

基于 MNN 和 MTCNN 的人脸检测

目前安卓端调用 mnn 的 demo 是将应用层（如 loadmodel, create session）逻辑写入了 java 层，对于不熟悉 java 的同学可能使用不是很方便，我这边提供一个将应用层写到 c++ 的 demo 演示。

调用的结构是这样的：

1. 最上层是 java 层，功能为传图给 c++ 层，显示计算后图像，不涉及 mnn 的逻辑
2. 中间层：jni 层，java 和 c++ 的接口层，只负责传递数据
3. 底层：c++ 层，负责图像的计算功能，例如本实例中的人脸检测功能

如何使用？

1. 本 demo 底层依赖 opencv 和 mnn 实现
 - a) mnn 已经提供了编译好的 demo 版本，
 - b) opencv 请到官网下载 release 版本，地址为 <https://opencv.org/releases/>
 - i. 下载完成后需要修改 CMakeLists.txt 中第 10 行代码指定 opencv 路径
 - ii. 若不想修改 CMakeLists.txt，可将 OpenCV-android-sdk 文件夹拷贝至 CMakeLists.txt 相同的文件夹下。
2. 将 models 中的 3 个模型文件拷贝至手机上，手机路径位于“内部存储\1”
 - a) 如果在手机端找不到“内部存储\1 文件夹”，请断点调试 native-lib.cpp 中第 44 行查看完整路径

C++ 代码层次划分：

- b) 本文聚焦于 c++ 实现部分，阅读此实现的起点为 native-lib.cpp 中的“deal(imgData, testfilePath);”，此接口为 c++ 实现的入口，它的输入为

当前的图像帧， 模型存储的路径， 功能为实时检测人脸框， 并将人脸框画在 imgData 上， 返回 imgData;

- c) Deal 函数内部调用 imgProcess 的 detFace 接口， 实现人脸检测功能
- d) detFace 内部调用 MTCNN 类， MTCNN 类对 mtcnn 人脸检测算法进行封装， MTCNN 类的实现位于 mtcnn.cpp
- e) MTCNN 在底层调用了 Inference_engine 类实现前传功能
- f) Inference_engine 对 mnn 库的外部接口进行封装， 统一为 load 和 infer 两个接口， 并且在这一层封装了 mnn 的逻辑功能