



Litter Detection Algorithm: [User Manual](#)

[Table of Contents](#)

1. Introduction	2
2. Tools	2
3. How to Interface the AWS ec2 Server	2
3.1 Logging on to the server	2
3.1.1 Windows	2
3.1.2 Mac	4
3.2 Initial configuration	5
3.3 Using Jupyter Notebook	6
3.4 Running the algorithm	7

1. Introduction

FixIT worked with Keep America Beautiful to create a Litter Detection Algorithm that will identify litter, visually, when given a particular image. This manual was written as a user-friendly guide for anyone that is interested in using our litter detection algorithm with no prior experience in computer science or programming.

2. Tools

TensorFlow

Amazon Web Services

- P2 Large Ubuntu Server

Jupyter Notebook

Windows:

- PuTTY

Mac:

- Terminal

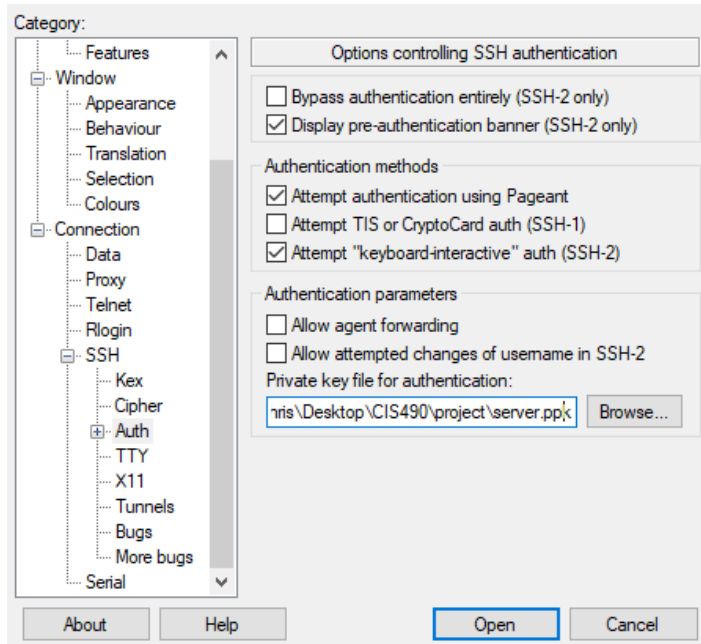
3. How to Interface the AWS ec2 Server

3.1 Logging on to the server

3.1.1 Windows

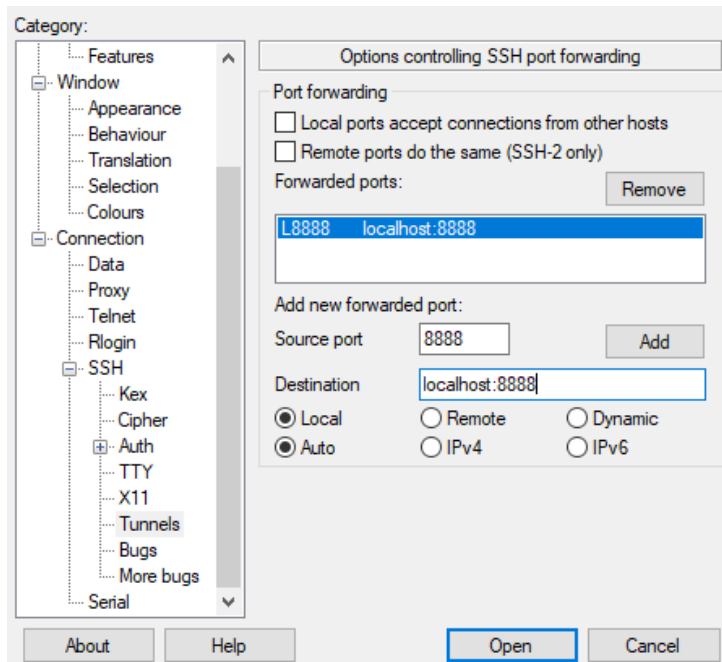
For the purposes of this project, Putty was used on Windows computers to interface the ec2 server.

1. Download and install Putty from <https://www.putty.org/>
2. Load in the private key under [Connection > SSH > Auth]
 - a. Click "browse" and select your private key file



3. Setup SSH tunneling under [Connection > SSH > Tunnels]. Enter these values:

Source Port: 8888
Destination: localhost:8888



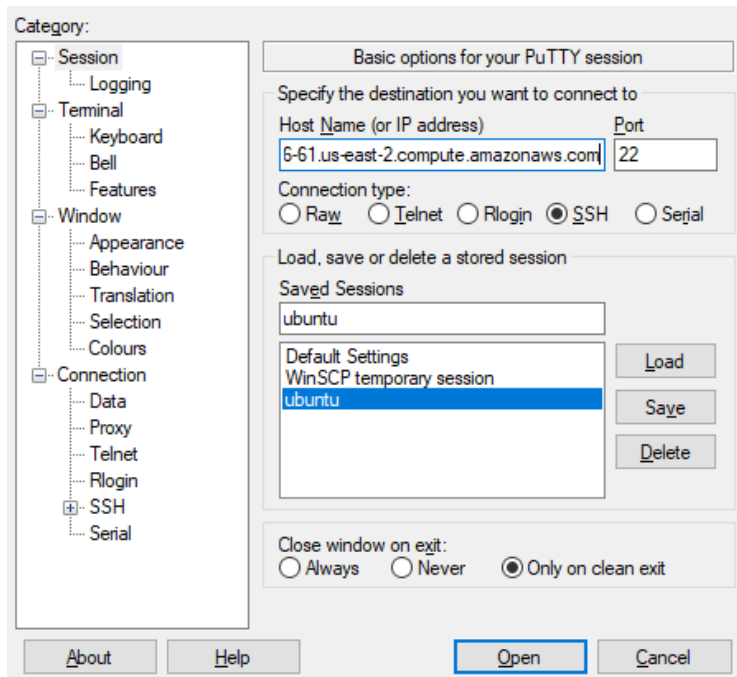
This is needed for jupyter notebook to work correctly.

4. Enter the server connection details under [Session]

IP: ec2-18-218-116-61.us-east-2.compute.amazonaws.com

Port: 22

Connection Type: SSH



5. Save the session and click open to login to the server.
6. Login as **ubuntu**.

3.1.2 Mac

For the purposes of this project, Terminal, a native Mac application, was used on Mac computers to interface the ec2 server.

1. In the Terminal application, change directories to the location of the private key file that you created when you launched the instance. This private key file will have the format: *filename.pem*. Execute the following command into the Terminal window:

```
cd path/to/file.pem
```

2. Use the **chmod** command to make sure that your private key file isn't publicly viewable. For example, if the name of your private key file is *my-key-pair.pem*, use the following command:

```
chmod 400 /path/my-key-pair.pem
```

3. Use the **ssh** command to connect to the instance. You specify the private key (.pem) file and `user_name@public_dns_name`. For example, if you used an Amazon Linux AMI, `user_name` is `ec2-user`. If you used Ubuntu, `user_name` is `ubuntu`. Replace `<ec2-domain>` with your actual domain address.

```
ssh -i "FixItServer.pem" -L 8157:127.0.0.1:8888
user_name@<public_dns_name>
```

The first time you login, you should see the following response:

```
The authenticity of host 'ec2-198-51-100-1.compute-1.amazonaws.com
(10.254.142.33)'
can't be established.
RSA key fingerprint is
1f:51:ae:28:bf:89:e9:d8:1f:25:5d:37:2d:7d:b8:ca:9f:f5:f1:6f.
Are you sure you want to continue connecting (yes/no)?
```

Type **yes** to connect. After your initial login, after you execute the `ssh` command, you will be automatically logged into the server.

3.2 Initial configuration

1. Activate Tensorflow environment by using the following command:

```
source ~/tensorflow/bin/activate
```

2. Change directories to the `models/research` directory.

```
cd ~/models/research
```

3. Compile the `protoc` library. Execute the following command:

```
protoc object_detection/protos/*.proto --python_out=.
```

4. Export the appropriate Python path. Execute the following command:

```
export PYTHONPATH=$PYTHONPATH:`pwd`:`pwd`/slim
```

3.3 Using Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. This application will be used to execute the algorithm on chosen test images. These test images are located in the `~/models/research/object_detection/test_images` folder.

1. Change directories `models/research/object_detection` by using the following command:

```
cd ~/models/research/object_detection
```

2. Run Jupyter Notebook using the following command:

```
jupyter-notebook
```

You should see the following results:

```
Serving notebooks from local directory:
/home/ubuntu/models/research/object_detection
[I 22:00:04.373 NotebookApp] 0 active kernels
[I 22:00:04.373 NotebookApp] The Jupyter Notebook is running at:
[I 22:00:04.373 NotebookApp]
http://localhost:8888/?token=dec0e2a8d30add83c31c735213336503524e16e9f952b5b5
[I 22:00:04.373 NotebookApp] Use Control-C to stop this server and shut down all
kernels (twice to skip confirmation).
[W 22:00:04.373 NotebookApp] No web browser found: could not locate runnable
browser.
[C 22:00:04.373 NotebookApp]
```

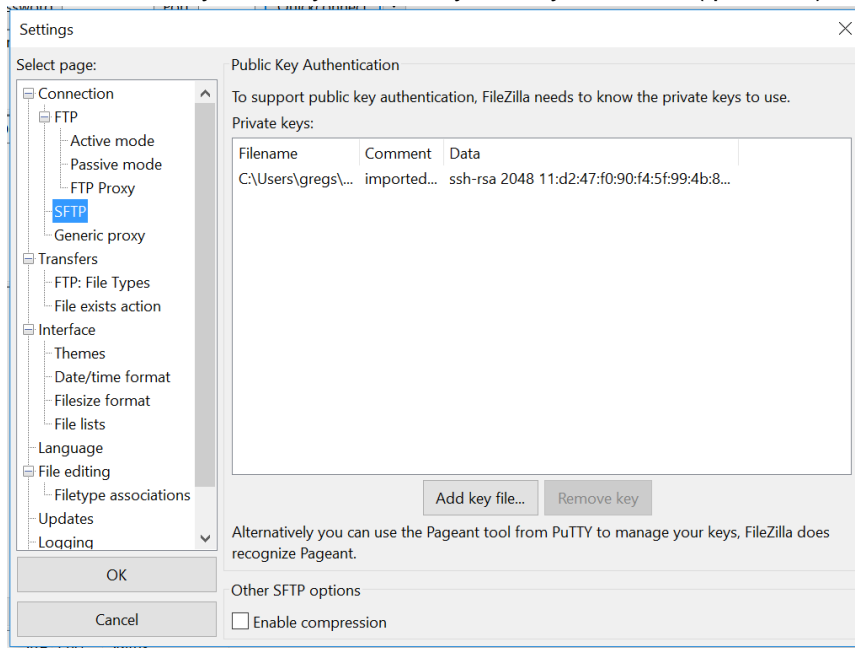
Copy/paste this **URL** into your browser when you connect for the first time,
to login with a token:

```
http://localhost:8888/?token=dec0e2a8d30add83c31c735213336503524e16e9f952b5b5
```

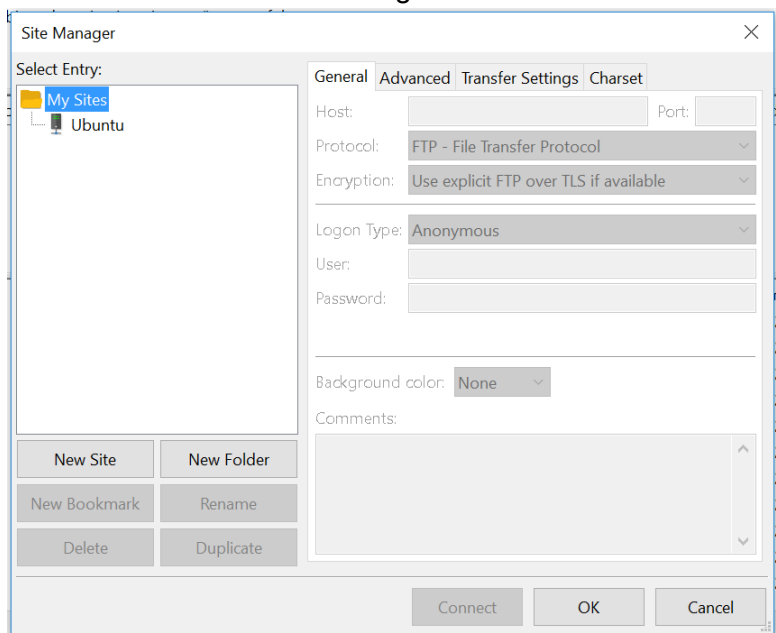
3. In any web browser type **localhost:8157** to access the Jupyter Notebook interface. Copy the token string (i.e. the red bolded line of text bolded in the above snippet) into the password textbox to login.

3.4 Running the algorithm

1. Choose images that you would like to test. It is important that the format needs to be .jpg.
2. Download a file transfer client, e.g. FileZilla. You need this client to upload your test images to the server.
3. Go to Edit > Settings and then to SFTP on the left side. Click “Add Key file...” and go to the directory where you saved your key to server (.pem file) and select it.



4. Go to File > Site Manager and click “New Site”.

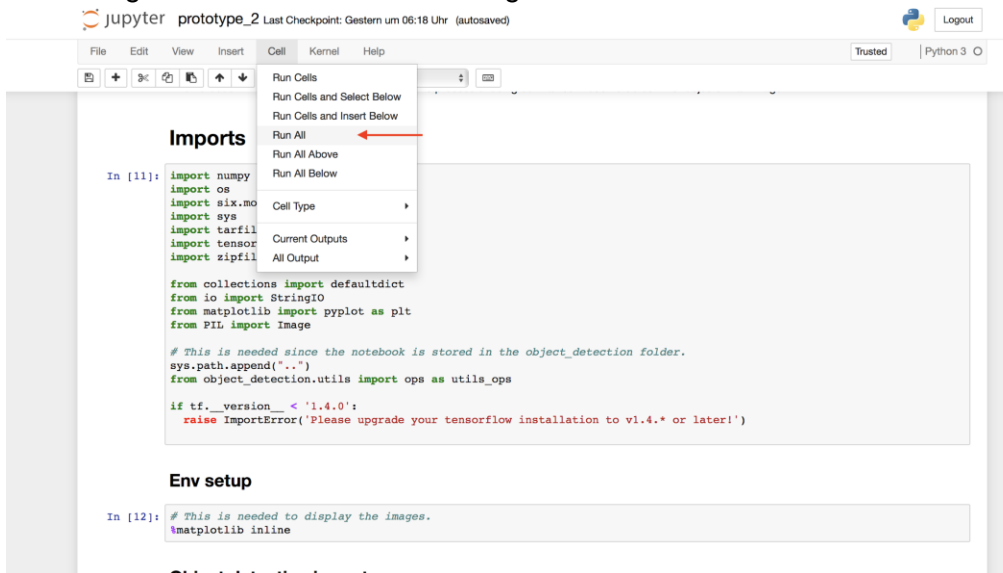


5. The “Host” is the domain of your aws server.
6. For “Protocol” select “SFTP”.
7. Choose “Normal” for “Logon Type”.
8. The “User” is the same username you use to log on to the server.

9. Type in your password for the server if you have one otherwise just leave it blank.
10. Press “Connect” to connect to the server
11. The left side it shows all the files saved locally on your PC and select the directory where you saved your test images here.

Local site: C:\Users\gregs\OneDrive\Documents\CIS 490\Schultz pictures\				Remote site: /home/ubuntu/models/research/object_detection/test_images					
<div><div></div><div>Schultz pictures</div><div>Timesheets</div><div>windows_v1.6.0</div><div>CIS341</div><div>CS 211 ppt</div><div>CS 443</div></div>				<div><div>home</div><div>ubuntu</div><div>.cache</div><div>.config</div><div>.gnupg</div><div>.ipython</div></div>					
Filename	Filesize	Filetype	Last modified	Filename	Filesize	Filetype	Last mod...	Permissions	Owner...
..				image4.jpg	60,437	JPG File	4/24/201...	-rwxr-xr-x	ubunt...
Test2.jpg	4,583,...	JPG File	4/26/2018 ...	image5.jpg	88,010	JPG File	4/24/201...	-rwxr-xr-x	ubunt...
Test5.jpg	4,725,...	JPG File	4/26/2018 ...	image6.jpg	48,530	JPG File	4/24/201...	-rw-rw-r--	ubunt...
Test4.jpg	6,693,...	JPG File	4/26/2018 ...	image7.jpg	44,402	JPG File	4/24/201...	-rw-rw-r--	ubunt...
Test3.jpg	5,708,...	JPG File	4/26/2018 ...	image8.jpg	73,073	JPG File	4/24/201...	-rw-rw-r--	ubunt...
Test1.jpg	6,222,...	JPG File	4/26/2018 ...	image9.jpg	1,188,...	JPG File	4/24/201...	-rw-rw-r--	ubunt...
				Test1.jpg	6,222,...	JPG File	4/26/201...	-rw-rw-r--	ubunt...
				Test2.jpg	4,583,...	JPG File	4/26/201...	-rw-rw-r--	ubunt...
				Test3.jpg	5,708,...	JPG File	4/26/201...	-rw-rw-r--	ubunt...
				Test4.jpg	6,693,...	JPG File	4/26/201...	-rw-rw-r--	ubunt...
				Test5.jpg	4,725,...	JPG File	4/26/201...	-rw-rw-r--	ubunt...

12. On the right side it shows all the files saved on the server. Go to home > ubuntu > models > research > object_detection > test_images. Here all the images that are executed by the algorithm are saved.
13. Move the images from your PC into that folder.
14. Open the Jupyter Notebook in your Browser as explained in the steps above. Navigate to the appropriate Notebook file (i.e. file ends in .ipynb). Then go to Cells > Run All to run the algorithm with the selected images



```

jupyter prototype_2 Last Checkpoint: Gestern um 06:18 Uhr (autosaved)

File Edit View Insert Cell Kernel Help Trusted Python 3

Run Cells
Run Cells and Select Below
Run Cells and Insert Below
Run All
Run All Above
Run All Below
Cell Type
Current Outputs
All Output

Imports
In [11]: import numpy
import os
import six
import sys
import tarfile
import tensorflow
import zipfile

from collections import defaultdict
from io import StringIO
from matplotlib import pyplot as plt
from PIL import Image

# This is needed since the notebook is stored in the object_detection folder.
sys.path.append("..")
from object_detection.utils import ops as utils_ops

if tf.__version__ < '1.4.0':
    raise ImportError('Please upgrade your tensorflow installation to v1.4.* or later!')

Env setup
In [12]: # This is needed to display the images.
matplotlib inline

Object detection imports

```

