

# [CS-8395 Spring 2020] Deep Learning in Med. Image Computing

## Assignment 1: Detection

### I. Purpose:

This homework contains a simple computer vision detection task, which is a warm up project for this class. From this assignment, you will:

1. Develop a 2D deep neural network for objective detection.
2. Be able to refer the similar platform based on assignment 0.
3. Have the processing pipeline for the following assignments.

The training data is small-scale, which allows you to train it on GPU or CPU.

### II. Grading and Submission

1. The assignment will be evaluated in a total of 150 scores. The basic scores are generally given based on the following table. Then, the scores will be further adjusted based on requirements in “Tasks” (as red color scores in Tasks).

	Basic Score	
Amazing Work	150	Design a new network with 6 roughly correct predictions on testing images (or withheld images).
	140	Have substantial improvements on an existing network with 6 roughly correct predictions on testing images (or withheld images).
Solid Project	130	Some improvements or directly using an existing network with at least 3 roughly correct predictions on testing images (or withheld images).
	120	Some improvements or directly using an existing network with at least 1 roughly correct prediction on testing images (or withheld images).
Significant Efforts	110	Some improvements or directly using an existing network with no roughly correct predictions on testing images (or withheld images).
Much Work Needed	90	Propose a method with some implementation.
Show Understanding	70	Propose a method without implementation.
Turn in Something	50	Barely written report.
No Turn in	0	

2. The assignment should be submitted in four formats:
  - i) Presentations should be submitted to brightspace as a ppt/pptx file with last name and VUID (e.g., “Huo\_huoy1.pptx”).
  - ii) A single PDF report file should be submitted to brightspace with last name and VUID (e.g., “Huo\_huoy1.pdf”). The PDF report consist presentation slides and code.  
(Please do not write any extra words)

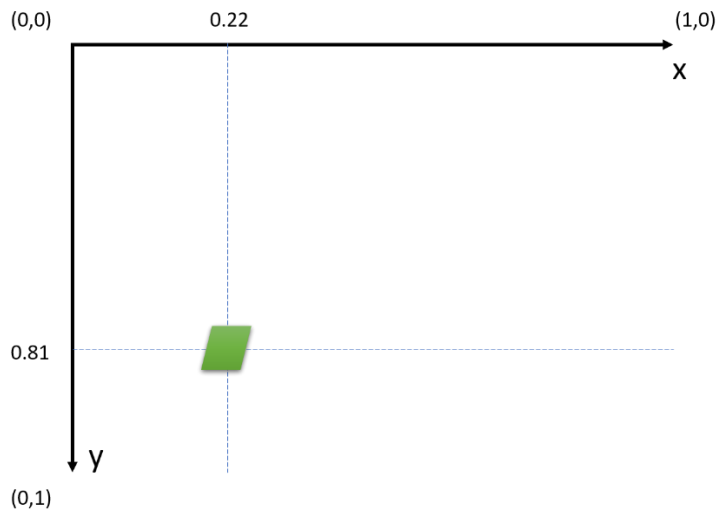
- iii) The same PDF file should also be printed (color/black) and please bring it to class.  
Don't forget to put your name and VUID on the first page of the report.
  - iv) All source code should be submitted to brightspace as a single zip file with last name and VUID (e.g., "Huo\_huoy1.zip").
3. The deadline of brightspace submission is 9:00am on Jan 28.  
The deadline of hardcopy report is 4:00pm on Jan 28.

### III. Description

This assignment is to implement an object detection deep neural network. The task is to find the location of the object from a single RGB image.

- 1. 105 training images are saved in "train" folder.
- 2. 10 validation images are saved in "validation" folder.
- 3. 6 testing images are saved in "test" folder.
- 4. The coordinates of training and validation images are saved in "labels" folder. The coordinates of testing images are not provided.
- 5. A set of withheld images are not released so that instructor is able to test the code on such images when it is necessary.

The coordinate system in labels.txt is like the following figure.



Each row in the labels.txt file means:

img\_path, x (coordinate of the object), y (coordinate of the object)

The coordinates are the relative coordinates that are normalized to 0 to 1. For example, if the image is 128x128. The coordinate for (64,64) is x=0.5, y=0.5.

The labels for train and validation images are provided. You can tune the network base on train and validation images to obtain the good combinations of hyper-parameters (learning rate, batch size etc.). The main training code should be named as "train.py".

Then, the trained model will be applied to the six testing images, which the labels are not provided. You need to write a "test.py" file, which load the trained model and apply to a

single jpg image. For example, we would run test.py on a jpg image and get coordinates with four digits precision.

```
>>> python test.py ~/test/121.jpg  
0.5000 0.5000
```

#### IV. Tasks:

The following tasks can be run on Windows, Mac or Linux, with/without GPU.

##### 1. Presentation

Each presentation is presented in 3 minutes. Less than 3 is totally fine, but please try to keep in within 3 minutes. (10 scores)

i) Title page with name, 1 slide. (5 scores)

ii) Introduction, 1 slide, (5 scores)

Summarize the task in 1 slide.

iii) Rationale, 1-2 slides (10 scores)

The method you referred and why did you use that.

iv) Method, 2 slides (40 scores)

Slide 1:

Show a figure of network structure. If you use existing networks (e.g. ResNet, VGG, AlexNet etc.), you can even copy paste the figure from google/paper. The purpose of the figure is to let the reader understand the method quickly.

Slide 2:

Show how did your format the input and output?

Slide 3:

Show did you use any interesting tricks for training? (e.g., preprocessing, postprocessing, data augmentation etc.)

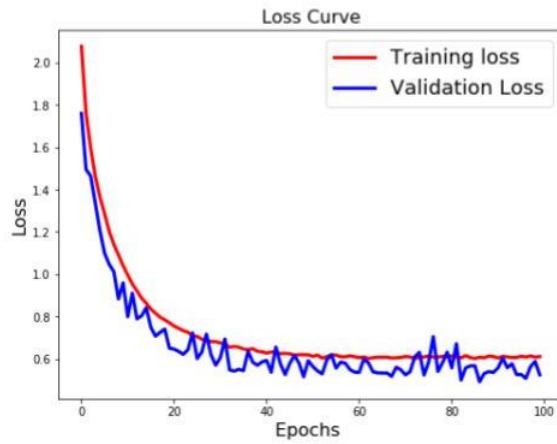
Slide 4:

Describe the hyper parameters used during training. (e.g., epoch number, batch size, learning rate, loss function, parameters of layers, optimizer, input number of channels, output number of channels, OS, GPU/CPU model).

v) Results, 3 slides (30 scores)

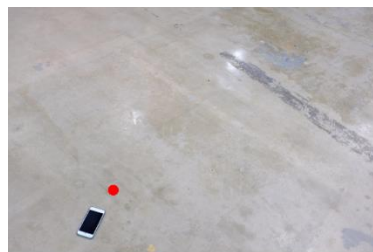
Slide 1:

One figure shows training (105 images) and validation loss (10 images) along with epochs, such as the following example figure.



Slide 2:

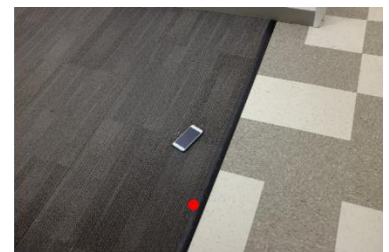
One figure shows the overlays of the six testing figures and your detection, such as the following example figure.



121.jpg



122.jpg



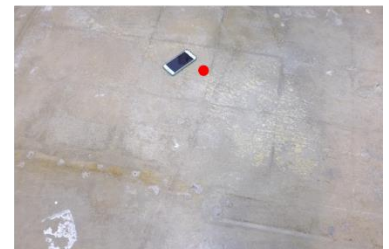
123.jpg



124.jpg



125.jpg



126.jpg

Slide 3:

One table like following example

Name	Coordinate 1	Coordinate 2
121.jpg	0.1111	0.1111
122.jpg	0.1111	0.1111
123.jpg	0.1111	0.1111
124.jpg	0.1111	0.1111
125.jpg	0.1111	0.1111
126.jpg	0.1111	0.1111

For Slide 1, 2 and 3 in Results, they should be included in the report, but optional in the presentation.

The roughly correct prediction means your output for testing images (and may be also for withhold images) are within a radius of 0.05 (normalized distance) centered on the object.

vi) Conclusion, 1-2 slides (10 scores)

Summarize the experiments (e.g., difficulties, limitations, or thoughts).

2. Code

Paste your code at the end of the report and submitted the e-version as a zip file.

The consistency of the method, results and the code would be evaluated by lecturer/TA.

i) Code for training and validation (15 scores)

The labels for train and validation images are provided. You can tune the network base on train and validation images to obtain the good combinations of hyper-parameters (learning rate, batch size etc.). The main training code should be named as "train.py".

ii) Code for testing (15 scores)

The trained model will be applied to the six testing images or withhold images, which the labels are not provided. You need to write a "test.py" file, which load the trained model and apply to a single jpg image. For example, we would run test.py on a jpg image and get coordinates with four digits precision.

```
>>> python test.py ~/test/121.jpg  
0.5000 0.5000
```

3. Submission (10 scores)

Submit report in both e-version and hardcopy.

Submit code in e-version.

Submit presentation in e-version.