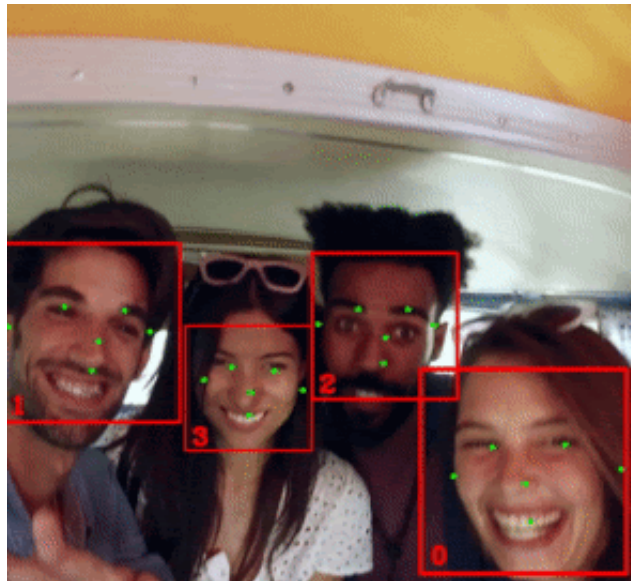


AI学习笔记--MediaPipe--软件环境篇

• 基本说明

MediaPipe 是 Google 开源的一套关于 ML 的相关框架。目前提供的是人脸、手掌等识别算法历程。



官网的学习文档地址：

<https://mediapipe.readthedocs.io>

源码的 Git 地址：

<https://github.com/google/mediapipe>

首先梳理一下总的逻辑，编译需要使用 bazel 工具，Mac下可以用 brew 工具 install 相应的工具链。例如终端输入命令：

```
→ ~ brew install bazel
Updating Homebrew...
==> Auto-updated Homebrew!
Updated 1 tap (homebrew/core).
==> New Formulae
aws-cdk                                llvm@8
==> Updated Formulae
ffmpeg ✓                               dosbox-x                               kops                                   paket                                  terragrunt
aws-okta                               exploitdb                               kubecfg                               pipx                                  tomcat@8
babel                                  git-annex                              llvm                                   rabbitmq                               webpack
bit                                    gmic                                   minio-mc                              range-v3
conan                                  goreleaser                             mono                                   rc
crystal                               jfrog-cli-go                           mosquitto                             reop
dbmate                                knot                                    nim                                    src
==> Deleted Formulae
llvm@3.9                               llvm@5
```

```
Warning: bazel 0.29.1 is already installed and up-to-date
To reinstall 0.29.1, run `brew reinstall bazel`
```

并且，实际配置中，发现 mediapipe 需要的软件配置情况如下：

- Bazel 版本在 0.24 以上，脚本中有对 bazel 版本的判断。
- Android SDK release 28.0.3 and above.
- Android NDK r17c and above.
- Android OS 版本需要在 5 以上（Android 5 可能需要安装 Google 全家桶，高版本无需安装，在源码中已经集成了相关的 TensorFlow 模块）

最后，可以使用 bazel 进行 build 工程，也可以导入到 Android Studio 进行 build，个人建议，AS 用于看代码和 写代码。build 还得看命令行，比如现在需要 build 一个 face 的历程，可以在命令终端中输入：

```
bazel build -c opt --config=android_arm
mediapipe/examples/android/src/java/com/google/mediapipe/apps/facedetection
cpu
```

这个例子中的一些配置都可以在 build 中找到相对应的说明。具体搜关键词自己去尝试。

• 模块说明

主要的代码区块分为以下几个部分

- calculators 用于计算和处理的模块
- Docs 文档说明
- example 历程
- framework 核心代码模块
- GPU GPU 加速模块
- Graphs 图形处理运算模块
- JAVA 与 Android 相关的模块
- models 模型库集合

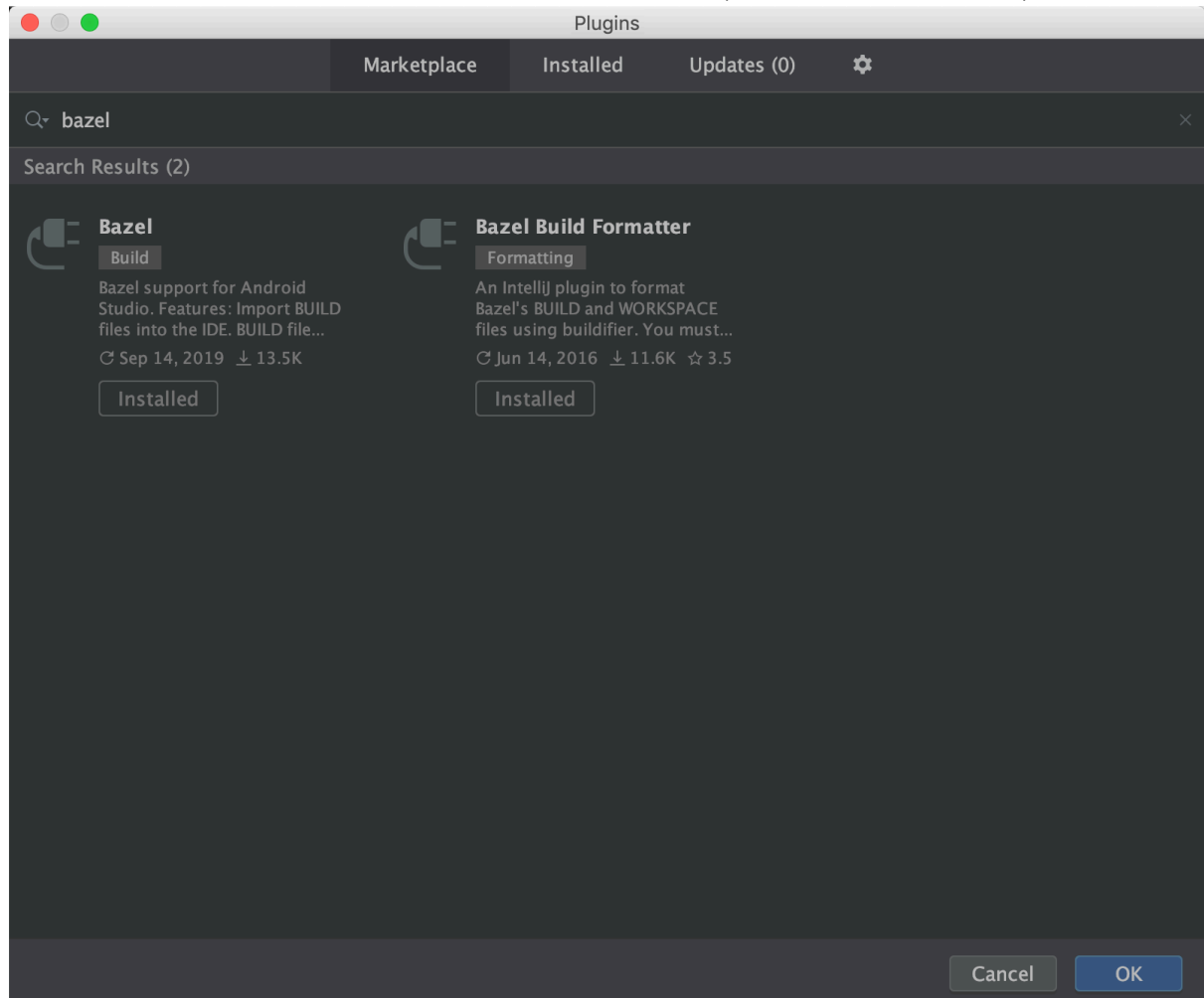
在第三方的模块中，我们找到了相应的模块信息：

- FFMpeg 用于视频编解码模块
- libYuv 用于 YUV 数据格式处理
- OpenCV 用于图形处理模块
- Protobuf Google 序列化
- TensorFlow ML 基本库
- Camera Camera 相关模块

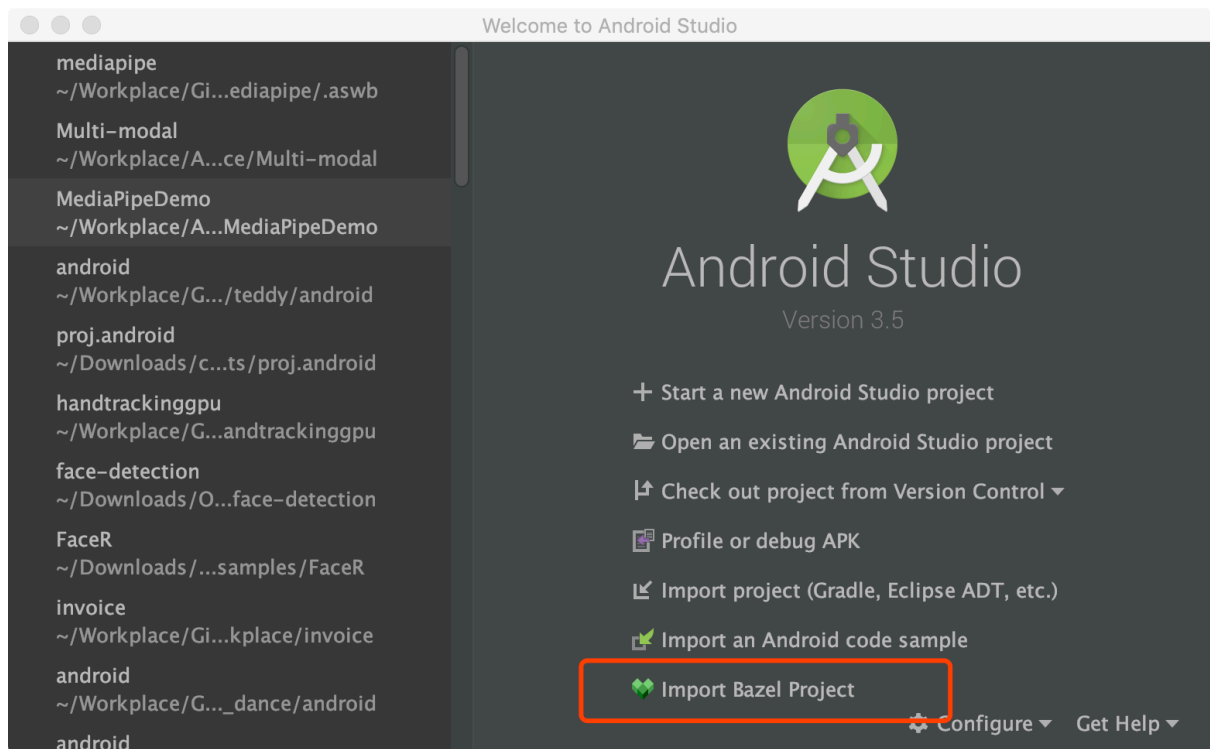
以上便是大概包含模块的说明情况。

• AS build

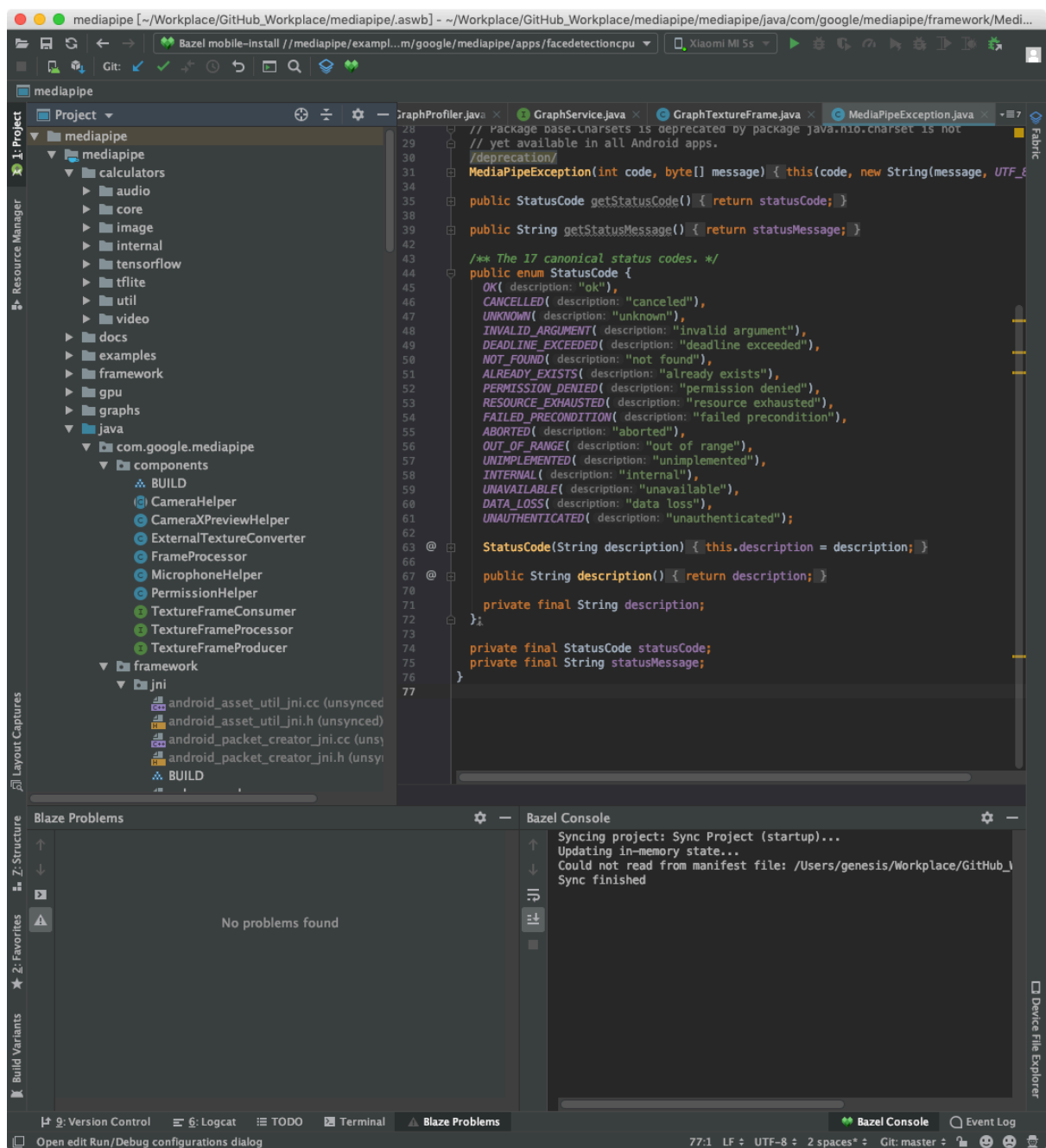
在 Android studio 3.5 版本中添加插件 Bazel 工具，在打开的页面中添加，如下：



添加完成后，重启会发现导入工程模式的方式多了一个选择。



在这里，我们导入从 Git clone 的代码：



自此，可以看到代码的跳转，整体模块的相关内容，不推荐使用 AS run 相关的代码，效率太低，太慢，还是在这里改代码，并且使用命令行去执行 build。如果需要配置 AS build 工程模块，需要按照官方文档第 9 和 10 的配置条件：

1. Connect an Android device to the workstation.
2. Select Run... | Edit Configurations....
 - * Select Templates | Bazel Command.
 - * Enter Target Expression:
//mediapipe/examples/android/src/java/com/google/mediapipe/apps/facedetectioncpu
 - * Enter Bazel command: mobile-install.
 - * Enter Bazel flags: -c opt --config=android_arm64.

- * Press the [+] button to add the new configuration.
- * Select Run to run the example app on the connected Android device.