COM2028 Artificial Intelligence
Semester 2, 2024-25

Coursework Guideline

Prof Gustavo Carneiro, Dr Frank Guerin, Dr Samaneh Kouchaki

3rd March 2025

1 Overview

This coursework concerns the automated classification of images through machine learning techniques. You will work on ultrasound data, where training and testing samples and their ground truth are provided. You will develop suitable classification techniques, a cross-validation experiment, hyper-parameter tuning strategy, an evaluation using specific metrics (accuracy, area under the receiver operating characteristic curve, area under the precision recall curve, precision, recall, and F1 score), visualise model performance through relevant plots (e.g., confusion matrix, receiver operating characteristic curve). You will need to submit a Jupyter notebook containing your code for training, visualisation and evaluation by **4pm on Friday 11th April**. Your submitted Jupyter notebook will be automatically marked based on pre-defined tests, so it is important that you follow the directions provided in this guideline.

2 Dataset and Code

You will use the dataset BreastMNIST from [1]. A Jupyter notebook with code on how to download the dataset, train and test a convolutional model is available from CW_AI_2425.ipynb.

The dataset BreastMNIST contains ultrasound images of size $28 \times 28 \times 1$ pixels. Each image can have two classes: {'0': 'malignant', '1': 'normal, benign'}. The dataset has been split into 546 training images, 78 validation images, and 156 testing images. More information about the original dataset can be found in [1].

The current baseline area under the receiver operating characteristic curve (AUC) and accuracy results [1] are reported in Table 1.

Table 1: Benchmark AUC and Accuracy results on BreasMNIST [1].

Method	AUC	Accuracy
ResNet-18 [1]	0.901	0.863
ResNet-50 [1]	0.857	0.812

3 Deliverables and Rubric

You should submit a Jupyter notebook, with your answers directly coded in the space provided in CW_AI_2425.ipynb. Here is a summary of the coursework.

Question 1 - Model Training and Tuning (40 Marks)

Question 1.1 (10 Marks)

Improve the validation AUC result by modifying hyperparameters (`num_epochs`, `learning_rate`, and `weight_decay_rate`) to achieve `val_AUC > 0.8`.

Question 1.2 (10 Marks)

Enhance model performance using data augmentation. Add the following augmentations to the training images:

- Random rotations of 10°.
- Random horizontal flips with a 50% probability.

Question 1.3 (10 Marks)

Compare the validation AUC results of two optimisers (Adam and SGD). Automatically select the best-performing optimiser based on the validation AUC results and run a final training with it.

Question 1.4 (10 Marks)

Build a new CNN model called `Net2` with the following architecture:

- **Layer 1:** Input convolutional layer with 32 filters (5x5), followed by ReLU activation and max pooling (kernel size = 3, stride = 3).
- **Layer fc:** Linear layer with 300 nodes, ReLU activation, and another linear layer with `num_classes` nodes.

Question 2 - Classification Metrics (10 Marks)

Objective: Write a function `compute_classification_metrics(all_outputs, all_targets)` to calculate:

- AUPR (Area Under the Precision-Recall Curve).
- Precision, Recall, and F1 Score for classifications with probability > 0.5.
- Use `precision_score`, `recall_score`, `f1_score`, `precision_recall_curve`, and `auc` from `sklearn.metrics`.

Question 3 - ROC Curve (10 Marks)

Objective: Plot the ROC curve for binary classification. You are given a function to plot the ROC curve but it lacks the necessary definition of which class's probability to select. In particular, we are interested in the classification of normal/benign cases. Fix the code by selecting the correct probability for the normal/benign cases from the `all_outputs` array.

Question 4 - Confusion Matrix (10 Marks)

Objective: Write a function `compute_confusion_matrix(all_outputs, all_targets)` that calculates the confusion matrix for the binary classification problem using the predicted outputs (`all_outputs`) and ground truth labels (`all_targets`).

Question 5 - Cross Validation (10 Marks)

Question 5.1 (10 Marks)

Fix the incorrect implementation of the cross-validation function `cross_validate_model_with_serious_mistake(combined_dataset, kf, learning_rate, weight_decay_rate, num_epochs)`. Correct it to a functional version: `cross_validate_model(combined_dataset, kf, learning_rate, weight_decay_rate, num_epochs)`.

Question 6 - Fine-Tuning Pre-Trained Model (20 Marks)

Question 6.1 - Preprocessing for ResNet18 (10 Marks)

Objective: Modify the `train_transform` and `val_test_transform` to:

- Resize the images to 224x224 pixels (use `transforms.Resize(224)`).
- Convert grayscale images to 3 RGB channels (use `transforms.Lambda(lambda x: x.repeat(3, 1, 1))`).
- Normalize the images using the ImageNet normalization statistics (use `transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])`).

Question 6.2 - Modify ResNet18 Architecture (10 Marks)

Objective: Load the pre-trained ResNet18 model from torchvision and modify the final fully connected layer to have `n_classes` output nodes.

- Ensure the model is initialized with weights from ImageNet (`weights=torchvision.models.ResNet18_Weights.IMAGENET1K_V1`).

Important Notes

1. **Follow the instructions carefully:** For each question, you will need to modify only the sections of the code that are specified. In particular, you'll see:
   # YOUR CODE HERE
   raise NotImplementedError()
   Please replace **raise NotImplementedError()** with your code. Do not change anything else in the notebook unless directed to do so.

2. **Testing and validation:** Each change you make will be automatically tested using a set of unit tests that are hidden from this notebook.

3. **Keep the code clean:** For each question, only modify the code within the cells indicated by the question. Do not add extra cells or change the overall structure of the notebook unless explicitly instructed.

4. **Error handling:** If you encounter any issues or errors, ensure you carefully debug your code. If you are unable to resolve the issue, ask for help, but remember that the solution must follow the given instructions.

5. **Final Submission:** Once you have completed all the questions, submit the notebook as instructed. Please do not change the notebook's file name. Make sure to run all the cells to ensure that your changes work correctly.

References

[1] Jiancheng Yang, Rui Shi, Donglai Wei, Zequan Liu, Lin Zhao, Bilian Ke, Hanspeter Pfister, and Bingbing Ni. Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. *Scientific Data*, 10(1):41, 2023.