

# Mediapipe

---

## 1. Introduction to mediapipe

MediaPipe is an open-source framework for developing machine learning applications for data stream processing developed by Google. It is a graph-based data processing pipeline for building applications that use a variety of data sources, such as video, audio, sensor data, and any time-series data. mediaPipe is cross-platform, running on embedded platforms (Raspberry Pi, etc.), mobile devices (iOS and Android), workstations, and servers, and supports mobile GPU acceleration. MediaPipe provides a cross-platform, customisable ML solution for real-time and streaming media, with its core framework implemented in C++ and language support for Java and Objective C. MediaPipe's key concepts include Packet, Stream, Calculator, Graph, and Stream. The main concepts of MediaPipe include Packet, Stream, Calculator, Graph and Subgraph.

MediaPipe features:

- End-to-End Acceleration: Built-in fast ML reasoning and processing accelerates even on common hardware;
- Build Once, Deploy Anywhere: Unified solution for Android, iOS, desktop/cloud, web and IoT;
- Ready-to-Use Solution: A cutting-edge ML solution that demonstrates the full functionality of the framework;
- Free and open source: framework and solution under Apache 2.0, fully extensible and customisable.

## 2. Use

### 2.1 Run the programme

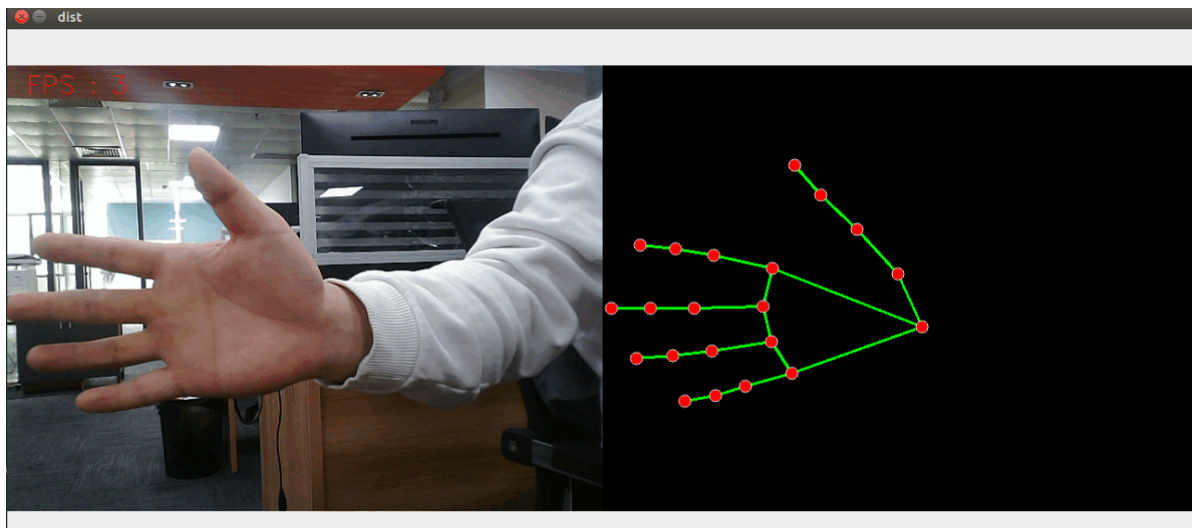
source code path reference.

```
/home/pi/cartographer_ws2/src/yahboom_mediapipe/yahboom_mediapipe
```

terminal input.

```
#手部检测#Hand detection
ros2 run yahboom_mediapipe 01_HandDetector
# 姿态检测# Attitude detection
ros2 run yahboom_mediapipe 02_PoseDetector
# 整体检测# Overall detection
ros2 run yahboom_mediapipe 03_Holistic
# 面部检测# Facial detection
ros2 run yahboom_mediapipe 04_FaceMesh
# 人脸识别# Face Recognition
ros2 run yahboom_mediapipe 05_FaceEyeDetection
```

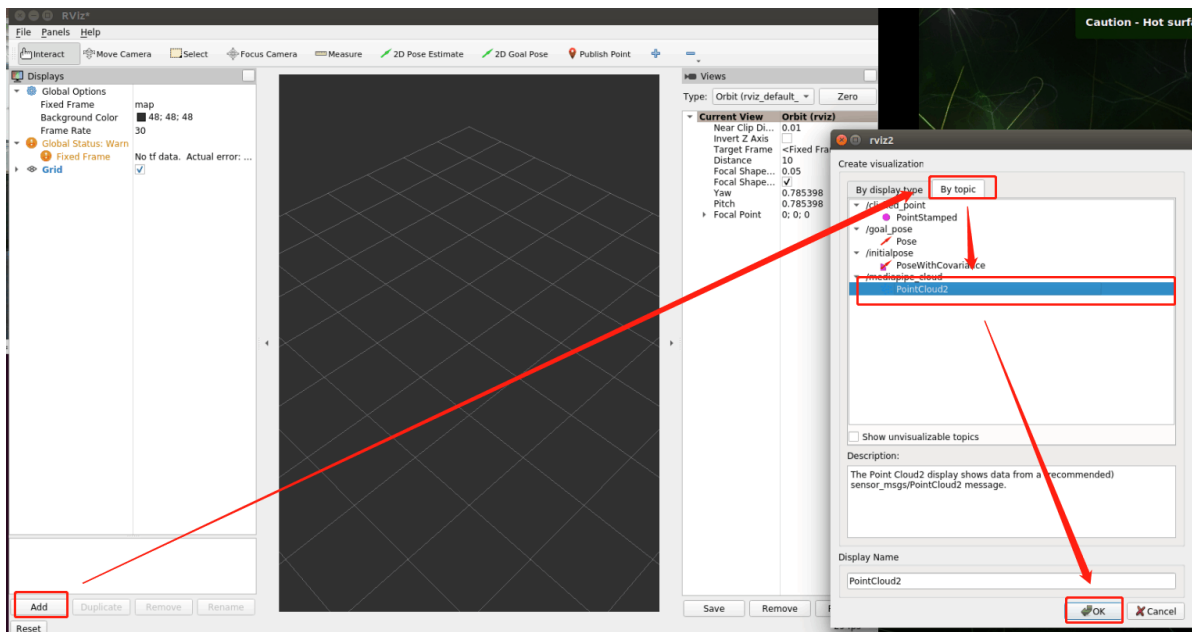
Taking hand detection as an example, the screenshot of the run is as follows

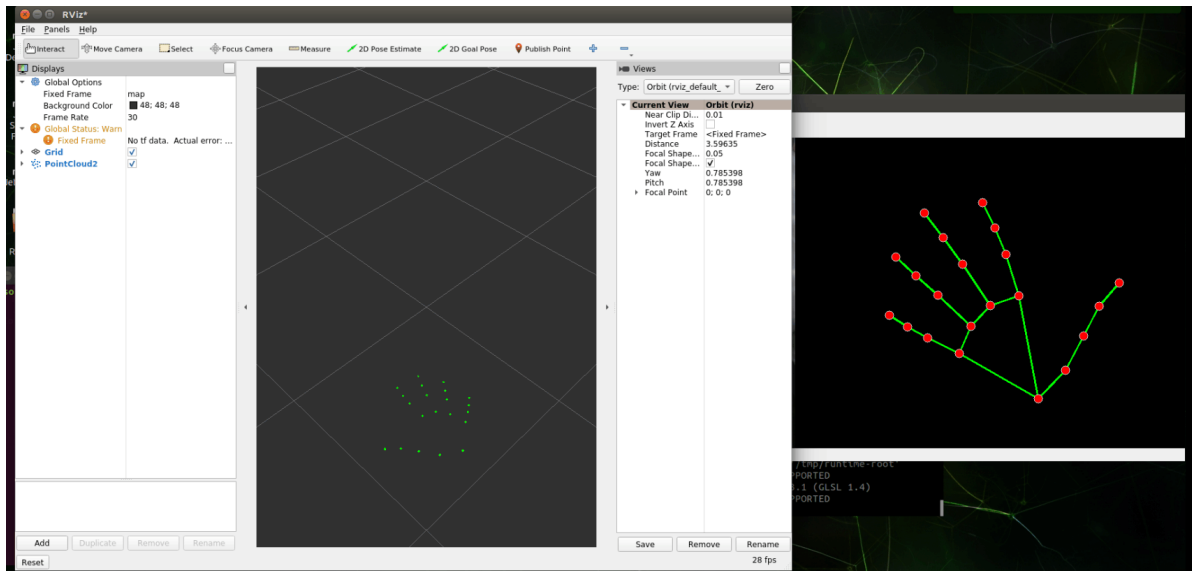


Alternatively, to view point cloud data, docker terminal input.

```
#运行点云发布程序# Run the point cloud publisher
ros2 run yahboom_point pub_point
#开启rviz查看点云# Turn on rviz to view point clouds
rviz2
```

Follow the steps below to add a point cloud topic to rviz.

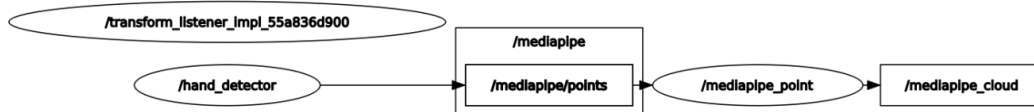




Above is a point cloud view of the program running HandDetector aka 01\_HandDetector. Point cloud viewing is only supported for the 01-05 demos.

The node topic communication graphs can be viewed via rqt\_graph, the

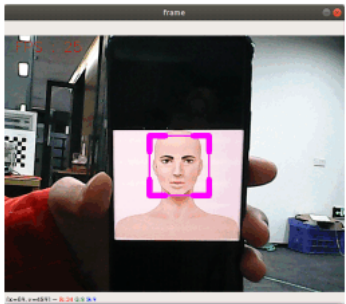
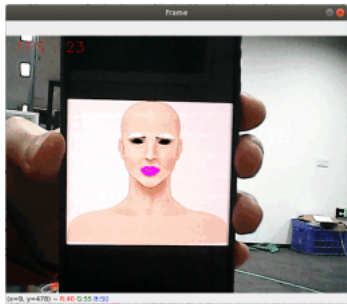
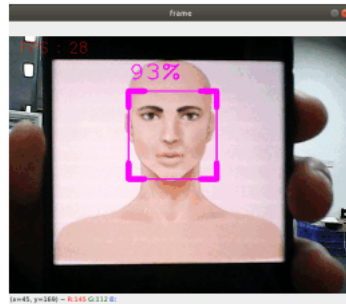
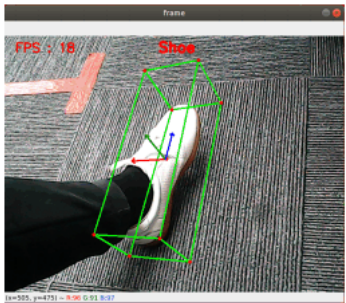
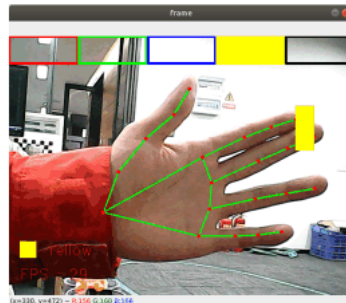
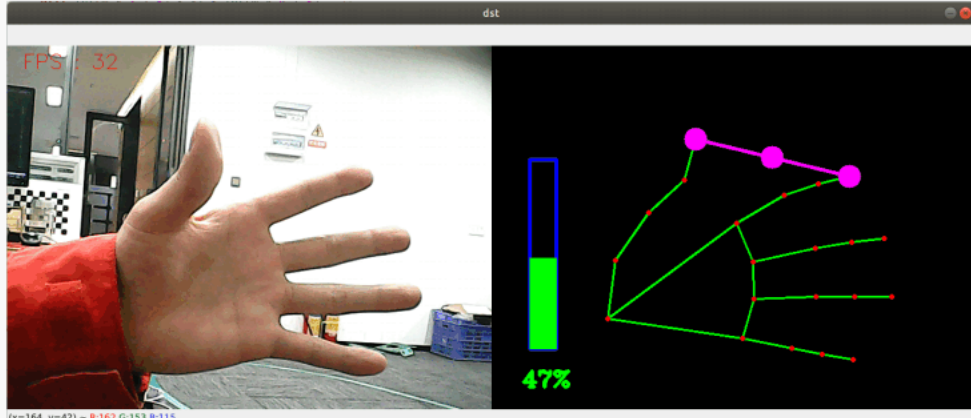
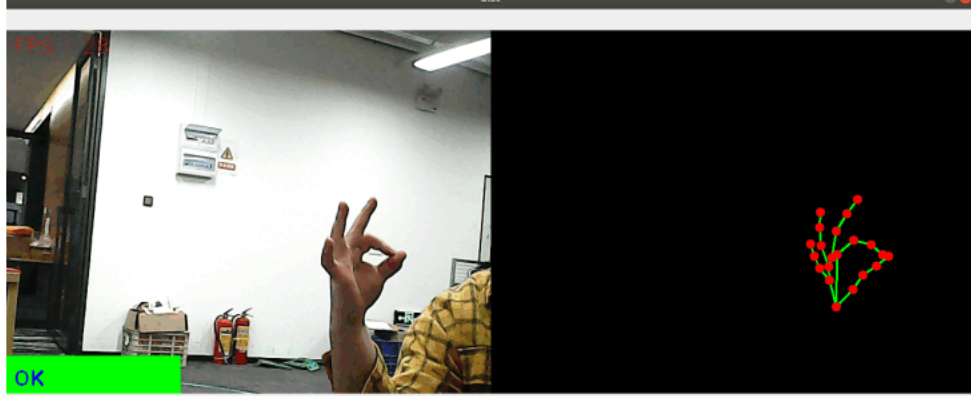
```
ros2 run rqt_graph rqt_graph
```



## 2.2 Other fun ways to play

In the Mechanical Dog terminal enter.

```
cd /home/pi/cartographer_ws2/src/yahboom_mediapipe/yahboom_mediapipe
#人脸特效#Face Effect
python3 06_FaceLandmarks.py
#人脸检测#Face Detection
python3 07_FaceDetection.py
#三维物体识别# 3D object recognition
python3 08_Objectron.py
#画笔# Brushes
python3 09_virtualPaint.py
#手指控制#Fingerprint control
python3 10_HandCtrl.py
#手势识别# Hand gesture recognition
python3 11_GestureRecognition.py
```

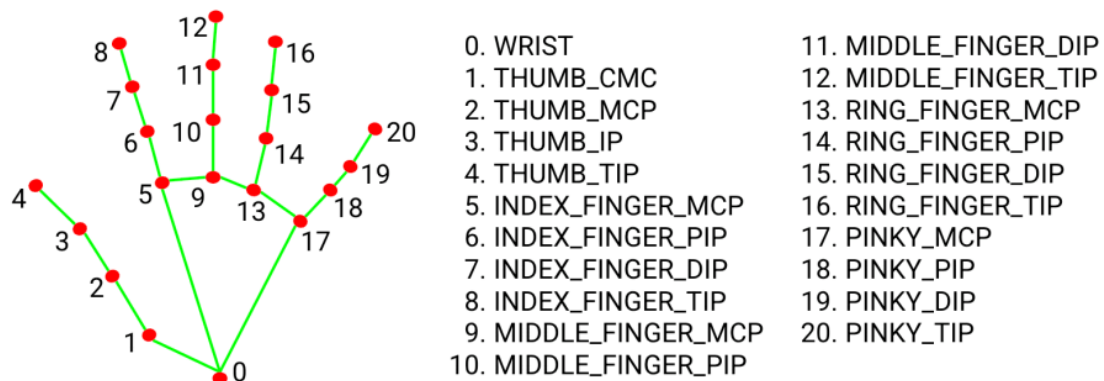
<p><b>05、face recognition</b></p> 	<p><b>06、face effect</b></p> 	<p><b>07、face detection</b></p> 
<p><b>08、3D object recognition</b></p> 		<p><b>09、brushes</b></p> 
<p><b>10、finger control</b></p>		
<p><b>11、gesture recognition</b></p>		

### 3. Mediapipe Hands

MediaPipe Hands is a high fidelity hand and finger tracking solution. It uses machine learning (ML) to infer the 3D coordinates of 21 hands from a single frame.

After performing palm detection on the entire image, the coordinates of the 21 3D hand joints within the detected hand regions are precisely keypoint-located by regression based on the hand marking model, i.e., direct coordinate prediction. The model learns a consistent internal representation of the hand pose and is robust even to partially visible hands and self-occlusion.

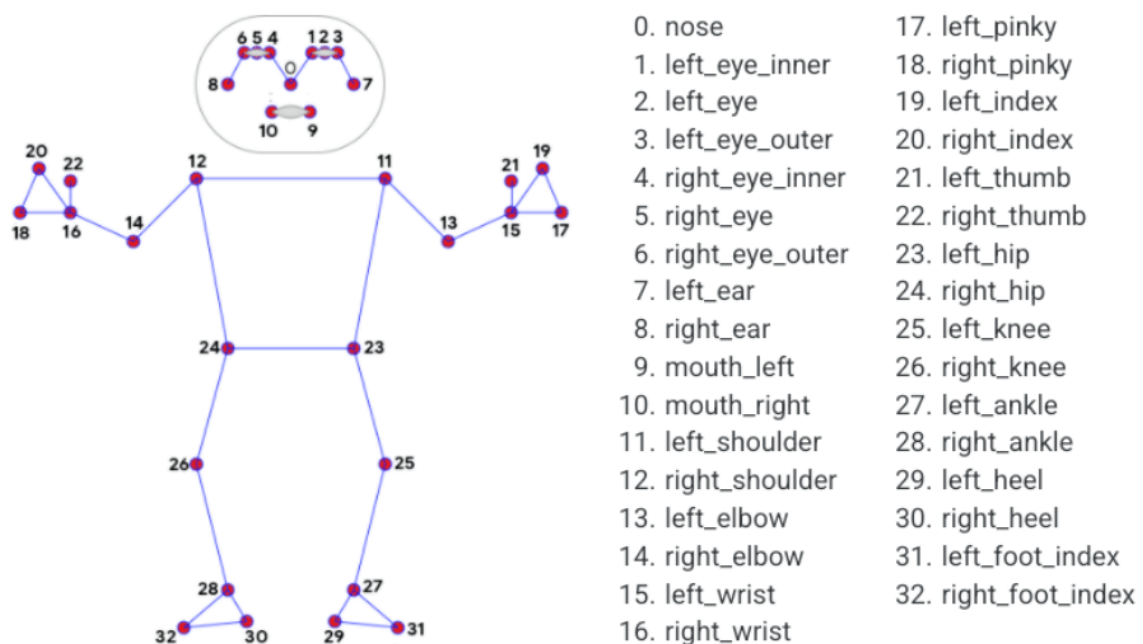
In order to obtain face-toface real data, approximately 30K real-world images were manually annotated using 21 3D coordinates, as shown below (Z-values were obtained from the image depth map, if Z-values were available for each of the corresponding coordinates). To better cover possible hand poses and to provide additional supervision of the nature of hand bases, high-quality synthetic hand models in various backgrounds were also drawn and mapped to the corresponding 3D coordinates.



## 4. Mediapipe Pose

MediaPipe Pose, an ML solution for high-fidelity body pose tracking, utilises the BlazePose study to infer 33 3D coordinates and full-life background segmentation masks from RGB video frames, which also powers the ML Kit pose detection API.

The landmark model in the MediaPipe pose predicts the locations of the 33 pose coordinates (see figure below).



## 5. dlib

The counterpart is the human face effect.

DLIB is a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real-world problems. It is widely used by industry and academia in areas such as robotics, embedded devices, mobile phones, and large, high-performance computing environments. The dlib library uses 68 points to mark important parts of the human face, such as 18-22 points for the right eyebrow and 51-68 points for the mouth. Faces are

detected using the `get_frontal_face_detector` module of the `dlib` library, and the `shape_predictor_68_face_landmarks.dat` feature data is used to predict the values of face features.