BMEG4320 Assignment 2

Classification of Retinal Diseases based on Deep Learning and OCT

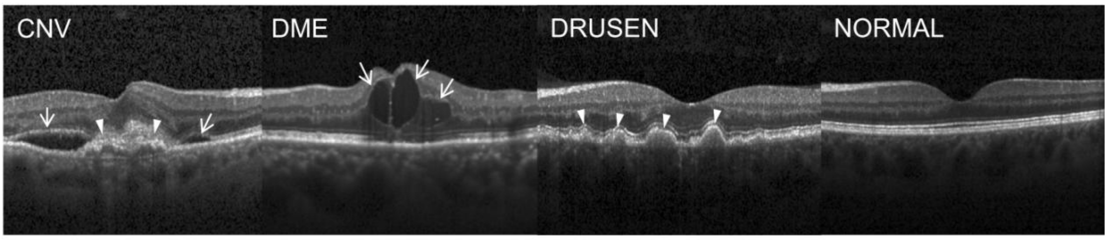
Date: 4/10/2023 (Wed) – 29/10/2023 (Sun)

1. Overview & Background

Artificial intelligence (AI), particularly deep learning (DL) techniques, can potentially revolutionize disease diagnosis and management. AI can perform complex classification tasks for human experts and rapidly review immense amounts of images.

Spectral-domain OCT uses light to capture high-resolution in vivo optical cross-sections of the retina that can be assembled into three-dimensional-volume images of living retinal tissue. It has become one of the most commonly performed medical imaging procedures and a standard of care for guiding the diagnosis and treatment of some of the leading causes of blindness worldwide.

In this assignment, you need to build a diagnostic model based on a DL framework for the 2D four-class classification of patients with common treatable blinding retinal diseases in OCT images, as shown in Fig 1. The dataset has been integrated into MedMNISTv2 (https://medmnist.com/) and divided into three parts: training set, validation set, and test set. More detailed information about this research can be found in the original paper [1].



Dataset	CNV	DME	DRUSEN	NORMAL	Total
Training set	33484	10213	7754	46026	97477
Validation set	3721	1135	862	5114	10832
Test set	250	250	250	250	1000

Fig 1. (Far left) choroidal neovascularization (CNV) with neovascular membrane (white arrowheads) and associated subretinal fluid (arrows). (Middle left) Diabetic macular edema (DME) with retinal-thickening-associated intraretinal fluid (arrows). (Middle right) Multiple drusen (arrowheads) present in early AMD. (Far right) Normal retina with preserved foveal contour and absence of any retinal fluid/edema [1].

The following documents would be given in this assignment:

1. '*DLframework.zip*'. We provide you with a runnable DL framework (skeleton program) for the classification of retinal diseases. Some codes are blocked, and you must write down your codes to finish the corresponding tasks. Note that other codes unrelated to the tasks cannot be modified. The dataset will be downloaded automatically via the functions in this framework. You can use the anaconda Spyder IDE to open and compile the codes in Python.

2.1 Task 1: Implementation of ResNet

Deep Convolutional Neural Network (CNN) plays an important role in the DL model performance. A Residual Neural Network (ResNet) is one of the typical network architectures. It is a network with skip connections that perform identity mappings, merged with the layer outputs by addition. This enables DL models with tens or hundreds of layers to train easily and approach better accuracy when going deeper. More detailed information could be found in the original paper [2].

In this task, you are required to implement the ResNet18 in our DL framework. Specifically, you need to fill in your codes in *Function forward(), Class ResNet (Line 67) in DLframework/models.py.* This task can help you understand the functions of different layers and the principles of ResNet.

2.2 Task 2: Implementation of Loss Function

The loss function is a method of evaluating how well your model is in terms of predicting the expected outcome. The loss function is used in the training process that uses back-propagation to minimize the error between the actual and predicted outcomes. The loss function needs to be minimized to improve its performance.

In this task, you are required to implement the *Cross-Entropy Loss function* in our framework (*DLframework/train_and_eval_prtorch.py, Line 266*) from scratch for multi-class classification, which means you **cannot** use the built-in functions *nn.CrossEntropyLoss()* or *F.CrossEntropy()* in Pytorch directly. This task can help you understand the computation process of loss functions.

2.3 Task 3: Implementation of Evaluation Metrics

Evaluation metrics are tied to DL tasks. There are different metrics for the functions of classification. Using various metrics for performance evaluation, we should be able to improve our model's overall predictive power before we roll it out for production on unseen data. Without doing a proper assessment of the DL model by using different evaluation metrics and only depending on accuracy, it can lead to a problem when the respective model is deployed on unseen data and may end in poor predictions.

In this task, you are required to implement the evaluation metrics to assess comprehensively your model's classification performances. Specifically, you need to understand and compute these metrics: Sensitivity (SEN), Specificity (SPE), Precision (PREC) and Recall (REC) in our framework (*DLframework/train_and_eval_prtorch.py, Line 247*). This task can help you understand the meaning of different evaluation metrics in medical image analysis.

2.4 Task 4: Open Challenge

In this task, you are required to conduct more techniques on the classification of retinal diseases. The higher the evaluation metrics (e.g., AUC, ACC, SEN, or SPE), the better. For example, you can try to use/plug any tricks in the DL framework to improve the model performances, such as data augmentations, other powerful network architectures/modules, and stage-of-the-art loss functions. On the other hand, you can adjust some hyper-parameters to determine whether they are the best, such as learning-rate, batch-size, and training epoch. This is an open question, and you can change the DL framework in any way you want. Please write down what you did in the technical report. **The final score will be based on your final classification performances and your efforts in this task!**

3 Technical Report

In this assignment, you are required to submit a technical report containing the implementation details, experimental results, and analysis of all tasks. The report should have clear formats (e.g., tables) to present the statistical results/comparison, and the maximum length is four pages. The report should include a brief abstract, introduction, methodology, experiment, result, discussion, and conclusion, just like the simplified version of the conference paper.

4 General Specification

You are required to complete all functionalities according to the specification with Python. Do not change the files' name or the functions' name. Please keep the log recording. Your source codes will be perused, and the object code will be tested! **NO PLAGIARISM**.

5 Marking scheme

a. Task 1 -- 10%
b. Task 2 -- 10%
c. Task 3 -- 10%
d. Task 4 -- 30%
e. Technical report -- 40%

6 Submission Guidelines

The folder you hand in must contain the following:

1. ‘*Technical Report.pdf*’.
2. ‘C*odes/*’ the folder includes the source codes of the modified DL framework, and the log recording (all *@model.csv* files).

Please rename the submitted folder as <your student ID>-Asgn2, and compress it into <your student ID>-Asgn2.zip, and upload it to blackboard system.

Reference

[1] Kermany D S, Goldbaum M, Cai W, et al. Identifying medical diagnoses and treatable diseases by image-based deep learning[J]. cell, 2018, 172(5): 1122-1131. e9.

[2] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.