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

Assignment 2: Classification

I. Purpose:

This homework contains a large-scale classification task for skin lesion detection:

1. Develop a 2D deep neural network on >10,000 medical image classification.
2. You will use 9015 skin lesion images as training, and 1000 as testing with 7 classes.
3. Understand the basic metrics for classification (acc, precision, recall, confusion matrix).
4. Be able to refer the similar platform based on assignment 0, and assignment 1.
5. Have the processing pipeline for the following assignments.

The training data is large-scale, please use GPU since CPU is probably too slow.

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	**(score can be adjusted based on the "Task" requirements)**
Amazing Work	150	Design a new network with > 80% accuracy on testing images.
	140	Have substantial improvements on an existing network with > 70% accuracy on testing images.
Solid Project	130	Some improvements or directly using an existing network with > 60% accuracy on testing images.
	120	Some improvements or directly using an existing network with at least > 50% accuracy on testing images.
Significant Efforts	110	Some improvements or directly using an existing network with no roughly correct predictions on testing 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").

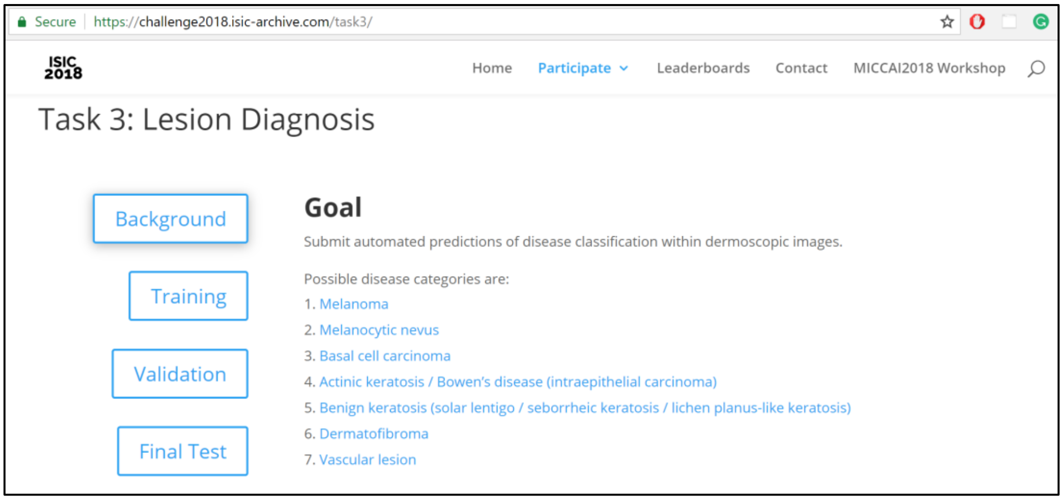
3. The deadline of braightspace submission is 9:00am on Feb 13.
   The deadline of hardcopy report is 4:00pm on Feb 13.

III. Description

This assignment is to implement a classification deep neural network. The link for downloading the data is provided in BrightSpace.

1. 9015 training images are saved in “train” folder.
2. 1000 testing images are saved in “test” folder.
3. The class labels are saved in “labels” folder.
4. A set of withheld images are not released so that instructor is able to test the code on such images when it is necessary.

Everything about the data can be found from ISIC website (https://challenge2018.isic-archive.com/task3/)



Task 3: Lesion Diagnosis

Goal

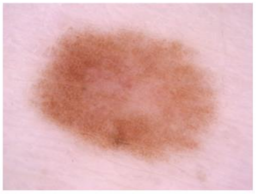
Submit automated predictions of disease classification within dermoscopic images.

Possible disease categories are:

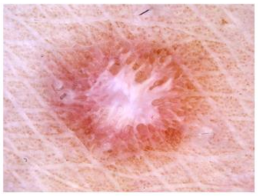
1. Melanoma
2. Melanocytic nevus
3. Basal cell carcinoma
4. Actinic keratosis / Bowen’s disease (intraepithelial carcinoma)
5. Benign keratosis (solar lentigo / seborrheic keratosis / lichen planus-like keratosis)
6. Dermatofibroma
7. Vascular lesion

The lesion types are

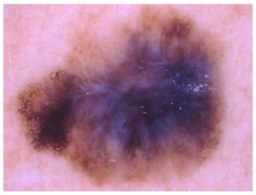
Nevus



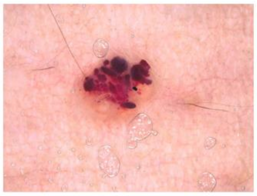
Dermatofibroma



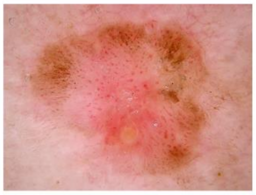
Melanoma



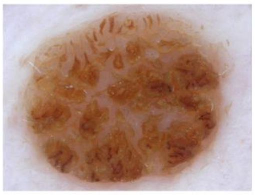
Vascular



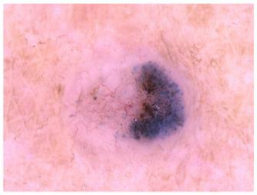
Pigmented Bowen’s



Pigmented Benign Keratoses



Basal Cell Carcinoma



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

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)

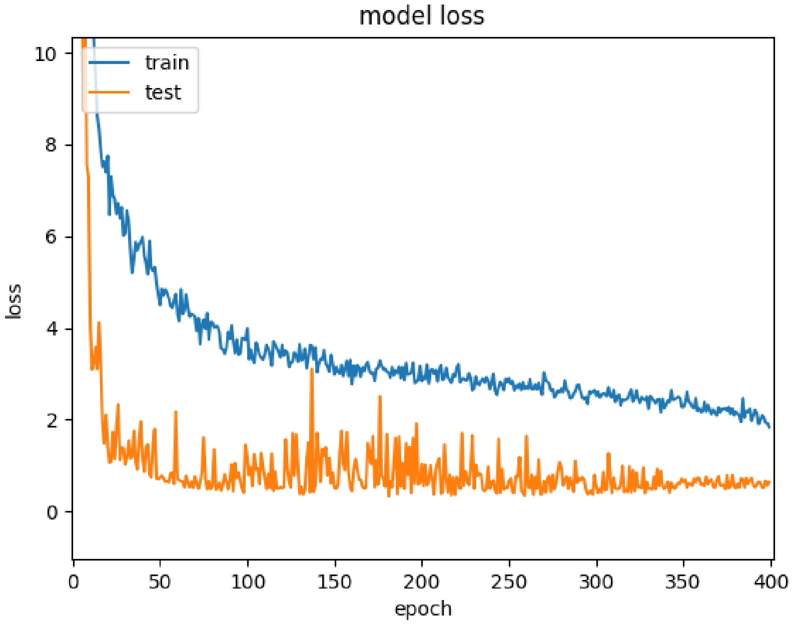
   ii) Introduction, 1 slide, (5 scores)
   Summarize the task in 1 slide.

   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 and testing loss along with epochs, such as the following example figure.



Slide 2:

One slide shows accuracy, recall, precision.

To get the scores, you can refer to the following websites.

The python code and definition of accuracy:

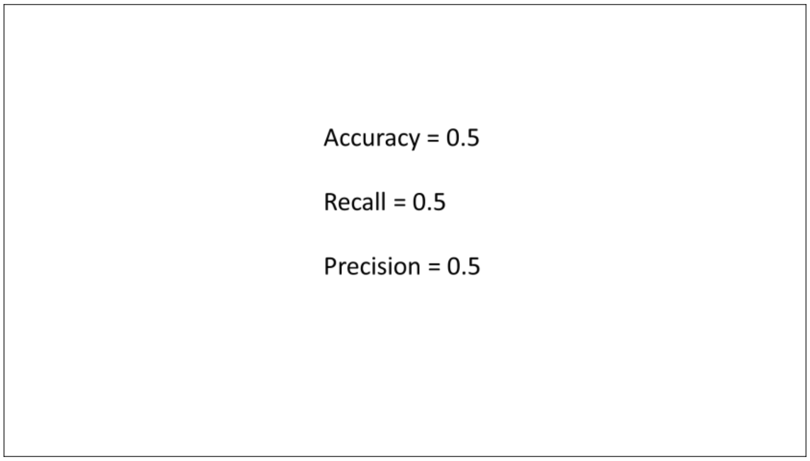
http://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html#sklearn.metrics.accuracy_score

The python code and definition of recall:

http://scikit-learn.org/stable/modules/generated/sklearn.metrics.recall_score.html#sklearn.metrics.recall_score

The python code and definition of recall:

http://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_score.html#sklearn.metrics.precision_score



Accuracy = 0.5

Recall = 0.5

Precision = 0.5

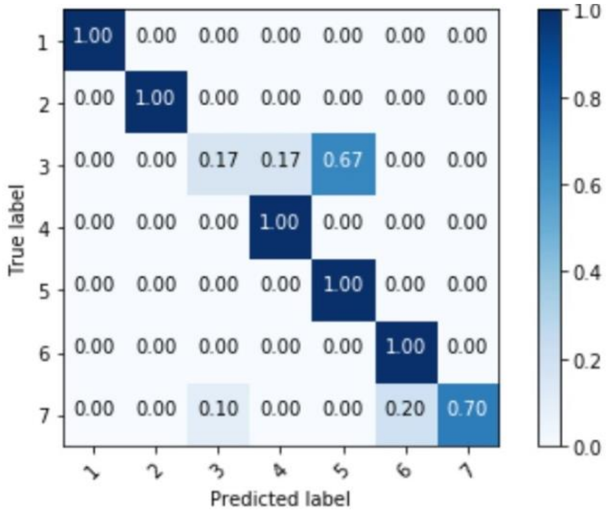
Slide 3:

One figure shows confusion matrix.

To get the matrix, you can refer to the following websites.

The python code and definition of confusion matrix:

http://scikit-learn.org/stable/modules/generated/sklearn.metrics.confusion_matrix.html



	1	2	3	4	5	6	7
1	1.00	0.00	0.00	0.00	0.00	0.00	0.00
2	0.00	1.00	0.00	0.00	0.00	0.00	0.00
3	0.00	0.00	0.17	0.17	0.67	0.00	0.00
4	0.00	0.00	0.00	1.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	1.00	0.00	0.00
6	0.00	0.00	0.00	0.00	0.00	1.00	0.00
7	0.00	0.00	0.10	0.00	0.00	0.20	0.70

You can decide if you will present your results slides or not during in class presentation.

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.

i) Code for training (15 scores)
The labels for > 9000 training images are provided. You can tune the network base on train 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 1000 testing images or withhold images, which the labels are 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/ ISIC_0033321.jpg
>>> 3
```

Please make sure labels of testing images are **ONLY** used for calculating the metrics, and **NOT** used during training or testing. If the lecturer feels the results are not "real", you might be asked to re-produce your training and testing on ACCRE in lecturer's office.

3. Submission (10 scores)

Submit report in both e-version and hardcopy.
Submit code in e-version.
Submit presentation in e-version.