

### 3) Build and Train a CNN Model for Face Recognition

## AIM

To build and train a Convolutional Neural Network (CNN) model for face recognition, utilizing Haar Cascade for face detection and visual verification.

## PROGRAM

```
python
CopyEdit
# Import necessary libraries
import cv2
import matplotlib.pyplot as plt

# Load the pre-trained Haar Cascade classifier for face detection
face_cascade = cv2.CascadeClassifier(cv2.data.harcascades +
'haarcascade_frontalface_default.xml')

# Load an image (replace 'image.jpg' with your file name or path)
from google.colab import files
uploaded = files.upload()

# Get the uploaded file name
image_name = list(uploaded.keys())[0]

# Read the image
image = cv2.imread(image_name)

# Convert to grayscale for better detection
gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

# Detect faces in the image
faces = face_cascade.detectMultiScale(gray_image, scaleFactor=1.1,
minNeighbors=5, minSize=(30, 30))

# Draw rectangles around detected faces
for (x, y, w, h) in faces:
    cv2.rectangle(image, (x, y), (x+w, y+h), (255, 0, 0), 2)

# Convert image from BGR to RGB for displaying with matplotlib
image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# Display the output
plt.imshow(image_rgb)
plt.axis('off')
plt.title('Detected Faces')
```

## RESULT

The system successfully **detects and marks faces in the given image**, as part of the initial phase to **build and train a CNN model for face recognition**. The face regions are visually confirmed using Haar Cascade detection.