

Low-Level Software Working

Subject: Engineering Capstone Project

Group Number: 21

Group Members: Deep Shah (8750086) | Jeevan Dsouza (8716171)

Team Name: Vision: A virtual assistant for the visually impaired



Text Recognition for an image stored in an S3 bucket

A sample image was uploaded to the AWS S3 bucket and by using boto3 and python we extracted the image from S3 and did text recognition on the sample image as shown below.



```
Run Command: vision.py
Detected text
-----
Detected text:See your future
Confidence: 99.96%
Id: 0
Type:LINE

Detected text:with new eyes
Confidence: 99.85%
Id: 1
Type:LINE
```

The screenshot shows the AWS Cloud9 IDE interface. At the top, there's a navigation bar with File, Edit, Find, View, Go, Run, Tools, Window, Support, Preview, and Run buttons. Below the navigation bar is a search bar labeled "Go to Anything (⌘ P)". On the left sidebar, there's a file tree showing a folder named "Vision - /home/ec2" containing "README.md" and "vision.py". A small "aws" icon is also present. The main workspace has two tabs: "Welcome" and "vision.py". The "vision.py" tab contains the following Python code:

```
1 #Copyright 2018 Amazon.com, Inc. or its affiliates. All Rights Reserved.
2 #PDX-License-Identifier: MIT-0 (For details, see https://github.com/awsdocs/amazon-rekognition-developer-guide/blob/master/LICENSE-SAMPLECODE.)
3
4 import boto3
5
6 AWS: Add Debug Configuration | AWS: Edit Debug Configuration
7
8 client=boto3.client('rekognition')
9
10 response=client.detect_text(Image={'S3Object':{'Bucket':bucket,'Name':photo}})
11
12 textDetections=response['TextDetections']
13 print ('Detected text\n-----')
14 for text in textDetections:
15     print ('Detected text:' + text['DetectedText'])
16     print ('Confidence: ' + '{:.2f}'.format(text['Confidence']) + "%")
17     print ('Id: {0}'.format(text['Id']))
18     if 'ParentId' in text:
19         print ('Parent Id: {0}'.format(text['ParentId']))
20         print ('Type:' + text['Type'])
21     print()
22
23 return len(textDetections)
```

The status bar at the bottom right indicates "24:12 Python Spaces: 4". Below the workspace, there's a terminal window titled "bash - *p-172-31-11-8.ec2.x vision.py - Stopped". It shows the output of the Python script:

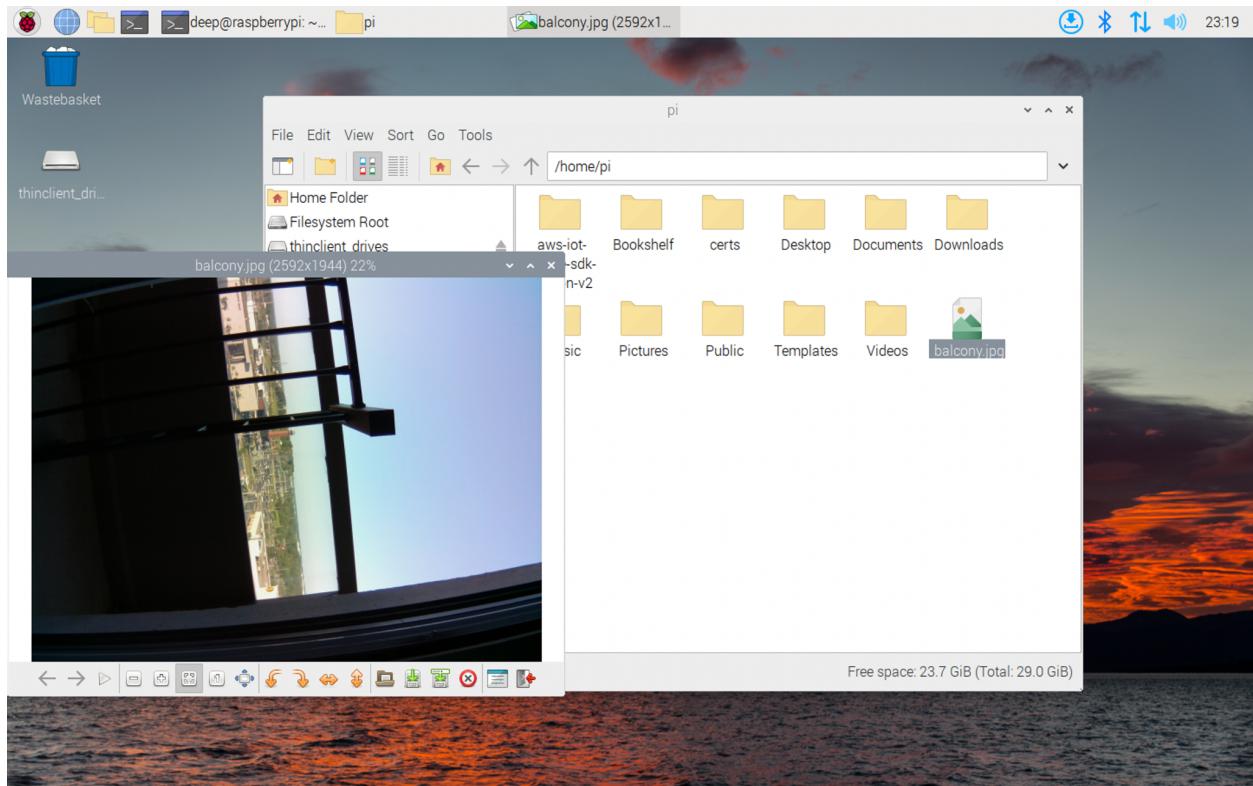
```
Detected text
-----
Detected text:See your future
Confidence: 99.96%
Id: 0
Type:LINE

Detected text:with new eyes
Confidence: 99.85%
Id: 1
Type:LINE
```

The terminal also shows "Command: vision.py" and "Runner: Python 3 CWD ENV".

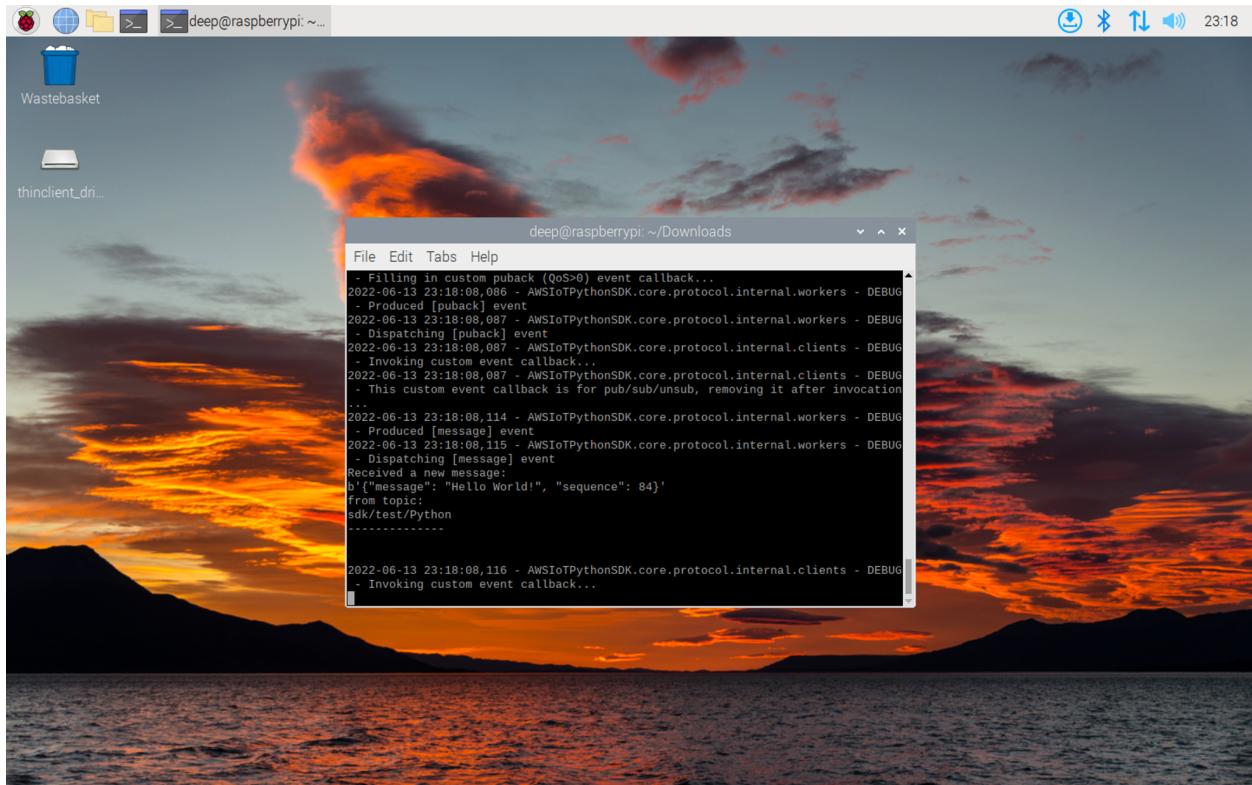
Installed Microsoft Remote Desktop and configured Pi cam

Microsoft Remote Desktop is installed to leverage the GUI capabilities of the Pi and a sample image was taken from the Pi cam as shown below.



Sent data from Raspberry Pi to AWS IOT Core

Our main goal was to make sure that we are able to send data from PI to the cloud. We successfully sent a stream of data from PI to AWS IOT Core by using the MQTT protocol.



The screenshot shows the AWS IoT console interface. On the left, there is a sidebar titled "AWS IoT" with several navigation options: Monitor, Connect, Test, and Manage. Under "Test", the "MQTT test client" option is highlighted. The main area displays a list of received MQTT messages. Each message is shown in a card format with a timestamp and a JSON payload. The payloads are identical, indicating successful test messages.

Message ID	Timestamp	Topic	Message Content
1	June 13, 2022, 18:18:51 (UTC-0400)	sdk/test/Python	{"message": "Hello World!", "sequence": 125}
2	June 13, 2022, 18:18:50 (UTC-0400)	sdk/test/Python	{"message": "Hello World!", "sequence": 124}
3	June 13, 2022, 18:18:49 (UTC-0400)	sdk/test/Python	{"message": "Hello World!", "sequence": 123}
4	June 13, 2022, 18:18:48 (UTC-0400)	sdk/test/Python	{}