CSPB-3202 Artificial Intelligence HW5 Kaggle
Maci Flanagan

Problem Description and Dataset

In this project, I participated in the Kaggle Histopathologic Cancer Detection competition, which focuses on detecting metastatic cancer in histopathologic images of lymph node tissue. The motivation behind this task is to develop automated tools that can assist medical professionals in diagnosing cancer more accurately and efficiently. The dataset provided by Kaggle consists of: Over 200,000 color .tif training images, each measuring 96x96 pixels; A CSV file (train_labels.csv) that provides a binary label for each image (1 = cancer, 0 = no cancer); A set of unlabeled test images used to evaluate model performance on Kaggle's platform.The primary evaluation metric for this competition is the Area Under the ROC Curve (AUC), which is appropriate for imbalanced binary classification problems like this one.

Exploratory Data Analysis

I began the project by loading the train_labels.csv file to inspect the distribution of labels. I found that the dataset is imbalanced, with a higher number of non-cancerous (label 0) images compared to cancerous (label 1) images. This observation was important because it informed my choice of evaluation metric (AUC instead of accuracy) and motivated me to consider techniques to handle imbalance, such as resampling or class weighting. To better understand the dataset, I visualized a sample of both cancerous and non-cancerous images. This helped me confirm that the images were small, color tissue samples, and appeared visually similar to medical microscope slides. This exploration also highlighted the need for careful image processing and model tuning, as the differences between classes were not visually obvious.

Modeling Approach

To prepare the data for modeling, I performed the following preprocessing steps: I resized images from 96x96 to 64x64 pixels to reduce computational load while retaining visual features;I scaled pixel values to the range [0, 1] by dividing by 255; I implemented a custom PyTorch Dataset class to efficiently load and process images in batches.

I built a Convolutional Neural Network (CNN) using PyTorch. The architecture consisted of two convolutional layers with ReLU activations and max pooling to reduce spatial dimensions, and a fully connected layer followed by a sigmoid output layer for binary classification.

Training Details

- Loss Function: Binary Cross-Entropy Loss, which is standard for binary classification tasks.
- Optimizer: Adam optimizer with a learning rate of 0.001.
- Batch Size: 16 images per batch.
- Epochs: I ran the model for multiple epochs, starting with a smaller subset of data to validate the pipeline before scaling up.

I used a validation split during training to monitor the model's performance and prevent overfitting.

Results and Analysis

After training the model, I generated predictions on the provided test set and formatted the results into a submission file. I uploaded this file to Kaggle and received the following leaderboard score: 0.7417

While the score was encouraging, there are several areas where the model could likely be improved:

- Training on the full dataset: Due to time and resource constraints, I initially worked with a smaller subset of the data. Training on the full dataset could improve performance.
- Data Augmentation: Techniques like random rotations, flips, or color shifts could help the model generalize better.
- Transfer Learning: Using a pre-trained model such as ResNet or VGG could leverage learned features from large image datasets to improve cancer detection performance.
- Handling Class Imbalance: Implementing class weighting or resampling methods could help address the label imbalance more effectively.

Conclusion

This project provided a valuable hands-on experience with deep learning, medical image classification, and real-world dataset challenges. I successfully loaded and explored a large medical imaging dataset, built and trained a CNN model using PyTorch, and generated predictions and submitted results to Kaggle for evaluation.

While my initial results showed promise, there is clear potential for improvement with more advanced techniques and more computational resources. I learned the importance of understanding data distribution, selecting appropriate evaluation metrics, and iterating on model design.

This project not only strengthened my skills in deep learning and model evaluation but also gave me experience participating in a competitive machine learning environment.

Github Link: https://github.com/mcflanaga/HW5-cancer-detection-3202