---

layout: post

title: Customize Image Classifier Machine Learning Foundation Services

subtitle: A way to retrain the model on SAP Cloud Platform

date: 2019-10-07

author: Jiayi

header-img: img/post-bg-ios10.jpg

catalog: true

tags:

- SAP API

- Image Classification

---

## Step 01: Install the Cloud Foundry Command Line Interface (CLI)

You can download it via https://github.com/cloudfoundry/cli#downloads

#### Test the Cloud Foundry CLI

open your terminal and input CF

![](/img/ImageClassifierSCP/0002.png)

#### Connect the CLI to the cloud region

You can find the API Endpoint in "Subaccount - Overview" section in your SCP account.

![](/img/ImageClassifierSCP/0001.png)

#### Log in using the CLI

![](/img/ImageClassifierSCP/0003.png)

## Step 02: Install the Machine Learning foundation plugin for SAP Cloud Platform CLI

You can download it here: https://tools.hana.ondemand.com/#mlfoundation

## Install the plugin

```python

cf install-plugin -f <extract directory>/sapmlcli

Verify the plugin is installed properly

```

![](/img/ImageClassifierSCP/0004.png)

## Step 03: Create service instance and service key

Go to SCP, login your Cloud Foundry environment, click "space", then you will see service marketplace and service instances in the left panel.

SAP Leonardo Machine Learning Foundation is pre-installed in CF environment.

![](/img/ImageClassifierSCP/0005.png)

Create a new instance with standard plan

![](/img/ImageClassifierSCP/0006.png)

![](/img/ImageClassifierSCP/0007.png)

You can also create a new instance via Cloud Foundry Command Line Interface Method

```python

cf create-service <service name> <plan> <instance name>

cf create-service ml-foundation-trial-beta standard my-ml-foundation

```

\*\*The next step is to create service key in order to use it in POSTMAN\*\*

![](/img/ImageClassifierSCP/0008.png)

## Step 04: Prepare your environment for the SAP Leonardo Machine Learning foundation Image Classification Retraining scenario

Get the Service Key details

![](/img/ImageClassifierSCP/0009.png)

Set the Machine Learning foundation plugin Configuration

```python

cf sapml config set auth\_server <authentication URL>

cf sapml config set job\_api <JOB\_SUBMISSION\_API\_URL>

cf sapml config set retraining\_image\_api <IMAGE\_RETRAIN\_API\_URL>

cf sapml config set ml\_foundation\_service\_name ml-foundation-trial-beta

cf sapml fs init

```

![](/img/ImageClassifierSCP/0010.png)

## Step 05: Prepare and upload your Dataset for Image Classification Retraining

The training dataset locates in http://download.tensorflow.org/example\_images/flower\_photos.tgz

Download and prepare the training dataset

```python

import os

import shutil

import wget

import tarfile

import numpy as np

# cleanup before starting

if os.path.exists('./flower\_photos.tgz'): os.remove('./flower\_photos.tgz')

shutil.rmtree('./flower\_photos', ignore\_errors=True)

shutil.rmtree('./flowers', ignore\_errors=True)

# download the dataset file with flowers

archive = wget.download('http://download.tensorflow.org/example\_images/flower\_photos.tgz')

# extract the tar file content

archivetar = tarfile.open(archive, "r:gz")

archivetar.extractall()

archivetar.close()

# fill the data directory with 10 files for try, 90% of the remainder for training, 5% for validation and 5% for test

flowerdirs = ['daisy', 'dandelion', 'roses', 'sunflowers', 'tulips']

targetdirs = ['training', 'validation', 'test']

targetdirs\_split = [.8, .5, 1]

for flowerdir in flowerdirs:

# list all files in dir

files = [f for f in os.listdir('./flower\_photos/' + flowerdir)]

# create the try directory

os.makedirs('./flowers/try/' + flowerdir)

# move file to the try directory

random\_files = np.random.choice(files, 10)

for idx\_fname, fname in enumerate(random\_files):

ffname = os.path.join('./flower\_photos/' + flowerdir, fname)

shutil.move(ffname, './flowers/try/' + flowerdir)

files.remove(fname)

# for targetdir in targetdirs:

for idx\_targetdir, targetdir in enumerate(targetdirs):

# create the data directory

os.makedirs('./flowers/' + targetdir + '/' + flowerdir)

# move file to the targetdir directory

random\_files = np.random.choice(files, int(len(files) \* targetdirs\_split[idx\_targetdir]), replace=False)

for fname in random\_files:

ffname = os.path.join('./flower\_photos/' + flowerdir, fname)

shutil.move(ffname, './flowers/' + targetdir + '/' + flowerdir)

files.remove(fname)

if os.path.exists('./flower\_photos.tgz'): os.remove('./flower\_photos.tgz')

shutil.rmtree('./flower\_photos', ignore\_errors=True)

```

The below is the screenshot of the downloaded dataset

![](/img/ImageClassifierSCP/0011.png)

Once configured, you can now transfer the prepared dataset using the following commands:

\*\*go to the folder where you store the dataset\*\*, then use the below code to transfer from local storage to remote storage

```python

cf sapml fs put flowers/test/daisy/ flowers/test/daisy/

cf sapml fs put flowers/test/dandelion/ flowers/test/dandelion/

cf sapml fs put flowers/test/roses/ flowers/test/roses/

cf sapml fs put flowers/test/sunflowers/ flowers/test/sunflowers/

cf sapml fs put flowers/test/tulips/ flowers/test/tulips/

cf sapml fs put flowers/training/daisy/ flowers/training/daisy/

cf sapml fs put flowers/training/dandelion/ flowers/training/dandelion/

cf sapml fs put flowers/training/roses/ flowers/training/roses/

cf sapml fs put flowers/training/sunflowers/ flowers/training/sunflowers/

cf sapml fs put flowers/training/tulips/ flowers/training/tulips/

cf sapml fs put flowers/validation/daisy/ flowers/validation/daisy/

cf sapml fs put flowers/validation/dandelion/ flowers/validation/dandelion/

cf sapml fs put flowers/validation/roses/ flowers/validation/roses/

cf sapml fs put flowers/validation/sunflowers/ flowers/validation/sunflowers/

cf sapml fs put flowers/validation/tulips/ flowers/validation/tulips/

```

![](/img/ImageClassifierSCP/0012.png)

![](/img/ImageClassifierSCP/0013.png)

## Step 06: Execute the Image Classification Model Retraining Job

Create a JSON file in your local storage

```python

{

"dataset": "flowers",

"modelName": "flowers"

}

```

Configure the Image Retraining Job and Check the Image Retraining Job Status

cf sapml retraining job\_submit image\_retrain.json -m image

cf sapml retraining jobs -m image

![](/img/ImageClassifierSCP/0015.png)

```python

cf sapml fs get flowers-2019-10-07t2201z009744/retraining.log ./retrain.log

```

Open the retrain.log file in your favorite text editor.

![](/img/ImageClassifierSCP/0018.png)

## Step 07: Deploy the Image Classification Retrained Model

Once a job is completed, the model will be automatically stored in the model repository.

Only model store in the repository can then be deployed.

To check the list of model in the repository, you can execute the following command:

![](/img/ImageClassifierSCP/0014.png)

Then Submit your Model for Deployment

![](/img/ImageClassifierSCP/0016.png)

check that your model was properly deployed

![](img/ImageClassifierSCP/0017.png)

## Step 08: Consume the Image Classification Retrained Model

Just like the Image Classification service, the retrained Image Classification service calculates and returns a list of classifications along with their probabilities for a given image using your predefined categories.

The only difference is in the URL to be called, where you will need to append the following /models/{model name}/versions/{model version}.

In Postman, create an environment called my-ml-foundation

![](/img/ImageClassifierSCP/0019.png)

Prepare OAuth Token request

![](/img/ImageClassifierSCP/0020.png)

Use below code to test

pm.environment.set("OAuthToken", decodeURIComponent(pm.response.json().access\_token))

![](/img/ImageClassifierSCP/0021.png)

Now you have all needed

![](/img/ImageClassifierSCP/0022.png)

Now we can use image classification API to detect the below picture

![](/img/ImageClassifierSCP/394990940\_7af082cf8d\_n.jpg)

https://mlftrial-image-classifier.cfapps.eu10.hana.ondemand.com/api/v2/image/classification/models/flowers/versions/1

![](/img/ImageClassifierSCP/0025.png)

![](/img/ImageClassifierSCP/0026.png)

### 99% rose!

Use a picture from SAP TechEd to call face detection API

![](/img/ImageClassifierSCP/SAP\_TechEd\_LV2018\_10772.jpg)

![](/img/ImageClassifierSCP/0027.png)

![](/img/ImageClassifierSCP/0028.png)