CMPE266 Project Final Report

SJSU SPR2023

Group 1 Xinyu He, Ling Qiu, Vineet Batthina

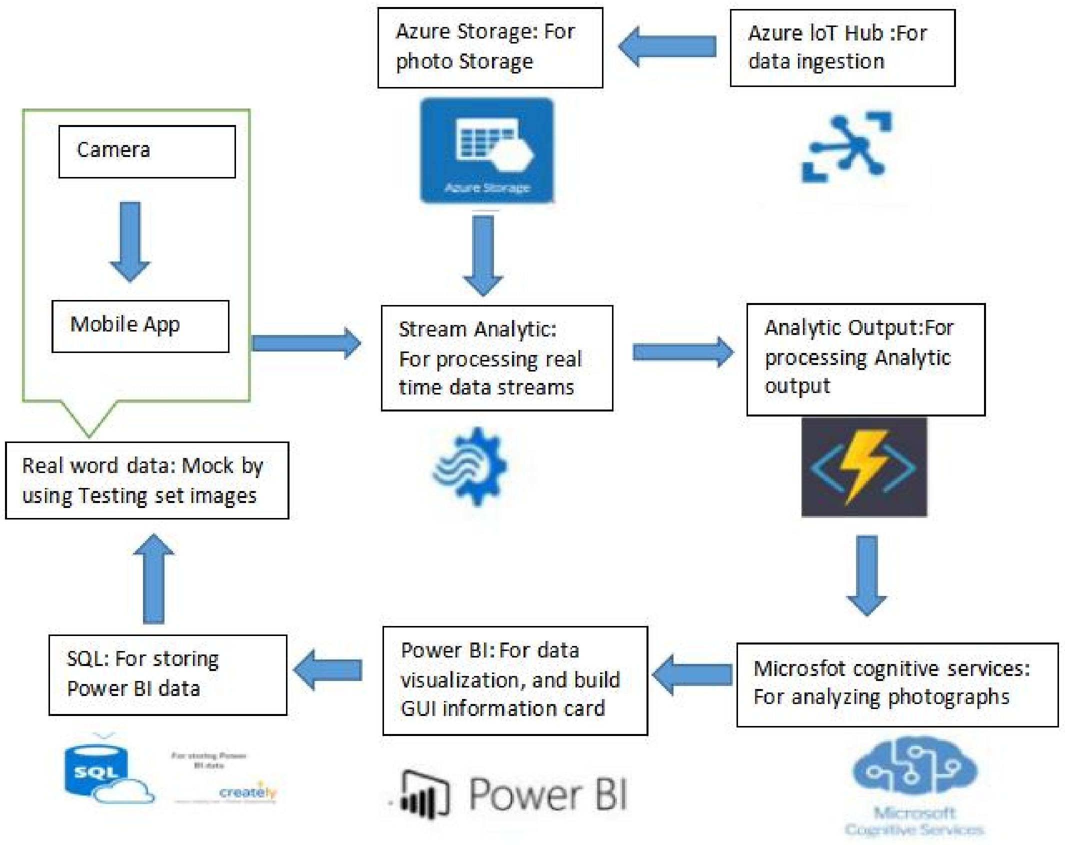
• Project description

When you are travel or do research in the wild word and meet unknown an animal, you can take picture through your camera or smart phone, connected the real-time data to a remote services to recognize the animal and receive the information of this animal fast, so that you can take action accordingly.

• Architectural Overview

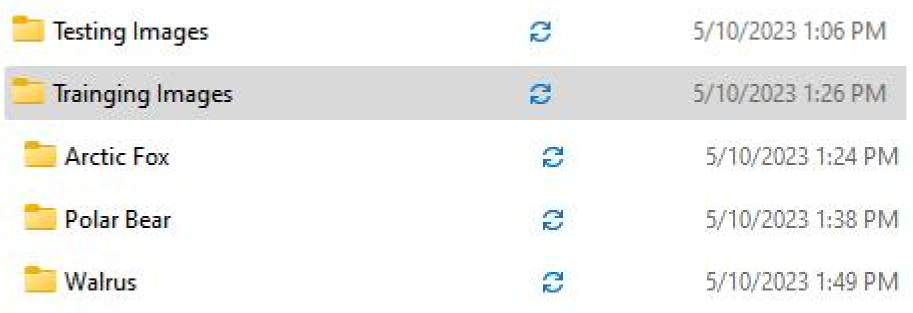
For the user side, in order to get the information of the unknown wild animal, they need to:
1. Take a picture.
2. Sending the picture from mobile app to remote service.
3. Get responding from remote services to get information card to tell detail about the animal.

For the remote service side, in order to achieve the goal for user, it need to implement following steps:
1. Input training set image to Azure loT Hub
2. Use Azure storage to storage trained set Photoes.
3. Receive the testing set images, which mock the real time data come from mobile app, and analytic according to the photo in storage.
4. Using stream analytic tools to processing Analytic output.
5. Using function tool to Analyzing phtograhps
6. Data visualization to make information card to respond to user.
7. SQL storage.
8. Return the information card to user

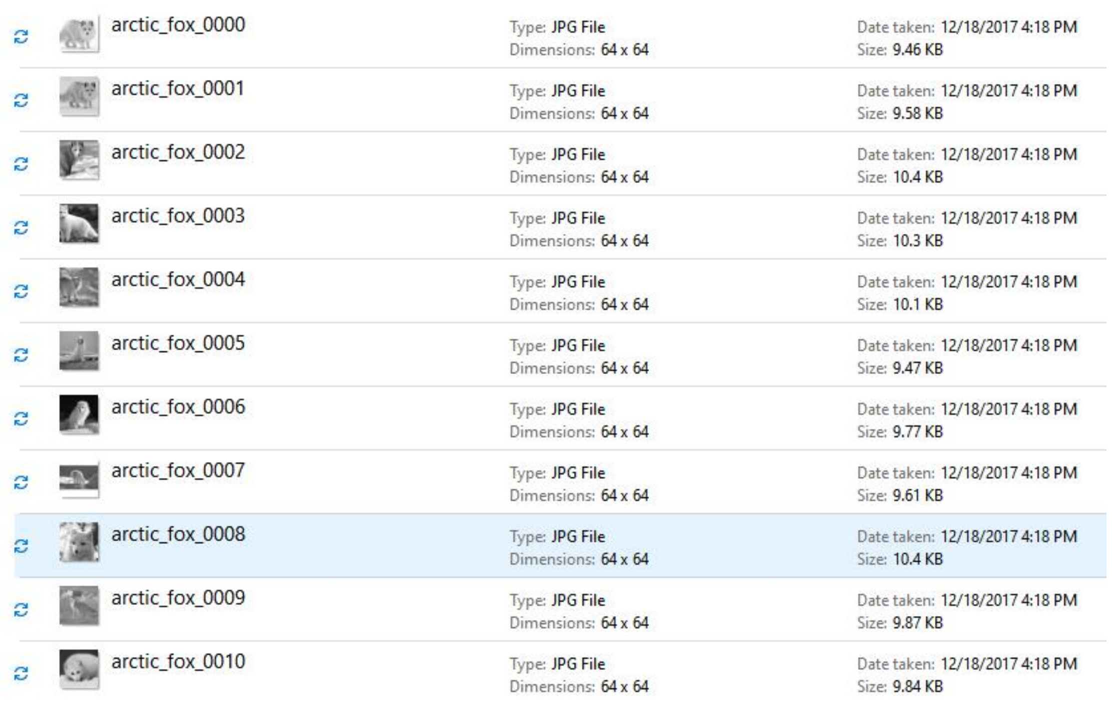

Azure Storage: For photo Storage
Azure loT Hub :For data ingestion
Azure Storage
Camera
Mobile App
Stream Analytic: For processing real time data streams
Analytic Output:For processing Analytic output
Real word data: Mock by using Testing set images
SQL: For storing Power BI data
Power BI: For data visualization, and build GUI information card
Microsfot cognitive services: For analyzing photographs
SQL
creately
Power BI
Microsoft Cognitive Services

• DB Schema, data statistics, indexes used

1. There will be three Three species : Arctic Fox, Polar Bear, Walrus
2. Training set: 100 images each, Testing set :15 images each to mock the real time data.
3. Sample data arctic fox is show as below.



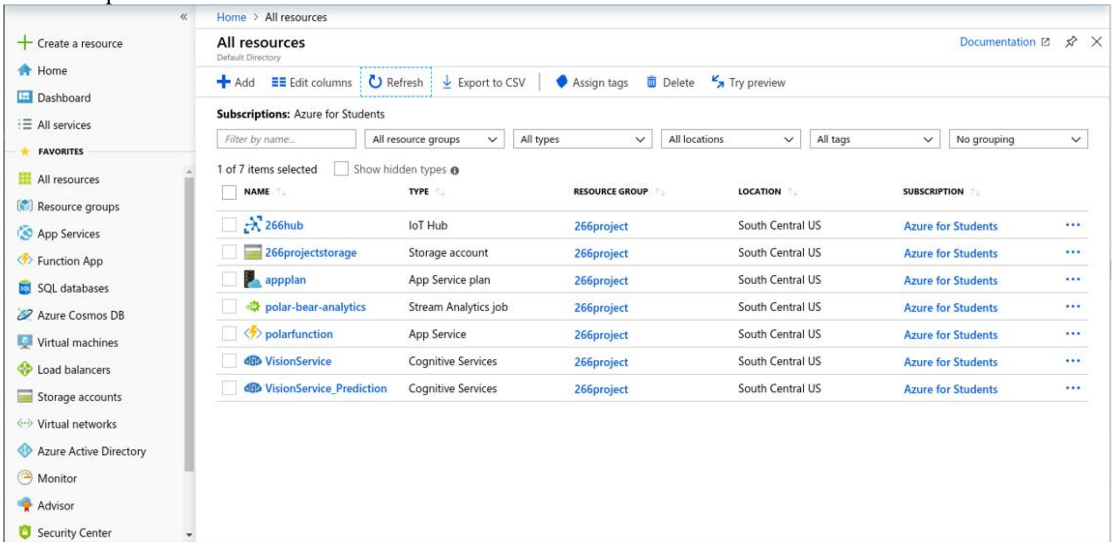
Testing Images	5/10/2023 1:06 PM
Trainging Images	5/10/2023 1:26 PM
Arctic Fox	5/10/2023 1:24 PM
Polar Bear	5/10/2023 1:38 PM
Walrus	5/10/2023 1:49 PM



arctic_fox_0000	Type: JPG File	Dimensions: 64 x 64	Date taken: 12/18/2017 4:18 PM	Size: 9.46 KB
arctic_fox_0001	Type: JPG File	Dimensions: 64 x 64	Date taken: 12/18/2017 4:18 PM	Size: 9.58 KB
arctic_fox_0002	Type: JPG File	Dimensions: 64 x 64	Date taken: 12/18/2017 4:18 PM	Size: 10.4 KB
arctic_fox_0003	Type: JPG File	Dimensions: 64 x 64	Date taken: 12/18/2017 4:18 PM	Size: 10.3 KB
arctic_fox_0004	Type: JPG File	Dimensions: 64 x 64	Date taken: 12/18/2017 4:18 PM	Size: 10.1 KB
arctic_fox_0005	Type: JPG File	Dimensions: 64 x 64	Date taken: 12/18/2017 4:18 PM	Size: 9.47 KB
arctic_fox_0006	Type: JPG File	Dimensions: 64 x 64	Date taken: 12/18/2017 4:18 PM	Size: 9.77 KB
arctic_fox_0007	Type: JPG File	Dimensions: 64 x 64	Date taken: 12/18/2017 4:18 PM	Size: 9.61 KB
arctic_fox_0008	Type: JPG File	Dimensions: 64 x 64	Date taken: 12/18/2017 4:18 PM	Size: 10.4 KB
arctic_fox_0009	Type: JPG File	Dimensions: 64 x 64	Date taken: 12/18/2017 4:18 PM	Size: 9.87 KB
arctic_fox_0010	Type: JPG File	Dimensions: 64 x 64	Date taken: 12/18/2017 4:18 PM	Size: 9.84 KB

• Tools used (refs and goal)

The following image is the tools actually used in Microsoft Azure.In this project, we are focus on the back end parts.



All resources

Default Directory

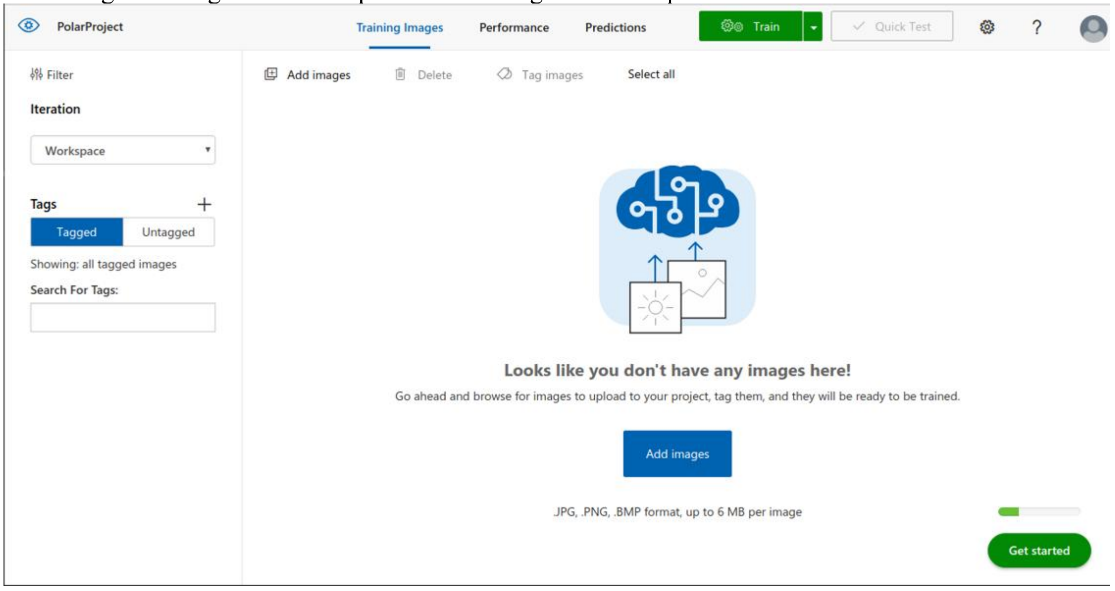
Subscriptions: Azure for Students

1 of 7 items selected

NAME	TYPE	RESOURCE GROUP	LOCATION	SUBSCRIPTION
266hub	IoT Hub	266project	South Central US	Azure for Students
266projectstorage	Storage account	266project	South Central US	Azure for Students
applan	App Service plan	266project	South Central US	Azure for Students
polar-bear-analytics	Stream Analytics job	266project	South Central US	Azure for Students
polarfunction	App Service	266project	South Central US	Azure for Students
VisionService	Cognitive Services	266project	South Central US	Azure for Students
VisionService_Prediction	Cognitive Services	266project	South Central US	Azure for Students

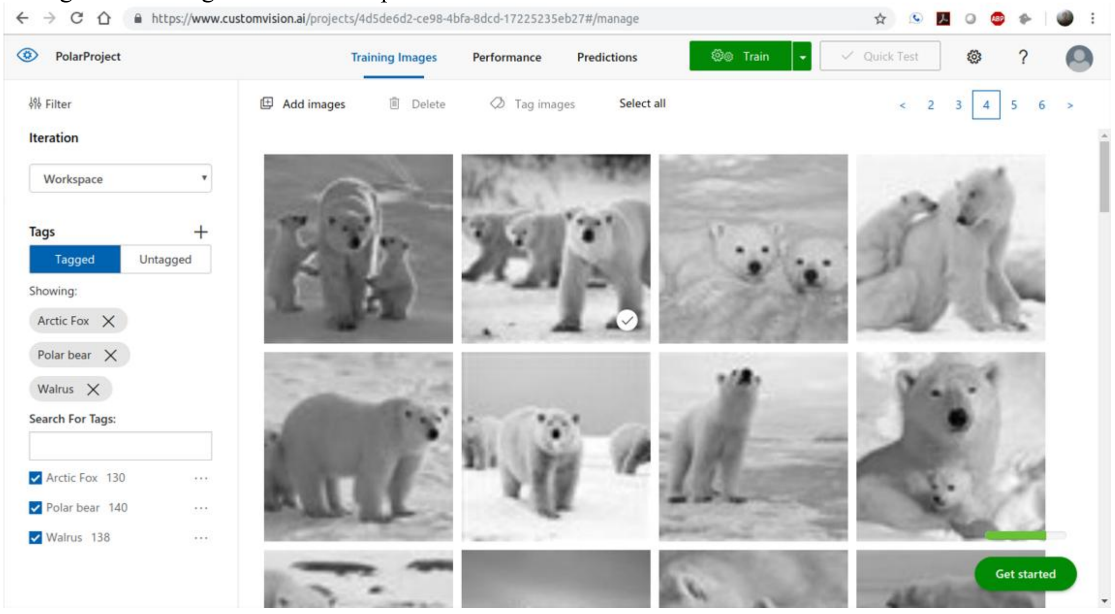
1.Microsoft Azure IoT Hub: This can be used to ingest any kind of streaming data via a simulated recording media for example - cameras.
2.Microsoft Azure Storage: Any kind of streaming data can be stored here.
3.Microsoft Azure Stream Analytic: This service can be used to process real-time data streams.
4.Microsoft Azure Function: This service can be used to process outputs from the Stream Analytic service.
5.Microsoft Custom Vision Service: This service can be used to analyze/recognize images.
6.Microsoft Power BI: This service can be used to create customized dashboards for data visualization.

• Features designed/implemented
Screenshots
1. Training data
Collecting 100 images for three species as training data and input to loT Hub.

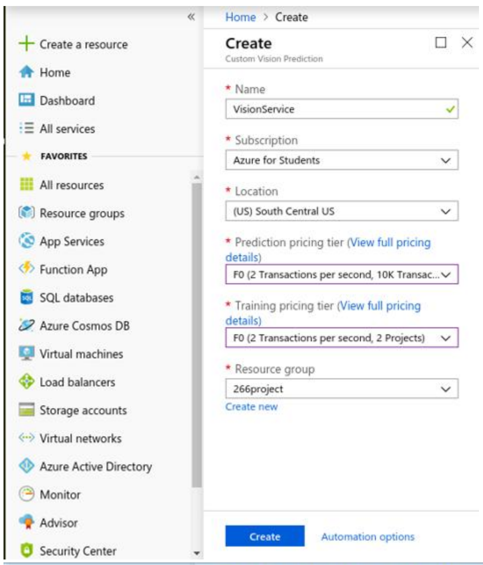


Looks like you don't have any images here!
Go ahead and browse for images to upload to your project, tag them, and they will be ready to be trained.
JPG, .PNG, .BMP format, up to 6 MB per image

2. Training data storage
Storage All training data for three species.



3. App Service Plan
Create a service plan and use the south central US as our remote service.



Create
Custom Vision Prediction

* Name
VisionService

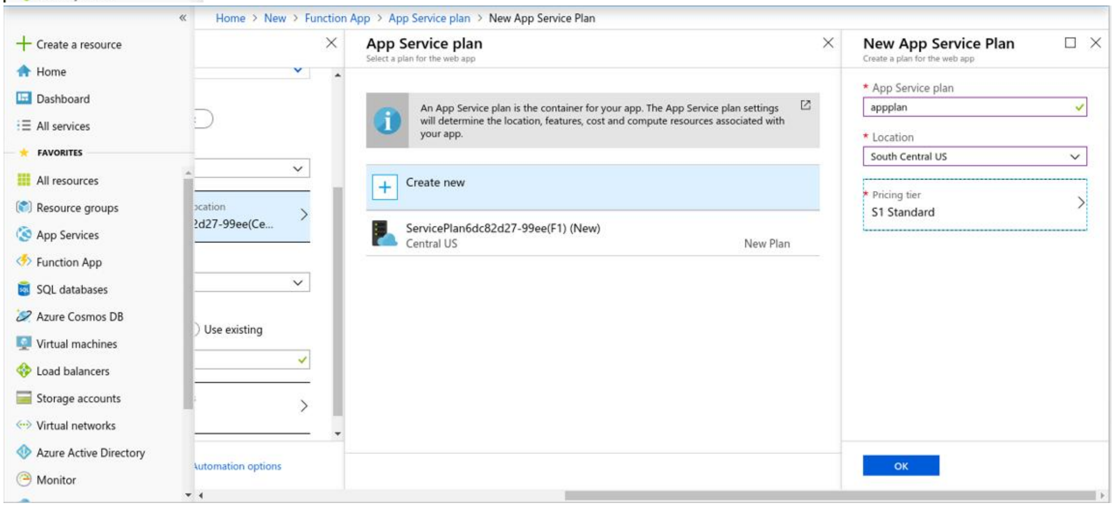
* Subscription
Azure for Students

* Location
(US) South Central US

* Prediction pricing tier (View full pricing details)
F0 (2 Transactions per second, 10K Transac...

* Training pricing tier (View full pricing details)
F0 (2 Transactions per second, 2 Projects)

* Resource group
266project
Create new



App Service plan
Select a plan for the web app

An App Service plan is the container for your app. The App Service plan settings will determine the location, features, cost and compute resources associated with your app.

Create new

ServicePlan6dc82d27-99ee(F1) (New)
Central US New Plan

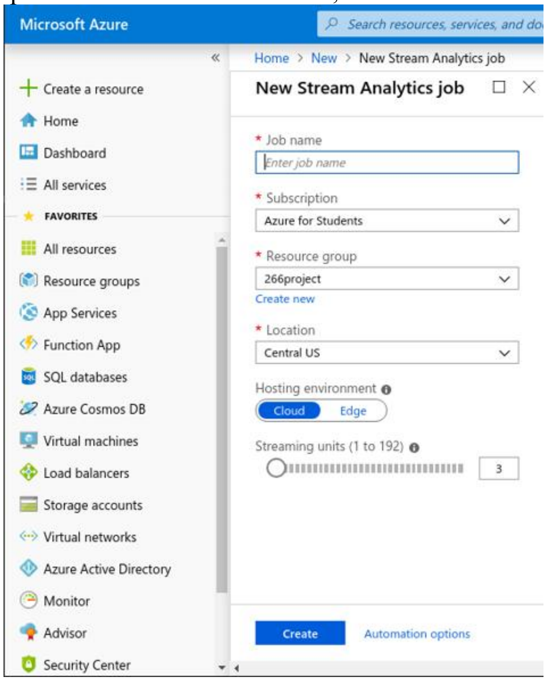
New App Service Plan
Create a plan for the web app

* App Service plan
appplan

* Location
South Central US

* Pricing tier
S1 Standard

4. Stream Analysis
 process real-time data streams, in this case camera input.



New Stream Analytics job

* Job name
Enter job name

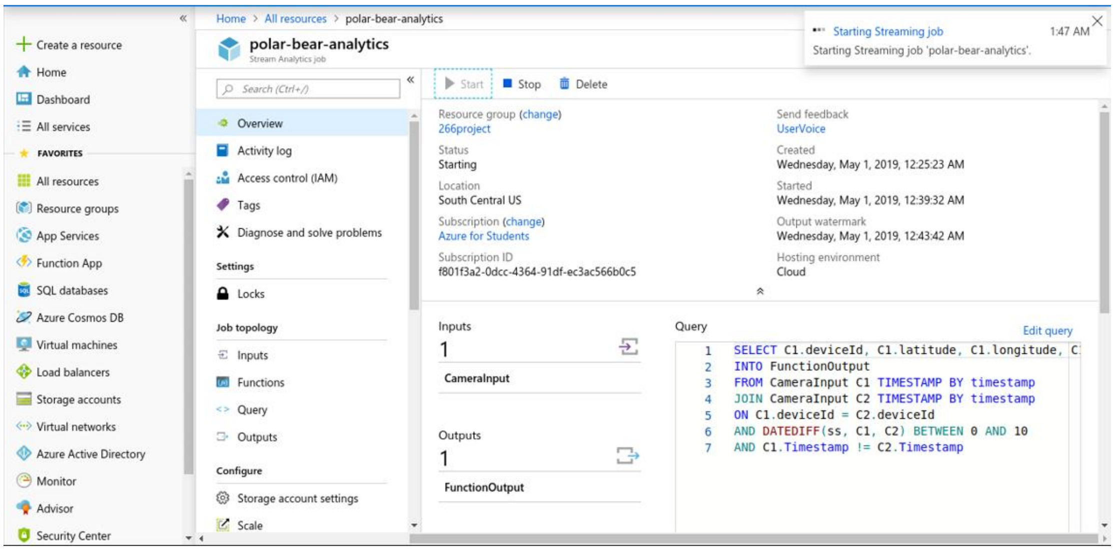
* Subscription
Azure for Students

* Resource group
266project
Create new

* Location
Central US

Hosting environment
[✓ Cloud] Edge

Streaming units (1 to 192)
3


Home > All resources > polar-bear-analytics

polar-bear-analytics
Stream Analytics job

Starting Streaming job — 1:47 AM
Starting Streaming job 'polar-bear-analytics'.

Start | Stop | Delete

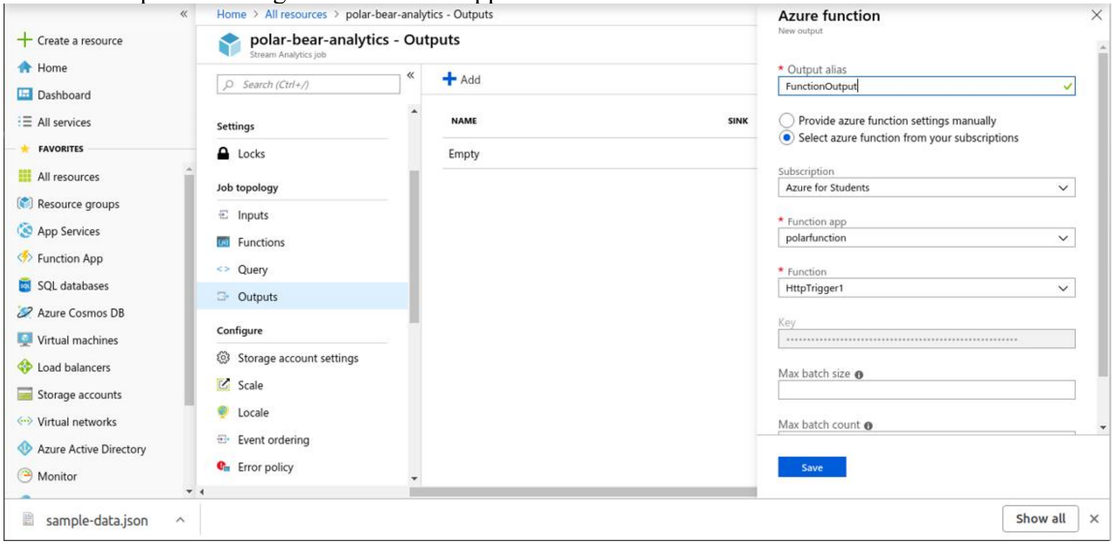
Resource group (change): 266project
Status: Starting
Location: South Central US
Subscription (change): Azure for Students
Subscription ID: f801f3a2-0dcc-4364-91df-ec3ac566b0c5
Send feedback: UserVoice
Created: Wednesday, May 1, 2019, 12:25:23 AM
Started: Wednesday, May 1, 2019, 12:39:32 AM
Output watermark: Wednesday, May 1, 2019, 12:43:42 AM
Hosting environment: Cloud

Inputs: 1 — CameraInput
Outputs: 1 — FunctionOutput

Query (Edit query)

```
SELECT C1.deviceId, C1.latitude, C1.longitude, C
INTO FunctionOutput
FROM CameraInput C1 TIMESTAMP BY timestamp
JOIN CameraInput C2 TIMESTAMP BY timestamp
ON C1.deviceId = C2.deviceId
AND DATEDIFF(ss, C1, C2) BETWEEN 0 AND 10
AND C1.Timestamp != C2.Timestamp
```

5. Stream Analysis output
Generate output according to the function App.


Home > All resources > polar-bear-analytics - Outputs

polar-bear-analytics - Outputs
Stream Analytics job

Add

NAME	SINK
Empty	

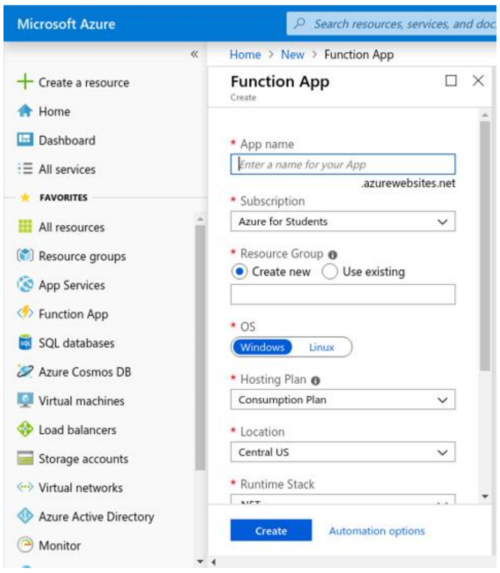
Azure function
New output

Output alias: FunctionOutput
( ) Provide azure function settings manually
(●) Select azure function from your subscriptions
Subscription: Azure for Students
Function app: polarfunction
Function: HttpTrigger1
Key: ••
Max batch size:
Max batch count:
Save

sample-data.json — Show all

Queries performed

1. Function: method - LSH.

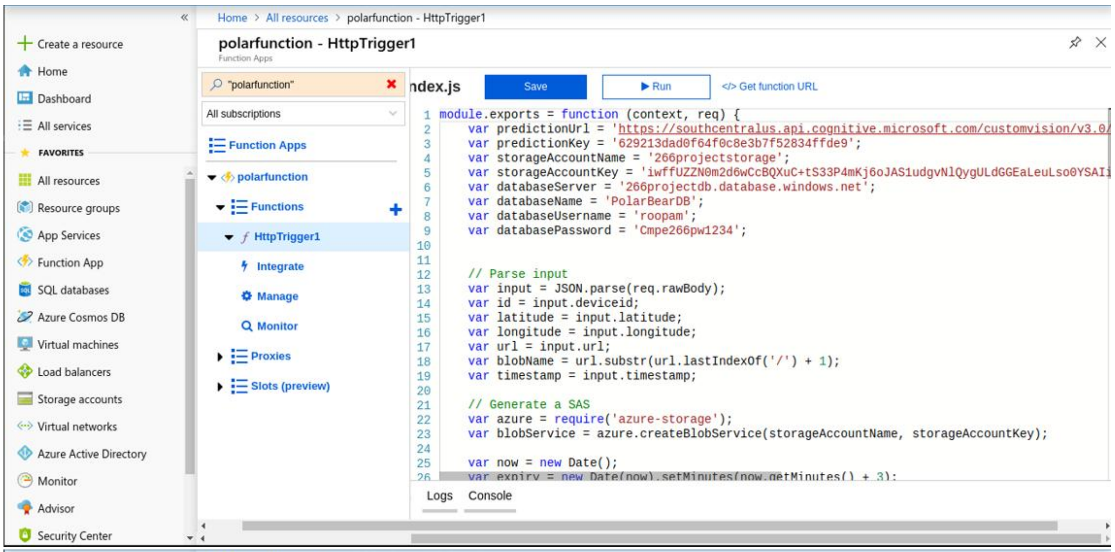

Microsoft Azure — Search resources, services, and doc

Home > New > Function App

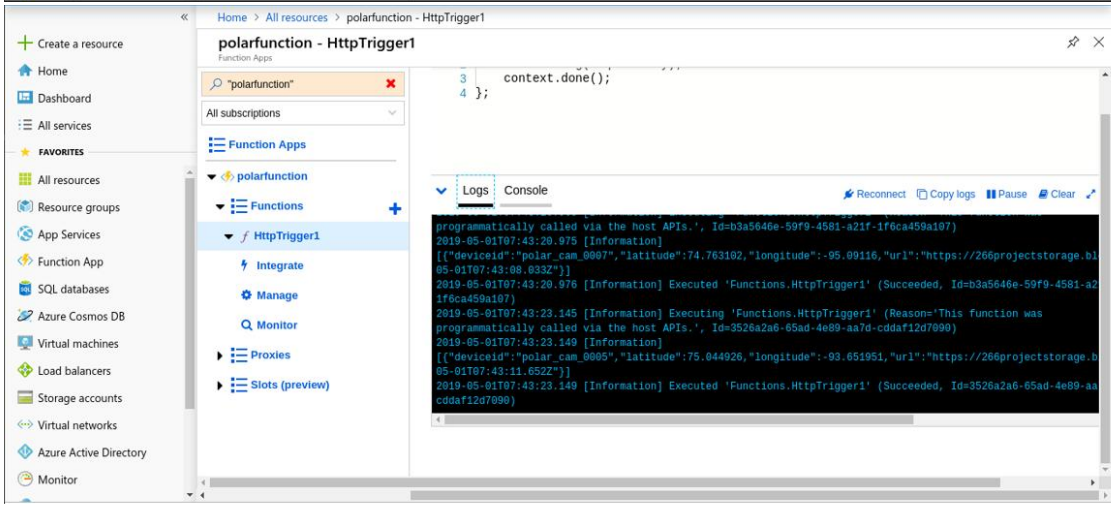
Function App
Create

* App name: Enter a name for your App .azurewebsites.net
* Subscription: Azure for Students
* Resource Group: [✓ Create new] Use existing
* OS: [✓ Windows] Linux
* Hosting Plan: Consumption Plan
* Location: Central US
* Runtime Stack: .NET

Create | Automation options



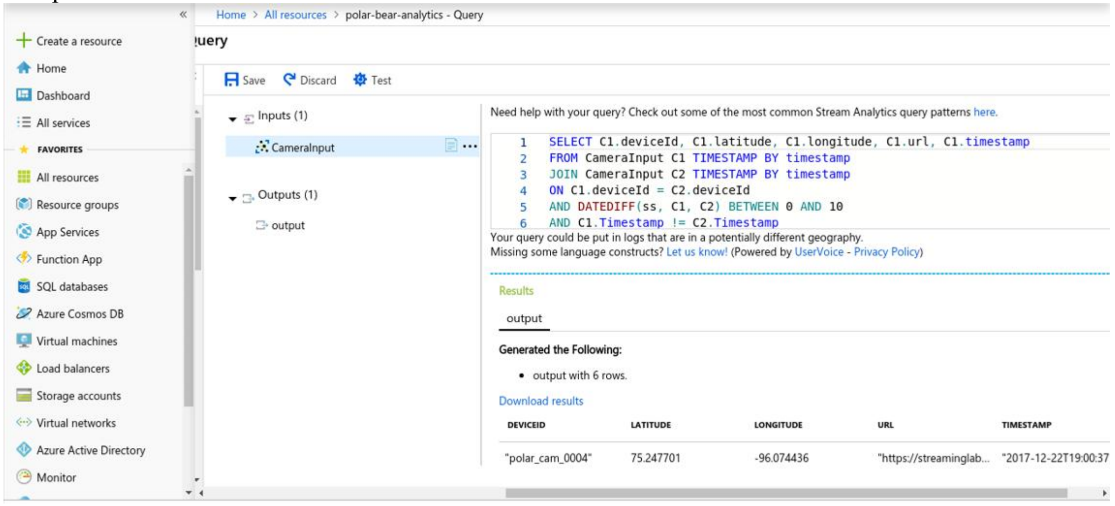
```
module.exports = function (context, req) {
    var predictionUrl = 'https://southcentralus.api.cognitive.microsoft.com/customvision/v3.0/
    var predictionKey = '629213dad0f64f0c8e3b7f52834ffde9';
    var storageAccountName = '266projectstorage';
    var storageAccountKey = 'iwffUZZN0m2d6wCcBQXuC+tS33P4mKj6oJAS1udgvNlQygULdGGEaLeuLso0YSAIi
    var databaseServer = '266projectdb.database.windows.net';
    var databaseName = 'PolarBearDB';
    var databaseUsername = 'roopam';
    var databasePassword = 'Cmpe266pw1234';


    // Parse input
    var input = JSON.parse(req.rawBody);
    var id = input.deviceid;
    var latitude = input.latitude;
    var longitude = input.longitude;
    var url = input.url;
    var blobName = url.substr(url.lastIndexOf('/') + 1);
    var timestamp = input.timestamp;

    // Generate a SAS
    var azure = require('azure-storage');
    var blobService = azure.createBlobService(storageAccountName, storageAccountKey);

    var now = new Date();
    var expiry = new Date(now).setMinutes(now.getMinutes() + 3);
```



```
    context.done();
};
```

```
programmatically called via the host APIs.', Id=b3a5646e-59f9-4581-a21f-1f6ca459a107)
2019-05-01T07:43:20.975 [Information]
[{"deviceid":"polar_cam_0007","latitude":74.763102,"longitude":-95.09116,"url":"https://266projectstorage.bl
05-01T07:43:08.033Z"}]
2019-05-01T07:43:20.976 [Information] Executed 'Functions.HttpTrigger1' (Succeeded, Id=b3a5646e-59f9-4581-a2
1f6ca459a107)
2019-05-01T07:43:23.145 [Information] Executing 'Functions.HttpTrigger1' (Reason='This function was
programmatically called via the host APIs.', Id=3526a2a6-65ad-4e89-aa7d-cddaf12d7090)
2019-05-01T07:43:23.149 [Information]
[{"deviceid":"polar_cam_0005","latitude":75.044926,"longitude":-93.651951,"url":"https://266projectstorage.b
05-01T07:43:11.652Z"}]
2019-05-01T07:43:23.149 [Information] Executed 'Functions.HttpTrigger1' (Succeeded, Id=3526a2a6-65ad-4e89-aa
cddaf12d7090)
```

2. Input



Need help with your query? Check out some of the most common Stream Analytics query patterns here.

```
SELECT C1.deviceId, C1.latitude, C1.longitude, C1.url, C1.timestamp
FROM CameraInput C1 TIMESTAMP BY timestamp
JOIN CameraInput C2 TIMESTAMP BY timestamp
ON C1.deviceId = C2.deviceId
AND DATEDIFF(ss, C1, C2) BETWEEN 0 AND 10
AND C1.Timestamp != C2.Timestamp
```

Your query could be put in logs that are in a potentially different geography.
Missing some language constructs? Let us know! (Powered by UserVoice - Privacy Policy)

Results

output

Generated the Following:

- output with 6 rows.

Download results

DEVICEID	LATITUDE	LONGITUDE	URL	TIMESTAMP
"polar_cam_0004"	75.247701	-96.074436	"https://streaminglab...	"2017-12-22T19:00:37

3. Test

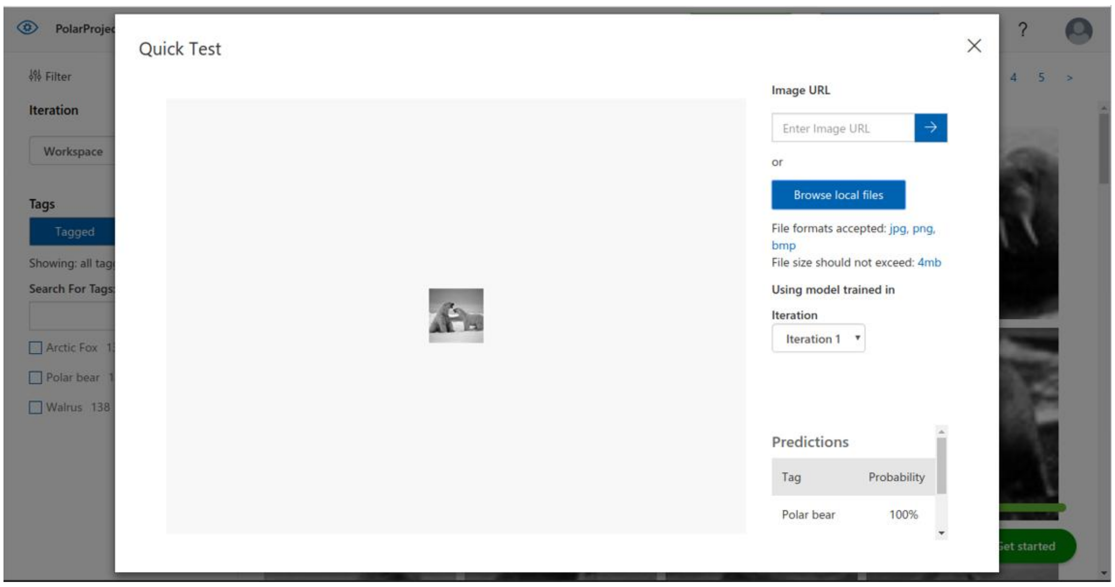

Quick Test

Image URL

Enter Image URL

or

Browse local files

File formats accepted: jpg, png, bmp

File size should not exceed: 4mb

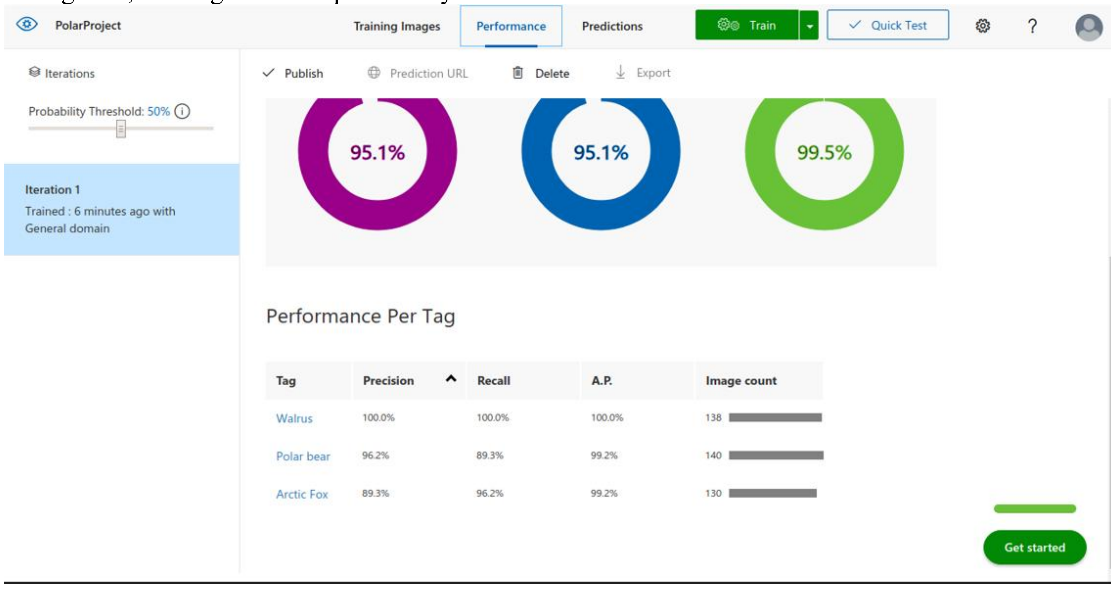
Using model trained in

Iteration: Iteration 1

Predictions

Tag	Probability
Polar bear	100%

4. Results
For the desire testing data from three species are performance well. However,if use other sepcies as testing case, it will generate unpredictably results.



Probability Threshold: 50%

Iteration 1
Trained : 6 minutes ago with General domain

95.1% 95.1% 99.5%

Performance Per Tag

Tag	Precision	Recall	A.P.	Image count
Walrus	100.0%	100.0%	100.0%	138
Polar bear	96.2%	89.3%	99.2%	140
Arctic Fox	89.3%	96.2%	99.2%	130

• Lessons learned
1.Microsoft Azure Tools:
Learn more about how to use this tools
2.Software develop life cycle(waterfall):
Draw flowchart at the beginning to help manage schedule better

• Evaluation/ Open issues/future work
Evaluation:

Open issues:
1. can not tell differences between similar species. For example: can not distinguish polar bear and other bears. Solution:need more factors help to distinguish, such as the location of the data sending.
Future work:
1. Front end Mobile app
2. Collecting more data for more species

• Systems requirements to install and run the project / environments
Microsoft Azure can be used in both Linux and windows system. All the service provide by Azure and do not require to install any software.
In our project, we are using windows system, and the Azure provided GUI interface.

• Github link with the code and sample collection
https://github.com/Jeremiah0715/CMPE266_Group_Project