In [ ]:
```
! activate ai-azure-c1

import sys

sys.path.append("/opt/conda/envs/ai-azure-c1/lib/python3.8/site-packages")
```

In [ ]:
```
!pip install Pillow==8.4
```

```
Collecting Pillow==8.4
  Downloading https://files.pythonhosted.org/packages/7d/2a/2fc11b54e2742db06297f7fa7f420a0e3069fdcf0e4b57dfec33f0b08622/Pillow-8.4.0.tar.gz (49.4MB)
    100% |████████████████████████████████| 49.4MB 768kB/s eta 0:00:01    96% |███████████████████████████████ | 47.9MB 57.8MB/s eta 0:00:01
Building wheels for collected packages: Pillow
  Running setup.py bdist_wheel for Pillow ... done
  Stored in directory: /root/.cache/pip/wheels/a7/69/9a/bba9fca6782340f88dbc378893095722a663cbc618e58fe401
Successfully built Pillow
scikit-image 0.14.2 has requirement dask[array]>=1.0.0, but you'll have dask 0.16.1 which is incompatible.
Installing collected packages: Pillow
  Found existing installation: Pillow 5.2.0
    Uninstalling Pillow-5.2.0:
      Successfully uninstalled Pillow-5.2.0
Successfully installed Pillow-8.4.0
```

In [ ]:
```
import io
import datetime
import pandas as pd
from PIL import Image
import requests
import io
import glob, os, sys, time, uuid

from matplotlib.pyplot import imshow
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
%matplotlib inline

from urllib.parse import urlparse
from io import BytesIO
from PIL import Image, ImageDraw

from video_indexer import VideoIndexer
from azure.cognitiveservices.vision.face import FaceClient
from azure.cognitiveservices.vision.face.models import TrainingStatusType
from msrest.authentication import CognitiveServicesCredentials
```

In [ ]:
```
def display_image(path):
    img = mpimg.imread(path)
    imgplot = plt.imshow(img)
    plt.show()
```

In [ ]:
```
video_path =  "/workspace/home/pascal-boarding-pass.mp4"

video_analysis = VideoIndexer(
    vi_subscription_key='e20396c824664dd48aa4d63b7ac6fbe6',
    vi_location='trial',
    vi_account_id='aaf5f40f-498c-4d1f-b444-aeb42181e814'
)
```

In [ ]:
```
video_analysis.check_access_token()
```

```
Getting video indexer access token...
Access Token: eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJWZXJzaW9uIjoiMi4wLjAuMCIsIktleVZlcnNpb24iOiI5NGQyMWY0ZDZlY2Y0MzRmOGUzYmRhMDVjMWU3MmRhZCIsIkFjY291bnRJZCI6ImFhZjVmNDBmLTQ5OGMtNGQxZi1iNDQ0LWFlYjQyMTgxZTgxNCIsIkFjY291bnR
UeXBlIjoiVHJpYWwiLCJQZXJtaXNzaW9uIjoiQ29udHJpYnV0b3IiLCJFeHRlcm5hbFVzZXJJZCI6IkIwOEQ2RDFFNkI0OTREQTFBRTc3MzQzRDc2Q0VBNDA2IiwiVXNlclR5cGUiOiJNaWNyb3NvZnRDb3JwQWFkIiwiSXNzdWVyTG9jYXRpb24iOiJUcmlhbCIsIm5iZiI6MTcwMzMyNDI4MiwiZ
XhwIjoxNzAzMzI4MTgyLCJpc3MiOiJodHRwczovL2FwaS52aWRlb2luZGV4ZXIuYWkvIiwiYXVkIjoiaHR0cHM6Ly9hcGkudmlkZW9pbmRleGVyLmFpLyJ9.DeqI-fbuxfrKmKEA7jGopXf1u-jUEdzXEkeHt1ewMkFqB2ANBqH_thdLf2xJ916lCDC6iRh_bMdTPsUZLfODn9hKhP53-ZPQHmn1tn
M1JhFii7PYPhLhywuuJiuxWi8m7Kc9D-GEaCfurp-nfbotsCeij4w7Csvdf_ReAh9gXLY2MOu_17BzByTXvOl-NR8rHhvvIPQO6Z29wG2cEUN32QZjHy4RqusJMkDM3eSHviySm9sITMpb3CxXuO4f71C3AzfB1J1bPCq0LpHlaMxuLu3gZiI9pNwEJxiMdU92E8h8mmEfIvgEg3adN2T4dNfXyQGC
xTLQ4E8c0CIslKjq1w
```

```
In [ ]: video_id = video_analysis.upload_to_video_indexer(
                                                   input_filename=video_path,
                                                   video_name='pascal-kiosk-30s',
                                                   video_language='English'
                                                    )
```

```
Uploading video to video indexer...
```

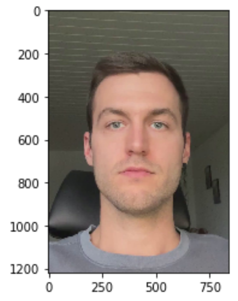
```
In [ ]: video_info = video_analysis.get_video_info(video_id, video_language='English')
```

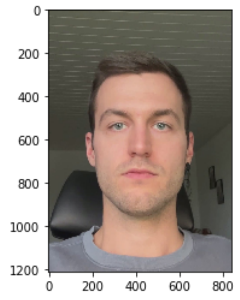
```
Getting video info for: 128fbab0a4
```

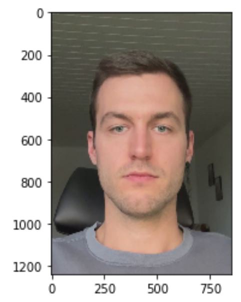
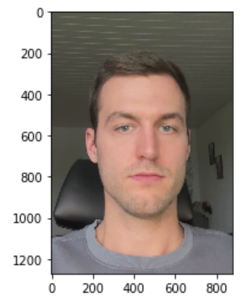
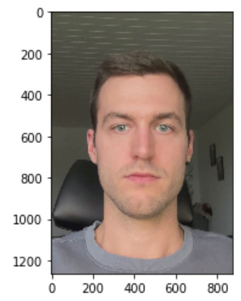
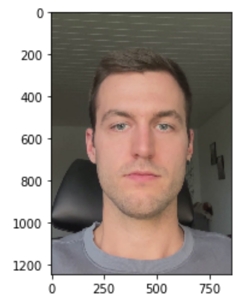
```
In [ ]: thumb_images = []
        thumb_id = []
        for thumb in video_info['videos'][0]['insights']['faces'][0]['thumbnails']:
            if 'fileName' in thumb and 'id' in thumb:
                file_name = thumb['fileName']
                id = thumb['id']
                video_img = video_analysis.get_thumbnail_from_video_indexer(video_id,  id)
                stream = io.BytesIO(video_img)
                img = Image.open(stream)
                thumb_images.append(img)
                thumb_id.append(thumb['id'])
```

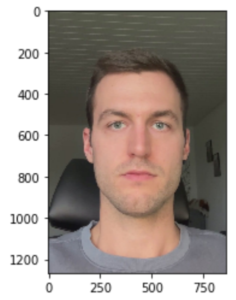
```
Getting thumbnail from video: 128fbab0a4, thumbnail: f5f7c4b0-3493-4e6d-998f-237060aa3229
Getting thumbnail from video: 128fbab0a4, thumbnail: c8a80ded-0051-462b-8d11-a86465162636
Getting thumbnail from video: 128fbab0a4, thumbnail: c1837812-2f4b-4511-b40b-5544fc5d17c4
Getting thumbnail from video: 128fbab0a4, thumbnail: d51ba933-2131-440d-9a5f-8d1a45b96205
Getting thumbnail from video: 128fbab0a4, thumbnail: 33c254bb-c8ab-487f-afbf-a7133e4f7688
Getting thumbnail from video: 128fbab0a4, thumbnail: b9269554-f3a7-41d1-9fe1-983cdc0156af
Getting thumbnail from video: 128fbab0a4, thumbnail: 9eec8011-4f3b-45c2-8d15-256fae4ef82b
Getting thumbnail from video: 128fbab0a4, thumbnail: 62922bfa-c54d-4c78-9658-7c735273da78
```

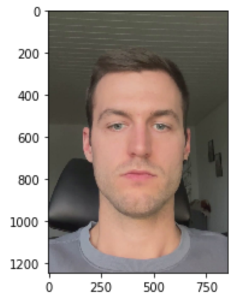
```
In [ ]: for plt_image in thumb_images:
            plt.figure()
            plt.imshow(plt_image)
```






0
200
400
600
800
1000
1200
0
250
500
750

0
200
400
600
800
1000
1200
0
200
400
600
800

0
200
400
600
800
1000
1200
0
200
400
600
800

0
200
400
600
800
1000
1200
0
250
500
750





In [ ]:
```python
n = 1
for image_save in thumb_images:
    print(type(image_save))
    image_save.save('train_image' + str(n) + '.jpg')
    n += 1
```

```
<class 'PIL.JpegImagePlugin.JpegImageFile'>
<class 'PIL.JpegImagePlugin.JpegImageFile'>
<class 'PIL.JpegImagePlugin.JpegImageFile'>
<class 'PIL.JpegImagePlugin.JpegImageFile'>
<class 'PIL.JpegImagePlugin.JpegImageFile'>
<class 'PIL.JpegImagePlugin.JpegImageFile'>
<class 'PIL.JpegImagePlugin.JpegImageFile'>
<class 'PIL.JpegImagePlugin.JpegImageFile'>
```

In [ ]:
```python
person_id = str(uuid.uuid4())
person_name = str(uuid.uuid4())
```

In [ ]:
```python
## This code is taken from Azure Face SDK
## --------------------------------------
def build_person_group(client, person_group_id, pgp_name):
    print('Create and build a person group...')
    # Create empty Person Group. Person Group ID must be lower case, alphanumeric, and/or with '-', '_'.
    print('Person group ID:', person_group_id)
    client.person_group.create(person_group_id = person_group_id, name=person_group_id)

    # Create a person group person.
    human_person = client.person_group_person.create(person_group_id, pgp_name)
    # Find all jpeg human images in working directory.
    human_face_images = [file for file in glob.glob('*.jpg') if file.startswith("train_image")]
    # Add images to a Person object
    for image_p in human_face_images:
        with open(image_p, 'rb') as w:
            client.person_group_person.add_face_from_stream(person_group_id, human_person.person_id, w)

    # Train the person group, after a Person object with many images were added to it.
    client.person_group.train(person_group_id)
    # Wait for training to finish.
    while (True):
        training_status = client.person_group.get_training_status(person_group_id)
        print("Training status: {}.".format(training_status.status))
        if (training_status.status is TrainingStatusType.succeeded):
            break
        elif (training_status.status is TrainingStatusType.failed):
            client.person_group.delete(person_group_id=PERSON_GROUP_ID)
            sys.exit('Training the person group has failed.')
        time.sleep(5)
```

In [ ]:
```
face_service_key = "daa35ffab0e2498090d181e20cbae73c"
face_service_endpoint = "https://udacity-face.cognitiveservices.azure.com/"

face_client = FaceClient(face_service_endpoint, CognitiveServicesCredentials(face_service_key))
```

In [ ]:
```
build_person_group(face_client, person_id, person_name)
```

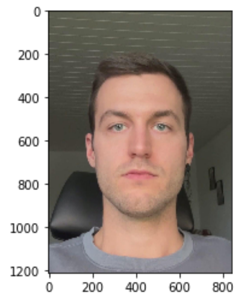
```
Create and build a person group...
Person group ID: d1261b5c-22ed-4dca-bb5a-b99da6431646
Training status: running.
Training status: succeeded.
```

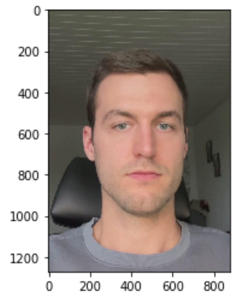
In [ ]:
```
person_id
```

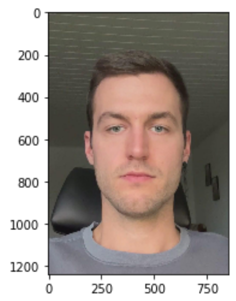
Out[ ]:
```
'd1261b5c-22ed-4dca-bb5a-b99da6431646'
```

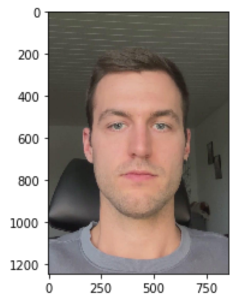
In [ ]:
```
test_image = "/workspace/home/train_image5.jpg"
```

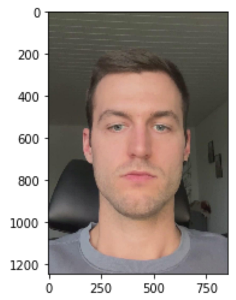
In [ ]:
```
train_image_list = [file for file in glob.glob('*.jpg') if file.startswith("train_image")]
for test_image_plt in train_image_list:
    display_image(test_image_plt)
```

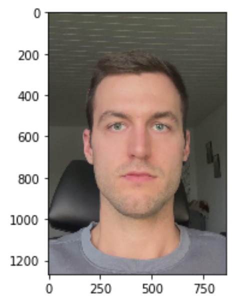


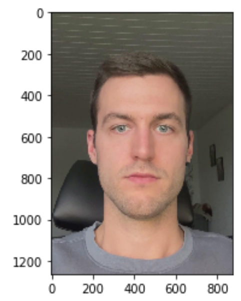


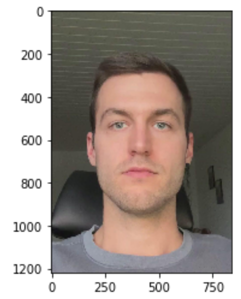

0
200
400
600
800
1000
1200
0
250
500
750


0
200
400
600
800
1000
1200
0
250
500
750


0
200
400
600
800
1000
1200
0
250
500
750


0
200
400
600
800
1000
1200
0
250
500
750





In [ ]:
```
test_image_read = open(test_image, 'rb')
faces = face_client.face.detect_with_stream(test_image_read)
face_id_list = []
for face in faces:
    face_id_list.append(face.face_id)
    print("Face ID:", face.face_id)
```

```
Face ID: 13ffd6b6-4754-42e9-8d61-a74bf1f07284
```

In [ ]:
```
face_id_list
```

Out[ ]:
```
['13ffd6b6-4754-42e9-8d61-a74bf1f07284']
```

In [ ]:
```
person_groupe_id = person_id
face_identity_result = face_client.face.identify(face_id_list, person_groupe_id)
```

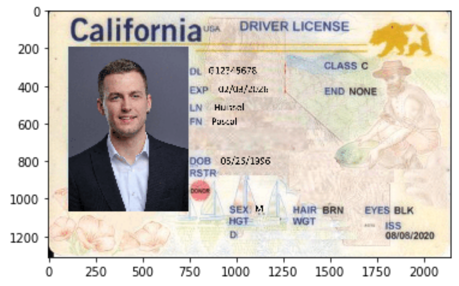
In [ ]:
```
for r in face_identity_result:
    for identity in r.candidates:
        print("The Identity is the same with confidence", identity.confidence * 100 , "%")
```

```
The Identity is the same with confidence 98.178 %
```

In [ ]:
```
id_card_path = "/workspace/home/ca-dl-pascal-huissel.png"
id_card_read = open(id_card_path, 'rb')
id_card_face = face_client.face.detect_with_stream(id_card_read)
id_card_face_list = []
for f in id_card_face:
    id_card_face_list.append(f.face_id)
    print("Face ID:", f.face_id)
```

```
Face ID: ae698ac1-4363-4ba4-add6-d587cdb78667
```

In [ ]:
```
display_image(id_card_path)
```


California USA DRIVER LICENSE

DL G12345678

EXP 02/04/2026

LN Hustal

FN Pascal

DOB 05/25/1956

RSTR

CLASS C

END NONE

SEX M

HGT

D

HAIR BRN

WGT

EYES BLK

ISS 08/08/2020

DONOR
0
200
400
600
800
1000
1200
0
250
500
750
1000
1250
1500
1750
2000

In [ ]:
```
face_identity_result = face_client.face.identify(id_card_face_list, person_groupe_id)
for r in face_identity_result:
    for identity in r.candidates:
        print("The Identity is the same with confidence", identity.confidence * 100 , "%")
```

```
The Identity is the same with confidence 67.242 %
```

In [ ]:
```
passenger_sentiments = video_info['summarizedInsights']['sentiments']

if len(passenger_sentiments) > 0:
    for sentiment in passenger_sentiments:
        print("Recognized passenger sentiment: ", sentiment)
else:
    print("No sentiments detected")
```

```
No sentiments detected
```

In [ ]:
```
passenger_emotions = video_info['summarizedInsights']['emotions']

if len(passenger_emotions) > 0:
    for emotion in passenger_emotions:
        print("Recognized passenger emotion: ", emotion)
else:
    print("No emotions detected")
```

```
No emotions detected
```

In [ ]:
```

```