



IBM Watson Studio – Visual Recognition Workshop

Prerequisites: IBM Cloud Setup, a Watson Studio service instance, IBM Cloud CLI, Python and Git. See the Setup and Installation Guide for more details.

Over the course of this workshop, you are going to be learning how to build a visual recognition model with Watson Studio, and then how to deploy an application using that model to the IBM Cloud.

NOTE: The sample data you will need is

- Training Data.zip
- Test Data.zip
- App Data.zip

You should be able to complete this tutorial in approximately 60 minutes, depending on loading and running times.

To get started, open the IBM Cloud site at <https://cloud.ibm.com>, sign in, and make sure that you have a Watson Studio Service Instance.



In this tutorial we will use **Visual Recognition** to tag and classify visual content using the Watson Visual Recognition service in this case it is dogs, the cats.zip data will be used to give the model a negative example class.

Part 1: Setting up the project

1. From the home screen of Watson Studio, click **Create a Project**.
2. Choose the **Visual Recognition** card.
3. For the Select a region field, select **US South**.
4. Click **Select** on the popup.
5. On the details page add the name of **"Finding Bruno" – Visual Recognition**. Here is an optional suggested description: "Investigating the Visual Recognition project service".

To learn more about the hosting services, see <https://bit.ly/2Q5YhvS> for more details

6. Scroll down to "Define Watson Visual Recognition" and click Add.
7. On the new page which is opened, create a new Visual Recognition service instance under the Lite plan.
8. Click **Create**.
9. On the confirm creation pop make the options default or what is listed here:
 - a. Region → US South
 - b. Plan → Lite
 - c. Resource Group → Default
 - d. Service name → This will be automatically generated
10. Click **Confirm**.
11. Return to the previous page and click the Refresh button to see your new service appear.
12. Click **Create**.
13. Name the model **Finding Bruno**.

Part 2: Adding Data

1. On the right side of the screen, find **Upload to project** and select **Browse**.
2. Add all of the **Training Data** zip files.
3. Drag the **Golden Retriever.zip**, **Husky.zip**, **German Shepherd.zip**, **Pug.zip** files from the right-hand side panel to the training area and this will create new training classes. Alternatively, you can select the radio buttons for each file and click the **Add to model** button.

This will import all the images into the model. Classes will be created for each separate compressed (.zip) files and will serve as positive examples. The General classifier returns classes from thousands of possible tags organized into categories and subcategories. Top-level categories include: animals; person and people-oriented information and activities; food; plants; sports; nature; transportation; and, many more.

Learn more on guidelines for training classifiers at: <https://bit.ly/2EbHNwa>.

4. Now upload the **Cats.zip** file.
5. And drag **Cats.zip** to the negative class box; this will make it the negative class.
6. Click the **Train Model** button. Training may take a few minutes to do.
7. When training is complete, click on the **Default Custom Model** tab in the top left corner of the page.
8. Click on the **Test** tab.
9. Either drag-and-drop or browse to add the **Testing Data** images to the workspace. The images will be evaluated, and a confidence score will be assigned to each class that the image is tested against.

A confidence factor of greater than .90 should provide a good level of certainty to the model. In this situation, most of the other classes will have confidence factors closer to 0.00.

If you have multiple classes with low-to-mid confidence factors, additional images may be needed to have a more accurate model.

10. Click on the **tab with the name of your project** in the top left corner of the page.
11. Click **Assets**. This will show you the imported assets and the model that you just created. Leave this open and start the next step.

Part 3: Running an app Locally Using Python and IBM cloud

Step 1: Run the app locally.

1. On your local computer, **open terminal or command prompt**, and navigate to the `dogs/sample-app` directory of the files you downloaded.
 - To find the directory in question, enter the command **"dir"** to see which directories you can get to, then enter **"cd"** plus the name of the directory you want to open. Keep using these two commands until you are in the directory ending in **"dogs/sample-app"**.
2. Enter one of these commands, depending on the system you are working in:
 - For **Windows**: `pip install -r requirements.txt`
 - For **Mac**, run this command instead: `sudo pip install --user -r requirements.txt`
3. After this command has run/installed, we navigate to the `server.py` file which is in `dogs/sample-app`.
4. Open the **server.py** file using a text editor, then go back to the **Watson Studio** page you have left open in your browser. You should be on the **Assets** page of your Visual Recognition project.
5. Under the Models tab, copy the **Model ID**.
6. Back in the **server.py** file, paste in the **Model ID** where it is indicated in the code (see picture below).

```
model_id = '' # <-- PASTE YOUR MODEL ID HERE
apikey   = '' # <-- PASTE YOUR APIKEY HERE
```

Step 2: The API key

1. In a new tab go to <https://cloud.ibm.com/>.
2. Under resource summary, **Click Services**.
3. Click the **Visual Recognition** service instance.



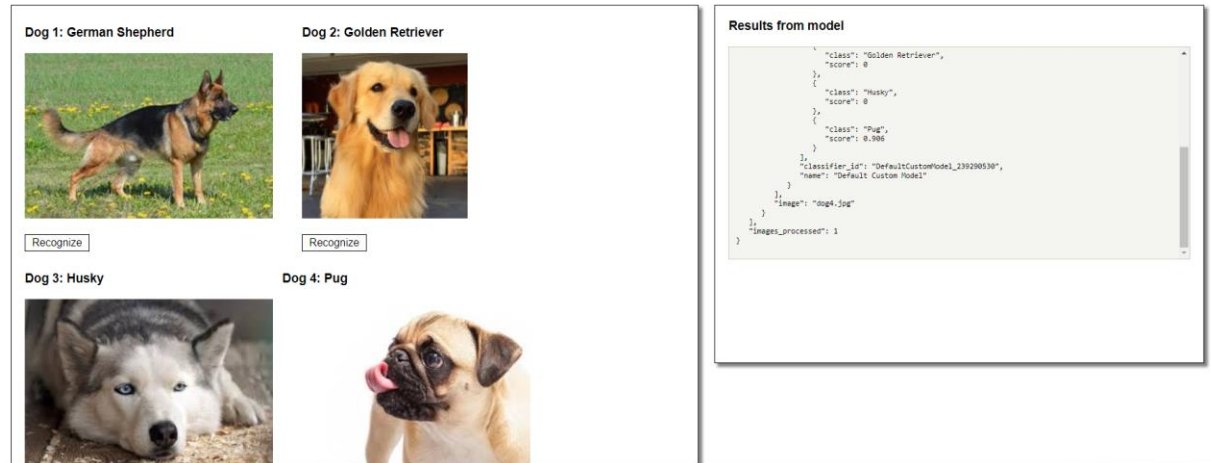
4. Copy the **API Key**.
5. Paste this in the **API key field** where indicated in the image earlier.

6. Close and then run the **server.py** file.
7. Enter **“localhost:8000”** into your browser. Your browser should appear like the image below.

Dogs

This web page demonstrates how you could implement an app that recognizes the dog breeds by using an AI model that has been trained to classify images of dogs:

- German Shepherd
- Golden Retriever
- Husky
- Pug



Step 4: Prepare the local app code

1. Open **manifest.yml**.
2. In the code find the line with **'app-name'** and replace this with a unique name.

The 'app-name' must be unique for every application to being pushed to the cloud world-wide.

Step 5: Deploy the app

1. Login to your IBM Cloud account via terminal or command prompt by enter the command **“ibmcloud login -sso”**
2. You will be prompted to open a window in a browser, **select 'Y' for Yes**, then copy the timed passcode and paste it into the terminal or command prompt.
3. Target the CloudFoundry API endpoint by entering the command **‘ibmcloud target --cf’**.
4. From within the app working directory (where the file server.py is located) push your app to the IBM Cloud by entering the command **‘ibmcloud app push’**.



Step 6: View the app online

1. You can save and close any of the files from this workshop you have open, because now you will go to <https://cloud.ibm.com/> in your browser.
2. Once logged in, from the **Resource Summary** card, click **Cloud Foundry Apps**.
3. In the accordion menu, open the resource which has the **unique app-name** which you put into manifest.yml earlier.
4. This is your application! It should have a green dot next to the name of your app that says **Running** next to it. To see the app, click **Visit App URL** next to the name of your app. ● Running [Visit App URL](#)

Over the course of this workshop, there have been two halves: building a visual recognition model and releasing it as an app to the world. For the first half, we got pictures of dogs and cats and taught our model how to distinguish between them and between breeds of dog. In the second half, we used IBM Cloud and Cloud Foundry to make our model into a functioning internet application.

See this link for more details on running Visual Recognition apps and APIs in Python:
<https://cloud.ibm.com/apidocs/visual-recognition?code=python>