Lab 3: Deep Learning

**Submission Instructions:**

1. **You will submit the modified Task1.ipynb, Task2.ipynb, ssd512.ipynb and ResNet50RetinaNet.ipynb files and a txt file lab3.txt.**
2. **You should zip all the above files and upload it in blackboard.**
3. **In lab3.txt provide answers to question asked in each tasks (Task1, Task2 and Task3). Sample lab3\_sample.txt is provided.**
4. **You do not need to submit the data files.**
5. **At the beginning of your lab3.txt, please clearly indicate contribution of individual members, e.g., who drafted which answers, and who reviewed which answers.**

**Before you start:**

Anaconda and Python:

* Introduction: https://www.anaconda.com/
* Download and install Anaconda 5.3 with python 3.7 version.
* Link: <https://www.anaconda.com/download/>
* Open Anaconda Prompt(with Administrative previlage) and create a virtual environment using the following command:
  + *(base) C:\> conda create –n deeplearning python=3.7 anaconda*
* Activate the new virtual environment using following command:
  + *(base) C:\> Activate deeplearning*
* The environment should change from “base” to “deeplearning”
  + *(deeplearning) C:\>\_*

Tensorflow:

* Introduction: <https://www.tensorflow.org/>
* Installation Steps:
  + In “deeplearning” virtual environment install tensorflow using following command:
    - *(deeplearning) C:\> conda install –c anaconda tensorflow*

H5py:

* Introduction: https://www.h5py.org/
* Installation Step:
  + *(deeplearning) C:\> conda install h5py*

Jupyter Notebook:

* Introduction: http://jupyter.org/
* Installation Step:
  + *(deeplearning) C:\> conda install jupyter notebook*

Keras:

* Introduction: <https://keras.io/>
* Installation Step:
  + *(deeplearning) C:\> pip install keras*

Other required installations:

* Numpy
  + Introduction: http://www.numpy.org/
  + Installation Step:
    - *(deeplearning) C:\> conda install numpy*
* Pandas
  + Introduction: <https://pandas.pydata.org/>
  + Installation Step:
    - *(deeplearning) C:\> conda install pandas*
* Opencv
  + Introduction: https://opencv-python-tutroals.readthedocs.io/en/latest/py\_tutorials/py\_tutorials.html
  + Installation Step:
    - *(deeplearning) C:\> conda install –c anaconda opencv*
* Matplotlib
  + Introduction: https://matplotlib.org/
  + Installation Step:
    - *(deeplearning) C:\> conda install –c conda-forge matplotlib*
* Pillow
  + Introduction: https://python-pillow.org/
  + Installation Step:
    - *(deeplearning) C:\> conda install pillow*
* Tqdm
  + Introduction: https://pypi.org/project/tqdm/
  + Installation Step:
    - *(deeplearning) C:\> conda install tqdm*
* Scikit-learn
  + Introduction: http://scikit-learn.org
  + Installation Step:
    - *(deeplearning) C:\> conda install -c anaconda scikit-learn*
* Imageio
  + Introduction: https://pypi.org/project/imageio/
  + Installation Step:
    - *(deeplearning) C:\> conda install -c conda-forge imageio*

**Task 1: Image Classification using Pre-Trained Model**

* We will be using a pre-trained ResNet50 model on imageNet to predict the classes of all 10 test images in folder “Task1”.
* For each test image, provide top 3 predicted classes and their confidence levels.
* Did you find any discrepancies between the predicted class and the actual class? Provide your observations.
* Note: Modify and run the script “Task1.ipynb” in folder *~/Lab3/Task1* to get the results.

**Task 2: Transfer Learning by Re-training Model** We will train the ResNet50 model with the images in *~/Task2/Train* folder.

* The train folder contains 2 sub folders. The sub folder name represents the class labels, including Guardrail\_Barriers and RumbleStrip.
* We have 50 training images for each class.
* Train your custom ResNet50 model with following parameters:
  + Epochs = 150
  + batch\_size = 16
  + Validation split = 0.3
* Classify the test images in *~/Lab/Task2/Test* using retrained ResNet50 model.
  + There are 10 Test Images for each class type (Guardrail\_Barriers and RumbleStrip)
  + How many of the test images were correctly classified?
* Note: Modify and run the script “Task2.ipynb” in folder ~/Lab3/Task2.

**Task 3: Object Detection Using Pre-trained Models**

* We will use two pre-trained Models for object detection.
  + RetinaNet
    - Before you start, Go to the repository ~\Lab3\Task3\*keras-retinanet* in anaconda prompt and execute `***pip install . --user***`.
    - Next run ‘***python setup.py build\_ext --inplace***’
    - Modify and run ResNet50RetinaNet.ipynb in folder (*~\Lab3\Task3\keras-retinanet*) to get the classification results on test images.
  + SSD
    - Modify and run ssd512.ipynb in folder (*~\Lab3\Task3\* *SSD512\_keras*) to get the classification results on test images.
* Test images are located in folder *~\Lab3\Task3\TestImages*
* For test image 1, which model has the better detection result? Were the models able to detect all the people in the image?
* In test image 2, which model was able to detect the laptop? What might be the reason the other model could not detect the laptop?