To prepare the code:

Download zip file from LumiNUS and extract to local drive (e.g.: D:\IRS-PRMLS-2019-09-10-IS1PT-GRP-FlowerPower\)

To prepare the data:

1. Download zipped data set: *flrnonflr-dataset.zip* from link: https://drive.google.com/drive/folders/1G13izEO_Oxs-4vOFDaFvAHjrwu-xibDN?usp=sharing
2. Copy and unzip into the same directory of the npz file (e.g.: D:\IRS-PRMLS-2019-09-10-IS1PT-GRP-FlowerPower\npz\).
   Please note the location should align with the path with *np.load(<path_to_npz>)* in code

(Note: more data with different pixel sizes and splits can be found at: https://drive.google.com/drive/folders/1_D9W2_4_8uSFNBHjEh0BIZJK846ujYEE?usp=sharing)

To prepare the Conda Environment:

1. To run *Image_Classification_Neural_Network_Final.ipynb*:
   - Open Anaconda command prompt
   - $ cd <code directoty> (e.g.: D:\PRMLS-MR-2019-09-10-IS1PT-GRP-FlowerPower\)
   - $ conda create --name tf-gpu

```
(base) D:\personal\NUS\Sem2\PRMLS\CA2\PRMLS-MR-2019-09-10-IS1PT-GRP-FlowerPower>conda create --name tf-gpu
WARNING: The conda.compat module is deprecated and will be removed in a future release.
Collecting package metadata: done
Solving environment: done
```

   - $ conda activate tf-gpu
   - $ conda install -c aaronzs tensorflow-gpu

```
(base) D:\personal\NUS\Sem2\PRMLS\CA2\PRMLS-MR-2019-09-10-IS1PT-GRP-FlowerPower>conda activate tf-gpu

(tf-gpu) D:\personal\NUS\Sem2\PRMLS\CA2\PRMLS-MR-2019-09-10-IS1PT-GRP-FlowerPower>conda install -c aaronzs tensorflow-gp
u
```

```
Downloading and Extracting Packages
certifi-2019.9.11    | 155 KB    | ############################################################################ | 100%
absl-py-0.8.0        | 161 KB    | ############################################################################ | 100%
openssl-1.1.1d       | 5.7 MB    | ############################################################################ | 100%
setuptools-41.2.0    | 685 KB    | ############################################################################ | 100%
astor-0.8.0          | 45 KB     | ############################################################################ | 100%
python-3.6.9         | 20.4 MB   | ############################################################################ | 100%
numpy-base-1.14.5    | 3.8 MB    | ############################################################################ | 100%
tensorflow-gpu-1.10. | 91.5 MB   | ############################################################################ | 100%
werkzeug-0.16.0      | 255 KB    | ############################################################################ | 100%
vs2015_runtime-14.16 | 2.4 MB    | ############################################################################ | 100%
libprotobuf-3.9.2    | 2.2 MB    | ############################################################################ | 100%
pip-19.2.3           | 1.9 MB    | ############################################################################ | 100%
tensorboard-1.10.0   | 3.3 MB    | ############################################################################ | 100%
ca-certificates-2019 | 165 KB    | ############################################################################ | 100%
gast-0.3.2           | 13 KB     | ############################################################################ | 100%
protobuf-3.9.2       | 588 KB    | ############################################################################ | 100%
wheel-0.33.6         | 58 KB     | ############################################################################ | 100%
numpy-1.14.5         | 35 KB     | ############################################################################ | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```

   - $ conda install -c anaconda cudatoolkit==9.0

```
Downloading and Extracting Packages
cudatoolkit-9.0       | 339.8 MB  | ##################################################################################### | 100%
protobuf-3.8.0        | 582 KB    | ##################################################################################### | 100%
libprotobuf-3.8.0     | 2.2 MB    | ##################################################################################### | 100%
vs2015_runtime-15.5.  | 2.2 MB    | ##################################################################################### | 100%
vc-14.1               | 5 KB      | ##################################################################################### | 100%
openssl-1.1.1         | 5.7 MB    | ##################################################################################### | 100%
certifi-2019.9.11     | 155 KB    | ##################################################################################### | 100%
ca-certificates-2019  | 165 KB    | ##################################################################################### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```

- $ conda install -c anaconda cudnn==7.1.4

```
Downloading and Extracting Packages
cudnn-7.1.4           | 192.3 MB  | ##################################################################################### | 100%
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
```

- $ jupyter notebook

2. To run specific VGG/AlexNet models, use: Conda ENV given from PSUPR-Day5 workshop: *ml1P13* (with GPU)
   - Open Anaconda command prompt
   - $ cd <code directoty> (e.g.: D:\PRMLS-MR-2019-09-10-IS1PT-GRP-FlowerPower\)
   - $ conda create -n ml1P13 python=3.6 numpy=1.15.1 opencv=3.4.2 matplotlib=2.2.3 tensorflow=1.13.1 tensorflow-gpu=1.13.1 cudatoolkit=9.0 cudnn=7.1.4 scipy=1.1.0 scikit-learn=0.19.1 pillow=5.1.0 spyder=3.3.2 cython=0.29.2 pathlib=1.0.1 ipython=7.2.0 yaml pandas keras keras-gpu pydot graphviz
   - $ conda activate ml1P13
   - $ jupyter notebook

3. To run specific ResNet/Inception/Inception ResNet/SE Inception ResNet models, use: above created local Conda ENV: *tf-gpu*.

4. To run models without GPU is not recommended as it can take long hours to run through the training section. Instead, please use colab at: https://colab.research.google.com, enable GPU and mount/upload respective dataset.

To run the code:

After jupyter notebook was launched, click run all or run each block individually.