

Face Expression

July 12, 2024

```
[44]: #Facial Expression Recognition_Palanichamy Naveen

# Listing the Contents of the train and test Directories

import os

# List files in the train directory
train_files = os.listdir(r'C:\Users\KPRIET\Downloads\archive')

# Loading Images from the Directory Structure

import os
import cv2
import numpy as np
from tensorflow.keras.utils import to_categorical
from sklearn.model_selection import train_test_split

# Define the path to the dataset
dataset_path = r'C:\Users\KPRIET\Downloads\archive'

# Define the emotion labels
emotion_labels = ['angry', 'disgust', 'fear', 'happy', 'neutral', 'sad', 'surprise']
num_classes = len(emotion_labels)

# Function to load images from a directory
def load_images_from_directory(directory, label):
    images = []
    labels = []
    for filename in os.listdir(directory):
        img_path = os.path.join(directory, filename)
        img = cv2.imread(img_path, cv2.IMREAD_GRAYSCALE)
        if img is not None:
            img = cv2.resize(img, (48, 48))
            img = img.astype('float32') / 255.0
            images.append(img)
            labels.append(label)
```

```

        return images, labels

# Load the training data
train_images = []
train_labels = []

for label, emotion in enumerate(emotion_labels):
    emotion_dir = os.path.join(dataset_path, 'train', emotion)
    images, labels = load_images_from_directory(emotion_dir, label)
    train_images.extend(images)
    train_labels.extend(labels)

# Convert lists to numpy arrays
train_images = np.array(train_images)
train_labels = np.array(train_labels)

# Expand dimensions to match the input shape for the model
train_images = np.expand_dims(train_images, -1)
train_labels = to_categorical(train_labels, num_classes=num_classes)

# Split the data into training and validation sets
X_train, X_val, y_train, y_val = train_test_split(train_images, train_labels,
    ↪test_size=0.2, random_state=42)

# Print the shape of the datasets
print('Training data shape:', X_train.shape)
print('Validation data shape:', X_val.shape)
print('Training labels shape:', y_train.shape)
print('Validation labels shape:', y_val.shape)

# Training the Model

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense,
    ↪Dropout

# Define the model
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(48, 48, 1)),
    MaxPooling2D((2, 2)),
    Dropout(0.25),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Dropout(0.25),
    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),

```

```

        Dropout(0.25),
        Flatten(),
        Dense(128, activation='relu'),
        Dropout(0.5),
        Dense(num_classes, activation='softmax')
    ])

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

    # Train the model
    model.fit(X_train, y_train, epochs=30, batch_size=64, validation_data=(X_val,
        y_val))

    # Save the model
    Face_Exp_Naveen = r'C:\Users\KPRIET\Downloads\Face_ Exp_Naveen.keras'
    model.save('Face_ Exp_Naveen.keras')

```

```

Training data shape: (22967, 48, 48, 1)
Validation data shape: (5742, 48, 48, 1)
Training labels shape: (22967, 7)
Validation labels shape: (5742, 7)
Epoch 1/30
359/359          21s 46ms/step -
accuracy: 0.2347 - loss: 1.8367 - val_accuracy: 0.3224 - val_loss: 1.7070
Epoch 2/30
359/359          15s 43ms/step -
accuracy: 0.3130 - loss: 1.6935 - val_accuracy: 0.4216 - val_loss: 1.5265
Epoch 3/30
359/359          16s 44ms/step -
accuracy: 0.4078 - loss: 1.5400 - val_accuracy: 0.4530 - val_loss: 1.4156
Epoch 4/30
359/359          16s 44ms/step -
accuracy: 0.4404 - loss: 1.4485 - val_accuracy: 0.4869 - val_loss: 1.3648
Epoch 5/30
359/359          17s 48ms/step -
accuracy: 0.4698 - loss: 1.3789 - val_accuracy: 0.5091 - val_loss: 1.3239
Epoch 6/30
359/359          18s 49ms/step -
accuracy: 0.4878 - loss: 1.3416 - val_accuracy: 0.5176 - val_loss: 1.2854
Epoch 7/30
359/359          17s 48ms/step -
accuracy: 0.4998 - loss: 1.3144 - val_accuracy: 0.5239 - val_loss: 1.2548
Epoch 8/30
359/359          16s 44ms/step -
accuracy: 0.5160 - loss: 1.2788 - val_accuracy: 0.5275 - val_loss: 1.2428

```

Epoch 9/30
 359/359 16s 44ms/step -
 accuracy: 0.5224 - loss: 1.2505 - val_accuracy: 0.5348 - val_loss: 1.2246

Epoch 10/30
 359/359 16s 45ms/step -
 accuracy: 0.5346 - loss: 1.2302 - val_accuracy: 0.5409 - val_loss: 1.2055

Epoch 11/30
 359/359 15s 42ms/step -
 accuracy: 0.5389 - loss: 1.2132 - val_accuracy: 0.5517 - val_loss: 1.1842

Epoch 12/30
 359/359 15s 42ms/step -
 accuracy: 0.5490 - loss: 1.1875 - val_accuracy: 0.5446 - val_loss: 1.1981

Epoch 13/30
 359/359 17s 47ms/step -
 accuracy: 0.5574 - loss: 1.1695 - val_accuracy: 0.5500 - val_loss: 1.1805

Epoch 14/30
 359/359 15s 43ms/step -
 accuracy: 0.5547 - loss: 1.1585 - val_accuracy: 0.5540 - val_loss: 1.1905

Epoch 15/30
 359/359 15s 42ms/step -
 accuracy: 0.5728 - loss: 1.1379 - val_accuracy: 0.5603 - val_loss: 1.1623

Epoch 16/30
 359/359 15s 42ms/step -
 accuracy: 0.5654 - loss: 1.1385 - val_accuracy: 0.5660 - val_loss: 1.1524

Epoch 17/30
 359/359 15s 43ms/step -
 accuracy: 0.5788 - loss: 1.1183 - val_accuracy: 0.5662 - val_loss: 1.1479

Epoch 18/30
 359/359 15s 43ms/step -
 accuracy: 0.5808 - loss: 1.1151 - val_accuracy: 0.5677 - val_loss: 1.1489

Epoch 19/30
 359/359 16s 46ms/step -
 accuracy: 0.5837 - loss: 1.1055 - val_accuracy: 0.5665 - val_loss: 1.1458

Epoch 20/30
 359/359 17s 48ms/step -
 accuracy: 0.5870 - loss: 1.0857 - val_accuracy: 0.5596 - val_loss: 1.1641

Epoch 21/30
 359/359 16s 45ms/step -
 accuracy: 0.5950 - loss: 1.0794 - val_accuracy: 0.5700 - val_loss: 1.1463

Epoch 22/30
 359/359 17s 48ms/step -
 accuracy: 0.6030 - loss: 1.0574 - val_accuracy: 0.5763 - val_loss: 1.1466

Epoch 23/30
 359/359 17s 46ms/step -
 accuracy: 0.5996 - loss: 1.0511 - val_accuracy: 0.5749 - val_loss: 1.1331

Epoch 24/30
 359/359 16s 45ms/step -
 accuracy: 0.6029 - loss: 1.0346 - val_accuracy: 0.5721 - val_loss: 1.1354

```
Epoch 25/30
359/359          16s 45ms/step -
accuracy: 0.6077 - loss: 1.0329 - val_accuracy: 0.5777 - val_loss: 1.1329
Epoch 26/30
359/359          17s 47ms/step -
accuracy: 0.6091 - loss: 1.0281 - val_accuracy: 0.5808 - val_loss: 1.1343
Epoch 27/30
359/359          16s 43ms/step -
accuracy: 0.6139 - loss: 1.0249 - val_accuracy: 0.5763 - val_loss: 1.1345
Epoch 28/30
359/359          16s 45ms/step -
accuracy: 0.6190 - loss: 1.0072 - val_accuracy: 0.5758 - val_loss: 1.1513
Epoch 29/30
359/359          16s 46ms/step -
accuracy: 0.6167 - loss: 0.9992 - val_accuracy: 0.5782 - val_loss: 1.1503
Epoch 30/30
359/359          18s 51ms/step -
accuracy: 0.6195 - loss: 1.0010 - val_accuracy: 0.5789 - val_loss: 1.1621
```

[]: