**Facial Emotion Recognition using Deep Learning**

*A Project Based Learning Report Submitted in partial fulfilment of the requirements for the award of the degree*

*of*

**Bachelor of Technology**

**in The Department of Electronics and Communication Engineering**

**Deep Network Architecture**

**(23EC2224F)**

Submitted by

**2310040049 - D. Navya**

**2310040058 - M. Sirisha**

**2310040065 - M. Divya**

Under the guidance of

**Dr. Ravi Boda**



Department of Electronics and Communication Engineering

Koneru Lakshmaiah Education Foundation, Aziz Nagar

Aziz Nagar – 500075

April - 2025.

**Abstract**

Facial Emotion Recognition (FER) via deep learning is a state-of-the-art method for recognizing human emotions from facial expressions. This project utilizes Convolutional Neural Networks (CNNs) to detect and study facial features for effective emotion classification, including happiness, sadness, anger, fear, surprise, and neutrality. The model is trained on massive datasets to enhance recognition accuracy and robustness for varying facial variations, lighting, and angles.

For the purpose of performance improvement, transfer learning, data augmentation, and fine-tuning techniques are utilized. The suggested system provides real-time emotion detection with high reliability and efficiency. It has broad applications in areas such as human-computer interaction, mental health monitoring, customer sentiment analysis, and security surveillance.

The application of deep learning allows the model to learn intricate facial patterns without the need for manual feature extraction, hence being versatile with different datasets. The system can be implemented in different applications such as mobile apps, smart surveillance, and virtual assistants, and it improves the user experience. The project contributes to affective computing by enhancing the accuracy and efficiency of FER systems, showing the potential of deep learning in real-time emotion recognition and analysis.

**Introduction**

Facial Emotion Recognition (FER) is an essential part of human-computer interaction that allows machines to recognize and respond to human emotions. Emotions are a key component of communication, affecting decision-making, behavior, and social interactions. Automated emotion detection increases applications across multiple fields, such as healthcare, security, marketing, and virtual assistants.

Deep learning, especially Convolutional Neural Networks (CNNs), has much enhanced FER by allowing automatic feature extraction and classification. Formerly, methods were based on handcrafted features, but deep learning omits manual selection of features, gaining more accuracy and robustness. With the possibility of training on extensive datasets, CNNs are capable of identifying emotions in various facial expressions, lighting, and demographics.

The purpose of this project is to build a real-time FER system based on deep learning approaches. The proposed model will recognize the emotions like happiness, sadness, anger, fear, surprise, and neutrality from face images. Preprocessing activities like face detection, normalization, and data augmentation are some of the essential steps to ensure optimal model performance. The proposed system can be deployed in multiple real-world applications like mental health tracking, customer emotion analysis, and intelligent surveillance.

Facial Emotion Recognition (FER) is an essential component of human-computer interaction, allowing machines to recognize and respond to human emotions. Emotions are a significant part of communication, affecting decision-making, behavior, and social interactions. Emotion detection automation improves applications across many fields, such as healthcare, security, marketing, and virtual assistants.

Deep learning, especially Convolutional Neural Networks (CNNs), has greatly enhanced FER through the ability to automatically extract features and classify them. Handcrafted features were used in traditional techniques, but deep learning does not require manual selection of features and is more accurate and robust. CNNs are trained on large datasets and can identify emotions under varying facial expressions, lighting, and demographics.

This project is to design a real-time FER system based on deep learning methods. The model is supposed to recognize happiness, sadness, anger, fear, surprise, and neutrality from face images. Face detection, normalization, and data augmentation are major preprocessing steps to improve model performance. The system can be applied to many real-world applications like mental health checking, customer mood analysis, and smart surveillance.

**METHODOLOGY**

The Facial Emotion Recognition (FER) system employs systematic deep learning procedures to deliver high accuracy with real-time operational capabilities. The system development follows several sequential steps which include selecting appropriate datasets as well as pre-processing them before designing models for training along with evaluation and software implementation.

**1. Data Collection and Preprocessing**

Data preprocessing stands as a vital initial point because it advances model accuracy levels and generalization traits. The following steps are performed:

**1.1 Dataset Selection**

Four publicly accessible databases such as FER-2013 together with CK+ and JAFFE and RAF-DB and AffectNet serve as training and testing resources.

The emotional annotations present in these datasets cover happiness, sadness, anger, fear, surprise, disgust together with neutral expressions.

**1.2 Face Detection and Extraction**

The isolation of facial regions within images happens through the implementation of Face detection algorithms which include Haar cascades, MTCNN, or OpenCV’s DNN module.

All initially extracted faces get adjusted to one of the standardized input dimensions such as 48×48 or 224×224 pixels.

**1.3 Data Normalization and Augmentation**

The normalization technique scales pixel values from 0 to 1 to establish equal distribution throughout all input data.

Model generalization receives improvement through the application of augmentation techniques.

**Rotation:** Helps the model recognize faces from different angles.

**Flipping:** Enhances robustness to orientation variations.

**Brightness adjustment:** Ensures consistency under varying lighting conditions.

Applying Gaussian noise prevents model overfitting thus improving the robustness of the model.

**2. Model Selection and Architecture Design**

CNNs as deep learning neural networks function to extract features while classifying emotional expressions in the input data.

**2.1 CNN-Based Approach**

A customized CNN model benefits from multiple convolutional layers together with ReLU activations and batch normalization that add dropout as a prevention against overfitting.

* The architecture consists of:
* Convolutional layers for feature extraction.
* Pooling layers for dimensionality reduction.
* Fully connected layers for classification.
* Softmax activation for final emotion prediction.

**2.2 Transfer Learning with Pre-trained Models**

Existed deep learning models VGG16, ResNet50, MobileNet and EfficientNet and InceptionNet undergo fine-tuning procedures for emotion detection application.

The adoption of transfer learning helps data scientists cut training durations and achieve better outcomes from limited training samples.

**2.3 Attention Mechanisms and Feature Fusion**

The study investigates complex attention mechanisms (SE-Net and CBAM) specifically to detect essential facial areas.

The recognition of emotions benefits from multiple signal integration between facial expressions and audio and physiological data.

**3. Training and Optimization**

The performance improvement of models depends on several training strategies used alongside optimization methods.

**3.1 Training Strategy**

The model uses Categorical Cross-Entropy Loss as its primary loss function for performing multi-class determination.

Adam along with RMSprop and SGD serve as compatible optimizers for technical performance evaluation. The choice depends on achieved results.

The combination of Batch Size with Learning Rate underwent needed adjustments through a process of hyperparameter tuning.

**3.2 Regularization Techniques**

Hidden neuron components in Dropout Layers become deactivated by chance in order to stop the model from fitting data patterns excessively.

L2 Regularization offers a mechanism to decrease model complexity.

**3.3 Hyperparameter Tuning**

The process of refining learning rate along with number of layers and filter size and activation functions uses Grid Search or Bayesian Optimization approaches.

**4. Model Evaluation and Performance Metrics**

A series of performance measures are used to conduct thorough assessment of the trained model to ensure its operational stability.

**4.1 Performance Metrics**

* **Accuracy**: Measures overall correctness.
* **Precision & Recall:** Evaluates the effectiveness of classification.
* **F1-score:** Ensures a balance between precision and recall.
* **Confusion Matrix:** Analyzes misclassification trends.
* **ROC-AUC Curve:** Assesses classification confidence.

**4.2 Validation Strategies**

**Train-Test Split:** 80% training, 10% validation, 10% testing.

The stability of models and model variance reduction becomes possible through K-Fold Cross-Validation.

**4.3 Real-Time Testing**

workers utilize an implementation of OpenCV and TensorFlow to evaluate real-time operational performance through the deployment of the model to streaming video content.

**5. Deployment and Real-Time Integration**

After successful training the developed model gets deployed within real-world environments.

**5.1 Real-Time Implementation**

A framework integrates OpenCV and TensorFlow to support real-time emotion detection through its system.

Testing of webcam-based emotion recognition has been developed.

**5.2 Edge Deployment**

The optimized FER models get implemented specifically for on-device processing through smartphone platforms or Raspberry Pi and Jetson Nano units.

Real-time inference occurs through the utilization of both TensorFlow Lite and ONNX technologies.

**5.3 Application Areas**

Healthcare: Emotion-based mental health analysis.

Human-Computer Interaction incorporates smart assistants and gaming as applications that depend on the technology.

* **Security & Surveillance:** Behavior analysis in public spaces.

The technology analyzes business sentiment as part of marketing and customer analytics operations.

**6. Future Enhancements**

* **Hybrid Emotion Recognition:** Combining facial expressions with voice and physiological signals.
* **3D Facial Emotion Recognition:** Improving accuracy with depth data.
* **Explainable AI (XAI):** Enhancing model interpretability in FER applications.

The method delivers an extensive solution for creating a scalable real-time system that performs Facial Emotion Recognition effectively.

**LITERATURE SURVEY**

Artificial Intelligence together with computer vision and affective computing make Facial Emotion Recognition (FER) one of the most extensively investigated fields. Before deep learning emerged traditional methods in FER depended on human engineers to create features that were later classified by human experts but deep learning automates both extraction and classification. The research field undergoes an evaluation of its past developments and existing studies within this section.

**1. Traditional Approaches in Facial Emotion Recognition**

The initial stage of FER consisted of manual facial feature extraction through methods based on features.

The geometric-based technique makes use of facial landmarks (eyes and nose and mouth) in order to create facial expression models

The appearance-based method included Geographic-based approaches that used texture and shape descriptors including Local Binary Patterns (LBP) together with Gabor filters and Histogram of Oriented Gradients (HOG).

Facial emotional information classification involved facial deformation modeling through Active Appearance Models (AAMs) and Active Shape Models (ASMs)

These techniques demonstrated poor accuracy levels because they responded poorly to changes in light exposure and the blocking of areas on the face as well as facial expression variations.

**2. Machine Learning-Based Methods**

Since the emergence of machine learning researchers began developing different classifiers for FER.

Support Vector Machines (SVms) work with extracted facial features to achieve emotion classification according .

Random Forests and Decision Trees: Applied for feature-based classification.

Researchers used Hidden Markov Models (HMMs) together with Principal Component Analysis (PCA) to perform feature dimensionality reduction The methodology increased accuracy levels yet continued to need extensive manual interventions during feature attribute customization.

**3. Deep Learning-Based Approaches**

Through deep learning FER experienced a revolutionary change when computers gained the ability to automatically extract necessary features.

The Convolutional Neural Networks system reached superior accuracy levels by establishing hierarchical learning pattern

VGG-Face, ResNet, and InceptionNet: Used for transfer learning in FER applications

RNNs together with LSTMs function in video-based FER to detect emotional changes across temporal dimensions

Attention Mechanisms: Focused on important facial regions for better classification

**4. Recent Advances and Hybrid Models**

The combination of Transformer-Based Models and their two main frameworks Vision Transformers and hybrid CNN-ViT architectures produces excellent results according to research by

Multi-Modal Approaches: Combining facial expressions with speech, EEG signals, and physiological data for enhanced recognition.

The integration of depth sensors such as Kinect helps emotional classification by improving performance in authentic real-world scenarios.

The application of MobileNet and EfficientNet lightweight models enables edge devices to execute FER in real time.

**5. Challenges and Research Gaps**

The progress made has not resolved all the problems which FER technology experiences.

The system needs to handle interruptions that occur when face masks or glasses cover the face.

Variability in expressions across cultures and demographics.

Real-time processing constraints for edge devices.

The present dataset contains ethical issues and demonstrates data bias characteristics.

**6. Conclusion**

Deep learning replaced feature-based methods to deliver a substantial boost in FER accuracy levels. Current scholarly efforts concentrate on developing multichannel identification systems and attentive network frameworks and real-time execution solutions. Future research needs to tackle dataset prejudice as well as develop compact models suitable for edge deployment systems and create interpretable artificial intelligence for emotional recognition systems.

**CODE:**

from zipfile import ZipFile

file\_name = "emojify.zip"

with ZipFile(file\_name, 'r') as zip:

zip.extractall()

print("Done")

import numpy as np

import cv2

from keras.models import Sequential

from keras.layers import Dense, Dropout, Flatten

from keras.layers import Conv2D

from keras.optimizers import Adam

from keras.layers import MaxPooling2D

from keras.preprocessing.image import ImageDataGenerator

train\_dir = 'train'

val\_dir = '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'))

emotion\_model.compile(loss='categorical\_crossentropy',optimizer=Adam(lr=0.0001, decay=1e-6),metrics=['accuracy'])

emotion\_model\_info = emotion\_model.fit\_generator(

train\_generator,

steps\_per\_epoch=28709 // 64,

epochs=50,

validation\_data=validation\_generator,

validation\_steps=7178 // 64)

#Saving the model

emotion\_model.save('model.h5')

from keras.models import load\_model

emotion\_model = load\_model('model.h5')

def emotion\_analysis(emotions):

objects = ('angry', 'disgust', 'fear', 'happy', 'sad', 'surprise', 'neutral')

y\_pos = np.arange(len(objects))

plt.bar(y\_pos, emotions, align='center', alpha=0.5)

plt.xticks(y\_pos, objects)

plt.ylabel('percentage')

plt.title('emotion')

plt.show()

#define Capturing an image on Colab from here: https://colab.research.google.com/notebook#fileId=1OnUy6eFE7XhdfGfAHDCqQxpwueTOj\_NO

from IPython.display import display, Javascript

from google.colab.output import eval\_js

from base64 import b64decode

def take\_photo(filename='photo.jpg', quality=0.8):

js = Javascript('''

async function takePhoto(quality) {

const div = document.createElement('div');

const capture = document.createElement('button');

capture.textContent = 'Capture';

div.appendChild(capture);

const video = document.createElement('video');

video.style.display = 'block';

const stream = await navigator.mediaDevices.getUserMedia({video: true});

document.body.appendChild(div);

div.appendChild(video);

video.srcObject = stream;

await video.play();

// Resize the output to fit the video element.

google.colab.output.setIframeHeight(document.documentElement.scrollHeight, true);

// Wait for Capture to be clicked.

await new Promise((resolve) => capture.onclick = resolve);

const canvas = document.createElement('canvas');

canvas.width = video.videoWidth;

canvas.height = video.videoHeight;

canvas.getContext('2d').drawImage(video, 0, 0);

stream.getVideoTracks()[0].stop();

div.remove();

return canvas.toDataURL('image/jpeg', quality);

}

''')

display(js)

data = eval\_js('takePhoto({})'.format(quality))

binary = b64decode(data.split(',')[1])

with open(filename, 'wb') as f:

f.write(binary)

return filename

take\_photo()

import cv2

def facecrop(image):

facedata = '/content/haarcascade\_frontalface\_alt.xml'

cascade = cv2.CascadeClassifier(facedata)

img = cv2.imread(image)

try:

minisize = (img.shape[1],img.shape[0])

miniframe = cv2.resize(img, minisize)

faces = cascade.detectMultiScale(miniframe)

for f in faces:

x, y, w, h = [ v for v in f ]

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

sub\_face = img[y:y+h, x:x+w]

cv2.imwrite('capture.jpg', sub\_face)

#print ("Writing: " + image)

except Exception as e:

print (e)

if name == 'main':

facecrop('/content/photo.jpg')

#Testing a file.

from keras.preprocessing import image

from keras.preprocessing.image import ImageDataGenerator

import numpy as np

import matplotlib.pyplot as plt

file = '/content/capture.jpg'

true\_image = image.load\_img(file)

img = image.load\_img(file, color\_mode="grayscale", target\_size=(48, 48))

x = image.img\_to\_array(img)

x = np.expand\_dims(x, axis = 0)

x /= 255

custom = emotion\_model.predict(x)

emotion\_analysis(custom[0])

x = np.array(x, 'float32')

x = x.reshape([48, 48]);

plt.imshow(true\_image)

plt.show()