

Face Detection & Recognition System

Overview

This document provides a detailed technical review of the face recognition system implemented using OpenVINO, FAISS, and other supporting technologies. It explains the models used, their input/output specifications, and the underlying processes for face detection, recognition, and similarity search.

Technologies Used

1. OpenVINO

OpenVINO is utilized for optimized inference of face detection and recognition models. It supports hardware acceleration on Intel CPUs, GPUs, and NPUs.

Key Components:

- **SCRFD (Face Detection):**
 - Model: `det_10g.onnx` (converted to OpenVINO IR format).
 - Input Shape: (1, 3, 360, 640) (Batch size, Channels, Height, Width).
 - Output Shape: Multiple tensors for bounding boxes, keypoints, and confidence scores.
 - Functionality: Detects faces in an image and optionally filters blurred faces using Laplacian variance.
 - **AdaFace (Face Recognition):**
 - Model: `adaface_r50.xml` (OpenVINO IR format for R50 architecture).
 - Input Shape: (1, 3, 112, 112) (Batch size, Channels, Height, Width).
 - Output Shape: (1, 512) (512-dimensional feature vector for face embeddings).
 - Functionality: Extracts high-dimensional embeddings for face recognition.
-

2. FAISS

FAISS is used for efficient similarity search and clustering of face embeddings. It supports both flat indexing and advanced techniques like IVF (Inverted File Index) and HNSW (Hierarchical Navigable Small World).

Key Components:

- **IndexFlatIP:**

- Metric: Inner Product (Cosine Similarity after normalization).
 - Used for small-scale datasets.
 - **IndexIVFFlat:**
 - Metric: Inner Product.
 - Parameters: `nlist=1024`, `nprobe=100`.
 - Used for large-scale datasets with faster search capabilities.
 - **IndexHNSWFlat:**
 - Metric: Inner Product (Cosine Similarity after normalization).
 - Parameters:
 - * `dim`: Dimensionality of the feature vectors (e.g., 512 for AdaFace embeddings).
 - * `M=32`: Number of neighbors per node in the graph.
 - Suitable for scenarios requiring high-speed retrieval with large embedding databases.
-

3. OpenCV

OpenCV is used for image preprocessing, video processing, and visualization. It handles tasks like resizing, normalization, and drawing bounding boxes.

4. Python

Python serves as the core programming language for implementation. Libraries like NumPy, JSON, and argparse are used for data handling and configuration.

Hardware Details

- **CPU:** Intel(R) Core(TM) Ultra 5 125H
 - **CPU Cores:** 18
 - **Memory Size:** 32 GB
 - **GPU:** Intel(R) Graphics (iGPU)
 - **NPU:** Intel(R) AI Boost - integrated NPU in **Meteor Lake** and newer CPUs
 - **Operating System:**
 - OS: Ubuntu 22.04
 - Kernel Version: 6.8.0-40-generic
-

Models and Their Specifications

1. SCRFD (Face Detection)

- **Model File:** `det_10g.onnx` (converted to OpenVINO IR format).
- **Input Shape:** (1, 3, 360, 640) (Batch size, Channels, Height, Width).
- **Output Shape:**
 - Bounding Boxes: (N, 4) (x_min, y_min, x_max, y_max).
 - Keypoints: (N, 10) (x, y coordinates for 5 keypoints).
 - Confidence Scores: (N,) (Detection confidence for each face).
- **Blur Detection:**
 - Threshold: `blur_threshold=50.0` (Laplacian variance).
 - Functionality: Filters out blurred faces before recognition.

2. AdaFace (Face Recognition)

- **Model File:** `adaface_r50.xml` (OpenVINO IR format for R50 architecture).
 - **Input Shape:** (1, 3, 112, 112) (Batch size, Channels, Height, Width).
 - **Output Shape:** (1, 512) (512-dimensional feature vector).
 - **Normalization:**
 - Mean: 127.5.
 - Standard Deviation: 127.5.
 - Input Range: [-1, 1].
 - **Similarity Metric:**
 - Cosine Similarity: $1 - \text{cosine_distance}$.
-

Workflow

1. Face Detection

- **Input:** Raw image or video frame.
- **Process:**
 - Resize to (640, 360).
 - Pass through SCRFD model.
 - Extract bounding boxes and keypoints.
 - Filter blurred faces using Laplacian variance.
- **Output:** Bounding boxes and keypoints for detected faces.

2. Face Recognition

- **Input:** Cropped face image (based on bounding box).
- **Process:**
 - Resize to (112, 112).
 - Normalize pixel values to [-1, 1].
 - Pass through AdaFace model.

- Extract 512-dimensional embedding.
- **Output:** Face embedding.

3. Similarity Search

- **Input:** Face embedding.
 - **Process:**
 - Normalize embedding using L2 normalization.
 - Search against FAISS index.
 - Retrieve top matches based on cosine similarity.
 - **Output:** Matched embeddings and similarity scores.
-

Performance Metrics

1. Face Detection

- **Average Detection Time:** ~10-15ms per frame (on GPU).
- **Accuracy:** High precision for frontal faces; reduced performance for occluded faces.

2. Face Recognition

- **Embedding Extraction Time:** ~5-10ms per face (on NPU).
- **Embedding Size:** 512 dimensions.
- **Similarity Thresholds:**
 - ≥ 0.45 : Likely the same person.
 - < 0.45 : Different person.

3. FAISS Search

- **Search Time:** ~10ms per query (IndexFlatIP with 100k embeddings).
- **Search Time:** ~1ms per query (IndexIVFFlat with 100k embeddings).
- **Search Time:** ~1ms per query (IndexHNSWFlat with 100k embeddings).
- **Index Size:** Scalable to millions of embeddings.

4. Encoding Time

- **Average Time Taken:** ~10ms per frame.

5. Total Frame Time

- **Total Time Taken:** ~50-60ms for a frame.

Key Observations:

- When running on `ONNX backend`, the average FPS is around **4-5**.

- While when running on **OpenVINO backend**, the average FPS is around **8-10** with search Index as **IndexFlatIP**, giving the best results.
- When running on **OpenVINO backend**, the average FPS is around **15-19** with search Index as **IndexIVFFlat**, with results very close to **IndexFlatIP**.
- When running on **OpenVINO backend**, the average FPS is around **17-21** with search Index as **IndexHNSWFlat**, but the results are poor.
- As FD + FR is only around 30% of total frame time so FP32, FP16 and INT8 are all performing similar, with difference of 0-2 FPS.
- **SCRFD** model at full precision is detecting blurred faces too so blur threshold is set to **50.0**.
- **SCRFD** model at half precision is detecting less blurred faces, so blur threshold is set to **25.0**.
- As we increase the blur threshold, the number of blurred faces detected is reduced and hence FPS is increased.
- **Tunable Parameters:**
 - **blur_threshold**: Threshold for detecting blurred faces.
 - **nlist**: Number of clusters for **IndexIVFFlat**.
 - **nprobe**: Number of probes for **IndexIVFFlat**.
 - **M**: Number of neighbors per node in the graph for **IndexHNSWFlat**.
 - **dim**: Dimensionality of the feature vectors for **IndexHNSWFlat**.
 - **cosine_threshold**: Similarity threshold for face recognition.

Sample Output Time:

Frames	FPS	FD	FR	Search	Draw	Encode	Frame Total
30	14.40	0.013	0.008	0.002	0.000	0.010	0.054
60	15.70	0.013	0.010	0.002	0.000	0.009	0.058
90	16.10	0.014	0.009	0.001	0.000	0.008	0.055
120	16.22	0.015	0.011	0.004	0.000	0.015	0.072
150	16.14	0.017	0.010	0.002	0.000	0.009	0.071
180	15.45	0.014	0.011	0.001	0.000	0.010	0.087
210	14.90	0.013	0.010	0.002	0.000	0.009	0.076
240	14.55	0.015	0.008	0.001	0.000	0.010	0.079
270	14.36	0.015	0.010	0.001	0.000	0.009	0.077
300	14.11	0.014	0.010	0.001	0.000	0.009	0.077
330	14.02	0.012	0.010	0.001	0.000	0.009	0.066
360	14.01	0.014	0.008	0.002	0.000	0.010	0.079
390	13.93	0.016	0.010	0.001	0.000	0.009	0.085
420	13.98	0.018	0.009	0.001	0.000	0.009	0.061
450	14.19	0.011	0.011	0.001	0.000	0.009	0.033
480	14.49	0.017	0.012	0.001	0.000	0.010	0.041
510	14.75	0.012	0.011	0.002	0.000	0.011	0.050
540	14.81	0.014	0.010	0.001	0.000	0.009	0.056
570	14.76	0.016	0.010	0.001	0.000	0.010	0.058
600	14.75	0.015	0.008	0.003	0.000	0.011	0.069

Frames	FPS	FD	FR	Search	Draw	Encode	Frame Total
630	14.69	0.016	0.010	0.002	0.000	0.011	0.071
660	14.71	0.012	0.009	0.001	0.000	0.010	0.067
690	14.79	0.012	0.010	0.002	0.000	0.009	0.045
720	15.02	0.019	0.015	0.003	0.000	0.010	0.046
750	15.22	0.015	0.014	0.002	0.000	0.014	0.046
780	15.43	0.015	0.013	0.002	0.000	0.009	0.055
810	15.63	0.014	0.013	0.002	0.000	0.008	0.037

File Structure

1. Model Files

- Located in `/home/sr/ov_fr/buffalo_1/`.
- Includes SCRFD and AdaFace models in ONNX and OpenVINO formats.

2. Database

- Located in `/home/sr/ov_fr/database/`.
- Contains scripts for embedding generation and database management.
- Also contains the original images and embedding files.

3. Face Detection

- Located in `/home/sr/ov_fr/fd/`.
- Includes SCRFD implementation and blur detection logic.

4. Face Recognition

- Located in `/home/sr/ov_fr/fr/`.
- Includes AdaFace implementations for ONNX and OpenVINO.

5. Utilities

- Located in `/home/sr/ov_fr/utlis/`.
- Includes scripts for embedding export, JSONL updates, and testing.

6. Results

- Located in `/home/sr/ov_fr/videos/output/ov`.
- Contains annotated videos.
- Also located in google drive. Structure of results is:
 - No_blur
 - * result_{fd_precision}_{fr_precision}.mp4
 - 25_blur

```
    * result_{fd_precision}_{fr_precision}.mp4
- 50_blur
    * result_{fd_precision}_{fr_precision}.mp4
```

Conclusion

This document provides a comprehensive technical overview of the face recognition system. It highlights the models, workflows, and performance metrics of the entire solution.