ChatGPT

# Open-Source Face Recognition Libraries (2024–2025)

The leading open-source face-recognition libraries (MIT/Apache-2.0 licensed) all run entirely offline (no cloud calls) and do not log data, preserving privacy. Key criteria are **accuracy on "in-the-wild" images**, **fast CPU inference**, and **ease of use**. Top candidates include **InsightFace**, **DeepFace (Serengil's library)**, **dlib** (and wrappers like Face_Recognition), plus legacy toolkits like **OpenFace**. All support modern Python (typically Python ≥3.7 or 3.9). For example, DeepFace requires Python ≥3.7 [1] (so Python 3.9+ is fine), and InsightFace has a PyPI package for Python 3 [2]. These libraries provide local face detection, alignment and embedding without any external API or cloud. Below we summarize each, then compare InsightFace, DeepFace and dlib in detail.

- **InsightFace (MIT)** – A state-of-the-art 2D/3D face analysis toolkit. Its **ArcFace/CosFace models** are **state-of-the-art**: e.g. ArcFace achieves ~99.4% on LFW [3]. It's packaged for Python (`pip install insightface` [2]) and uses ONNX Runtime by default (GPU optional). Installation is straightforward (just `pip install insightface` plus `pip install onnxruntime` for CPU-only or `onnxruntime-gpu` for CUDA [4]). InsightFace runs **very fast on CPU** – benchmarks show ~0.015–0.021 seconds per image (50–66 FPS) on typical resolutions [5] – much faster than older models. The library is fully offline (models auto-download from repo) and MIT-licensed [6].

- **DeepFace (Serengil, MIT)** – A Python framework that *wraps* multiple face-recognition models (VGG-Face, FaceNet, OpenFace, Facebook DeepFace, DeepID, ArcFace, Dlib, etc.) [7]. It offers plug-and-play face verification, recognition and attribute analysis via a simple API. Installation is easy via `pip install deepface` [8] (Python 3.7+ [1]). However, DeepFace pulls in heavy dependencies (TensorFlow/Keras by default), making it **slower on CPU**. In practice, DeepFace's inference time ranges from *tens* to *hundreds* of milliseconds per image. For example, one study found DeepFace taking ~0.05–0.7 s per image (depending on resolution) [5], far slower than InsightFace. Accuracy depends on the chosen model: DeepFace wraps FaceNet (~99.63% LFW [9]) and ArcFace, but its default VGG-Face model is ~97.8% LFW [10]. In summary, DeepFace is **easy to install and use** [8], but on a CPU it lags InsightFace in speed. It remains offline and MIT-licensed [1].

- **Dlib (Boost-licensed)** – A C++ library with Python bindings for face detection and recognition. Dlib's **deep metric model** (a ResNet-29 variant) maps faces to 128D vectors. It achieves ~99.38% on LFW [11], comparable to SOTA. In Python, a popular wrapper is the Face_Recognition library (MIT) which uses dlib under the hood. **Installation is the hardest** of the three: it requires CMake, a C++ compiler and matching Python/Boost versions [12]. On Windows especially this often fails; on Linux/mac, `pip install dlib` may work if a prebuilt wheel exists. Once installed, inference runs on CPU in C++ code. Dlib's inference speed is moderate: reports indicate on a Jetson Nano it took ~0.5–0.8 s per image [13], so on a modern desktop CPU it might be a few hundred milliseconds per face. This is slower than InsightFace's 15–20 ms and in the same ballpark as DeepFace. However, dlib's code is well-optimized (multi-threaded) and its accuracy is very high. The dlib library is fully offline (all models are bundled) and uses a permissive Boost license [14].

- **Face_Recognition (MIT)** – A high-level Python library by Ageitgey. It's essentially a thin wrapper around dlib's face model. Face_Recognition is extremely easy to use (just `pip install face_recognition`) [15], but note that it *requires* dlib to be installed first. It reports the same 99.38% LFW accuracy (it *is* dlib's model) [16]. Installation is easier than raw dlib (it bundles wheels for many platforms), but on unsupported systems you still need to build dlib. Inference speed is similar to dlib's (hundreds of ms per face).

- **Other notable libraries:**

- **OpenFace (Apache-2.0)** – An older FaceNet-like toolkit. Accuracy is lower (~93.8% on LFW [17]) and installation (Torch-based) can be tricky.
- **facenet-pytorch (MIT)** – A modern PyTorch re-implementation of FaceNet. State-of-art accuracy (99.63% LFW [9]), runs on CPU/GPU, easy `pip install facenet-pytorch`.
- **OpenCV (Apache)** – Provides some classical face recognizers (Eigenfaces, LBPH) and DNN support. Accuracy is much lower than deep models. (Not state-of-art, so generally not preferred if SOTA accuracy is needed.)

## Accuracy in the Wild

Among these libraries, **InsightFace** (ArcFace models) currently yields the highest accuracy on public benchmarks [18]. In practice it generalizes very well to unconstrained "in-the-wild" photos. Dlib's model is also very accurate (99.38% LFW [11]). FaceNet (as wrapped by DeepFace or facenet-pytorch) likewise reports ~99.6% [9]. By contrast, older models (e.g. OpenFace, Facebook DeepFace 2014) are in the high-90s or ~97–98%. All of the above outperform human-baseline (~97.5% LFW [19]). In short, any of InsightFace, DeepFace (with its ArcFace/FaceNet models) or dlib will deliver **very high "in-the-wild" accuracy**, with InsightFace generally being state-of-art [18].

## Installation & Ease of Use

- **InsightFace:** Simply `pip install insightface` [2]. OnnxRuntime is required: run `pip install onnxruntime` (CPU) or `onnxruntime-gpu` for GPU support [4]. No compiler is needed. Models are fetched automatically or placed in `~/.insightface/models/`. Overall installation is straightforward.

- **DeepFace:** Simply `pip install deepface` [8]. It will pull in Keras/TensorFlow or PyTorch. This is easy (no compilation) but results in a large install footprint. Once installed, the interface is very user-friendly. It supports Python 3.7+ [1].

- **dlib (and Face_Recognition):** This can be **challenging**. Dlib's Python binding must match your compiler and Python (see Davis King's note on mismatches [12]). On Linux/mac, one can often do `pip install dlib` if build tools are available. On Windows, manual installation (via Conda or visual C++) is often required. In summary, **InsightFace/DeepFace install via pip with few hassles, whereas dlib may require manual build.**

## Inference Speed (CPU)

On a standard desktop CPU, **InsightFace** is fastest. In controlled tests it processed an image in **~15–21 ms** [5]. By contrast, **DeepFace** (with its default models) ran in **tens to hundreds of ms** (e.g. 52–733 ms as image size grows [5]). **Dlib/Face_Recognition** also runs on the order of a few hundred ms per face: one report on Jetson Nano was ~500–800 ms/image [13] (on a slower CPU); on a faster desktop one might see ~50–200 ms. In summary, *InsightFace can achieve real-time rates (≈30–60 FPS)* on a CPU, whereas DeepFace and dlib are slower (often single-digit FPS). Using GPU (with `onnxruntime-gpu` or TensorFlow) can improve DeepFace/dlib speeds, but the question requested CPU-focused performance.

## Summary

All the above libraries are **free, local, and have strong accuracy**. InsightFace (ArcFace/CosFace) currently leads in accuracy and speed [5] [18]. DeepFace offers ease-of-use and flexibility (multiple models) at the cost of heavier inference. Dlib's model is very accurate, but its Python install is harder and inference slower. In practice, for a "best overall" pick: **InsightFace** (MIT-licensed) is recommended for top accuracy and fastest CPU inference. DeepFace (MIT) is great for rapid prototyping, and face_recognition/dlib (Boost-licensed) is solid if installation is solved. All run offline and respect privacy, meeting the specified criteria.

**Sources:** Official docs and benchmarks (InsightFace [6] [5], DeepFace PyPI and tutorials [1] [8], dlib author notes [12], and published comparisons [5] [11] [3] [13]).

---


[1] [7] [8] deepface · PyPI
https://pypi.org/project/deepface/

[2] [4] insightface · PyPI
https://pypi.org/project/insightface/

[3] [9] [10] [11] [17] [18] [19] Master Facial Recognition with DeepFace in Python
https://viso.ai/computer-vision/deepface/

[5] Impact of Image Resolution on Age Estimation with DeepFace and InsightFace
https://arxiv.org/pdf/2511.14689

[6] GitHub - deepinsight/insightface: State-of-the-art 2D and 3D Face Analysis Project
https://github.com/deepinsight/insightface

[12] An interview with Davis King, creator of the dlib toolkit - PyImageSearch
https://pyimagesearch.com/2017/03/13/an-interview-with-davis-king-creator-of-the-dlib-toolkit/

[13] Face Detection/Recognition In Python that gets 5FPS or more? - Jetson Nano - NVIDIA Developer Forums
https://forums.developer.nvidia.com/t/face-detection-recognition-in-python-that-gets-5fps-or-more/253256

[14] dlib C++ Library - License
https://dlib.net/license.html

[15] [16] face-recognition · PyPI
https://pypi.org/project/face-recognition/