

3. FACE RECOGNITION

EX.N0 : 3	CNN MODEL FOR FACE RECOGNITION
<u>DATE : 03/03/2025</u>	

AIM:

To build and train a Convolutional Neural Network (CNN) for face recognition using a dataset such as the Labelled Faces in the Wild (LFW) dataset. This model will be capable of recognizing faces from the dataset.

ALGORITHM:

Step 1: Import necessary libraries.

Step 2: Load and pre-process the LFW dataset.

Step 3: Build the CNN model using Keras Sequential API.

Step 4: Compile the model with appropriate loss function and optimizer.

Step 5: Train the model on the training dataset.

Step 6: Evaluate the model on the test dataset.

Step 7: Visualize a few predictions to validate performance.

PROGRAM:

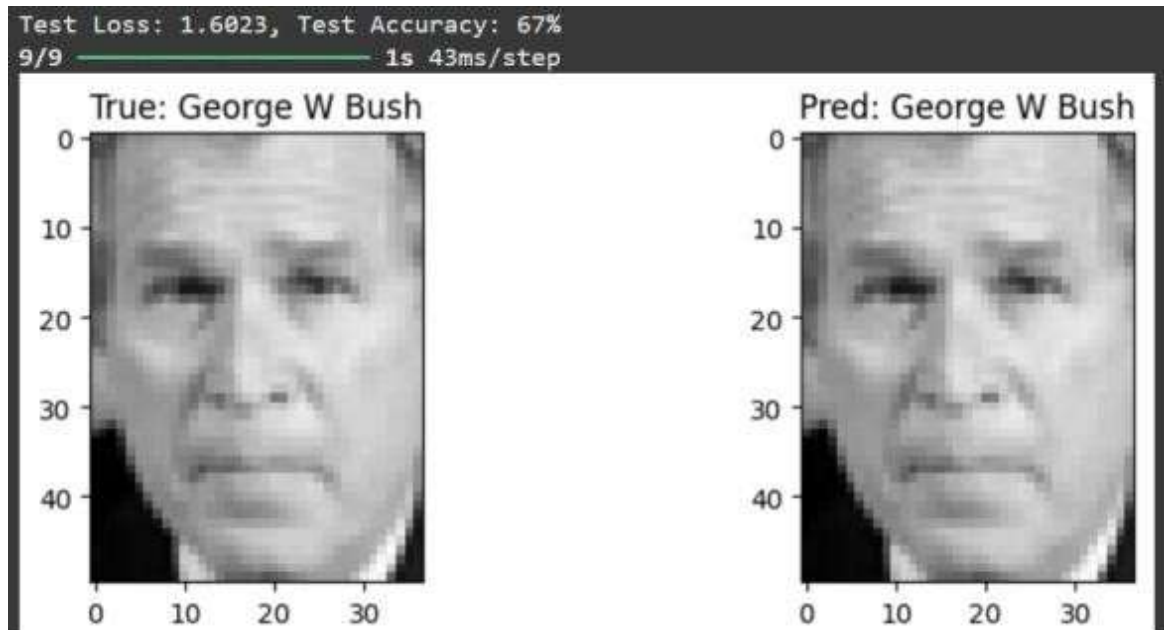
```
import tensorflow as tf
from tensorflow.keras import datasets, layers, models
from sklearn.model_selection import train_test_split
from sklearn import datasets
import numpy as np
```

```

lfw_data = datasets.fetch_lfw_people(min_faces_per_person=70, resize=0.4)
images = lfw_data.images # Image data
labels = lfw_data.target # Labels
images = images / 255.0
X_train, X_test, y_train, y_test = train_test_split(images, labels, test_size=0.3, random_state=42)
model = models.Sequential([
    layers.InputLayer(input_shape=(50, 37, 3)), # LFW image shape
    layers.Conv2D(32, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(128, (3, 3), activation='relu'),
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dense(len(lfw_data.target_names), activation='softmax') # Output layer
])
model.compile(optimizer='adam',
    loss='sparse_categorical_crossentropy',
    metrics=['accuracy'])
model.fit(X_train, y_train, epochs=10, batch_size=32)
test_loss, test_accuracy = model.evaluate(X_test, y_test)
print(f'Test Accuracy: {test_accuracy * 100:.2f}%')
model.save("face_recognition_model.h5")
import matplotlib.pyplot as plt
predictions = model.predict(X_test)
for i in range(5):
    plt.imshow(X_test[i])
    plt.title(f'Predicted: {lfw_data.target_names[predictions[i].argmax()]}, Actual:
    {lfw_data.target_names[y_test[i]]}')
    plt.show()

```

OUTPUT:



RESULT:

Thus the Program has been executed successfully and verified.