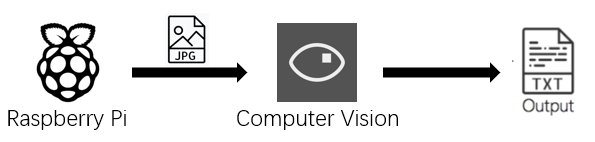
**Objective**: In this lab you will use Azure computer Vision to process license plate image and convert any information on the license plate into text.

**Required Setup**: Connect GrovePi+ board to RPi and have all GrovePi+ libraries installed.

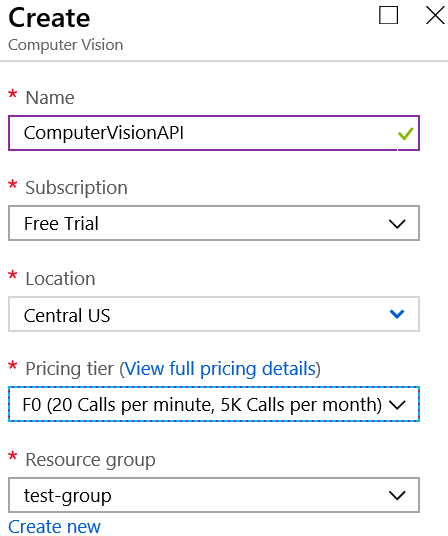
**Parts:**

* RPi 3 B
* 1 x Raspberry Pi camera
* A laptop

Sometime in life, you were driving in the middle of the night. Suddenly, you noticed something flashed for a second. After a week, you received a speeding ticket from the local police department. Do you think that there are people behind the camera to monitor the traffic all the time? No. They are using computers and AI as well. Today we are going to mimic that process using Azure Computer Vision.

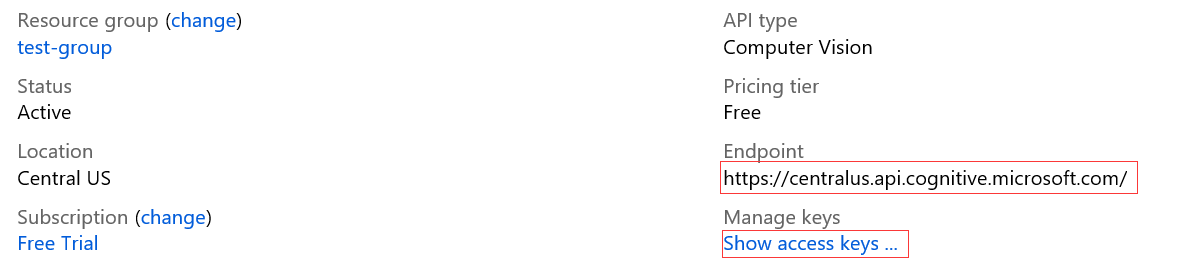


**Step 1.** Sign into the Azure portal using <https://portal.azure.com/#home>. You should see your dashboard.

**Step 2.** Click on ‘+Create a resource’ in the main navigation menu. In the search box, type ‘Computer vision’, find the service and choose it then click on ‘Create’ at the bottom.

**Step 3.** In the prompted window, for the ‘Name’, type ‘ComputerVisionAPI’; For ‘Pricing tier’, choose ‘F0(20 Calls per minute, 5K Calls per month)’. This means you can use this service 20 times in a minute and 5 thousand times for a month for free. The ‘Resource group’ should be ‘test-group’, which we created in previous week. Click on ‘Create’ at the bottom.

**Step 4.** Wait for it to complete deployment and click on ‘Go to resource’. Navigate to ‘Overview’ and you will see the ‘Endpoint’, which is out API endpoint to access this service. Click on ‘Show access keys…’ and copy the ‘Key 1’ from the prompted window. Store both the ‘Endpoint’ and ‘Key 1’ to somewhere for further usage.



**Step 5.** Back to the Raspberry Pi. Turn off the Raspberry Pi and unscrewed the GrovePi off the Raspberry Pi board. Locate the camera port, gently pull up on the edges of the plastic clip and insert the camera ribbon. Make sure it’s the right way round. Push the plastic clip into place.

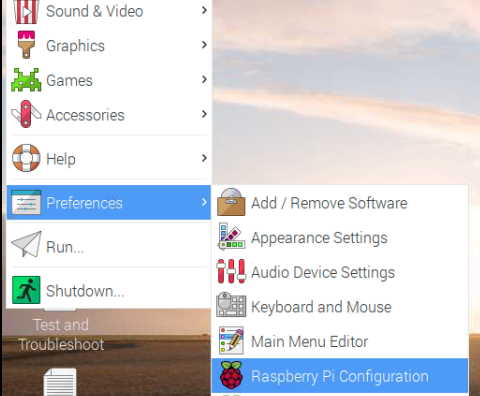


**Step 6.** Feel free to put the GrovePi back on, but we won’t use any sensors this week.

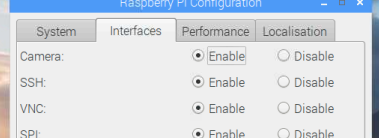
**Step 7.** Boot up the Raspberry Pi and download ‘week 9.py’ from the folder.

**Step 8.** First, we need to do some configuration to be able to use the Pi camera.

**Step 9.** Open the menu and choose ‘Raspberry Pi Configuration’.



**Step 10.** Switch to the ‘Interfaces’ and make sure ‘Camera’ is enabled. Click ‘OK’, then reboot your Pi.



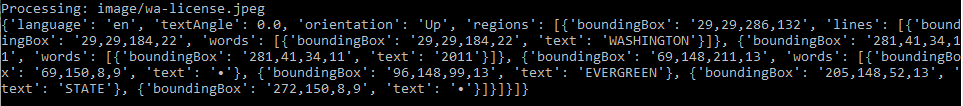
**Step 11.** Download ‘camera.py’ from the folder. Feel free to use either Thonny or terminal to run this script. You should be getting a new file on the desktop called ‘image.jpg’.

**Step 12.** Now download ‘license\_plate.jpeg’ from the folder or use Google to search a license plate and take a picture with the Pi camera. (There might be a chance that Pi Camera is not working, feel free to use ‘wa-license.jpeg’ in the folder directly) Make sure you use a .jpeg image file.

**Step 13.** Download ‘week 8.py’ from the folder. Keep the script on the desktop and create a folder called ‘image’ on the desktop as well. Put the license plate image in the ‘image’ folder. Open the ‘week 8.py’ with Thonny, modify the API\_KEY and ENDPOINT on line 5 and 6 by using the information you stored in **Step 4**.

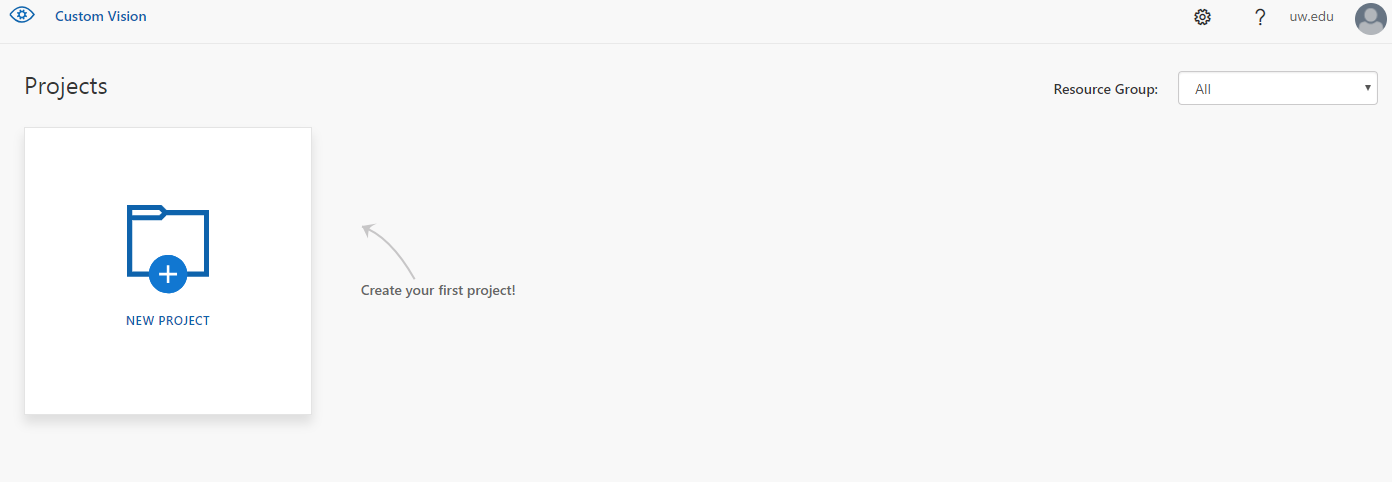


**Step 14.** You should be seeing following output in the console and the script would generate a ‘output.txt’ file.

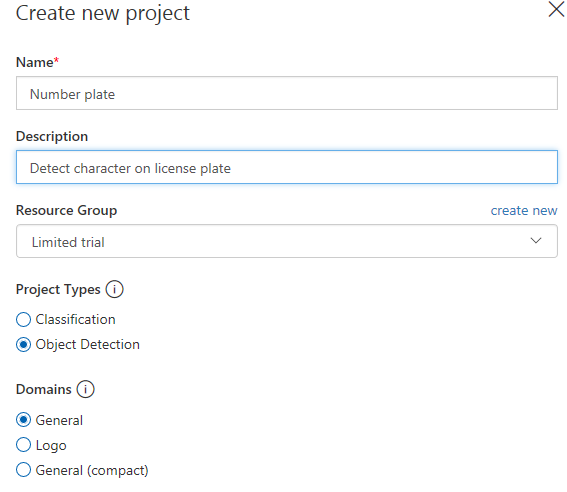


**Step 15.** In the ‘output.txt’ file, you should see “WASHINGTON”, “2011”, and “EVERGREEN STATE”. Obviously, the computer vision service is not picking up the license plate number. The reason is that the service we are using is called ‘OCR’, which stands for Optical Character Recognition. However, although the [document](https://westus.dev.cognitive.microsoft.com/docs/services/5adf991815e1060e6355ad44/operations/56f91f2e778daf14a499e1fc) says it recognizes characters, it’s actually designed to focus on detecting words, fragments, or sentence that makes sense. “WASHINGTON” is an actual word; “EVERGREEN STATE” is a valid fragment. When it reads the license plate number which doesn’t make much sense of being a word, it would skip it. On the other hand, we are using a free version of the service which probably limit the computing power to analyze the image. That will also cause in-complete detection as well.

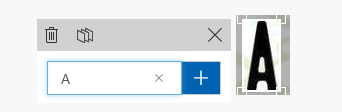
**Step 15.** So now, can we do better? Azure provides another additional service called Custom Vision. Click on this link: <https://www.customvision.ai/> . Sign in and accept the terms of service. You should see the dashboard.



**Step 16.** Click ‘NEW PROJECT’. In the ‘Name’, type ‘Number plate’; you can leave the description blank. For ‘project type’, select ‘Object Detection’. Click ‘Create project’.



**Step 17.** Download ‘A’ folder and there are 15 images of ‘A’ in it. Click on ‘Add images’, upload all 15 images. Click on the first image, hover the cursor on the image and select the region. Type ‘A’ to tag this region.



**Step 18.** Click on the right arrow on the right of the prompted window to go to the next image.

**Step 19.** Keep doing **Step 18** for the rest of images. After tagged all the images, click on ‘Train’ in the top menu.



**Step 20.** As a summary, what you just did above was using these 15 images to teach the engine and algorithm behind the custom vision to understand what letter ‘A’ on the license plate is. Use Google to search more images (minimum 15 images per letter or per number) for all the letters and numbers and train them.

**Step 21.** Then click on ‘Quick Test’. Upload your license plate image and let the engine recognize the characters. You are likely to get better results if you used more images as resources of the recognition engine.



Reference:

1. <https://www.taygan.co/blog/2018/4/28/image-processing-with-cognitive-services>
2. <https://nicksnettravels.builttoroam.com/post/2018/05/24/Building-a-Number-Plate-Identification-Service-in-5-Minutes-with-Microsofts-Custom-Vision-Service.aspx>
3. https://www.henkboelman.com/object-detection-with-microsoft-custom-vision/