FACE RECOGNITION SYSTEM DOCUMENTATION

Introduction

This document provides an overview and usage instructions for a face recognition system implemented in Python. The system consists of three main components:

1. **Face Dataset Creation:** Captures facial images and organizes them into a dataset.
2. **Recognizer Training:** Trains a face recognizer using the created dataset.
3. **Real-time Face Recognition:** Utilizes the trained recognizer to perform face recognition in real-time.

Components

1. Face Dataset Creation

Class: CreateDataset

- **Attributes:**
  - face_cascade_path: Path to the Haar Cascade XML file for face detection.
  - face_id: Initial face ID for labeling captured images.
  - face_images_per_id: Number of face images to capture per ID.
  - face_counter: Initial face counter.
  - dataset_folder: Path to the main dataset folder.
- **Methods:**
  - detect_faces(img, gray): Detects faces in a given image.
  - draw_rectangles_and_save_images(img, gray, faces, subfolder): Draws rectangles around detected faces, saves images, and displays them.
  - start_capturing(): Initiates the face capturing process.

Usage:

```
face_dataset_creator = CreateDataset(
    face_id=args.face_id,
    face_images_per_id=args.images_per_id,
    face_counter=args.face_counter,
    dataset_folder=args.dataset_path,
)

face_dataset_creator.start_capturing()
```

2. Recognizer Training

Class: FaceTrainer

- **Attributes:**
  - data_path: Path to the dataset folder.
  - trainer_folder: Path to the folder where the trained recognizer will be saved.
  - cascade_path: Path to the Haar Cascade XML file for face detection.
  - recognizer: LBPH Face Recognizer.
  - detector: Cascade Classifier for face detection.

- **Methods:**
  - _get_image_id(image_path): Extracts the face ID from an image path.
  - _get_images_and_labels(): Processes images from the dataset and extracts face samples and corresponding IDs.
  - train_recognizer(): Trains the LBPH Face Recognizer.

Usage:

```
face_trainer = FaceTrainer(data_path=args.dataset_path)
if face_trainer.train_recognizer():
   print("Recognizer trained successfully!")
else:
   print("Failed to train recognizer.")
```

3. Real-time Face Recognition

Class: RealTimeFaceRecognizer

- **Attributes:**
  - dataset_path: Path to the dataset folder.
  - trainer_path: Path to the trained recognizer file.
  - cascade_path: Path to the Haar Cascade XML file for face detection.
  - recognizer: LBPH Face Recognizer.
  - face_cascade: Cascade Classifier for face detection.
  - font: Font type for displaying text.
  - names: List of names extracted from the dataset.
- **Methods:**
  - get_names_from_dataset(): Retrieves names from the dataset.
  - load_recognizer(): Loads the trained recognizer.
  - detect_and_recognize_faces(img): Detects and recognizes faces in real-time.
  - start_recognition(): Initiates the real-time face recognition process.

Usage:

```
real_time_recognizer = RealTimeFaceRecognizer(
   dataset_path=args.dataset_path,
   trainer_path=args.trainer_path,
   cascade_path=args.cascade_path,
)

real_time_recognizer.start_recognition()
```

Conclusion

This face recognition system provides a comprehensive solution for capturing facial data, training a recognizer, and performing real-time face recognition. Users can customize parameters and paths based on their requirements.
