Convolutional Neural Network Training and Optimization for Histopathology Image Classification

Dataset Description

For this project I used the PathMNIST dataset from the MedMNIST2D collection. PathMNIST is a colorectal cancer histology dataset in which each original sample is a 3×224×224 RGB tissue patch. In MedMNIST, the 2D datasets are uniformly downsampled to 3×28×28 to create a lightweight benchmark suitable for rapid experimentation. The dataset has 89,996 training images, 10,004 validation images, and 7,180 test images, with a total of 107,180 labeled histology samples across nine tissue categories.

Methods and Techniques

For the classification task, I chose ResNet-18, a convolutional neural network known for its reliability and efficiency. To fully take advantage of ResNet-18 and ensure effective feature extraction, the input images were resized to 64×64. This preprocessing step preserved more spatial and texture information, which helps improve performance on medical image classification tasks.

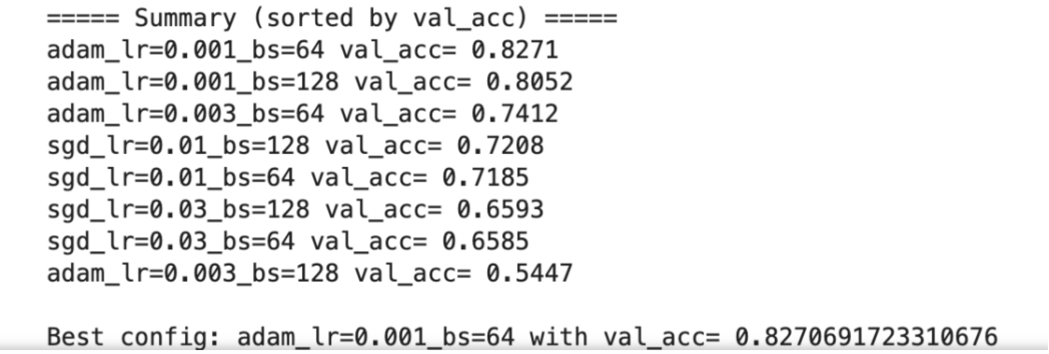
I started by training a baseline model using a baseline configuration to ensure that the full pipeline worked correctly.

Baseline configuration:
- Learning rate: 1e-3
- Batch size: 128
- Epochs: 10
- Optimizer: Adam
- Weight decay: 1e-4
- Scheduler: StepLR (step_size = 5, gamma = 0.1)
- Early stopping patience: 3

Using this setup, I trained a ResNet-18 created with build_resnet18 (n_channels, n_classes, pretrained=False). The loss and accuracy were computed for both training and validation sets and saved in a history dictionary. After training, the function returned best_model and history. This baseline showed a best validation accuracy of 0.9725.

Hyperparameter Tuning

A hyperparameter search was performed on a 20% of the training set due to the long run time required for full-dataset tuning. The search explored two optimizers (Adam and SGD), a total of four learning rates (1×10^{-3} and 3×10^{-3} for Adam, and 1×10^{-2} and 3×10^{-2} for SGD), and two batch sizes (64 and 128). Each experiment was trained for three epochs to provide a fast but informative comparison. Across all combinations, Adam with a learning rate of 1×10^{-3} and batch size of 64 produced the highest validation accuracy.



```
===== Summary (sorted by val_acc) =====
adam_lr=0.001_bs=64 val_acc= 0.8271
adam_lr=0.001_bs=128 val_acc= 0.8052
adam_lr=0.003_bs=64 val_acc= 0.7412
sgd_lr=0.01_bs=128 val_acc= 0.7208
sgd_lr=0.01_bs=64 val_acc= 0.7185
sgd_lr=0.03_bs=128 val_acc= 0.6593
sgd_lr=0.03_bs=64 val_acc= 0.6585
adam_lr=0.003_bs=128 val_acc= 0.5447

Best config: adam_lr=0.001_bs=64 with val_acc= 0.8270691723310676
```

The loss and accuracy curves for the best hyperparameter setting (Adam, learning rate = 0.001, batch size = 64) show a consistent learning (Figure 1 and Figure 2) with training and validation loss show a decrease over three epochs, whereas accuracy shows an increase. By the final epoch, validation accuracy slightly exceeds training accuracy, suggesting that the model generalizes well to unseen data.

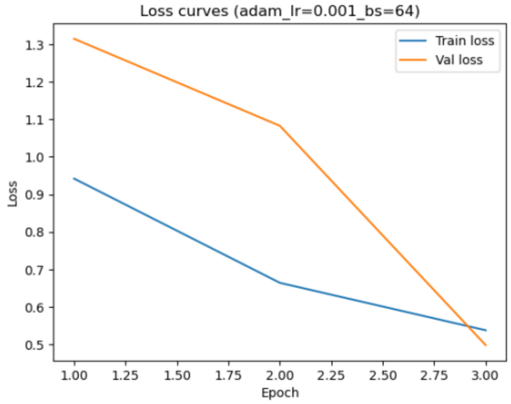


Figure 1: Training and validation loss curves (optimizer = Adam, lr = 0.001, batch size = 64, epochs = 3).

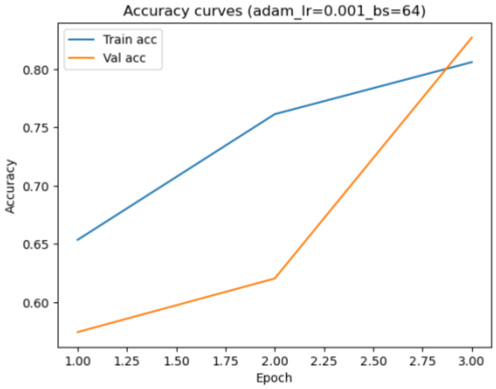


Figure 2: Training and validation accuracy curves (optimizer = Adam, lr = 0.001, batch size = 64, epochs = 3).

The best configuration resulted in a validation accuracy of about 0.827 after the 3-epoch hyperparameter search. After applying these hyperparameters on the full PathMNIST training set, the tuned model slightly overperformed the baseline model by achieving validation accuracy of 0.9761 (compare to 0.9725). The final evaluation on the test set showed a test accuracy of about 0.834 and an AUC of about 0.972, thus, demonstrating strong discriminative performance across tissue classes.

Reflection and Future Work

The optimized baseline model showed good accuracy and AUC scores when tested on the PathMNIST dataset. Further improvements to the pipeline could be achieved by testing data augmentation strategies that are typically used for medical images, testing deeper networks (e.g. ResNet-50 and DenseNet-121). Additionally, exploring methods that deal with class imbalance can further improve accuracy, especially for underrepresented classes.