CAP 5516 - Medical Image Computing (Spring 2026)

Programming Assignment #1 (30 points)

Due: 3/1/2026, 11:59 PM (EST)

Deep Learning-based Pneumonia Detection Using Chest X-Ray Images

According to the World Health Organization (WHO), pneumonia kills about 2 million children under 5 years old every year and is consistently estimated as the single leading cause of childhood mortality [1]. In this assignment, you will apply deep learning techniques to detect pneumonia based on pediatric chest X-rays.

1. Dataset

Dataset to be used for this assignment: Chest X-Ray Images (Pneumonia)
Where to download the dataset: https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia
Brief description of the dataset: The dataset is organized into 3 folders (train, test, val) and contains subfolders for each image category (Pneumonia/Normal), see Fig. 1. There are 5,863 X-Ray images (JPEG) and 2 categories (Pneumonia/Normal). Chest X-ray images (anterior-posterior) were selected from retrospective cohorts of pediatric patients of one to five years old from Guangzhou Women and Children's Medical Center, Guangzhou. All chest X-ray imaging was performed as part of patients' routine clinical care.

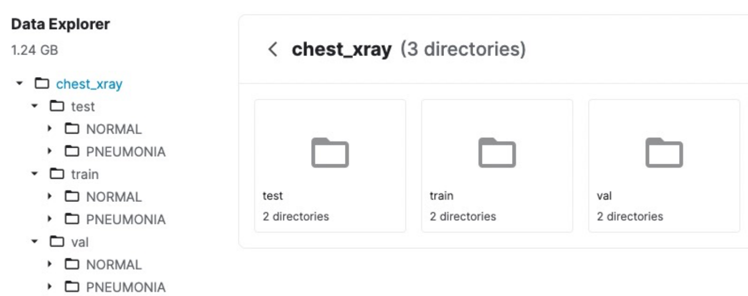


Fig. 1 Overview of the dataset structure.

2. Task

The task is to apply an off-the-shelf CNN or Transformer architecture for Pneumonia classification using X-ray images (i.e., binary classification - Pneumonia/Normal). You can choose any network architecture you want. Since you may have limited computing resources, please consider using small networks such as ResNet-18, MobileNet, MobileNet-v2, ViT-Tiny, etc.

The train/val/test splits are provided within the dataset. The "train" set is for model training, the "val" set is for validating the model generalization ability (e.g., preventing model overfitting) and hyperparameters tuning (e.g., learning rate), and "test" set is for model testing.

Assume ResNet-18 is used as the network backbone, please complete the following two sub-tasks.

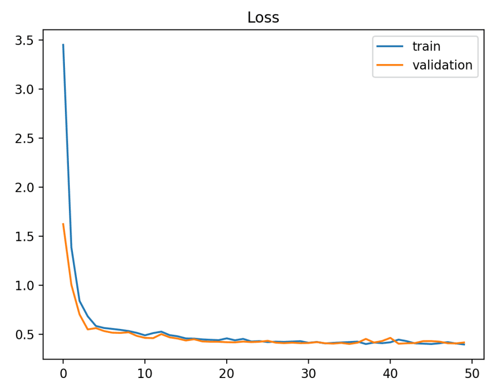
Task 1.1 – Train the model (e.g., ResNet-18) from scratch (i.e., random initialization of the model parameters) using the training X-ray images.

Task 1.2 – Leverage the pre-trained ResNet-18 (the same CNN used in Task 1.1) model on the ImageNet and fine tune the model on the target X-ray images (training set).

By using the techniques learned from class (e.g., data augmentation, model regularization, etc.), the goal is to achieve the best testing accuracy you can for Task 1.1 and Task 1.2.
More training tricks or good practices can be found in [2].

3. What to report

For both sub-tasks, you should report:
1) The implementation details (e.g., network architecture, learning rate, batch size, training epoch, etc.).
2) A graph showing the training and validation loss curves. Fig. 2 shows an example. Tensorboard provides an easy way to track and visualize metrics such as loss and accuracy, see [4-7] for more information.
3) The overall classification accuracy on the testing set. The accuracy for each class (i.e., Pneumonia/Normal).
4) Show several failure cases (i.e., misclassified images). **(Optional):** Use CAM [8] or Grad-CAM [9] to provide "visual explanations" for several predictions made by the model. This is very helpful for analyzing failure cases. Software to use - **TorchCAM: class activation explorer:** https://github.com/frgfm/torch-cam



Epoch
Fig. 2. An example of loss curves [3] for model diagnosis.

4. What to submit

(1) A report for this assignment. The quality of the report is important.
(2) Clean code and clear instructions (e.g., a readme file) to reproduce your results. If you choose to host the code on GitHub, please provide the GitHub link.

References

[1] Rudan, Igor, et al. "Epidemiology and etiology of childhood pneumonia." Bulletin of the world health organization 86 (2008): 408-416B.
[2] He, Tong, et al. "Bag of tricks for image classification with convolutional neural networks." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.
[Code]
https://github.com/weiaicunzai/Bag_of_Tricks_for_Image_Classification_with_Convolutional_Neural_Networks
[3] https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance/
[4] https://pytorch.org/tutorials/recipes/recipes/tensorboard_with_pytorch.html
[5] https://pytorch.org/docs/stable/tensorboard.html
[6] https://pytorch.org/tutorials/intermediate/tensorboard_tutorial.html
[7] https://www.tensorflow.org/tensorboard/scalars_and_keras
[8] Zhou, Bolei, et al. "Learning deep features for discriminative localization." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. [Code] https://github.com/zhoubolei/CAM

[9] Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." Proceedings of the IEEE international conference on computer vision. 2017. [Code] https://github.com/ramprs/grad-cam/