DLib 配置与 pip 安装

【操作步骤】

1. apt 安装一些基础软件
2. 配置 DLib 人脸探测，尝试启动
3. 手工安装 face_recognition
   a) 查看 HA 中 dlib_face_detect.py 源程序
   b) pip 安装的程序包选择次序
   c) pip 安装中的 USERBASE
   d) pip 安装中的本地 wheel 包

【参考】

- Dlib 人脸探测配置说明文档
  https://www.home-assistant.io/components/image_processing.dlib_face_detect/
- apt-get 安装命令
  ```
  sudo apt-get install libatlas-base-dev cmake
  ```
- Dlib 人脸探测配置（example_9_1_1.yaml）
  ```
  image_processing:
    - platform: dlib_face_detect
      scan_interval: 1000000
      source:
        - entity_id: camera.cam7
  ```
- 手工安装 face_recognition
  ```
  export PYTHONUSERBASE=/home/pi/.homeassistant/deps
  pip3 install face_recognition==1.0.0 --upgrade --user
  ```

本地 DLib 人脸探测

【操作步骤】

1. 使用 vlc 配置一个抓屏摄像头
2. Dlib 人脸探测配置解说
3. 使用 packages 方式保存配置文件
4. 配置 whitelist_external_dirs
5. 演示人脸探测，观察 CPU 状态

【参考】

- VLC http mjpeg 输出配置

  :sout=#transcode{vcodec=MJPG,vb=800,scale=自动,acodec=none,scodec=none}:standard{access=http{mime=multipart/x-mixed-replace;boundary=7b3cc56e5f51db803f790dad720ed50a},mux=mpjpeg,dst=:8888/} :no-sout-all :sout-keep

- DLib 人脸探测配置说明文档

  https://www.home-assistant.io/components/image_processing.dlib_face_detect/

- DLib 人脸探测配置（example_9_2_1.yaml）

```
# example_9_2_1.yaml
image_processing:
  - platform: dlib_face_detect
    scan_interval: 1000000
    source:
      - entity_id: camera.cam_input
        name: face
script:
  dlib_face_detect:
    alias: 人脸探测并保存图片
    sequence:
      - service: image_processing.scan
        data:
          entity_id: image_processing.face
      - service: camera.snapshot
        data:
          entity_id: camera.cam_input
          filename: '/home/pi/Pictures/face.jpg'
camera:
  - platform: local_file
    name: image_to_be_processed
    file_path: /home/pi/Pictures/face.jpg
```

本地 DLib 人脸识别

【操作步骤】

1. 使用 vlc 配置一个抓屏摄像头（同上一个视频）
2. 上传人物标准照片
3. DLib 人脸识别配置解说
4. 使用 packages 方式配置
5. 演示人脸识别，观察 CPU 状态

【参考】

- VLC http mjpeg 输出配置（同上一篇）

  :sout=#transcode{vcodec=MJPG,vb=800,scale=自

  动,acodec=none,scodec=none}:standard{access=http{mime=multipart/x-mixed-replace;

  boundary=7b3cc56e5f51db803f790dad720ed50a},mux=mpjpeg,dst=:8888/} :no-sout-all :sout-keep

- DLib 人脸识别配置说明文档

  https://www.home-assistant.io/components/image_processing.dlib_face_identify/

- DLib 人脸识别配置（example_9_3_1.yaml）

```
# example_9_3_1.yaml
image_processing:
  - platform: dlib_face_identify
    scan_interval: 1000000
    source:
      - entity_id: camera.cam_input
        name: face_id
    faces:
      Trump: /home/pi/Pictures/trump.jpg
      Obama: /home/pi/Pictures/obama.jpg
      Clinton: /home/pi/Pictures/clinton.jpg
script:
  dlib_face_identify:
    alias: 人脸识别
    sequence:
      - service: image_processing.scan
        data:
          entity_id: image_processing.face_id
      # - service: camera.snapshot
        # data:
          # entity_id: camera.cam_input
          # filename: '/home/pi/Pictures/face.jpg'
# camera:
  # - platform: local_file
    # name: image_to_be_processed
    # file_path: /home/pi/Pictures/face.jpg

automation:
  - alias: Clinton coming
    trigger:
      platform: event
      event_type: image_processing.detect_face
      event_data:
        entity_id: image_processing.face_id
        name: 'Clinton'
    action:
      service: persistent_notification.create
      data:
        title: '发现认识的人'
        message: '克林顿出现在屏幕上'
  - alias: Obama coming
    trigger:
      platform: event
```

```
    event_type: image_processing.detect_face
    event_data:
      entity_id: image_processing.face_id
      name: 'Obama'
  action:
    service: persistent_notification.create
    data:
      title: '发现认识的人'
      message: '奥巴马出现在屏幕上'
- alias: Trump coming
  trigger:
    platform: event
    event_type: image_processing.detect_face
    event_data:
      entity_id: image_processing.face_id
      name: 'Trump'
  action:
    service: persistent_notification.create
    data:
      title: '发现认识的人'
      message: '川普出现在屏幕上'
```

微软人脸特征检测

【操作步骤】

1. 申请免费的微软认知/人脸服务 key
2. 微软人脸特征检测配置解说
3. 使用 vlc 配置一个抓屏摄像头（同上一个视频）
4. 效果实验

【参考】

- 微软人脸云服务
  https://azure.microsoft.com/zh-cn/services/cognitive-services/face/
- HA 中微软人脸探测配置说明文档
  https://www.home-assistant.io/components/image_processing.microsoft_face_detect/
- 微软人脸探测配置样例（example_9_4_1.yaml）

```
# example_9_4_1.yaml
microsoft_face:
  api_key: xxxxxxxxxxxxxx
  azure_region: westcentralus

image_processing:
  - platform: microsoft_face_detect
    scan_interval: 1000000
    source:
      - entity_id: camera.cam_input
        name: ms_face_feature
    attributes:
      - age
      - gender
      - glasses

script:
  ms_face_detect:
    alias: 微软人脸特征识别
    sequence:
      - service: image_processing.scan
        data:
          entity_id: image_processing.ms_face_feature

automation:
  - alias: Somebody appearing
    trigger:
      platform: event
      event_type: image_processing.detect_face
      event_data:
        entity_id: image_processing.ms_face_feature
    action:
      service: tts.google_say
      data_template:
        message: >
          {% if trigger.event.data.glasses=="ReadingGlasses" %}
            {% set message = '眼镜' %}
              {% if trigger.event.data.gender=="male" %}
                {% set message=message+'男' %}
              {% else %}
                {% set message=message+'女' %}
              {% endif %}
          {% else %}
            {% if trigger.event.data.gender=="male" %}
              {% set message='男人' %}
            {% else %}
              {% set message='女人' %}
            {% endif %}
          {% endif %}
          发现一个{{ message }}，大概{{ trigger.event.data.age|int }}岁。
```

- VLC http mjpeg 输出配置（同上一篇）

:sout=#transcode{vcodec=MJPG,vb=800,scale=自
动,acodec=none,scodec=none}:standard{access=http{mime=multipart/x-mixed-replace;
boundary=7b3cc56e5f51db803f790dad720ed50a},mux=mpjpeg,dst=:8888/} :no-sout-all :sout-keep

微软人脸识别与认证

【操作步骤】

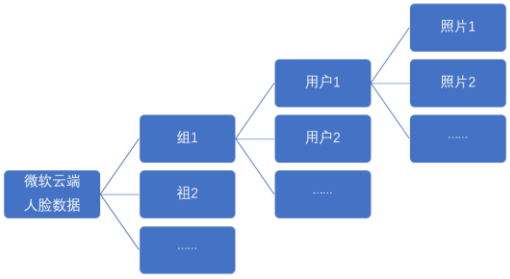
1. 准备标准照待用
2. 在微软人脸云服务中配置组、用户、标准照片
3. 微软人脸识别配置解说
4. 使用 vlc 配置一个抓屏摄像头（同上一个视频）
5. 效果实验

【参考】

- 微软人脸云服务 API 说明

  https://westcentralus.dev.cognitive.microsoft.com/docs/services/563879b61984550e40cbbe8d/operations/563879b61984550f30395236

- 微软人脸云端数据结构



- 相关 API 调用命令
  - 创建组

```
curl -X PUT "https://westcentralus.api.cognitive.microsoft.com/face/v1.0/persongroups/presidents" \
 -H "Content-Type: application/json" \
 -H "Ocp-Apim-Subscription-Key: 4211c17af9b14ff581a41492bd7b069b" \
 --data-ascii "{'name': 'PresidentsInUS', 'userData': 'Presidents of United States'}"
```

  - 创建用户

```
curl -X POST "https://westcentralus.api.cognitive.microsoft.com/face/v1.0/persongroups/presidents/persons" \
 -H "Content-Type: application/json" \
 -H "Ocp-Apim-Subscription-Key: 4211c17af9b14ff581a41492bd7b069b" \
 --data-ascii "{'name': 'Clinton', 'userData': 'Bill Clinton'}"
```

  - 上传用户照片

```
curl -X POST \
"https://westcentralus.api.cognitive.microsoft.com/face/v1.0/persongroups/presidents/persons/bb6d222a-8956-40ee-8a78-e00ccd3c6503/persistedFaces" \
-H "Ocp-Apim-Subscription-Key: 4211c17af9b14ff581a41492bd7b069b" \
-H "Content-Type: application/octet-stream" \
--data-binary "@/home/pi/Pictures/clinton.jpg"
```

  - 训练组

```
curl -X POST "https://westcentralus.api.cognitive.microsoft.com/face/v1.0/persongroups/presidents/train" \
 -H "Ocp-Apim-Subscription-Key: 4211c17af9b14ff581a41492bd7b069b" \
 --data-ascii ""
```

- HA 中微软人脸识别配置说明文档

  https://www.home-assistant.io/components/image_processing.microsoft_face_identify/

- 微软人脸识别与认证配置样例（example_9_5_1.yaml）

```
# example_9_5_1.yaml
microsoft_face:
  api_key: 4211c17af9b14ff581a41492bd7b069b
  azure_region: westcentralus

image_processing:
  - platform: microsoft_face_identify
    scan_interval: 1000000
    group: presidents
    confidence: 10
    source:
      - entity_id: camera.cam_input
        name: ms_face_identify

script:
  ms_face_identify_script:
    alias: 微软人脸识别
    sequence:
      - service: image_processing.scan
        data:
          entity_id: image_processing.ms_face_identify

automation:
  - alias: Clinton Identify
    trigger:
      platform: event
      event_type: image_processing.detect_face
      event_data:
        entity_id: image_processing.ms_face_identify
        name: 'Clinton'
    action:
      service: persistent_notification.create
      data_template:
        title: '人脸认证'
        message: '克林顿出现在摄像头中，可信度{{ trigger.event.data.confidence }}%'
```

- VLC http mjpeg 输出配置（同上一篇）

```
:sout=#transcode{vcodec=MJPG,vb=800,scale=自
动,acodec=none,scodec=none}:standard{access=http{mime=multipart/x-mixed-replace;
boundary=7b3cc56e5f51db803f790dad720ed50a},mux=mpjpeg,dst=:8888/} :no-sout-all :sout-keep
```

Facebox-在 docker 中运行人脸识别

【操作步骤】

1. 安装 docker
2. 获得 MB_KEY
3. 启动 machinebox/facebox
4. 在 facebox 的 web 界面中传入标准照并进行识别操作
5. 在 HA 中配置与使用 facebox

【参考】

- Linux 下 docker 的安装

  ```
  curl -fsSL get.docker.com -o get-docker.sh && sh get-docker.sh
  ```

- Windows 下 docker 的安装
  https://docs.docker.com/docker-for-windows/install/
- facebox 网站
  https://machinebox.io/docs/facebox
- MB_KEY 获得
  https://machinebox.io/account
- docker 运行 facebox

  ```
  docker run -d --name=facebox --restart=always -p 9999:8080 -e "MB_KEY=$MB_KEY"    machinebox/facebox
  ```

- 测试照片 URL

  https://a57.foxnews.com/static.foxnews.com/foxnews.com/content/uploads/2018/09/1862/1048/billhillaryclintonreut.jpg?ve=1&tl=1

- HA 中 facebox 配置样例（example_9_6_1.yaml）

```
# example_9_6_1.yaml
image_processing:
  - platform: facebox
    scan_interval: 1000000
    ip_address: 192.168.3.156
    port: 9999
    confidence: 10
    source:
      - entity_id: camera.cam_input
        name: facebox_identify

script:
  ms_face_identify_script:
    alias: facebox 人脸识别
    sequence:
      - service: image_processing.scan
        data:
          entity_id: image_processing.facebox_identify

automation:
  - alias: Clinton Identify
    trigger:
      platform: event
      event_type: image_processing.detect_face
      event_data:
        entity_id: image_processing.facebox_identify
        name: 'Clinton'
    action:
      service: persistent_notification.create
      data_template:
        title: '人脸认证'
        message: '克林顿出现在摄像头中，可信度{{ trigger.event.data.confidence }}%'
```