Please install the required Python modules/SDKs

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

import sys

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

In [ ]:
```
from azure.ai.formrecognizer import FormRecognizerClient
from azure.ai.formrecognizer import FormTrainingClient
from azure.core.credentials import AzureKeyCredential
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import pandas as pd
import os
from datetime import datetime
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
import io
import glob, os, sys, time, uuid
from azure.cognitiveservices.vision.customvision.training import CustomVisionTrainingClient
from azure.cognitiveservices.vision.customvision.prediction import CustomVisionPredictionClient
from azure.cognitiveservices.vision.customvision.training.models import ImageFileCreateBatch, ImageFileCreateEntry, Region
from msrest.authentication import ApiKeyCredentials
```

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

In [ ]:
```
fligth_manifest_path = "/workspace/home/MyFlightManifest.csv"
fligth_manifest_df = pd.read_csv(fligth_manifest_path)
```

In [ ]:
```
fligth_manifest_df
```

Out[ ]:

	Flight No	Origin	Destination	Date	Time	First Name	Last Name	Sex	SeatNo	DateofBirth	DoBValidation	PersonValidation	LuggageValidation	NameValidation	BoardingPassValidation
0	234	San Francisco	Chicago	April 20 2022	10:00 AM PST	Avkash	Chauhan	M	20A	01/01/1990	False	False	False	False	False
1	234	San Francisco	Chicago	April 20 2022	10:00 AM PST	James	Jackson	M	25B	10/12/1956	False	False	False	False	False
2	234	San Francisco	Chicago	April 20 2022	10:00 AM PST	Libby	Herold	F	3D	02/10/1996	False	False	False	False	False
3	234	San Francisco	Chicago	April 20 2022	10:00 AM PST	James	Webb	M	1A	12/15/1970	False	False	False	False	False
4	234	San Francisco	Chicago	April 20 2022	10:00 AM PST	Radha	Kumar	F	34B	03/05/1994	False	False	False	False	False
5	234	San Francisco	Chicago	April 20 2022	10:00 AM PST	Sameer	Kumar	M	34A	01/25/1990	False	False	False	False	False
6	289	Stuttgart	New York	December 21 2023	15:00 AM PST	Pascal	Huissel	M	5A	05/26/1996	False	False	False	False	False

In [ ]:
```
endpoint_form_recognizer = "https://udacity-fr-project.cognitiveservices.azure.com/"
key_form_recognizer = "a9ccd13b54544a20bb8e724976f7cf20"
training_data_url = "https://udacitykioskstorage.blob.core.windows.net/passengerdata?sp=racwdli&st=2023-12-21T14:01:29Z&se=2023-12-30T22:01:29Z&spr=https&sv=2022-11-02&sr=c&sig=FKu4N%2FtaLeZC6vQKIaPBJ5MyJI%2Fk9aJFqQ71b1vd

form_training_client = FormTrainingClient(endpoint=endpoint_form_recognizer, credential=AzureKeyCredential(key_form_recognizer))
form_recognizer_client = FormRecognizerClient(endpoint=endpoint_form_recognizer, credential=AzureKeyCredential(key_form_recognizer))
```

In [ ]:
```
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
UeXBlIjoiVHJpYWwiLCJQZXJtaXNzaW9uIjoiQ29udHJpYnV0b3IiLCJFeHRlcm5hbFVzZXJJZCI6IkIwOEQ2RDFFNkI0OTREQTFBRTc3MzQzRDc2Q0VBNDA2IiwiVXNlclR5cGUiOiJNaWNyb3NvZnRDb3JwQWFkIiwiSXNzdWVyTG9jYXRpb24iOiJUcmlhbCIsIm5iZiI6MTcwMzM0ODYwMSwiZ
XhwIjoxNzAzMzUyNTAxLCJpc3MiOiJodHRwczovL2FwaS52aWRlb2luZGV4ZXIuYWkvIiwiYXVkIjoiaHR0cHM6Ly9hcGkudmlkZW9pbmRleGVyLmFpLyJ9.Wogfzxa851nj0SUwgcbfltaONwuKeEE3aNYf2dj4Y0zx8QTC0C-SkAffdgaYDYGvr8osB6s9gitq6w4Q0Kuycb74wjtN5B_GztKJ3w
SgDvUvv06S7N1bP5hOPu413K-vNJf8n4ihMKaR74PO677xCdeZWOQSeK99g4dNpW2jhjkuvYYn7cOdZNid-Cyq5ou6p8qkGGMYwhSHahIipGhSb5fDpCfliOQmoXk4XBZtQWz6s9SniQCXaTksR-2uJx6NpMvOzf_lYgDj10L5narsBHjFqqg1fyRhjlFOJsyayTdOr74Q9ctX2hUvNRO9SpG3ooK2
ba0MzHueL8xIm9oI0g
```

In [ ]:
```
pred_endpoint = "https://udacityligtherdetection-prediction.cognitiveservices.azure.com/"
pred_key = "ba7ce73bb517493db18927c49e94d379"
pred_resource_id = "/subscriptions/a70a7281-34b4-43ff-932a-1f9171daff2c/resourceGroups/aind-246477/providers/Microsoft.CognitiveServices/accounts/udacityligtherdetection-Prediction"

pred_cred = ApiKeyCredentials(in_headers={"Prediction-key": pred_key})
prediction_model = CustomVisionPredictionClient(endpoint=pred_endpoint, credentials=pred_cred)
```

In [ ]:
```
def id_text_extraction(id_content_path):
    """
    Extracts text from image using Azure form recognizer
    """
    with open(os.path.join (id_content_path), "rb") as image:
        id_content = form_recognizer_client.begin_recognize_identity_documents(image.read(), content_type = "image/png")
        id_content_result = id_content.result()

    return id_content_result
```

In [ ]:
```
training_process = form_training_client.begin_training(training_data_url, use_training_labels=True, include_subfolders=False)
trained_model = training_process.result()
print("Trained model status: ", trained_model.status)

# Code based on: L3 Exercise 3 - Custom Form 1040 Recognizer Exercise Solution.ipynb
for submodel in trained_model.submodels:
    print(
        "The trained model with form type '{}' has overall accuracy '{}' and recognized the following fields: {}".format(
            submodel.form_type,
            submodel.accuracy,
            ", ".join(
                [
                    field.label if field.label else name
                    for name, field in submodel.fields.items()
                ]
            ),
        )
    )
```

```
Trained model status:  ready
The trained model with form type 'custom:ff732f68-b342-4496-ada0-d52f7e24f959' has overall accuracy '0.992' and recognized the following fields: Baggage, Boarding Time, Carrier, Class, Date, Destination, Flight No., Gate,
Origin, Passenger Name, Seat
```

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 [ ]:
```
## 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.
```

```python
    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 [ ]:
```python
def boarding_pass_extraction(boarding_pass_content_path):

    test_boarding_pass = form_recognizer_client.begin_recognize_custom_forms_from_url(model_id=trained_model.model_id, form_url=boarding_pass_content_path)
    test_boarding_pass_result = test_boarding_pass.result()

    return test_boarding_pass_result
```

In [ ]:
```python
def check_3_way_name(id_text, boarding_pass_text):
    id_first_name = id_text.fields.get("FirstName").value
    id_last_name = id_text.fields.get("LastName").value
    boarding_pass_name = boarding_pass_text.fields.get("Passenger Name").value
    boarding_pass_first_name = boarding_pass_name.split(" ")[0]
    boarding_pass_last_name = boarding_pass_name.split(" ")[1]

    if id_first_name == boarding_pass_first_name and id_last_name == boarding_pass_last_name:
        if fligth_manifest_df['First Name'].isin([id_first_name]).any() and fligth_manifest_df['Last Name'].isin([id_last_name]).any():
           name_idx = fligth_manifest_df.index[fligth_manifest_df['Last Name']==id_last_name].tolist()[0]
           fligth_manifest_df.loc[name_idx, 'NameValidation'] = 'True'
           return 'True'
    else:
        return 'False'
```

In [ ]:
```python
def check_DoB(id_text):
    DoB = id_text.fields.get("DateOfBirth").value
    id_last_name = id_text.fields.get("LastName").value
    DoB = DoB.strftime('%m/%d/%Y')
    name_idx = fligth_manifest_df.index[fligth_manifest_df['Last Name']==id_last_name].tolist()[0]
    DoB_fligth = fligth_manifest_df._get_value(name_idx, 'DateofBirth')
    DoB_fligth_date = datetime.strptime(DoB_fligth, '%m/%d/%Y').date().strftime('%m/%d/%Y')

    if DoB_fligth_date == DoB:
        fligth_manifest_df.loc[name_idx, 'DoBValidation'] = 'True'
        return 'True'
    else:
        return 'False'
```

In [ ]:
```python
def check_boarding_pass(boarding_pass_text):
    boarding_pass_name = boarding_pass_text.fields.get("Passenger Name").value
    boarding_pass_first_name = boarding_pass_name.split(" ")[0]
    boarding_pass_last_name = boarding_pass_name.split(" ")[1]

    boarding_pass_fligth_num = boarding_pass_text.fields.get("Flight No.").value
    boarding_pass_seat_num = boarding_pass_text.fields.get("Seat").value
    boarding_pass_origin = boarding_pass_text.fields.get("Origin").value
    boarding_pass_destination = boarding_pass_text.fields.get("Destination").value
    boarding_pass_time = boarding_pass_text.fields.get("Boarding Time").value
    boarding_pass_date = boarding_pass_text.fields.get("Date").value
    name_idx = fligth_manifest_df.index[fligth_manifest_df['Last Name']==boarding_pass_last_name].tolist()[0]

    fm_fligth_num = fligth_manifest_df._get_value(name_idx, 'Flight No')
    fm_seat_num = fligth_manifest_df._get_value(name_idx, 'SeatNo')
    fm_origin = fligth_manifest_df._get_value(name_idx, 'Origin')
    fm_destination = fligth_manifest_df._get_value(name_idx, 'Destination')
    fm_time = fligth_manifest_df._get_value(name_idx, 'Time')
    fm_date = fligth_manifest_df._get_value(name_idx, 'Date')

    boarding_pass_date = boarding_pass_date.replace(",", "")
    fm_fligth_num = str(fm_fligth_num)

    if boarding_pass_fligth_num == fm_fligth_num and fm_seat_num == boarding_pass_seat_num and boarding_pass_origin == fm_origin and boarding_pass_destination == fm_destination and boarding_pass_time == fm_time and boardin
        fligth_manifest_df.loc[name_idx, 'BoardingPassValidation'] = 'True'
        return "True"

    else:
         return "False"
```

In [ ]:
```
id_card_pascal = "/workspace/home/digital_id/ca-dl-pascal-huissel.png"
id_card_avkash = "/workspace/home/digital_id/ca-dl-avkash.png"
id_card_james = "/workspace/home/digital_id/ca-dl-james-webb.png"
```

In [ ]:
```
id_text_pascal = id_text_extraction(id_card_pascal)
id_text_avkash = id_text_extraction(id_card_avkash)
id_text_james = id_text_extraction(id_card_james)
```

In [ ]:
```
boarding_pass_pascal = "https://udacitykioskstorage.blob.core.windows.net/passengerdata/boarding_pascal.pdf?sp=r&st=2023-12-21T14:21:09Z&se=2023-12-25T22:21:09Z&spr=https&sv=2022-11-02&sr=b&sig=NH9VFDB%2BDQuoRY%2F0Okpoy7du
boarding_pass_avkash = "https://udacitykioskstorage.blob.core.windows.net/passengerdata/boarding-avkash.pdf?sp=r&st=2023-12-21T14:21:47Z&se=2023-12-25T22:21:47Z&spr=https&sv=2022-11-02&sr=b&sig=ct%2BxqSQ2NPm1uPLNuwc6QE%2Bj
boarding_pass_james = "https://udacitykioskstorage.blob.core.windows.net/passengerdata/boarding-james-webb.pdf?sp=r&st=2023-12-21T14:22:26Z&se=2023-12-26T22:22:26Z&spr=https&sv=2022-11-02&sr=b&sig=L9W76A1CH3Sh4gVxNj4gpSFm6
```

In [ ]:
```
boarding_pass_text_pascal = boarding_pass_extraction(boarding_pass_pascal)
boarding_pass_text_avkash = boarding_pass_extraction(boarding_pass_avkash)
boarding_pass_text_james = boarding_pass_extraction(boarding_pass_james)
```

In [ ]:
```
pascal_name_check = check_3_way_name(id_text_pascal[0], boarding_pass_text_pascal[0])
avkash_name_check = check_3_way_name(id_text_avkash[0], boarding_pass_text_avkash[0])
james_name_check = check_3_way_name(id_text_james[0], boarding_pass_text_james[0])
```

In [ ]:
```
pascal_dob_check = check_DoB(id_text_pascal[0])
avkash_dob_check = check_DoB(id_text_avkash[0])
james_dob_check = check_DoB(id_text_james[0])
```

In [ ]:
```
pascal_boardind_pass_check = check_boarding_pass(boarding_pass_text_pascal[0])
avkash_boardind_pass_check = check_boarding_pass(boarding_pass_text_avkash[0])
james_boardind_pass_check = check_boarding_pass(boarding_pass_text_james[0])
```

In [ ]:
```
video_pascal =  "/workspace/home/digital-video/pascal-boarding-pass.mp4"
video_avkash = "/workspace/home/digital-video/avkash-boarding-pass.mp4"
video_id_pascal = video_analysis.upload_to_video_indexer(
                                                   input_filename=video_pascal,
                                                   video_name='pascal-kiosk-30s',
                                                   video_language='English'
                                                    )
video_id_avkash = video_analysis.upload_to_video_indexer(
                                                   input_filename=video_avkash,
                                                   video_name='avkash-kiosk-30s',
                                                   video_language='English'
                                                    )
```

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

In [ ]:
```
def get_person_from_video(video_id_person):
    video_info = video_analysis.get_video_info(video_id_person, video_language='English')
    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_person,  id)
            stream = io.BytesIO(video_img)
            img = Image.open(stream)
            thumb_images.append(img)
            thumb_id.append(thumb['id'])

    n = 1
    for image_save in thumb_images:
        print(type(image_save))
        image_save.save('train_image' + str(n) + '.jpg')
        n += 1

    person_id = str(uuid.uuid4())
    person_name = str(uuid.uuid4())
    build_person_group(face_client, person_id, person_name)

    return person_id
```

In [ ]:
```
def get_person_from_id_card(id_card_person_path):
    id_card_read = open(id_card_person_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)
    return id_card_face_list
```

In [ ]:
```
def PersonValidation(id_card_face_list, video_person_id):
    face_identity_result = face_client.face.identify(id_card_face_list, video_person_id)
    PersonValidation_flag = False
    for r in face_identity_result:
        for identity in r.candidates:
            print("The Identity is the same with confidence", identity.confidence * 100 , "%")
            if identity.confidence > 0.65:
                PersonValidation_flag = True
    return PersonValidation_flag
```

In [ ]:
```
person_id_pascal = get_person_from_video(video_id_pascal)
```

```
Getting video info for: 21af5d7acb
Getting thumbnail from video: 21af5d7acb, thumbnail: 14dff1fa-a1f5-423f-a676-4dec99e36ba0
Getting thumbnail from video: 21af5d7acb, thumbnail: f7fc2958-ea35-497b-8639-768bcfd96983
Getting thumbnail from video: 21af5d7acb, thumbnail: 6206dcb8-2b15-4fab-9d64-1bfcc9d6ad92
Getting thumbnail from video: 21af5d7acb, thumbnail: d3e9c16c-447a-49ab-a88b-f141a7f3e198
Getting thumbnail from video: 21af5d7acb, thumbnail: 0b821f02-d176-499b-9a50-d5f404daa4ca
Getting thumbnail from video: 21af5d7acb, thumbnail: c7aff424-4250-4f02-b937-bcb5b3060eba
Getting thumbnail from video: 21af5d7acb, thumbnail: 3d52a895-cf07-470e-ba88-acf22ecfc09b
Getting thumbnail from video: 21af5d7acb, thumbnail: 9d83f10a-97aa-4a2c-a53e-00b5ad4c8347
<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'>
Create and build a person group...
Person group ID: cf24f4e8-18ca-4742-bb81-0564d72a6ae0
Training status: running.
Training status: succeeded.
```

In [ ]:
```
person_id_avkash = get_person_from_video(video_id_avkash)
```
```
Getting video info for: dc76fa5b96
Getting thumbnail from video: dc76fa5b96, thumbnail: f16ec1d9-999f-4e92-8d5b-b50665a2816d
Getting thumbnail from video: dc76fa5b96, thumbnail: 453bf0a0-0f4a-40d6-af90-241bfe3bce09
Getting thumbnail from video: dc76fa5b96, thumbnail: 92191ca6-8379-4cfe-bc84-597b0332a974
Getting thumbnail from video: dc76fa5b96, thumbnail: 895ede83-01e3-47dc-bbee-1061ebe1f803
Getting thumbnail from video: dc76fa5b96, thumbnail: 30df4881-f08d-404e-b6e7-2c94a42ba041
Getting thumbnail from video: dc76fa5b96, thumbnail: 77687546-f343-48b6-ac0d-ee28749c9082
Getting thumbnail from video: dc76fa5b96, thumbnail: d353c46b-c935-4af7-b255-3539893bf40a
<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'>
Create and build a person group...
Person group ID: df3b8ac4-ef0a-4a2b-bfa5-be4e67ed7048
Training status: running.
Training status: succeeded.
```
In [ ]:
```
pascal_id_face_list = get_person_from_id_card(id_card_pascal)
```
```
Face ID: b5f159bf-94bc-427c-9eb2-34b8dd0b0e00
```
In [ ]:
```
avkash_id_face_list = get_person_from_id_card(id_card_avkash)
```
```
Face ID: 7d89cf18-1c40-49ff-a068-7e37c134395a
```
In [ ]:
```
PersonValidation_Pascal = PersonValidation(pascal_id_face_list, person_id_pascal)
```
```
The Identity is the same with confidence 67.242 %
```
In [ ]:
```
PersonValidation_avkash = PersonValidation(avkash_id_face_list, person_id_avkash)
```
```
The Identity is the same with confidence 79.671 %
```
In [ ]:
```
if PersonValidation_Pascal:
    fligth_manifest_df.loc[6, 'PersonValidation'] = 'True'
```
In [ ]:
```
if PersonValidation_avkash:
    fligth_manifest_df.loc[0, 'PersonValidation'] = 'True'
```
In [ ]:
```
# The personal identity check was only carried out for two people, as no more videos were available
```
In [ ]:
```
fligth_manifest_df
```
Out[ ]:

	Flight No	Origin	Destination	Date	Time	First Name	Last Name	Sex	SeatNo	DateofBirth	DoBValidation	PersonValidation	LuggageValidation	NameValidation	BoardingPassValidation
0	234	San Francisco	Chicago	April 20 2022	10:00 AM PST	Avkash	Chauhan	M	20A	01/01/1990	True	True	False	True	True
1	234	San Francisco	Chicago	April 20 2022	10:00 AM PST	James	Jackson	M	25B	10/12/1956	False	False	False	False	False
2	234	San Francisco	Chicago	April 20 2022	10:00 AM PST	Libby	Herold	F	3D	02/10/1996	False	False	False	False	False
3	234	San Francisco	Chicago	April 20 2022	10:00 AM PST	James	Webb	M	1A	12/15/1970	True	False	False	True	True
4	234	San Francisco	Chicago	April 20 2022	10:00 AM PST	Radha	Kumar	F	34B	03/05/1994	False	False	False	False	False
5	234	San Francisco	Chicago	April 20 2022	10:00 AM PST	Sameer	Kumar	M	34A	01/25/1990	False	False	False	False	False
6	289	Stuttgart	New York	December 21 2023	15:00 AM PST	Pascal	Huissel	M	5A	05/26/1996	True	True	False	True	True

In [ ]:
```
def perform_prediction_ligther_detection(test_image_path, model_id, model_iteration_name):
    with open(os.path.join (test_image_path), "rb") as test_image:
        results = prediction_model.detect_image(model_id, model_iteration_name, test_image.read())
        # Display the results.
        for n, prediction in enumerate(results.predictions):
            if prediction.probability > 0.1:
                print("\t" + prediction.tag_name +
                    ": {0:.2f}%".format(prediction.probability * 100))
            elif n == 0:
                 print("\t" + prediction.tag_name +
                    ": {0:.2f}%".format(prediction.probability * 100))
```

In [ ]:
```
publish_name_ligther_model = "detect-lighter-model-V2"
project_id_ligther_model = "2a42a6c3-1158-49a7-a078-a54b7ff4ea8b"
```

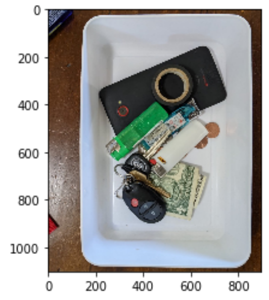
In [ ]:
```
Luggage_image_path = "/workspace/home/ligther_test_images/"

Luggage_image_list = []
for root, dirs, files in os.walk(Luggage_image_path):
    for file in files:
        if file.endswith(".jpg"):
            Luggage_image_list.append(os.path.join(root, file))
```

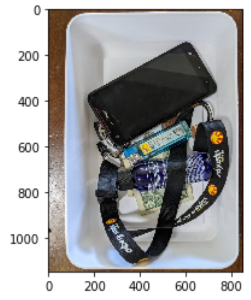
In [ ]:
```
Luggage_image_list
```

Out[ ]:
```
['/workspace/home/ligther_test_images/lighter_test_set_2of5.jpg',
 '/workspace/home/ligther_test_images/lighter_test_set_3of5.jpg',
 '/workspace/home/ligther_test_images/lighter_test_set_1of5.jpg',
 '/workspace/home/ligther_test_images/lighter_test_set_5of5.jpg',
 '/workspace/home/ligther_test_images/lighter_test_set_4of5.jpg']
```

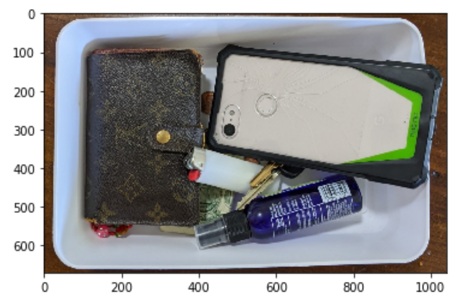
In [ ]:
```
for Luggage_image in Luggage_image_list:
    display_image(Luggage_image)
    perform_prediction_ligther_detection(Luggage_image, project_id_ligther_model, publish_name_ligther_model)
```



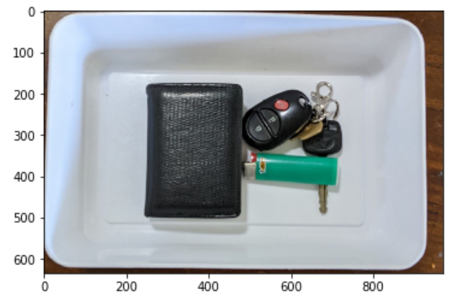
```
	lighter: 31.19%
	lighter: 16.46%
	lighter: 12.89%
```



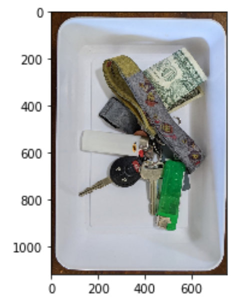
```
	lighter: 6.59%
```



```
lighter: 65.05%
lighter: 26.22%
lighter: 18.40%
```



```
lighter: 14.10%
lighter: 13.18%
lighter: 12.88%
```



```
lighter: 31.70%
lighter: 23.01%
```

In [ ]:
```
# write the dataframe back to loadedcsv file
fligth_manifest_df.to_csv(fligth_manifest_path, index=False)
```

In [ ]:
```
fligth_manifest_df
```

Out[ ]:

	Flight No	Origin	Destination	Date	Time	First Name	Last Name	Sex	SeatNo	DateofBirth	DoBValidation	PersonValidation	LuggageValidation	NameValidation	BoardingPassValidation
0	234	San Francisco	Chicago	April 20 2022	10:00 AM PST	Avkash	Chauhan	M	20A	01/01/1990	True	True	False	True	True
1	234	San Francisco	Chicago	April 20 2022	10:00 AM PST	James	Jackson	M	25B	10/12/1956	False	False	False	False	False
2	234	San Francisco	Chicago	April 20 2022	10:00 AM PST	Libby	Herold	F	3D	02/10/1996	False	False	False	False	False
3	234	San Francisco	Chicago	April 20 2022	10:00 AM PST	James	Webb	M	1A	12/15/1970	True	False	False	True	True
4	234	San Francisco	Chicago	April 20 2022	10:00 AM PST	Radha	Kumar	F	34B	03/05/1994	False	False	False	False	False
5	234	San Francisco	Chicago	April 20 2022	10:00 AM PST	Sameer	Kumar	M	34A	01/25/1990	False	False	False	False	False
6	289	Stuttgart	New York	December 21 2023	15:00 AM PST	Pascal	Huissel	M	5A	05/26/1996	True	True	False	True	True

In [ ]:

```python
def create_message_for_kiosk(passenger_name):
    passenger_first_name = passenger_name.split(" ")[0]
    passenger_last_name = passenger_name.split(" ")[1]
    name_idx = fligth_manifest_df.index[fligth_manifest_df['Last Name']==passenger_last_name].tolist()[0]

    fm_fligth_num = str(fligth_manifest_df._get_value(name_idx, 'Flight No'))
    fm_seat_num = fligth_manifest_df._get_value(name_idx, 'SeatNo')
    fm_origin = fligth_manifest_df._get_value(name_idx, 'Origin')
    fm_destination = fligth_manifest_df._get_value(name_idx, 'Destination')
    fm_time = fligth_manifest_df._get_value(name_idx, 'Time')
    fm_date = fligth_manifest_df._get_value(name_idx, 'Date')

    fm_DoBValidation = fligth_manifest_df._get_value(name_idx, 'DoBValidation')
    fm_PersonValidation = fligth_manifest_df._get_value(name_idx, 'PersonValidation')
    fm_LuggageValidation = fligth_manifest_df._get_value(name_idx, 'LuggageValidation')
    fm_NameValidation = fligth_manifest_df._get_value(name_idx, 'NameValidation')
    fm_BoardingPassValidation = fligth_manifest_df._get_value(name_idx, 'BoardingPassValidation')

    fm_DoBValidation_out = "unsuccessful"
    fm_PersonValidation_out = "unsuccessful"
    fm_LuggageValidation_out = "unsuccessful"
    fm_NameValidation_out = "unsuccessful"
    fm_BoardingPassValidation_out = "unsuccessful"

    if fm_DoBValidation == 'True':
        fm_DoBValidation_out = "successful"

    if fm_PersonValidation == 'True':
        fm_PersonValidation_out = "successful"

    if fm_LuggageValidation == 'True':
        fm_LuggageValidation_out = "successful"

    if fm_NameValidation == 'True':
        fm_NameValidation_out = "successful"

    if fm_BoardingPassValidation == 'True':
        fm_BoardingPassValidation_out = "successful"

    print("Dear " + passenger_first_name + " " + passenger_last_name + ",\n\n" +
            "Thank you for using our Kiosk. We are pleased to inform you that you have completed the check-in process for your flight " +
            fm_fligth_num + " from " + fm_origin + " to " + fm_destination + " on " + fm_date + " at " + fm_time + ".\n" +
            "Your seat number is " + fm_seat_num + ".\n" +
            "The Fligt No. is " + fm_fligth_num + ".\n\n" +
            "Your check-in validation status is as follows:\n" +
            "Name Validation: " + fm_NameValidation_out + ".\n" +
            "Date of Birth Validation: " + fm_DoBValidation_out + ".\n" +
            "Person Validation: " + fm_PersonValidation_out + ".\n" +
            "Luggage Validation: " + fm_LuggageValidation_out + ".\n" +
            "Boarding Pass Validation: " + fm_BoardingPassValidation_out + ".\n\n" +
            "If not all validations were successful, please contact our customer service at the check-in counter.\n\n"
            "We wish you a pleasant flight.\n\n" +
            "Best regards,\n" +
            "The Kiosk Team")
```

In [ ]:
```python
create_message_for_kiosk("Pascal Huissel")
```

```
Dear Pascal Huissel,

Thank you for using our Kiosk. We are pleased to inform you that you have completed the check-in process for your flight 289 from Stuttgart to New York on December 21 2023 at 15:00 AM PST.
Your seat number is 5A.
The Fligt No. is 289.

Your check-in validation status is as follows:
Name Validation: successful.
Date of Birth Validation: successful.
Person Validation: successful.
Luggage Validation: unsuccessful.
Boarding Pass Validation: successful.

If not all validations were successful, please contact our customer service at the check-in counter.

We wish you a pleasant flight.

Best regards,
The Kiosk Team
```

In [ ]:
```python
create_message_for_kiosk("Avkash Chauhan")
```

```
Dear Avkash Chauhan,

Thank you for using our Kiosk. We are pleased to inform you that you have completed the check-in process for your flight 234 from San Francisco to Chicago on April 20 2022 at 10:00 AM PST.
Your seat number is 20A.
The Fligt No. is 234.

Your check-in validation status is as follows:
Name Validation: successful.
Date of Birth Validation: successful.
Person Validation: successful.
Luggage Validation: unsuccessful.
Boarding Pass Validation: successful.

If not all validations were successful, please contact our customer service at the check-in counter.

We wish you a pleasant flight.

Best regards,
The Kiosk Team
```

In [ ]:
```python
create_message_for_kiosk("James Webb")
```

```
Dear James Webb,

Thank you for using our Kiosk. We are pleased to inform you that you have completed the check-in process for your flight 234 from San Francisco to Chicago on April 20 2022 at 10:00 AM PST.
Your seat number is 1A.
The Fligt No. is 234.

Your check-in validation status is as follows:
Name Validation: successful.
Date of Birth Validation: successful.
Person Validation: unsuccessful.
Luggage Validation: unsuccessful.
Boarding Pass Validation: successful.

If not all validations were successful, please contact our customer service at the check-in counter.

We wish you a pleasant flight.

Best regards,
The Kiosk Team
```