IoT and Facial Recognition at Scale:

Using Amazon's DeepLens to search for matches from the US's Missing Persons Database

—

Andrew Carlson, Vicki Foss, Gerard Kelly, Michelle Liu

August 7, 2018

Background

Why we are doing this

In the United States alone, over 600,000 individuals go missing each year.

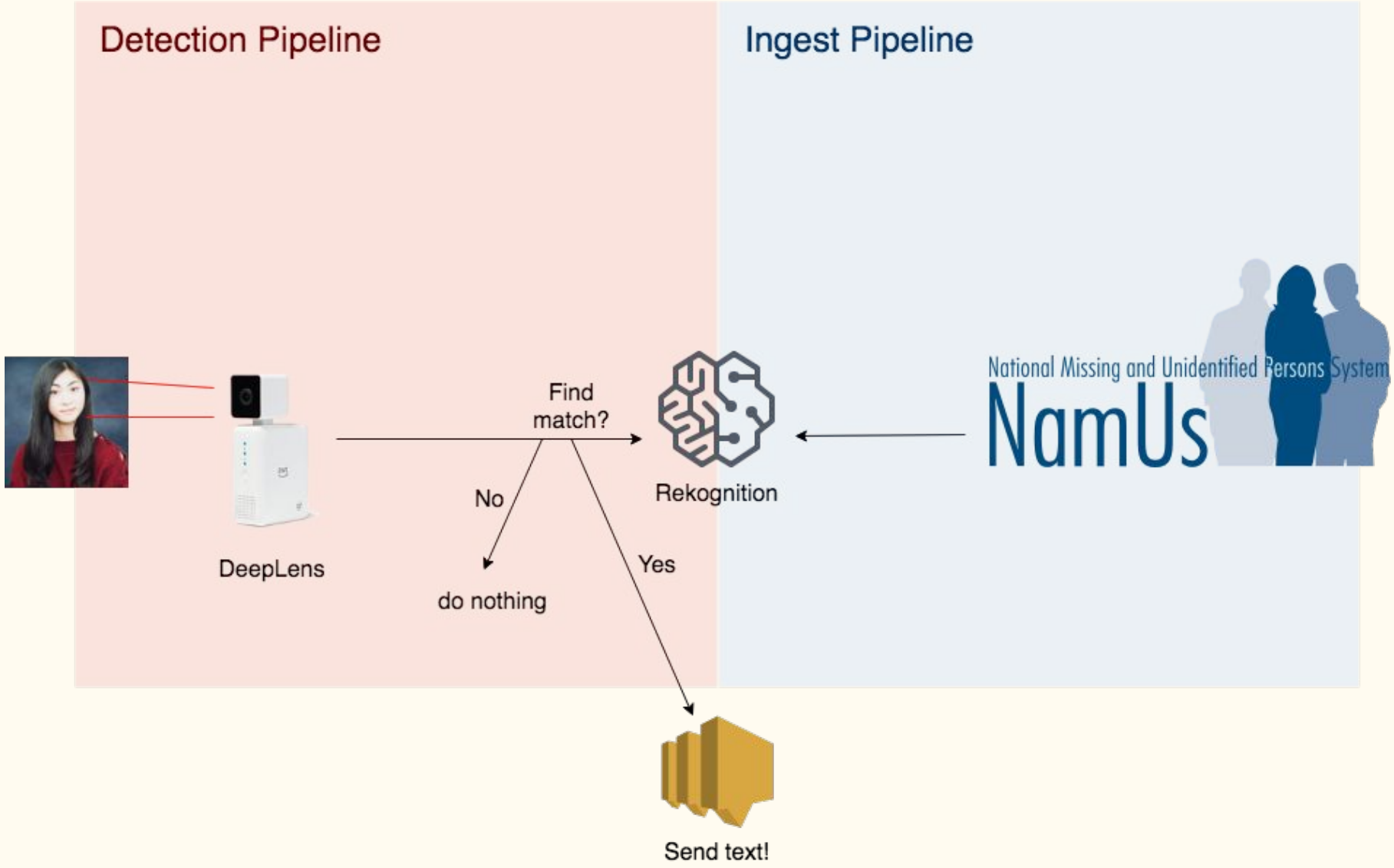
The general public is not very reliable for being able to recognize missing persons, nor for reporting a sighting to the correct authorities if they *do* recognize someone as a missing person.

There is a race and gender disparity in the missing persons cases that receive the most media attention.

Technology can be a better partner than the general public in the search for missing persons.

- Computers surpass humans in facial recognition
- Deep learning models can be trained to identify hundreds of thousands of faces
- Ability to alert authorities automatically in real-time
- Cloud computing makes such a system scalable

Technology Stack


Detection Pipeline
Ingest Pipeline
National Missing and Unidentified Persons System
NamUs
Find
match?
No
Yes
Rekognition
DeepLens
do nothing
Send text!

Tools

Amazon Web Services

 S3 - object storage

 EC2 - elastic compute

 Lambda - event-driven functions

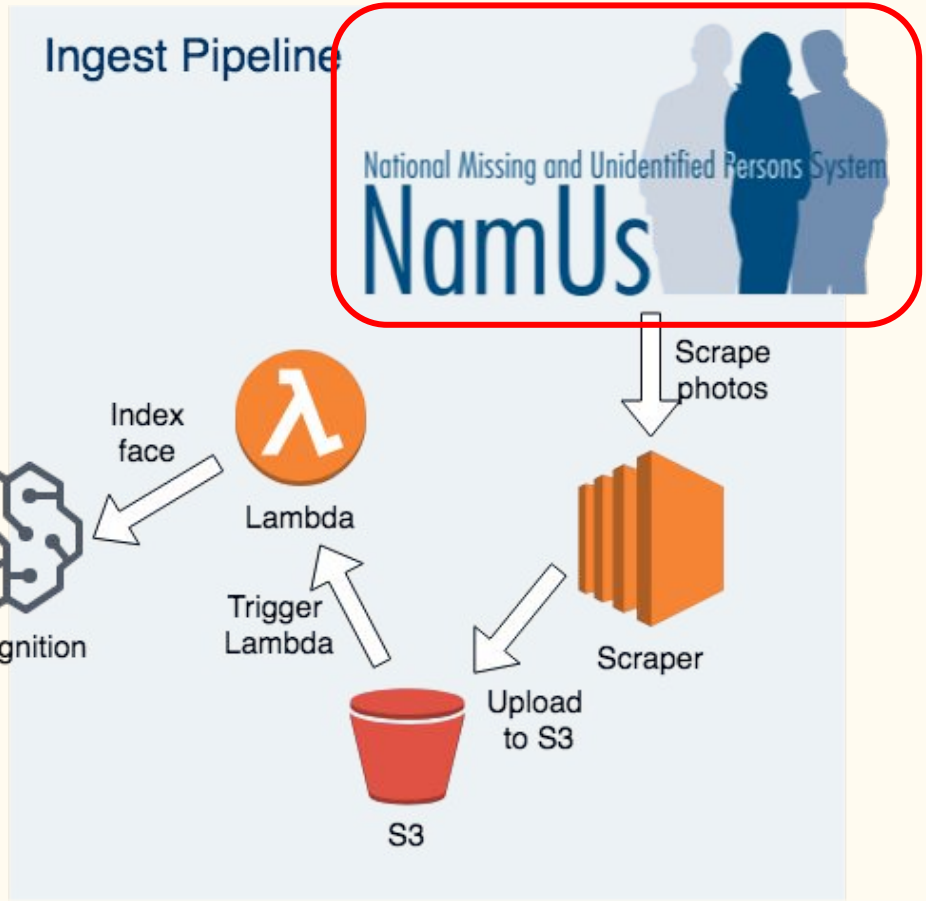
 DynamoDB - document database

 Rekognition - facial recognition API

 SNS - notification service

 IAM Roles - credentials & permissions

 DeepLens - deep learning enabled camera


Ingest Pipeline
National Missing and Unidentified Persons System
NamUs
Scrape photos
Index face
Lambda
Rekognition
Trigger Lambda
Scraper
Upload to S3
S3

Missing Person / NamUs #MP51557

Suzanna Rodriguez, Female, Hispanic / Latino

Date of Last Contact	Missing From	Missing Age	Current Age
August 2, 2018	Seadrift, Texas	25 Years	25 Years

Case Information

Demographics

Missing Age
25 Years

Current Age
25 Years

First Name
Suzanna

Middle Name
--

Last Name
Rodriguez

Nickname/Alias
--

Sex
Female

Race / Ethnicity
Hispanic / Latino

Height
5' 5" (65 Inches)

Weight
140 lbs

Circumstances

Date of Last Contact
August 2, 2018

NamUs Case Created
August 3, 2018

Last Known Location Map

Location
Seadrift, Texas

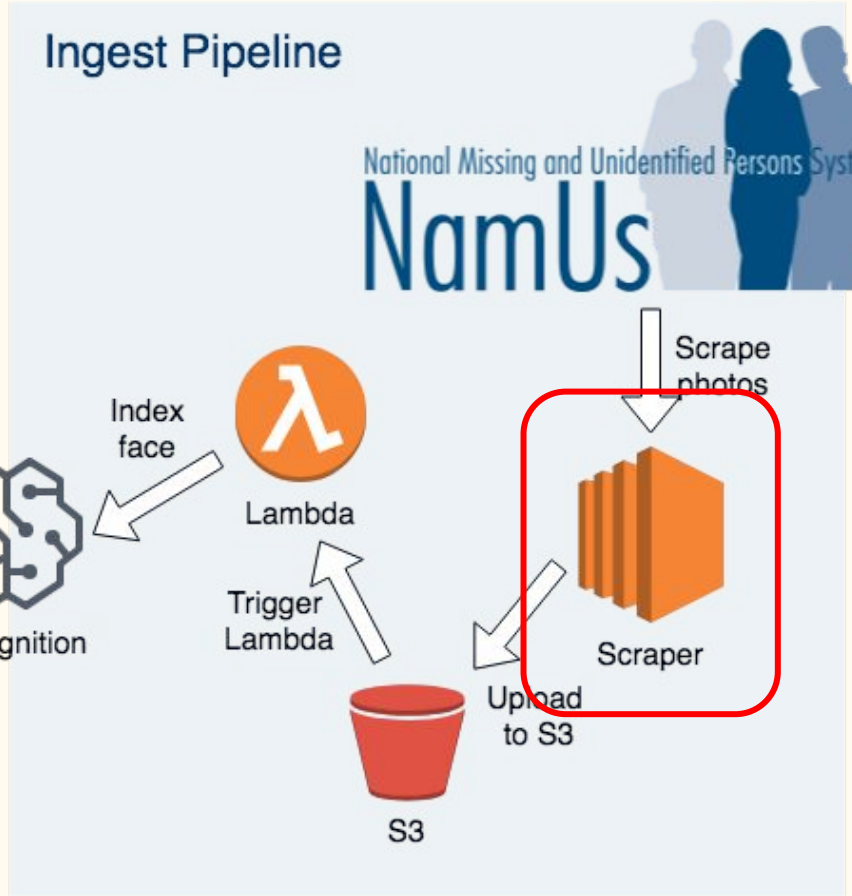
County
Calhoun County

Circumstances of Disappearance
Last seen on August 2, 2018.

```
curl 'https://www.namus.gov/api/CaseSets/NamUs/MissingPersons/Cases/51557' | \
  jq '.images[].files.original'
```

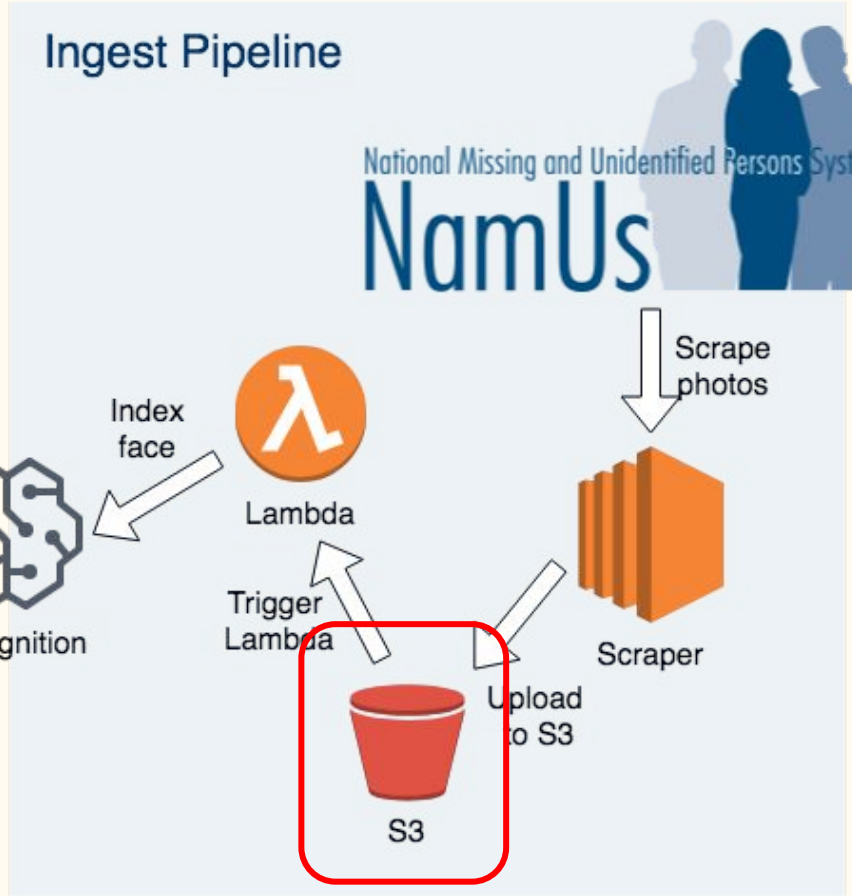

```
{
  "mimeType": "image/pjpeg",
  "height": 890,
  "width": 680,
  "href":
"/api/CaseSets/NamUs/MissingPersons/Cases/51557/Images/83699/Original"
}
```


```
curl -O
'https://www.namus.gov/api/CaseSets/NamUs/MissingPersons/Cases/51557/Images/83
699/Original'
```


Ingest Pipeline
National Missing and Unidentified Persons System
NamUs
Scrape photos
Index face
Lambda
Rekognition
Trigger Lambda
Scraper
Upload to S3
S3

```python
import requests, boto3
s3 = boto3.resource("s3")
bucket = s3.Bucket("missingpeopledb")

for id in ids_to_download:
    # first get all metadata, including image URLs
    metadata = requests.get(make_case_url(id), verify = False).json()

    for ind, image in enumerate(metadata['images']):
        # stream the object directly into a file
        image_resp = requests.get(make_image_url(image), verify = False, stream = True)
        image_filename = str(id) + '-' + str(ind) +'.jpg'
        with open('namus/' +  image_filename, 'wb') as image_file:
            for chunk in image_resp:
                image_file.write(chunk)

        # upload the image to our S3 bucket
        bucket.upload_file('namus/' + image_filename, image_filename)
```

* error-handling and bookkeeping code elided for clarity


Ingest Pipeline
National Missing and Unidentified Persons System
NamUs
Scrape photos
Index face
Lambda
Rekognition
Trigger Lambda
Scraper
Upload to S3
S3

Amazon S3 > missingpeopledb

Overview | Properties | Permissions | Management

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

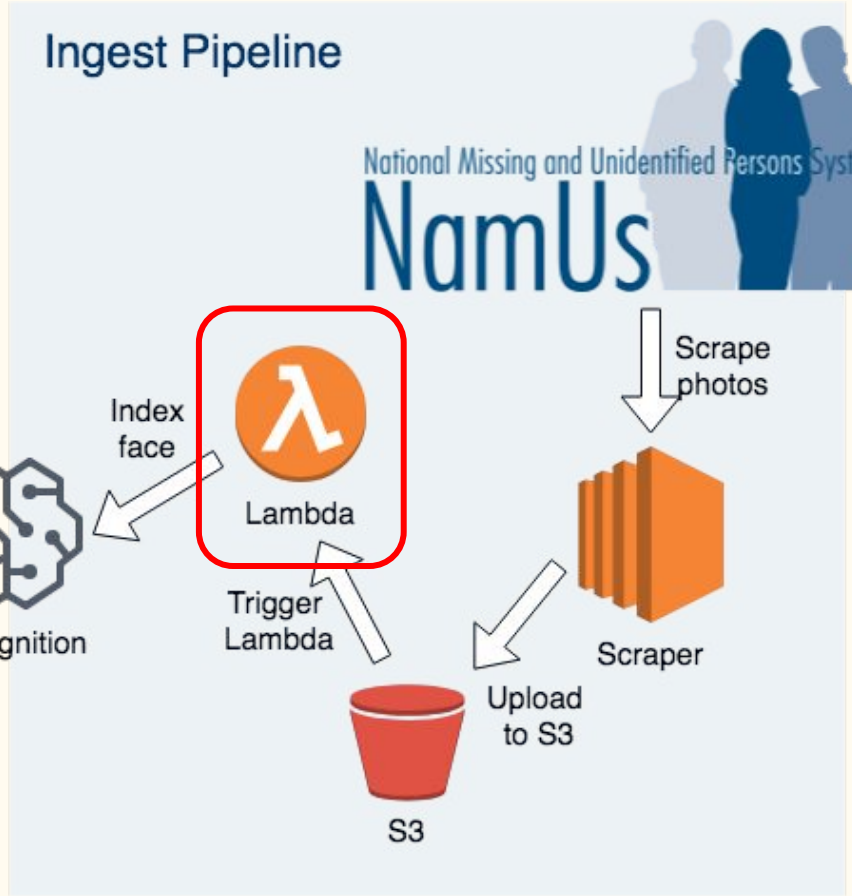
Upload | + Create folder | More | US East (N. Virginia)

Viewing 1 to 300 >

Name	Last modified	Size	Storage class
1-0.jpg	Jul 13, 2018 11:38:59 PM GMT-0700	32.0 KB	Standard
1-1.jpg	Jul 13, 2018 11:39:00 PM GMT-0700	40.0 KB	Standard
1-2.jpg	Jul 13, 2018 11:39:01 PM GMT-0700	40.0 KB	Standard
1-3.jpg	Jul 13, 2018 11:39:03 PM GMT-0700	48.0 KB	Standard
10-0.jpg	Jul 13, 2018 11:39:18 PM GMT-0700	24.0 KB	Standard
10-1.jpg	Jul 13, 2018 11:39:19 PM GMT-0700	32.0 KB	Standard
100-0.jpg	Jul 13, 2018 11:42:30 PM GMT-0700	32.0 KB	Standard
10000-0.jpg	Jul 14, 2018 9:24:51 PM GMT-0700	32.0 KB	Standard

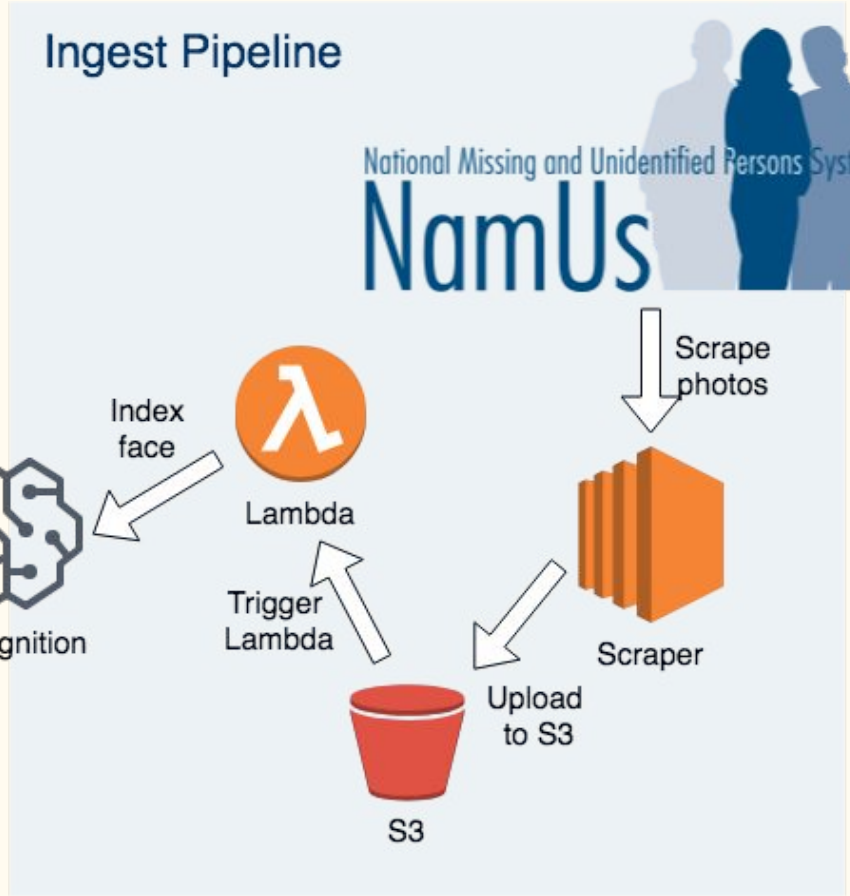
NamUs image stats

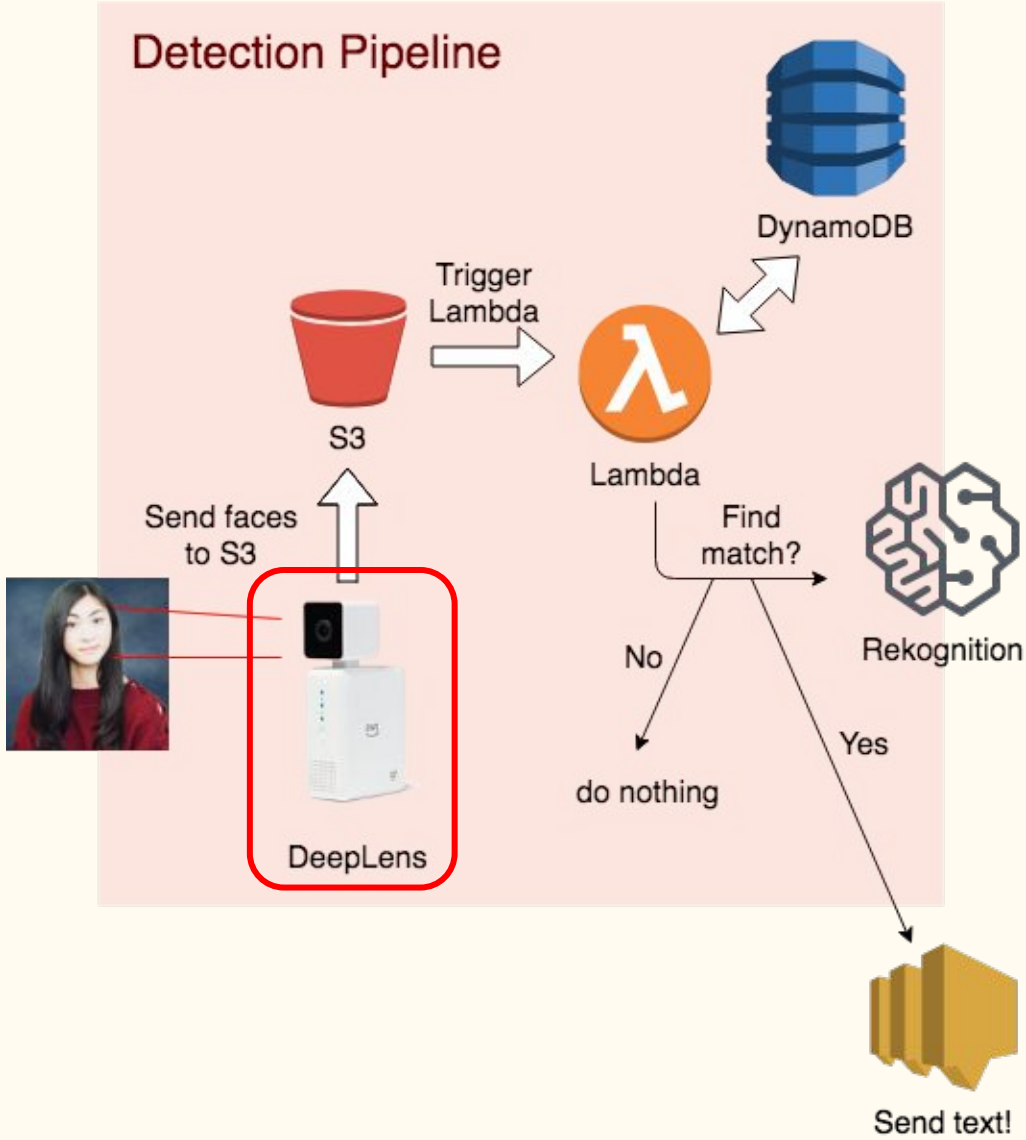
- 14,444 missing people
- 12,294 of those records contain at least one image
- Average of about two images per person, resulting in 24,620 images


Ingest Pipeline
National Missing and Unidentified Persons System
NamUs
Scrape photos
Index face
Lambda
Rekognition
Trigger Lambda
Scraper
Upload to S3
S3

```python
import boto3, urllib
rekognition = boto3.client("rekognition")

def handler(event, context):
    s3_info = event["Records"][0]["s3"]

    # Get the object from the event
    bucket = s3_info["bucket"]["name"]
    key = urllib.unquote_plus(s3_info["object"]["key"].encode("utf8"))

    # tells rekognition to index the new image
    results = rekognition.index_faces(
        CollectionId="missingpeople",
        Image={
            'S3Object': {
                'Bucket': bucket,
                'Name': key
            }
        },
        ExternalImageId=external_id,
        DetectionAttributes=['ALL']
    )
    print(results)
```

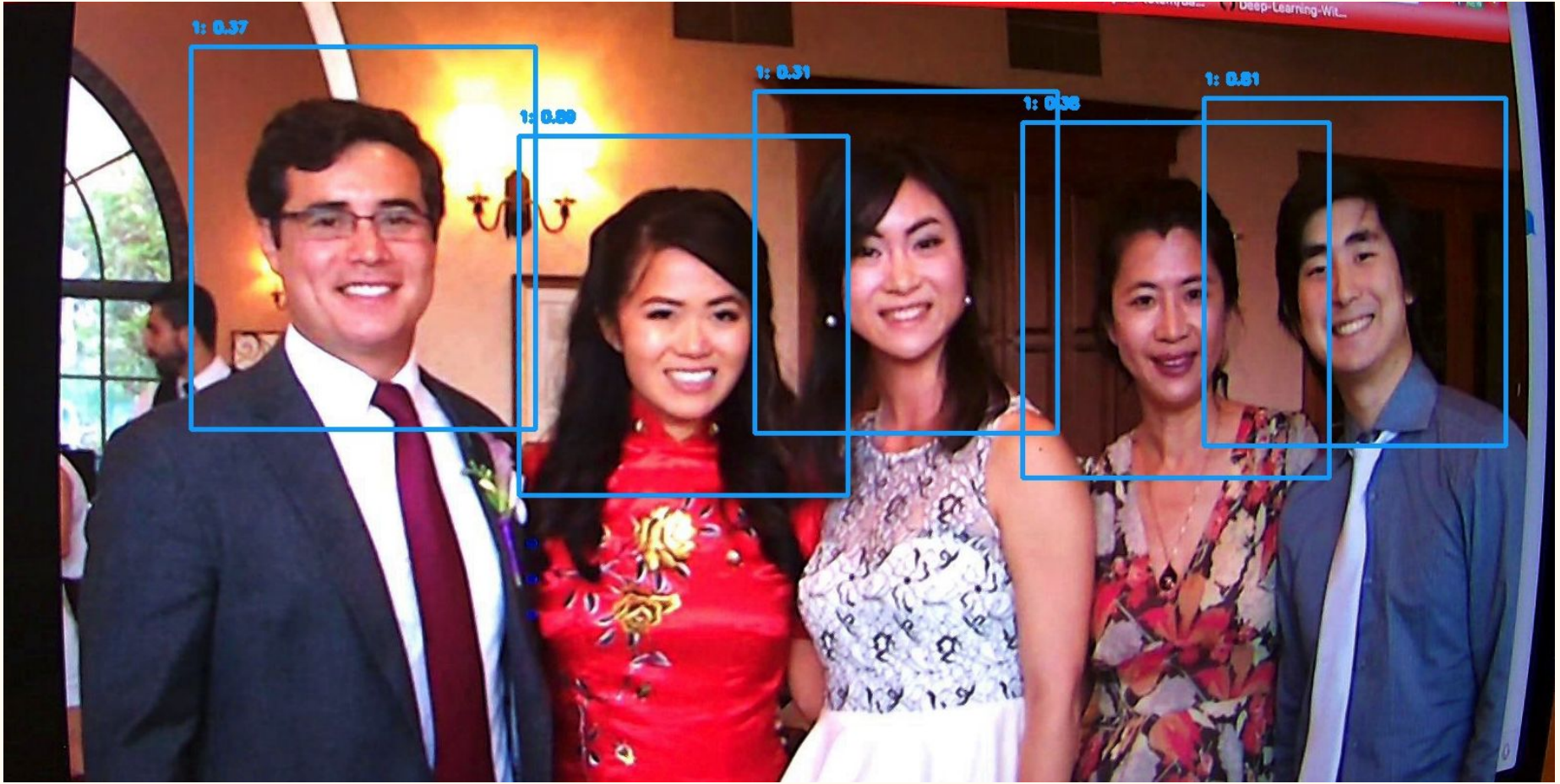
* error-handling and validation code elided for clarity


Ingest Pipeline
National Missing and Unidentified Persons System
NamUs
Scrape photos
Index face
Lambda
Rekognition
Trigger Lambda
Scraper
Upload to S3
S3

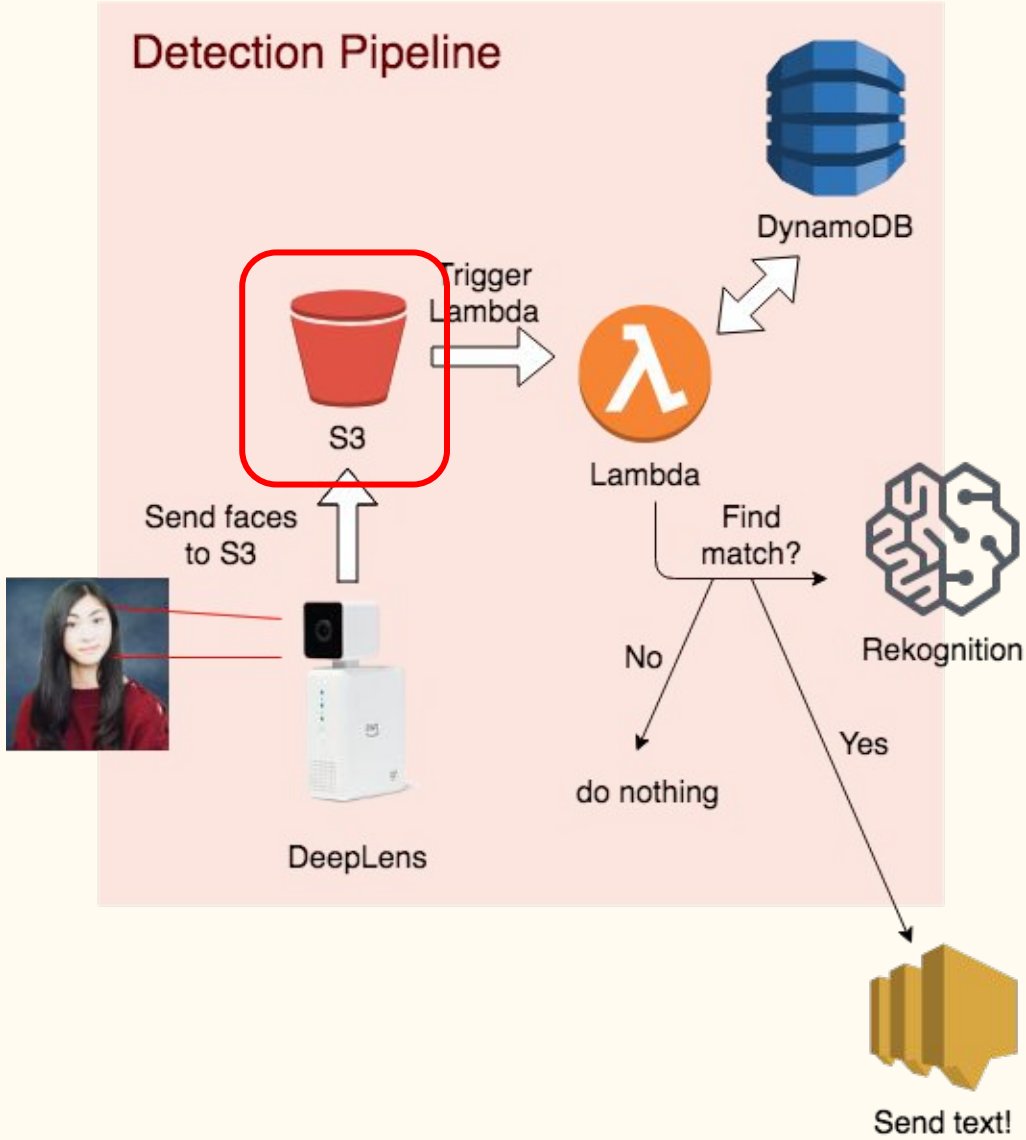

Detection Pipeline
DynamoDB
Trigger
Lambda
S3
Lambda
Send faces
to S3
Find
match?
Rekognition
No
Yes
do nothing
DeepLens
Send text!

DeepLens

- Wireless deep learning enabled video camera
- Easy to deploy either a pre-trained or custom deep learning models that can detect faces, objects, activities, texts, etc.
- Lambda functions - when it detects human faces, the project stream containing the frame is extracted and the image of the face is cropped out and upload to a S3 bucket



Example of project output stream


Detection Pipeline
DynamoDB
Trigger Lambda
S3
Lambda
Send faces to S3
Find match?
Rekognition
No
Yes
do nothing
DeepLens
Send text!

Amazon S3 > deeplensfacedetected

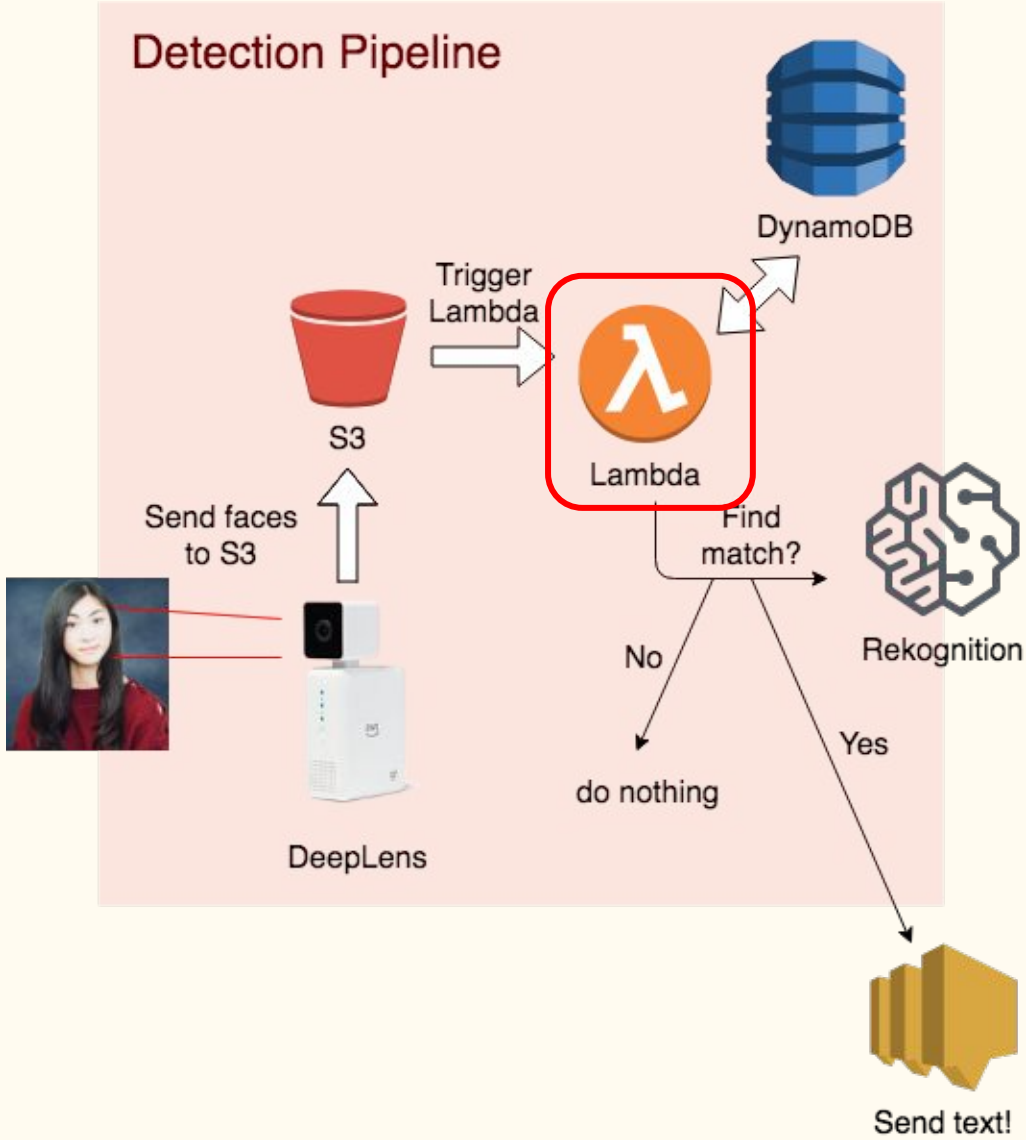
Overview | Properties | Permissions | Management

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

Upload | + Create folder | More | US East (N. Virginia)

Viewing 1 to 28

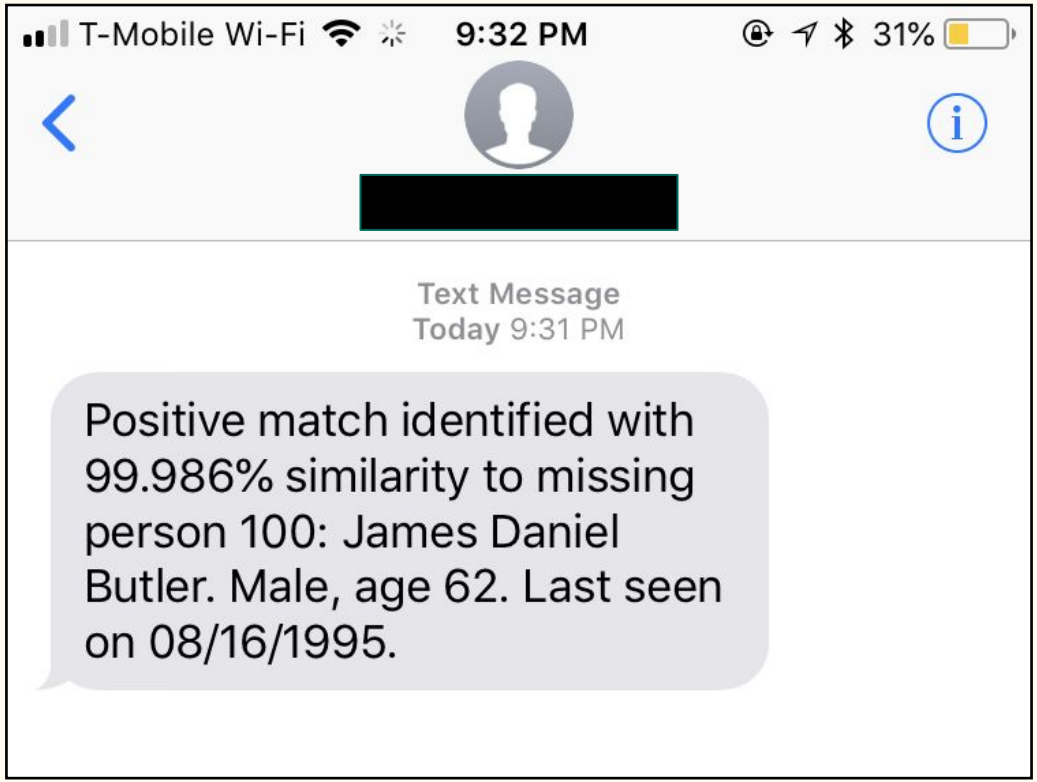
Name	Last modified	Size	Storage class
image-20180719-052026.jpg	Jul 18, 2018 10:20:27 PM GMT-0700	31.3 KB	Standard
image-20180719-052028.jpg	Jul 18, 2018 10:20:30 PM GMT-0700	25.3 KB	Standard
image-20180719-052356.jpg	Jul 18, 2018 10:23:57 PM GMT-0700	28.8 KB	Standard
image-20180719-052357.jpg	Jul 18, 2018 10:23:58 PM GMT-0700	28.8 KB	Standard
image-20180719-052358.jpg	Jul 18, 2018 10:23:59 PM GMT-0700	28.1 KB	Standard
image-20180719-052400.jpg	Jul 18, 2018 10:24:01 PM GMT-0700	25.6 KB	Standard
image-20180719-052505.jpg	Jul 18, 2018 10:25:07 PM GMT-0700	28.6 KB	Standard
image-20180719-052508.jpg	Jul 18, 2018 10:25:09 PM GMT-0700	26.8 KB	Standard
image-20180719-052509.jpg	Jul 18, 2018 10:25:10 PM GMT-0700	27.3 KB	Standard


Detection Pipeline
DynamoDB
Trigger
Lambda
S3
Lambda
Send faces
to S3
Find
match?
Rekognition
No
Yes
do nothing
DeepLens
Send text!

```python
import boto3, urllib, datetime
dynamodb = boto3.client('dynamodb')
sns = boto3.client('sns')
rekognition = boto3.client('rekognition')

def lambda_handler(event, context):
    # Get the object from the event
    bucket = event['Records'][0]['s3']['bucket']['name']
    key = urllib.unquote_plus(event['Records'][0]['s3']['object']['key'].encode('utf8'))

    result = rekognition.search_faces_by_image(
        CollectionId="missingpeople",
        Image={
            'S3Object': {
                'Bucket': bucket,
                'Name': key
            }
        },
        MaxFaces=1
    )

    # send text if face matches a missing person, and if we haven't texted recently
    if result['FaceMatches']:
        externalid = int(result['FaceMatches'][0]['Face']['ExternalImageId'])
        should_send_text = check_dynamo_if_texted_recently(dynamodb, externalid)
        if should_send_text:
            update_dynamo(dynamodb, externalid)
            msg = create_text_msg(gxternalid)
            sns.publish(PhoneNumber='+16262263799', Message=msg)
```

* error-handling and data-munging code elided for clarity

It’s a match!


T-Mobile Wi-Fi
9:32 PM
31%
[redacted]
Text Message
Today 9:31 PM
Positive match identified with 99.986% similarity to missing person 100: James Daniel Butler. Male, age 62. Last seen on 08/16/1995.

Demo!

Results

- It works!
- End-to-end, it takes ~2 seconds to get an alert when a missing person is seen by the DeepLens
- Built realtime application with highly scalable infrastructure based on serverless backend.
- Easy to add new edge devices and more buckets. Lambda scales automatically.

Future Work

Extensions to the ingestion pipeline:

- Add additional faces to the index using missing persons databases from other countries
- Include other persons of interest (e.g. wanted criminals, fugitives, terrorists)

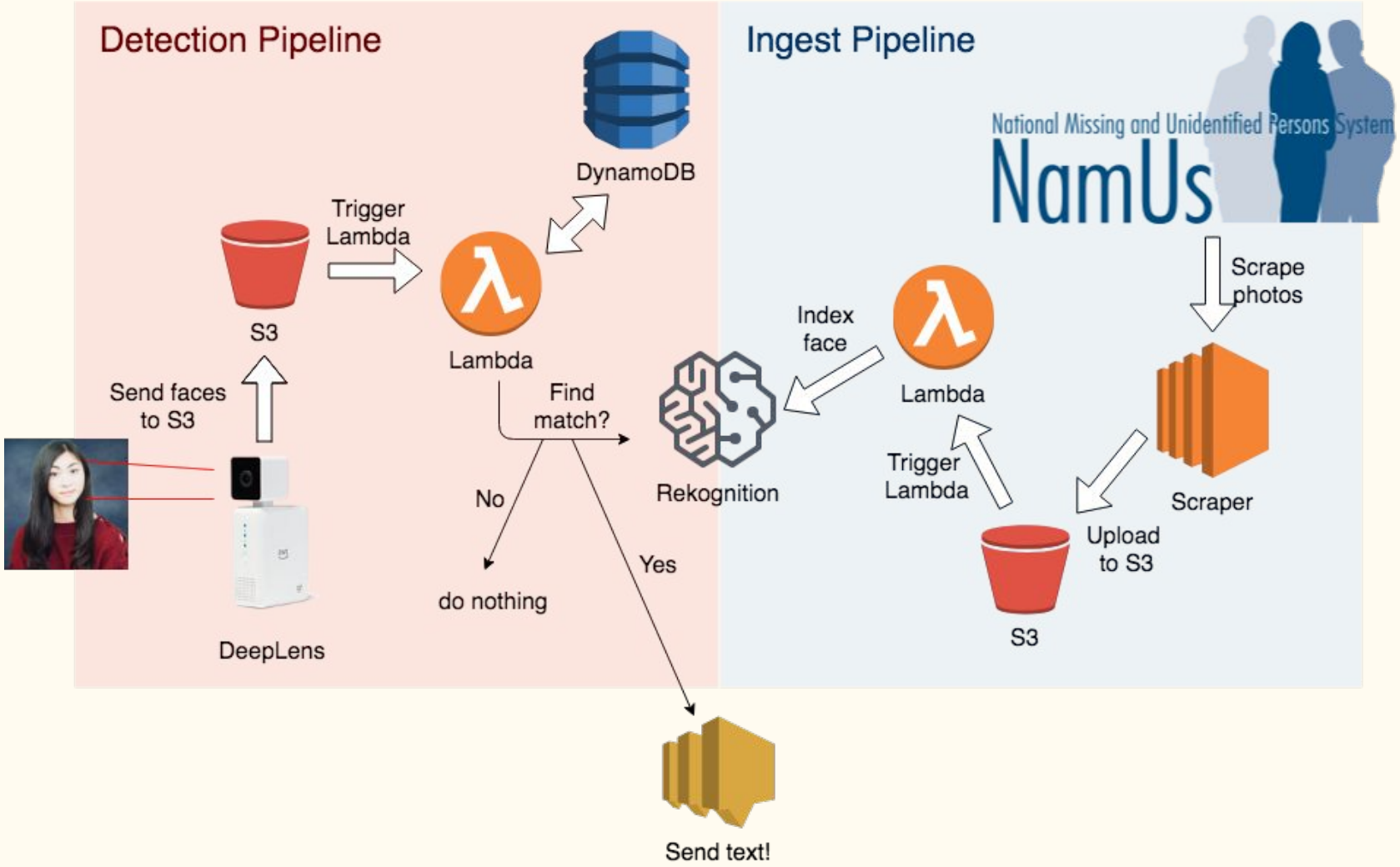
Extensions to the detection pipeline:

- In addition to the DeepLens edge device, we could source images from social media feeds and the web

Implementation Challenges

- Amazon Rekognition is not perfect; false positives and false negatives are possible.
- Facial analysis systems are notorious for misclassifying the faces of women and people of color at much higher rates than they do for faces of white men. Testing this with Amazon Rekognition was beyond the scope of our project, but something we would want to get a baseline on before implementing our system at scale.
- Significant non-technological barriers to the implementation of this system include privacy concerns and the need for government allies.

Questions?


Detection Pipeline
Ingest Pipeline
DynamoDB
National Missing and Unidentified Persons System
NamUs
Trigger Lambda
S3
Lambda
Scrape photos
Index face
Lambda
Send faces to S3
Find match?
Rekognition
Trigger Lambda
Scraper
No
Upload to S3
Yes
do nothing
DeepLens
S3
Send text!