

Face API

Have you ever wondered how your camera detects faces? Or how that app everyone was talking about detects age by looking at an image? Or how your photos get sorted on their own? For this project, you will use the Microsoft Azure's Face API. Azure can do all this and much more in a matter of minutes.

Let's get started.

Requirements

You will be using the FaceAPI to do the following:

- 1) Detect a face in an image
- 2) Get face attributes: age, gender, headPose, smile, facialHair, glasses, emotion, hair, makeup, occlusion, accessories, blur, exposure and noise.
- 3) Show the image with the detected faces with rectangles around the faces.

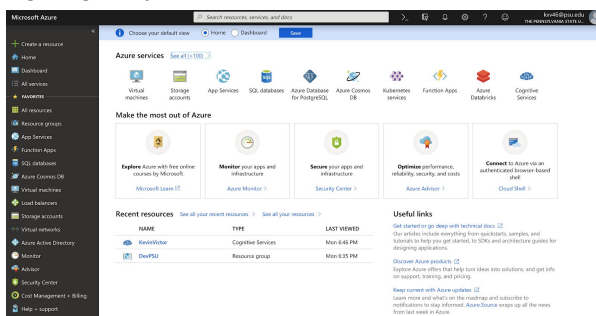
Installation

For this project we will be using Microsoft Azure for students, it is free and does not require a credit card!

Start by clicking this link <https://azure.microsoft.com/en-us/?cdn=disable>

Then click **start free** then **start free** again on the next screen.

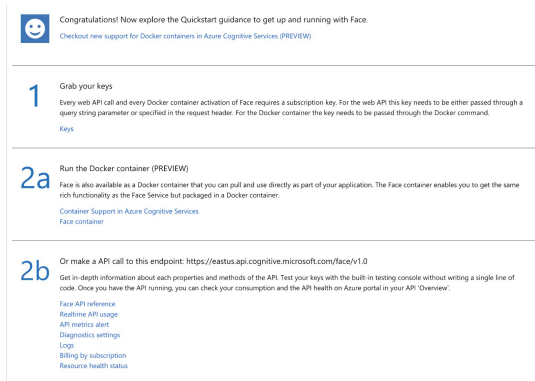
You should then be able to sign in with your penn state access account through microsoft. After signing up you should then be taken to the Azure portal which looks like this!



Next we will need to get our Face API. In the search bar look up **cognitive services**. Next you will need to create a cognitive service. In the marketplace search **Face**.

Then, create a new FaceAPI, where you will need to fill in some information (pricing tier is F0) and create a new resource group (this can be named anything).

After deploying you should then see your new service on the Dashboard! After clicking on your service you should be taken to the quickstart page that looks like this.



Now all you need is to retrieve one of your keys so we can move on to actual programming! Please feel free to explore the API and read the documentation to see how powerful this software actually is.

Instructions

This may seem intimidating at first, but don't worry. We'll help you through it. Now open your preferred text editor with **detectFace.py** to get started.

```
First, pip install cognitive_face
      pip install requests
      pip install Pillow
```

(Windows user might need to use **py -m pip install** or **python -m pip install**)

Then, we will need to do all of our imports. Please import `cognitive_face`, `requests`. Then `BytesIO` from `io` (only `BytesIO` is from `io`), and finally `Image`, `ImageDraw` from `PIL` (you may need to `pip install` these required packages)

Then under FaceAPI details you will need to put your newly acquired **subscription key** in order to activate the API.

Next we will handle our output parameters that the API allows us to use. First set **returnFaceId** to true, and **returnFaceAttributes** to comma separated values of the descriptions we want outputted from our project requirements. (age,gender,etc) (NO SPACES should be between these they must be comma separated as a string, or else you will get an error)

We then need to establish the image we will be using which will be this google stock image of the amazing Steve Jobs.

https://upload.wikimedia.org/wikipedia/commons/thumb/f/f5/Steve_Jobs_Headshot_2010-CROP2.jpg/220px-Steve_Jobs_Headshot_2010-CROP2.jpg

Next we initialize our API with the web server in order to retrieve the output data (this is done for you)

If you print the code we have so far you should be able to output all of the face attributes of our image!

Lastly, we will draw the rectangles on the image using Pillow so we can visually identify where the faces are. Under the **getRectangle** function fill in the correct dictionary keys for **rect** in order to get the boundaries of each rectangle. (hint: bottom is the sum of left and height, while right is the sum of top and width)

Finally, we will download the image, outline the rectangles in red, and use **img.show()** display the marked image in a browser. (this is done for you)

Some things to look out for:

- Make sure to use your own subscription key that is valid and active
- Pay close attention to the url (make sure it is the correct region of service)
- pip install the required packages
 - PIL might have some issues on mac, Pillow works also since it is a fork of PIL

Submission

Confirm that what you did worked as expected by reading the generated JSON carefully. It should have the details and attributes as expected. Please submit your code as well as the output of your code to canvas.

Awesome! You have now successfully used an API to detect faces on an image. However we have only scratched the surface of what this service can do. Try testing with multiple photos or photos with multiple people in it (attributes are then listed in the order of the largest rectangle detected).

Extra

You have now built a basic script to detect and read attributes and emotion from an image, build on this further with a use case of your choice like:

- Emotion detector
- Age Detector
- Group photos (using Face-Group API)
- Identify a particular person (using Face-Identify API)
- Find similar faces