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
   1. 查看HA中dlib\_face\_detect.py源程序
   2. pip安装的程序包选择次序
   3. pip安装中的USERBASE
   4. 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

注：最新版的vlc，可能会将8888仅监听到ipv6上；可以修改为：dst=x.x.x.x:8888/

其中，x.x.x.x是你的pc的ip地址

* 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

entity\_id: "all"

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】

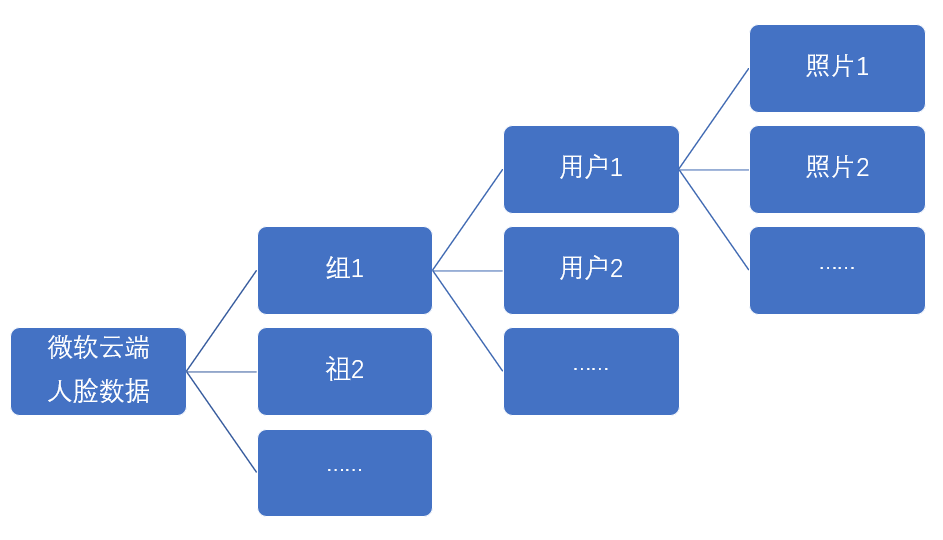
1. 准备标准照待用
2. 在微软人脸云服务中配置组、用户、标准照片
3. 微软人脸识别配置解说
4. 使用vlc配置一个抓屏摄像头（同上一个视频）
5. 效果实验

【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 }}%'