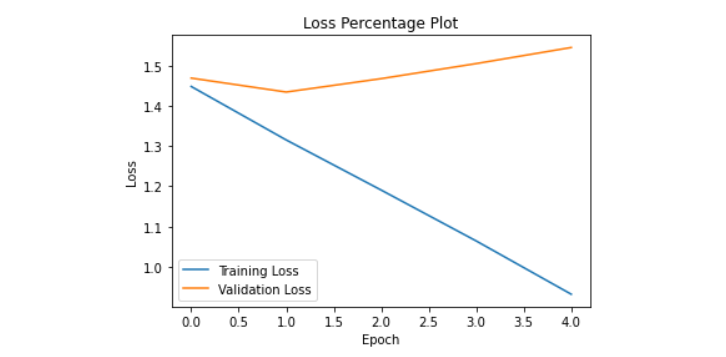
****

Figure 4.3.4 Accuracy and loss plot

**CONCLUSION**

In the realm of computer vision and deep learning, this project presents an innovative real-time emotion recognition system employing a Convolutional Neural Network (CNN) in conjunction with TensorFlow and OpenCV. The model is trained on a dataset of facial expressions, and its architecture, while straightforward, exhibits a commendable ability to discern and classify emotions accurately. The preprocessing steps, including rescaling and augmentation using the ImageDataGenerator, contribute to the model's adaptability to various facial expressions and lighting conditions. The decision to use grayscale images enhances computational efficiency while retaining the essential features for emotion prediction.

An integral component of the system lies in the integration of the Haar Cascade classifier for face detection in the live video feed. This not only streamlines the real-time aspect of the application but also ensures the efficient identification of facial regions for subsequent emotion analysis. The interactive display of predicted emotions overlaid on the video feed, complete with bounding boxes around detected faces, provides a visually compelling demonstration of the model's capabilities. This integration of machine learning with computer vision techniques fulfills the practical demands of real-time applications, making it accessible and engaging.

**BIBLIOGRAPHY**

1. https://www.kaggle.com/datasets//real time emotion recognition
2. <https://www.analyticsvidhya.com/blog/2017/06/transfer-learning-the-art-of-fine-tuning-a-pre-trained-model/>
3. [https://ieeexplore.ieee.org/document](https://ieeexplore.ieee.org/document/10242024)
4. <https://www.mathworks.com/help/images/deep-learning.html>
5. https://docs.opencv.org/4.x/d9/df8/tutorial\_root.html