

# “AWS Rekognition Video Analysis to Solve a Security Problem”

by

Muhammad Maaz Khan

Abdallah Abdelwahab Mohammed

Seeta Kumari

Nnenna Diana

Spring 2020

May 6<sup>th</sup>, 2020

# Problem

- What is the problem that the tool aims to solve (top-level and sub-problem)?
  - Reduce human error as much as possible in the context of digital surveillance.
  - Humans not that great at monitoring for rare events across multiple video streams, might be hours long.
  - A 2014 study done by Harvard found frequent false negative errors in visual screening tasks
- Why it is a problem?



Manually intensive



Slow and error-prone



Expensive

# Problem

- How severe is the problem?
  - Moderately severe.
  - But if solved can have high positive impact on video surveillance.
- How do you want to address the problem?
  - To reduce human efforts by using Artificial Intelligence (Amazon AWS Rekognition).
  - Transcribe videos to text autonomously and analyze video by searching the transcribed text.



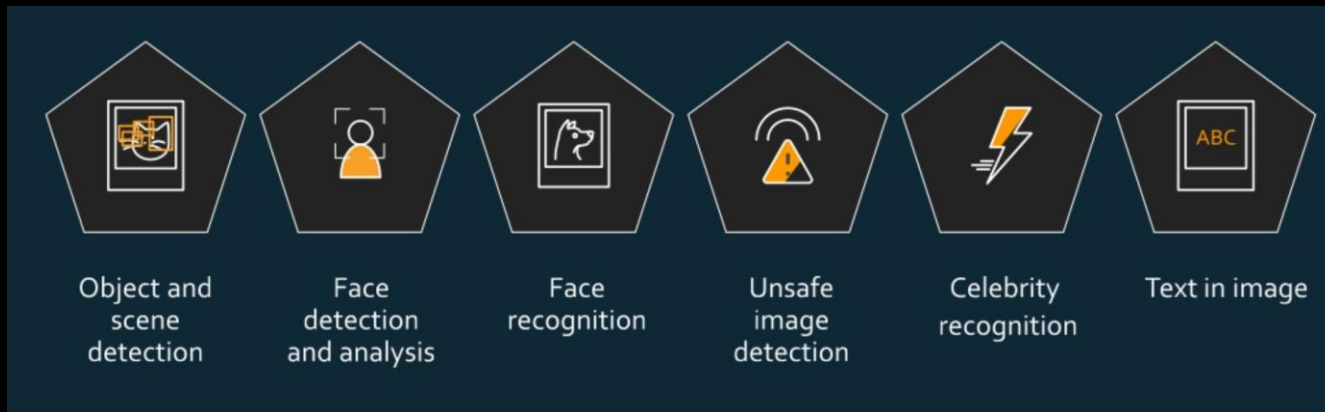
amazon Rekognition

# Problem

- Goal?
  - To reduce human efforts by using Artificial Intelligence (Amazon AWS Rekognition).
  - Transcribe videos to text autonomously and analyze video by searching the transcribed text.

# Background

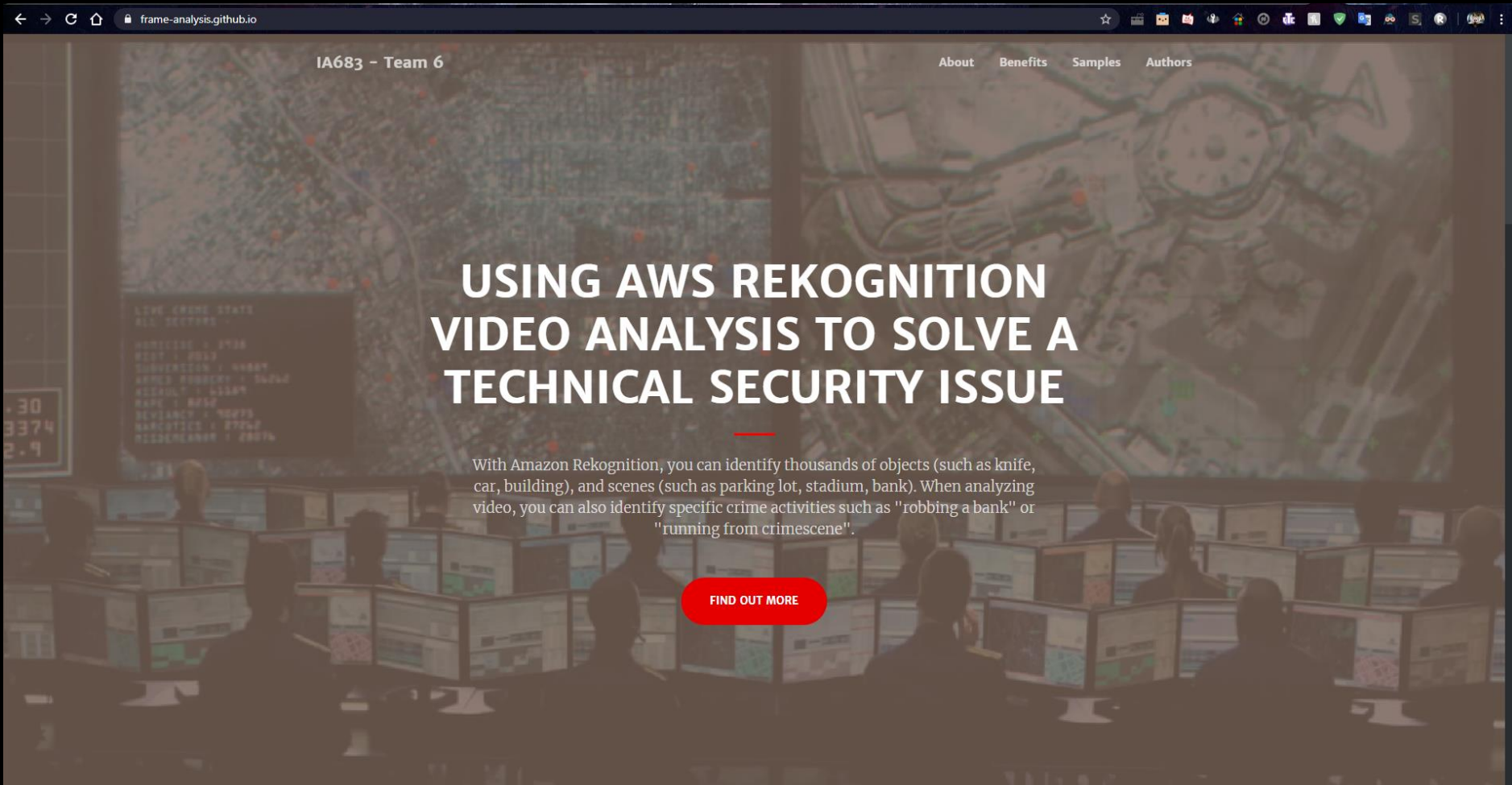
- What must we know about in order to understand the tool?
  - Amazon Rekognition makes it easy to add image and video analysis to your applications using deep learning technology.
  - With Amazon Rekognition, you can identify:



- Can detect any inappropriate content.
- Provides highly accurate facial analysis and facial search capabilities.

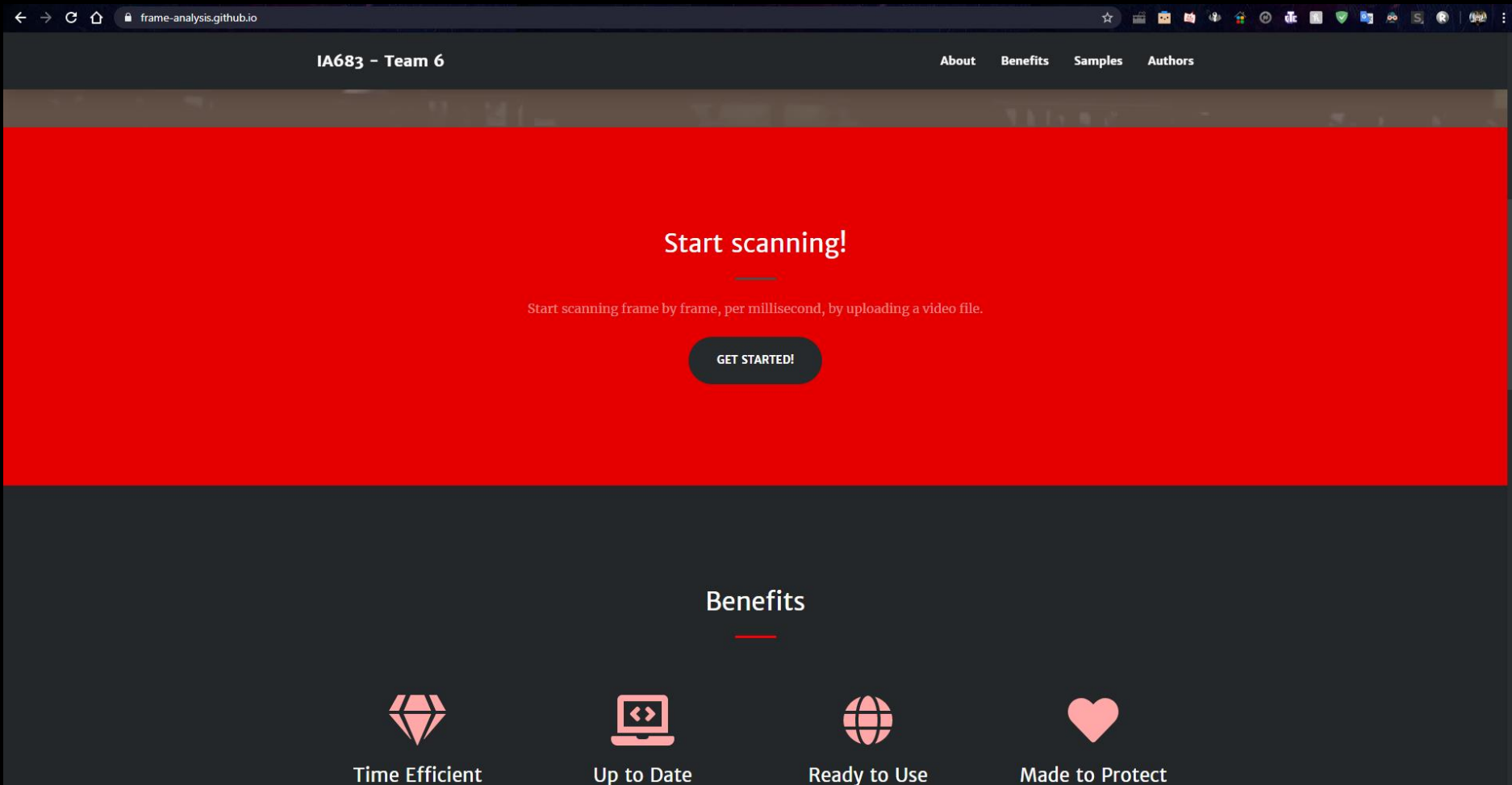
# Tool

- What is the tool like?



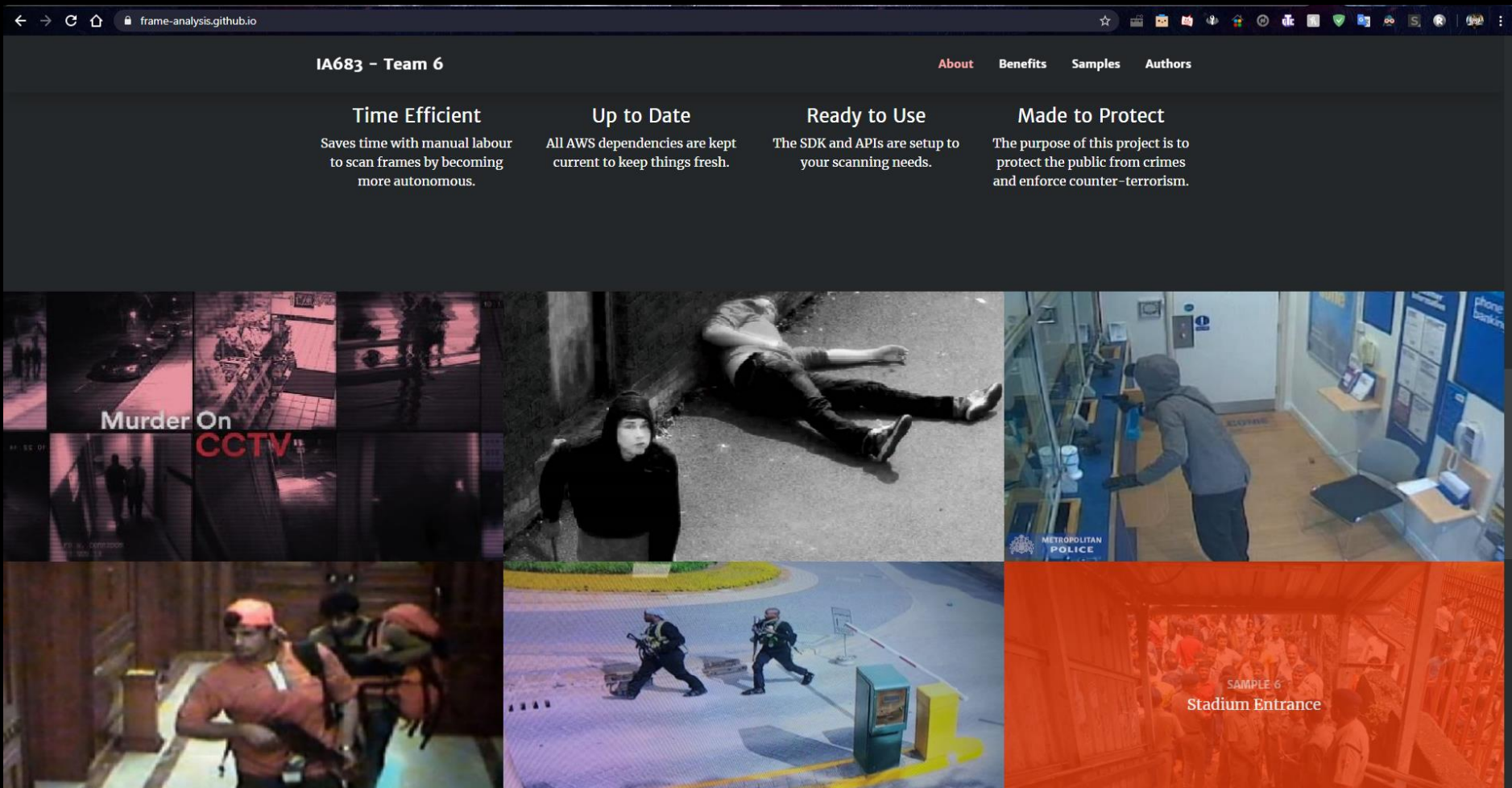
# Tool

- What is the tool like?



# Tool

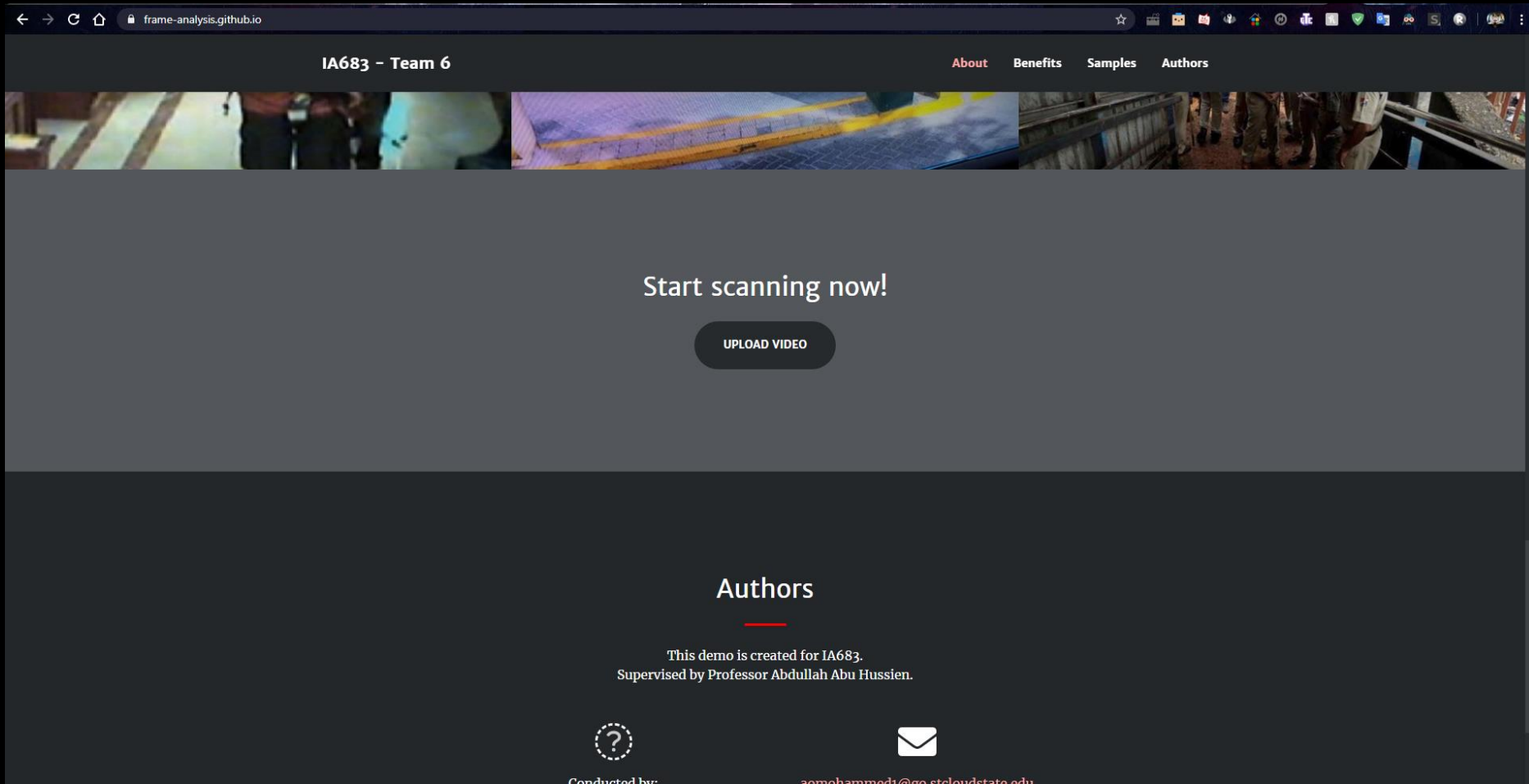
- What is the tool like?





# Tool

- What is the tool like?



# Tool

- How does it work?

```
def StartTextDetection(self):
    response=self.rek.start_text_detection(Video={'S3Object': {'Bucket': self.bucket, 'Name': self.video}},
        NotificationChannel={'RoleArn': self.roleArn, 'SNSTopicArn': self.snsTopicArn})

    self.startJobId=response['JobId']
    print('Start Job Id: ' + self.startJobId)

def GetTextDetectionResults(self):
    maxResults = 10
    paginationToken = ''
    finished = False

    while finished == False:
        response = self.rek.get_text_detection(JobId=self.startJobId,
                                                MaxResults=maxResults,
                                                NextToken=paginationToken)

        print('Codec: ' + response['VideoMetadata']['Codec'])
        print('Duration: ' + str(response['VideoMetadata']['DurationMillis']))
        print('Format: ' + response['VideoMetadata']['Format'])
        print('Frame rate: ' + str(response['VideoMetadata']['FrameRate']))
        print()

        for textDetection in response['TextDetections']:
            text=textDetection['TextDetection']

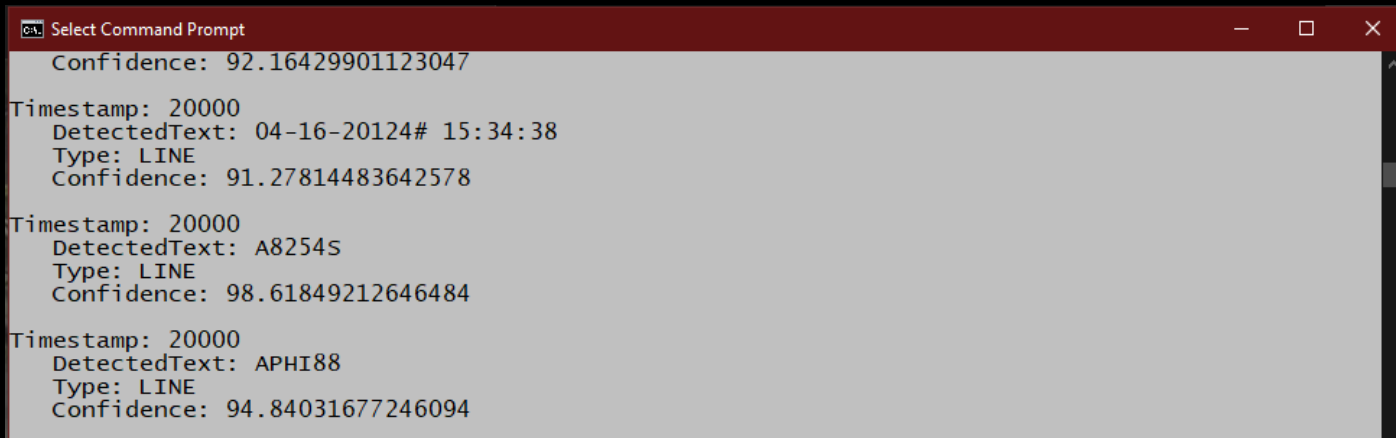
            print("Timestamp: " + str(textDetection['Timestamp']))
            print("    DetectedText: " + text['DetectedText'])
            print("    Type: " + text['Type'])
            print("    Confidence: " + str(text['Confidence']))
```

# Tool

- How does it work?

```
D:\Desktop - WD\Desktop stuff\University\SCSU\2nd Sem\IA_683\Final Project>python videoanalysis.py
Start Job Id: e9eaae8af82e7ac4b50c5957fa673617218f81bc276d59cd1ec6cc3b84de285e
.....e9eaae8af82e7ac4b50c5957fa673617218f81bc276d59cd1ec6cc3b84de285e
SUCCEEDED
Matching Job Found:e9eaae8af82e7ac4b50c5957fa673617218f81bc276d59cd1ec6cc3b84de285e
Codec: h264
Duration: 29960
Format: QuickTime / MOV
Frame rate: 25.0
```

Shows the video has been analyzed successfully. (Figure above)



```
Select Command Prompt
Confidence: 92.16429901123047
Timestamp: 20000
DetectedText: 04-16-20124# 15:34:38
Type: LINE
Confidence: 91.27814483642578
Timestamp: 20000
DetectedText: A8254s
Type: LINE
Confidence: 98.61849212646484
Timestamp: 20000
DetectedText: APHI88
Type: LINE
Confidence: 94.84031677246094
```

Text detected along with the timestamp. in this case at 20 seconds into the video.

# Tool

- How does it work?



Proof that the results obtained are accurate. Video is stopped at 20 seconds to show that the text detected is in fact visible in the video at the 20 second mark.

# Tool

- How does it work?





ia683finalproject

Overview Properties Permissions Management Access points

🔍 Type a prefix and press Enter to search. Press ESC to clear.

Upload Create folder Download Actions ▾ US East (N. Virginia) ↻

Viewing 1 to 4

<input type="checkbox"/>	Name ▾	Last modified ▾	Size ▾	Storage class ▾
<input type="checkbox"/>	 TestVideo.mp4	May 5, 2020 5:10:49 PM GMT-0500	9.1 MB	Standard
<input type="checkbox"/>	 motorway.jpg	May 5, 2020 3:15:18 PM GMT-0500	66.2 KB	Standard
<input type="checkbox"/>	 numberplate.jpg	May 5, 2020 3:41:26 PM GMT-0500	287.3 KB	Standard
<input type="checkbox"/>	 videosec20.PNG	May 5, 2020 4:01:56 PM GMT-0500	990.4 KB	Standard

Viewing 1 to 4

AWS S3 bucket used to store the video from where AWS Rekognition picks up the video to be analyzed. This is where the videos would be uploaded to.

# Tool

- From a user perspective? What key design decisions did the authors make and the rationale for those decisions?
  - In order for the tool to be easy-to-use and user-friendly, we created a front-end web application that is currently accessible to the public.
  - Hide unnecessary elements (e.g. BoundingBox and Polygons) in order to make it read for the user to read from the output.

```
#print("    Geometry:")
# for geometry in text['Geometry']:
#     print ("        Bounding box")
#     print ("            Top: " + str(geometry['BoundingBox']['Top']))
#     print ("            Left: " + str(geometry['BoundingBox']['Left']))
#     print ("            Width: " + str(geometry['BoundingBox']['Width']))
#     print ("            Height: " + str(geometry['BoundingBox']['Height']))
#     print ("        Polygon")
#     print ("            X: " + str(geometry['Polygon']['X']))
#     print ("            Y: " + str(geometry['Polygon']['Y']))
print()
```

# Evaluation Method

- How did you evaluate your project? (tool/outcomes)

- Method 1

Research a security issue that could have been better solved with this tool. Work on existing data and see if we can get the correct outcomes out of it.

- Method 2

Create a scenario, Record an activity out of it and try to resolve the issue.

# Evaluation Results

- What results and/or findings came out of the evaluation?
  - The size of giving people's face recognition is only 4096.
  - Storing size is only 20 million faces in the database.
  - The quality of a video also makes the results suffer. if a low pixel is inputted. it might not generate the desired results. making it difficult to work in developing countries.
- Do the results support the authors' claims?
  - The authors' claims were that it was low cost, which is true. That it was integrated with other AWS services, which is true as well.



# Discussion

- Did the authors discuss any interesting insights or implications for design based on their work?
  - Using Amazon's Web Services has its perks. Using the information gathered from their sources and applying the tool we have come to these interesting insights and benefits of using Rekognition Video Analysis:



## Time Efficient

Saves time with manual labour to scan frames by becoming more autonomous.



## Up to Date

All AWS dependencies are kept current to keep things fresh.



## Ready to Use

The SDK and APIs are setup to your scanning needs.



## Made to Protect

The purpose of this project is to protect the public from crimes and enforce counter-terrorism.

# Discussion

- Future work?
  - The team would like to carry this project further and make the tool fully-functional through our web application
    - Found here: <https://frame-analysis.github.io/>
  - With the help of Amazon, Inc. have the tool become open source for Government security sectors.
  - Implement a Browser-Based Upload using HTTPS POST
    - Found here:  
<https://docs.aws.amazon.com/AmazonS3/latest/API/sigv4-post-example.html>
- Open research areas?
  - Attain real-time responses through live footage, e.g. CCTV.

# Claims/Conclusion

- What key claims did the authors make about their tool?
  - Amazon Rekognition claims that it is easy to add image and video analysis to your applications using proven, highly scalable, deep learning technology
    - That requires no machine learning expertise to use.
  - Be able to identify objects, people, text, scenes, and activities in images and videos, as well as detect any inappropriate content.
  - Provides highly accurate facial analysis and facial search capabilities that you can use to detect, analyze, and compare faces for a wide variety of user verification, people counting, and public safety use cases.

# Claims/Conclusion

- What key findings supported those claims? What, if any, other key contributions did they make?
  - Using their SDK, in Python, we were successfully able to recreate the tool and specify which API and feature we wanted to use and implement.