



数字识别操作文档

文件状态： [] 正在修改 [] 正在发布	当前版本：	V1.0
	作者：	张振丹
	完成日期：	2019.1.12
	审核：	Adolph
	完成日期：	2019.1.16

版本历史

版本号	作者	修改日期	修改说明	审核	备注
V1.0	张振丹	2019.1.12	初始版本	Adolph	



目录

前言:	3
一、准备工作:	3
二、接线方式:	3
三、拷贝程序:	4
四、进入指定路径:	5
五、运行程序:	6
六、识别过程:	7
七、退出程序:	8
八、附件:	8



前言：

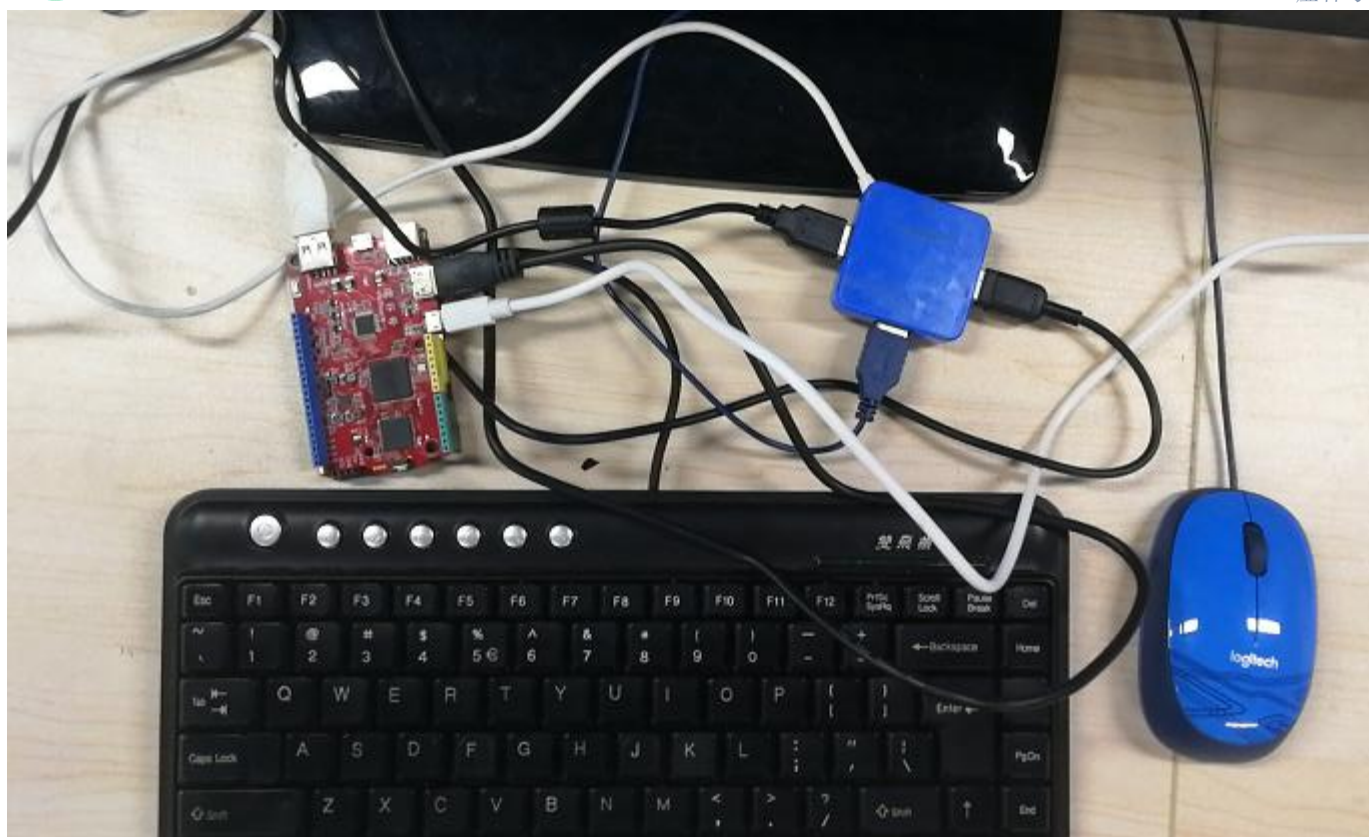
本文档介绍了如何在虚谷板上运行数字识别程序

一、准备工作：

- 虚谷板
- 显示器
- 鼠标
- 键盘
- HDMI 线
- 摄像头
- USB HUB
- 数字识别程序
- 良好的 Micro USB OTG 线

二、接线方式：

将虚谷板连接上 USB HUB，再将鼠标、键盘、显示器、摄像头接在 HUB 上，最后将虚谷板上电源，如下图所示：



三、拷贝程序：

将数字识别程序拷贝到一个目录下，例如我将程序拷贝到 /home/scope/Desktop/目录下，如下图所示：



四、进入指定路径：

进入到刚才存放文件的路径，进入的方法如下：

按 Ctrl+Alt+t 组合按键打开终端，在终端上输入 `cd /home/scope/Desktop/`，按下回车，如果没有提示就说明我们已经成功进入到该路径下了，如下图所示：



```
Terminal 终端 - scope@localhost: ~/Desktop
文件(F) 编辑(E) 视图(V) 终端(T) 标签(A) 帮助(H)
scope@localhost:~$ cd /home/scope/Desktop/
scope@localhost:~/Desktop$
```

五、运行程序：

运行数字识别程序：

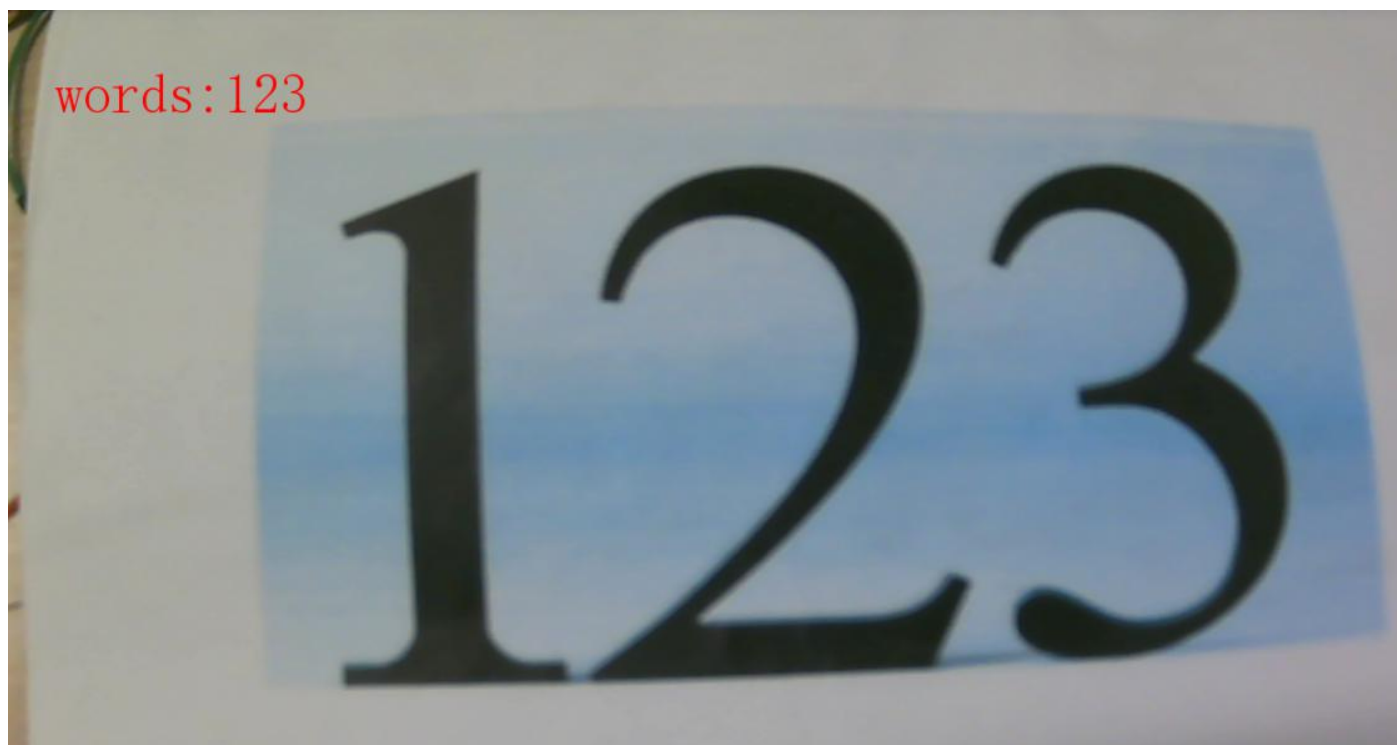
在终端上输入 `python3 words.py`, 按下回车, 会在屏幕上看到如下界面就说明程序已经运行成功（注意：初次运行可能有点慢，需要等几秒钟）：

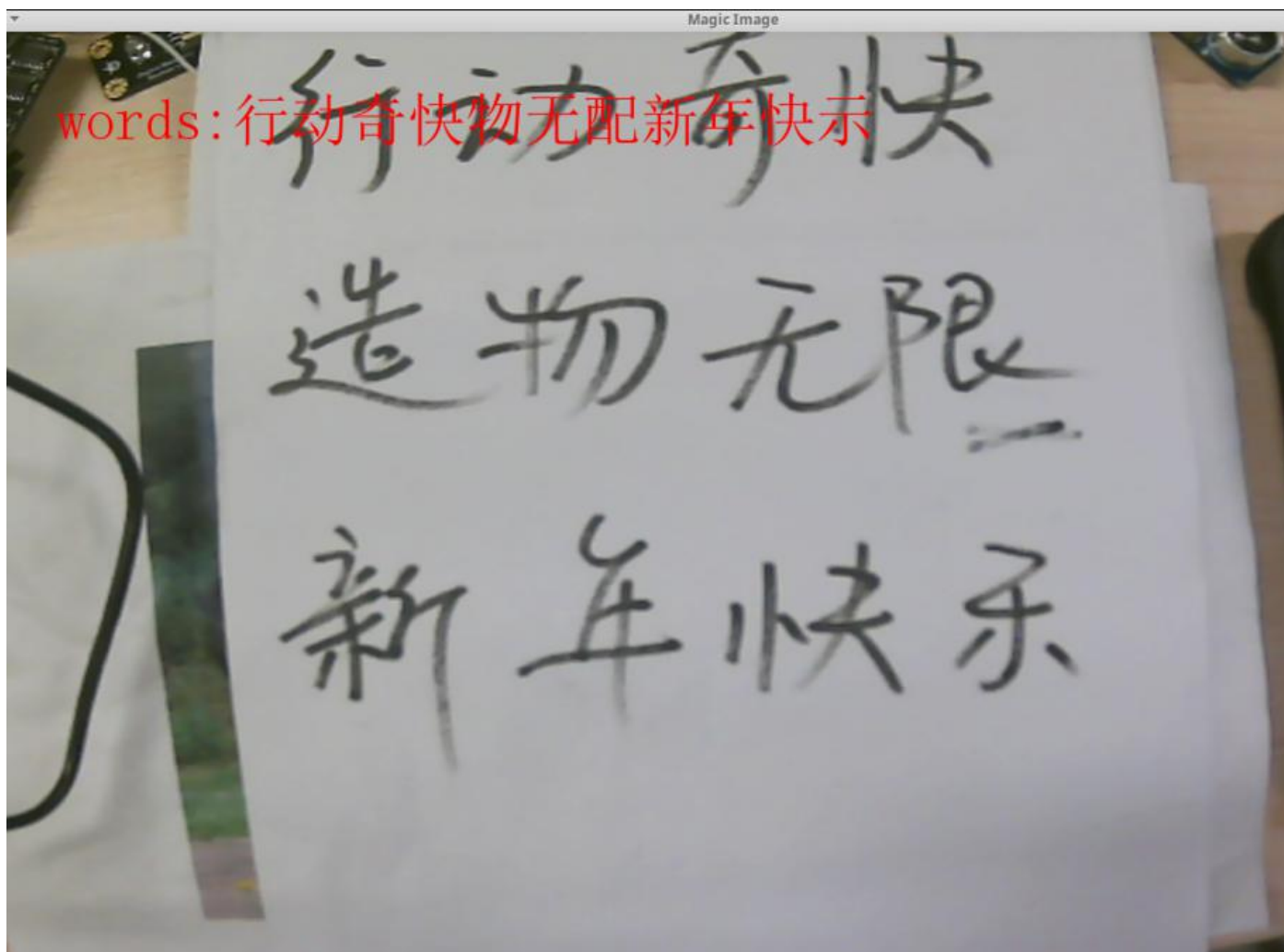




六、识别过程：

识别过程必须联网，将带有数字文字的图片放在摄像头前面，等待一些时间，会出现如下图片：





七、退出程序：

如果想退出程序，在键盘上按 q 键，程序就会自动退出

八、附件：

程序说明

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Date of establishment: November 27, 2018

@author: zhangzd
"""
```

```
import cv2 #导入 cv2 模块
import requests #导入 requests 模块
```




```
import json #导入 json 模块
import threading #导入 threading 模块
import time #导入时间模块
import base64 #导入 base64 模块
import numpy as np #导入 numpy 模块
from PIL import Image, ImageDraw, ImageFont #导入 PIL 模块
```

```
access_token = "" #定义 sccess_token 变量
access_key = ""
API_KEY = "1QUgy55fjIqQgENwxM12vj4Y" #定义 API_KEY 变量
SECRET_KEY = "IApdXqXgVoNDKExlVBC1Zf6Ex3E28BCi" #定义 SECRET_KEY 变量
```

```
frame = None #定义 frame 变量
now_time = 0 #定义 now_time 变量
number = None #定义 number 变量
```

```
def cving_to_b64(img):
    """
    图片转换函数，将二进制图片转换为 base64 加密格式
    """
    try:
        image = cv2.imencode('.jpg', img)[1] #将图片格式转换(编码)成流数据，赋值到内存缓存中
        base64_data = str(base64.b64encode(image))[2:-1] #将图片加密成 base64 格式的数据
        return base64_data #返回加密后的结果
    except Exception as e:
        return "error"
```

```
def get_ai_access_token():
    """
    获取 token 值
    """
    url = "https://aip.baidubce.com/oauth/2.0/token?grant_type=" + \
        "client_credentials&client_id=%s&client_secret=%s" % (API_KEY, SECRET_KEY)
    try:
        response = requests.get(url)
        res_text = response.text
        res_json = json.loads(res_text)
        return str(res_json["access_token"])
    except Exception:
        return "error"
```



```
def get_number(img64):
url = "https://aip.baidubce.com/rest/2.0/ocr/v1/general_basic"
url = url + "?access_token=" + access_token
data = {
"image": img64, "type": 'commontext'
}
try:
response = requests.post(url, files=None, data=data) #向 url 地址发送请求并获取响应
res_text = response.text #获取响应后的结果并将结果转化为字符串格式
res_json = json.loads(res_text) #将字符串转化为 json 格式
return res_json #返回 json 数据
except Exception:
return "error"
```

```
def post_request(frame, nt):
if time.time() - nt > 3: #判断时间差是否大于 3
global now_time #声明 now_time 是全局变量
now_time = time.time() #给 now_time 重新赋值为当前秒数
img64 = cvimg_to_b64(frame) #调用 cvimg_to_b64 函数
res = get_number(img64) #调用 get_number 函数
global number #声明 number 是全局变量
try:
if res["words_result"]: #判断 res 字典中是否含有 words_result
if len(res["words_result"]) == 1: #判断长度
number = res["words_result"][0]["words"]
else:
#将多行内容合并成一行赋值给 number
number = res["words_result"][0]["words"]
for i in range(1, len(res["words_result"])):
number += (res["words_result"][i]["words"])
return
else:
number = None
except:
number = None
```

```
def put_Text(cvimg, text, location, size=50):
"""
将字符信息显示在屏幕上
"""
cvimg = Image.fromarray(cv2.cvtColor(cvimg, cv2.COLOR_BGR2RGB))
draw = ImageDraw.Draw(cvimg)
fontText = ImageFont.truetype("./simsun.ttc", size, encoding="utf-8")
draw.text(location, text, (255,0,0), font=fontText)
```



```
# cvimg = cv2.putText(cvimg, text, location, cv2.FONT_HERSHEY_SIMPLEX, size, (51, 102, 255), 2,)
cvimg = cv2.cvtColor(np.array(cvimg), cv2.COLOR_RGB2BGR)
return cvimg
```

```
def main():
    """
    程序主函数
    """
    token = get_ai_access_token()
    if token != "error":
        global access_token
        access_token = token
    cap = cv2.VideoCapture(0) #创建摄像头对象
    global now_time #声明 now_time 为全局变量
    now_time = time.time() #将当前时间秒数赋值给 now_time
    while (True): #创建一个死循环用于循环读取摄像头数据
        ret, frame = cap.read() #从摄像头中读取一张图片
        if ret == True: #判断是否读取成功
            #创建一个 1280x800 的窗口
            frame1 = cv2.resize(frame, (1280, 800), interpolation=cv2.INTER_LINEAR)
            #创建一个线程用于处理读取到的图片
            t = threading.Thread(target=post_request, args=(frame, now_time,), name='POST_REQUEST')
            t.start() #启动这个线程
            print(number) #打印动物信息
            #在画布上写字
            if number == None:
                frame1 = put_Text(frame1, "Waiting...", (50, 50))
            else:
                try:
                    frame1 = put_Text(frame1, str(number), (200, 50))
                    frame1 = put_Text(frame1, "words:", (50, 50))
                except Exception:
                    pass
            cv2.imshow('Magic Image', frame1)
            if cv2.waitKey(1) & 0xFF == ord('q'): #等待键盘输入 q 后退出程序
                break
        cap.release()
    cv2.destroyAllWindows()
```

```
if __name__ == "__main__":
    main()
```



VBBoard