CITS5551 &CITS5552 Software Engineering Design Project

Weed Detection - User Manual

Jiaqi Han 22292229

I. Introduction

For the weed detection project, I have deployed Mask_RCNN weed detection in the Google Cloud Platform. The user can use Jupyter Notebook interface to achieve all functions visually. I also provided some simple commands to achieve all functions through SSH server.

II. Terms Explanations

Staff: System staffs who responsible for routine maintenance and system upgrade.
User: System users for weed detection, like weed experts.
System Directory Tree: Code and data of the system saved in "Mask_RCNN" folder.
/Mask_RCNN folder

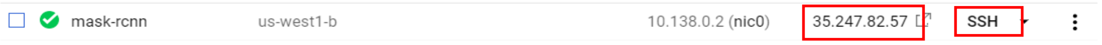
|__ mrcnn folder: contained the main code for Mask_RCNN algorithm and .h5 model for further retrain.
|__ samples folder: contained Weed_retrain.py and Weed_detection.py
|__ image folder: stored the farmer images
|__ logs folder: saved the retrained model and logs
|__ train_data folder: stored the training data for the model retrain.
|__ Farmer folder: saved the farm's information: farmer's model, dataset and images

III. Staff Guide

This section is the guide for staffs to maintenance the system and upgrade weed detection model.

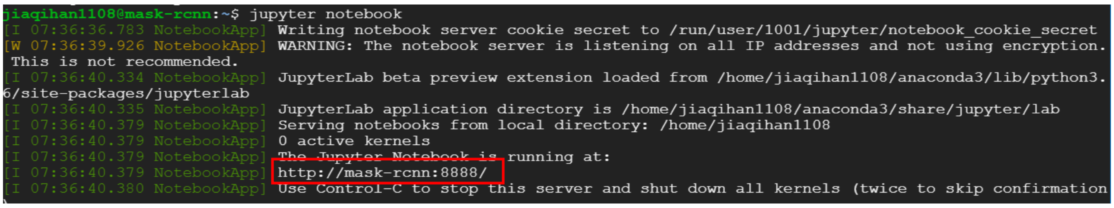
1. Connect to the system

Start the "mask-rcnn" instance for the system and get the instance running like figure1, then click on the "SSH" button to start the terminal interface for the system. The user can directly input commands in this terminal. The user also can connect to the system visually by Jupyter notebook like figure2. We have a Static external IP shown in figure1 for the project.



□ mask-rcnn us-west1-b 10.138.0.2 (nic0) 35.247.82.57 SSH

Figure 1: 'mask-rcnn' Weed Detection VM instances of Google Cloud platform with a Static External IP



```
jiaqihan1108@mask-rcnn:~$ jupyter notebook
[I 07:36:36.783 NotebookApp] Writing notebook server cookie secret to /run/user/1001/jupyter/notebook_cookie_secret
[W 07:36:39.926 NotebookApp] WARNING: The notebook server is listening on all IP addresses and not using encryption.
 This is not recommended.
[I 07:36:40.334 NotebookApp] JupyterLab beta preview extension loaded from /home/jiaqihan1108/anaconda3/lib/python3.
6/site-packages/jupyterlab
[I 07:36:40.335 NotebookApp] JupyterLab application directory is /home/jiaqihan1108/anaconda3/share/jupyter/lab
[I 07:36:40.379 NotebookApp] Serving notebooks from local directory: /home/jiaqihan1108
[I 07:36:40.379 NotebookApp] 0 active kernels
[I 07:36:40.379 NotebookApp] The Jupyter Notebook is running at:
[I 07:36:40.379 NotebookApp] http://mask-rcnn:8888/
[I 07:36:40.380 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation
```

Figure 2: Running Jupyter Notebook of the system

The Jupyter Notebook of the instances is running with Static External IP address (35.247.82.57), so the link is http:// 35.247.82.57:8888/.

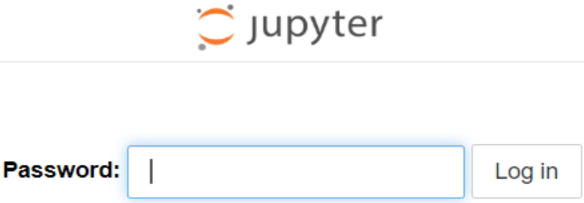
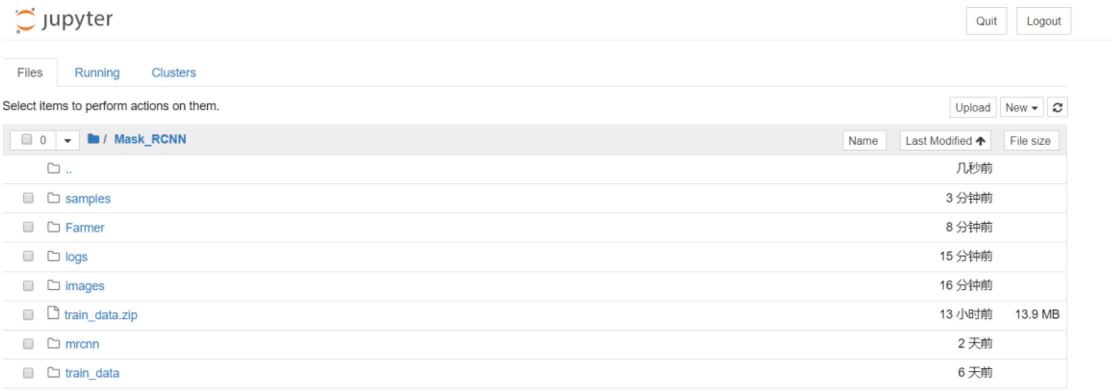


Figure 3: The Password of the system is “hanjiaqi”



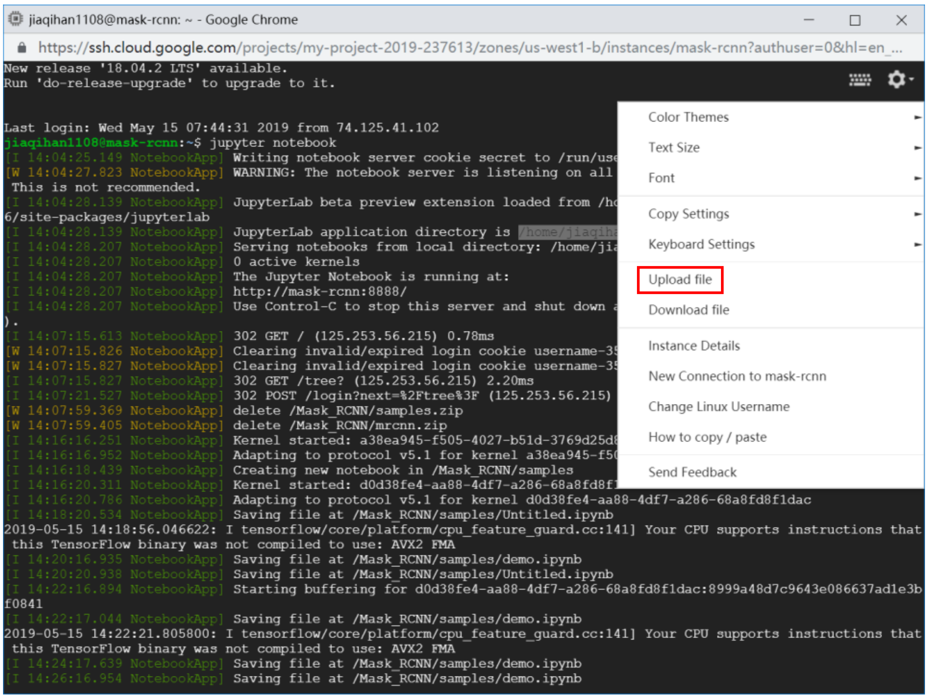
Select items to perform actions on them.

0 / Mask_RCNN

Name	Last Modified	File size
..	几秒前	
samples	3 分钟前	
Farmer	8 分钟前	
logs	15 分钟前	
images	16 分钟前	
train_data.zip	13 小时前	13.9 MB
mrcnn	2 天前	
train_data	6 天前	

Figure 4: The Jupyter Notebook of the Weed Detection System

2. Adding new training data

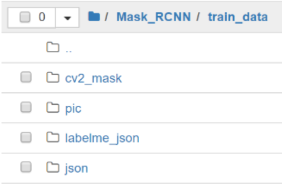

jiaqihan1108@mask-rcnn: ~ - Google Chrome
https://ssh.cloud.google.com/projects/my-project-2019-237613/zones/us-west1-b/instances/mask-rcnn?authuser=0&hl=en_...
Color Themes
Text Size
Font
Copy Settings
Keyboard Settings
Upload file
Download file
Instance Details
New Connection to mask-rcnn
Change Linux Username
How to copy / paste
Send Feedback

```
New release '18.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.


Last login: Wed May 15 07:44:31 2019 from 74.125.41.102
jiaqihan1108@mask-rcnn:~$ jupyter notebook
[I 14:04:25.149 NotebookApp] Writing notebook server cookie secret to /run/use
[W 14:04:27.823 NotebookApp] WARNING: The notebook server is listening on all
 This is not recommended.
[I 14:04:28.139 NotebookApp] JupyterLab beta preview extension loaded from /ho
6/site-packages/jupyterlab
[I 14:04:28.139 NotebookApp] JupyterLab application directory is /home/jiaqiha
[I 14:04:28.207 NotebookApp] Serving notebooks from local directory: /home/jia
[I 14:04:28.207 NotebookApp] 0 active kernels
[I 14:04:28.207 NotebookApp] The Jupyter Notebook is running at:
[I 14:04:28.207 NotebookApp] http://mask-rcnn:8888/
[I 14:04:28.207 NotebookApp] Use Control-C to stop this server and shut down a
).
[I 14:07:15.613 NotebookApp] 302 GET / (125.253.56.215) 0.78ms
[W 14:07:15.826 NotebookApp] Clearing invalid/expired login cookie username-35
[W 14:07:15.827 NotebookApp] Clearing invalid/expired login cookie username-35
[I 14:07:15.827 NotebookApp] 302 GET /tree? (125.253.56.215) 2.20ms
[I 14:07:21.527 NotebookApp] 302 POST /login?next=%2Ftree%3F (125.253.56.215)
[W 14:07:59.369 NotebookApp] delete /Mask_RCNN/samples.zip
[W 14:07:59.405 NotebookApp] delete /Mask_RCNN/mrcnn.zip
[I 14:16:16.251 NotebookApp] Kernel started: a38ea945-f505-4027-b51d-3769d25d8
[I 14:16:16.952 NotebookApp] Adapting to protocol v5.1 for kernel a38ea945-f50
[I 14:16:18.439 NotebookApp] Creating new notebook in /Mask_RCNN/samples
[I 14:16:20.311 NotebookApp] Kernel started: d0d38fe4-aa88-4df7-a286-68a8fd8f1
[I 14:16:20.786 NotebookApp] Adapting to protocol v5.1 for kernel d0d38fe4-aa88-4df7-a286-68a8fd8f1dac
[I 14:18:20.534 NotebookApp] Saving file at /Mask_RCNN/samples/Untitled.ipynb
2019-05-15 14:18:56.046622: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that
 this TensorFlow binary was not compiled to use: AVX2 FMA
[I 14:20:16.935 NotebookApp] Saving file at /Mask_RCNN/samples/demo.ipynb
[I 14:20:20.938 NotebookApp] Saving file at /Mask_RCNN/samples/Untitled.ipynb
[I 14:22:16.894 NotebookApp] Starting buffering for d0d38fe4-aa88-4df7-a286-68a8fd8f1dac:8999a48d7c9643e086637ad1e3b
f0841
[I 14:22:17.044 NotebookApp] Saving file at /Mask_RCNN/samples/demo.ipynb
2019-05-15 14:22:21.805800: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions that
 this TensorFlow binary was not compiled to use: AVX2 FMA
[I 14:24:17.639 NotebookApp] Saving file at /Mask_RCNN/samples/demo.ipynb
[I 14:26:16.954 NotebookApp] Saving file at /Mask_RCNN/samples/demo.ipynb
```

Figure 5: Upload train dataset by SSH

Because of the large size of the training dataset, using Jupyter Notebook’s upload function always costs too much time. A better way is to upload the dataset by SSH service and unzip the compressed package with command “unzip”. Remember to move dataset to right location, since the default upload path is “home/jiaqihan1108/”.



□ 0 ▾ / Mask_RCNN / train_data

..

□ cv2_mask

□ pic

□ labelme_json

□ json

Figure 6: Training dataset content (Huan's Part)

The quality of the training data (especially the mask quality) can be check with the "Mask_RCNN_testing.ipynb" in the samples folder. If you get the mask like figure7 with class "Weed", this training data can be used for model re-train.

```
image_id 12
```

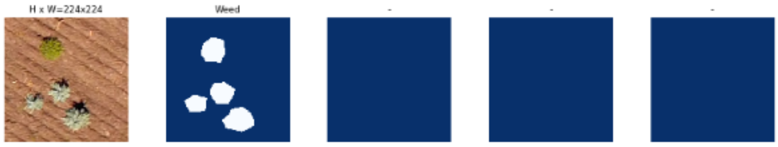
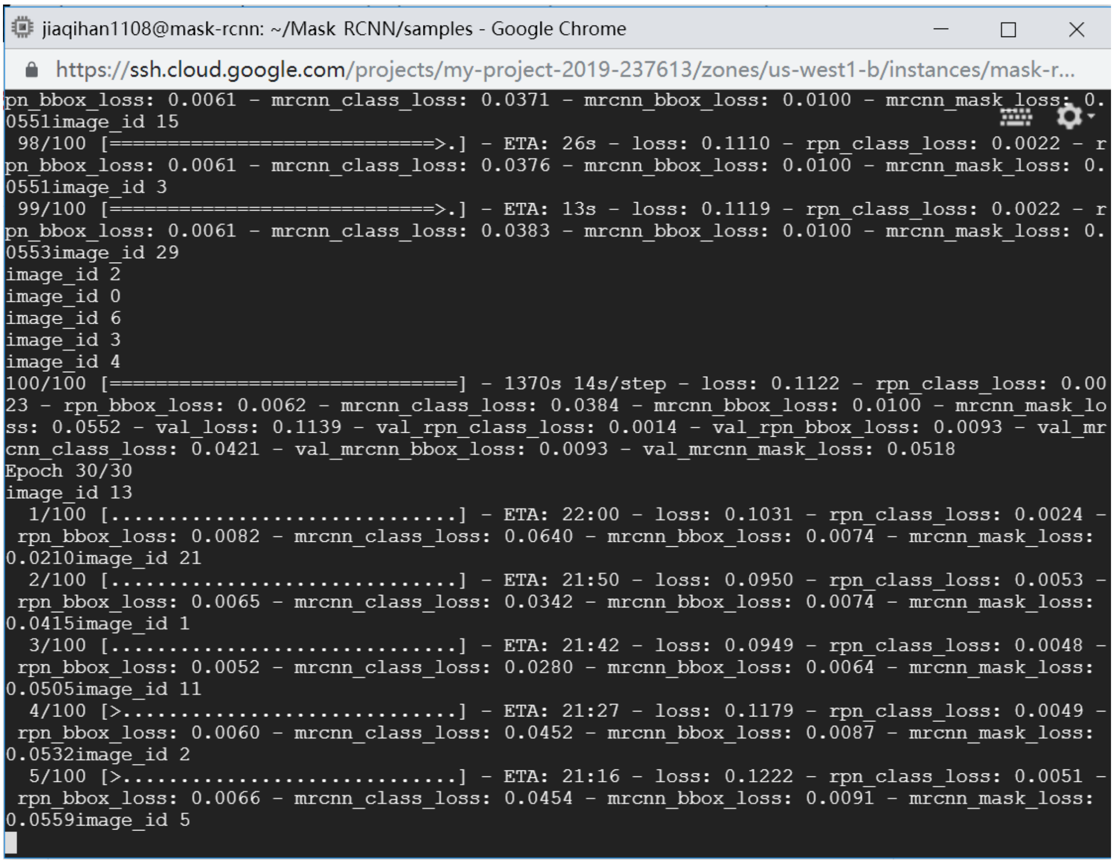


Figure 7: Testing the quality of the training dataset

3. Re-training Model



jiaqihan1108@mask-rcnn: ~/Mask RCNN/samples - Google Chrome

https://ssh.cloud.google.com/projects/my-project-2019-237613/zones/us-west1-b/instances/mask-r...

```
pn_bbox_loss: 0.0061 - mrcnn_class_loss: 0.0371 - mrcnn_bbox_loss: 0.0100 - mrcnn_mask_loss: 0.
0551image_id 15
 98/100 [============================>.] - ETA: 26s - loss: 0.1110 - rpn_class_loss: 0.0022 - r
pn_bbox_loss: 0.0061 - mrcnn_class_loss: 0.0376 - mrcnn_bbox_loss: 0.0100 - mrcnn_mask_loss: 0.
0551image_id 3
 99/100 [============================>.] - ETA: 13s - loss: 0.1119 - rpn_class_loss: 0.0022 - r
pn_bbox_loss: 0.0061 - mrcnn_class_loss: 0.0383 - mrcnn_bbox_loss: 0.0100 - mrcnn_mask_loss: 0.
0553image_id 29
image_id 2
image_id 0
image_id 6
image_id 3
image_id 4
100/100 [==============================] - 1370s 14s/step - loss: 0.1122 - rpn_class_loss: 0.00
23 - rpn_bbox_loss: 0.0062 - mrcnn_class_loss: 0.0384 - mrcnn_bbox_loss: 0.0100 - mrcnn_mask_lo
ss: 0.0552 - val_loss: 0.1139 - val_rpn_class_loss: 0.0014 - val_rpn_bbox_loss: 0.0093 - val_mr
cnn_class_loss: 0.0421 - val_mrcnn_bbox_loss: 0.0093 - val_mrcnn_mask_loss: 0.0518
Epoch 30/30
image_id 13
  1/100 [..............................] - ETA: 22:00 - loss: 0.1031 - rpn_class_loss: 0.0024 -
 rpn_bbox_loss: 0.0082 - mrcnn_class_loss: 0.0640 - mrcnn_bbox_loss: 0.0074 - mrcnn_mask_loss:
0.0210image_id 21
  2/100 [..............................] - ETA: 21:50 - loss: 0.0950 - rpn_class_loss: 0.0053 -
 rpn_bbox_loss: 0.0065 - mrcnn_class_loss: 0.0342 - mrcnn_bbox_loss: 0.0074 - mrcnn_mask_loss:
0.0415image_id 1
  3/100 [..............................] - ETA: 21:42 - loss: 0.0949 - rpn_class_loss: 0.0048 -
 rpn_bbox_loss: 0.0052 - mrcnn_class_loss: 0.0280 - mrcnn_bbox_loss: 0.0064 - mrcnn_mask_loss:
0.0505image_id 11
  4/100 [>.............................] - ETA: 21:27 - loss: 0.1179 - rpn_class_loss: 0.0049 -
 rpn_bbox_loss: 0.0060 - mrcnn_class_loss: 0.0452 - mrcnn_bbox_loss: 0.0087 - mrcnn_mask_loss:
0.0532image_id 2
  5/100 [>.............................] - ETA: 21:16 - loss: 0.1222 - rpn_class_loss: 0.0051 -
 rpn_bbox_loss: 0.0066 - mrcnn_class_loss: 0.0454 - mrcnn_bbox_loss: 0.0091 - mrcnn_mask_loss:
0.0559image_id 5
```

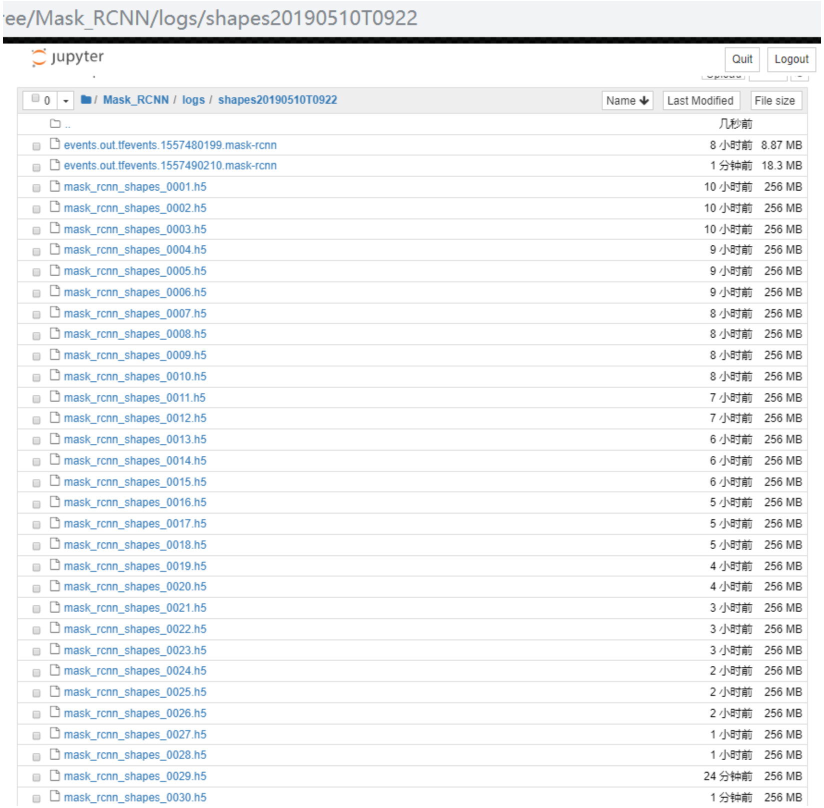
Figure 8: Weed Detection Model Re-train

Directly run the python file "./Weed_retrain.py" to re-train the model in the SSH terminal interface. Firstly, get into the python file's path "Mask_RCNN/samples/" using "cd Mask_RCNN/samples/" command; Then check the training dataset path, if necessary, change the default dataset path: "dataset_root_path =

os.path.join(ROOT_DIR, "train_data")" using "vim" command; Finally, retrain the model using command "./Weed_retrain.py". The model starts to retrain when shown like figure 8 and this stage will cost a long time more than 5 hours.

What's more, the default orignal trained weights file is mask_rcnn_coco.h5 model. If you want to train the model with the last Weed Detection model, just change the "init_with = "coco"" to "init_with = "last"".

The retrained model will be saved automatically in the logs folder in epoch, shown like figure 9, the "mask_rcnn_shapes_0030" is the model we what. Then the staff can use this new model to detect the weed more accurately.



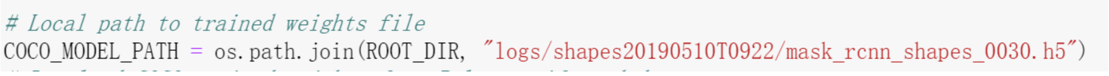
0 / Mask_RCNN / logs / shapes20190510T0922

Name	Last Modified	File size
..	几秒前	
events.out.tfevents.1557480199.mask-rcnn	8 小时前	8.87 MB
events.out.tfevents.1557490210.mask-rcnn	1 分钟前	18.3 MB
mask_rcnn_shapes_0001.h5	10 小时前	256 MB
mask_rcnn_shapes_0002.h5	10 小时前	256 MB
mask_rcnn_shapes_0003.h5	10 小时前	256 MB
mask_rcnn_shapes_0004.h5	9 小时前	256 MB
mask_rcnn_shapes_0005.h5	9 小时前	256 MB
mask_rcnn_shapes_0006.h5	9 小时前	256 MB
mask_rcnn_shapes_0007.h5	8 小时前	256 MB
mask_rcnn_shapes_0008.h5	8 小时前	256 MB
mask_rcnn_shapes_0009.h5	8 小时前	256 MB
mask_rcnn_shapes_0010.h5	8 小时前	256 MB
mask_rcnn_shapes_0011.h5	7 小时前	256 MB
mask_rcnn_shapes_0012.h5	7 小时前	256 MB
mask_rcnn_shapes_0013.h5	6 小时前	256 MB
mask_rcnn_shapes_0014.h5	6 小时前	256 MB
mask_rcnn_shapes_0015.h5	6 小时前	256 MB
mask_rcnn_shapes_0016.h5	5 小时前	256 MB
mask_rcnn_shapes_0017.h5	5 小时前	256 MB
mask_rcnn_shapes_0018.h5	5 小时前	256 MB
mask_rcnn_shapes_0019.h5	4 小时前	256 MB
mask_rcnn_shapes_0020.h5	4 小时前	256 MB
mask_rcnn_shapes_0021.h5	3 小时前	256 MB
mask_rcnn_shapes_0022.h5	3 小时前	256 MB
mask_rcnn_shapes_0023.h5	3 小时前	256 MB
mask_rcnn_shapes_0024.h5	2 小时前	256 MB
mask_rcnn_shapes_0025.h5	2 小时前	256 MB
mask_rcnn_shapes_0026.h5	2 小时前	256 MB
mask_rcnn_shapes_0027.h5	1 小时前	256 MB
mask_rcnn_shapes_0028.h5	1 小时前	256 MB
mask_rcnn_shapes_0029.h5	24 分钟前	256 MB
mask_rcnn_shapes_0030.h5	1 分钟前	256 MB

Figure 8: Weed Detection Model Re-train result saved in logs folder

4. Updating new model for weed detection

Change the model path in "Weed_detection.ipynb" like figure9.

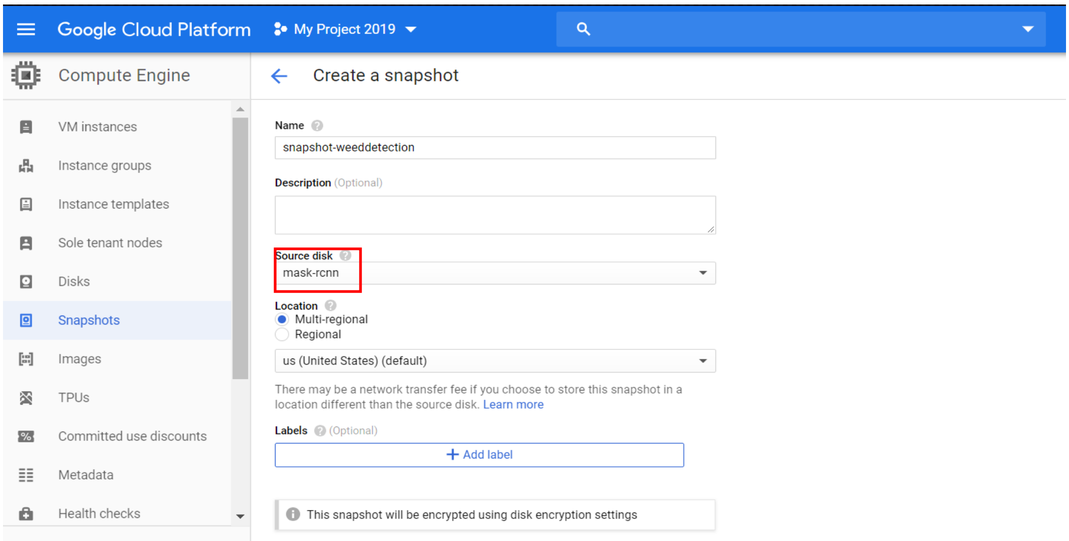


```
# Local path to trained weights file
COCO_MODEL_PATH = os.path.join(ROOT_DIR, "logs/shapes20190510T0922/mask_rcnn_shapes_0030.h5")
```

Figure 9: Add New Weed Detection Model for weed detecting

5. Restoring the instance

Through snapshots we can restore and recreate the instance. A snapshot should be created once the instance is upgraded to a new version or implemented new functions. Create a snapshot as shown in Figure 10.



Create a snapshot

Name
snapshot-weeddetection

Description (Optional)

Source disk
mask-rcnn

Location
● Multi-regional
○ Regional

us (United States) (default)

There may be a network transfer fee if you choose to store this snapshot in a location different than the source disk. Learn more

Labels (Optional)

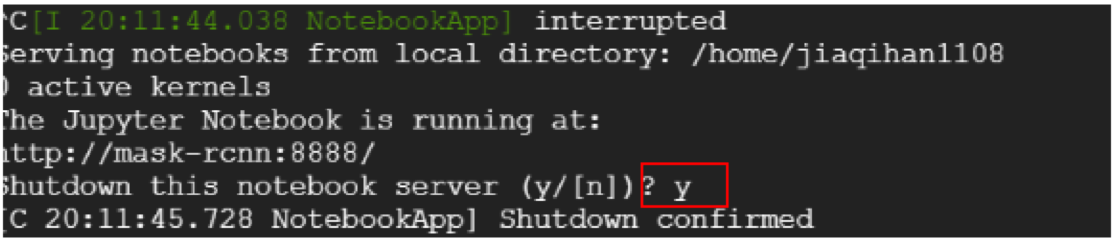
+ Add label

This snapshot will be encrypted using disk encryption settings

Figure 10: Create Snapshot of mask-rcnn instance

6. Shutting down the system

Shutting down the Jupyter Notebook with “Ctrl + c” and input “y” as shown in figure 11. Finally, stop the instance. What’s more, when you shut down the instance, the “Weed Detect” button on the website will be invalid.



```
^C[I 20:11:44.038 NotebookApp] interrupted
Serving notebooks from local directory: /home/jiaqihan1108
) active kernels
The Jupyter Notebook is running at:
nttp://mask-rcnn:8888/
Shutdown this notebook server (y/[n])? y
[C 20:11:45.728 NotebookApp] Shutdown confirmed
```

Figure 11: Shut down the Jupyter Notebook

IV. User

This section is the guide for users who use this system for weed detection.

1. Connecting to the system and shutting down the system

Mentioned in III.1 and III.6

2. Upload farmer images for detection

The User can upload farm images through Jupyter notebook one by one in the Mask_RCNN/image folder or upload compressed images’ package and unzip the package using “unzip file_name.zip” command, just as we mentioned in III.2

For better user experience, you also can create new folder with famer’s name in the “Farmer” folder, and then upload all farmer’s images in his/her folder.

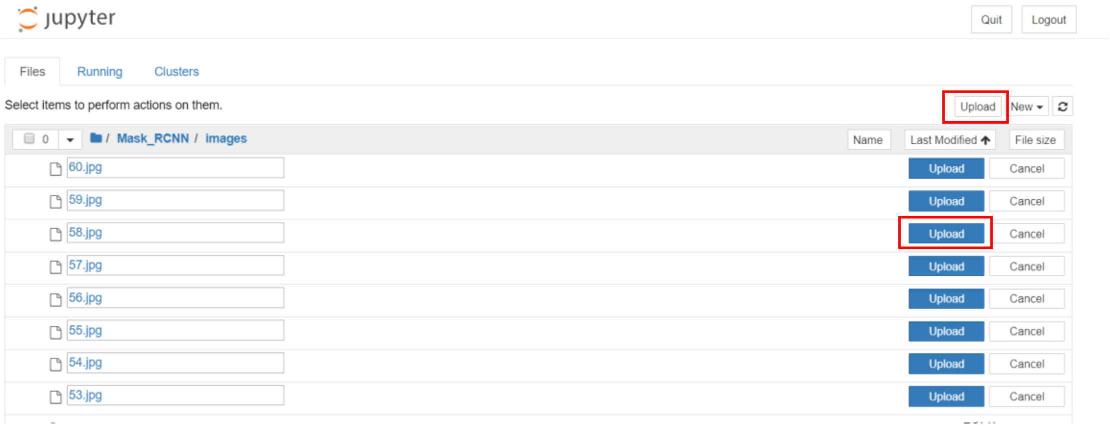


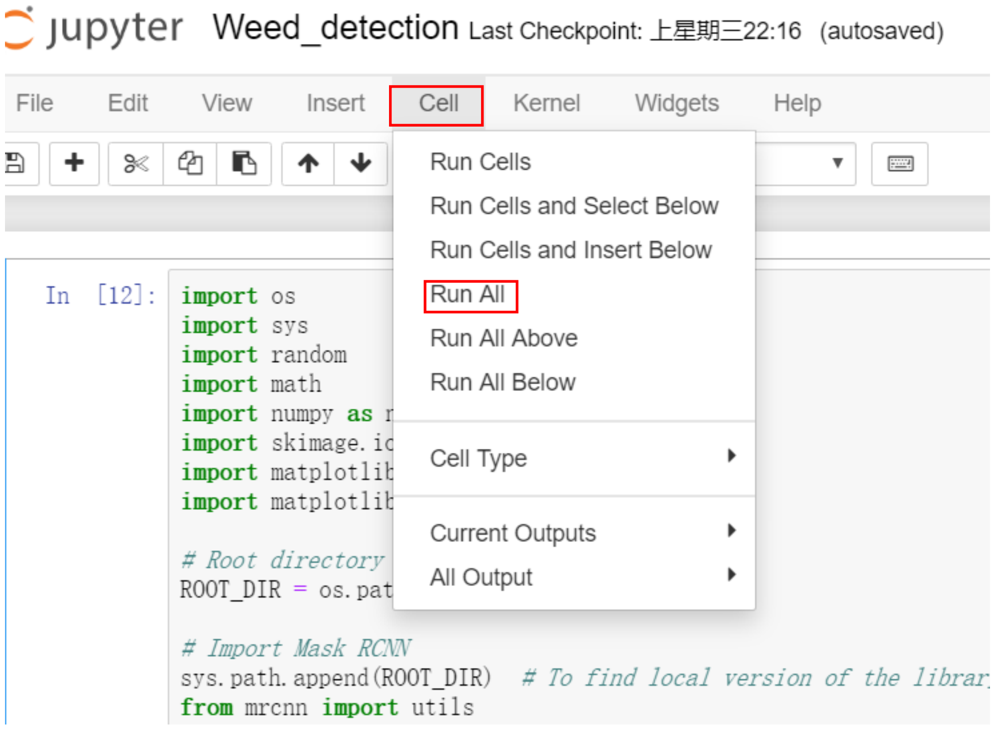
Figure 12: Upload farmer images

3. Running the detection function

The file for detection is saved in the samples folder named as Weed_detection.ipynb, click it and it will open automatically.

Setting the path of detection images, run detection on the farmer's image folder: IMAGE_DIR = os.path.join(ROOT_DIR, "Farmer/farmer_folder_name/images") → Click "Cell" →Click "Run All" → Wait until all In[*] become In[number]

The detection results will show directly at the bottom of this page.



```
In [12]: import os
         import sys
         import random
         import math
         import numpy as r
         import skimage.ic
         import matplotlib
         import matplotlib

         # Root directory
         ROOT_DIR = os.pat

         # Import Mask RCNN
         sys.path.append(ROOT_DIR)  # To find local version of the librar
         from mrcnn import utils
```

Figure 13: Upload farmer images

4. Analyzing the detection results and save the result in local machine

The Weed area will be covered by different color like shown in figure14, right-click and choose “save” to save it on local PC.

The class of detection and the predictive value is shown in the left-up corner of the bounding box. The weed experts can analyze the detection results to decide whether the system staffs need to retrain the model for this farmer, and provide feedback to staffs and farmers.

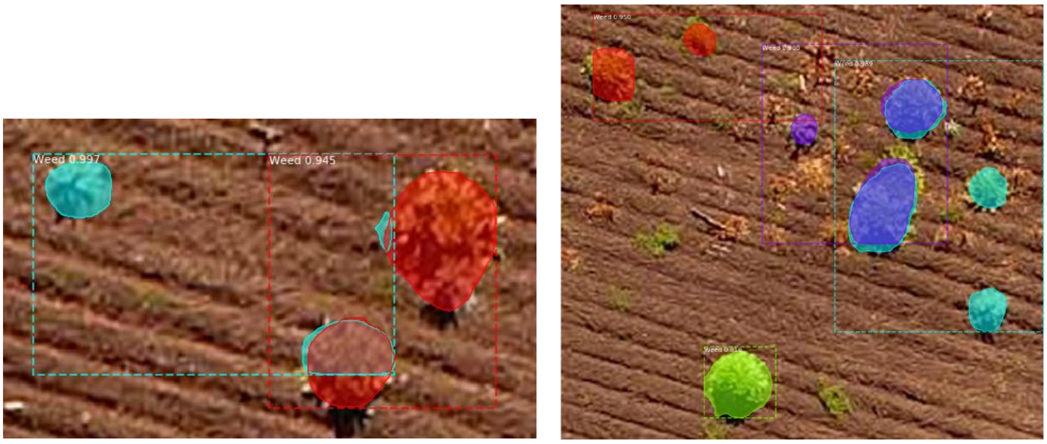
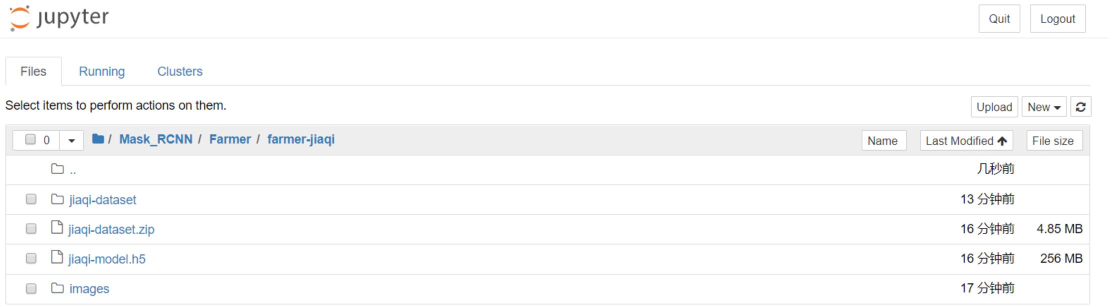


Figure 14: Get the final detection results

5. Farmer Folder

Then organize all the farmer information into Farmer folder, for providing better services, as shown in figure 15:



Select items to perform actions on them.

/ Mask_RCNN / Farmer / farmer-jiaqi

Name	Last Modified	File size
..	几秒前	
jiaqi-dataset	13 分钟前	
jiaqi-dataset.zip	16 分钟前	4.85 MB
jiaqi-model.h5	16 分钟前	256 MB
images	17 分钟前	

Figure 15: The farmer Jiaqi’s folder