SVKM MITHIBAI COLLEGE OF SCIENCE,COMMERCE AND ARTS MUMBAI

FACIAL EMOTION RECOGNITION

DEEP LEARNING PROJECT

MSC COMPUTER SCIENCE

NAME : VIKRAM VISHNU PANDEY

DIV :F

ROLL NO: 03

YEAR :2024-2025

MENTORED BY : JAYASHREE RAVI MA’AM

Project Introduction

Over the past decade, the Indian education has experienced significant transformation due to the rise of web-based learning services, particularly eLearning platforms. The global eLearning market is projected to grow eightfold over the next five years.

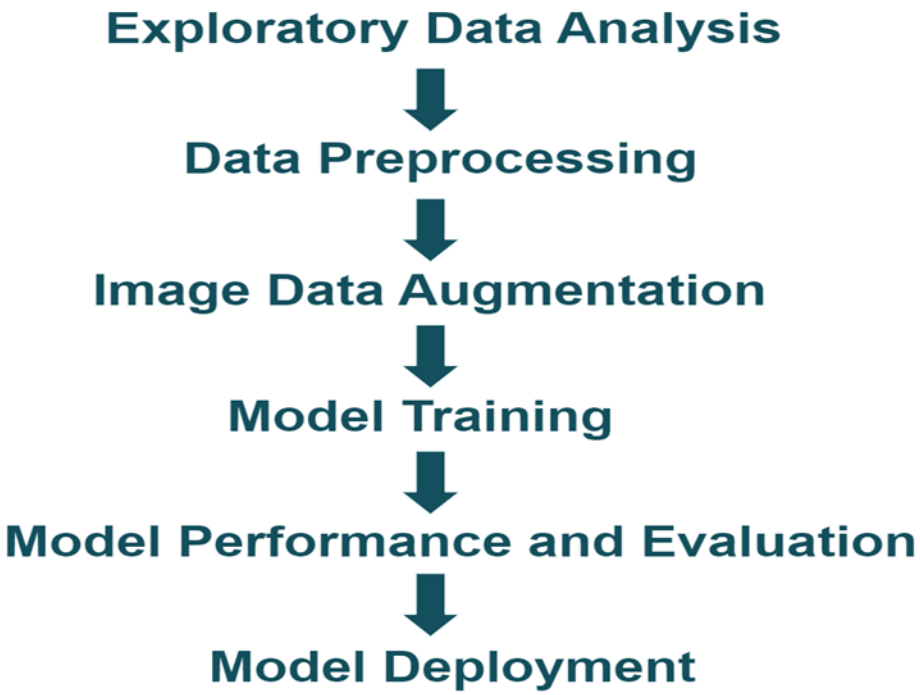
Despite this rapid growth, digital learning faces several challenges compared to traditional brick-and-mortar classrooms. One major challenge is ensuring quality learning for students. While digital platforms may offer superior content quality, assessing whether students are comprehending the material in real-time remains an open question. In physical classrooms, teachers can observe students' facial expressions and emotions, adjusting their teaching pace accordingly and identifying those who need extra attention.

Digital classrooms, often conducted via video conferencing software like Zoom, make it difficult for teachers to monitor the mood and engagement of a medium-sized class (25-50 students). This lack of surveillance can lead to decreased student focus on the content. However, digital platforms have the advantage of leveraging data and machine learning. They generate data in the form of video, audio, and text, which can be analyzed using deep learning algorithms. These **systems not only address the surveillance issue** but **also eliminate human bias**, translating information from the teacher's observations into quantifiable data that can be **analyzed** and tracked.

EXPLANATION

1. Here , I will try to solve the above-mentioned challenge by applying deep learning algorithms to live video data.
2. The solution to this problem is by recognizing facial emotions.
3. This is a few shot learning live face emotion detection system. The model should be able to real-time
4. identify the emotions of students in a live class

Flow of the project



Model Architecture:

1. Exploratory Data Analysis (EDA)

- **Objective: To Understand the dataset's structure, distribution, and potential issues.**
- **Steps:**
    - **Data Visualization: Plot the distribution of different emotions.**
    - **Summary Statistics: Calculate mean, median, and standard deviation for pixel values.**
    - **Missing Values: Identify and handle any missing or corrupted data.**
    - **Class Imbalance: Check for imbalanced classes and plan for handling them.**

2. Data Preprocessing

- **Objective: To Prepare the data for model training.**
- **Steps:**

- **Face Detection: Use algorithms like MTCNN to detect faces in images.**
- **Normalization: Scale pixel values to a range (e.g., 0 to 1).**
- **Resizing: Resize images to a consistent size (e.g., 48x48 pixels).**
- **Data Augmentation: Apply transformations to increase data diversity.**

3. Image Data Augmentation

- **Objective: Enhance the training dataset by creating variations of existing images.**
- **Techniques:**
  - **Rotation: Rotate images by small angles.**
  - **Flipping: Flip images horizontally or vertically.**
  - **Scaling: Zoom in or out on images.**
  - **Translation: Shift images horizontally or vertically.**
  - **Brightness Adjustment: Change the brightness of images.**
  -

4. Model Training

- **Objective: Train a model to recognize facial emotions.**
- **Steps:**
  - **Model Selection: Choose a suitable model architecture**
  - **Loss Function: Use a loss function like categorical cross-entropy.**
  - **Optimizer: Select an optimizer like Adam or SGD.**
  - **Training: Train the model on the preprocessed and augmented data.**
  - **Validation: Validate the model on a separate validation set to monitor performance.**

5. Model Performance Evaluation

- **Objective: Assess the model's performance on unseen data.**
- **Metrics:**
  - **Accuracy: Overall correctness of the model.**
  - **Precision, Recall, F1-Score: Evaluate performance for each class.**
  - **Confusion Matrix: Visualize true vs. predicted classifications.**

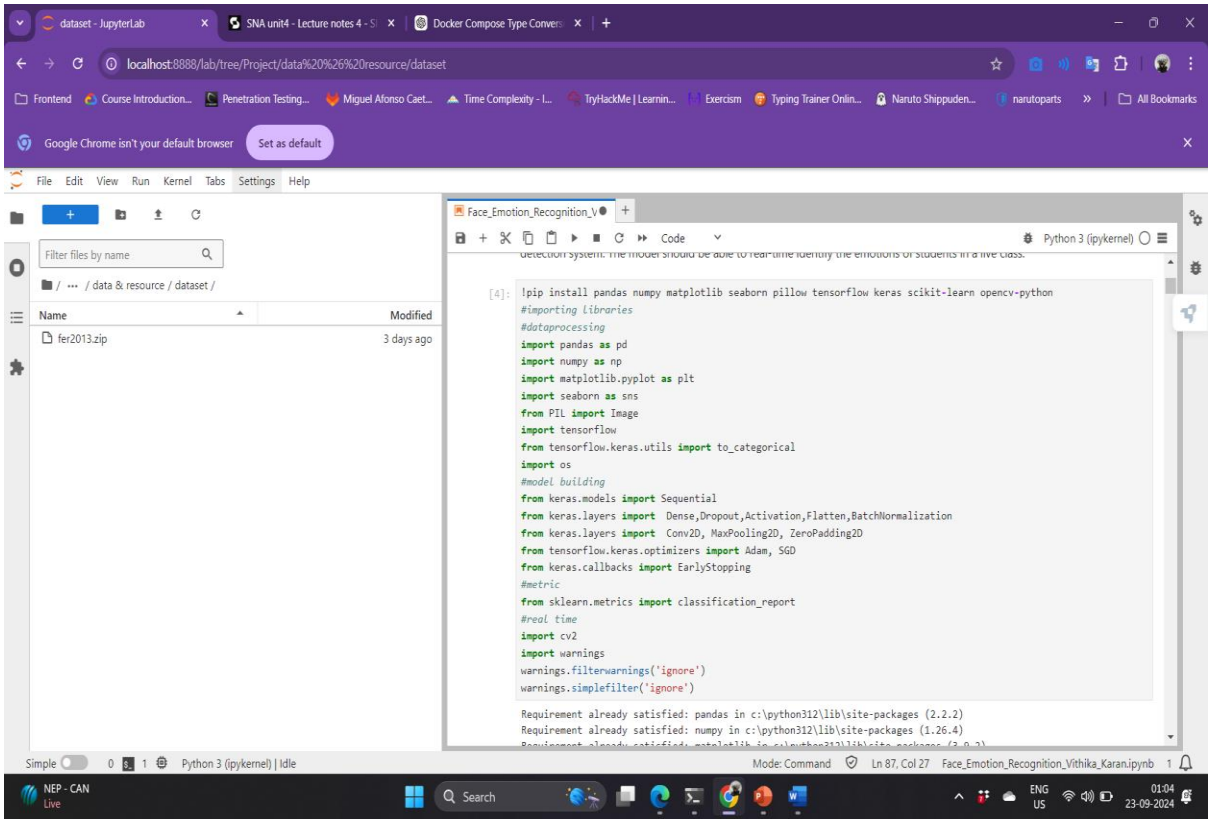
    - **ROC Curve: Assess the trade-off between true positive and false positive rates.**

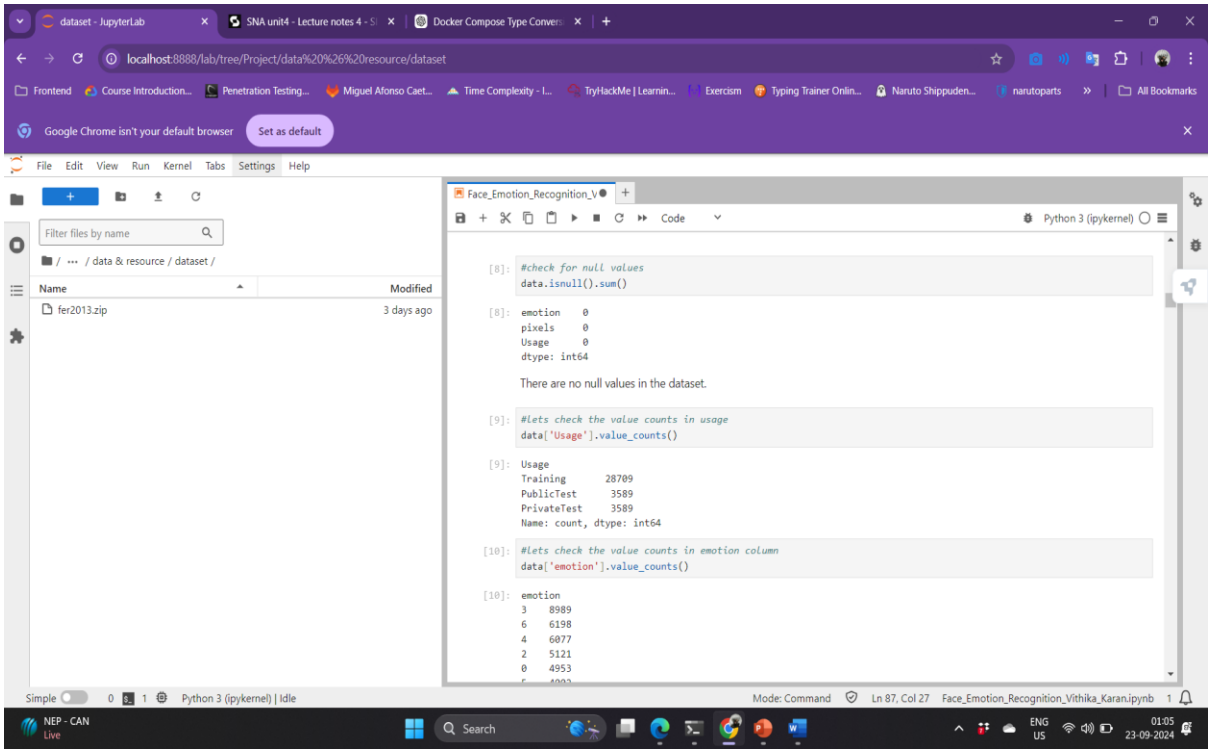
6. Model Deployment

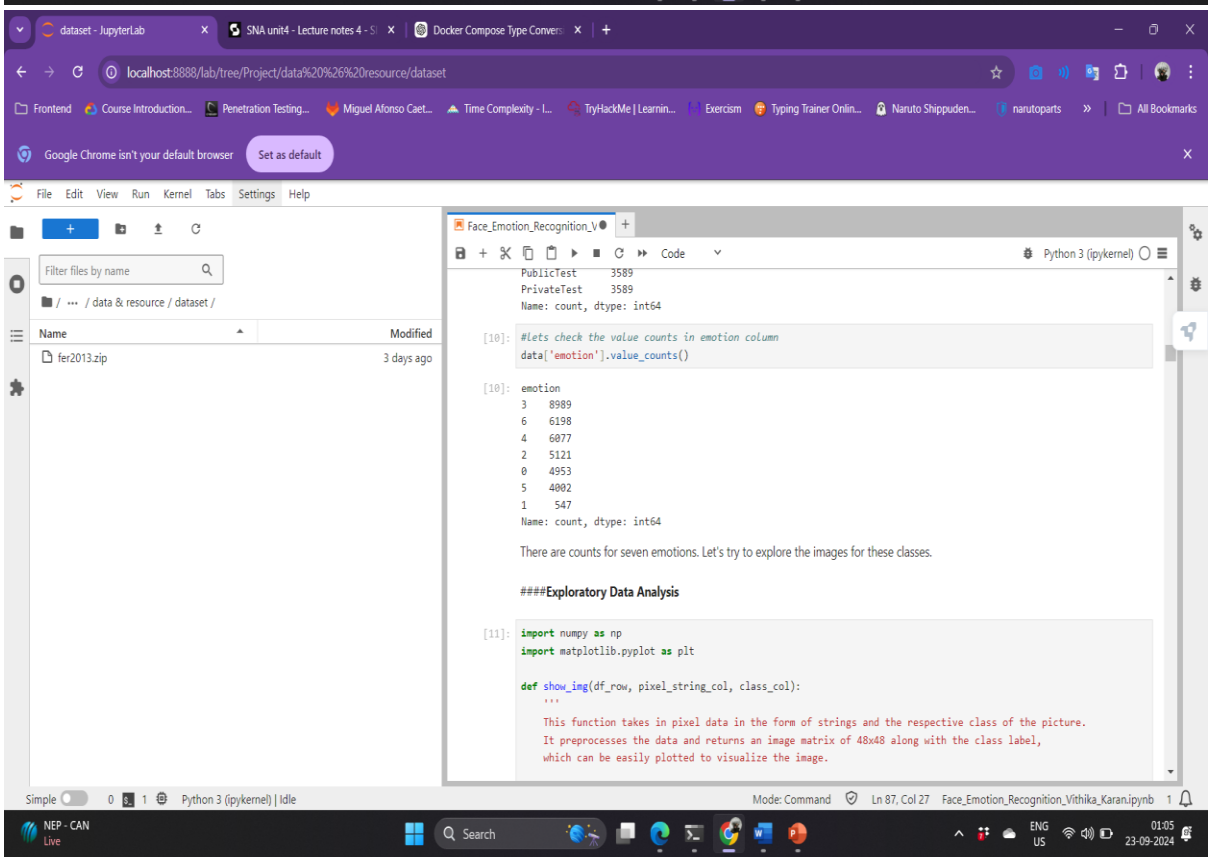
- **Objective: Integrate the trained model into a live system.**
- **Steps:**
    - **Real-Time Inference: Implement real-time emotion detection from video streams.**

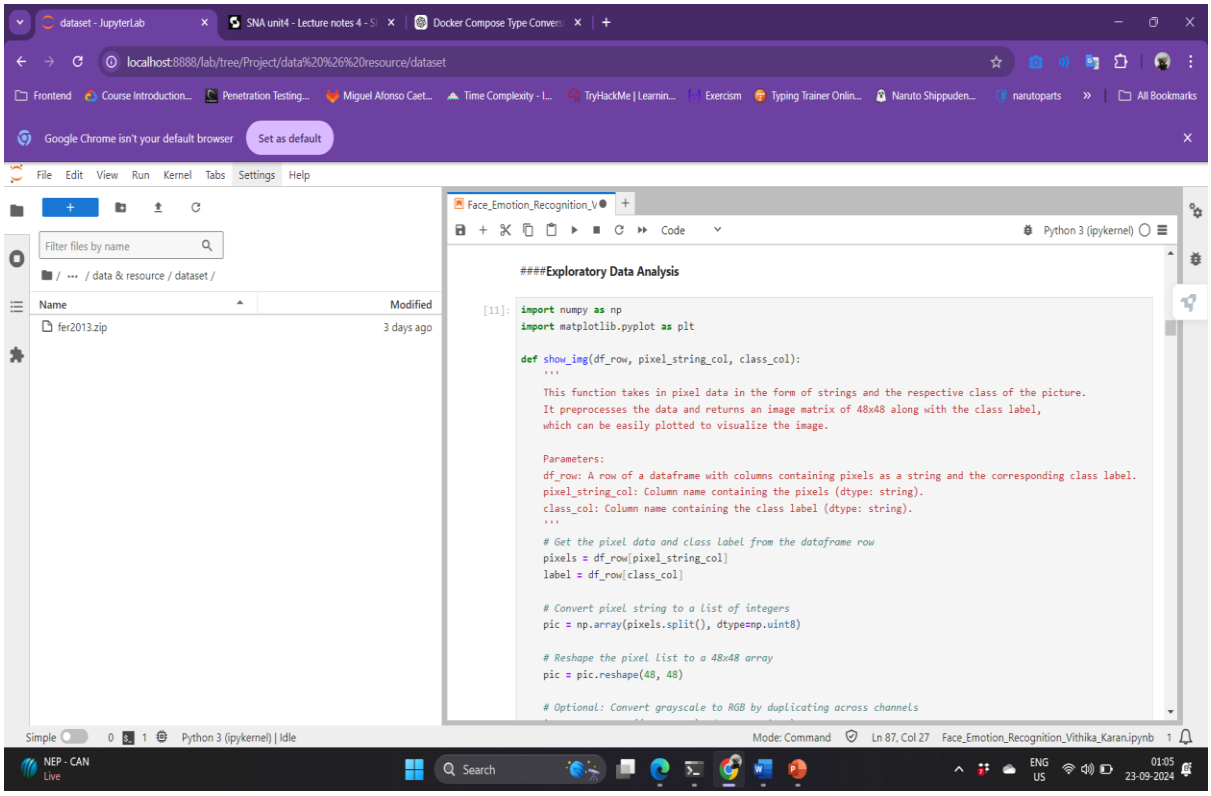
Code :

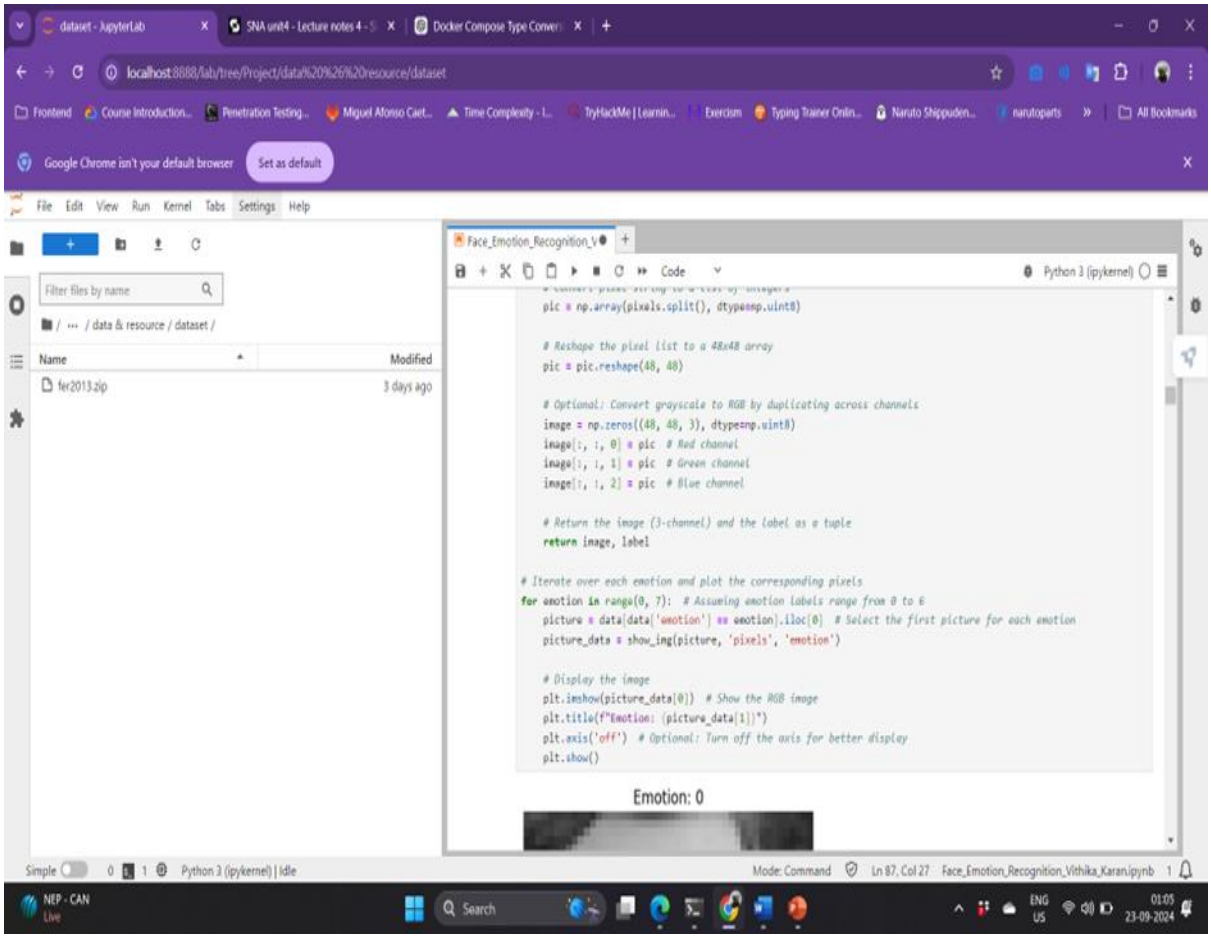
Notebook Screenshot along with complete Explanation.....within it below in detail

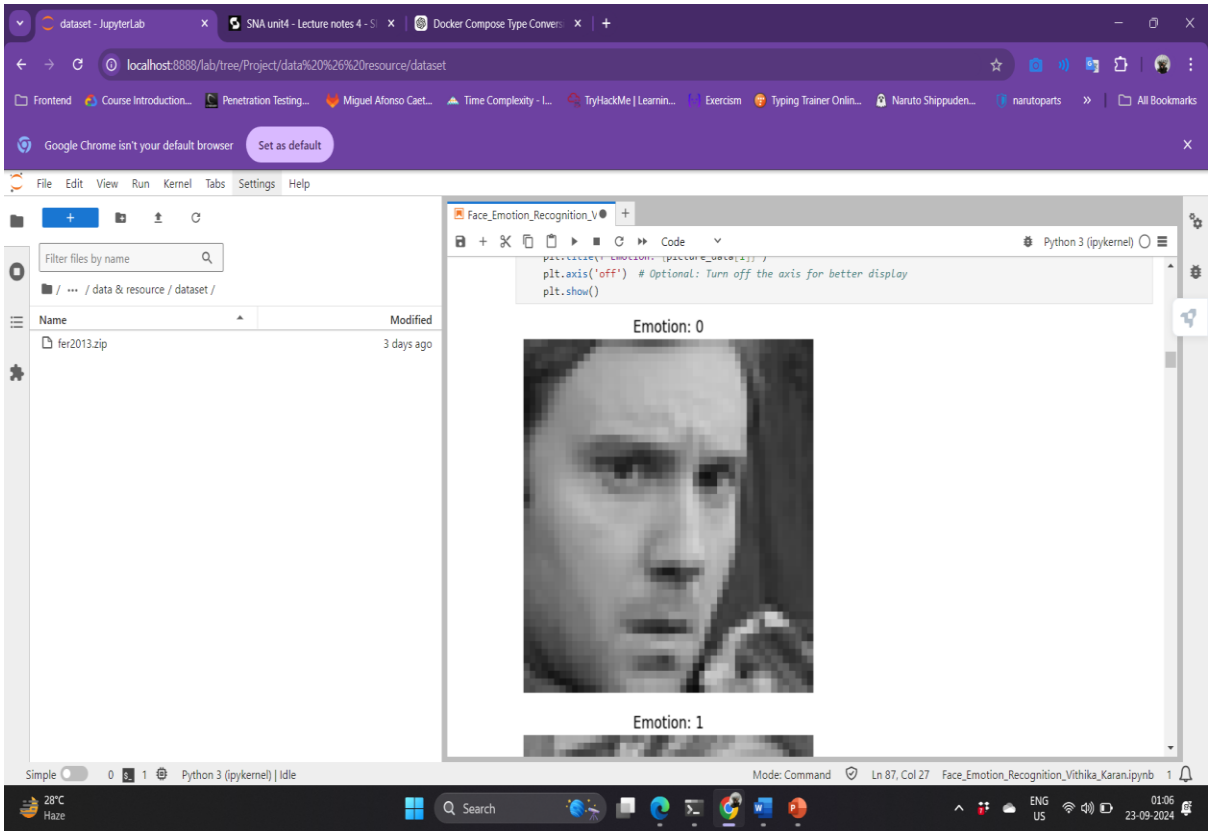


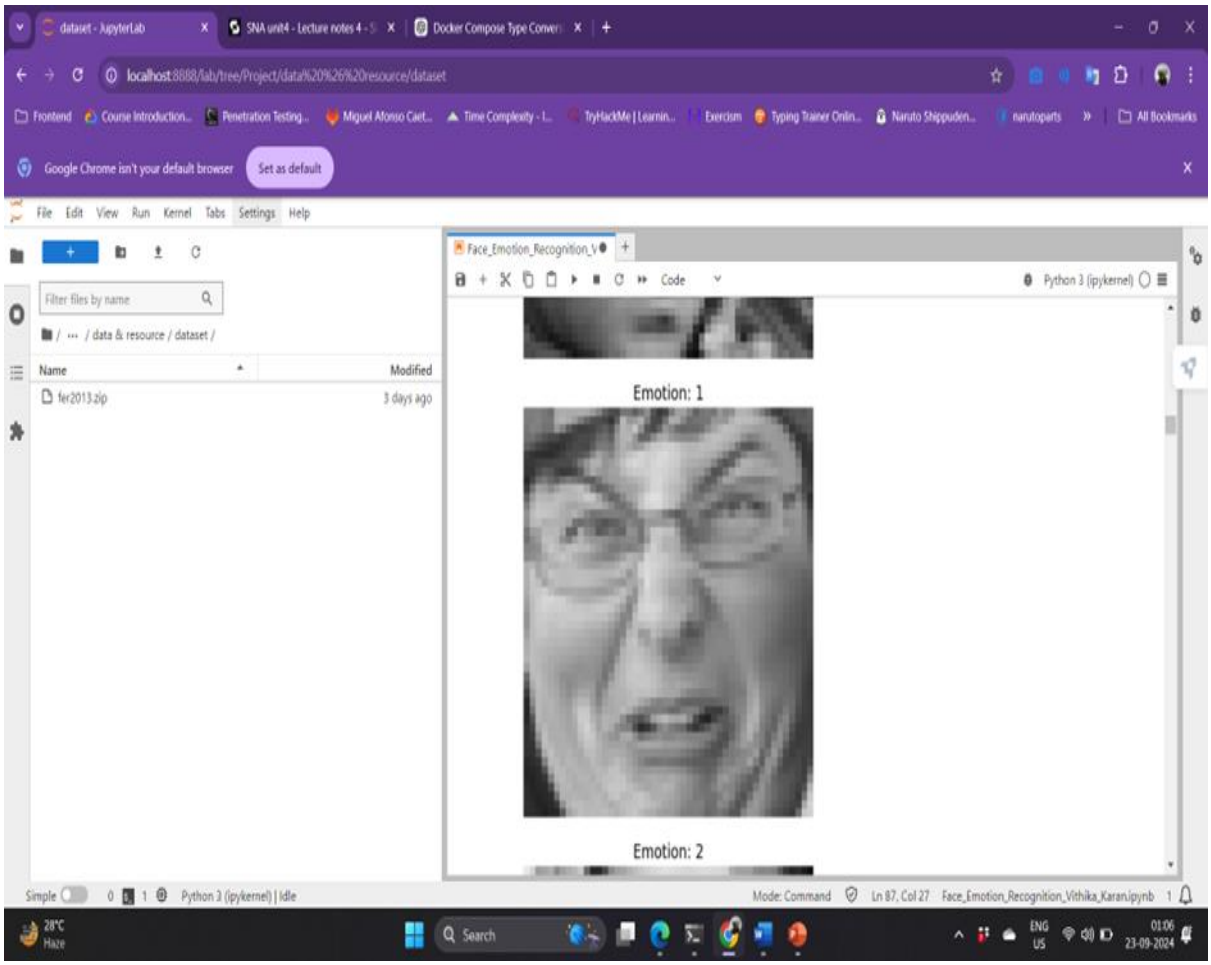

dataset - JupyterLab
SNA unit4 - Lecture notes 4 - S
Docker Compose Type Conver
localhost:8888/lab/tree/Project/data%20%26%20resource/dataset
Google Chrome isn't your default browser
Set as default
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ data & resource / dataset /
Name
Modified
fer2013.zip
3 days ago
Face_Emotion_Recognition_V
Code
Python 3 (ipykernel)
[8]: #check for null values
data.isnull().sum()
[8]: emotion 0
pixels 0
Usage 0
dtype: int64
There are no null values in the dataset.
[9]: #lets check the value counts in usage
data['Usage'].value_counts()
[9]: Usage
Training 28709
PublicTest 3589
PrivateTest 3589
Name: count, dtype: int64
[10]: #lets check the value counts in emotion column
data['emotion'].value_counts()
[10]: emotion
3 8989
6 6198
4 6077
2 5121
0 4953
Simple
Python 3 (ipykernel) | Idle
Mode: Command
Ln 87, Col 27
Face_Emotion_Recognition_Vithika_Karan.ipynb
NEP - CAN
Live
Search
ENG
US
01:05
23-09-2024

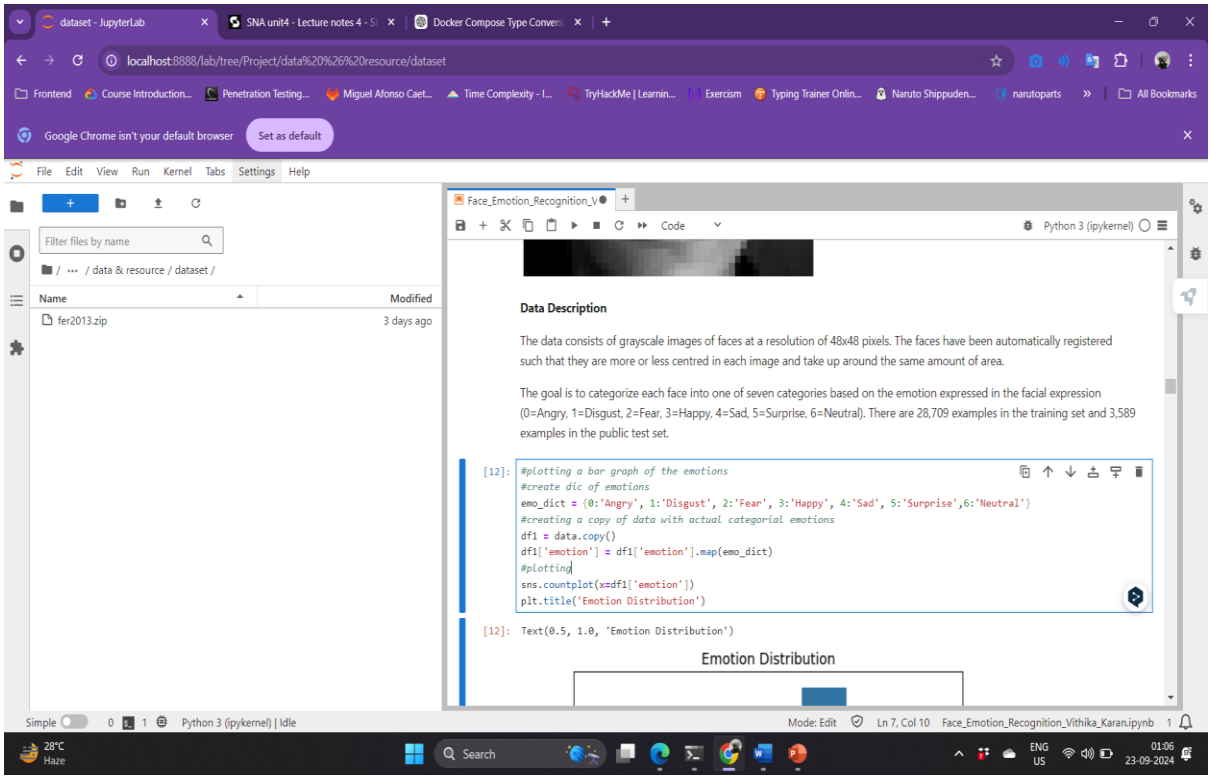

dataset - JupyterLab
SNA unit4 - Lecture notes 4 - S
Docker Compose Type Conver
localhost:8888/lab/tree/Project/data%20%26%20resource/dataset
Google Chrome isn't your default browser
Set as default
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ data & resource / dataset /
Name
Modified
fer2013.zip
3 days ago
Face_Emotion_Recognition_V
Code
Python 3 (ipykernel)
PublicTest 3589
PrivateTest 3589
Name: count, dtype: int64
[10]: #lets check the value counts in emotion column
data['emotion'].value_counts()
[10]: emotion
3 8989
6 6198
4 6077
2 5121
0 4953
5 4002
1 547
Name: count, dtype: int64
There are counts for seven emotions. Let's try to explore the images for these classes.
####Exploratory Data Analysis
[11]: import numpy as np
import matplotlib.pyplot as plt
def show_img(df_row, pixel_string_col, class_col):
'''
This function takes in pixel data in the form of strings and the respective class of the picture.
It preprocesses the data and returns an image matrix of 48x48 along with the class label,
which can be easily plotted to visualize the image.
Simple
Python 3 (ipykernel) | Idle
Mode: Command
Ln 87, Col 27
Face_Emotion_Recognition_Vithika_Karan.ipynb
NEP - CAN
Live
Search
ENG
US
01:05
23-09-2024

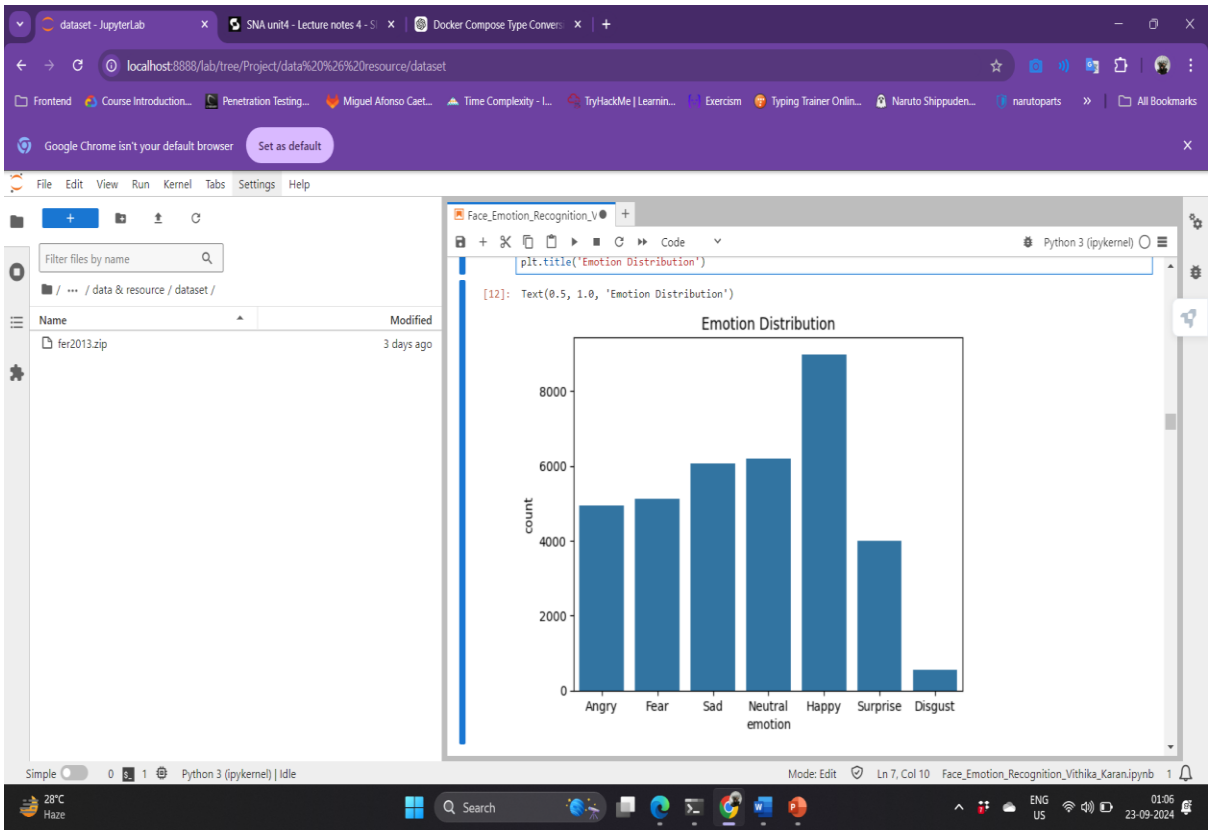

dataset - JupyterLab
SNA unit4 - Lecture notes 4 - S
Docker Compose Type Conver
localhost:8888/lab/tree/Project/data%20%26%20resource/dataset
Google Chrome isn't your default browser
Set as default
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ data & resource / dataset /
Name
Modified
fer2013.zip
3 days ago
Face_Emotion_Recognition_V
Code
Python 3 (ipykernel)
####Exploratory Data Analysis
[11]:
import numpy as np
import matplotlib.pyplot as plt
def show_img(df_row, pixel_string_col, class_col):
'''
This function takes in pixel data in the form of strings and the respective class of the picture.
It preprocesses the data and returns an image matrix of 48x48 along with the class label,
which can be easily plotted to visualize the image.
Parameters:
df_row: A row of a dataframe with columns containing pixels as a string and the corresponding class label.
pixel_string_col: Column name containing the pixels (dtype: string).
class_col: Column name containing the class label (dtype: string).
'''
# Get the pixel data and class label from the dataframe row
pixels = df_row[pixel_string_col]
label = df_row[class_col]
# Convert pixel string to a list of integers
pic = np.array(pixels.split(), dtype=np.uint8)
# Reshape the pixel list to a 48x48 array
pic = pic.reshape(48, 48)
# Optional: Convert grayscale to RGB by duplicating across channels
Simple
Python 3 (ipykernel) | Idle
Mode: Command
Ln 87, Col 27
Face_Emotion_Recognition_Vithika_Karan.ipynb
NEP - CAN
Live
Search
ENG
US
01:05
23-09-2024

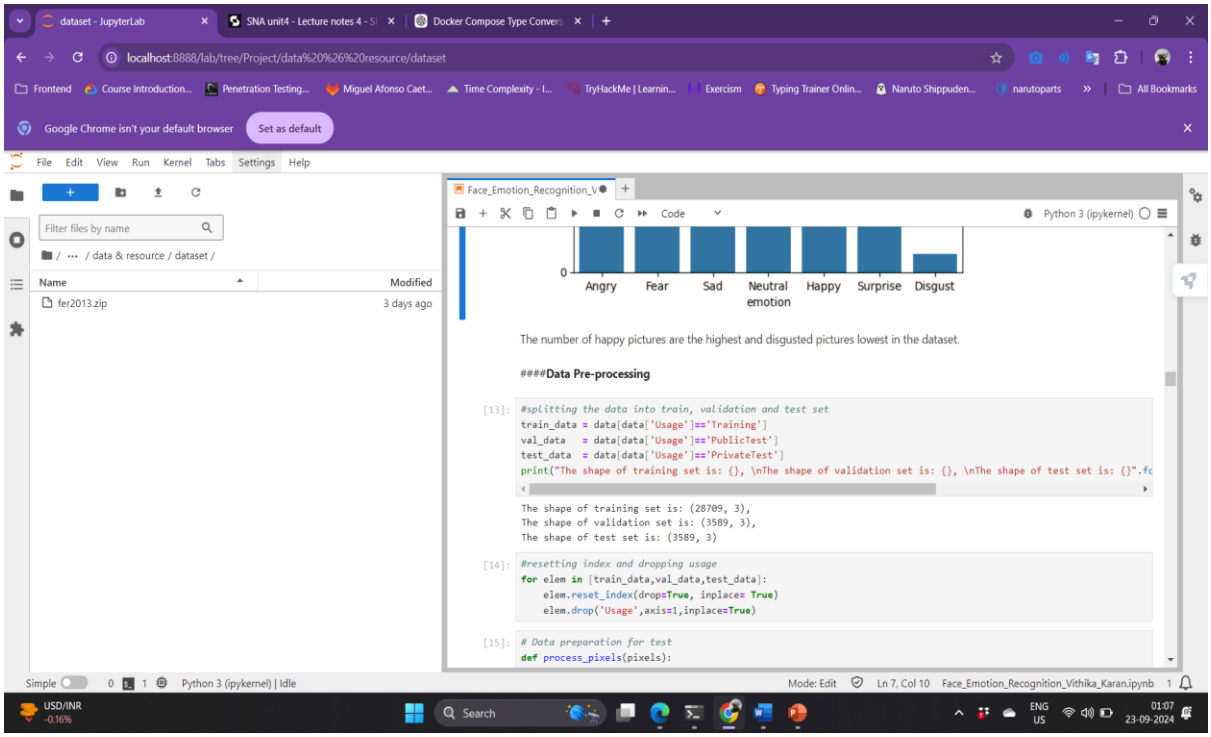

dataset - JupyterLab
SNA unit4 - Lecture notes 4 - S
Docker Compose Type Conver
localhost:8888/lab/tree/Project/data%20%26%20resource/dataset
Google Chrome isn't your default browser
Set as default
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ data & resource / dataset /
Name
Modified
fer2013.zip
3 days ago
Face_Emotion_Recognition_V
Code
Python 3 (ipykernel)
pic = np.array(pixels.split(), dtype=np.uint8)
# Reshape the pixel list to a 48x48 array
pic = pic.reshape(48, 48)
# Optional: Convert grayscale to RGB by duplicating across channels
image = np.zeros((48, 48, 3), dtype=np.uint8)
image[:, :, 0] = pic # Red channel
image[:, :, 1] = pic # Green channel
image[:, :, 2] = pic # Blue channel
Return the image (3-channel) and the label as a tuple
return image, label
Iterate over each emotion and plot the corresponding pixels
for emotion in range(0, 7): # Assuming emotion labels range from 0 to 6
picture = data[data['emotion'] == emotion].iloc[0] # Select the first picture for each emotion
picture_data = show_img(picture, 'pixels', 'emotion')
# Display the image
plt.imshow(picture_data[0]) # Show the RGB image
plt.title(f"Emotion: {picture_data[1]}")
plt.axis('off') # Optional: Turn off the axis for better display
plt.show()
Emotion: 0
Simple
Python 3 (ipykernel) | Idle
Mode: Command
Ln 87, Col 27
Face_Emotion_Recognition_Vithika_Karan.ipynb
NEP - CAN
Live
Search
ENG
US
01:05
23-09-2024

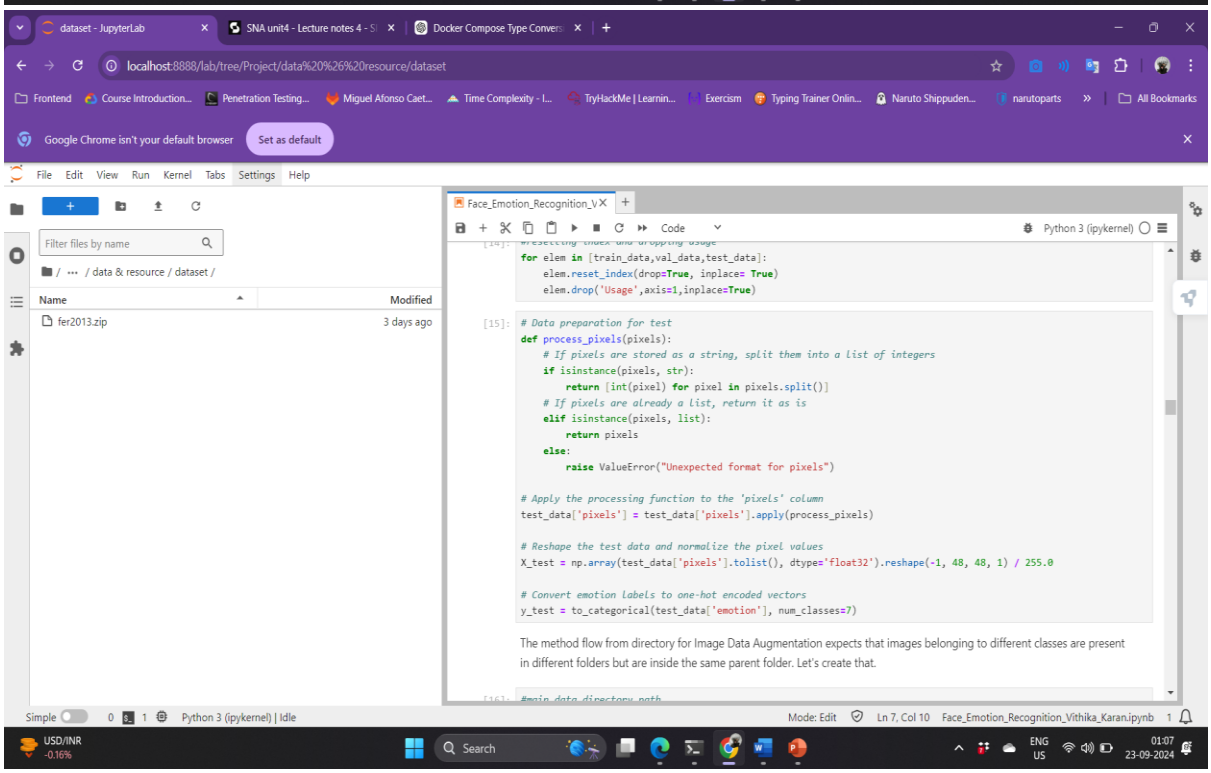

dataset - JupyterLab
SNA unit4 - Lecture notes 4 - S
Docker Compose Type Conver
localhost:8888/lab/tree/Project/data%20%26%20resource/dataset
Google Chrome isn't your default browser
Set as default
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ data & resource / dataset /
Name
Modified
fer2013.zip
3 days ago
Face_Emotion_Recognition_V
Code
Python 3 (ipykernel)
plt.axis('off') # Optional: Turn off the axis for better display
plt.show()
Emotion: 0
Emotion: 1
Simple
Python 3 (ipykernel) | Idle
Mode: Command
Ln 87, Col 27
Face_Emotion_Recognition_Vithika_Karan.ipynb
28°C
Haze
Search
ENG
US
01:06
23-09-2024

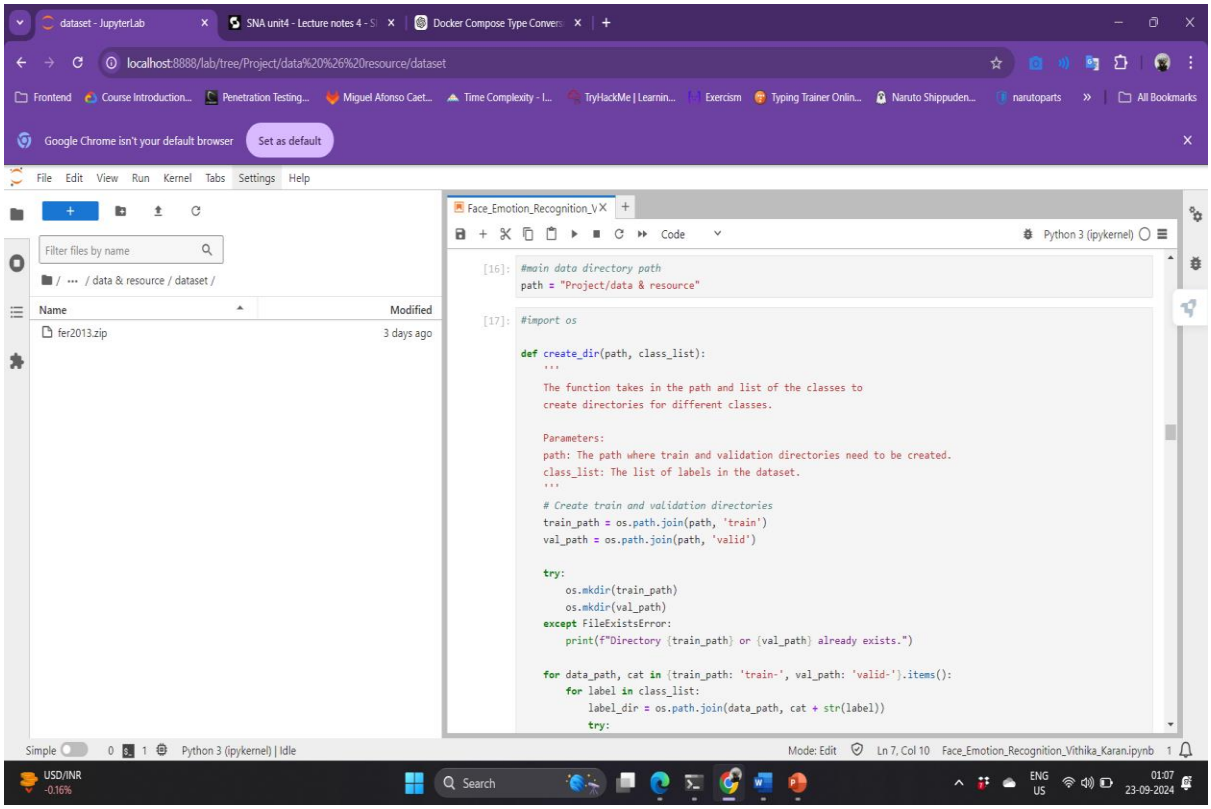

dataset - JupyterLab
SNA unit4 - Lecture notes 4 - S
Docker Compose Type Conver
localhost:8888/lab/tree/Project/data%20%26%20resource/dataset
Google Chrome isn't your default browser
Set as default
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ data & resource / dataset /
Name
Modified
fer2013.zip
3 days ago
Face_Emotion_Recognition_V
Code
Python 3 (ipykernel)
Emotion: 1
Emotion: 2
Simple
Python 3 (ipykernel) | Idle
Mode: Command
Ln 87, Col 27
Face_Emotion_Recognition_Vithika_Karan.ipynb
28°C
Haze
Search
ENG
US
01:06
23-09-2024

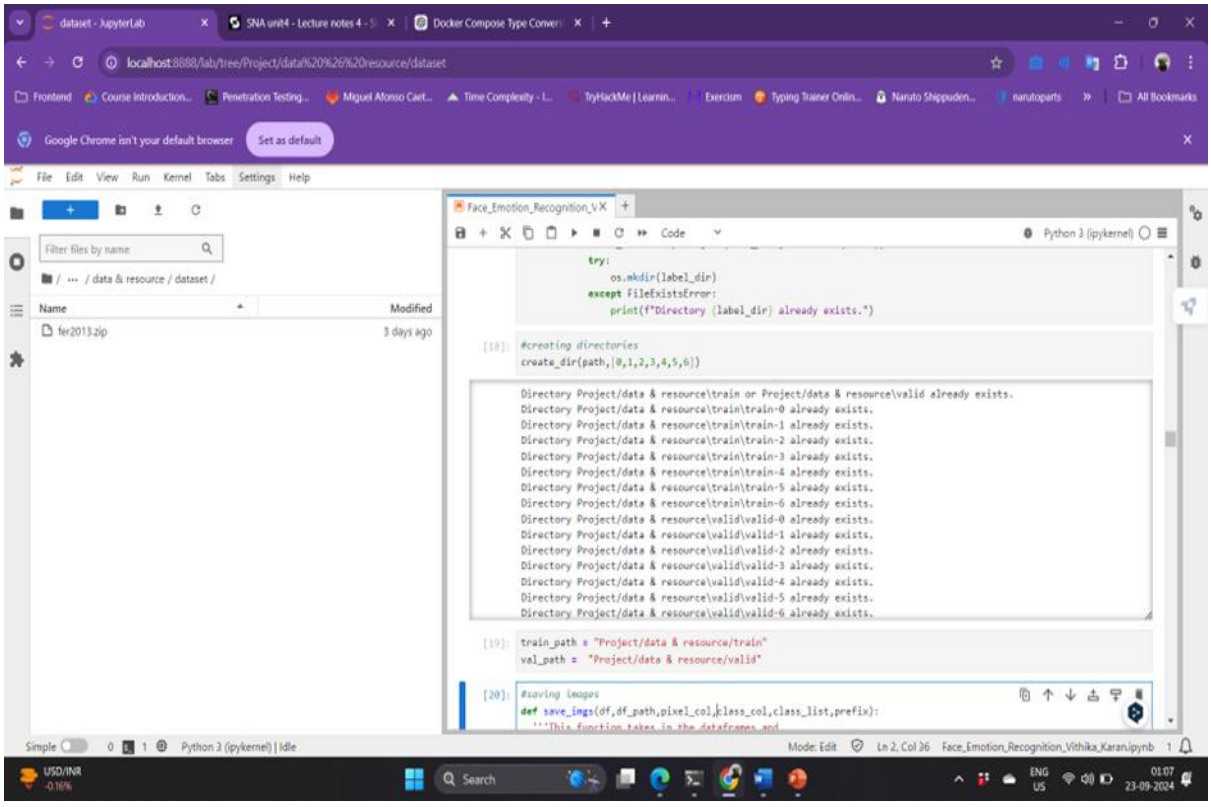

dataset - JupyterLab
SNA unit4 - Lecture notes 4 - S
Docker Compose Type Convers
localhost:8888/lab/tree/Project/data%20%26%20resource/dataset
Google Chrome isn't your default browser
Set as default
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ data & resource / dataset /
Name
Modified
fer2013.zip
3 days ago
Face_Emotion_Recognition_V
Code
Python 3 (ipykernel)
Data Description
The data consists of grayscale images of faces at a resolution of 48x48 pixels. The faces have been automatically registered such that they are more or less centred in each image and take up around the same amount of area.
The goal is to categorize each face into one of seven categories based on the emotion expressed in the facial expression (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral). There are 28,709 examples in the training set and 3,589 examples in the public test set.
[12]:
#plotting a bar graph of the emotions
#create dic of emotions
emo_dict = {0:'Angry', 1:'Disgust', 2:'Fear', 3:'Happy', 4:'Sad', 5:'Surprise',6:'Neutral'}
#creating a copy of data with actual categorial emotions
df1 = data.copy()
df1['emotion'] = df1['emotion'].map(emo_dict)
#plotting
sns.countplot(x=df1['emotion'])
plt.title('Emotion Distribution')
[12]:
Text(0.5, 1.0, 'Emotion Distribution')
Emotion Distribution
Simple
Python 3 (ipykernel) | Idle
Mode: Edit
Ln 7, Col 10
Face_Emotion_Recognition_Vithika_Karan.ipynb
28°C
Haze
Search
ENG
US
01:06
23-09-2024

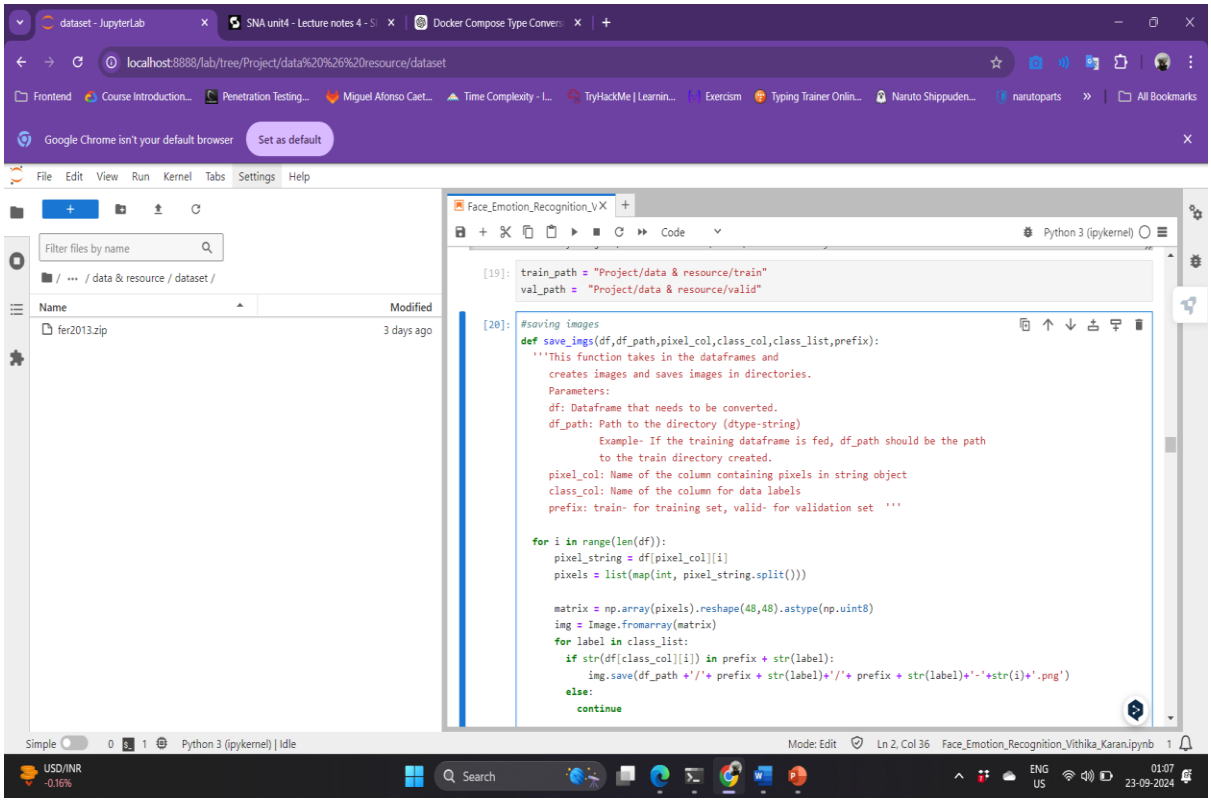

dataset - JupyterLab
SNA unit4 - Lecture notes 4 - S
Docker Compose Type Convers
localhost:8888/lab/tree/Project/data%20%26%20resource/dataset
Google Chrome isn't your default browser
Set as default
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ data & resource / dataset /
Name
Modified
fer2013.zip
3 days ago
Face_Emotion_Recognition_V
Code
Python 3 (ipykernel)
plt.title('Emotion Distribution')
[12]:
Text(0.5, 1.0, 'Emotion Distribution')
Emotion Distribution
count
8000
6000
4000
2000
0
Angry
Fear
Sad
Neutral
Happy
Surprise
Disgust
emotion
Simple
Python 3 (ipykernel) | Idle
Mode: Edit
Ln 7, Col 10
Face_Emotion_Recognition_Vithika_Karan.ipynb
28°C
Haze
Search
ENG
US
01:06
23-09-2024

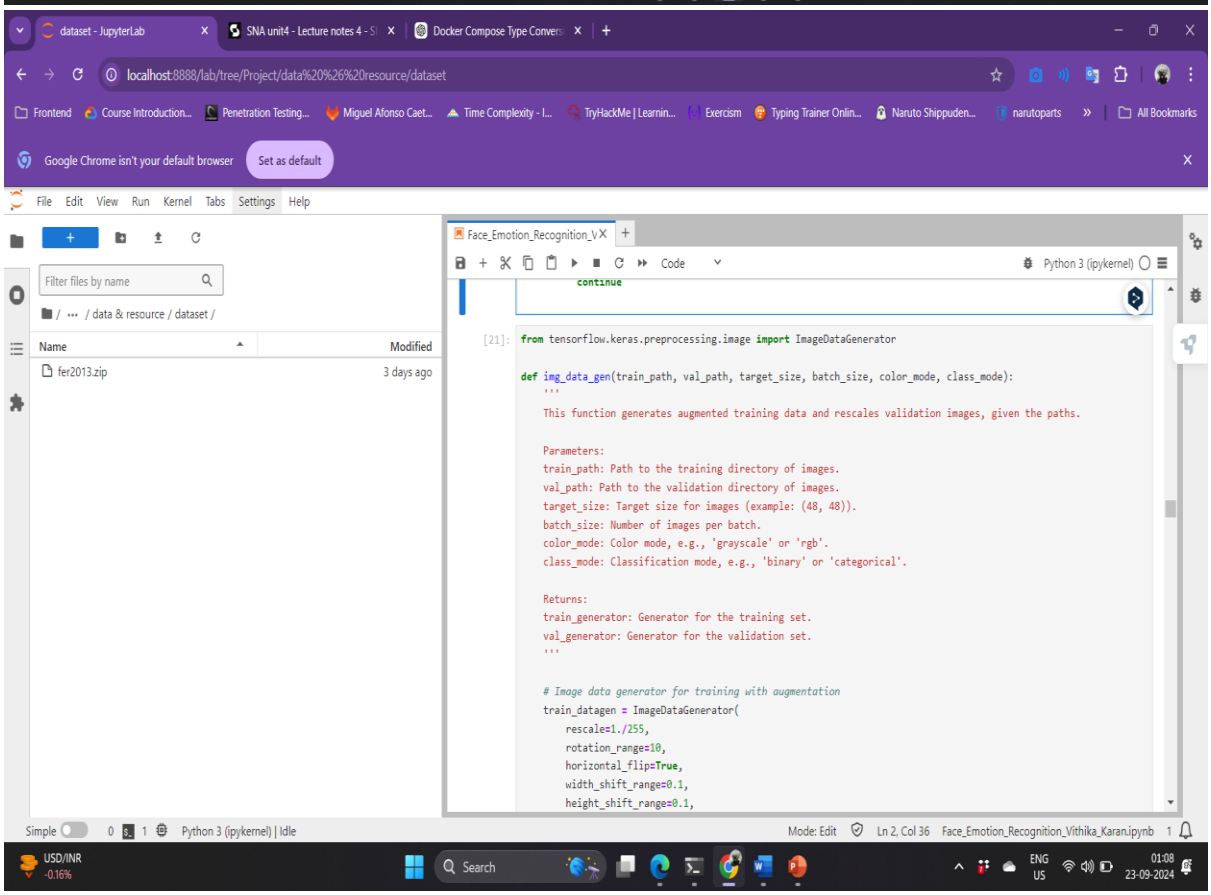

Angry
Fear
Sad
Neutral
Happy
Surprise
Disgust
emotion
0
The number of happy pictures are the highest and disgusted pictures lowest in the dataset.
####Data Pre-processing
#splitting the data into train, validation and test set
train_data = data[data['Usage']=='Training']
val_data = data[data['Usage']=='PublicTest']
test_data = data[data['Usage']=='PrivateTest']
print("The shape of training set is: {}, \nThe shape of validation set is: {}, \nThe shape of test set is: {}".fo
The shape of training set is: (28709, 3),
The shape of validation set is: (3589, 3),
The shape of test set is: (3589, 3)
#resetting index and dropping usage
for elem in [train_data,val_data,test_data]:
elem.reset_index(drop=True, inplace= True)
elem.drop('Usage',axis=1,inplace=True)
# Data preparation for test
def process_pixels(pixels):

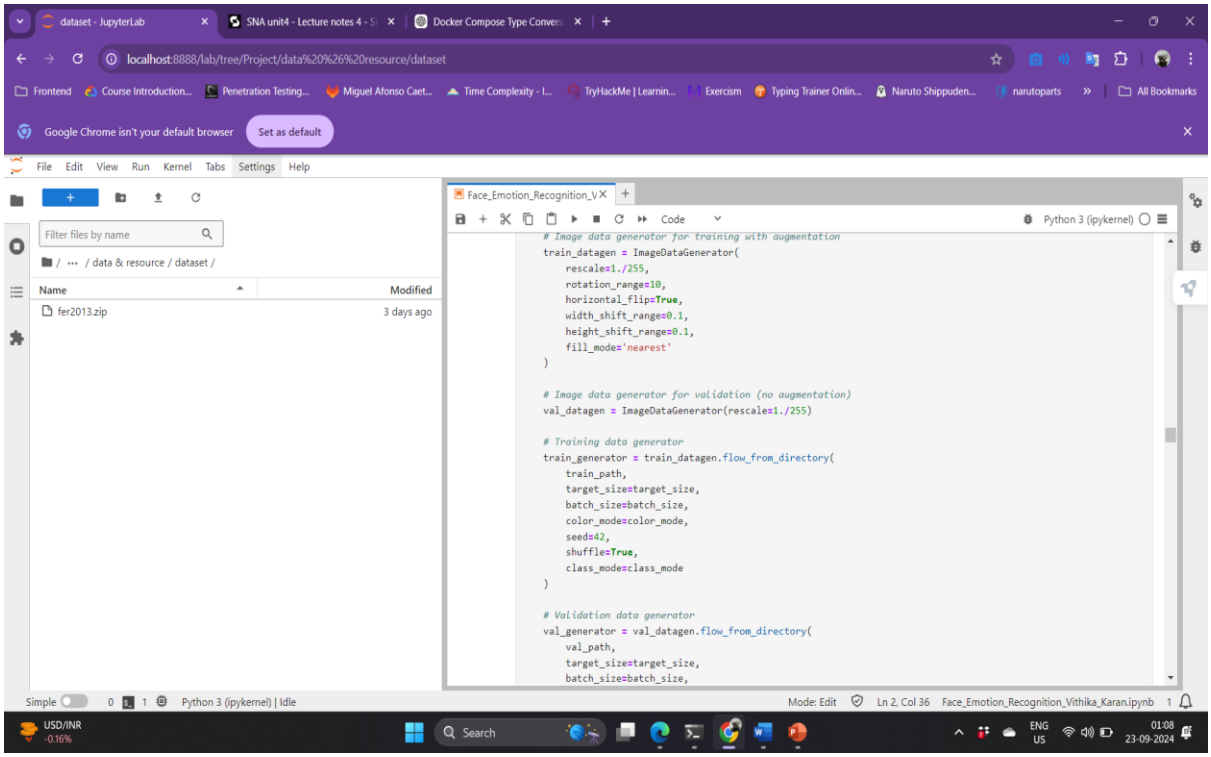

for elem in [train_data,val_data,test_data]:
elem.reset_index(drop=True, inplace= True)
elem.drop('Usage',axis=1,inplace=True)
# Data preparation for test
def process_pixels(pixels):
# If pixels are stored as a string, split them into a list of integers
if isinstance(pixels, str):
return [int(pixel) for pixel in pixels.split()]
# If pixels are already a list, return it as is
elif isinstance(pixels, list):
return pixels
else:
raise ValueError("Unexpected format for pixels")
Apply the processing function to the 'pixels' column
test_data['pixels'] = test_data['pixels'].apply(process_pixels)
# Reshape the test data and normalize the pixel values
X_test = np.array(test_data['pixels'].tolist(), dtype='float32').reshape(-1, 48, 48, 1) / 255.0
# Convert emotion labels to one-hot encoded vectors
y_test = to_categorical(test_data['emotion'], num_classes=7)
The method flow from directory for Image Data Augmentation expects that images belonging to different classes are present in different folders but are inside the same parent folder. Let's create that.

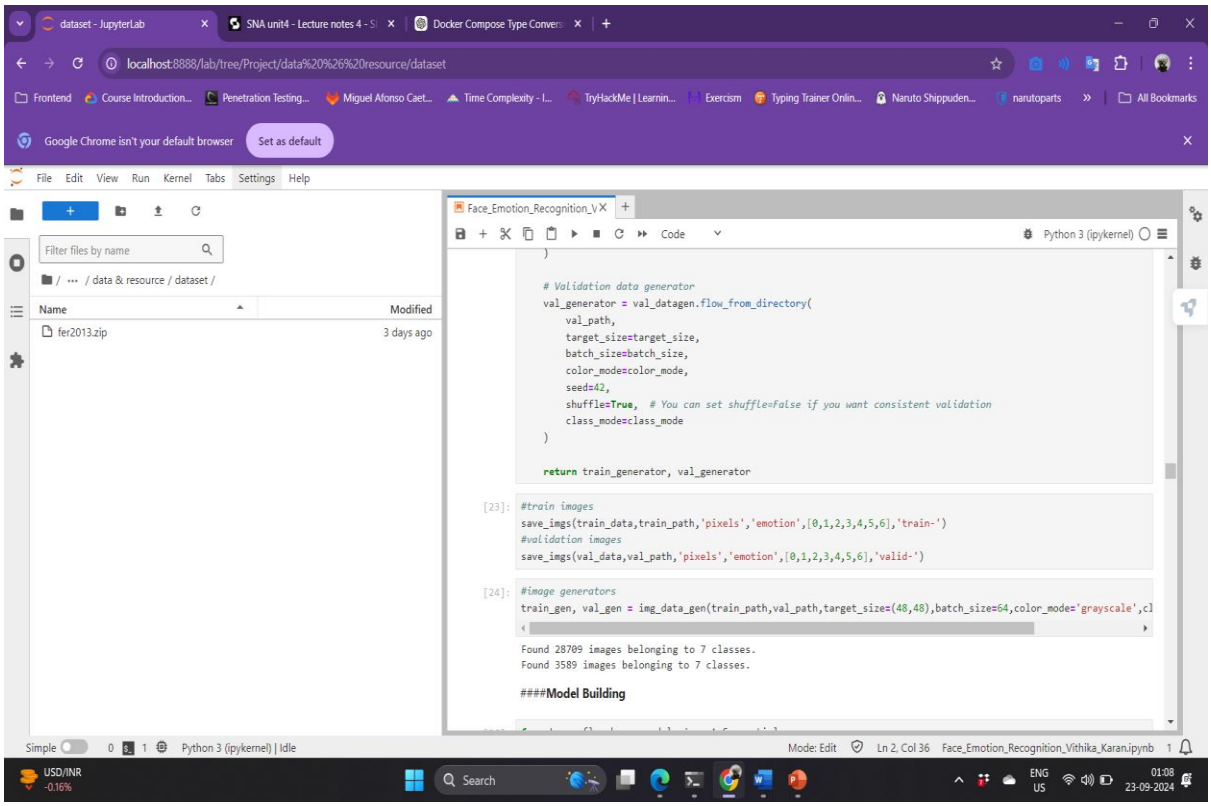

dataset - JupyterLab
SNA unit4 - Lecture notes 4 - S
Docker Compose Type Conver
localhost:8888/lab/tree/Project/data%20%26%20resource/dataset
Google Chrome isn't your default browser
Set as default
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ data & resource / dataset /
Name
Modified
fer2013.zip
3 days ago
Face_Emotion_Recognition_V
Code
Python 3 (ipykernel)
[16]:
#main data directory path
path = "Project/data & resource"
[17]:
#import os
def create_dir(path, class_list):
'''
The function takes in the path and list of the classes to
create directories for different classes.
Parameters:
path: The path where train and validation directories need to be created.
class_list: The list of labels in the dataset.
'''
# Create train and validation directories
train_path = os.path.join(path, 'train')
val_path = os.path.join(path, 'valid')
try:
os.mkdir(train_path)
os.mkdir(val_path)
except FileExistsError:
print(f"Directory {train_path} or {val_path} already exists.")
for data_path, cat in {train_path: 'train-', val_path: 'valid-'}.items():
for label in class_list:
label_dir = os.path.join(data_path, cat + str(label))
try:
Simple
Python 3 (ipykernel) | Idle
Mode: Edit
Ln 7, Col 10
Face_Emotion_Recognition_Vithika_Karan.ipynb
USD/INR
-0.16%
Search
ENG
US
01:07
23-09-2024

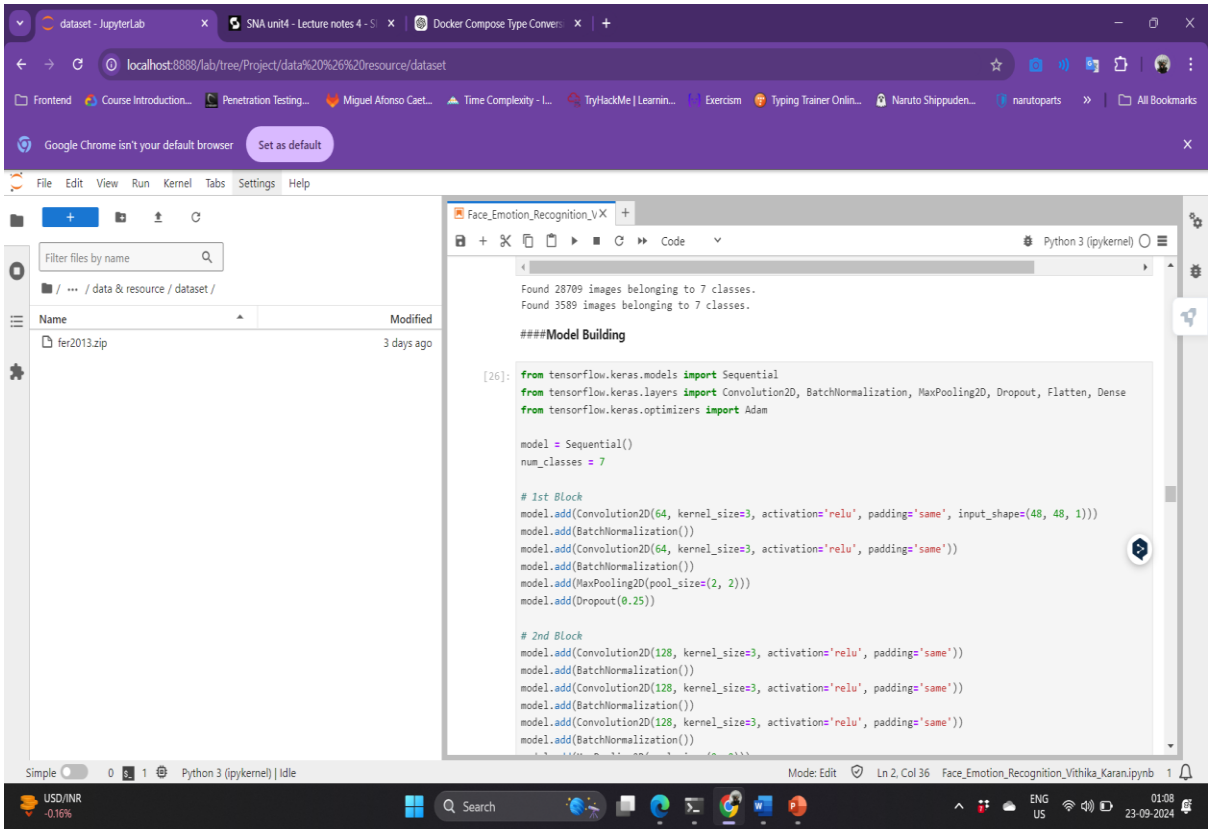

dataset - JupyterLab
SNA unit4 - Lecture notes 4 - S
Docker Compose Type Conver
localhost:8888/lab/tree/Project/data%20%26%20resource/dataset
Google Chrome isn't your default browser
Set as default
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ data & resource / dataset /
Name
Modified
fer2013.zip
3 days ago
Face_Emotion_Recognition_V
Code
Python 3 (ipykernel)
try:
os.mkdir(label_dir)
except FileExistsError:
print(f"Directory {label_dir} already exists.")
[18]:
#creating directories
create_dir(path,[0,1,2,3,4,5,6])
Directory Project/data & resource\train or Project/data & resource\valid already exists.
Directory Project/data & resource\train\train-0 already exists.
Directory Project/data & resource\train\train-1 already exists.
Directory Project/data & resource\train\train-2 already exists.
Directory Project/data & resource\train\train-3 already exists.
Directory Project/data & resource\train\train-4 already exists.
Directory Project/data & resource\train\train-5 already exists.
Directory Project/data & resource\train\train-6 already exists.
Directory Project/data & resource\valid\valid-0 already exists.
Directory Project/data & resource\valid\valid-1 already exists.
Directory Project/data & resource\valid\valid-2 already exists.
Directory Project/data & resource\valid\valid-3 already exists.
Directory Project/data & resource\valid\valid-4 already exists.
Directory Project/data & resource\valid\valid-5 already exists.
Directory Project/data & resource\valid\valid-6 already exists.
[19]:
train_path = "Project/data & resource/train"
val_path = "Project/data & resource/valid"
[20]:
#saving images
def save_imgs(df,df_path,pixel_col,klass_col,class_list,prefix):
'''This function takes in the dataframes and
Simple
Python 3 (ipykernel) | Idle
Mode: Edit
Ln 2, Col 36
Face_Emotion_Recognition_Vithika_Karan.ipynb
USD/INR
-0.16%
Search
ENG
US
01:07
23-09-2024

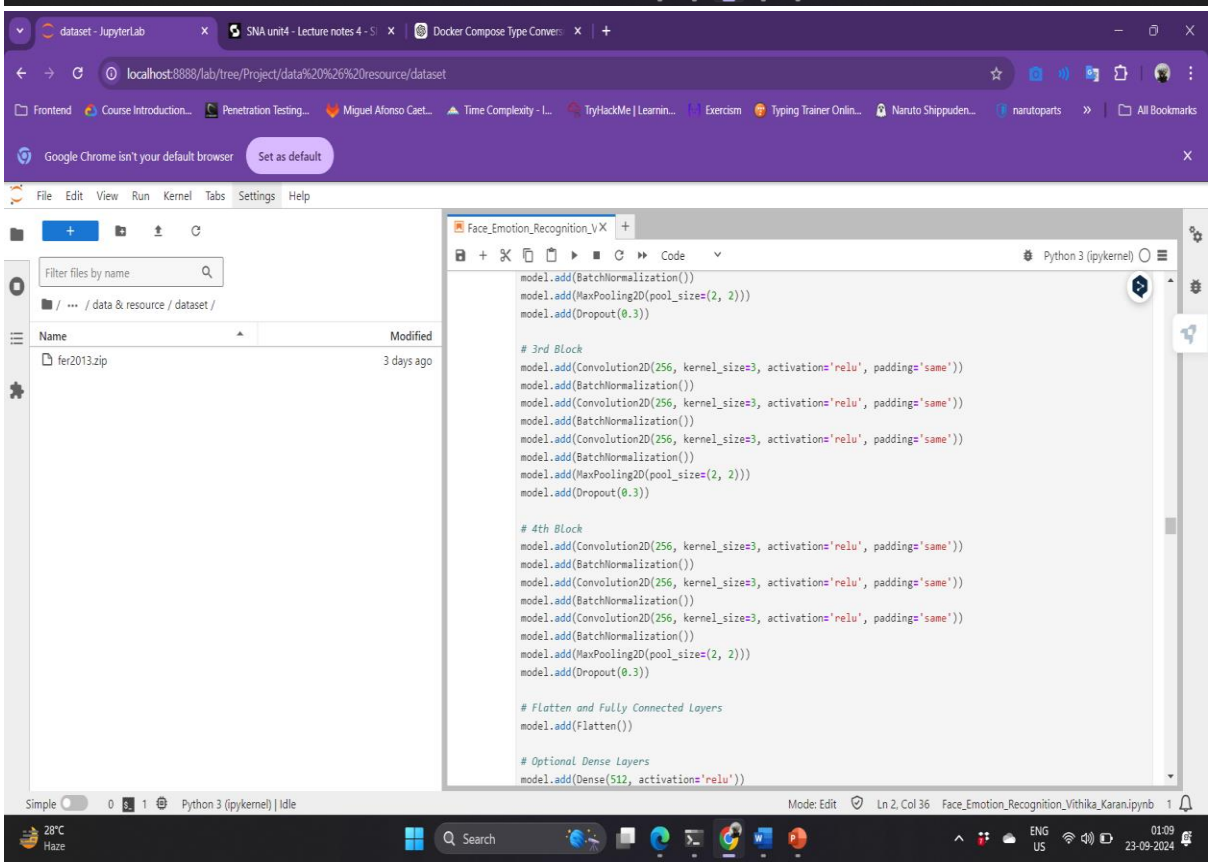

dataset - JupyterLab
SNA unit4 - Lecture notes 4 - S
Docker Compose Type Convers
localhost:8888/lab/tree/Project/data%20%26%20resource/dataset
Google Chrome isn't your default browser
Set as default
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ data & resource / dataset /
Name
Modified
fer2013.zip
3 days ago
Face_Emotion_Recognition_V
Code
Python 3 (ipykernel)
[19]:
train_path = "Project/data & resource/train"
val_path = "Project/data & resource/valid"
[20]:
#saving images
def save_imgs(df,df_path,pixel_col,class_col,class_list,prefix):
'''This function takes in the dataframes and
creates images and saves images in directories.
Parameters:
df: Dataframe that needs to be converted.
df_path: Path to the directory (dtype-string)
Example- If the training dataframe is fed, df_path should be the path
to the train directory created.
pixel_col: Name of the column containing pixels in string object
class_col: Name of the column for data labels
prefix: train- for training set, valid- for validation set '''
for i in range(len(df)):
pixel_string = df[pixel_col][i]
pixels = list(map(int, pixel_string.split()))
matrix = np.array(pixels).reshape(48,48).astype(np.uint8)
img = Image.fromarray(matrix)
for label in class_list:
if str(df[class_col][i]) in prefix + str(label):
img.save(df_path +'/'+ prefix + str(label)+'/'+ prefix + str(label)+'-'+str(i)+'.png')
else:
continue
Simple
Python 3 (ipykernel) | Idle
Mode: Edit
Ln 2, Col 36
Face_Emotion_Recognition_Vithika_Karan.ipynb
USD/INR
-0.16%
Search
ENG
US
01:07
23-09-2024

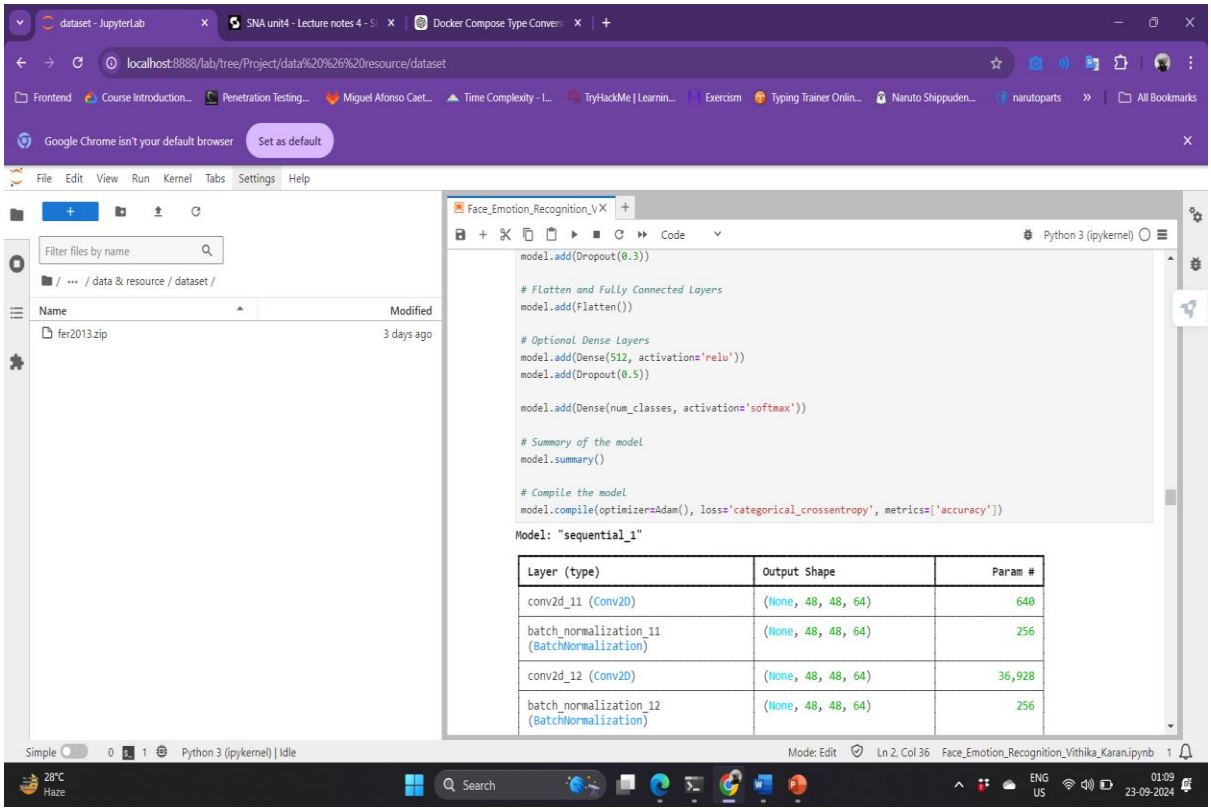

dataset - JupyterLab
SNA unit4 - Lecture notes 4 - S
Docker Compose Type Convers
localhost:8888/lab/tree/Project/data%20%26%20resource/dataset
Google Chrome isn't your default browser
Set as default
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ data & resource / dataset /
Name
Modified
fer2013.zip
3 days ago
Face_Emotion_Recognition_V
Code
Python 3 (ipykernel)
continue
[21]:
from tensorflow.keras.preprocessing.image import ImageDataGenerator
def img_data_gen(train_path, val_path, target_size, batch_size, color_mode, class_mode):
'''
This function generates augmented training data and rescales validation images, given the paths.
Parameters:
train_path: Path to the training directory of images.
val_path: Path to the validation directory of images.
target_size: Target size for images (example: (48, 48)).
batch_size: Number of images per batch.
color_mode: Color mode, e.g., 'grayscale' or 'rgb'.
class_mode: Classification mode, e.g., 'binary' or 'categorical'.
Returns:
train_generator: Generator for the training set.
val_generator: Generator for the validation set.
'''
# Image data generator for training with augmentation
train_datagen = ImageDataGenerator(
rescale=1./255,
rotation_range=10,
horizontal_flip=True,
width_shift_range=0.1,
height_shift_range=0.1,
Simple
Python 3 (ipykernel) | Idle
Mode: Edit
Ln 2, Col 36
Face_Emotion_Recognition_Vithika_Karan.ipynb
USD/INR
-0.16%
Search
ENG
US
01:08
23-09-2024

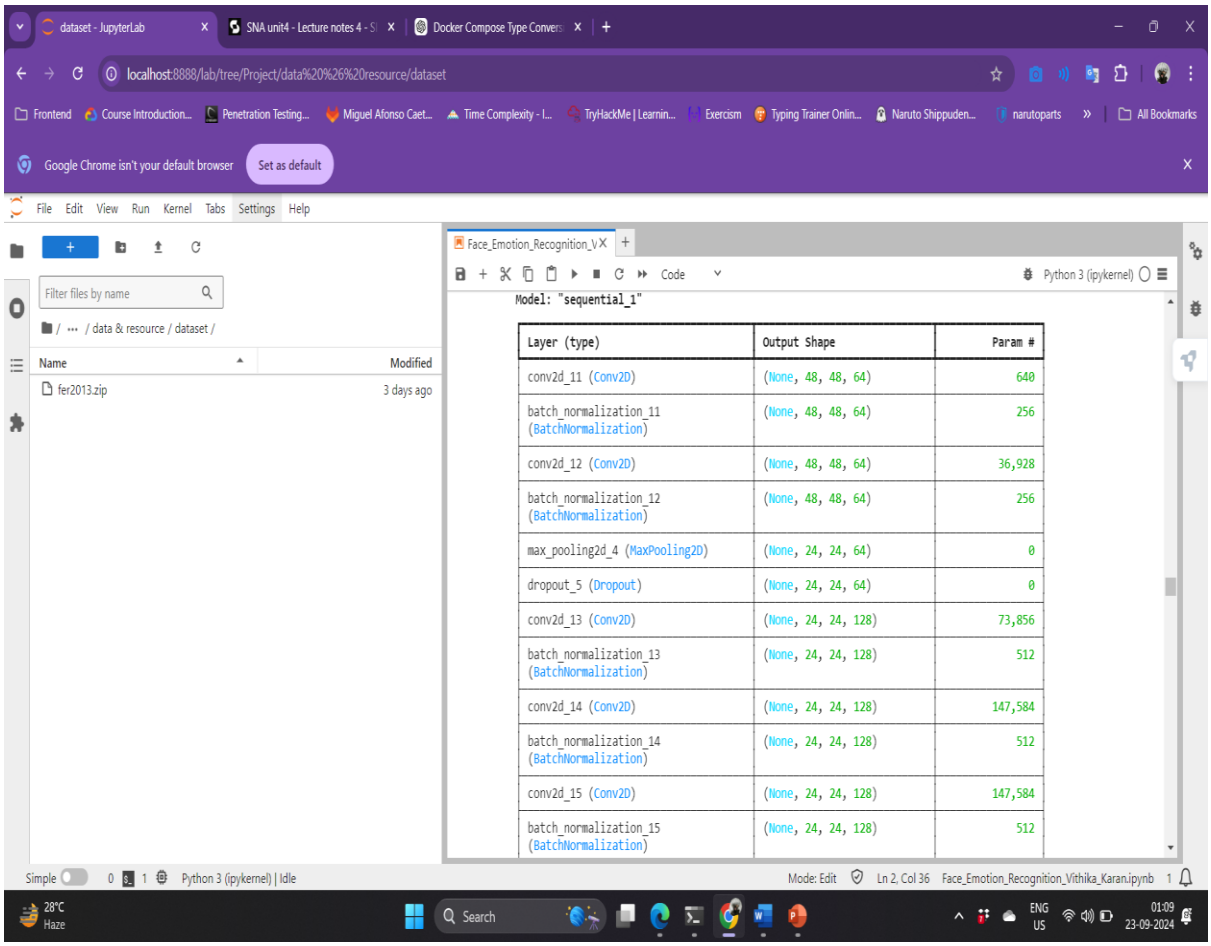

dataset - JupyterLab
SNA unit4 - Lecture notes 4 - Si
Docker Compose Type Convers
localhost:8888/lab/tree/Project/data%20%26%20resource/dataset
Google Chrome isn't your default browser
Set as default
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ data & resource / dataset /
Name
Modified
fer2013.zip
3 days ago
Face_Emotion_Recognition_V
Code
Python 3 (ipykernel)
# Image data generator for training with augmentation
train_datagen = ImageDataGenerator(
rescale=1./255,
rotation_range=10,
horizontal_flip=True,
width_shift_range=0.1,
height_shift_range=0.1,
fill_mode='nearest'
)
# Image data generator for validation (no augmentation)
val_datagen = ImageDataGenerator(rescale=1./255)
# Training data generator
train_generator = train_datagen.flow_from_directory(
train_path,
target_size=target_size,
batch_size=batch_size,
color_mode=color_mode,
seed=42,
shuffle=True,
class_mode=class_mode
)
# Validation data generator
val_generator = val_datagen.flow_from_directory(
val_path,
target_size=target_size,
batch_size=batch_size,
Simple
Python 3 (ipykernel) | Idle
Mode: Edit
Ln 2, Col 36
Face_Emotion_Recognition_Vithika_Karan.ipynb
USD/INR
-0.16%
Search
ENG
US
01:08
23-09-2024

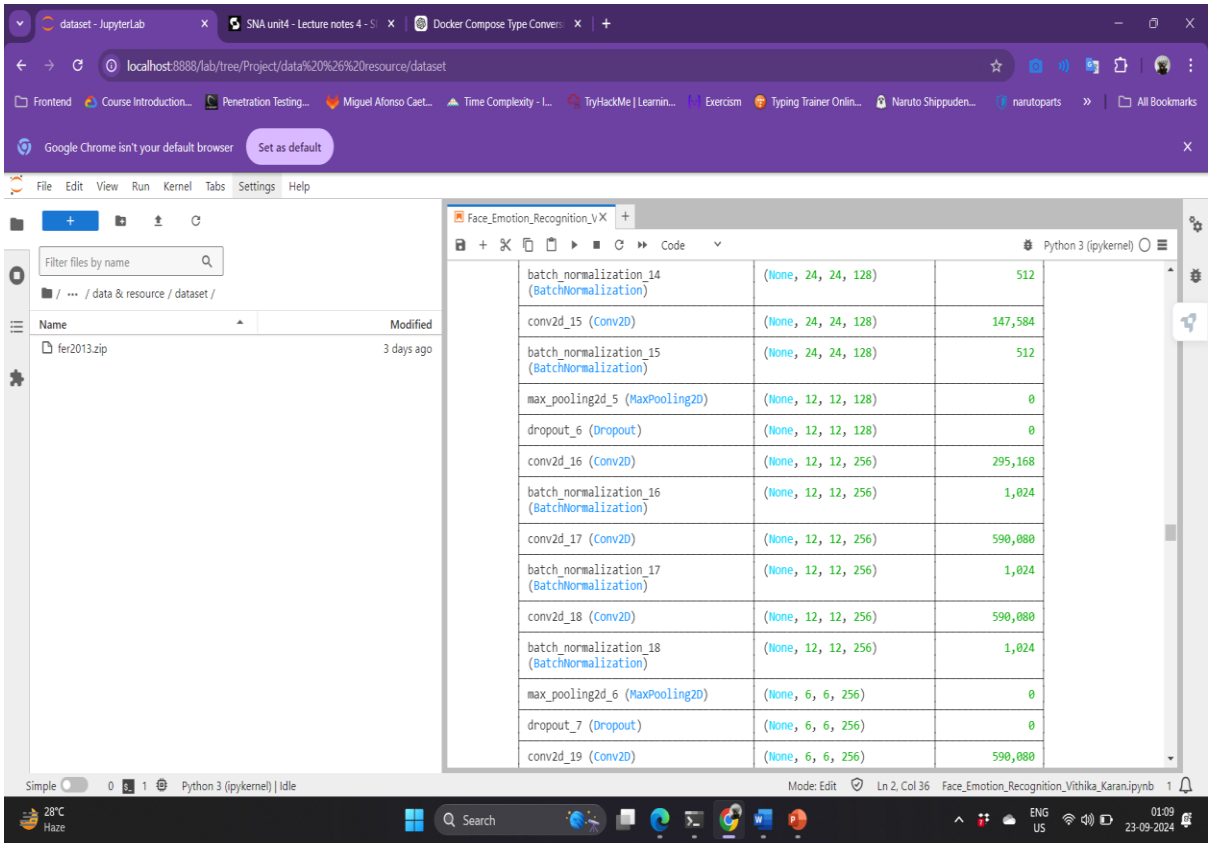

dataset - JupyterLab
SNA unit4 - Lecture notes 4 - Si
Docker Compose Type Convers
localhost:8888/lab/tree/Project/data%20%26%20resource/dataset
Google Chrome isn't your default browser
Set as default
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ data & resource / dataset /
Name
Modified
fer2013.zip
3 days ago
Face_Emotion_Recognition_V
Code
Python 3 (ipykernel)
)
# Validation data generator
val_generator = val_datagen.flow_from_directory(
val_path,
target_size=target_size,
batch_size=batch_size,
color_mode=color_mode,
seed=42,
shuffle=True, # You can set shuffle=False if you want consistent validation
class_mode=class_mode
)
return train_generator, val_generator
[23]:
#train images
save_imgs(train_data,train_path,'pixels','emotion',[0,1,2,3,4,5,6],'train-')
#validation images
save_imgs(val_data,val_path,'pixels','emotion',[0,1,2,3,4,5,6],'valid-')
[24]:
#image generators
train_gen, val_gen = img_data_gen(train_path,val_path,target_size=(48,48),batch_size=64,color_mode='grayscale',cl
Found 28709 images belonging to 7 classes.
Found 3589 images belonging to 7 classes.
####Model Building
Simple
Python 3 (ipykernel) | Idle
Mode: Edit
Ln 2, Col 36
Face_Emotion_Recognition_Vithika_Karan.ipynb
USD/INR
-0.16%
Search
ENG
US
01:08
23-09-2024

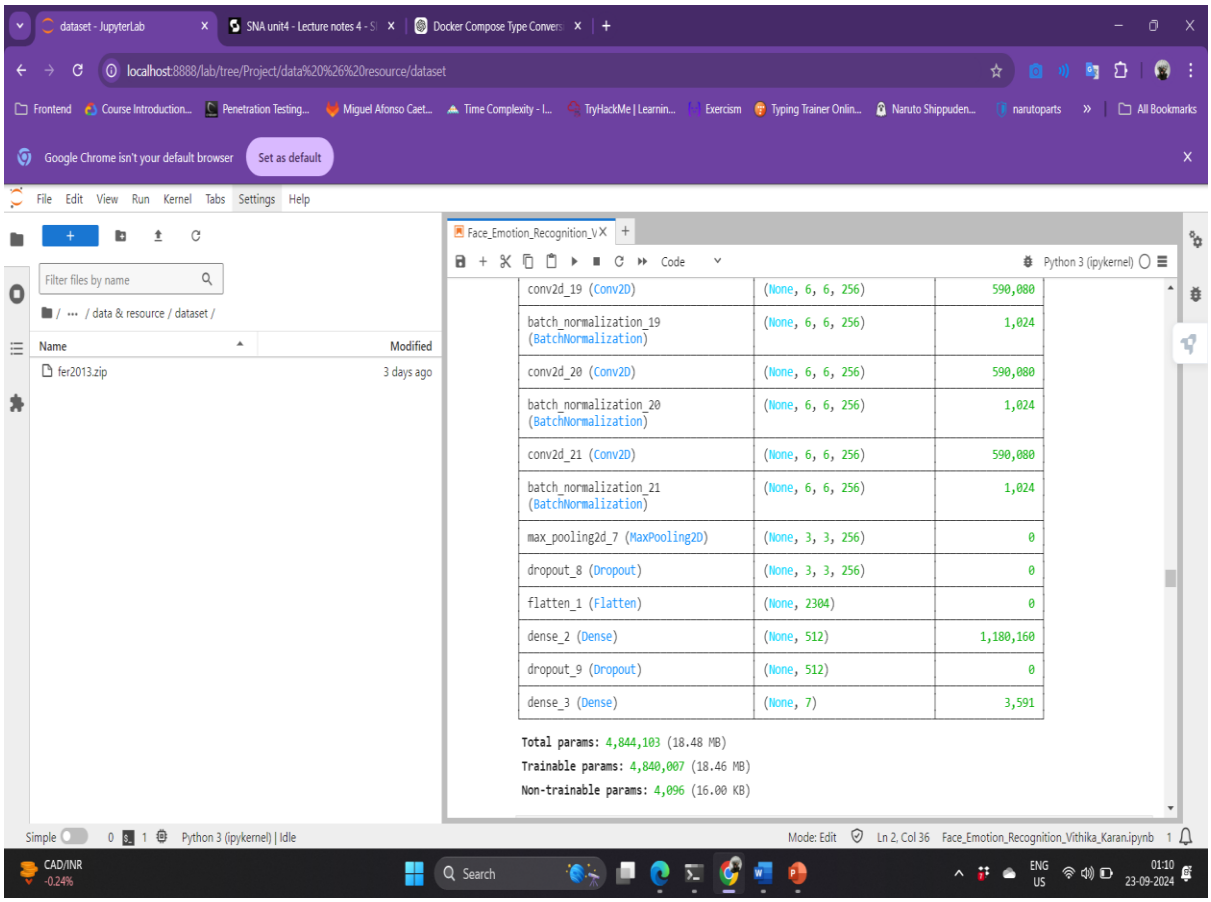

dataset - JupyterLab
SNA unit4 - Lecture notes 4 - S
Docker Compose Type Convers
localhost:8888/lab/tree/Project/data%20%26%20resource/dataset
Google Chrome isn't your default browser
Set as default
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ data & resource / dataset /
Name
Modified
fer2013.zip
3 days ago
Face_Emotion_Recognition_V
Code
Python 3 (ipykernel)
Found 28709 images belonging to 7 classes.
Found 3589 images belonging to 7 classes.
####Model Building
[26]:
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D, BatchNormalization, MaxPooling2D, Dropout, Flatten, Dense
from tensorflow.keras.optimizers import Adam
model = Sequential()
num_classes = 7
# 1st Block
model.add(Convolution2D(64, kernel_size=3, activation='relu', padding='same', input_shape=(48, 48, 1)))
model.add(BatchNormalization())
model.add(Convolution2D(64, kernel_size=3, activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
# 2nd Block
model.add(Convolution2D(128, kernel_size=3, activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(Convolution2D(128, kernel_size=3, activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(Convolution2D(128, kernel_size=3, activation='relu', padding='same'))
model.add(BatchNormalization())
Simple
Python 3 (ipykernel) | Idle
Mode: Edit
Ln 2, Col 36
Face_Emotion_Recognition_Vithika_Karan.ipynb
USD/INR
-0.16%
Search
ENG
US
01:08
23-09-2024

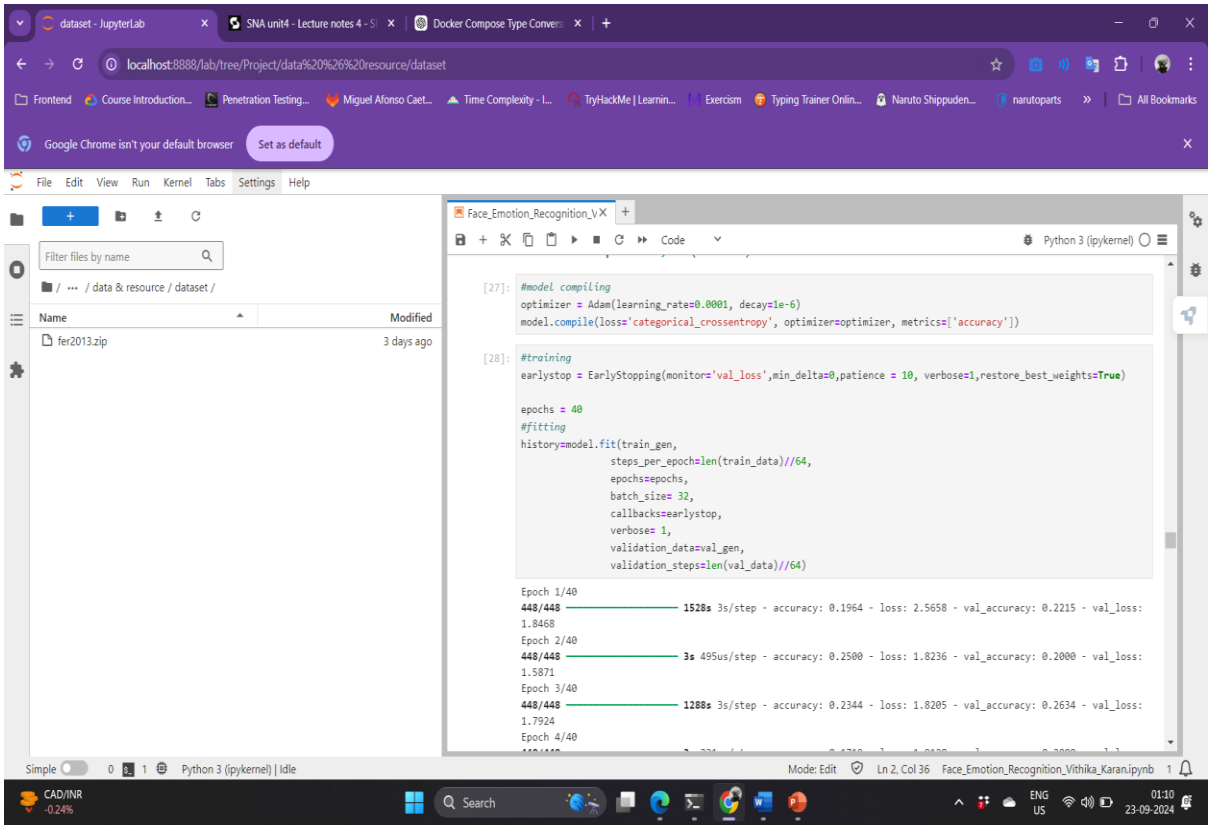

dataset - JupyterLab
SNA unit4 - Lecture notes 4 - S
Docker Compose Type Convers
localhost:8888/lab/tree/Project/data%20%26%20resource/dataset
Google Chrome isn't your default browser
Set as default
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ data & resource / dataset /
Name
Modified
fer2013.zip
3 days ago
Face_Emotion_Recognition_V
Code
Python 3 (ipykernel)
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.3))
# 3rd Block
model.add(Convolution2D(256, kernel_size=3, activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(Convolution2D(256, kernel_size=3, activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(Convolution2D(256, kernel_size=3, activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.3))
# 4th Block
model.add(Convolution2D(256, kernel_size=3, activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(Convolution2D(256, kernel_size=3, activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(Convolution2D(256, kernel_size=3, activation='relu', padding='same'))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.3))
# Flatten and Fully Connected Layers
model.add(Flatten())
# Optional Dense Layers
model.add(Dense(512, activation='relu'))
Simple
Python 3 (ipykernel) | Idle
Mode: Edit
Ln 2, Col 36
Face_Emotion_Recognition_Vithika_Karan.ipynb
28°C
Haze
Search
ENG
US
01:09
23-09-2024

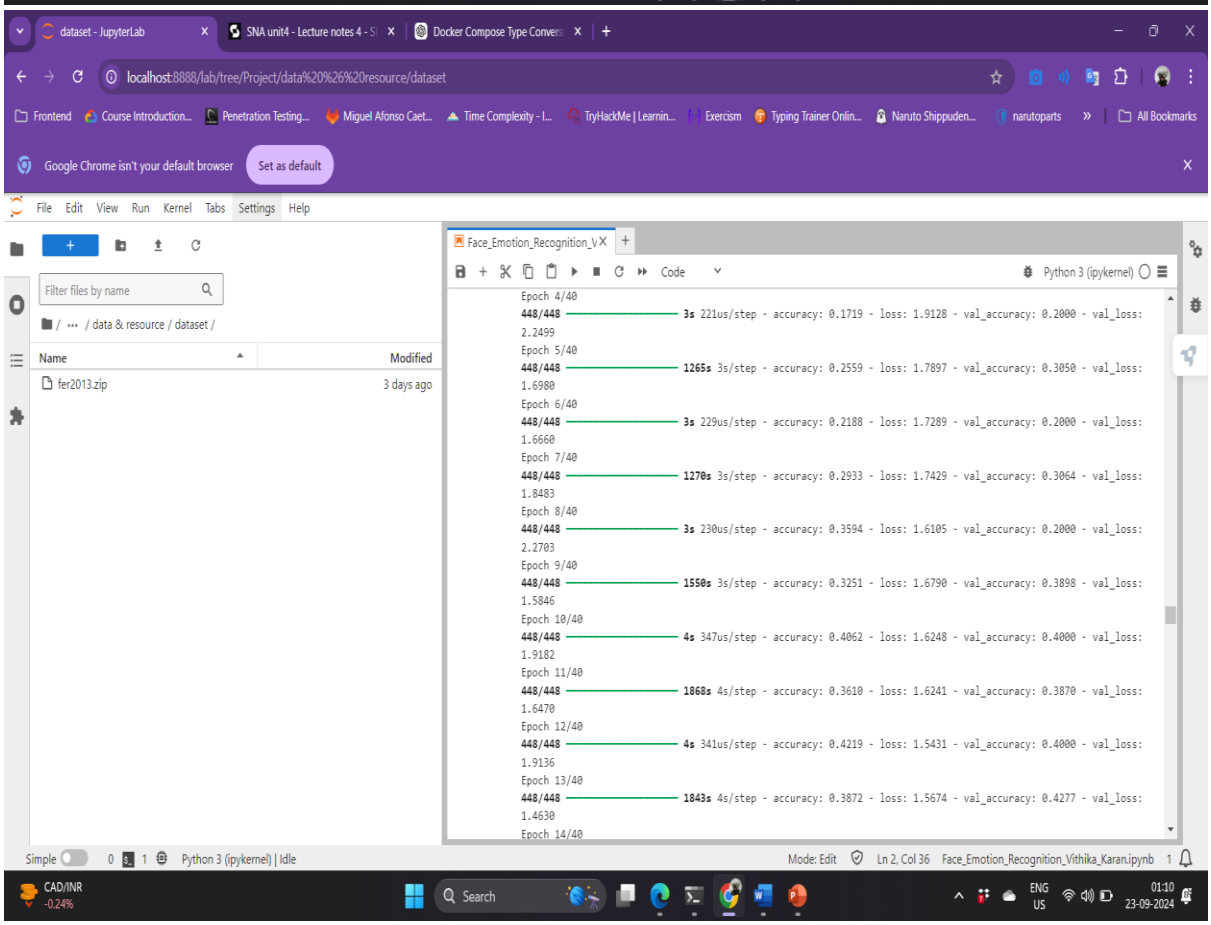

dataset - JupyterLab
SNA unit4 - Lecture notes 4 - S
Docker Compose Type Convers
localhost:8888/lab/tree/Project/data%20%26%20resource/dataset
Google Chrome isn't your default browser
Set as default
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ ... / data & resource / dataset /
Name
Modified
fer2013.zip
3 days ago
Face_Emotion_Recognition_V
Code
Python 3 (ipykernel)
model.add(Dropout(0.3))
# Flatten and Fully Connected Layers
model.add(Flatten())
# Optional Dense Layers
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))
# Summary of the model
model.summary()
# Compile the model
model.compile(optimizer=Adam(), loss='categorical_crossentropy', metrics=['accuracy'])
Model: "sequential_1"
Layer (type) Output Shape Param #
conv2d_11 (Conv2D) (None, 48, 48, 64) 640
batch_normalization_11 (BatchNormalization) (None, 48, 48, 64) 256
conv2d_12 (Conv2D) (None, 48, 48, 64) 36,928
batch_normalization_12 (BatchNormalization) (None, 48, 48, 64) 256
Simple
Python 3 (ipykernel) | Idle
Mode: Edit
Ln 2, Col 36
Face_Emotion_Recognition_Vithika_Karan.ipynb
28°C
Haze
Search
ENG
US
01:09
23-09-2024

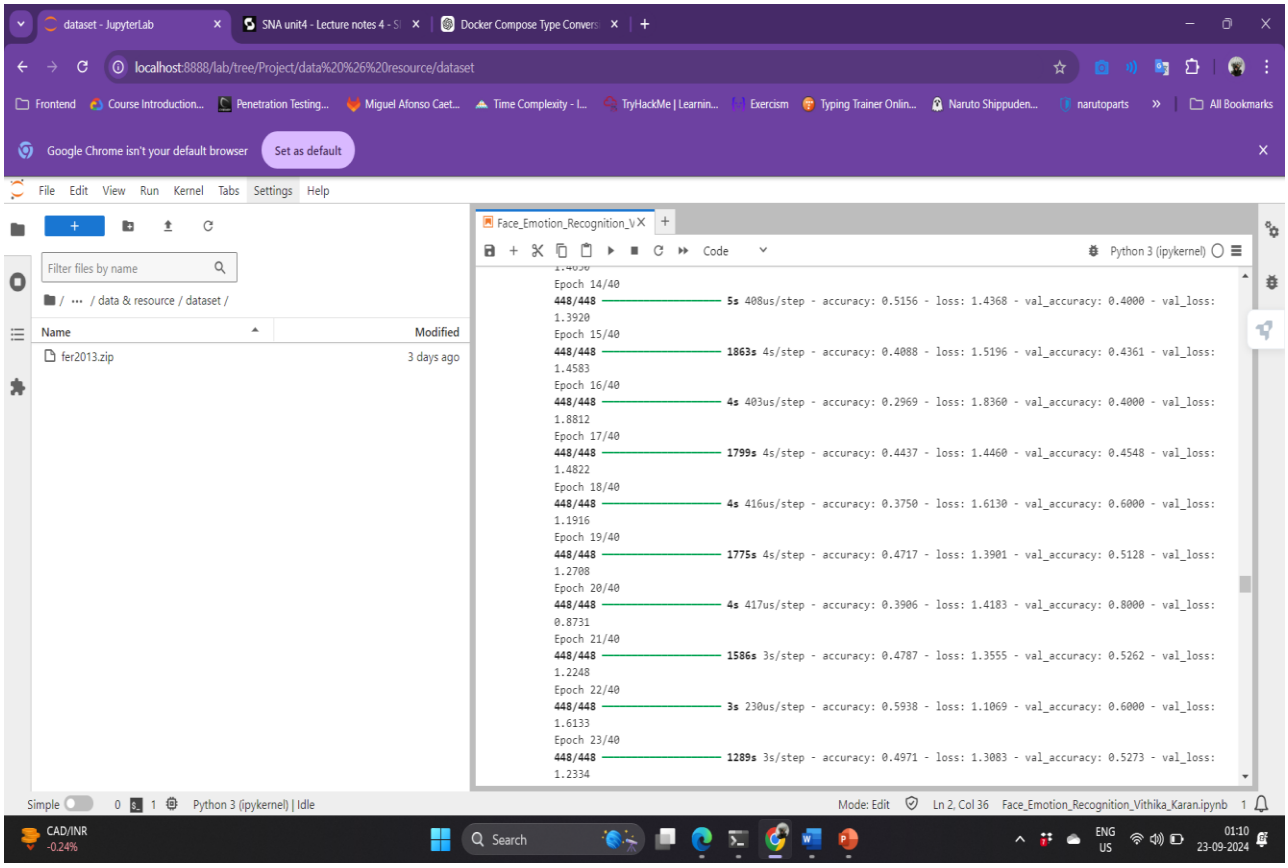

dataset - JupyterLab
SNA unit4 - Lecture notes 4 - S
Docker Compose Type Convers
localhost:8888/lab/tree/Project/data%20%26%20resource/dataset
Google Chrome isn't your default browser
Set as default
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ ... / data & resource / dataset /
Name
Modified
fer2013.zip
3 days ago
Face_Emotion_Recognition_V
Code
Python 3 (ipykernel)
Model: "sequential_1"
Layer (type) Output Shape Param #
conv2d_11 (Conv2D) (None, 48, 48, 64) 640
batch_normalization_11 (BatchNormalization) (None, 48, 48, 64) 256
conv2d_12 (Conv2D) (None, 48, 48, 64) 36,928
batch_normalization_12 (BatchNormalization) (None, 48, 48, 64) 256
max_pooling2d_4 (MaxPooling2D) (None, 24, 24, 64) 0
dropout_5 (Dropout) (None, 24, 24, 64) 0
conv2d_13 (Conv2D) (None, 24, 24, 128) 73,856
batch_normalization_13 (BatchNormalization) (None, 24, 24, 128) 512
conv2d_14 (Conv2D) (None, 24, 24, 128) 147,584
batch_normalization_14 (BatchNormalization) (None, 24, 24, 128) 512
conv2d_15 (Conv2D) (None, 24, 24, 128) 147,584
batch_normalization_15 (BatchNormalization) (None, 24, 24, 128) 512
Simple
Python 3 (ipykernel) | Idle
Mode: Edit
Ln 2, Col 36
Face_Emotion_Recognition_Vithika_Karan.ipynb
28°C
Haze
Search
ENG
US
01:09
23-09-2024

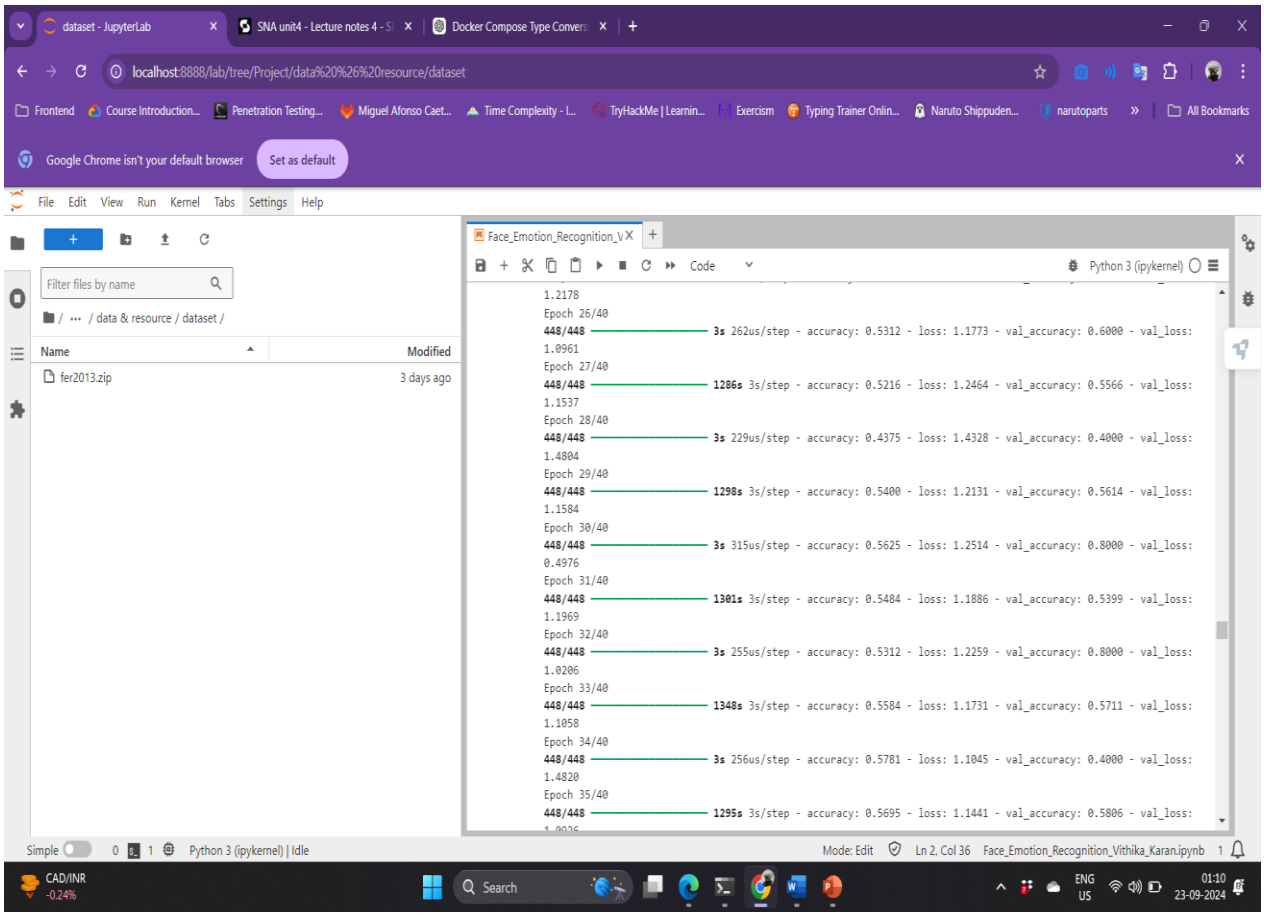

dataset - JupyterLab
SNA unit4 - Lecture notes 4 - Si
Docker Compose Type Convers
localhost:8888/lab/tree/Project/data%20%26%20resource/dataset
Google Chrome isn't your default browser
Set as default
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ ... / data & resource / dataset /
Name
Modified
fer2013.zip
3 days ago
Face_Emotion_Recognition_V
Code
Python 3 (ipykernel)
batch_normalization_14 (BatchNormalization) (None, 24, 24, 128) 512
conv2d_15 (Conv2D) (None, 24, 24, 128) 147,584
batch_normalization_15 (BatchNormalization) (None, 24, 24, 128) 512
max_pooling2d_5 (MaxPooling2D) (None, 12, 12, 128) 0
dropout_6 (Dropout) (None, 12, 12, 128) 0
conv2d_16 (Conv2D) (None, 12, 12, 256) 295,168
batch_normalization_16 (BatchNormalization) (None, 12, 12, 256) 1,024
conv2d_17 (Conv2D) (None, 12, 12, 256) 590,080
batch_normalization_17 (BatchNormalization) (None, 12, 12, 256) 1,024
conv2d_18 (Conv2D) (None, 12, 12, 256) 590,080
batch_normalization_18 (BatchNormalization) (None, 12, 12, 256) 1,024
max_pooling2d_6 (MaxPooling2D) (None, 6, 6, 256) 0
dropout_7 (Dropout) (None, 6, 6, 256) 0
conv2d_19 (Conv2D) (None, 6, 6, 256) 590,080
Simple
Python 3 (ipykernel) | Idle
Mode: Edit
Ln 2, Col 36
Face_Emotion_Recognition_Vithika_Karan.ipynb
28°C Haze
Search
ENG US
01:09 23-09-2024


dataset - JupyterLab
SNA unit4 - Lecture notes 4 - Si
Docker Compose Type Convers
localhost:8888/lab/tree/Project/data%20%26%20resource/dataset
Google Chrome isn't your default browser
Set as default
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ ... / data & resource / dataset /
Name
Modified
fer2013.zip
3 days ago
Face_Emotion_Recognition_V
Code
Python 3 (ipykernel)
conv2d_19 (Conv2D) (None, 6, 6, 256) 590,080
batch_normalization_19 (BatchNormalization) (None, 6, 6, 256) 1,024
conv2d_20 (Conv2D) (None, 6, 6, 256) 590,080
batch_normalization_20 (BatchNormalization) (None, 6, 6, 256) 1,024
conv2d_21 (Conv2D) (None, 6, 6, 256) 590,080
batch_normalization_21 (BatchNormalization) (None, 6, 6, 256) 1,024
max_pooling2d_7 (MaxPooling2D) (None, 3, 3, 256) 0
dropout_8 (Dropout) (None, 3, 3, 256) 0
flatten_1 (Flatten) (None, 2304) 0
dense_2 (Dense) (None, 512) 1,180,160
dropout_9 (Dropout) (None, 512) 0
dense_3 (Dense) (None, 7) 3,591
Total params: 4,844,103 (18.48 MB)
Trainable params: 4,840,007 (18.46 MB)
Non-trainable params: 4,096 (16.00 KB)
Simple
Python 3 (ipykernel) | Idle
Mode: Edit
Ln 2, Col 36
Face_Emotion_Recognition_Vithika_Karan.ipynb
CAD/INR -0.24%
Search
ENG US
01:10 23-09-2024


dataset - JupyterLab
SNA unit4 - Lecture notes 4 - Si
Docker Compose Type Convers
localhost:8888/lab/tree/Project/data%20%26%20resource/dataset
Google Chrome isn't your default browser
Set as default
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ data & resource / dataset /
Name
Modified
fer2013.zip
3 days ago
Face_Emotion_Recognition_V
Code
Python 3 (ipykernel)
[27]:
#model compiling
optimizer = Adam(learning_rate=0.0001, decay=1e-6)
model.compile(loss='categorical_crossentropy', optimizer=optimizer, metrics=['accuracy'])
[28]:
#training
earlystop = EarlyStopping(monitor='val_loss',min_delta=0,patience = 10, verbose=1,restore_best_weights=True)
epochs = 40
#fitting
history=model.fit(train_gen,
steps_per_epoch=len(train_data)//64,
epochs=epochs,
batch_size= 32,
callbacks=earlystop,
verbose= 1,
validation_data=val_gen,
validation_steps=len(val_data)//64)
Epoch 1/40
448/448 1528s 3s/step - accuracy: 0.1964 - loss: 2.5658 - val_accuracy: 0.2215 - val_loss: 1.8468
Epoch 2/40
448/448 3s 495us/step - accuracy: 0.2500 - loss: 1.8236 - val_accuracy: 0.2000 - val_loss: 1.5871
Epoch 3/40
448/448 1288s 3s/step - accuracy: 0.2344 - loss: 1.8205 - val_accuracy: 0.2634 - val_loss: 1.7924
Epoch 4/40
Simple 0 1 Python 3 (ipykernel) | Idle
Mode: Edit Ln 2, Col 36 Face_Emotion_Recognition_Vithika_Karan.ipynb 1
CAD/INR -0.24%
Search
ENG US
01:10 23-09-2024


dataset - JupyterLab
SNA unit4 - Lecture notes 4 - Si
Docker Compose Type Convers
localhost:8888/lab/tree/Project/data%20%26%20resource/dataset
Google Chrome isn't your default browser
Set as default
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ data & resource / dataset /
Name
Modified
fer2013.zip
3 days ago
Face_Emotion_Recognition_V
Code
Python 3 (ipykernel)
Epoch 4/40
448/448 3s 221us/step - accuracy: 0.1719 - loss: 1.9128 - val_accuracy: 0.2000 - val_loss: 2.2499
Epoch 5/40
448/448 1265s 3s/step - accuracy: 0.2559 - loss: 1.7897 - val_accuracy: 0.3050 - val_loss: 1.6980
Epoch 6/40
448/448 3s 229us/step - accuracy: 0.2188 - loss: 1.7289 - val_accuracy: 0.2000 - val_loss: 1.6660
Epoch 7/40
448/448 1270s 3s/step - accuracy: 0.2933 - loss: 1.7429 - val_accuracy: 0.3064 - val_loss: 1.8483
Epoch 8/40
448/448 3s 230us/step - accuracy: 0.3594 - loss: 1.6105 - val_accuracy: 0.2000 - val_loss: 2.2703
Epoch 9/40
448/448 1550s 3s/step - accuracy: 0.3251 - loss: 1.6790 - val_accuracy: 0.3898 - val_loss: 1.5846
Epoch 10/40
448/448 4s 347us/step - accuracy: 0.4062 - loss: 1.6248 - val_accuracy: 0.4000 - val_loss: 1.9182
Epoch 11/40
448/448 1868s 4s/step - accuracy: 0.3610 - loss: 1.6241 - val_accuracy: 0.3870 - val_loss: 1.6470
Epoch 12/40
448/448 4s 341us/step - accuracy: 0.4219 - loss: 1.5431 - val_accuracy: 0.4000 - val_loss: 1.9136
Epoch 13/40
448/448 1843s 4s/step - accuracy: 0.3872 - loss: 1.5674 - val_accuracy: 0.4277 - val_loss: 1.4630
Epoch 14/40
Simple 0 1 Python 3 (ipykernel) | Idle
Mode: Edit Ln 2, Col 36 Face_Emotion_Recognition_Vithika_Karan.ipynb 1
CAD/INR -0.24%
Search
ENG US
01:10 23-09-2024


dataset - JupyterLab
SNA unit4 - Lecture notes 4 - S
Docker Compose Type Convers
localhost:8888/lab/tree/Project/data%20%26%20resource/dataset
Google Chrome isn't your default browser
Set as default
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ data & resource / dataset /
Name
Modified
fer2013.zip
3 days ago
Face_Emotion_Recognition_V
Code
Python 3 (ipykernel)
Epoch 14/40
448/448 5s 408us/step - accuracy: 0.5156 - loss: 1.4368 - val_accuracy: 0.4000 - val_loss: 1.3920
Epoch 15/40
448/448 1863s 4s/step - accuracy: 0.4088 - loss: 1.5196 - val_accuracy: 0.4361 - val_loss: 1.4583
Epoch 16/40
448/448 4s 403us/step - accuracy: 0.2969 - loss: 1.8360 - val_accuracy: 0.4000 - val_loss: 1.8812
Epoch 17/40
448/448 1799s 4s/step - accuracy: 0.4437 - loss: 1.4460 - val_accuracy: 0.4548 - val_loss: 1.4822
Epoch 18/40
448/448 4s 416us/step - accuracy: 0.3750 - loss: 1.6130 - val_accuracy: 0.6000 - val_loss: 1.1916
Epoch 19/40
448/448 1775s 4s/step - accuracy: 0.4717 - loss: 1.3901 - val_accuracy: 0.5128 - val_loss: 1.2708
Epoch 20/40
448/448 4s 417us/step - accuracy: 0.3906 - loss: 1.4183 - val_accuracy: 0.8000 - val_loss: 0.8731
Epoch 21/40
448/448 1586s 3s/step - accuracy: 0.4787 - loss: 1.3555 - val_accuracy: 0.5262 - val_loss: 1.2248
Epoch 22/40
448/448 3s 230us/step - accuracy: 0.5938 - loss: 1.1069 - val_accuracy: 0.6000 - val_loss: 1.6133
Epoch 23/40
448/448 1289s 3s/step - accuracy: 0.4971 - loss: 1.3083 - val_accuracy: 0.5273 - val_loss: 1.2334
Simple
Python 3 (ipykernel) | Idle
Mode: Edit
Ln 2, Col 36
Face_Emotion_Recognition_Vithika_Karan.ipynb
CAD/INR -0.24%
Search
ENG US
01:10 23-09-2024


dataset - JupyterLab
SNA unit4 - Lecture notes 4 - S
Docker Compose Type Convers
localhost:8888/lab/tree/Project/data%20%26%20resource/dataset
Google Chrome isn't your default browser
Set as default
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ data & resource / dataset /
Name
Modified
fer2013.zip
3 days ago
Face_Emotion_Recognition_V
Code
Python 3 (ipykernel)
1.2178
Epoch 26/40
448/448 3s 262us/step - accuracy: 0.5312 - loss: 1.1773 - val_accuracy: 0.6000 - val_loss: 1.0961
Epoch 27/40
448/448 1286s 3s/step - accuracy: 0.5216 - loss: 1.2464 - val_accuracy: 0.5566 - val_loss: 1.1537
Epoch 28/40
448/448 3s 229us/step - accuracy: 0.4375 - loss: 1.4328 - val_accuracy: 0.4000 - val_loss: 1.4804
Epoch 29/40
448/448 1298s 3s/step - accuracy: 0.5400 - loss: 1.2131 - val_accuracy: 0.5614 - val_loss: 1.1584
Epoch 30/40
448/448 3s 315us/step - accuracy: 0.5625 - loss: 1.2514 - val_accuracy: 0.8000 - val_loss: 0.4976
Epoch 31/40
448/448 1301s 3s/step - accuracy: 0.5484 - loss: 1.1886 - val_accuracy: 0.5399 - val_loss: 1.1969
Epoch 32/40
448/448 3s 255us/step - accuracy: 0.5312 - loss: 1.2259 - val_accuracy: 0.8000 - val_loss: 1.0206
Epoch 33/40
448/448 1348s 3s/step - accuracy: 0.5584 - loss: 1.1731 - val_accuracy: 0.5711 - val_loss: 1.1058
Epoch 34/40
448/448 3s 256us/step - accuracy: 0.5781 - loss: 1.1045 - val_accuracy: 0.4000 - val_loss: 1.4820
Epoch 35/40
448/448 1295s 3s/step - accuracy: 0.5695 - loss: 1.1441 - val_accuracy: 0.5806 - val_loss:
Simple
Python 3 (ipykernel) | Idle
Mode: Edit
Ln 2, Col 36
Face_Emotion_Recognition_Vithika_Karan.ipynb
CAD/INR -0.24%
Search
ENG US
01:10 23-09-2024

```
448/448 ━━━━━━━━━━━━━━━━━━━━ 1295s 3s/step - accuracy: 0.5695 - loss: 1.1441 - val_accuracy: 0.5806 - val_loss:
1.0926
Epoch 36/40
448/448 ━━━━━━━━━━━━━━━━━━━━ 3s 260us/step - accuracy: 0.5781 - loss: 1.0125 - val_accuracy: 0.8000 - val_loss:
0.6906
Epoch 37/40
448/448 ━━━━━━━━━━━━━━━━━━━━ 1651s 4s/step - accuracy: 0.5740 - loss: 1.1319 - val_accuracy: 0.5781 - val_loss:
1.1211
Epoch 38/40
448/448 ━━━━━━━━━━━━━━━━━━━━ 4s 380us/step - accuracy: 0.6406 - loss: 1.0494 - val_accuracy: 0.2000 - val_loss:
1.1991
Epoch 39/40
448/448 ━━━━━━━━━━━━━━━━━━━━ 1736s 4s/step - accuracy: 0.5842 - loss: 1.1157 - val_accuracy: 0.5806 - val_loss:
1.1065
Epoch 40/40
448/448 ━━━━━━━━━━━━━━━━━━━━ 4s 333us/step - accuracy: 0.5625 - loss: 1.0326 - val_accuracy: 0.2000 - val_loss:
1.7910
Epoch 40: early stopping
Restoring model weights from the end of the best epoch: 30.
```

Restored the best model weights, which was for the epoch 45 with the following result:

- loss: 0.7075 - accuracy: 0.7381 - val_loss: 0.9825 - val_accuracy: 0.6643

####Model Evaluation

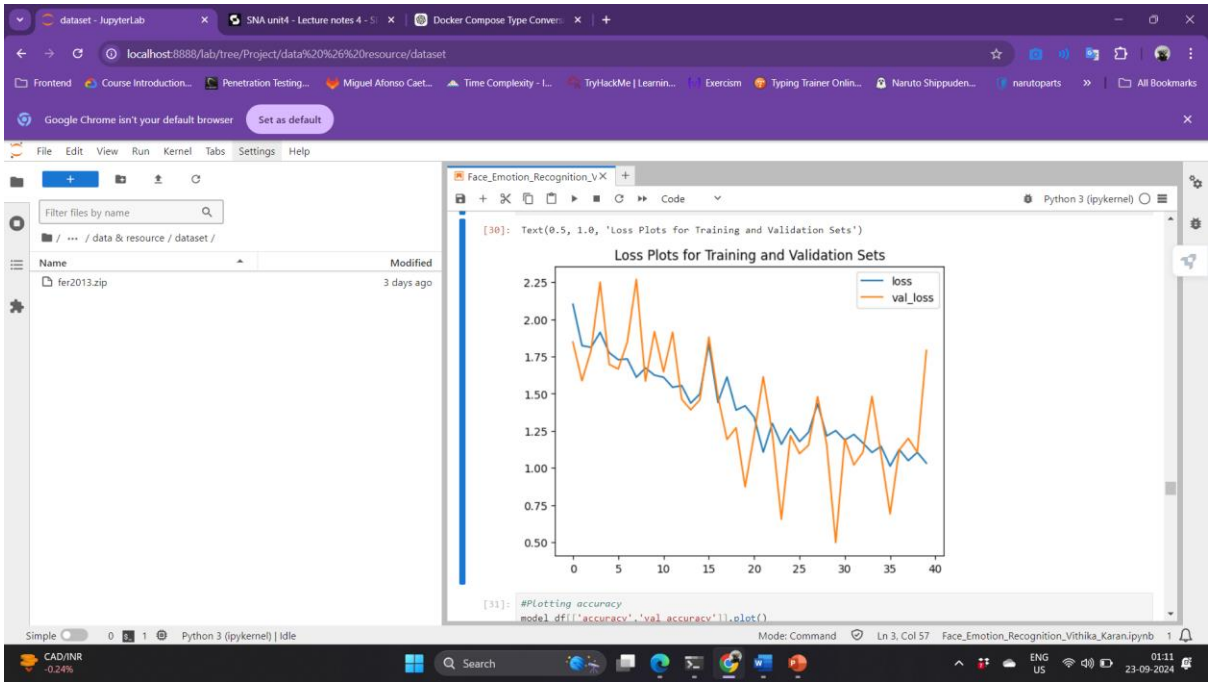
[29]:
```
#saving the history of the model in dataframe
model_df = pd.DataFrame(model.history.history)
```

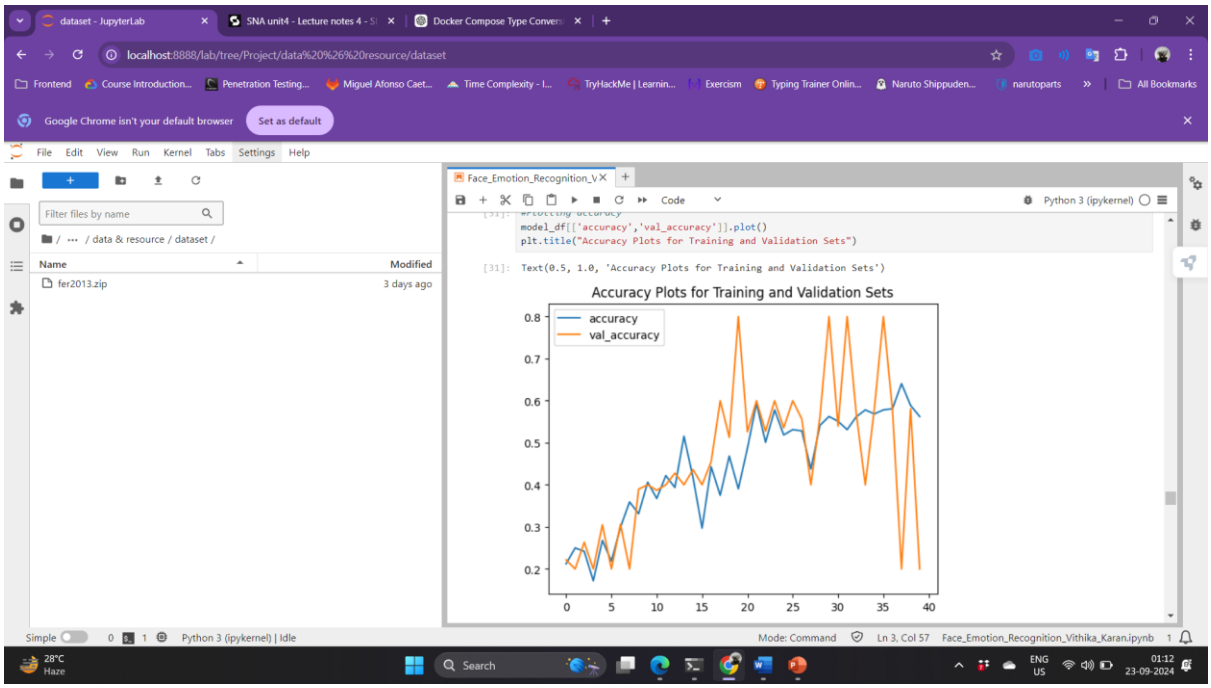
####Model Evaluation

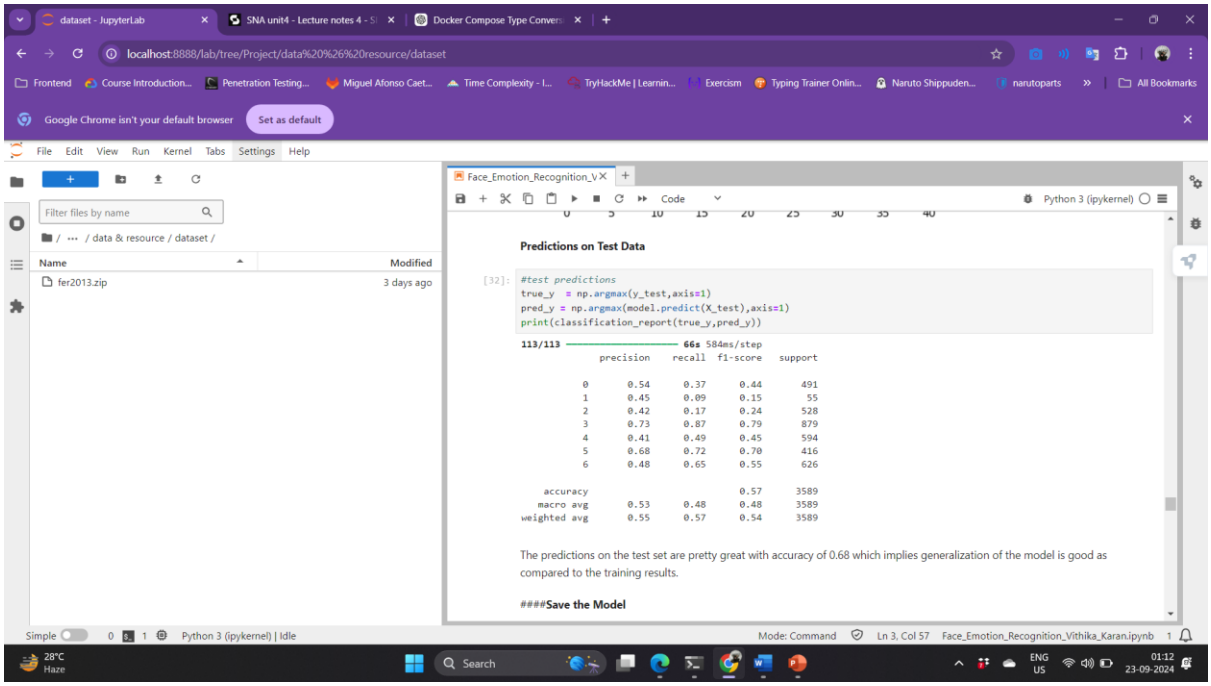
[29]:
```
#saving the history of the model in dataframe
model_df = pd.DataFrame(model.history.history)
```

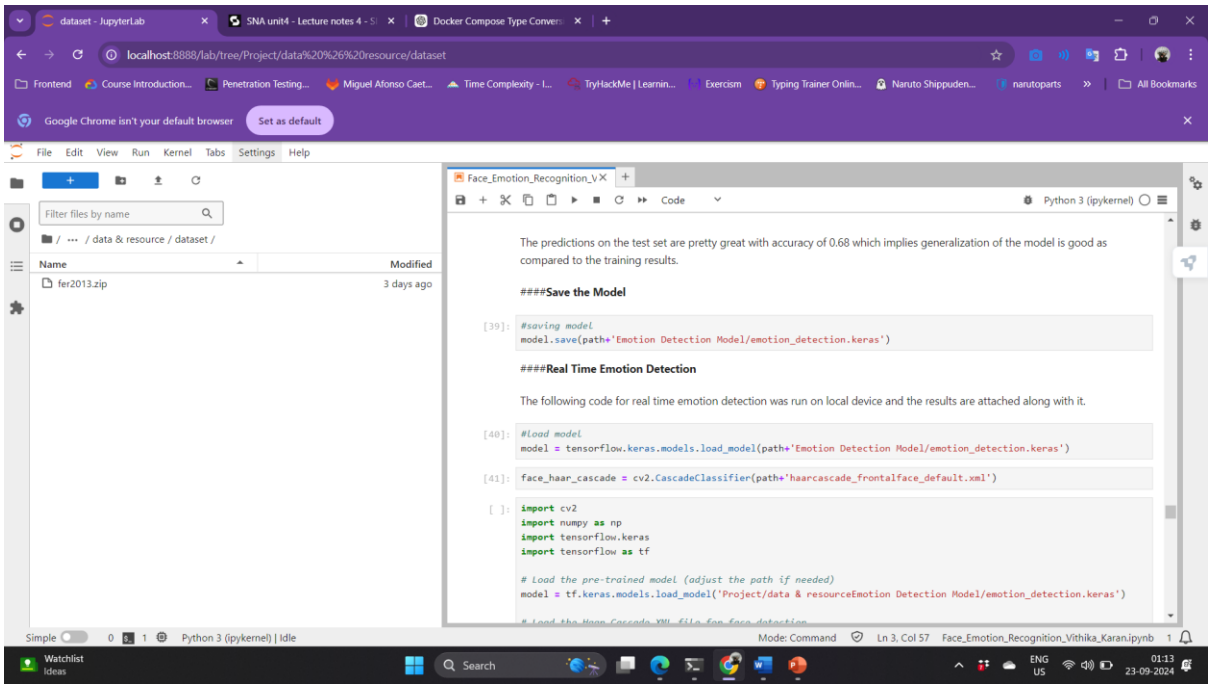
[30]:
```
#Plotting Loss
model_df[['loss','val_loss']].plot()
plt.title("Loss Plots for Training and Validation Sets")
```

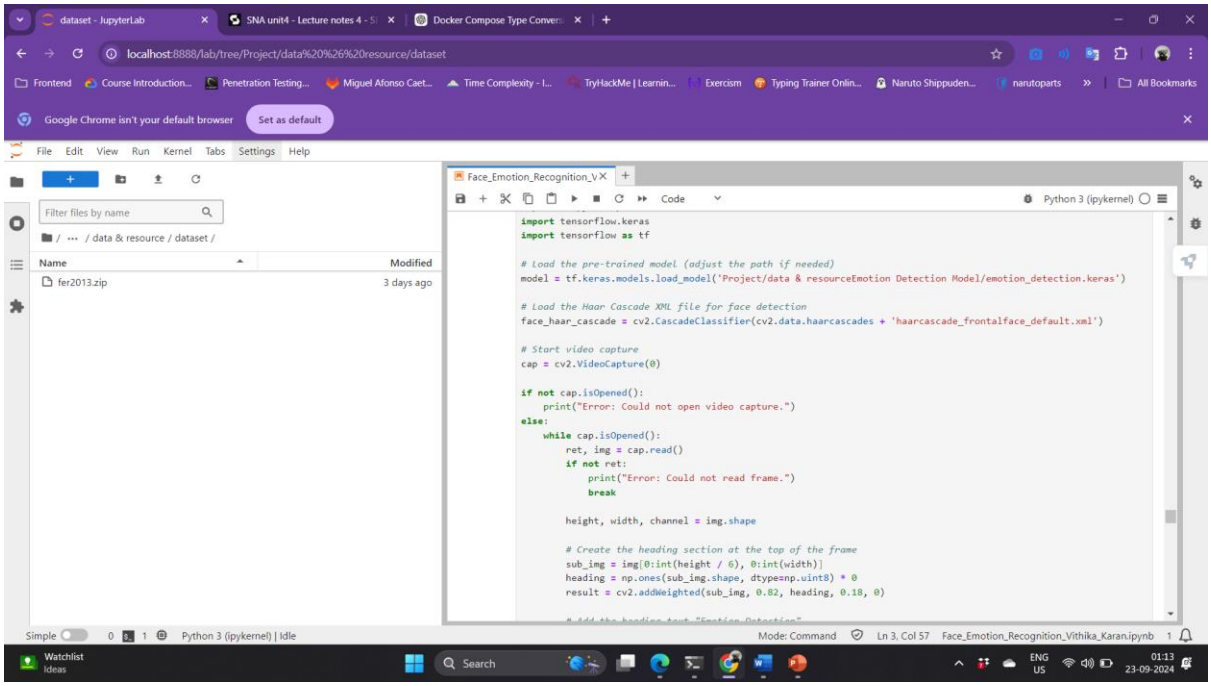
[30]:
```
Text(0.5, 1.0, 'Loss Plots for Training and Validation Sets')
```

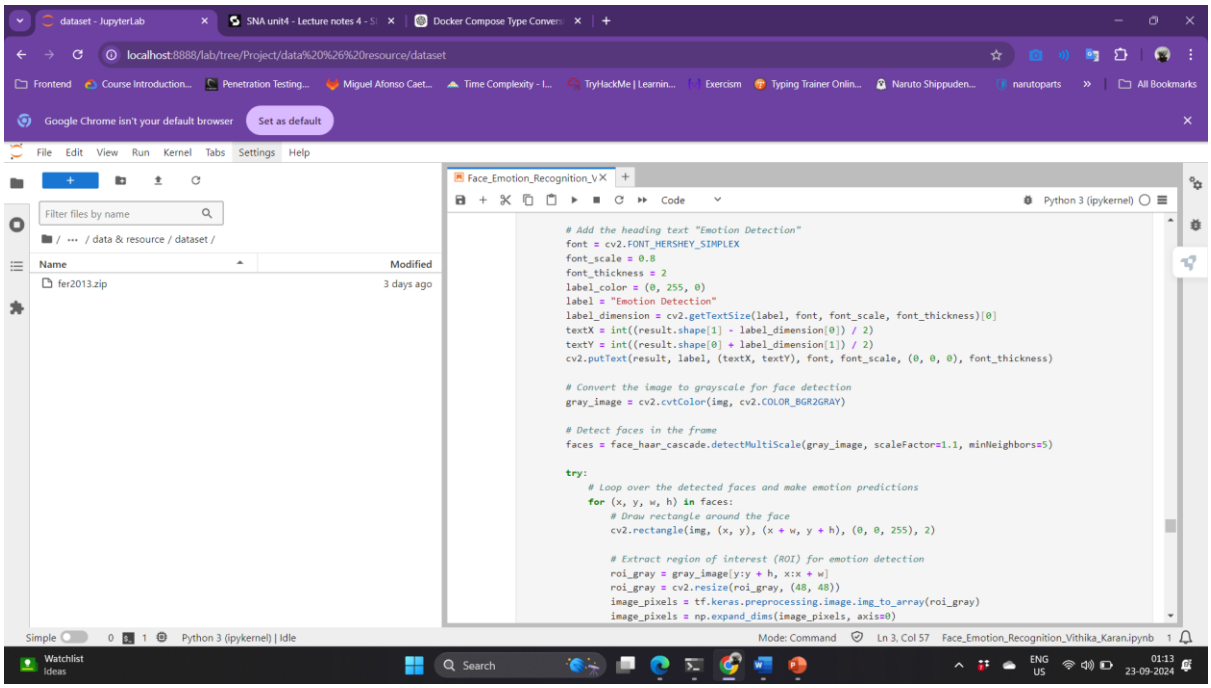

dataset - JupyterLab
SNA unit4 - Lecture notes 4 - Si
Docker Compose Type Convers
localhost:8888/lab/tree/Project/data%20%26%20resource/dataset
Google Chrome isn't your default browser
Set as default
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ data & resource / dataset /
Name
Modified
fer2013.zip
3 days ago
Face_Emotion_Recognition_V
Code
Python 3 (ipykernel)
[30]: Text(0.5, 1.0, 'Loss Plots for Training and Validation Sets')
Loss Plots for Training and Validation Sets
loss
val_loss
[31]: #Plotting accuracy
model_df[['accuracy','val_accuracy']].plot()
Simple
Python 3 (ipykernel) | Idle
Mode: Command
Ln 3, Col 57
Face_Emotion_Recognition_Vithika_Karan.ipynb
CAD/INR
-0.24%
Search
ENG
US
01:11
23-09-2024

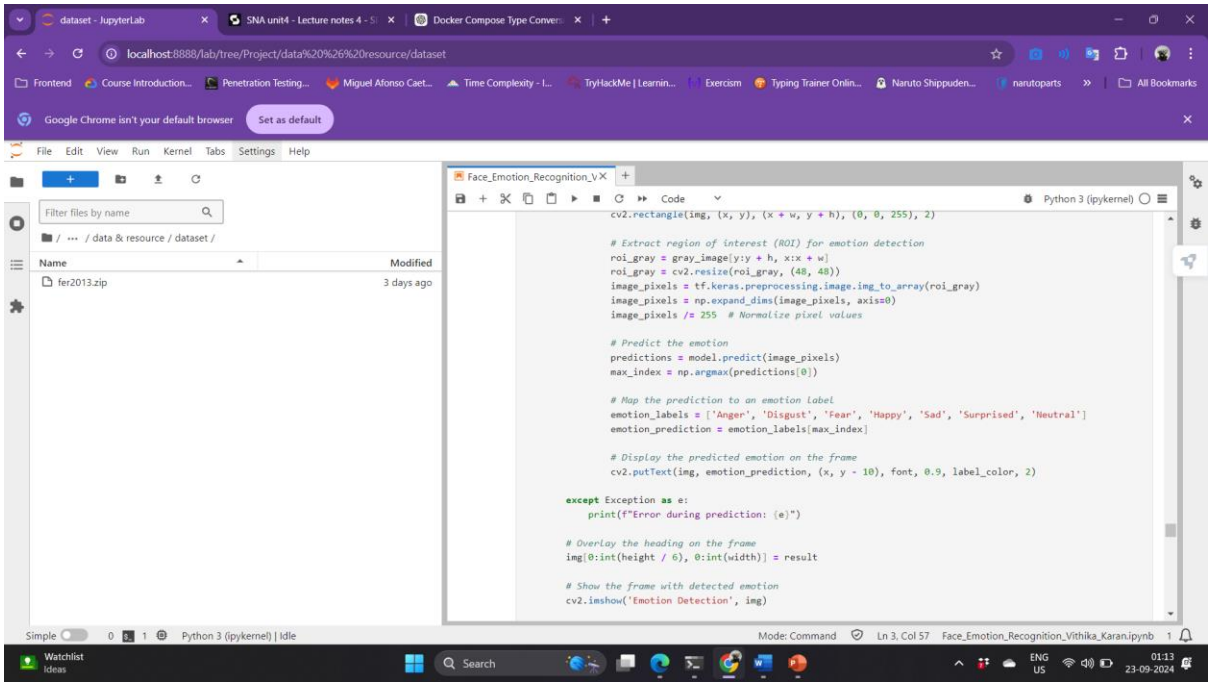

dataset - JupyterLab
SNA unit4 - Lecture notes 4 - Si
Docker Compose Type Convers
localhost:8888/lab/tree/Project/data%20%26%20resource/dataset
Google Chrome isn't your default browser
Set as default
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ data & resource / dataset /
Name
Modified
fer2013.zip
3 days ago
Face_Emotion_Recognition_V
Code
Python 3 (ipykernel)
model_df[['accuracy','val_accuracy']].plot()
plt.title("Accuracy Plots for Training and Validation Sets")
[31]: Text(0.5, 1.0, 'Accuracy Plots for Training and Validation Sets')
Accuracy Plots for Training and Validation Sets
accuracy
val_accuracy
Simple
Python 3 (ipykernel) | Idle
Mode: Command
Ln 3, Col 57
Face_Emotion_Recognition_Vithika_Karan.ipynb
28°C
Haze
Search
ENG
US
01:12
23-09-2024

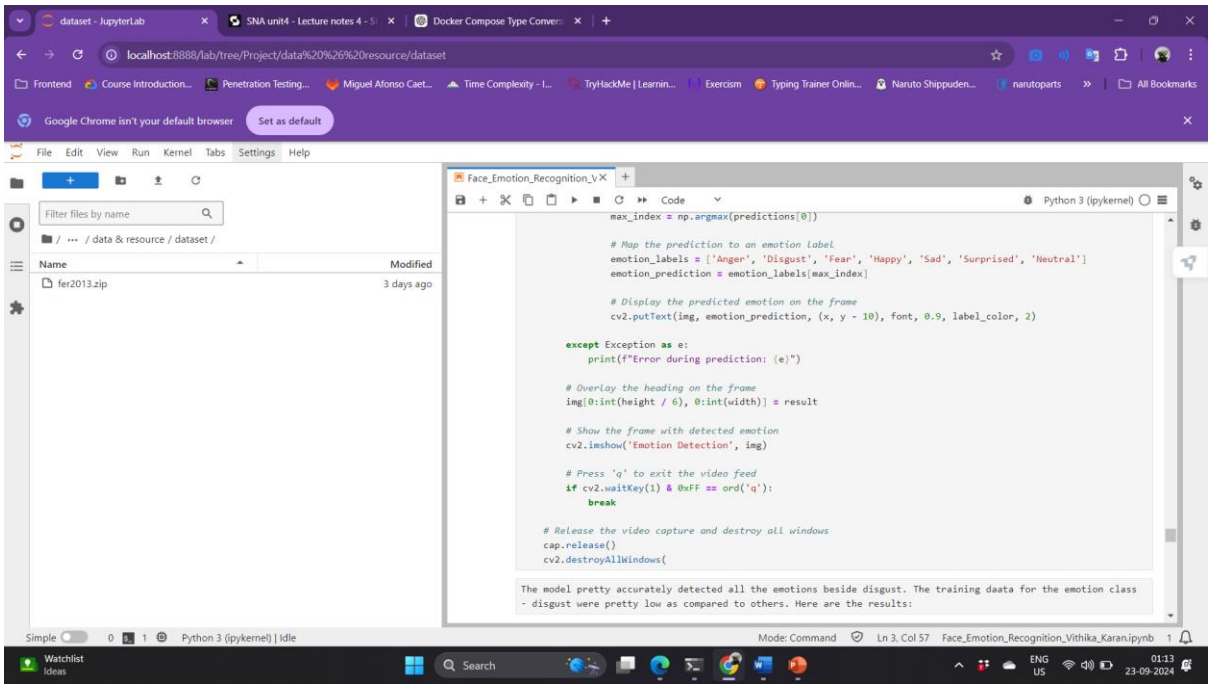

dataset - JupyterLab
SNA unit4 - Lecture notes 4 - S
Docker Compose Type Convers
localhost:8888/lab/tree/Project/data%20%26%20resource/dataset
Google Chrome isn't your default browser
Set as default
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ data & resource / dataset /
Name
Modified
fer2013.zip
3 days ago
Face_Emotion_Recognition_V
Code
Python 3 (ipykernel)
Predictions on Test Data
[32]:
#test predictions
true_y = np.argmax(y_test,axis=1)
pred_y = np.argmax(model.predict(X_test),axis=1)
print(classification_report(true_y,pred_y))
113/113 66s 584ms/step
precision recall f1-score support
0 0.54 0.37 0.44 491
1 0.45 0.09 0.15 55
2 0.42 0.17 0.24 528
3 0.73 0.87 0.79 879
4 0.41 0.49 0.45 594
5 0.68 0.72 0.70 416
6 0.48 0.65 0.55 626
accuracy 0.57 3589
macro avg 0.53 0.48 0.48 3589
weighted avg 0.55 0.57 0.54 3589
The predictions on the test set are pretty great with accuracy of 0.68 which implies generalization of the model is good as compared to the training results.
####Save the Model
Simple 0 1 Python 3 (ipykernel) | Idle
Mode: Command Ln 3, Col 57 Face_Emotion_Recognition_Vithika_Karan.ipynb 1
28°C Haze
Search
ENG US
01:12 23-09-2024


dataset - JupyterLab
SNA unit4 - Lecture notes 4 - S
Docker Compose Type Convers
localhost:8888/lab/tree/Project/data%20%26%20resource/dataset
Google Chrome isn't your default browser
Set as default
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ data & resource / dataset /
Name
Modified
fer2013.zip
3 days ago
Face_Emotion_Recognition_V
Code
Python 3 (ipykernel)
The predictions on the test set are pretty great with accuracy of 0.68 which implies generalization of the model is good as compared to the training results.
####Save the Model
[39]:
#saving model
model.save(path+'Emotion Detection Model/emotion_detection.keras')
####Real Time Emotion Detection
The following code for real time emotion detection was run on local device and the results are attached along with it.
[40]:
#Load model
model = tensorflow.keras.models.load_model(path+'Emotion Detection Model/emotion_detection.keras')
[41]:
face_haar_cascade = cv2.CascadeClassifier(path+'haarcascade_frontalface_default.xml')
[ ]:
import cv2
import numpy as np
import tensorflow.keras
import tensorflow as tf
# Load the pre-trained model (adjust the path if needed)
model = tf.keras.models.load_model('Project/data & resourceEmotion Detection Model/emotion_detection.keras')
Simple 0 1 Python 3 (ipykernel) | Idle
Mode: Command Ln 3, Col 57 Face_Emotion_Recognition_Vithika_Karan.ipynb 1
Watchlist Ideas
Search
ENG US
01:13 23-09-2024

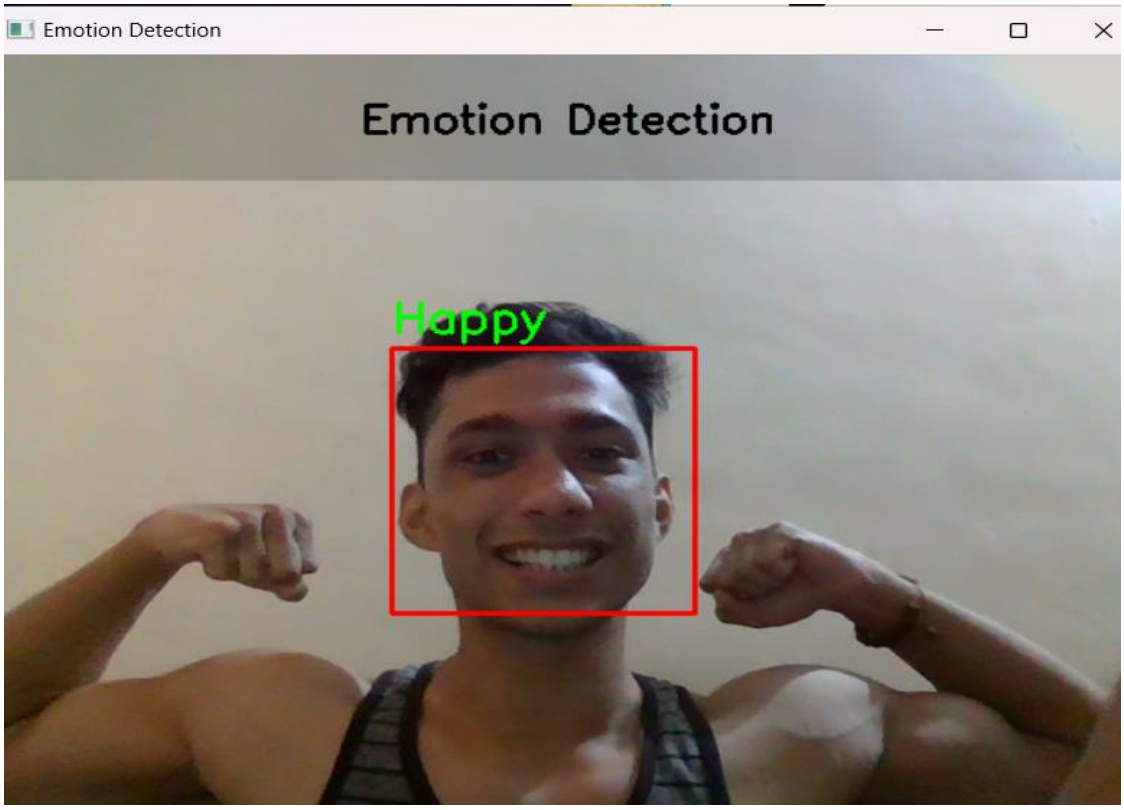

dataset - JupyterLab
SNA unit4 - Lecture notes 4 - Si
Docker Compose Type Convers
localhost:8888/lab/tree/Project/data%20%26%20resource/dataset
Google Chrome isn't your default browser
Set as default
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ data & resource / dataset /
Name
Modified
fer2013.zip
3 days ago
Face_Emotion_Recognition_V
Code
Python 3 (ipykernel)
import tensorflow.keras
import tensorflow as tf
# Load the pre-trained model (adjust the path if needed)
model = tf.keras.models.load_model('Project/data & resourceEmotion Detection Model/emotion_detection.keras')
# Load the Haar Cascade XML file for face detection
face_haar_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
# Start video capture
cap = cv2.VideoCapture(0)
if not cap.isOpened():
print("Error: Could not open video capture.")
else:
while cap.isOpened():
ret, img = cap.read()
if not ret:
print("Error: Could not read frame.")
break
height, width, channel = img.shape
# Create the heading section at the top of the frame
sub_img = img[0:int(height / 6), 0:int(width)]
heading = np.ones(sub_img.shape, dtype=np.uint8) * 0
result = cv2.addWeighted(sub_img, 0.82, heading, 0.18, 0)
Simple
Python 3 (ipykernel) | Idle
Mode: Command
Ln 3, Col 57
Face_Emotion_Recognition_Vithika_Karan.ipynb
Watchlist
Ideas
Search
ENG
US
01:13
23-09-2024

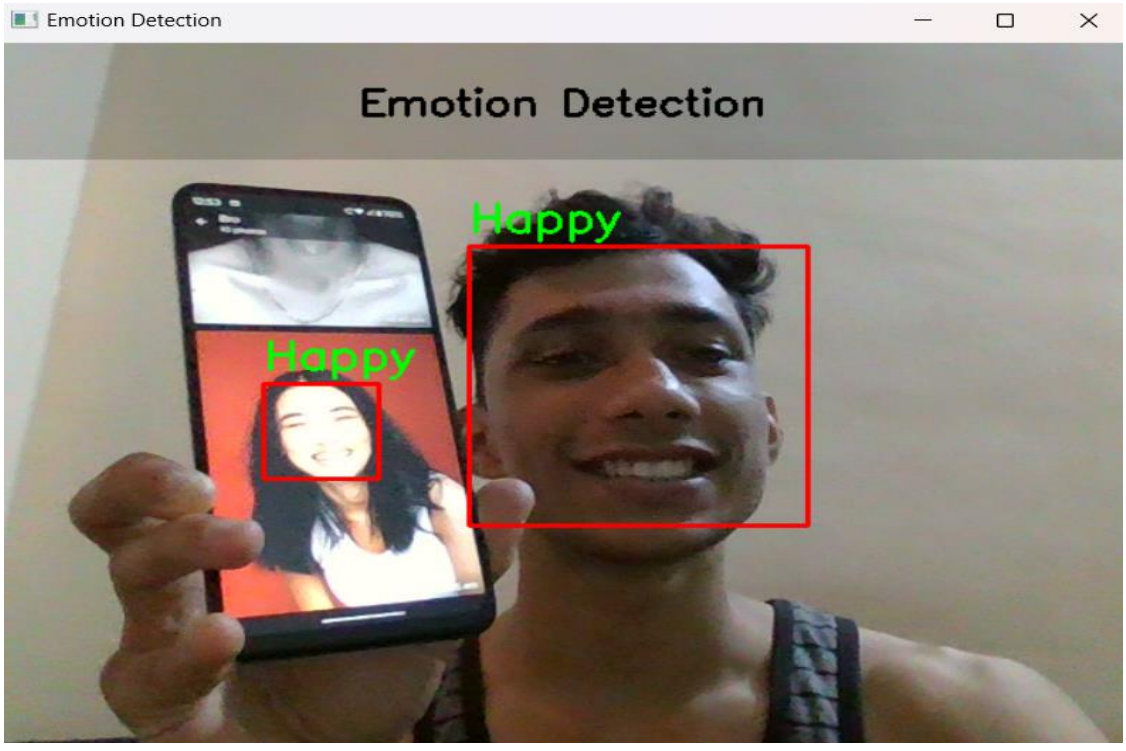

dataset - JupyterLab
SNA unit4 - Lecture notes 4 - Si
Docker Compose Type Convers
localhost:8888/lab/tree/Project/data%20%26%20resource/dataset
Google Chrome isn't your default browser
Set as default
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ data & resource / dataset /
Name
Modified
fer2013.zip
3 days ago
Face_Emotion_Recognition_V
Code
Python 3 (ipykernel)
# Add the heading text "Emotion Detection"
font = cv2.FONT_HERSHEY_SIMPLEX
font_scale = 0.8
font_thickness = 2
label_color = (0, 255, 0)
label = "Emotion Detection"
label_dimension = cv2.getTextSize(label, font, font_scale, font_thickness)[0]
textX = int((result.shape[1] - label_dimension[0]) / 2)
textY = int((result.shape[0] + label_dimension[1]) / 2)
cv2.putText(result, label, (textX, textY), font, font_scale, (0, 0, 0), font_thickness)
# Convert the image to grayscale for face detection
gray_image = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
# Detect faces in the frame
faces = face_haar_cascade.detectMultiScale(gray_image, scaleFactor=1.1, minNeighbors=5)
try:
# Loop over the detected faces and make emotion predictions
for (x, y, w, h) in faces:
# Draw rectangle around the face
cv2.rectangle(img, (x, y), (x + w, y + h), (0, 0, 255), 2)
# Extract region of interest (ROI) for emotion detection
roi_gray = gray_image[y:y + h, x:x + w]
roi_gray = cv2.resize(roi_gray, (48, 48))
image_pixels = tf.keras.preprocessing.image.img_to_array(roi_gray)
image_pixels = np.expand_dims(image_pixels, axis=0)
Simple
Python 3 (ipykernel) | Idle
Mode: Command
Ln 3, Col 57
Face_Emotion_Recognition_Vithika_Karan.ipynb
Watchlist
Ideas
Search
ENG
US
01:13
23-09-2024

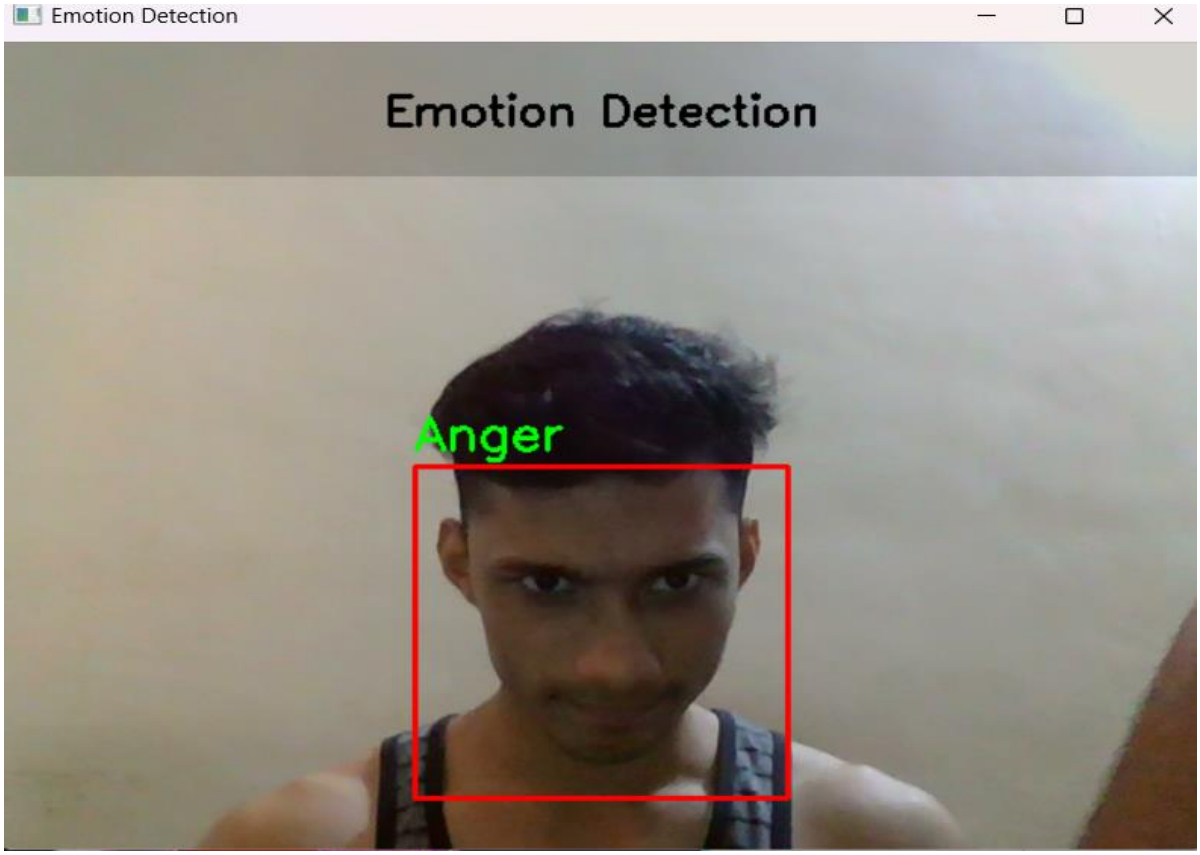

dataset - JupyterLab
SNA unit4 - Lecture notes 4 - Si
Docker Compose Type Convers
localhost:8888/lab/tree/Project/data%20%26%20resource/dataset
Google Chrome isn't your default browser
Set as default
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ data & resource / dataset /
Name
Modified
fer2013.zip
3 days ago
Face_Emotion_Recognition_V
Code
Python 3 (ipykernel)
cv2.rectangle(img, (x, y), (x + w, y + h), (0, 0, 255), 2)
# Extract region of interest (ROI) for emotion detection
roi_gray = gray_image[y:y + h, x:x + w]
roi_gray = cv2.resize(roi_gray, (48, 48))
image_pixels = tf.keras.preprocessing.image.img_to_array(roi_gray)
image_pixels = np.expand_dims(image_pixels, axis=0)
image_pixels /= 255 # Normalize pixel values
# Predict the emotion
predictions = model.predict(image_pixels)
max_index = np.argmax(predictions[0])
# Map the prediction to an emotion label
emotion_labels = ['Anger', 'Disgust', 'Fear', 'Happy', 'Sad', 'Surprised', 'Neutral']
emotion_prediction = emotion_labels[max_index]
# Display the predicted emotion on the frame
cv2.putText(img, emotion_prediction, (x, y - 10), font, 0.9, label_color, 2)
except Exception as e:
print(f"Error during prediction: {e}")
# Overlay the heading on the frame
img[0:int(height / 6), 0:int(width)] = result
# Show the frame with detected emotion
cv2.imshow('Emotion Detection', img)
Simple
Python 3 (ipykernel) | Idle
Mode: Command
Ln 3, Col 57
Face_Emotion_Recognition_Vithika_Karan.ipynb
Watchlist
Ideas
Search
ENG
US
01:13
23-09-2024

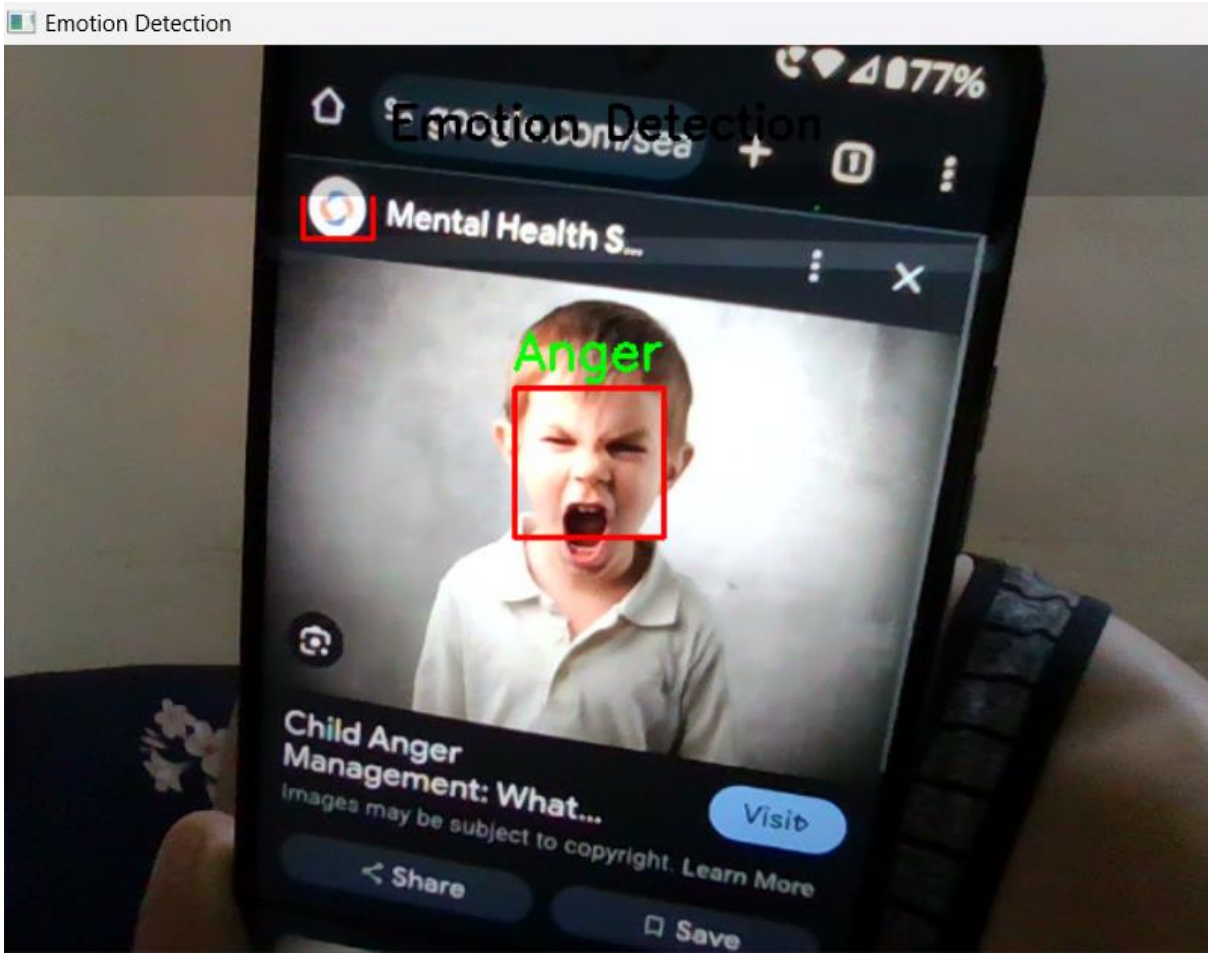

dataset - JupyterLab
SNA unit4 - Lecture notes 4 - Si
Docker Compose Type Convers
localhost:8888/lab/tree/Project/data%20%26%20resource/dataset
Google Chrome isn't your default browser
Set as default
File Edit View Run Kernel Tabs Settings Help
Filter files by name
/ data & resource / dataset /
Name
Modified
fer2013.zip
3 days ago
Face_Emotion_Recognition_V
Code
Python 3 (ipykernel)
max_index = np.argmax(predictions[0])
# Map the prediction to an emotion label
emotion_labels = ['Anger', 'Disgust', 'Fear', 'Happy', 'Sad', 'Surprised', 'Neutral']
emotion_prediction = emotion_labels[max_index]
# Display the predicted emotion on the frame
cv2.putText(img, emotion_prediction, (x, y - 10), font, 0.9, label_color, 2)
except Exception as e:
print(f"Error during prediction: {e}")
# Overlay the heading on the frame
img[0:int(height / 6), 0:int(width)] = result
# Show the frame with detected emotion
cv2.imshow('Emotion Detection', img)
# Press 'q' to exit the video feed
if cv2.waitKey(1) & 0xFF == ord('q'):
break
# Release the video capture and destroy all windows
cap.release()
cv2.destroyAllWindows(
The model pretty accurately detected all the emotions beside disgust. The training daata for the emotion class - disgust were pretty low as compared to others. Here are the results:
Simple
Python 3 (ipykernel) | Idle
Mode: Command
Ln 3, Col 57
Face_Emotion_Recognition_Vithika_Karan.ipynb
Watchlist
Ideas
Search
ENG
US
01:13
23-09-2024

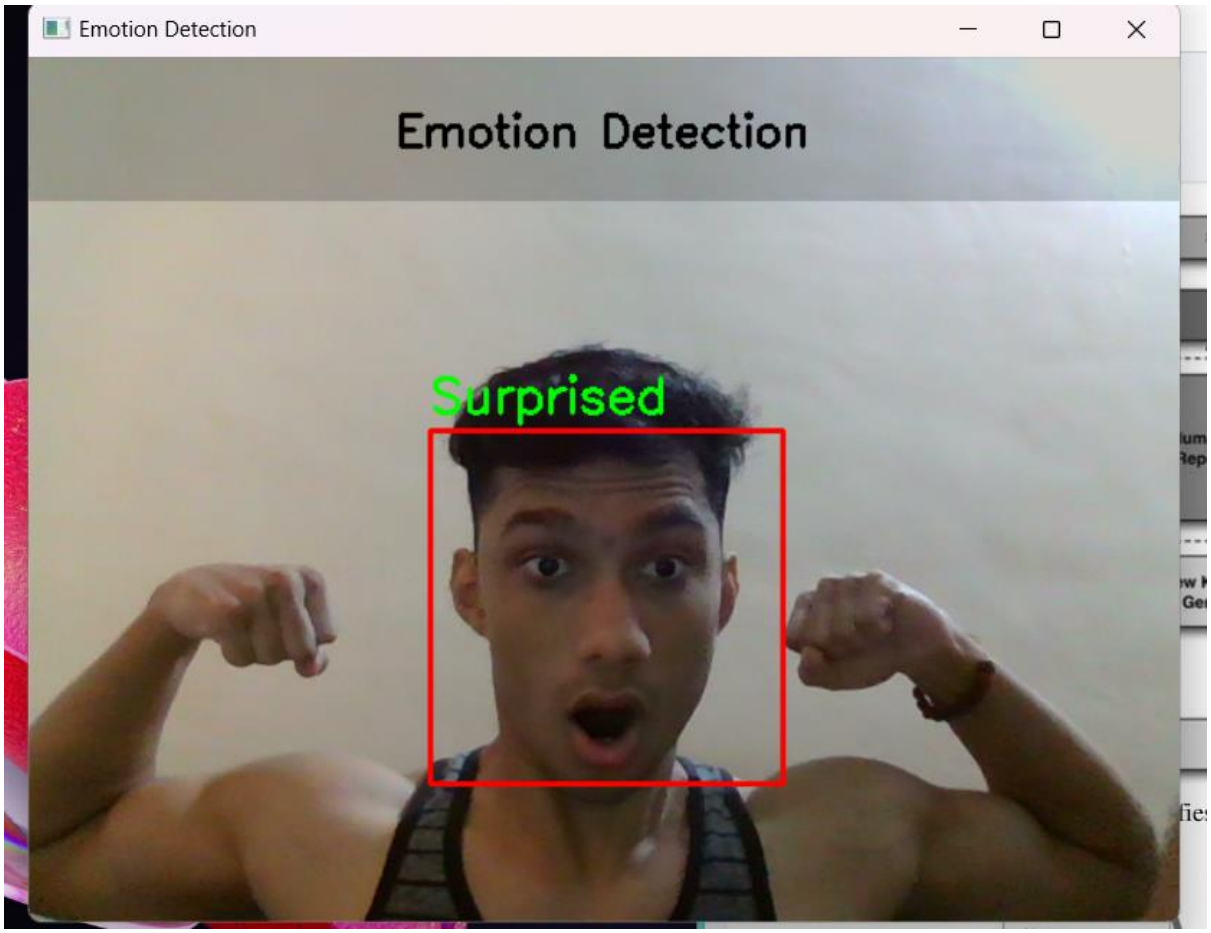
LIVE FACIAL EXPRESSION AND EMOTION RESULTS:

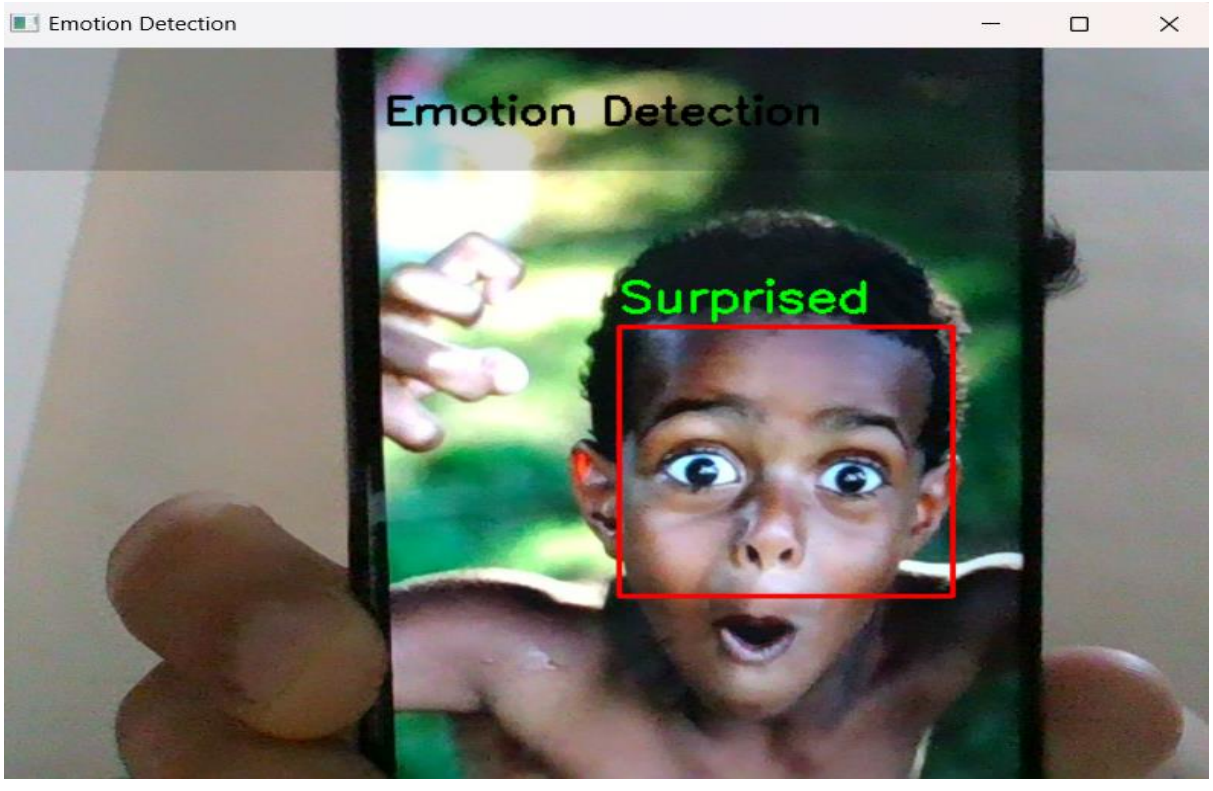


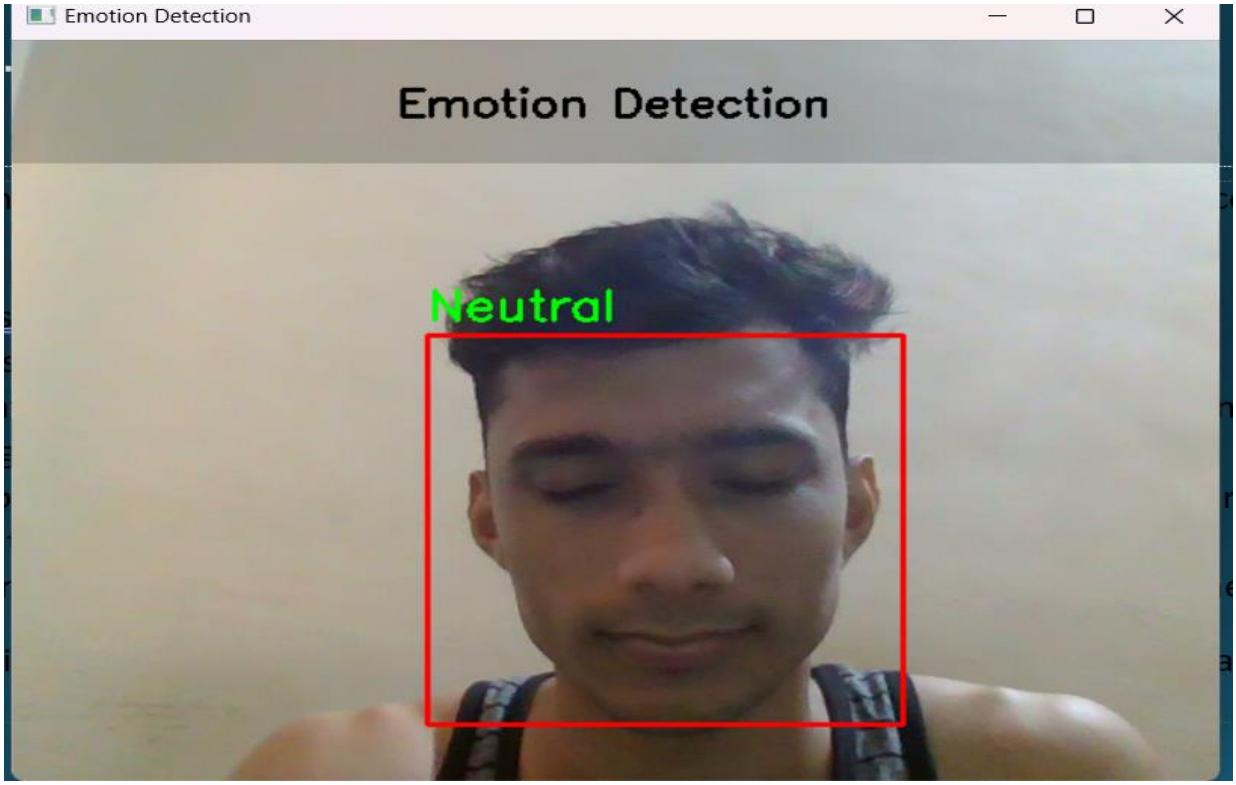


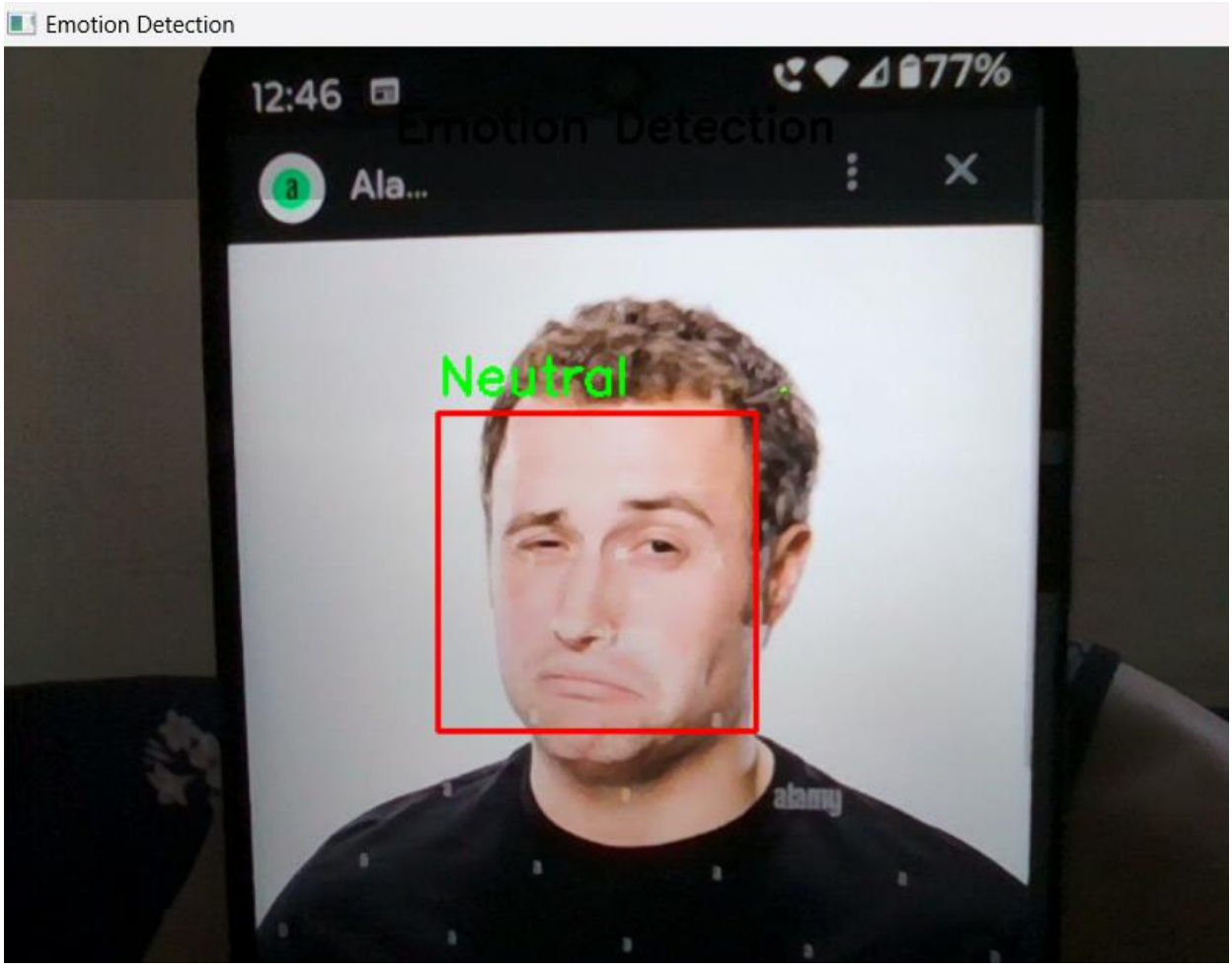

Emotion Detection
Emotion Detection
Anger

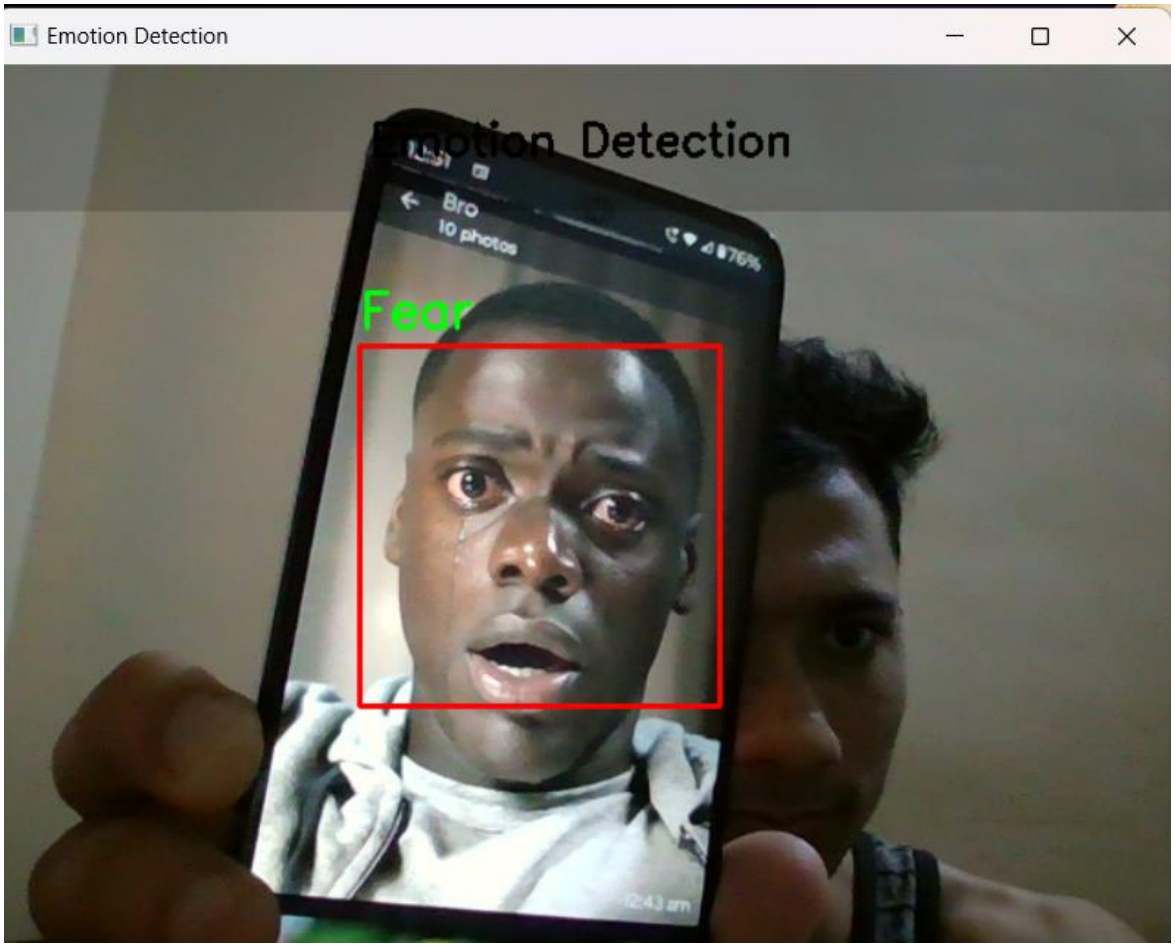

Emotion Detection
Emotion Detection
Mental Health S...
Anger
Child Anger Management: What...
Visit
Images may be subject to copyright. Learn More
Share
Save

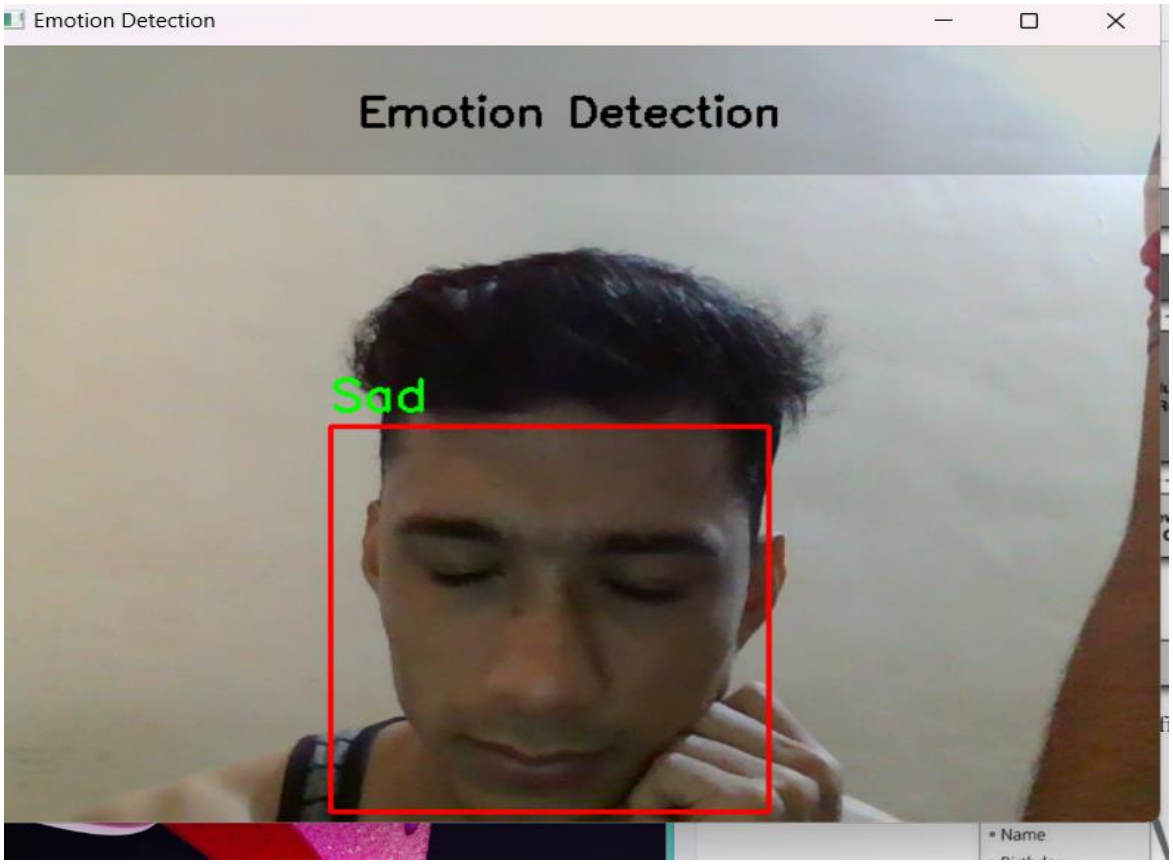

Emotion Detection
Emotion Detection
Surprised

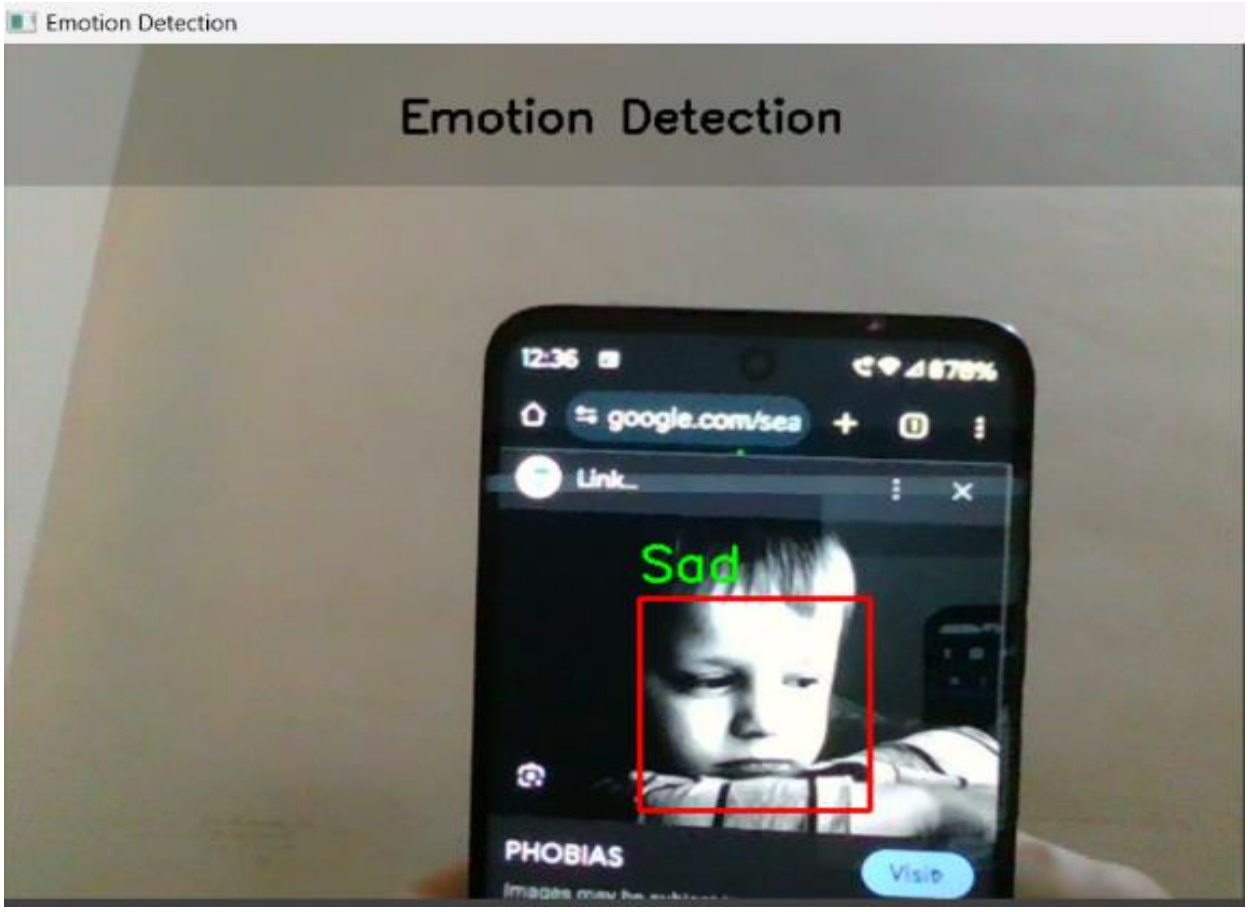

Emotion Detection
Emotion Detection
Surprised


Emotion Detection
Emotion Detection
Neutral


Emotion Detection
12:46
77%
Emotion Detection
Ala...
Neutral


Emotion Detection
Emotion Detection
Bro
10 photos
Fear


Emotion Detection
Emotion Detection
Sad
Name


Emotion Detection
Emotion Detection
12:36
google.com/sea
Link...
Sad
PHOBIAS

Conclusion

Some important conclusions in the project includes:

Best training result

At Epoch 30

Model training result were - accuracy: 0.5625 - loss: 1.2514 - val_accuracy: 0.8000 - val_loss: 0.4976

-here top val_accuracy of 0.8000 was achieved

At Epoch 39

Model training result were - accuracy: 0.6406 - loss: 1.0494 - val_accuracy: 0.2000 - val_loss: 1.1991 at epoch 39

-here top accuracy of 0.6406 was achieved

- Model Test Results:

```
113/113 ━━━━━━━━━━━━━━━━━━━━ 66s 584ms/step
              precision    recall  f1-score   support

           0       0.54      0.37      0.44       491
           1       0.45      0.09      0.15        55
           2       0.42      0.17      0.24       528
           3       0.73      0.87      0.79       879
           4       0.41      0.49      0.45       594
           5       0.68      0.72      0.70       416
           6       0.48      0.65      0.55       626

    accuracy                           0.57      3589
   macro avg       0.53      0.48      0.48      3589
weighted avg       0.55      0.57      0.54      3589
```

The predictions on the test set are pretty great with accuracy of 0.68 which implies generalization of the model is good as compared to the training results.

(0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral).

Overall Performance:

- **Accuracy**: 0.68 (68%), indicating the model correctly predicts the emotions 68% of the time.

Summary:

The model performs best on the “Happy” class with high precision, recall, and F1-score, indicating it can accurately and consistently identify happy faces. However, it struggles with the “Disgust” classes, showing lower precision and recall. The overall accuracy of 68% suggests the model generalizes well from the training data to the test data.

Note: disgust images in the above dataset were comparatively lesser compare to happy , sad etc....