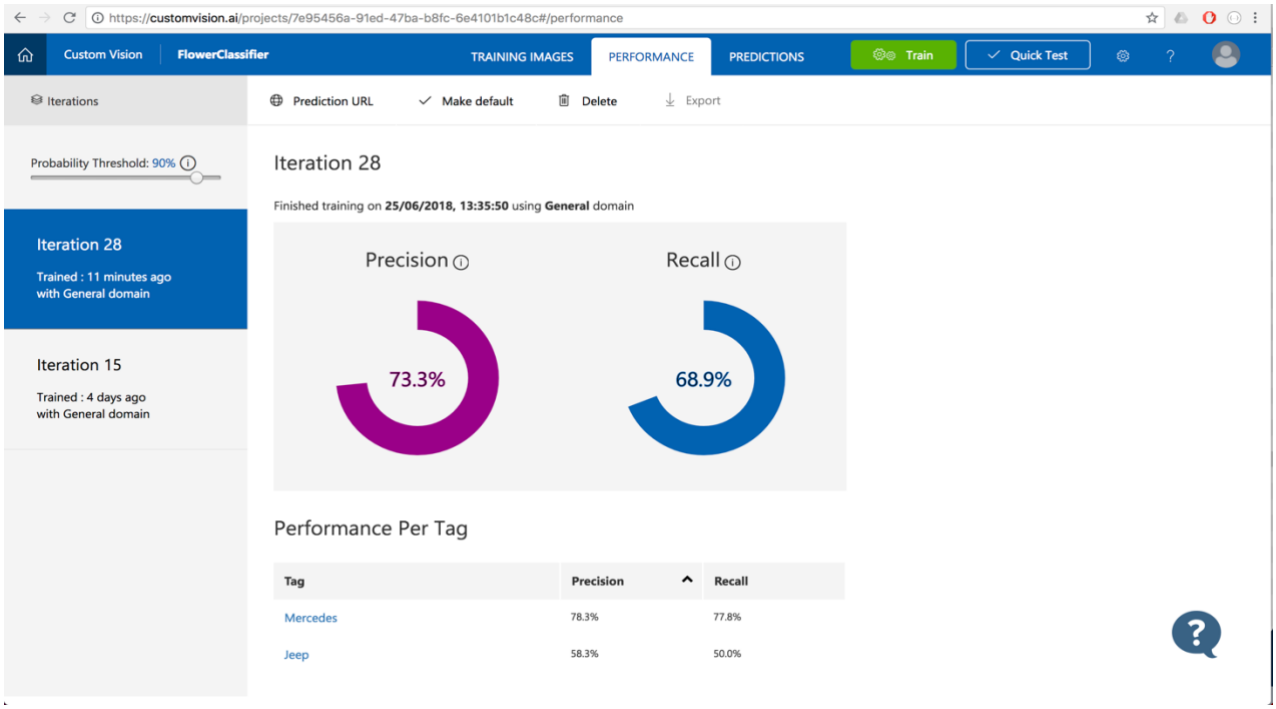
Homework 4

For my custom AI experiment, **Initially**, I have uploaded 20 images of two cars, Jeep and Mercedes. Next, I trained and built the model for classification of two different cars.

In image classification, we first feed different types of images and use classification technique to train and build a model. Once we are done with building the classification model, we test our model's precision and prediction accuracy by feeding more images without any classification or label assigned to them. If we get less accurate results on the testing phase then we add more images and train the model again for better accuracy and precision. This loop continues until we get satisfactory results.

In this project, as mentioned above I have taken two cars and uploaded some pictures of each category and labeled them accordingly Jeep and Mercedes. Set the *threshold* to 90% that is *Minimum probability score for a prediction to be valid when calculating precision and recall* and then I called 'Train' functionality of the system to check the precision and recall percentage.

Iteration 1: with small input set (Jeep and Mercedes cars)



This is our first iteration (Once model is built, we are ready to test it against some samples).

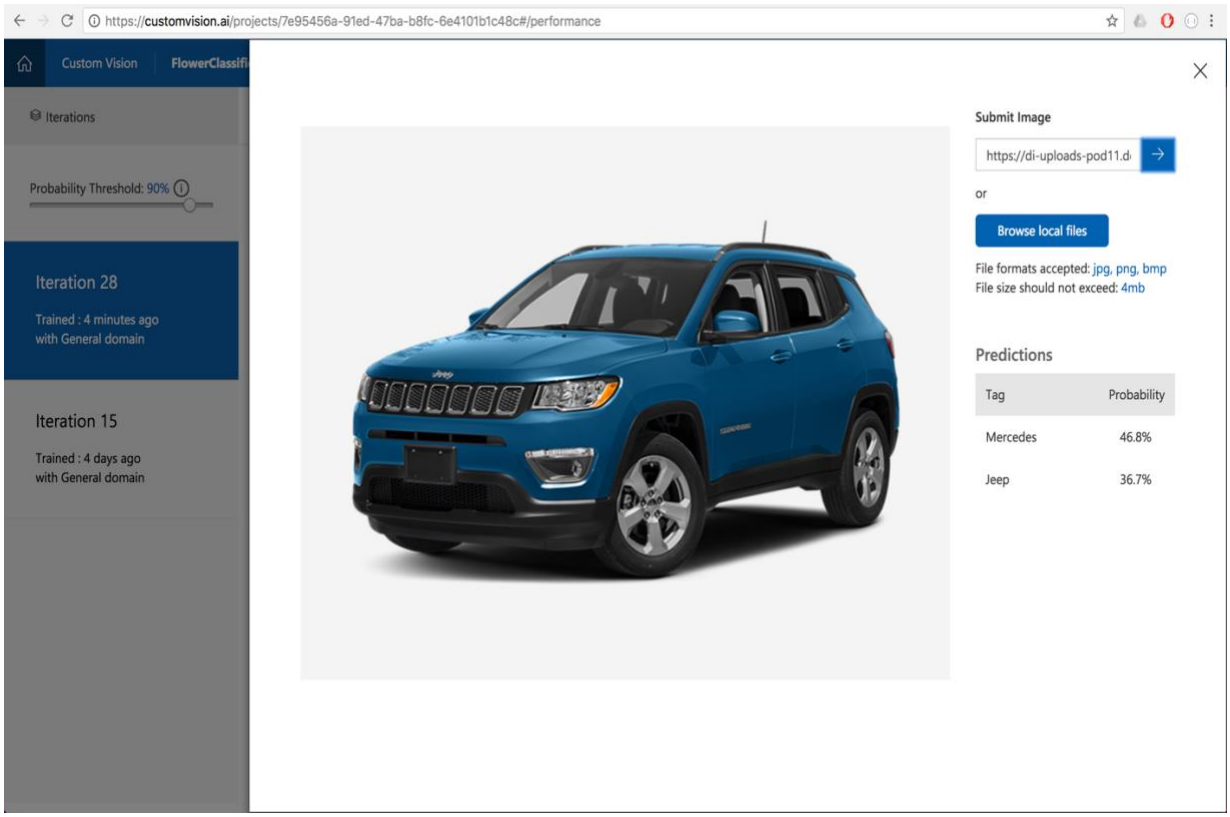
In the first iteration, we overall got 73.3% precision and 68.9% recall.

The **precision** percentage **will tell us: if a tag is predicted by our model, how likely is that to be right?**

The **Recall** percentage **will tell us: out of the tags which should be predicted correctly, what percentage did our model correctly find?**

Here in the first iteration with some we trained our system with small number of inputs so there 73.3% likelihood that next test input will be correctly identified or categorized as Jeep or Mercedes.

Now let's test the model, For the testing purpose we take a **Jeep image** and pass to initial iteration result model. As an output we get 36.7% likelihood of jeep and 46.8% confidence to the Mercedes which not so good. To reconfirm our output, we can use custom ai's API by passing iteration id, prediction id and project id along with image URL. The request would http post and image URL will be a post parameter.



API URL:
https://southcentralus.api.cognitive.microsoft.com/customvision/v2.0/Prediction/7e95456a-91ed-47ba-b8fc-6e4101b1c48c/url?iterationId=**1469d08d-795d-450f-8598-ae70f7894e23**

Here is http post request and response for the **first iteration** of our custom AI project for the Jeep image.

HTTP Post request: We passed new Jeep image as an argument

```
Host: southcentralus.api.cognitive.microsoft.com

Prediction-Key: f8fe79b850a6471c9ab1a5fec9f7cae1

Content-Type: application/json {

  "Url": "https://di-uploads-pod11.dealerinspire.com/raylaethemchryslerdodgejeep/uploads/2017/09/2018-Jeep-Compass-Hero.png"

}
```

Response content:


```
Date: Mon, 25 Jun 2018 21:01:34 GMT

Content-Length: 389

Content-Type: application/json; charset=utf-8 {

  "id": "64077899-0de8-4aac-a8a3-378ae18ed9db",

  "project": "7e95456a-91ed-47ba-b8fc-6e4101b1c48c",

  "iteration": "1469d08d-795d-450f-8598-ae70f7894e23",

  "created": "2018-06-25T21:01:34.487003Z",

  "predictions": [{

    "probability": 0.468212515,

    "tagId": "b2af220e-a4f4-43b7-9c98-c13a6c9f8ca1",

    "tagName": "Mercedes"

  }, {

    "probability": 0.367403924,

    "tagId": "8e958793-dfd4-40c5-92ee-b861c7005b5e",
```

```
  "tagName": "Jeep"

 }]

}
```


Now let's test the model against **Mercedes** car, For the testing purpose we take an image of **Mercedes image** and pass to **initial iteration** model. As an output we get 47.4% likelihood of Mercedes and 27.5% confidence to the Jeep which not so good but better as compare to Jeep confidence by this model. There is 47% percentage chance that image is Mercedes car.

HTTP Post request for Mercedes car, Iteration one

```
Host: southcentralus.api.cognitive.microsoft.com

Prediction-Key: f8fe79b850a6471c9ab1a5fec9f7cae1

Content-Type: application/json {

 "Url": "https://i.ytimg.com/vi/RhDlsFL2SIg/maxresdefault.jpg"

}
```

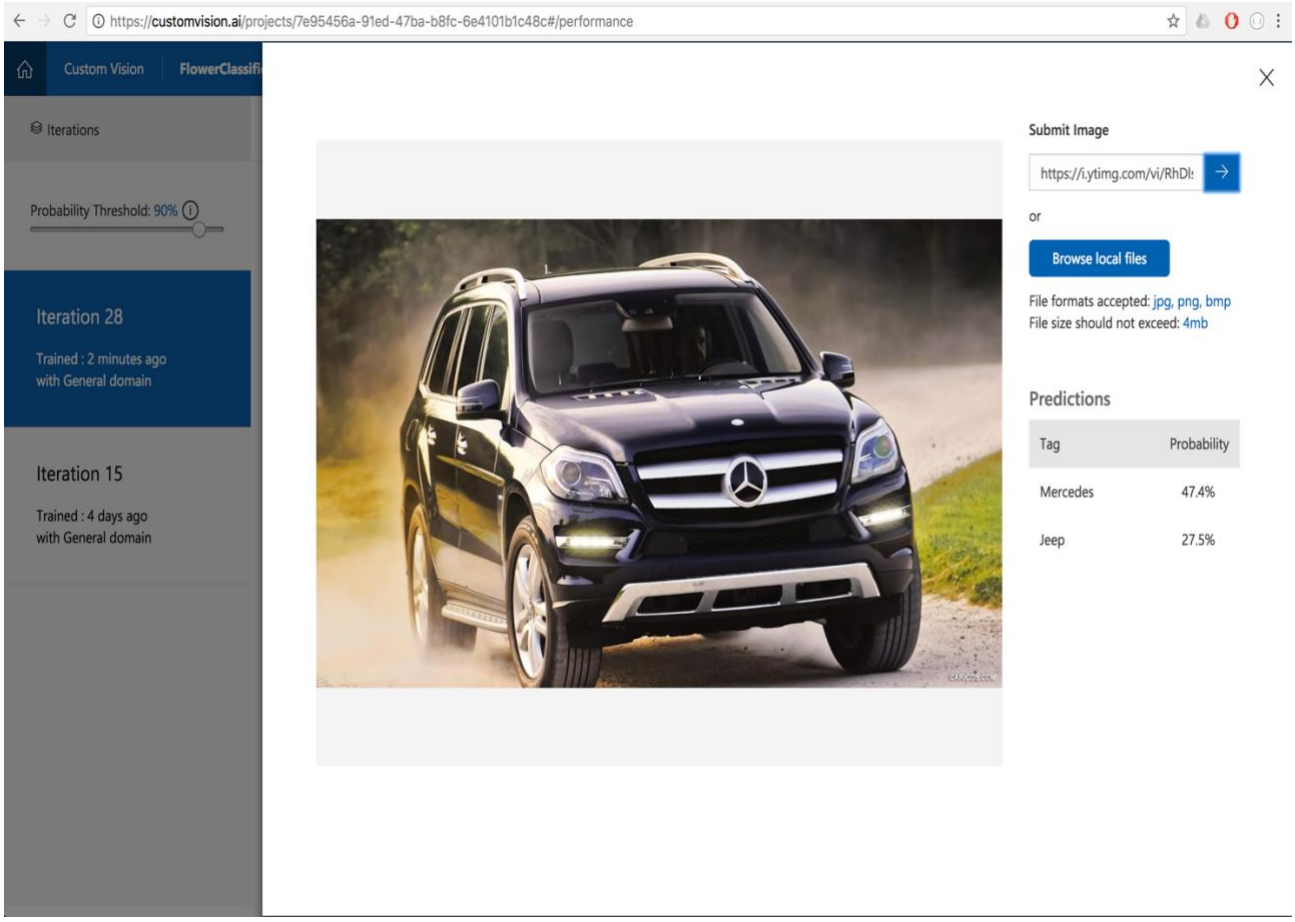
Http Response:


```
Date: Wed, 27 Jun 2018 05:37:12 GMT

Content-Type: application/json; charset=utf-8 {

 "id": "c13ec41e-f033-4123-8aec-3a02a98eb6c2",

 "project": "7e95456a-91ed-47ba-b8fc-6e4101b1c48c",

 "iteration": "1469d08d-795d-450f-8598-ae70f7894e23",

 "created": "2018-06-27T05:37:12.2842814Z",

 "predictions": [{

  "probability": 0.474129379,

  "tagId": "b2af220e-a4f4-43b7-9c98-c13a6c9f8ca1",

  "tagName": "Mercedes"
```

```
  }, {
    "probability": 0.275250763,
    "tagId": "8e958793-dfd4-40c5-92ee-b861c7005b5e",
    "tagName": "Jeep"
  }]
}
```


Iteration one had very less input sets so our model was not so strong. The model had very vast boundaries and somewhat able to fit both the different images in both category that means not able to clearly differentiate Jeep and Mercedes.

To get more accurate result, we need to supply more images of each category so that new model can have more features from each category and when we go for the testing, our model can extract feature from test set and try to compare with both classified results and gives the confidence level from each resulting set.
If our test set image features have higher degree of matching features with one of the sets then we'll have higher chance to getting that tag.

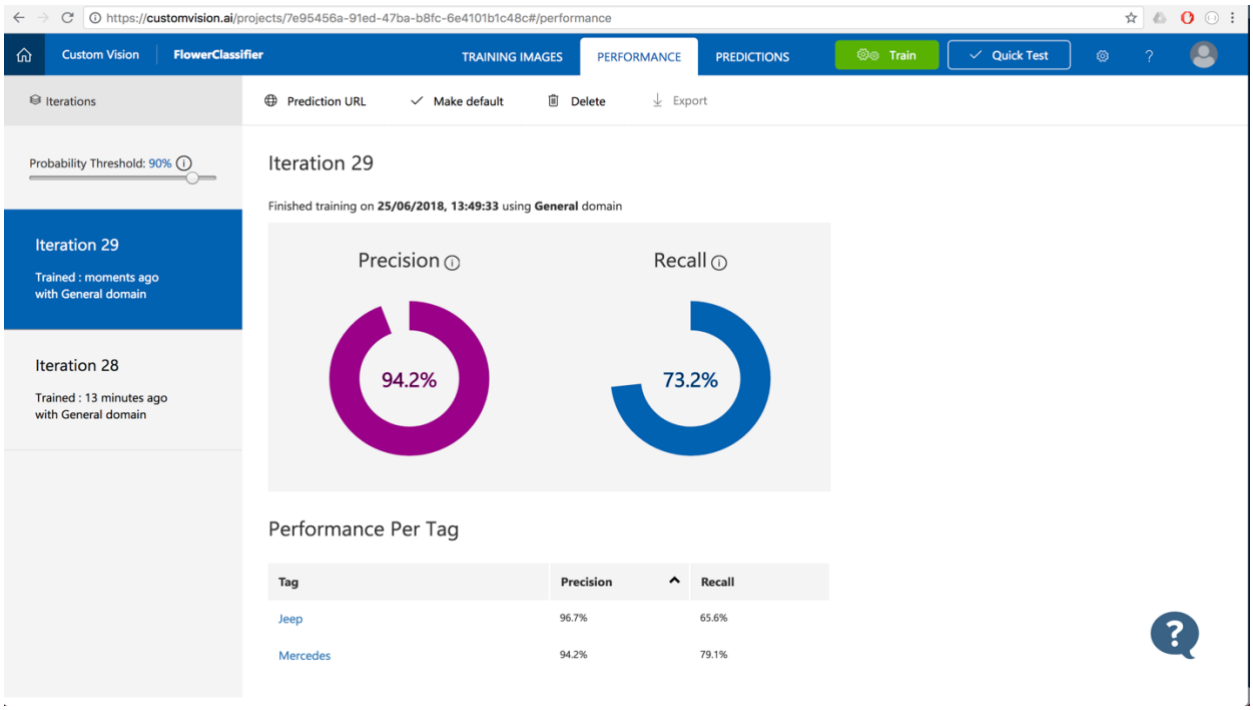


Iteration 2:

So basically, we can say that our initial model was not good and not so accurate. It's precision and accuracy are low. To improve the precision and confidence we need to give more **labeled** images and train our system again. In this project we have fed around 100 more images for each category of the cars and labeled them accordingly before calling the 'train' functionality of the system.

In this way custom AI system will extract more features from each categorized image for the greater precision and recall. Once we are done with this step we get following results.

Here in the **second iteration** we trained our system with **very large dataset** of inputs so there **94.2% likelihood** that next test input will be correctly identified or categorized as Jeep or Mercedes and 73.2% as recall confidence.



Now let's test the new précised model **against same input set from iteration one**. For the testing purpose we take a **same Jeep image** from iteration one and pass to second iteration model. As an output we get 93.7% likelihood of **jeep** and only 16.8% confidence to the Mercedes which is way better than iteration one.

To reconfirm our output, we can use custom ai's API by passing iteration id, prediction id and project id along with same jeep image URL. The request would be http post and image URL will be a post parameter.

API URL:
https://southcentralus.api.cognitive.microsoft.com/customvision/v2.0/Prediction/7e95456a-91ed-47ba-b8fc-6e4101b1c48c/url?iterationId=**81f53285-e3b4-44e3-885a-9988fc35f6c8**

HTTPS Post Request: We passed same Jeep image as an argument

Host: southcentralus.api.cognitive.microsoft.com

Prediction-Key: f8fe79b850a6471c9ab1a5fec9f7cae1

Content-Type: application/json {

"Url": "https://di-uploads-pod11.dealerinspire.com/raylaethemchryslerdodgejeep/uploads/2017/09/2018-Jeep-Compass-Hero.png"

}

HTTP Response:

Date: Mon, 25 Jun 2018 20:59:48 GMT

Content-Length: 388

Content-Type: application/json; charset=utf-8 {

"id": "ca86bc49-5d98-407e-ae10-38790761717b",

"project": "7e95456a-91ed-47ba-b8fc-6e4101b1c48c",

"iteration": "81f53285-e3b4-44e3-885a-9988fc35f6c8",

"created": "2018-06-25T20:59:48.4573752Z",

"predictions": [{

"probability": 0.9310171,

"tagId": "8e958793-dfd4-40c5-92ee-b861c7005b5e",

"tagName": "Jeep"

}, {

"probability": 0.166567564,

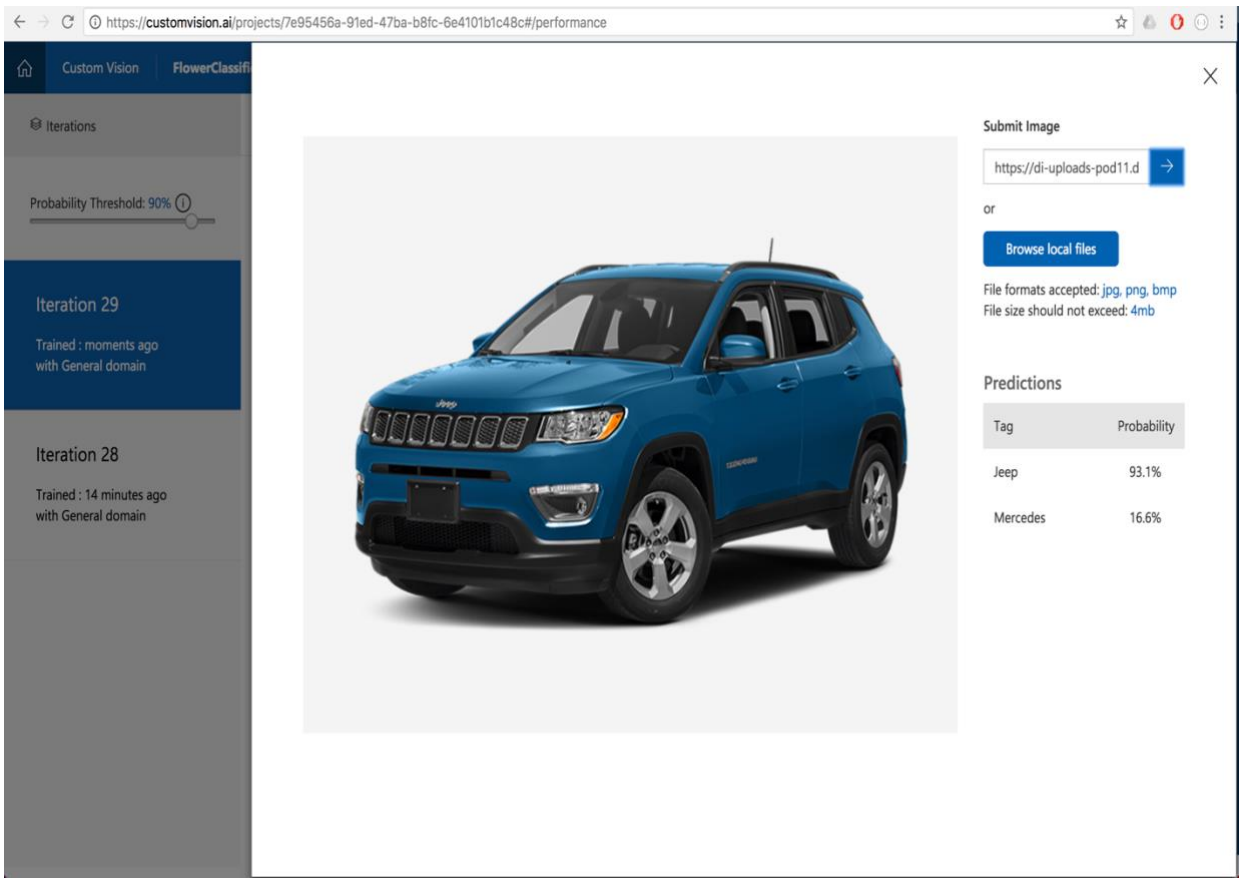
"tagId": "b2af220e-a4f4-43b7-9c98-c13a6c9f8ca1",

"tagName": "Mercedes"

}]

}

In **iteration one** we got only 36% confidence for this same Jeep image however in **iteration two** we received 93% confidence for that same image passed in the argument.



Now let's test the new model for **Mercedes** car, For the testing purpose we take a same image of **Mercedes image** from iteration one and pass to **second iteration** model. As a result, we get 99% likelihood of **Mercedes** and less than one percentage confidence to the Jeep which very precise and way better as compare to iteration one.

HTTP Post request: **We passed same Mercedes image from iteration one as an argument**

```
Host: southcentralus.api.cognitive.microsoft.com

Prediction-Key: f8fe79b850a6471c9ab1a5fec9f7cae1

Content-Type: application/json {

  "Url": "https://i.ytimg.com/vi/RhDlsFL2SIg/maxresdefault.jpg"

}
```

Https Response for Mercedes car, Iteration two:

apim-request-id: 30fc5a46-549f-4706-a4cc-ed009fceb90a

Strict-Transport-Security: max-age=31536000; includeSubDomains; preload

x-content-type-options: nosniff

Date: Wed, 27 Jun 2018 05:39:57 GMT

Content-Length: 388

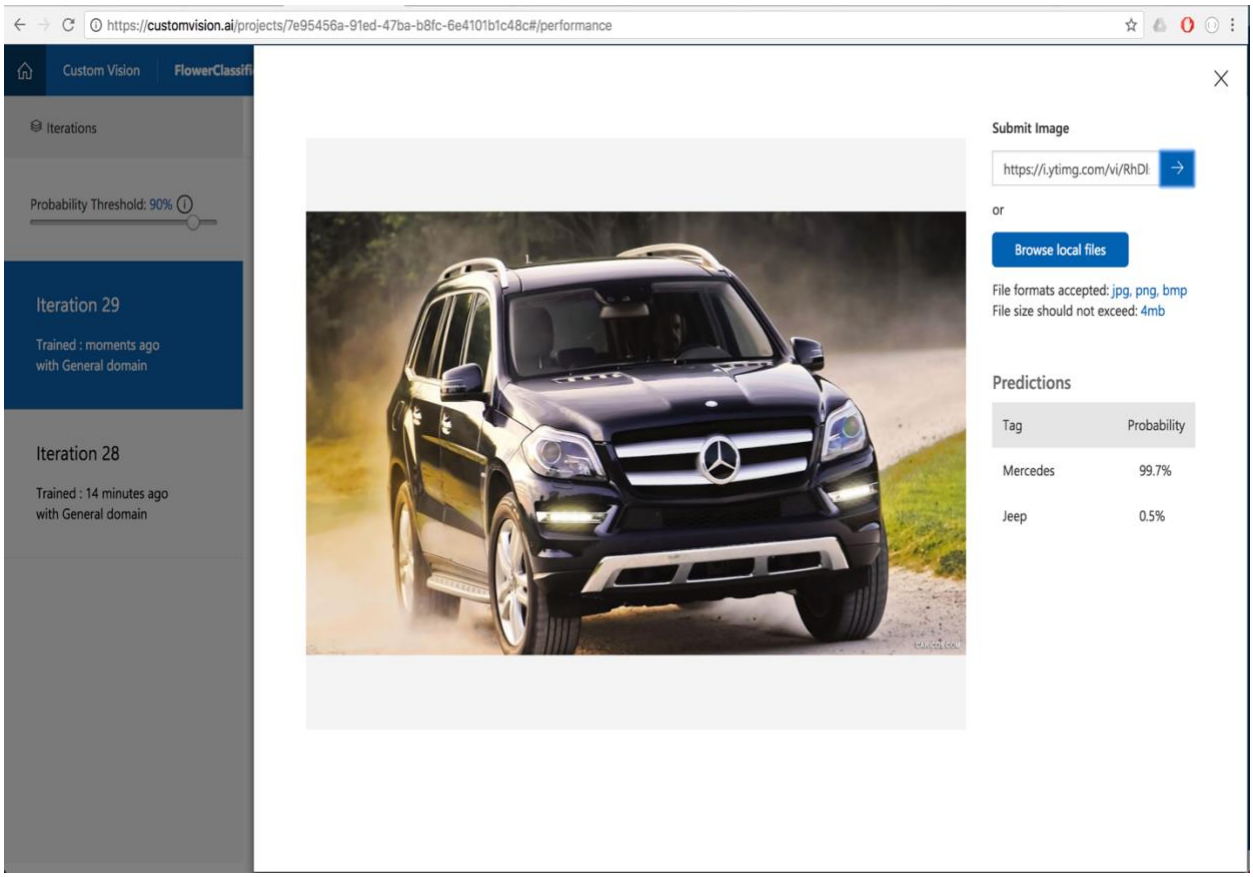
Content-Type: application/json; charset=utf-8

```
{
  "id": "30fc5a46-549f-4706-a4cc-ed009fceb90a",
  "project": "7e95456a-91ed-47ba-b8fc-6e4101b1c48c",
  "iteration": "81f53285-e3b4-44e3-885a-9988fc35f6c8",
  "created": "2018-06-27T05:39:57.2326194Z",
  "predictions": [{
    "probability": 0.9977333,
    "tagId": "b2af220e-a4f4-43b7-9c98-c13a6c9f8ca1",
    "tagName": "Mercedes"
  }, {
    "probability": 0.005058851,
    "tagId": "8e958793-dfd4-40c5-92ee-b861c7005b5e",
    "tagName": "Jeep"
  }]
}
```


This happened because we were able to create a better classification model by supplying more variety of image of each car category and our system was able to extract variety of

information or features from each category. The more feature list we have, the more chances that our test set will get greater précised result.

For the better results, it's always good to have more corelated inputs of each category and train system again and again. however, try to avoid overfitting the model as well. Because in case of overfitting, our model may fail to categories the input set if input set just outside the boundary.



Difficulties faced in the project:

(a) Initially understanding the problem was very difficult however TA's made good effort in setting up the examples and explaining the steps to build the classification model.

(b) Choosing the images of two different objects which are co-related but belong to two different categories was very difficult choice.

(c) Next building the model for iteration one was easy however precision and recall percentage was not good so we had to supply more accurate labeled images and train the system again. So, selecting and supplying the right images was challenge. I guess google image search and more trail-error concepts were very helpful in this case :).

(d) We had to reconfirm the confidence data with API. For this purpose, Microsoft provides cognitive services. Basically, online API calls to send http GET/POST requests. We just have to mention the project id, iteration id to uniquely identify the project and iterations.