17IT005

AIM:

Case study of real-life example using cloud framework/tool/project.

TEAM MEMBERS:

	Student Name: Palak Amin

DESCRIPTION OF THE SCENARIO:

Easy communication at scale, disposable "burner" phones, and simple photo-editing tools— these are just some of the ways the digital, connected world was helping traffickers operate anonymously, one step ahead of the law, especially those who traffic children online.

PROBLEM:

Abusers had figured out how to use advanced technology to facilitate their exploitation of children. A 2012 Thorn survey of child sex-trafficking survivors found that 75 percent had been sold online, most of them through escort websites. One way for law enforcement agents to catch traffickers and rescue kids would be to identify and investigate escort ads that seem to feature minors, but there are many obstacles to doing this. One of the most significant obstacles is the massive volume of these ads. As Thorn started work on a tool that could help investigators identify children being exploited, more than 150,000 new escort ads were being posted online every day in the United States. Problems of scale like this cry out for automated processes, but another big obstacle is the fact that escort ad data is anything but consistent: traffickers use code words, slang, and reused content to complicate detection by machines.

SOLUTION USING CLOUD: (AWS/GCP/AZURE/IBM/ORACLE etc.)

To help investigators overcome the obstacle mentioned above and other hurdles, Thorn collaborated with AWS and Digital Reasoning, a company specializing in cognitive-computing solutions, to build Spotlight. Spotlight's machine-learning models analyze new escort ads in real time and use intelligent image analysis and natural-language processing (NLP) to flag those that match risk profiles developed in cooperation with law enforcement agencies. Officers can set customized alerts and search Spotlight’s constantly growing database of ads to aid in their investigations.

Let us see some of the functionalities of Amazon Rekognition using python and S3 bucket.

First we need to connect aws using boto3 to our python project, the code for the same is:

```
import csv
import boto3


with open('new_user_credentials.csv', 'r') as input:
    next(input)
    reader = csv.reader(input)
    for line in reader:
        access_key_id = line[2]
        secret_access_key = line[3]
```

```
client = boto3.client('rekognition', aws_access_key_id = access_key_id, aws_secret_access_key = secret_access_key, region_name = 'us-east-1')
```

Now, let us see various cases for image recognition.

First let us see object recognition, the image we will be using for the same is:

The code for object identification:

```
photo = 'hot_air_ballon.jpg'
response = client.detect_labels(Image={'S3Object': {'Bucket': 'palak-ac', 'Name': photo}})
```

Output:

```
C:\Users\HP\PycharmProjects\aws\venv\Scripts\python.exe C:/Users
{'Labels': [{'Name': 'Balloon', 'Confidence': 99.44635009765625,
```

```
[{'Name': 'Ball'}]}, {'Name': 'Ball', 'Confidence': 99.44635009765625,
```

Now, another identification provided by aws recognition is of the content contained by the image, such as the moderation for supervision label, to filter the content that is or should be seen by children under supervision or not seen at all.

The code for moderation label:

```
response = client.detect_moderation_labels(Image={'S3Object': {'Bucket': 'palak-ac', 'Name': photo}})
```

Output:

```
C:\Users\HP\PycharmProjects\aws\venv\Scripts\python.exe C:/Users/HP/PycharmProjects/aws/rekognisation.py
{'ModerationLabels': [{'Confidence': 67.67680358886719, 'Name': 'Suggestive', 'ParentName': ''}, {'Confidence': 67.67680358886719, 'Name': 'Female Swimwear Or Underwear', 'ParentName': 'Suggestive

Process finished with exit code 0
```

Now, let us try another method that that is used to trace the expressions or the accessories of any image. The image used is:

The code is:

```
photo = 'smiling_girl.jpg'
response = client.detect_faces(Image={'S3Object': {'Bucket':'palak-ac', 'Name': photo}}, Attributes = ['ALL'])
```

Output:

```
AgeRange': {'Low': 22, 'High': 34}, 'Smile': {'Value': True, 'Confidence': 99.14017486572266}, 'Eyeglasses': {'Value': True, 'Confidence': 99.19371795654297}, 'Sunglasses': {'Value': True, 'Confidence
```

We can see the age, glasses, smile and all the attributes are estimated with level of confidence.

What if we give an image such as below with the same person different epressions? The output can be listed using for loop as:

We can see in the below image the value of smile is true for first index and false for second.

```
C:\Users\HP\PycharmProjects\aws\venv\Scripts\python.exe C:/Users/HP/PycharmProjects/aws/rekognisation.py
{'BoundingBox': {'Width': 0.4116973876953125, 'Height': 0.7690223455429077, 'Left': 0.04011255502700806, 'Top': 0.09249822795391083}, 'AgeRange': {'Low': 22, 'High': 34}, 'Smile': {'Value': True,
==========
{'BoundingBox': {'Width': 0.4111221432685852, 'Height': 0.7555440664291382, 'Left': 0.544808030128479, 'Top': 0.14054429531097412}, 'AgeRange': {'Low': 27, 'High': 43}, 'Smile': {'Value': False, '
==========

Process finished with exit code 0
```

We can also identify a celebrity using methods of the same such as let us try a picture of Linus Torvalds.

```
photo = 'random_guy.jpg'
response = client.recognize_celebrities(Image={'S3Object': {'Bucket':'palak-ac', 'Name': photo}})
```

Output:

```
C:\Users\HP\PycharmProjects\aws\venv\Scripts\python.exe C:/Users/HP/PycharmProjects/aws
{'CelebrityFaces': [{'Urls': ['www.imdb.com/name/nm1127735'], 'Name': 'Linus Torvalds',
```

Now let us see what happens when we give this algorithm a picture with 2 celebrities in it:

```
C:\Users\HP\PycharmProjects\aws\venv\Scripts\python.exe C:/Users/HP/PycharmProjects/aws/rekc
{'Urls': ['www.imdb.com/name/nm1907769'], 'Name': 'Elon Musk', 'Id': '3CI7QV9d', 'Face': {'E
===============
{'Urls': ['www.imdb.com/name/nm3212916'], 'Name': 'Mark Zuckerberg', 'Id': 'mm7At2u', 'Face'
===============

Process finished with exit code 0
```

We can also match faces from one source picture to another target picture, it may be a group image or a solo image such as:

The code for face matching:

```
source_photo = 'priyanka.jpg
target_photo = 'priyanka_group.jpg'
response = client.compare_faces(SourceImage = {'S3Object': {'Bucket':'palak-ac', 'Name': source_photo}},TargetImage = {'S3Object': {'Bucket':'palak-ac', 'Name': target_photo}} )
```

Output:

```
C:\Users\HP\PycharmProjects\aws\venv\Scripts\python.exe C:/Users/HP/PycharmProjects/aws/rekognisation.py
FaceMatches
{'Similarity': 99.98332977294922, 'Face': {'BoundingBox': {'Width': 0.22388336062431335, 'Height': 0.3245522081851959, 'Left': 0.3258302807807924, 'Top': 0.39236968755722046},
UnmatchedFaces
{'BoundingBox': {'Width': 0.2718127369880676, 'Height': 0.44287171959877014, 'Left': 0.0017828397685661912, 'Top': 0.3238799273967743}, 'Confidence': 99.99285888671875, 'Landmar
{'BoundingBox': {'Width': 0.24204567074775696, 'Height': 0.29369401931762695, 'Left': 0.5386145114898682, 'Top': 0.454824835062027}, 'Confidence': 99.99769592285156, 'Landmarks'
{'BoundingBox': {'Width': 0.29593440890312195, 'Height': 0.3342958390712738, 'Left': 0.17400573194026947, 'Top': 0.03448854014277458}, 'Confidence': 99.99279022216797, 'Landmark

Process finished with exit code 0
```

We can see 1 image is matched with the source photo, to see which one is matched we can check the left, right, x-coordinate and y-coordinate of the matched image.

We can also use these methods to identify text from an image which is very useful in getting a date or number plate of any vehicle from the image. For example, consider the following image and used code for text identification.

```
photo = 'quote.jpg'
response = client.detect_text(Image={'S3Object': {'Bucket':'palak-ac', 'Name': photo}})
```

Output:

```
C:\Users\HP\PycharmProjects\aws\venv\Scripts\python.exe C:/Users/HP/PycharmProjects/aws/rekognisation.py
{'TextDetections': [{'DetectedText': 'Animals share with us the', 'Type': 'LINE', 'Id': 0, 'Confidence': 98.10678100585938,
 'DetectedText': 'privilege of having a soul.', 'Type': 'LINE', 'Id': 1, 'Confidence': 99.27879333496094,
{'DetectedText': '-Pythagoras', 'Type': 'LINE', 'Id': 2, 'Confidence': 99.1493911743164,
```

Every line is correctly identified as we can see through the pictures.

Also, we have uploaded every picture in the S3 bucket and are accessing it through the bucket as S3 object, where bucket name is palak-ac and region is the same as our client. Also, the bucket is publicly accessible.

LIST OF CLOUD SERVICES USED:

Services used: S3 Bucket

Policies used: AmazonS3FullAccess, AmazonRekognitionFullAccess

LEARNING OUTCOME:

Note that the images accepted by these methods or any other aws recognition methods are only with the extension .jpg and .png. Along with this many famous companies are already using this feature of aws such as Thorn for their application Spotlight, which helps in law enforcement identify child-trafficking victims faster, Aella Credit empowers underbanked individuals by using Amazon Rekognition for identity verification, Marinus Analytics fights human trafficking using Amazon Rekognition and many more.

REFERENCE:

1. https://aws.amazon.com/rekognition/the-facts-on-facial-recognition-with-artificial-intelligence/
2. https://aws.amazon.com/getting-started/hands-on/detect-analyze-compare-faces-rekognition/
3. https://docs.aws.amazon.com/rekognition/latest/dg/what-is.html
4. https://docs.aws.amazon.com/s3/index.html