CS 534 Group Project Assignment 4 (100 Points)

Due: 11:59 p.m. on 04/23/2023

Project Objective: The goals of this assignment are to help you understand the concepts and the algorithms that we will discuss/have discussed in Week 12 ~ Week 14.

Note:

(1) *This project is to be done by* ***EACH GROUP****. No help besides the textbook, materials, and the instructor/TA should be taken. Copying any answers or part of answers from other sources, including your classmate groups, will earn you a grade of zero.*

(2) *Your program must be developed and implemented in the PyCharm-like IDE, or 10% of the graded score is deducted. Please check and choose the one from here: https://realpython.com/python-ides-code-editors-guide/, as the suggestion. Note that we* ***DO NOT*** *use the Jupyter as the IDE.*

(3) *Assignments are accepted in their assigned Canvas drop box without penalty if they are received by 11:59PM EST on the due date, or 10% of the graded score is deducted for the late submission per day. Work submitted after one week of its original due date will not be accepted.*

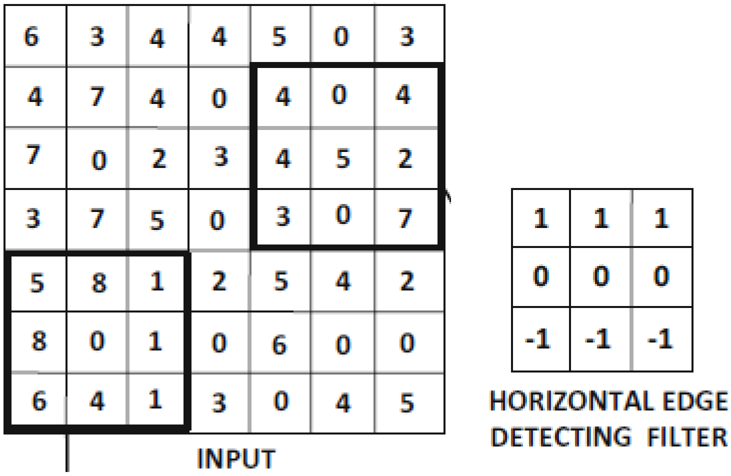
Project Deliverables: Submit your answers and solutions in a ***zip*** file that includes the write-ups of Question 1 in a ***pdf*** file and your ***python codes*** of Question 2 to Canvas.

A. Project Questions:

Question 1 (30 Points)

Part 1. Consider a 1-dimensional time-series with values [2, 1, 3, 4, 7]. Perform a convolution with a 1-dimensional filter [1, 0, 1] without padding. Show all the computational steps to get the full credit.

Part 2. Compute the convolution of the below input with the horizontal edge detection filter. Use a stride of 1 without padding. Show all the computational steps to get the full credit.



6	3	4	4	5	0	3
4	7	4	0	4	0	4
7	0	2	3	4	5	2
3	7	5	0	3	0	7
5	8	1	2	5	4	2
8	0	1	0	6	0	0
6	4	1	3	0	4	5

INPUT

1	1	1
0	0	0
-1	-1	-1

HORIZONTAL EDGE DETECTING FILTER

Part 3. Perform a 4 × 4 max pooling at stride 1 of the above same input. Show all the computational steps to get the full credit.

B. Project Development

Question 2 (70 Points)

In this project, you will develop and implement a pipeline that will train and test the AlexNet (See Below) to perform a binary classification on melanoma and naevus images.

AlexNet

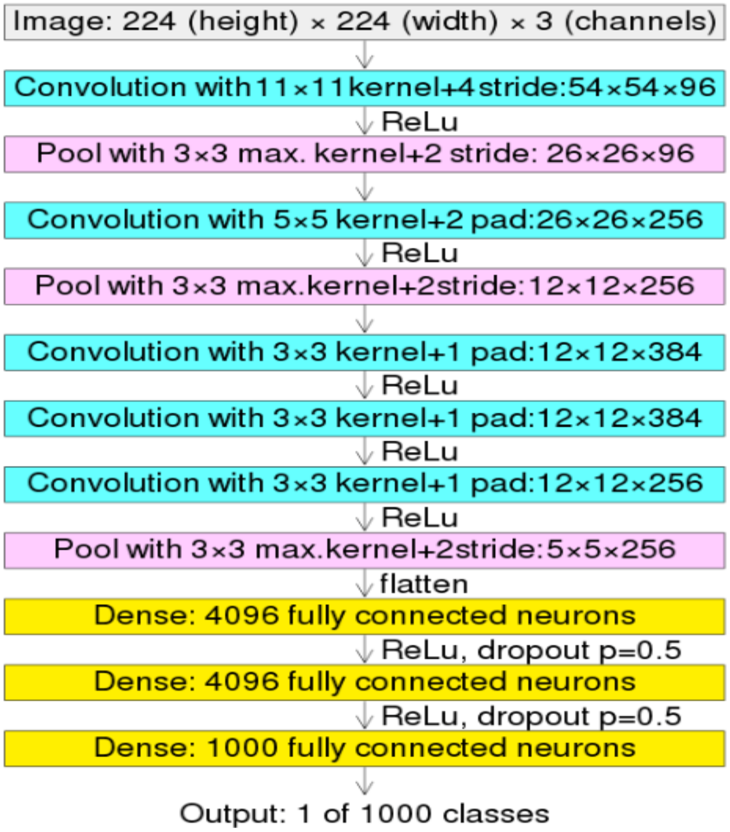


Image: 224 (height) × 224 (width) × 3 (channels)
Convolution with 11×11 kernel+4 stride:54×54×96
ReLu
Pool with 3×3 max. kernel+2 stride: 26×26×96
Convolution with 5×5 kernel+2 pad:26×26×256
ReLu
Pool with 3×3 max.kernel+2stride:12×12×256
Convolution with 3×3 kernel+1 pad:12×12×384
ReLu
Convolution with 3×3 kernel+1 pad:12×12×384
ReLu
Convolution with 3×3 kernel+1 pad:12×12×256
ReLu
Pool with 3×3 max.kernel+2stride:5×5×256
flatten
Dense: 4096 fully connected neurons
ReLu, dropout p=0.5
Dense: 4096 fully connected neurons
ReLu, dropout p=0.5
Dense: 1000 fully connected neurons
Output: 1 of 1000 classes

(1). Study and investigate the raw dataset provided by https://www.cs.rug.nl/~imaging/databases/melanoma_naevi/.

(2). If you are interested, please feel free to read the "AlexNet" paper (https://papers.nips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf) as a reference.

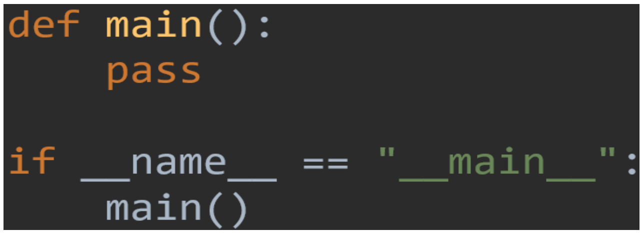
(3). As the raw dataset consists of 70 melanoma and 100 naevus images, for the learning and practice purposes, you can randomly select 70 naevus images to make your dataset balanced, i.e., 70 melanoma images and 70 naevus images. You can use any open-source AlexNet code if you prefer, but I would recommend the Pytorch version. Here are the links as a reference:

- https://pytorch.org/hub/pytorch_vision_alexnet/
- https://pytorch.org/vision/main/models/generated/torchvision.models.alexnet.html
- https://github.com/pytorch/vision/blob/main/torchvision/models/alexnet.py

(4). After (3), divide the above dataset into training (50 melanoma and 50 naevus images) and testing (20 melanoma and 20 naevus) sets, respectively. The total number of images is 140. You will use the training portion, 100 images, to train the AlexNet with **num_classes: int = 2**. Due to the small training samples, please try different dropout rate between 0 and 1 by using 5-fold cross validation to see which dropout rate will deliver the highest image classification accuracy. Please print the best dropout rate and its image classification accuracy on the output.

(5). After (4), use the testing dataset, 40 images, to evaluate and show the image classification accuracy of the best dropout-rate AlexNet that you just obtained from above.

(6). Submit **your original 140 raw images**, i.e., training (50 melanoma and 50 naevus images) and testing (20 melanoma and 20 naevus) sets, and your **pipeline** (**AlexNetPredictorPipeline.py**), including (3), (4), and (5), using the given construct to Canvas. That is, when your TA executes your pipeline with your provided raw data, all the above outputs will be displayed on your TA's computer console output.



```
def main():
    pass

if __name__ == "__main__":
    main()
```

Grading Criteria: Your answers must be complete and clear.

Checkpoints	**Points Possible**
Project Question	30 Points
Project Development	70 Points
(1). Proper Naming Conventions and Program Documentation on Your Codes	5 Points
(2). Compliable Code: **AlexNetPredictorPipeline.py**	10 Points
(3). Make Your Dataset Balanced	5 Points
(4). Divide the above dataset into training (50 melanoma and 50 naevus images) and testing (20 melanoma and 20 naevus) sets, respectively. Try different dropout rate between 0 and 1 by using 5-fold cross validation to see which dropout rate will deliver the highest image classification accuracy. Print the best dropout rate and its image classification accuracy on the output.	30 Points
(5). Use the testing dataset, 40 images, to evaluate and show the image classification accuracy of the best dropout-rate AlexNet that you just obtained from above	20 Points
Total	100 Points