Enhanced Face Recognition System

🚀 Features

Core Capabilities

- **Advanced Image Enhancement: Noise reduction, CLAHE, histogram equalization, and sharpening**
- **Multiple Detection Methods: HOG, CNN, and OpenCV Haar Cascade fallback systems**
- **Database Storage: SQLite-based embedding storage with quality metrics**
- **Enhanced Matching: Multiple similarity algorithms with weighted scoring**
- **Comprehensive Testing: 6 different testing modes for various use case**

Image Processing Enhancements

- **Non-local means denoising for noise reduction**
- **Contrast Limited Adaptive Histogram Equalization (CLAHE)**
- **Unsharp mask filtering for better sharpness**
- **Automatic image resizing for optimal face detection**
- **Multi-level image enhancement with PIL and OpenCV**

Testing Modes

1. **Testing: Process all images in testing folder**
2. **Single Image Testing: Test individual images**
3. **Interactive Mode: Real-time image path input**
4. **Detailed Batch Results: Comprehensive statistics and reporting**
5. **Database Information Batch: View stored embeddings and quality metrics**
6. **Menu-Driven Interface: User-friendly navigation system**

📊 Testing Modes

Mode 1: Batch Testing
Tests all images in testing_images/ folder with real-time detailed analysis.

Mode 2: Single Image Testing
Test one specific image with full diagnostic information.

Mode 3: Interactive Testing
Continuous testing mode - enter image paths until you quit.

Mode 4: Detailed Batch Results
Batch testing with comprehensive statistics and success rate analysis.

Mode 5: Database Information
View stored face embeddings, quality scores, and person statistics.

Mode 6: Exit
Clean program termination.

🔧 Advanced Features

Image Enhancement Pipeline

1. **Noise Reduction: Non-local means denoising**
2. **Contrast Enhancement: Histogram equalization + CLAHE**
3. **Sharpening: Gaussian blur with unsharp mask**
4. **Quality Assessment: Automatic image quality scoring**
5. **Intelligent Resizing: Maintains aspect ratio for optimal detection**

Multi-Method Face Detection

1. **Primary: HOG model with upsampling**
2. **Secondary: CNN model (if available)**
3. **Fallback: OpenCV Haar Cascade**
4. **Aggressive: Higher upsampling for difficult cases**

Advanced Similarity Matching

- **Cosine Similarity: Primary similarity metric**
- **Euclidean Distance: Secondary validation**
- **Dot Product: Tertiary confirmation**
- **Weighted Scoring: Quality and confidence-based weighting**
- **Dynamic Thresholding: Adaptive matching thresholds**

☑ Performance Metrics

The system provides comprehensive metrics:

- **Overall Similarity Score: Weighted combination of multiple metrics**
- **Individual Similarity Scores: Cosine, Euclidean, Dot Product**

- **Image Quality Assessment: Sharpness and brightness analysis**
- **Detection Confidence: Face size and clarity metrics**
- **Success Rate Statistics: Batch processing performance**

🎨 Sample Output

Processing: test_photo.jpg
✓ Found 1 face(s)

Analyzing Face #1:
✅ MATCH FOUND!
Person: John Doe
ID: 5
Overall Similarity: 0.847
Cosine Similarity: 0.823
Euclidean Similarity: 0.756
Quality Score: 0.678
🎯 This photo is matching with ID 5 and name John Doe
□ High Confidence Match

⚙️ Configuration

Similarity Thresholds
Default threshold: 0.5 (adjustable in code)

- **High Confidence: > 0.8**
- **Medium Confidence: 0.65 - 0.8**
- **Low Confidence: 0.5 - 0.65**

Image Enhancement Parameters

- **Denoising strength: 10**
- **CLAHE clip limit: 3.0**
- **Sharpness enhancement: 2.5x**
- **Minimum image size: 300x300 pixels**

📝 Requirements

Core Dependencies

- **opencv-python>=4.5.0**
- **numpy>=1.21.0**
- **face-recognition>=1.3.0**

- **Pillow>=8.0.0**
- **scikit-learn>=1.0.0**
- **scikit-image>=0.18.**

🗀 Project Structure

```
your_project/
│
├── face_training.py        ← training script
├── face_testing.py         ← recognition script
├── face_embeddings.db      ← database created after training
│
├── recognize_images/       ← training images
│   ├── person1/
│   │   ├── image1.jpg
│   │   └── image2.jpg
│   │
│   └── person2/
│       └── image1.jpg
│
└── testing_images/         ← images to test recognition
    ├── test1.jpg
    ├── test2.jpg
    └── test3.jpg
```