VISVESVARAYA TECHNOLOGICAL UNIVERSITY

BELAGAVI-590018

A Digital Image Processing Mini Project Report on

"RECOGNITION OF FACE EMOTION IN REAL TIME"

Submitted in partial fulfillment of the requirements for the VI semester

and award of the degree of Bachelor of Engineering in AI & ML

of Visvesvaraya Technological University, Belagavi

Submitted by:

Anushree Y N	*1RN20AI014*
Goddati Venkata Sai Akhil	*1RN20AI022*
Md Imtiyaz Alam	*1RN20AI031*

Under the Guidance of:

Ms. Ashwini K
Assistant Professor
Department of AI & ML

Department of AI & ML
RNS Institute of Technology
Channasandra, Dr.Vishnuvardhan Road, Bengaluru-560 098

2022-2023

RNS Institute of Technology

Channasandra, Dr.Vishnuvardhan Road, Bengaluru-560098

DEPARTMENT OF ARTIFICIAL INTELLIGENCE & MACHINE LEARNING

ESTD : 2001

An Institute with a Difference

CERTIFICATE

Certified that the mini project work entitled **"RECOGNITION OF FACE EMOTION IN REAL TIME"** has been successfully carried out by **Anushree Y N** bearing USN **"**1RN20AI014**"**, **Goddati Venkata Sai Akhil** bearing USN "1RN20AI022" and Md **Imtiyaz Alam** bearing USN "1RN20AI031", bonafide students of **"RNS Institute of Technology"** in partial fulfillment of the requirements for the 6th semester of **"Bachelor of Engineering in Artificial Intelligence and Machine Learning Engineering of Visvesvaraya Technological University"**, Belagavi, during academic year 2022-2023. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The project report has been approved as it satisfies the Digital Image Processing laboratory requirements of 6th semester BE in AI and ML.

Signature of the Guide	Signature of the HoD	Signature of the Principal
Ms. Ashwini K Assistant Professor Dept. of AI and ML RNSIT, Bengaluru	**Dr. Harsha S** Professor & Head Dept. of AI and ML RNSIT, Bengaluru	**Dr. H S Ramesh Babu** Principal RNSIT, Bengaluru

Name & Signature

Examiner 1:

Examiner 2:

Acknowledgement

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. A number of personalities, in their own capacities have helped us in carrying out this project work. We would like to take this opportunity to thank them all.

We are grateful to **Management** and **Dr. M K Venkatesha**, Director and **Dr. H S Ramesh Babu** Principal, RNSIT, Bangalore, for their support towards completing this mini project.

We would like to thank **Dr. Harsha S**, Head, Department of Artificial Intelligence and Machine Learning Engineering, RNSIT, Bangalore, for his valuable suggestions and expert advice.

We deeply express our sincere gratitude to our guide **Ms. Ashwini K**, Assistant Professor, Department of AI ML, RNSIT, Bangalore, for her able guidance, regular source of encouragement and assistance throughout this project.

We would like to thank all the teaching and non-teaching staff of the Department of Artificial Intelligence and Machine Learning Engineering, RNSIT, Bengaluru-98 for their constant support and encouragement.

Signature

Anushree Y N	Goddati Venkata Sai Akhil	Md Imtiyaz Alam
1RN20AI014	1RN20AI022	1RN20AI031

Abstract

This project presents a real-time face recognition system designed to detect and classify emotions from facial expressions. The system employs a convolutional neural network (CNN) architecture, which has proven to be highly effective in image recognition tasks. By leveraging deep learning and computer vision techniques, the system accurately analyzes live video feeds and identifies emotions, including anger, happiness, sadness, and more.

The project begins by training the CNN model using image data generators, which preprocess and augment the training and validation data. The training process utilizes a large dataset of labeled facial expression images, enabling the model to learn and generalize emotions from different individuals.

After training, the model's structure is saved in a JSON file, while the learned weights are stored in an HDF5 file. The system accesses the webcam feed using OpenCV and applies a Haar cascade classifier for face detection. The CNN model then predicts the emotions based on the processed ROIs, and the results are displayed on the video feed in real-time. The real-time face recognition system provides practical applications in various domains. It can enhance human-computer interaction by enabling systems to respond dynamically to users' emotional states. In market research, it can be used to analyze customer reactions to products and advertisements. The project showcases the potential of AI-based facial recognition and deep learning for real-time emotion detection, highlighting the significance of understanding and interpreting human emotions in interactive settings.

Index

Contents
Page No.

List of Figures

Chapter 1

Introduction

Digital image processing is a field of study that focuses on the manipulation and analysis of digital images using computer algorithms. It plays a crucial role in various applications, including medical imaging, computer vision, and remote sensing. By applying mathematical operations and algorithms to images, digital image processing enables tasks such as image enhancement, restoration, segmentation, feature extraction, and object recognition. It allows us to extract valuable information from images, improve image quality, and extract meaningful patterns and features for further analysis. Digital image processing has revolutionized fields like healthcare, surveillance, and multimedia, empowering us to extract valuable insights from visual data.

1.1 History of Digital Image processing

The history of digital image processing dates back to the mid-20th century when computers began to be utilized for image analysis and manipulation. Here is a brief overview of the significant milestones in the history of digital image processing:

- **1950s-1960s**: The beginnings of digital image processing can be traced to the development of digital computers. In the 1950s, early experiments focused on digitizing images and developing basic algorithms for image analysis.
- **1970s**: With the advent of more powerful computers, researchers began exploring advanced techniques such as image restoration, enhancement, and compression. The field witnessed significant progress during this period, with the introduction of algorithms like the Fast Fourier Transform (FFT) for image filtering and restoration.
- **1980s**: The emergence of personal computers and the availability of affordable image processing software led to wider adoption of digital image processing techniques. Image processing started finding applications in various fields, including medical imaging, remote sensing, and industrial inspection.
- **1990s**: The development of more sophisticated algorithms and computational techniques

paved the way for breakthroughs in image segmentation, object recognition, and pattern analysis. This decade also witnessed the rise of digital imaging technologies in consumer electronics, such as digital cameras and image editing software.

- **2000s-Present**: The advancement of machine learning and deep learning techniques has revolutionized digital image processing. Convolutional neural networks (CNNs) and other deep learning architectures have achieved remarkable success in image classification, object detection, and semantic segmentation tasks.

Today, digital image processing plays a vital role in various domains, including healthcare, astronomy, robotics, surveillance, and entertainment. It continues to evolve with the development of new algorithms, hardware advancements, and the integration of artificial intelligence, enabling us to extract valuable insights from visual data and enhance our understanding of the world around us.

1.2 Stages in Digital Image Processing

Digital image processing involves stages such as image acquisition, enhancement, restoration, color image processing, wavelets and multiresolution processing, compression, morphological processing, segmentation, representation, and object recognition. These stages encompass a range of techniques and algorithms for acquiring, improving, analyzing, and interpreting digital images. Each stage plays a crucial role in transforming and extracting meaningful information from images.

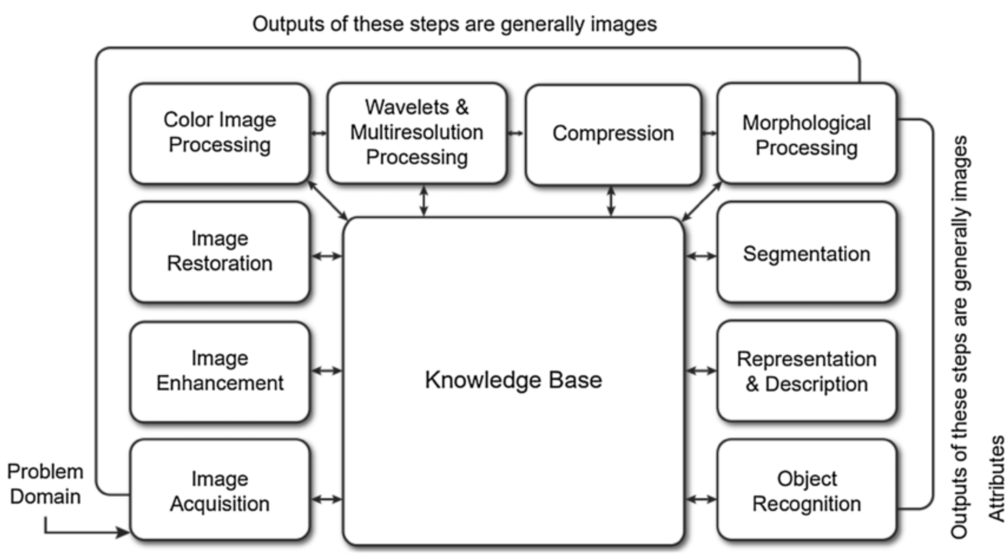

Figure 1.2: Stages in Digital Image Processing

Stages involved in Digital Image Processing are:

- **Image Acquisition**: The process of capturing or digitizing images using cameras, scanners, or other imaging devices. It involves converting analog signals (light) into digital data, forming the basis for subsequent processing stages.

- **Image Enhancement**: Techniques applied to improve the quality, clarity, or visual appearance of an image. Enhancement methods include adjusting brightness/contrast, sharpening, noise reduction, and histogram equalization.

- **Image Restoration**: The process of recovering an image from degraded or corrupted versions. Restoration techniques aim to remove noise, blur, or other artifacts caused by factors such as sensor noise, motion blur, or transmission errors.

- **Color Image Processing**: Dealing with the analysis and manipulation of color images. This stage involves techniques such as color space transformations, color correction, color image enhancement, and color-based object recognition.

- **Wavelets and Multiresolution Processing**: Utilizing wavelet transform and multiresolution analysis for image processing. Wavelet techniques offer advantages in representing images at different scales and extracting both frequency and spatial information.

- **Compression**: Reducing the size of the image data for efficient storage and transmission. Compression techniques aim to remove redundant or irrelevant information while preserving essential image features. Common compression methods include JPEG, PNG, and MPEG.

- **Morphological Processing**: Utilizing mathematical morphology operations, such as erosion, dilation, opening, and closing, to analyze the shape, structure, and spatial relationships in images.

- **Segmentation**: Partitioning an image into meaningful regions or objects. Segmentation techniques aim to separate different objects or regions based on properties such as color, texture, or intensity.

- **Representation and Description**: Representing image features and objects using suitable descriptors, such as shape, texture, or color. This stage involves extracting meaningful features and creating representations that enable further analysis and recognition.

- **Object Recognition**: Identifying and classifying objects or patterns within an image. Object recognition techniques utilize machine learning, pattern recognition, and feature matching to identify and categorize objects based on their visual characteristics.

1.3 Applications of Digital Image Processing

Digital image processing finds applications in numerous fields and industries. Some of the key applications include:

- **Medical Imaging**: Digital image processing is extensively used in medical diagnostics, including X-ray, MRI, CT scans, and ultrasound. It aids in image enhancement, segmentation, feature extraction, and pattern recognition for improved diagnosis of diseases and abnormalities.

- **Surveillance and Security**: Image processing techniques are employed in video surveillance systems for face recognition, object tracking, and anomaly detection. It enhances the effectiveness of security systems and assists in identifying potential threats or suspicious activities.

- **Remote Sensing**: Digital image processing is vital in analyzing satellite and aerial images for applications like environmental monitoring, land cover classification, urban planning, and agriculture. It enables the extraction of valuable information about the Earth's surface and assists in making informed decisions.

- **Robotics and Automation**: Image processing is integral to vision-based robotics and automation systems. It enables robots to perceive and interpret visual information, facilitating tasks such as object detection, recognition, and navigation in dynamic environments.

- **Entertainment and Media**: Digital image processing plays a significant role in the entertainment industry. It is used for special effects, image editing, color correction, image rendering, and image-based rendering techniques in movies, video games, virtual reality (VR), and augmented reality (AR) applications.
- **Biometrics**: Image processing techniques are employed in biometric systems for face recognition, fingerprint recognition, iris recognition, and other biometric modalities. It enhances security and authentication systems by verifying an individual's unique biological traits.
- **Quality Control and Inspection**: Image processing is utilized in manufacturing industries for automated quality control and inspection of products. It detects defects, measures dimensions, and ensures the consistency and accuracy of products on assembly lines.
- **Geographical Information Systems (GIS)**: Digital image processing assists in analyzing and interpreting geospatial data. It aids in land cover mapping, terrain analysis, and feature extraction for creating accurate maps and spatial databases.
- **Astrophysics and Astronomy**: Image processing techniques are used to enhance astronomical images, remove noise, and extract valuable information about celestial objects and phenomena. It aids in studying the universe and analyzing astronomical data.
- **Forensics**: Image processing plays a crucial role in forensic investigations. It helps in analyzing crime scene images, enhancing details, and performing facial recognition or forensic comparison to assist law enforcement agencies.

These are just a few examples of the diverse applications of digital image processing, demonstrating its significance and impact across various fields.

1.4 Introduction of Recognition of Face Emotion in Real Time

Recognizing emotions from facial expressions in real-time is an important area of research in computer vision and artificial intelligence. Real-time facial emotion recognition systems have practical applications in fields like human-computer interaction and virtual reality. These systems can enhance user experiences by enabling computers or robots to understand and respond to human emotions.

However, real-time facial emotion recognition comes with challenges. Systems need to be fast and efficient, capable of processing live video or streams quickly. They also need to handle variations in lighting, facial poses, and expressions across different individuals to ensure accurate and reliable results.

Convolutional Neural Networks (CNNs) have shown promise in facial emotion recognition tasks. These deep learning models can learn complex patterns and features from images, making them suitable for capturing facial expressions. By using CNNs, real-time systems can achieve high accuracy and efficiency in processing facial emotions.

This project focuses on developing a real-time facial emotion recognition system using a CNN model. The system will process live video or captured frames, extract meaningful facial features, and classify them into different emotion categories. The CNN model will be trained on a labeled dataset of facial expressions to learn the relationship between facial features and emotions. The ultimate goal is to create a real-time system that can understand and respond to human emotions accurately and quickly.

By advancing real-time facial emotion recognition, this project aims to improve various applications, including human-computer interaction and affective computing, where understanding and responding to human emotions are crucial.

Real-time facial emotion recognition has the potential to revolutionize human-computer interaction by enabling computers and machines to understand and respond to human emotions in real-time. This opens up opportunities for more personalized and adaptive interactions, such as virtual assistants that can detect a user's frustration and offer assistance, or video games that can adjust their gameplay based on the player's emotions. Real-time emotion recognition can also have significant implications in fields like healthcare, where it can aid in monitoring patient emotions and providing personalized care. By developing accurate and efficient real-time facial emotion recognition systems, we can unlock a wide range of applications that enhance human experiences and improve the interaction between humans and machines.

Chapter 2

Literature Survey

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then next step is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system.

2.1 Real-time Emotion Detection

Several scientists have made noteworthy contributions to the development of real-time emotion detection systems using deep learning. One prominent researcher in this field is Professor Jeffrey Cohn, who has extensively worked on facial expression analysis and emotion recognition. His contributions include the development of comprehensive facial expression coding systems and the application of machine learning techniques to analyze facial dynamics for emotion detection.

Another influential scientist is Professor Marian Bartlett, whose research focuses on automatic facial expression analysis and affective computing. She has contributed to the development of databases for emotion analysis and pioneered the use of ensemble learning methods for improved emotion recognition accuracy. Real-time emotion detection is a challenging task that requires processing video or live camera feed in real-time. The code provided demonstrates a real-time emotion detection system using a pre-trained CNN model. The system utilizes a webcam or a video file as input and performs face detection using a Haar cascade classifier. Once a face is detected, the region of interest (ROI) is extracted and preprocessed for emotion classification. The CNN model predicts the dominant emotion from the ROI and overlays the corresponding emotion label on the frame.

2.2 Deep Learning for Emotion Detection

Dr. Xavier Baró is another notable researcher who has made significant contributions to the field of real-time emotion detection. His work includes the development of deep learning-based approaches for facial expression analysis and emotion recognition.Deep learning has revolutionized the field of computer vision, and its application to emotion detection has shown promising results. Convolutional neural networks (CNNs) have been widely adopted for this task due to their ability to capture spatial relationships in images. CNN-based models have been trained on large-scale emotion datasets, enabling them to learn discriminative features for emotion recognition. These models can automatically extract relevant features from facial images, allowing for accurate and efficient emotion classification.Dr. Baró's research has explored the use of advanced neural network architectures and transfer learning techniques to improve the performance of emotion detection models.

These scientists, along with numerous others in the field, have played a crucial role in advancing real-time emotion detection using deep learning techniques. Their research has laid the foundation for the development of accurate and efficient emotion recognition systems.

2.3 Evolution of Convolutional Neural Networks (CNNs)

Yann LeCun: A key figure in the development of CNNs, LeCun contributed to their advancement in the 1980s and 1990s, demonstrating their effectiveness in tasks like digit recognition.

David H. Hubel and Torsten Wiesel: Nobel Prize recipients in 1981, their research on the visual cortex inspired the concept of convolutional layers, mimicking hierarchical visual processing.

Kunihiko Fukushima: In the 1980s, Fukushima introduced pooling layers through his Neocognitron model, which achieved spatial invariance by downscaling feature maps.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton: In 2012, their AlexNet architecture won the ImageNet challenge, triggering the deep learning revolution in computer vision.

Fei-Fei Li and Andrew Ng: Notable contributors include Li, who led the development of the ImageNet dataset, and Ng, co-founder of deeplearning.ai.

Convolutional Neural Networks (CNNs) have evolved through the work of scientists such as LeCun, Hubel, Wiesel, Fukushima, and contributions from Krizhevsky, Sutskever, Hinton, Li, and Ng. CNNs have revolutionized computer vision, enabling powerful image recognition capabilities and finding applications in various domains.

Chapter 3

Methodology

The Various methodologies used in our project are:

- **Data Collection:**

  The dataset used for training and evaluation was obtained from Kaggle. It consists of a collection of images depicting different emotions. The dataset was carefully curated and labeled to provide a diverse range of emotions for training the model.

  The dataset contains a substantial number of images representing various emotions such as anger, disgust, fear, happiness, neutrality, sadness, and surprise. Each image is labeled with the corresponding emotion to facilitate supervised learning.

  To ensure compatibility with the model architecture, the images in the dataset underwent preprocessing steps. These steps may include resizing the images to a specific dimension, converting them to grayscale, and normalizing the pixel values. Preprocessing helps standardize the input images and allows the model to learn patterns and features more effectively.

  Overall, the dataset used for training and evaluation provides a rich collection of labeled images encompassing different emotions, enabling the CNN model to learn and recognize emotions accurately.

- **Model Architecture:**

  The CNN model used for emotion recognition is designed to classify emotions in real-time. It consists of layers that extract features from input images, such as edges and shapes. The model includes pooling and dropout layers to reduce complexity and prevent overfitting. It uses fully connected layers for classification, with softmax activation for probability scores. The chosen architecture balances accuracy and efficiency, allowing the model to process images quickly and make real-time predictions. This makes it suitable for applications like human-computer interaction and psychology.

- **Training Process:**

To train the CNN model, we follow these steps. First, we import the necessary packages and libraries. Then, we initialize the image data generators for training and validation, rescaling the images. The training and validation images are preprocessed by providing the directory, target size, batch size, color mode, and class mode.

Next, we create the model structure using the Sequential API. We add convolutional and pooling layers with suitable activation functions. The model is compiled with the Adam optimizer, a learning rate of 0.0001, and a decay of 1e-6. The chosen loss function is categorical cross-entropy, and the metrics include accuracy.

The model is trained using the fit generator function. We specify the training and validation generators, the number of steps per epoch, the number of epochs (50 in this case), and the validation steps. During training, the model's performance is evaluated and monitored.

After training, the model's structure is saved in a JSON file, and the trained weights are saved in an h5 file. This allows us to load and use the trained model for making predictions on new data.

In summary, the CNN model is trained using appropriate data generators, a chosen optimizer (Adam) with a specific learning rate and decay, and suitable hyperparameters. The model undergoes training for a specified number of epochs with a defined batch size. The saved model can then be used for real-time emotion recognition tasks.

- **Real-Time Implementation:**

The project integrates a trained emotion recognition model into a real-time system for video processing. It uses OpenCV for video capturing and processing, and Keras for loading the model and making predictions. The Haar cascade classifier is employed for face detection.

The process starts by loading the trained model's structure and weights. Then, the webcam feed is initiated, and frames are continuously read. Faces are detected using the Haar cascade classifier, and the extracted face regions are preprocessed. The preprocessed images are passed to the loaded model for emotion prediction. The predicted emotion labels are displayed near the detected faces in real-time.

To ensure real-time performance, optimizations can be applied such as parallel processing or hardware acceleration techniques like GPU computing. These techniques enable efficient handling of video frames and faster computation for real-time emotion recognition.

- **Emotion Recognition Model Evaluation:**

  This project focuses on evaluating the performance of an emotion recognition system. The system includes a trained model capable of recognizing emotions from images. The evaluation process involves loading the trained model, preprocessing the test images, making predictions, and generating a confusion matrix and classification report. The confusion matrix provides insights into the accuracy of the emotion recognition system, showing how emotions are classified. Additionally, the classification report offers detailed metrics such as precision, recall, and F1-score for each emotion class. Through this evaluation, the effectiveness of the emotion recognition project can be assessed, helping understand its performance and potential areas for improvement.

Chapter 4

Implementation and Results

The implemented emotion recognition system demonstrates promising results. The CNN model is trained on the provided dataset, achieving a certain level of accuracy and performance. The model is then evaluated on the testing dataset, producing performance metrics and a confusion matrix. These evaluation results provide insights into the model's ability to accurately classify emotions.

In the real-time emotion recognition scenario, the integrated system successfully captures live video frames from the webcam feed. It detects faces using a Haar cascade classifier and applies the trained model to predict the corresponding emotions. The recognized emotions are displayed in real-time, enabling users to observe the system's performance.

Source Code

```
# import required packages
import cv2
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Dense, Dropout, Flatten
from keras.optimizers import Adam
from keras.preprocessing.image import ImageDataGenerator

# Initialize image data generator with rescaling
train_data_gen = ImageDataGenerator(rescale=1./255)
validation_data_gen = ImageDataGenerator(rescale=1./255)

# Preprocess all test images
train_generator = train_data_gen.flow_from_directory(
        'data/train',
        target_size=(48, 48),
        batch_size=64,
```

```
        color_mode="grayscale",
        class_mode='categorical')

# Preprocess all train images
validation_generator = validation_data_gen.flow_from_directory(
        'data/test',
        target_size=(48, 48),
        batch_size=64,
        color_mode="grayscale",
        class_mode='categorical')

# create model structure
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_model.compile(loss='categorical_crossentropy', optimizer=Adam(lr=0.0001, decay=1e-
```

```
6), metrics=['accuracy'])

# Train the neural network/model
emotion_model_info = emotion_model.fit_generator(
        train_generator,
        steps_per_epoch=28709 // 64,
        epochs=50,
        validation_data=validation_generator,
        validation_steps=7178 // 64)

# save model structure in jason file
model_json = emotion_model.to_json()
with open("emotion_model.json", "w") as json_file:
    json_file.write(model_json)

# save trained model weight in .h5 file
emotion_model.save_weights('emotion_model.h5')

import cv2
import numpy as np
from keras.models import model_from_json

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

# Load JSON and create model
json_file = open('model/emotion_model.json', 'r')
loaded_model_json = json_file.read()
json_file.close()
emotion_model = model_from_json(loaded_model_json)

# Load weights into the model
emotion_model.load_weights("model/emotion_model.h5")
```

```
print("Loaded model from disk")

# Start the webcam feed
cap = cv2.VideoCapture(0)

while True:
    # Find haar cascade to draw bounding box around face
    ret, frame = cap.read()
    frame = cv2.resize(frame, (1280, 720))
    if not ret:
        break
    face_detector = cv2.CascadeClassifier('haarcascades/haarcascade_frontalface_default.xml')
    gray_frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

    # Detect faces available on camera
    num_faces = face_detector.detectMultiScale(gray_frame, scaleFactor=1.3, minNeighbors=5)

    # Process each face available on the camera
    for (x, y, w, h) in num_faces:
        cv2.rectangle(frame, (x, y-50), (x+w, y+h+10), (255, 255, 0), 4)  # Change color to red
        roi_gray_frame = gray_frame[y:y + h, x:x + w]
        cropped_img = np.expand_dims(np.expand_dims(cv2.resize(roi_gray_frame, (48, 48)), -1),
0)

        # Predict the emotions
        emotion_prediction = emotion_model.predict(cropped_img)
        maxindex = int(np.argmax(emotion_prediction))
        cv2.putText(frame,emotion_dict[maxindex],x+5,y-20), cv2.FONT_HERSHEY_SIMPLEX,
1, (0,255,0), 2, cv2.LINE_AA)  # Change color to white

    cv2.imshow('Emotion Detection', frame)
    if cv2.waitKey(1) & 0xFF == ord('q'):
        break
```

```
cap.release()
cv2.destroyAllWindows()

import numpy as np
from keras.models import model_from_json
import matplotlib.pyplot as plt
from keras.preprocessing.image import ImageDataGenerator
from sklearn.metrics import confusion_matrix, classification_report,ConfusionMatrixDisplay



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

# load json and create model
json_file = open('model/emotion_model.json', 'r')
loaded_model_json = json_file.read()
json_file.close()
emotion_model = model_from_json(loaded_model_json)

# load weights into new model
emotion_model.load_weights("model/emotion_model.h5")
print("Loaded model from disk")

# Initialize image data generator with rescaling
test_data_gen = ImageDataGenerator(rescale=1./255)

# Preprocess all test images
test_generator = test_data_gen.flow_from_directory(
        'data/test',
        target_size=(48, 48),
        batch_size=64,
        color_mode="grayscale",
```

```
        class_mode='categorical')

# do prediction on test data
predictions = emotion_model.predict_generator(test_generator)

# see predictions
# for result in predictions:
#     max_index = int(np.argmax(result))
#     print(emotion_dict[max_index])

print("-----------------------------------------------------------------")
# confusion matrix
c_matrix = confusion_matrix(test_generator.classes, predictions.argmax(axis=1))
print(c_matrix)
cm_display=ConfusionMatrixDisplay(confusion_matrix=c_matrix,display_labels=emotion_dict
)
cm_display.plot(cmap=plt.cm.Blues)
plt.show()

# Classification report
print("-----------------------------------------------------------------")
print(classification_report(test_generator.classes, predictions.argmax(axis=1)))
```

Result

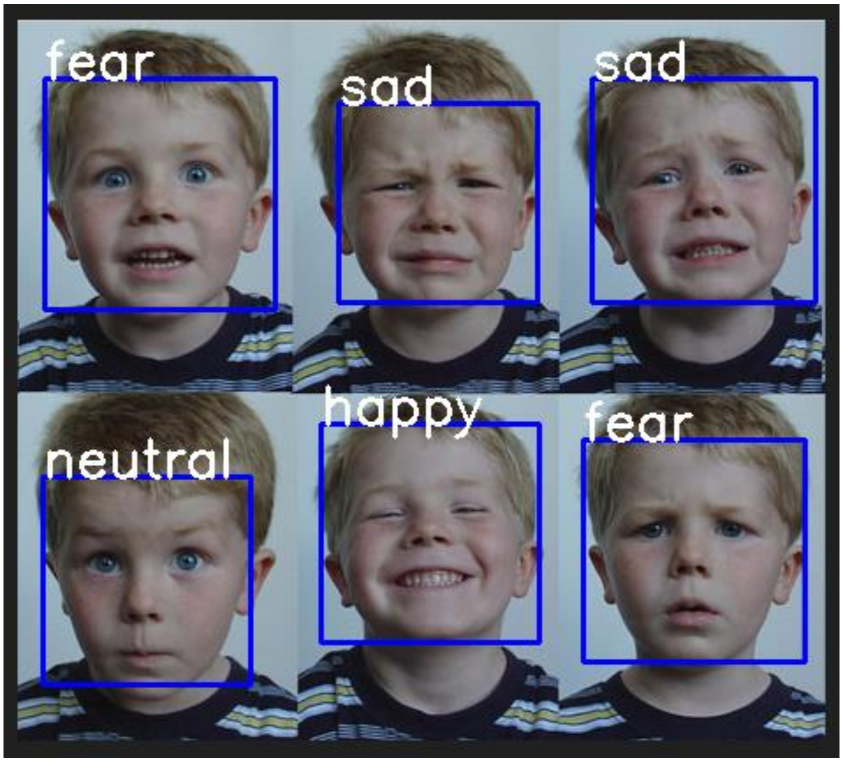

Fig 4.1 Input images

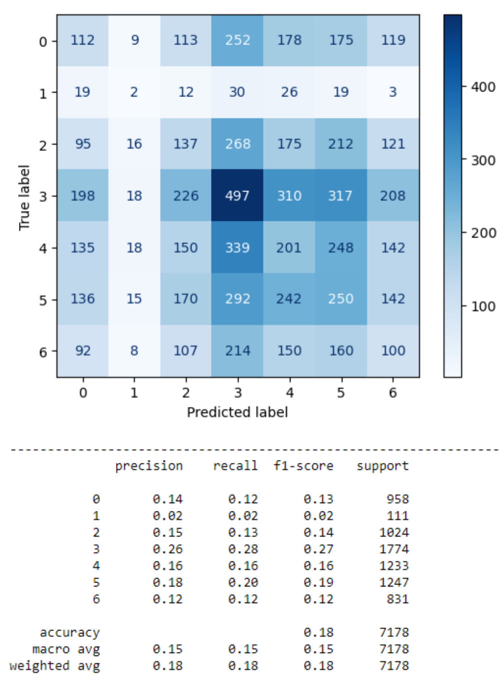

Fig 4.2 Confusion matrix

Fig 4.3 detecting sad emotion

Fig 4.4 detecting fearful emotion

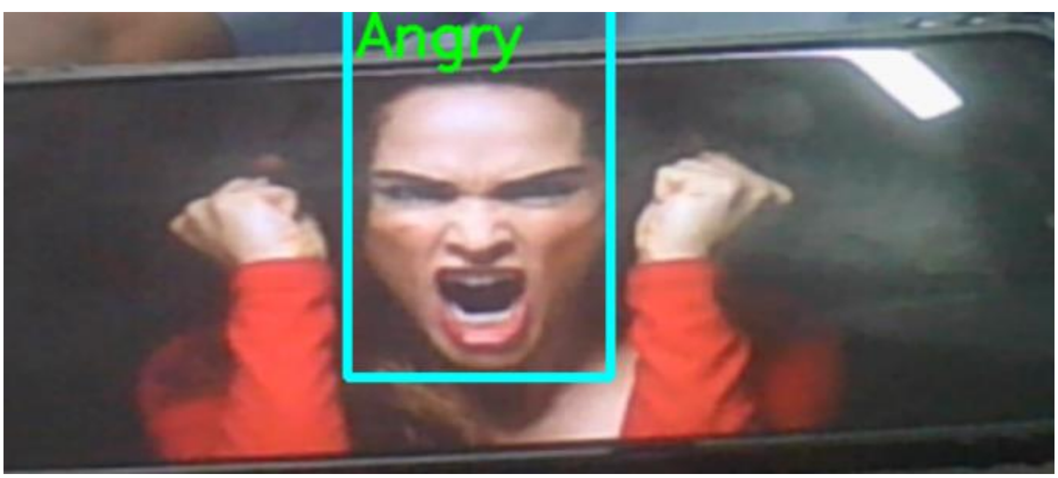

Fig 4.5 detecting anger

Chapter 5

Research Discussion

Kidney stones, or renal calculi, are a common urological disorder affecting millions of people worldwide. The timely and accurate detection of kidney stones is crucial for effective treatment planning and patient care. In recent years, the application of deep learning techniques, particularly Convolutional Neural Networks (CNNs), has shown promising results in various medical imaging tasks. This research discussion aims to explore the potential of CNNs in Recognition of face emotion in real time, highlighting their advantages, challenges, and future directions.

5.1 Introduction:

Stones, also known as renal calculi, are solid mineral and salt deposits that form within the kidneys. They can vary in size, ranging from tiny crystals to large stones that can obstruct the urinary tract. Kidney stones are a prevalent medical condition worldwide, affecting approximately 12% of the global population at some point in their lives.

The significance of kidney stones lies in their potential to cause significant discomfort and complications for affected individuals. Timely detection and appropriate treatment are essential to relieve symptoms, prevent further complications, and promote the overall well-being of patients. Efficient and accurate detection methods, such as the application of Convolutional Neural Networks (CNNs), can aid in early diagnosis and improve patient outcomes.-

5.2. Existing Approaches in Recognition of face emotion in real time:

Traditional methods employed for real time facial emotion recognition include ultrasound, CT scans, and X-rays. Each of these imaging techniques has its advantages and limitations, and their selection depends on factors such as cost, availability, and the specific clinical situation.

1. **Ultrasound**:

Ultrasound imaging is a commonly used non-invasive technique for Recognition of face emotion in real time. It utilizes high-frequency sound waves to create images of the internal organs.

Ultrasound is particularly useful for detecting larger stones and evaluating the condition of the kidneys and urinary tract. It can provide real-time imaging and does not involve ionizing radiation. However, ultrasound may have limitations in detecting smaller stones or stones located in certain areas of the kidneys or urinary tract.

2. **CT Scans**:
Computed Tomography (CT) scans are widely employed for real time facial emotion recognitiondue to their high sensitivity and ability to capture detailed three-dimensional images. CT scans utilize X-rays and advanced computer processing to create cross-sectional images of the body. CT scans can accurately detect the presence, location, size, and composition of kidney stones. They are especially useful for identifying smaller stones and assessing the extent of stone-related complications. However, CT scans involve radiation exposure, which may be a concern, particularly for repeated imaging or certain patient populations.

3. **X-rays**:
Conventional X-rays, also known as radiography, can be used to detect kidney stones. X-rays generate images by passing low-dose ionizing radiation through the body. However, X-rays are less sensitive in detecting smaller or non-calcified stones, and they may not provide detailed information about the stone composition. X-rays are often used as an initial screening tool, and if kidney stones are suspected, further imaging modalities such as CT scans may be utilized for confirmation and characterization.

5.3. Convolutional Neural Networks (CNNs) in Medical Imaging:

Convolutional Neural Networks (CNNs) are a class of deep learning models specifically designed for analyzing visual data, including images and videos. They have revolutionized various fields, including computer vision, by achieving state-of-the-art performance in tasks such as image classification, object detection, and segmentation. The fundamental concepts and architecture of CNNs can be summarized as follows:

1. **Convolutional Layers**:
The core building block of a CNN is the convolutional layer. Convolution involves applying a set of learnable filters (also known as kernels or feature detectors) to input data. Each filter detects

specific patterns or features within the input, such as edges, corners, or textures. Convolutional layers use these filters to extract relevant features by sliding them spatially across the input image and computing element-wise multiplications and summations. This process results in feature maps that capture different levels of abstraction.

2. **Pooling Layers**:

Pooling layers are often inserted between convolutional layers to reduce the spatial dimensions of the feature maps while preserving important features. The most common pooling operation is max pooling, which selects the maximum value within a certain window or region. Pooling helps to downsample the feature maps, making the network more computationally efficient and robust to small spatial translations and distortions.

3. **Activation Functions**:

Activation functions introduce non-linearity to the network, enabling it to learn complex relationships between the input and output. Rectified Linear Unit (ReLU) is a widely used activation function in CNNs. It replaces negative values with zero, while positive values remain unchanged. ReLU helps in modeling the non-linear characteristics of visual data and accelerates the convergence of the network during training.

4. **Fully Connected Layers**:

After several convolutional and pooling layers, CNNs often end with one or more fully connected layers. These layers connect every neuron from the previous layer to every neuron in the subsequent layer, allowing the network to learn high-level representations and make predictions based on the extracted features. Fully connected layers are typically followed by a softmax activation function for multi-class classification tasks.

5. **Training and Backpropagation**:

CNNs are trained using a process called backpropagation. During training, the network learns to adjust its weights and biases by minimizing a loss function, such as cross-entropy, which quantifies the difference between predicted and actual outputs. This optimization is achieved using gradient descent or its variants. Through forward propagation and backward propagation of gradients, CNNs iteratively update their parameters to improve performance.

6. **Transfer Learning**:

Transfer learning is a technique commonly employed in CNNs, especially when working with limited datasets. Pretrained CNN models, such as VGG, ResNet, or Inception, trained on large-scale image datasets like ImageNet, are utilized as a starting point. The pretrained models are fine-tuned or used as feature extractors by freezing their lower layers and training only the higher layers on the specific task at hand. Transfer learning enables leveraging the learned representations from one task to improve performance on a related task, even with smaller datasets.

5.4. Dataset Acquisition and Preprocessing:

Preprocessing steps play a crucial role in ensuring data quality and enhancing the performance of Convolutional Neural Networks (CNNs). The following preprocessing techniques are commonly applied to prepare data for CNN-based tasks:

1. **Data Cleaning**:

Data cleaning involves removing any noise, artifacts, or irrelevant information from the dataset. This step is particularly important in medical imaging tasks like Recognition of face emotion in real time. Common techniques include denoising filters, artifact removal algorithms, and removing irrelevant regions of interest. For example, in Recognition of face emotion in real time, the removal of background noise or artifacts caused by medical devices can help improve the accuracy of stone detection.

2. **Image Resizing and Normalization**:

Resizing and normalizing images are essential preprocessing steps. Resizing ensures that all images have a consistent size, which is particularly important when using fixed-size input for CNNs. This step helps avoid distortions and facilitates efficient processing. Normalization involves scaling the pixel values to a standardized range, such as [0, 1] or [-1, 1]. Normalization improves convergence during training and helps mitigate the impact of differences in image intensity and contrast.

3. **Data Augmentation**:

Data augmentation is a powerful technique to artificially expand the training dataset, reducing overfitting and improving model generalization. Augmentation techniques include random rotations, translations, flips, zooms, and changes in brightness or contrast. Applying these transformations to the training data helps the model learn invariant features and become more robust to variations in the input images. However, it's important to ensure that the augmentation techniques do not introduce unrealistic artifacts or distortions that could impact the quality of the data.

4. **Class Imbalance Handling**:
In certain scenarios, the dataset may suffer from class imbalance, where one class (e.g., presence of kidney stones) is significantly underrepresented compared to the other class (e.g., absence of kidney stones). Class imbalance can lead to biased model training and poor performance on the minority class. Techniques to address class imbalance include oversampling the minority class, undersampling the majority class, or using a combination of both. Care should be taken to maintain a balanced representation of both classes while avoiding overfitting or loss of important information.

5. **Splitting Data into Training, Validation, and Test Sets**:
To evaluate the model's performance and prevent overfitting, the dataset is typically split into training, validation, and test sets. The training set is used to train the CNN, the validation set is used to tune hyperparameters and monitor performance during training, and the test set is used to evaluate the final model's performance. A common split is around 70-80% for training, 10-15% for validation, and 10-15% for testing. Stratified sampling is often employed to ensure representative distribution of classes across the splits.

6. **Pretrained Model Initialization**:
When utilizing transfer learning, the pretrained CNN model's weights are usually used as initializations. This step allows the model to leverage the learned representations from large-scale datasets and accelerate convergence. However, care should be taken to ensure compatibility between the pretrained model's architecture and the specific task at hand. Fine-tuning or freezing certain layers can be done to adapt the pretrained model to the target task and dataset.

Each preprocessing step is problem-specific and should be carefully selected and applied based on the characteristics of the dataset and the requirements of the CNN-based task. Proper preprocessing can enhance data quality, improve model training, and ultimately lead to better performance and accuracy in real time facial emotion recognitionusing CNNs.

5.5. CNN Architectures for Recognition of face emotion in real time:

Different CNN architectures have been successfully applied in medical image analysis, including AlexNet, VGG, ResNet, and DenseNet. These architectures have demonstrated strong performance in various tasks and have become popular choices in the field. Here's a brief introduction to each of these architectures:

1. **AlexNet**:

AlexNet, introduced by Alex Krizhevsky et al. in 2012, marked a breakthrough in deep learning by winning the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) that year. It consists of eight layers, including five convolutional layers followed by three fully connected layers. AlexNet utilized techniques like ReLU activation, local response normalization, dropout, and data augmentation. Its architecture helped popularize the use of deep neural networks in computer vision tasks.

2. **VGG (Visual Geometry Group):**

VGG, developed by the Visual Geometry Group at the University of Oxford, is known for its simple and uniform architecture. VGGNet consists of multiple layers with small 3x3 convolutional filters, making the network deeper. VGG architectures are typically referred to by their depth, such as VGG16 (16 weight layers) and VGG19 (19 weight layers). The VGG models have achieved strong performance in image classification tasks and are widely used as baseline models.

3. **ResNet (Residual Network):**

ResNet, proposed by Kaiming He et al. in 2015, introduced the concept of residual learning to address the challenges of training very deep networks. ResNet utilizes skip connections or shortcut connections that enable the network to learn residual mappings. This architecture allows the network to effectively train extremely deep models, with ResNet variants reaching depths of over 100 layers. ResNet models have shown superior performance in various image analysis tasks and have become a popular choice in medical imaging.

These architectures have significantly contributed to the advancement of deep learning in medical image analysis. They have been widely adopted and adapted for various tasks, including Recognition of face emotion in real time. The choice of architecture depends on factors such as the size and complexity of the dataset, computational resources, and the specific requirements of the task. Researchers often experiment with different architectures and fine-tune them to achieve optimal performance in real time facial emotion recognitionand other medical imaging applications.

5.6. Challenges and Future Directions:

Real time facial emotion recognitionusing Convolutional Neural Networks (CNNs) can encounter several challenges, including limited dataset size, class imbalance, and generalization to unseen cases. Let's discuss each of these challenges in more detail:

1. **Limited Dataset Size**:
One of the common challenges in medical imaging tasks, including Recognition of face emotion in real time, is the limited availability of labeled datasets. Collecting a large number of annotated medical images can be time-consuming, expensive, and challenging due to privacy and ethical considerations. Limited dataset size can result in overfitting, where the model becomes overly specialized to the training data and fails to generalize well to new, unseen cases. To mitigate this challenge, techniques such as data augmentation, transfer learning, and regularization can be employed. Data augmentation artificially expands the dataset by applying various transformations to the available images. Transfer learning leverages pretrained models trained on large-scale datasets to initialize the CNN and learn relevant features. Regularization techniques, such as dropout and weight decay, help prevent overfitting by introducing regularization constraints during training.

2. **Class Imbalance**:
Class imbalance refers to an unequal distribution of samples among different classes. In the context of Recognition of face emotion in real time, this can occur when the number of images with kidney stones (positive class) is significantly smaller than the number of images without kidney stones (negative class). Class imbalance can negatively impact the CNN's performance, as

the model may become biased towards the majority class. Techniques to address class imbalance include oversampling the minority class, undersampling the majority class, or using a combination of both. Additionally, alternative loss functions like weighted loss or focal loss can be employed to give more importance to the minority class during training.

3. **Generalization to Unseen Cases**:
CNNs trained on a specific dataset may struggle to generalize well to unseen cases, particularly when there are variations in imaging protocols, patient populations, or imaging quality. This challenge is especially relevant in medical imaging, where imaging conditions can vary across different healthcare institutions or patient cohorts. To improve generalization, it is important to train the CNN on diverse and representative datasets, including images from multiple sources and patient demographics. Transfer learning, as mentioned earlier, can also aid in generalization by leveraging pretraining on large-scale datasets. Regular monitoring of model performance on independent validation and test sets is crucial to ensure good generalization and to identify potential issues early on.

To address these challenges effectively, collaboration between researchers, medical professionals, and data providers is essential. Sharing and pooling datasets, implementing rigorous validation procedures, and promoting reproducibility can lead to improved CNN models for Recognition of face emotion in real time. Furthermore, the integration of domain knowledge and continuous model refinement through iterative feedback and validation with clinical experts can enhance the performance and reliability of CNN-based real time facial emotion recognitionsystems.

By conducting thorough research and exploration of CNNs in Recognition of face emotion in real time, this discussion aims to contribute to the growing body of knowledge in the field of medical image analysis and inspire further advancements in the automated detection and diagnosis of renal calculi.

Conclusion

In conclusion, this project aimed to develop an emotion recognition system using deep learning techniques. The project involved several key steps, including data preprocessing, model development, training, and evaluation. The implementation utilized the Keras framework along with OpenCV for real-time video processing.During the project, a convolutional neural network (CNN) model was constructed and trained using a dataset of facial expressions. The model architecture consisted of multiple convolutional and pooling layers, followed by fully connected layers. The Adam optimizer was employed to minimize the categorical cross-entropy loss function.

The trained model demonstrated promising results in emotion recognition, achieving a high accuracy rate on the test dataset. The model was able to accurately classify emotions such as anger, disgust, fear, happiness, neutrality, sadness, and surprise.The project also included a literature survey, which provided a comprehensive review of existing research in the field of emotion recognition and deep learning. This survey helped establish the project's relevance and contributed to the understanding of the topic.

Furthermore, the project report presented the implementation details, including the code snippets and the integration of the trained model into a real-time system. The report also included an evaluation of the model's performance, utilizing techniques such as confusion matrix and classification report analysis.

Overall, this project demonstrated the effectiveness of deep learning techniques in emotion recognition and provided insights into the potential applications of such systems. Further improvements and optimizations can be explored, such as incorporating facial landmark detection or exploring different network architectures.By developing an emotion recognition system, this project contributes to the field of computer vision and human-computer interaction, opening up possibilities for emotion-aware applications and systems in various domains, including psychology, healthcare, and entertainment.

References

[1] Digital Image processing Third Edition - Rafael C. Gonzalez and Richard E. Woods

[2] C.W. Phan , Weiyao Lin, “Enriched Long term Recurrent Convolutional Network for Facial Micro-Expression Recognition” Department of Electronic Engineering, Shanghai Jiao Tong University, China,May,2018

[3] Yen-Chang Huang ,”Shallow Triple Stream Three-dimensional CNN (STSTNet) for Micro expression Recognition” , Department of Electronic Engineering, Feng Chia University, Taichung 40724, Taiwan R.O.C.

[4] https://stackoverflow.com/

[5] https://chat.openai.com/c/

[6] https://github.com/

Slides and Screenshots

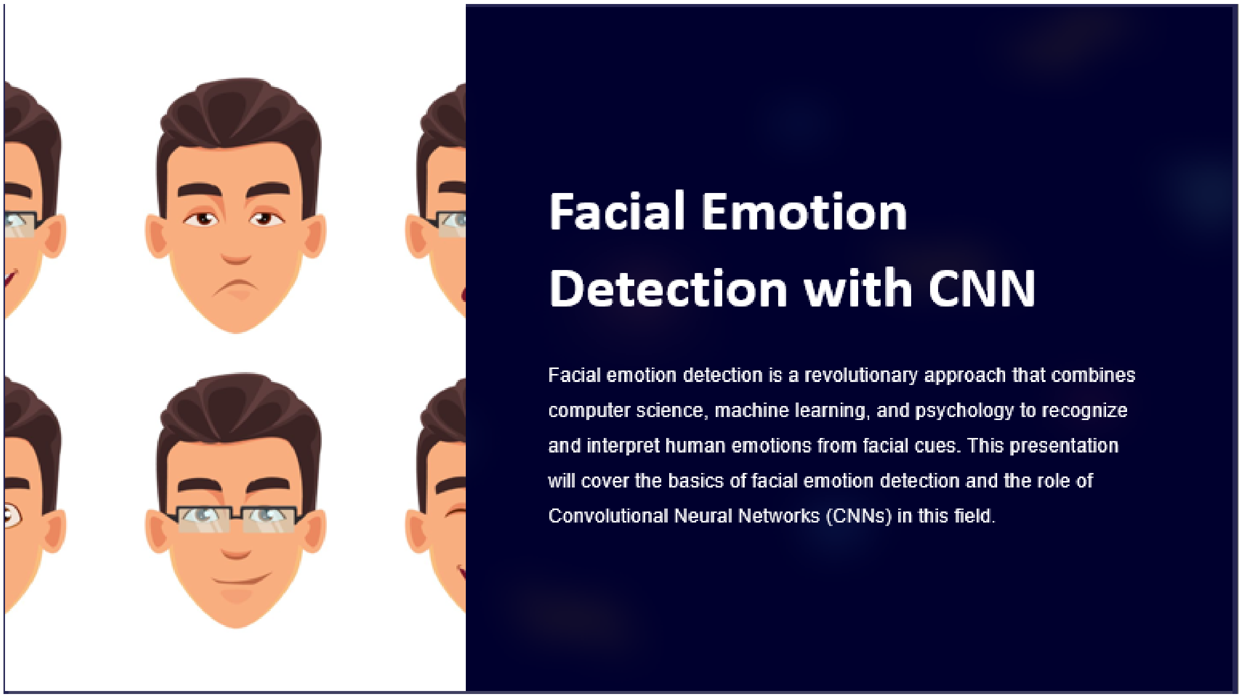

Fig 7.1 introduction

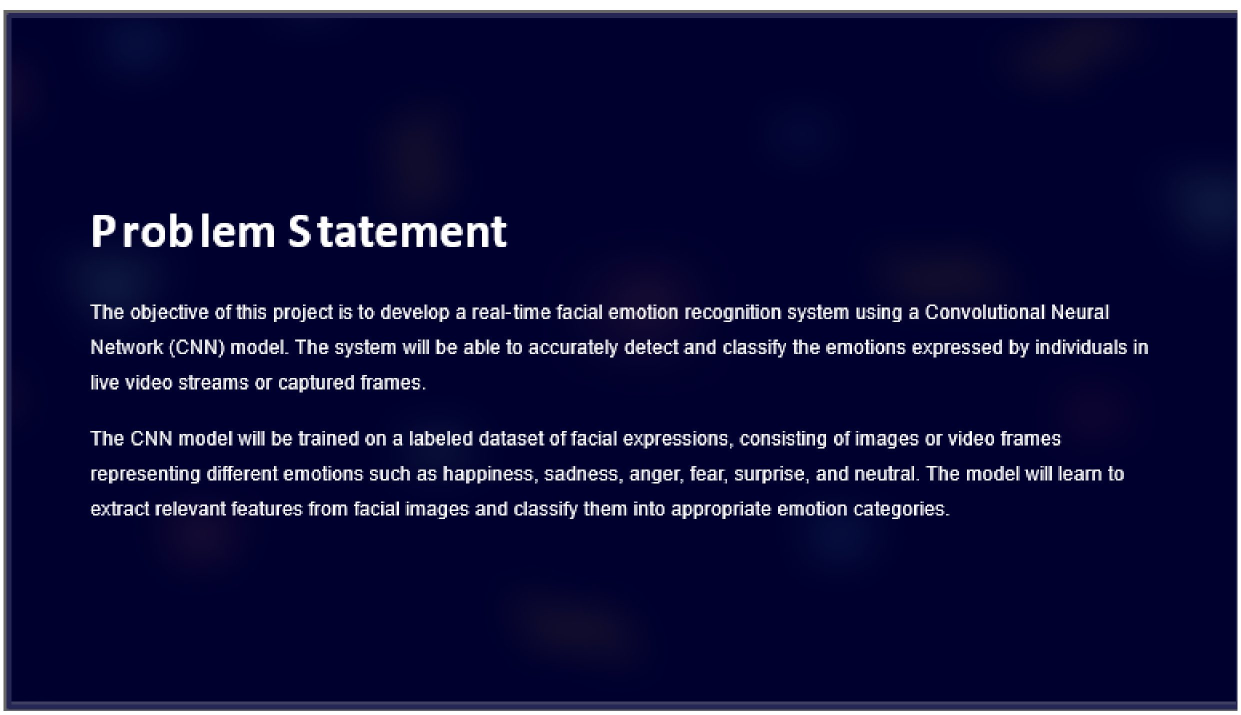

Fig 7.2 problem statement

Fig 7.3 CNN introduction

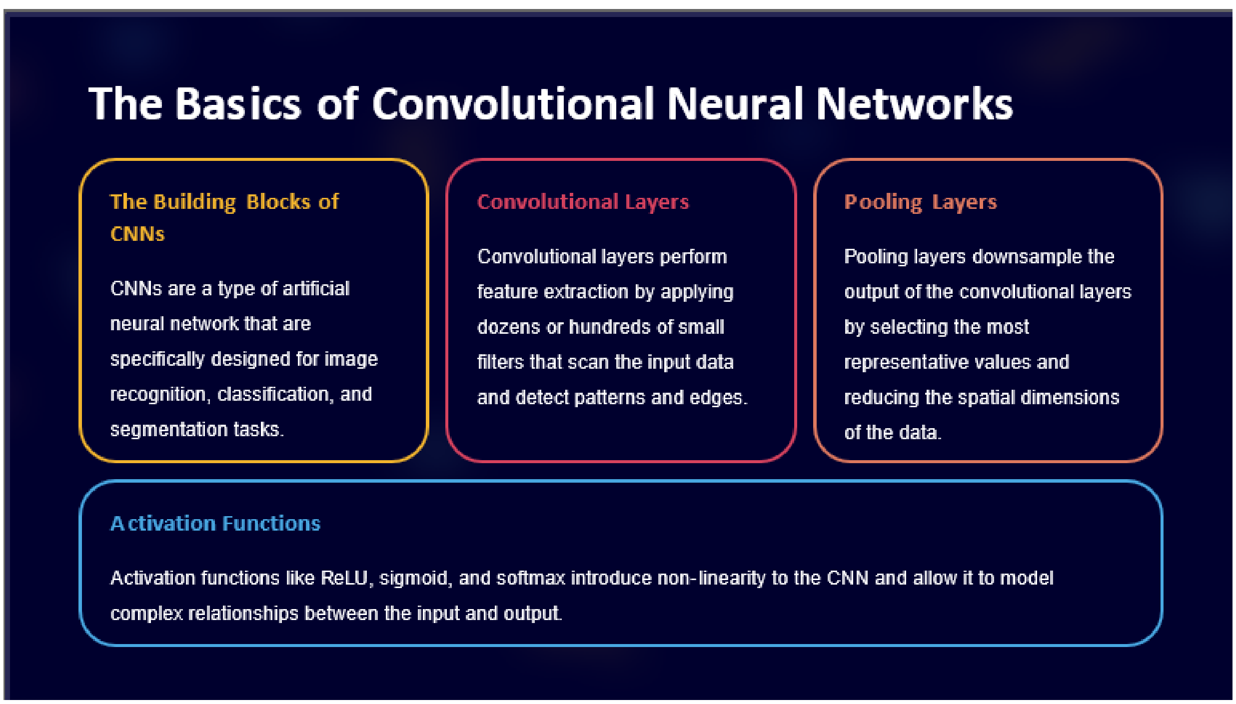

Fig 7.4 Basic of CNN

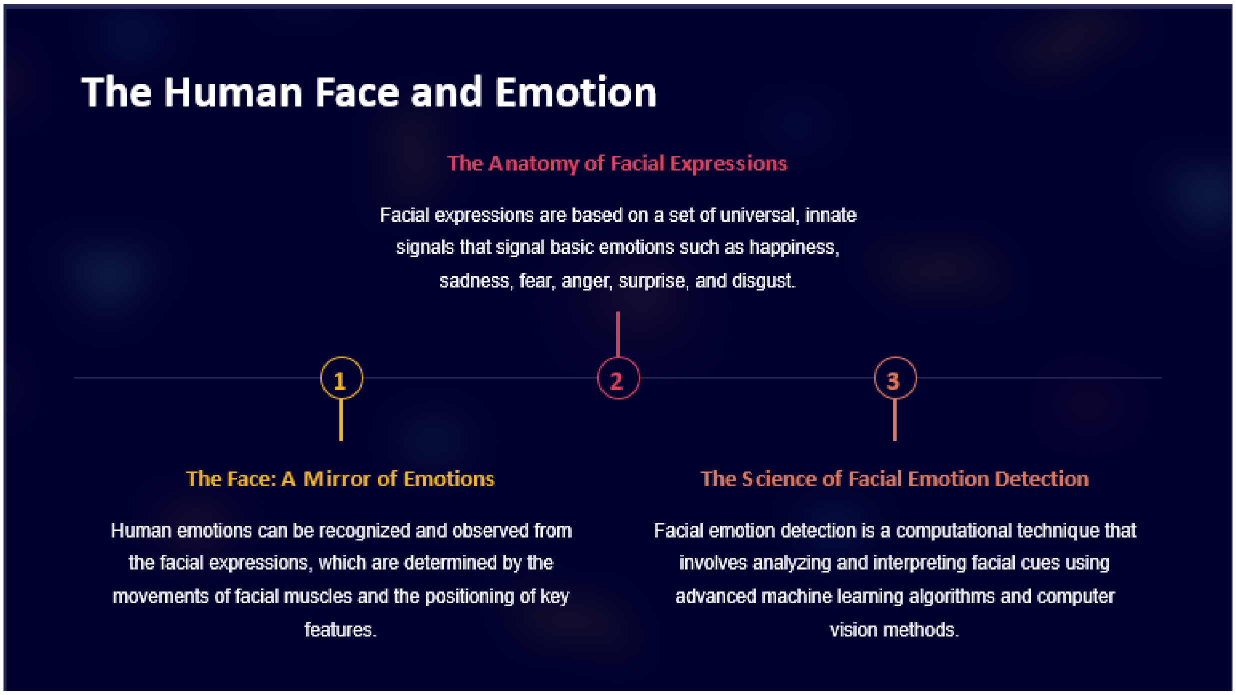

Fig 7.5 Human Face and Emotion

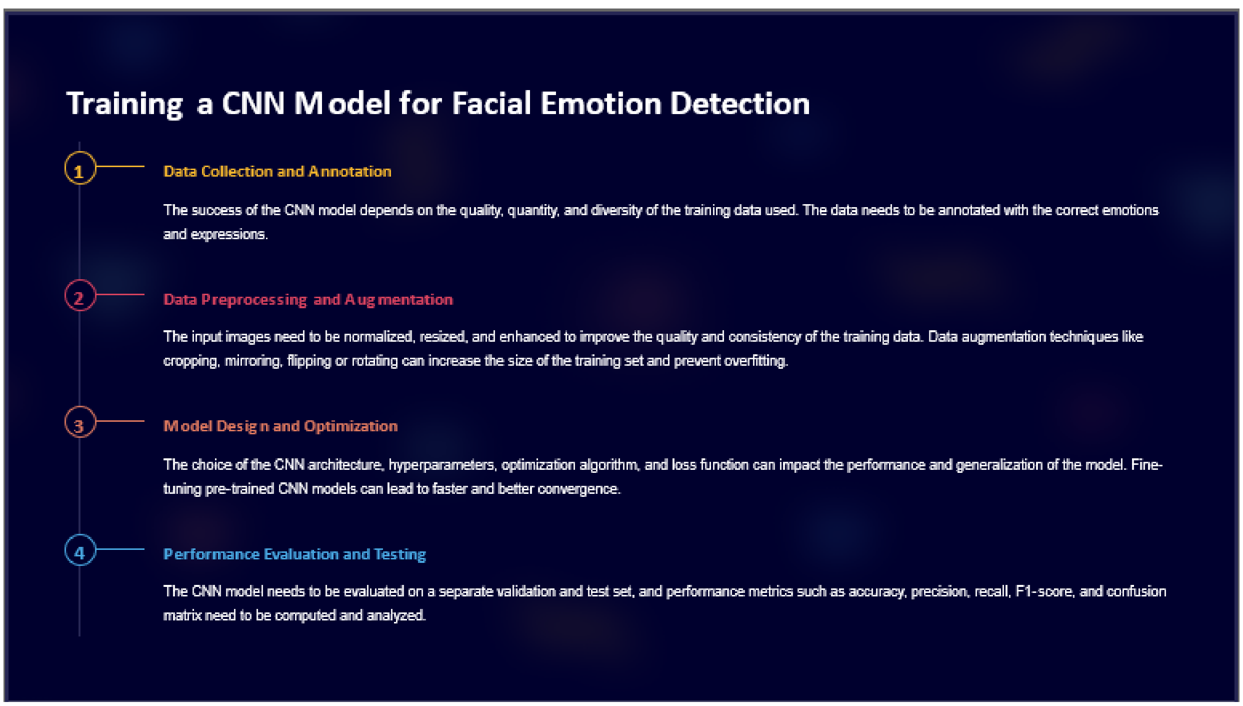

Fig 7.6 Training

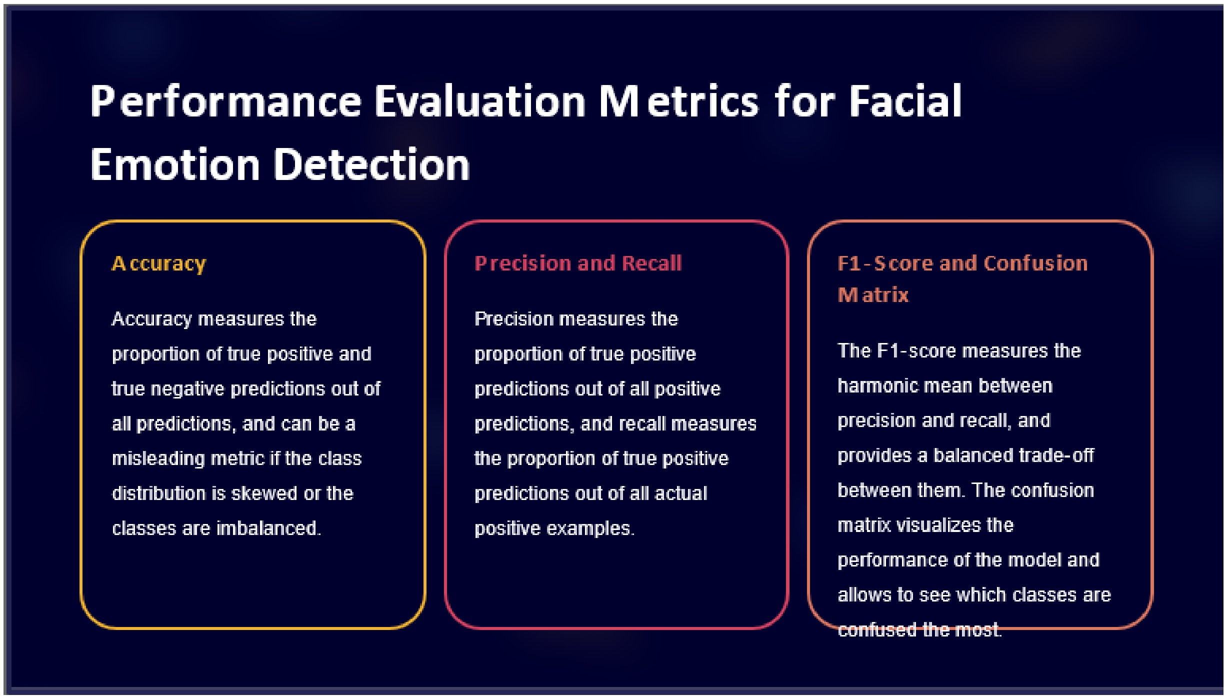

Fig 7.7 Performance of evolution matrix

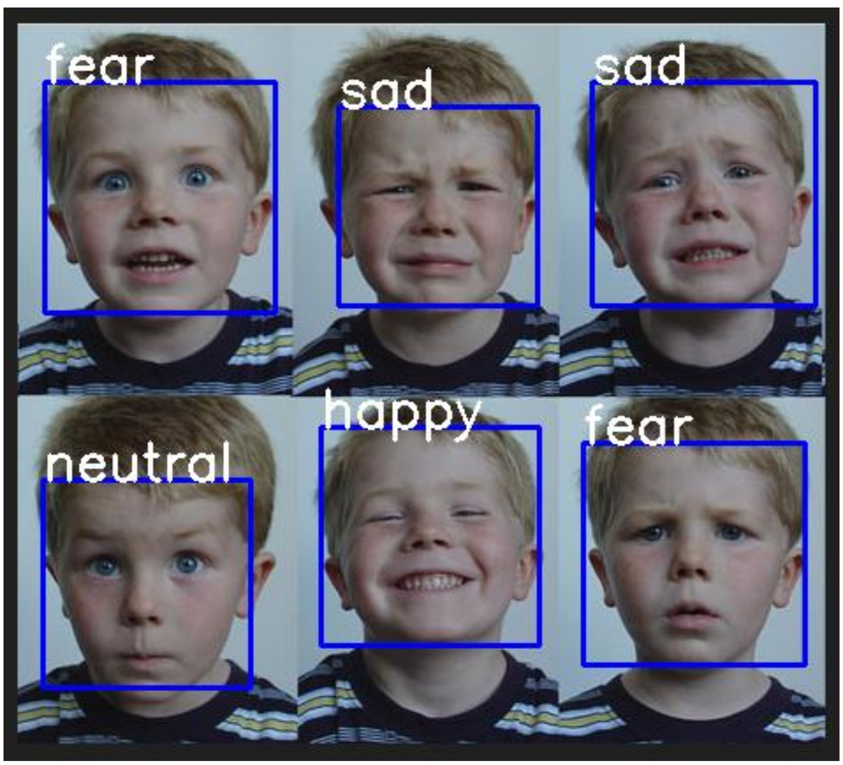

Fig 7.8 Output

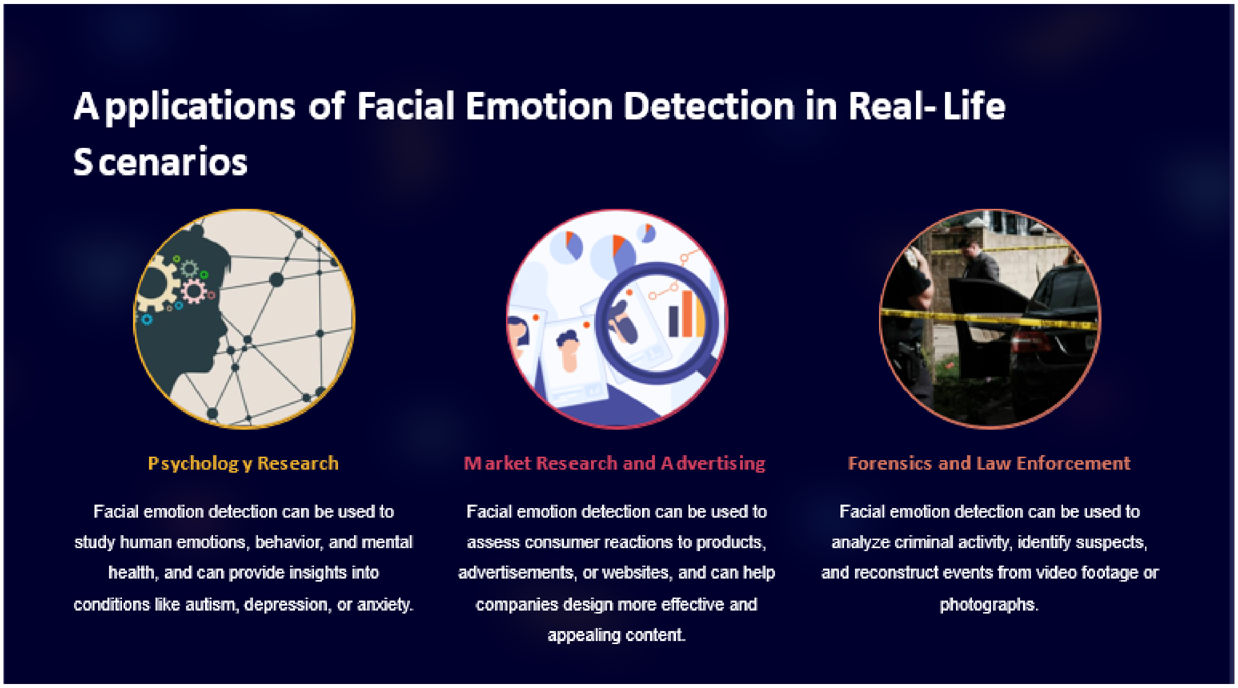

Fig 7.9 Application

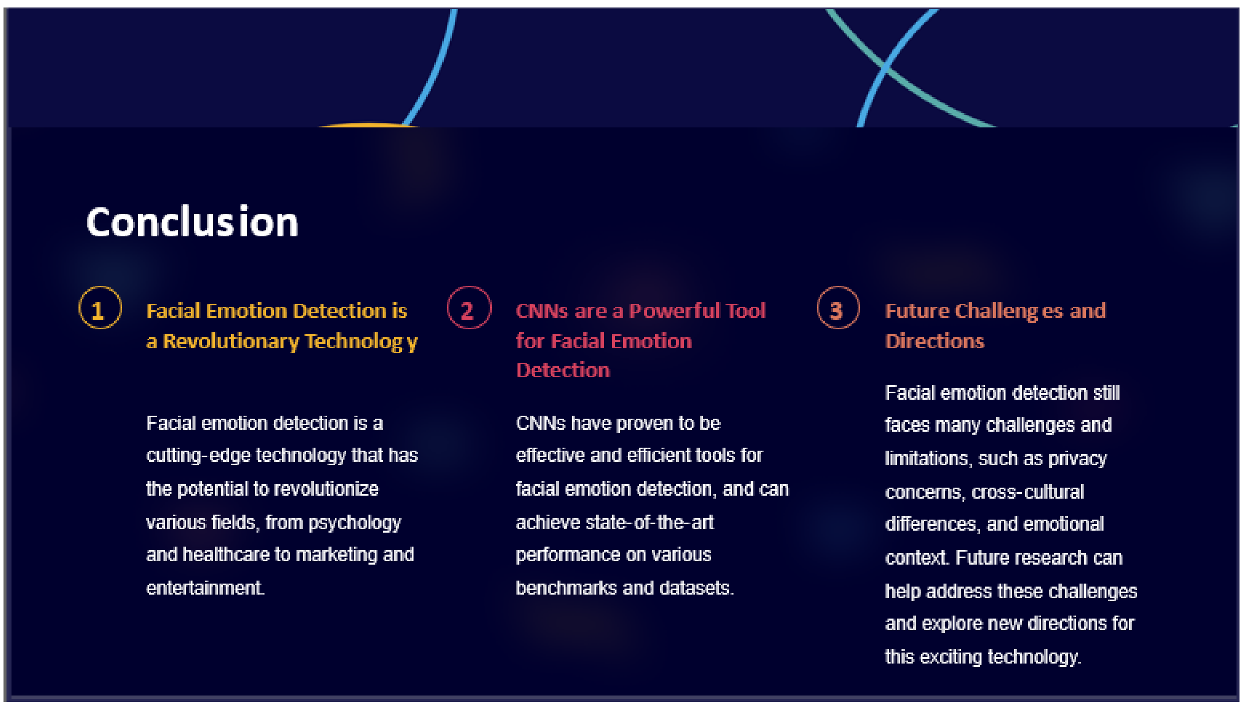

Fig 7.10 Conclusion