BTI3423 MACHINE VISION

PROJECT REPORT
FACE DETECTION AND EMOTION RECOGNITION

SEM I 2023/2024

GROUP MEMBERS

NAME	MATRIC ID
NIK NURUL SYAZWANI BINTI NIK FAZIR	TP21002
ARSHADA NADIA BINTI ABDULLAH	TP21003

LECTURER'S NAME
AHMAD SHAHRIZAN BIN ABDUL GHANI

FACULTY OF MANUFACTURING AND MECHATRONIC ENGINEERING TECHNOLOGY

UNIVERSITI MALAYSIA PAHANG

26600 PEKAN, PAHANG

MALAYSIA

INTRODUCTION

This project endeavors to develop an intelligent system capable of perceiving and interpreting human emotions by analyzing facial expressions captured in images or video streams. Understanding and recognizing emotions through facial cues are crucial aspects of human interaction and communication. Utilizing fundamental computer techniques, this project aims to bridge this understanding by deciphering how individuals feel based on their facial movements.

At the heart of this endeavor lies a series of intricate tasks. Firstly, the system must proficiently detect human faces within images or video footage. Subsequently, it must delve deeper, leveraging sophisticated algorithms and image processing methodologies to discern the emotional states depicted by these detected facial expressions. Finally, the project aims to visually represent these identified emotions in an easily understandable format.

By harnessing the capabilities of computer programs and specialized methodologies, the primary objective is to refine face detection algorithms, enabling accurate identification of facial features and the extraction of emotional cues. These cues encompass a spectrum of emotions, encompassing joy, sadness, surprise, anger, and more nuanced feelings. The system aims to comprehend and classify these emotions to create a comprehensive emotional analysis.

The significance of this project extends across numerous domains. The amalgamation of computer learning with face detection techniques is envisioned to craft a system that not only discerns facial identities but also comprehends the emotional nuances behind these identities. The potential applications span diverse fields, including but not limited to enhancing human-computer interaction, deciphering emotional content embedded in textual information, facilitating advancements in healthcare by detecting emotional states, and refining entertainment experiences through personalized emotional responses.

Through the convergence of face detection and emotion understanding, this initiative aspires to unlock innovative pathways for machines to discern, interpret, and respond to human emotions solely through the analysis of facial expressions. The pursuit of such technological advancements aims to redefine human-computer interaction paradigms and deepen our understanding of emotional intelligence within technological frameworks.

OBJECTIVES

- Develop a Face Detection System: Implement a robust system capable of accurately detecting human faces within images or video streams, addressing challenges related to varying poses, lighting conditions, and occlusions.

- Emotion Recognition: Create an intelligent model capable of analyzing facial expressions to recognize and classify various emotions exhibited by individuals, including but not limited to happiness, sadness, anger, surprise, and neutrality.

- Integration of Face Detection and Emotion Recognition: Integrate the developed face detection and emotion recognition models to work cohesively, allowing for the simultaneous detection of faces and the analysis of emotional cues within live video streams or static images.

- Model Accuracy and Performance: Aim for high accuracy and robust performance of the developed models in accurately identifying faces and interpreting emotional states. This includes optimizing the model's precision, recall, and overall classification accuracy.

- Real-Time Application: Enable real-time processing of facial expressions to swiftly recognize emotions, allowing for immediate feedback or response in applications such as human-computer interaction or emotion-driven interfaces.

- Generalization and Adaptability: Ensure the developed system's adaptability and generalization across diverse datasets and scenarios, allowing for its potential application in various domains beyond the scope of this project.

- Innovation and Advancements: Aim to contribute innovative solutions and advancements in the field of computer vision, human-computer interaction, and emotional analysis through the amalgamation of face detection and emotion recognition techniques.

- Future Scope and Potential Applications: Explore potential applications and future enhancements, paving the way for advancements in fields such as healthcare, entertainment, robotics, and sentiment analysis.

METHODOLOGY

The methodology for the Face Detection and Emotion Recognition project, amalgamating the FER2013 and fer2013plus datasets, involves a multi-step approach encompassing data preparation, model architecture design, training, validation, testing, model deployment, and real-time emotion analysis for images and video streams.

Data Preparation:

The combined dataset, an amalgamation of FER2013 and fer2013plus datasets, offers a diverse array of labeled facial expression images. This dataset is meticulously divided into three subsets: training, validation, and testing. The images undergo preprocessing to standardize their size to 48x48 pixels, ensuring consistency across the dataset. Such standardization streamlines the subsequent model training process.

Model Architecture Design:

A Convolutional Neural Network (CNN) architecture, constructed using Keras with TensorFlow as its backend, serves as the foundation for emotion recognition. This model comprises various layers, starting with Conv2D layers for feature extraction, followed by MaxPooling2D layers for downsampling, leading to deeper layers of Conv2D and MaxPooling2D. The architecture culminates in a flattened layer, dense layers incorporating Rectified Linear Unit (ReLU) activation for feature processing, and an output layer employing the softmax activation function to classify emotions into eight distinct categories.

Training Process:

Data augmentation and normalization techniques are employed through the ImageDataGenerator. Rescaling standardizes pixel values, while data augmentation diversifies the dataset by applying transformations like rotation, zoom, and flip to the training images. The model is compiled using the Adam optimizer and categorical cross-entropy loss function, ensuring optimal weight adjustments during training. The iterative process involves epochs, with the model learning from the augmented dataset to minimize the loss function and optimize classification accuracy.

Validation and Testing:

The model's efficacy is assessed using a validation dataset, enabling continuous evaluation of its performance after each training epoch. The evaluation metrics from the validation set guide adjustments to prevent overfitting and ensure robustness. Subsequently, the model's accuracy and generalization capability are thoroughly tested on an independent test dataset, measuring its ability to accurately predict emotions on previously unseen images.

Model Deployment:

Upon achieving satisfactory performance, the trained model is saved in a '.keras' file format. This file becomes the core component of the deployed system, ready for implementation in detecting faces and recognizing emotions in various scenarios, such as images and video streams.

Emotion Analysis for Images and Video Streams:

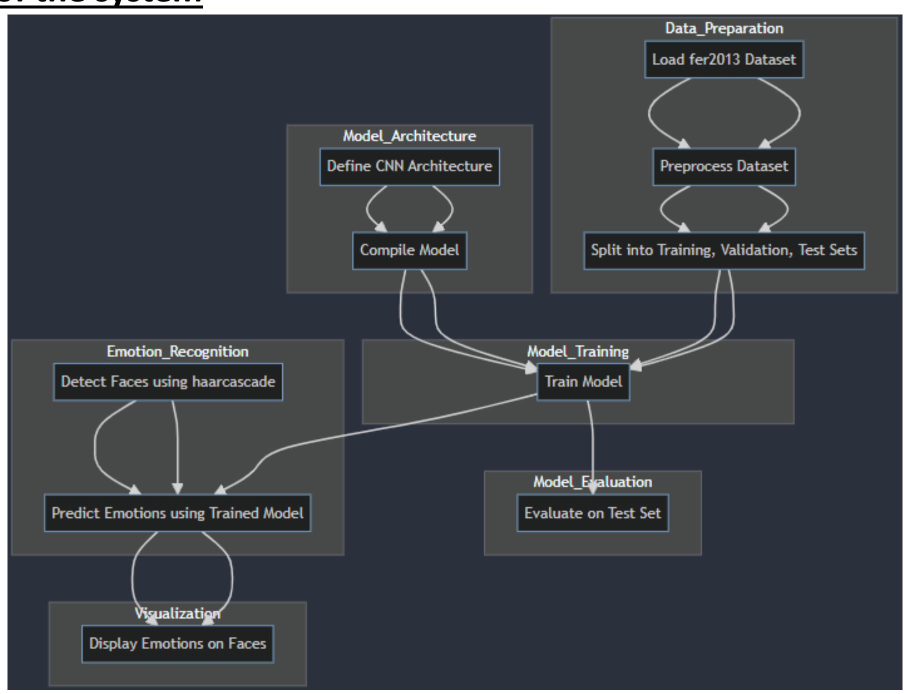
For image analysis, an implemented function leverages OpenCV, employing the pre-trained emotion detection model. It reads images, performs face detection, extracts facial regions, preprocesses them to match the model's input shape, predicts emotions, and overlays the detected emotions on the respective faces within the image.

Real-time Emotion Detection in Video Streams:

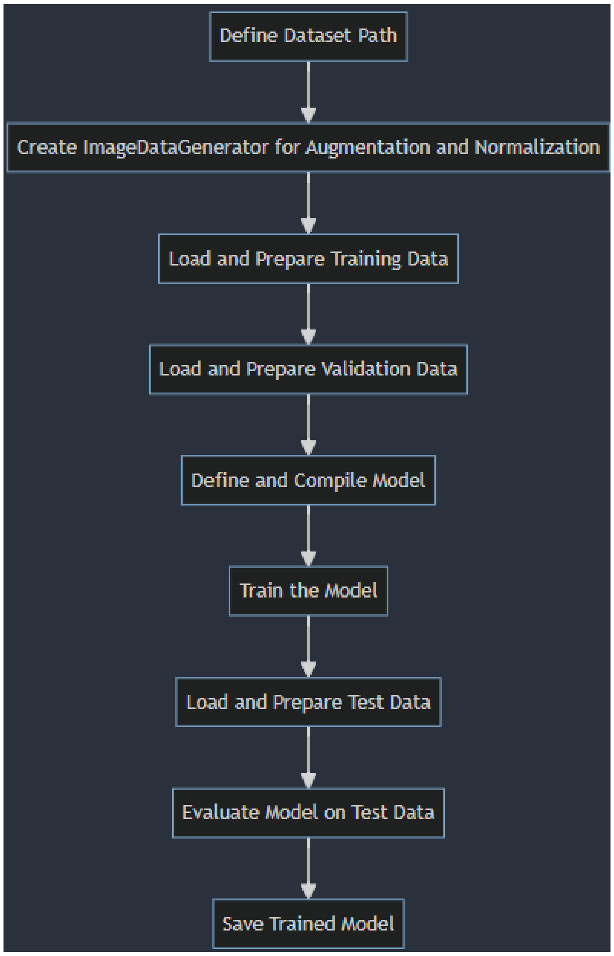
Similarly, for video streams, another function is employed, continuously capturing frames, detecting faces, predicting emotions, and overlaying the identified emotions onto the video frames. The result is a real-time video output exhibiting emotions detected within the stream.

This comprehensive methodology converges data preprocessing, model development, training, validation, testing, and deployment, culminating in a sophisticated system proficient in detecting faces and recognizing emotions within images or live video streams, thereby facilitating diverse applications in human-computer interaction, surveillance, and emotional analysis domains.

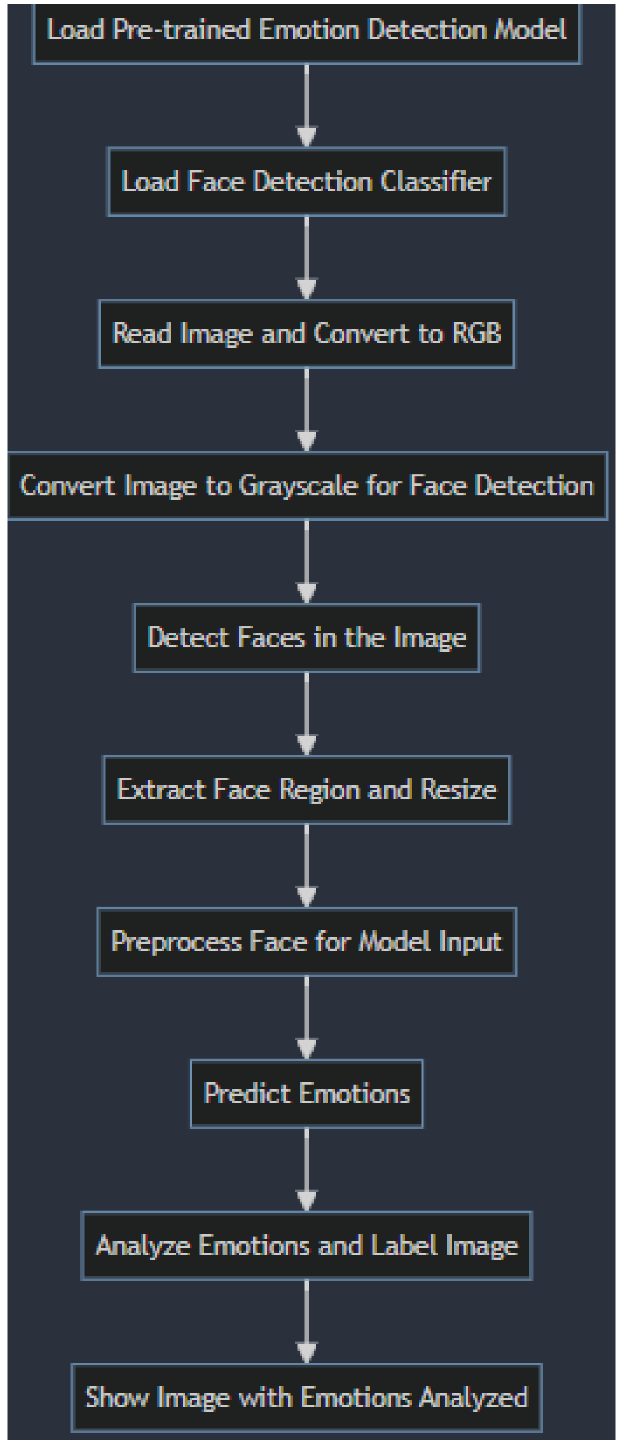
Flowchart of the system



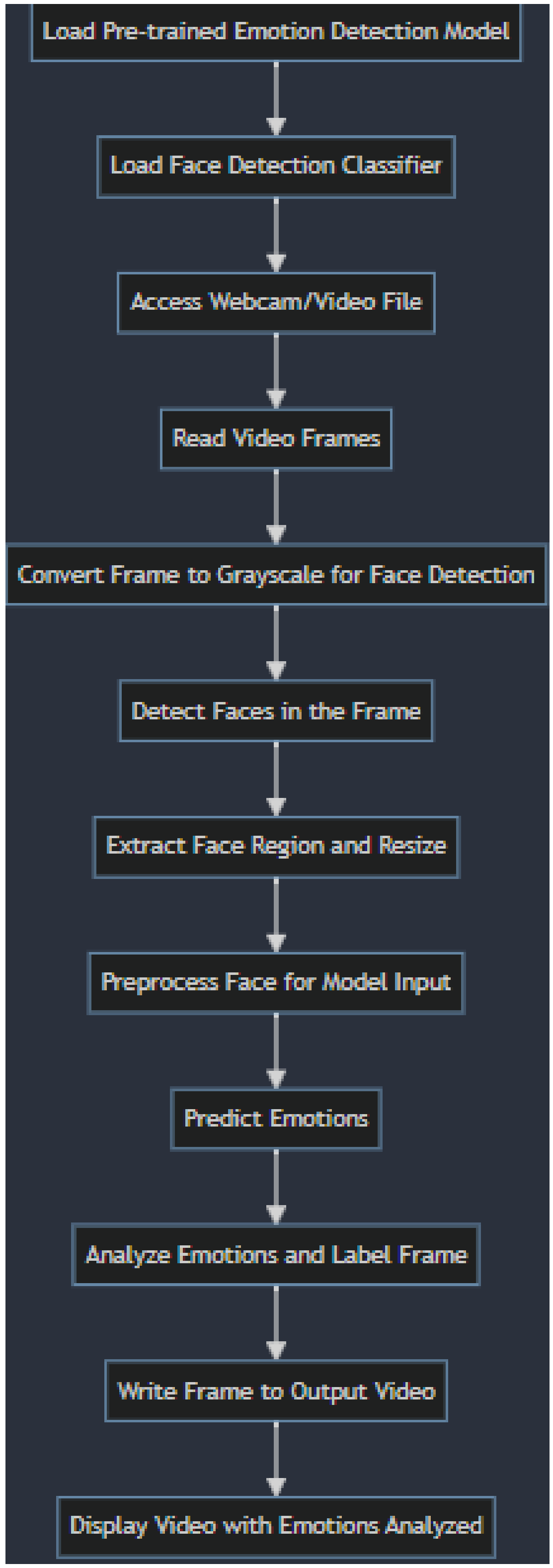
Flowchart of the whole system



Flowchart for training model



Flowchart for emotion detection in image



Flowchart for emotion detection in video stream

CODING

For training and testing using CNN

```
# -*- coding: utf-8 -*-
"""
Created on Sun Dec 17 01:10:25 2023

@author: syazw
"""
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense

# Define the path to your dataset
dataset_path = r'C:\Users\syazw\Desktop\BTI3423 MACHINE VISION\Project\Project Face
Detection & Emotion Recognition\combined'

# Constants for image size and batch size
img_width, img_height = 48, 48
batch_size = 50

# Create an ImageDataGenerator for data augmentation and normalization
datagen = ImageDataGenerator(rescale=1./255, validation_split=0.20)

# Load and prepare the training data
train_generator = datagen.flow_from_directory(
    dataset_path + '/train',
    target_size=(img_width, img_height),
    batch_size=batch_size,
    class_mode='categorical',
    subset='training')

# Load and prepare the validation data
validation_generator = datagen.flow_from_directory(
    dataset_path + '/train',
    target_size=(img_width, img_height),
    batch_size=batch_size,
    class_mode='categorical',
    subset='validation')
```

```
# Define and compile the model
model = Sequential() #Creates linear stack of layers for the model
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(img_width, img_height, 3)))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Conv2D(256, (3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dense(8, activation='softmax'))  # Output layer with 8 classes for emotions

#Configures the model for training
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# Train the model
history = model.fit(
    train_generator,
    steps_per_epoch=train_generator.samples // batch_size,
    epochs=10,
    validation_data=validation_generator,
    validation_steps=validation_generator.samples // batch_size)

# Load and prepare the test data
test_generator = datagen.flow_from_directory(
    dataset_path + '/test',
    target_size=(img_width, img_height),
    batch_size=batch_size,
    class_mode='categorical')

# Evaluate the model on the test data
test_loss, test_accuracy = model.evaluate(test_generator, steps=test_generator.samples //
batch_size)
print(f"Test Accuracy after further augmentation: {test_accuracy * 100:.2f}%")

# Save the trained model
model.save('trained_emotion_model3_2.keras')
```

<u>For Image</u>

```
# -*- coding: utf-8 -*-
"""
Created on Sun Dec 17 00:56:37 2023

@author: syazw
"""
import cv2
import numpy as np
from tensorflow.keras.models import load_model

# Load the pre-trained emotion detection model
emotion_model = load_model('trained_emotion_model3_2.keras')

# Load the face detection classifier
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +
'haarcascade_frontalface_default.xml')

# Function to detect and analyze emotions in the image
def detect_emotions(image_path):
    image = cv2.imread(image_path)
    image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)  # Convert BGR to RGB

    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)  # Convert to grayscale for face detection

    #Detects faces in the grayscale image using Haar cascade classifier
    faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5, minSize=(30,
30))

    #Iterates through the detected faces
    for (x, y, w, h) in faces:
        face_roi = gray[y:y + h, x:x + w] # Extract face region in grayscale
        face_roi_color = image_rgb[y:y + h, x:x + w]  # Extract face region in color

        face_roi = cv2.resize(face_roi, (48, 48))
        face_roi_color = cv2.resize(face_roi_color, (48, 48))  # Resize color image

        # Convert grayscale face to RGB (3 channels)
        face_roi_color = cv2.cvtColor(face_roi_color, cv2.COLOR_RGB2BGR)

        # Preprocess the face to match the model input shape
        face_roi_color = face_roi_color.astype('float') / 255.0
```

```
        face_roi_color = np.expand_dims(face_roi_color, axis=0)

        # Predict emotions using the pre-trained model
        prediction = emotion_model.predict(face_roi_color)
        emotion = np.argmax(prediction)
        emotions = ["anger", "contempt", "disgust", "fear", "happiness", "neutral", "sadness",
"surprise"]
        # Match the predicted emotion index to the predefines list of emotions
        emotion_text = emotions[emotion]

        # Draw rectangle around detected face and label the emotion
        cv2.rectangle(image, (x, y), (x + w, y + h), (255, 0, 0), 2)
        cv2.putText(image, emotion_text, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.9, (255, 0, 0),
2)

    # Show the image with emotions analyzed
    cv2.imshow('Emotion Detection', image)
    cv2.waitKey(0)
    cv2.destroyAllWindows()

# Call the function with the image path
detect_emotions(r"C:\Users\syazw\Desktop\BTI3423 MACHINE VISION\Project\Project Face
Detection & Emotion Recognition\4.jpg")
```

For video stream

```
# -*- coding: utf-8 -*-
"""
Created on Sun Dec 17 01:01:39 2023

@author: syazw
"""
import cv2
import numpy as np
from tensorflow.keras.models import load_model

# Load the pre-trained emotion detection model
emotion_model = load_model('trained_emotion_model3_2.keras')

# Load the face detection classifier
face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades +
'haarcascade_frontalface_default.xml')
```

```
# Function to detect and analyze emotions in a video stream
def detect_emotions_in_video():
    cap = cv2.VideoCapture(1)  # Access webcam; change to 'filename.mp4' for video file

    # Get video frame properties
    width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
    height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
    fps = int(cap.get(cv2.CAP_PROP_FPS))

    # Define the codec and create VideoWriter object for the output video
    fourcc = cv2.VideoWriter_fourcc(*'XVID')
    out = cv2.VideoWriter('output_video2.mp4', fourcc, fps, (width, height))

    #enter the loop to continuously  read freames from video stream
    while True:
        ret, frame = cap.read() # Read a frame from the video capture
        if not ret:
            break

        gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY) #Converts frame to grayscale for face detection
        faces = face_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))

        for (x, y, w, h) in faces:
            face_roi = gray[y:y + h, x:x + w]
            face_roi_color = frame[y:y + h, x:x + w]

            face_roi = cv2.resize(face_roi, (48, 48))
            face_roi_color = cv2.resize(face_roi_color, (48, 48))
            face_roi_color = cv2.cvtColor(face_roi_color, cv2.COLOR_BGR2RGB)

            face_roi_color = face_roi_color.astype('float') / 255.0
            face_roi_color = np.expand_dims(face_roi_color, axis=0)

            prediction = emotion_model.predict(face_roi_color)
            emotion = np.argmax(prediction)
            emotions = ["anger", "contempt", "disgust", "fear", "happiness", "neutral", "sadness", "surprise"]
            emotion_text = emotions[emotion]
```

```
            cv2.rectangle(frame, (x, y), (x + w, y + h), (255, 0, 0), 2)
            cv2.putText(frame, emotion_text, (x, y - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.9, (255, 0,
0), 2)

        out.write(frame)  # Write the processed frame to the output video file

        # Loops break when press 'q' key, release webcam, closes the output video file
        cv2.imshow('Emotion Detection', frame)
        if cv2.waitKey(1) & 0xFF == ord('q'):
            break

    cap.release()
    out.release()  # Release VideoWriter
    cv2.destroyAllWindows()

# Call the function to start detecting emotions in the video stream and save the result
detect_emotions_in_video()
```

RESULTS AND DISCUSSION

1. Model Performance:

The model's training journey revealed promising yet moderate outcomes. Trained on the amalgamated dataset from FER2013 and fer2013plus, each containing eight distinct emotion classes, the model depicted commendable performance. Throughout 10 epochs, a discernible progression was witnessed in both accuracy and loss metrics. Commencing at a training accuracy of 76.44%, it showcased an upward trend but eventually converged, reaching a validation accuracy plateau of around 67.30%. This plateau suggests a possible saturation in the model's learning capacity concerning the provided data. The training process was efficient, indicating a substantial learning curve for the network in understanding the emotional nuances encoded in the dataset.

In contrast, during testing, the model exhibited a slightly lower accuracy of 62.40%. While it successfully recognized prevalent emotions such as anger, fear, happiness, neutral, sadness, and surprise, it encountered challenges in accurately identifying 'disgusted' and certain other emotions absent in the dataset. This discrepancy implies a potential need for further refinement to enhance the model's capability in comprehending less frequently expressed emotions.

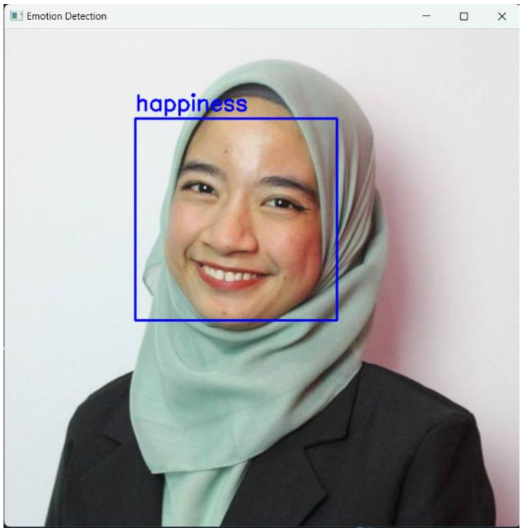
```
In [4]: runfile('C:/Users/syazw/Desktop/BTI3423 MACHINE VISION/Project/Project Face Detection & Emotion Recognition/train3.py', wdir='C:/Users/syazw/Desktop/BTI3423 MACHINE VISION/Project/
Project Face Detection & Emotion Recognition')
Found 45680 images belonging to 8 classes.
Found 11415 images belonging to 8 classes.
Epoch 1/10
913/913 [==============================] - 326s 354ms/step - loss: 1.5171 - accuracy: 0.4175 - val_loss: 1.2887 - val_accuracy: 0.5320
Epoch 2/10
913/913 [==============================] - 279s 305ms/step - loss: 1.1314 - accuracy: 0.5912 - val_loss: 1.1497 - val_accuracy: 0.5873
Epoch 3/10
913/913 [==============================] - 289s 316ms/step - loss: 1.0033 - accuracy: 0.6405 - val_loss: 1.1127 - val_accuracy: 0.6122
Epoch 4/10
913/913 [==============================] - 237s 260ms/step - loss: 0.9197 - accuracy: 0.6708 - val_loss: 1.0725 - val_accuracy: 0.6286
Epoch 5/10
913/913 [==============================] - 247s 270ms/step - loss: 0.8480 - accuracy: 0.6951 - val_loss: 1.0383 - val_accuracy: 0.6425
Epoch 6/10
913/913 [==============================] - 396s 434ms/step - loss: 0.7881 - accuracy: 0.7144 - val_loss: 1.0540 - val_accuracy: 0.6428
Epoch 7/10
913/913 [==============================] - 355s 388ms/step - loss: 0.7359 - accuracy: 0.7334 - val_loss: 1.0587 - val_accuracy: 0.6556
Epoch 8/10
913/913 [==============================] - 355s 389ms/step - loss: 0.6979 - accuracy: 0.7454 - val_loss: 1.0659 - val_accuracy: 0.6613
Epoch 9/10
913/913 [==============================] - 226s 248ms/step - loss: 0.6585 - accuracy: 0.7565 - val_loss: 1.0316 - val_accuracy: 0.6754
Epoch 10/10
913/913 [==============================] - 188s 205ms/step - loss: 0.6265 - accuracy: 0.7644 - val_loss: 1.0397 - val_accuracy: 0.6730
Found 14277 images belonging to 8 classes.
285/285 [==============================] - 31s 108ms/step - loss: 1.1240 - accuracy: 0.6240
Test Accuracy after further augmentation: 62.40%
```

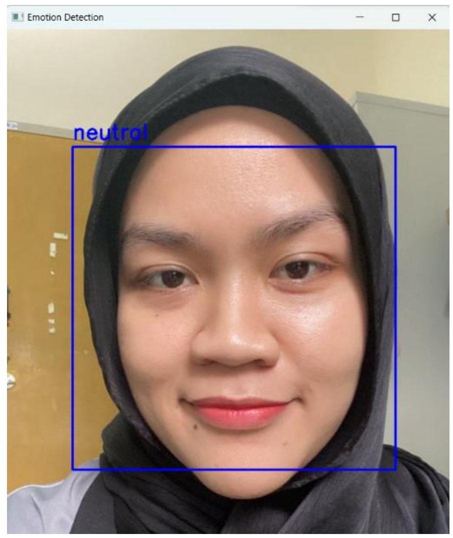
2. Training Process:

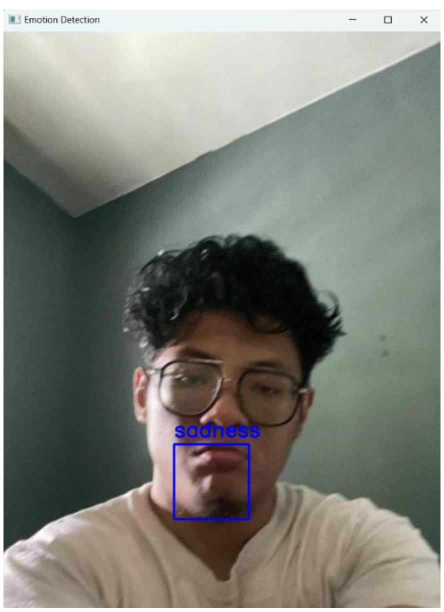
The iterative training process illustrated consistent improvements across epochs. However, the incremental gain in accuracy and simultaneous loss reduction demonstrated diminishing returns, indicative of the model's approaching saturation. Employing data augmentation techniques did not substantially elevate the model's performance, signifying limited enhancement from additional data manipulations. This highlights the need for a more diverse dataset to propel the model's accuracy further.

3. Real-time Emotion Recognition:

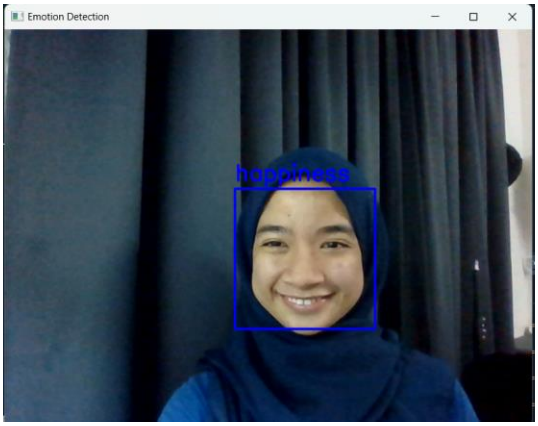
Image Analysis: Upon applying the model to images, its proficiency in identifying prominent emotions like anger, fear, happiness, neutral, sadness, and surprise was commendable. Nonetheless, it struggled with recognizing 'disgusted' and other emotions not extensively represented in the dataset. This indicates a potential gap in the model's ability to comprehend emotions lacking sufficient training samples.

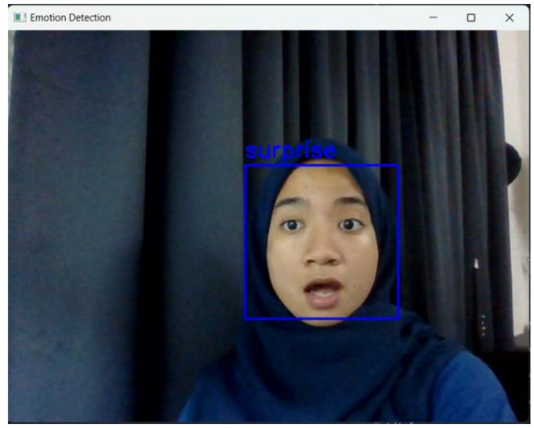


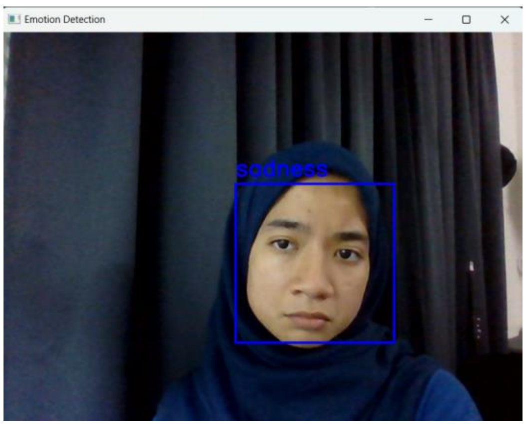


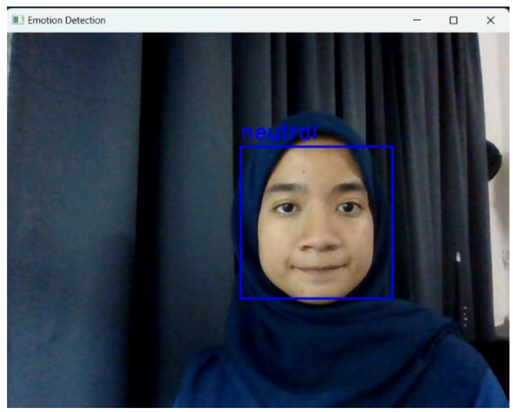


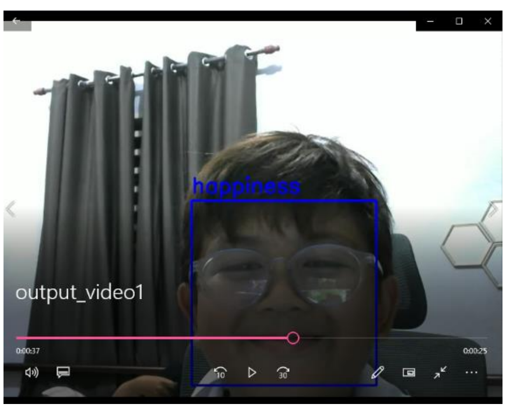
Video Challenges: The real-time analysis of video streams presented inherent challenges in discerning subtle emotions such as 'disgusted' and 'contempt.' Variations in facial expressions and nuances specific to these emotions posed significant hurdles, impacting the model's accuracy in real-time emotion analysis.


Emotion Detection
happiness


Emotion Detection
surprise


Emotion Detection
sadness

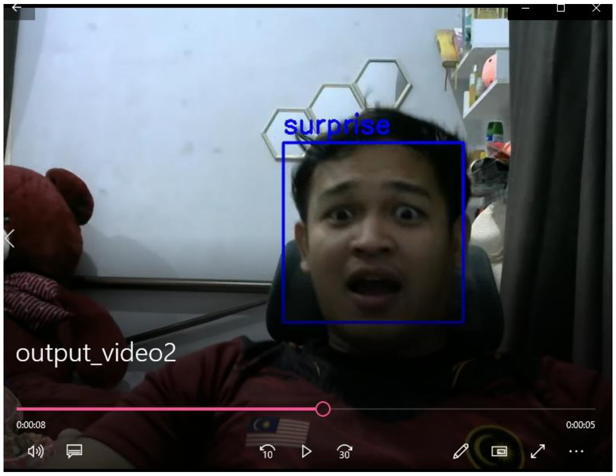

Emotion Detection
neutral

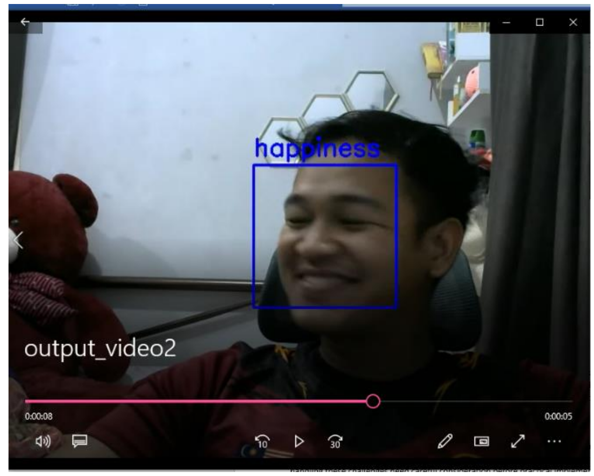

happiness
output_video1


anger


neutral
neutral


sadness
output_video2





4. Accuracy Discussion:

The model exhibited notable strength in recognizing prevalent emotions yet encountered difficulties in identifying less commonly expressed emotions. This limitation can be attributed to imbalances in the dataset's representation of various emotions, potentially hindering the model's capability to generalize well across all emotion classes.

5. Model Deployment:

Deploying the model in real-world scenarios would necessitate addressing the challenges encountered in recognizing less frequently expressed emotions. Strategies for handling these challenges need careful consideration before practical implementation to ensure accurate emotion detection across diverse scenarios.

6. Limitations and Future Scope:

Limitations: The model's limitations stem from imbalances and inadequate representations of certain emotions within the dataset. Moreover, its struggles with recognizing less frequently expressed emotions highlight the need for a more comprehensive and diverse dataset.

Improvement Opportunities: Future enhancements involve acquiring a more diverse dataset that offers a balanced representation of emotions. Exploring advanced model architectures or ensemble methods could potentially mitigate the challenges encountered in accurately identifying subtle or less frequently expressed emotions, such as 'disgusted' and 'contempt.'

CONCLUSION

The undertaking of developing a system for Face Detection and Emotion Recognition has been an enlightening journey marked by notable successes and inherent challenges. The amalgamation of FER2013 and fer2013plus datasets provided a robust foundation for training the emotion detection model, enabling it to discern prominent emotions with commendable accuracy.

The model exhibited commendable proficiency in recognizing prevalent emotions like anger, fear, happiness, neutrality, sadness, and surprise, indicating a robust understanding of these expressions. However, challenges arose in identifying less frequently expressed emotions, notably 'disgusted' and 'contempt.' These difficulties stemmed from imbalances and inadequacies in the dataset's representation, underscoring the significance of a more comprehensive and balanced dataset for improved model generalization.

Throughout the training process, the model showcased consistent improvements in accuracy, albeit reaching a saturation point, indicating diminishing returns with prolonged training. The validation accuracy plateaued around 67.30%, suggesting a potential limit in the model's capacity to learn further from the provided dataset.

Real-time emotion recognition, particularly in video streams, posed challenges due to the subtlety and variability in facial expressions, affecting the model's accuracy in discerning less commonly expressed emotions. Enhancing the model's capability to accurately recognize a broader spectrum of emotions, especially those less frequently observed, remains a significant area for improvement.

In conclusion, while the developed model demonstrated commendable performance in recognizing prevalent emotions, there exist opportunities for refinement and augmentation. Addressing dataset imbalances, exploring diverse datasets, refining model architectures, and implementing advanced techniques will be pivotal for enhancing the model's accuracy and broadening its capability to perceive nuanced emotions. Embracing these improvements will enable the system to be more adept at real-world emotion recognition scenarios, facilitating a more comprehensive understanding of human expressions.

REFERENCES

1. *Sklearn.linear_model.linearregression*. scikit. (n.d.). https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html
2. SubhadityaMukherjee. (2022, February 21). *Fer2013Plus*. Kaggle. https://www.kaggle.com/datasets/subhaditya/fer2013plus
3. Sambare, M. (2020, July 19). Fer-2013. Kaggle. https://www.kaggle.com/datasets/msambare/fer2013/data