Build and train a CNN model for Face recognition

1.Library Imports and Dataset Path Setup

- The script uses TensorFlow, OpenCV, NumPy, scikit-learn, and Matplotlib.
- It defines the target image size and constructs the path to the dataset directory, where face images are stored with filenames in the format `<label>_<index>.jpg`.

2. Mount Google Drive

- Mounts Google Drive in Colab to access the dataset stored in a specific folder (`Dataset/Faces`).

3. Image Loading and Label Extraction

- A custom function loads images from the dataset folder.
- Each image is resized to 100x100 and assigned a label based on the filename prefix (e.g., "Alice_1.jpg" → label "Alice").
- The script skips any files that are unreadable or corrupted.

4. Label Encoding

- String labels (names) are encoded into numerical labels using `LabelEncoder`.
- The number of unique classes (faces) is determined for use in the output layer.

5. Dataset Splitting and Normalization

- The image data and encoded labels are split into training and testing sets (80/20 split).
- Pixel values are normalized to the range [0, 1].

6. CNN Model Definition

- A Sequential CNN model is created with the following architecture:
    - Four convolutional blocks with increasing filter sizes (32 → 128), each followed by max pooling.
    - A flatten layer to convert feature maps into a vector.
    - A dense layer with 512 units and ReLU activation.
    - An output softmax layer with `num_classes` units for multi-class classification.

7. Model Compilation and Training

- The model is compiled with the Adam optimizer and sparse categorical crossentropy loss.
- It is trained for 10 epochs with a batch size of 32 and validated on the test data.

8. Model Evaluation

- The model is evaluated on the test set and the test accuracy is printed.

9. Face Detection and Visualization

- OpenCV's Haar Cascade classifier is used to detect faces in images (`sample1.jpg` and `sample2.jpg`).
- Detected faces are highlighted with green rectangles.
- The original and annotated images are displayed side by side using Matplotlib.