Standard Operating Procedure (SOP)

Ovarian Cancer Subtype Classification

Background and Rationale

Ovarian cancer is a leading cause of gynecological cancer mortality, characterized by five primary histological subtypes: Clear Cell Carcinoma (CC), Endometrioid Carcinoma (EC), High-Grade Serous Carcinoma (HGSC), Low-Grade Serous Carcinoma (LGSC), and Mucinous Carcinoma (MC). Accurate identification of these subtypes is crucial for effective diagnosis and personalized treatment strategies. This project aims to develop a deep neural network (DNN) model to classify ovarian cancer subtypes using histopathological image patches, thereby enhancing diagnostic precision and supporting clinical decision-making.

Dependencies

The following packages are required to execute this pipeline:

- `torch : 2.5.1` - Deep learning framework.
- `torchvision : 0.20.1` - Computer vision utilities for PyTorch.
- `tqdm = 4.67.1` - Progress bar utility for loops and processes.
- `numpy = 2.0.2` - Numerical computation library.
- `timm = 1.0.12` - Pretrained image models and utilities.
- `Pillow = 11.0.0` - Image processing library.
- `scipy = 1.13.1` - Scientific computing library.
- `pandas = 2.2.3` - Data analysis and manipulation library.
- `h5py = 3.12.1` - HDF5 file handling for large datasets.
- `scikit-learn = 1.5.2` - Machine learning library
- `matplotlib = 3.9.3` - Plot library

Dataset

The dataset must be organized as follows:

```
data/<dataset_name>/patches/<patch_size>/Mix/<subtype>/<slide_name>/<patch_size>/
```

Example:

```
data/MKobel/patches/1024/Mix/CC/112962/1024/20
```

- **Training Data:** 1 slide per subtype, totaling 1,000 patches (200 per subtype).
- **Test Data:** 1 slide per subtype, totaling 200 patches (40 per subtype).

You can Find the dataset at data folder on github.

Usage

- Install Nextflow (version: 24.10.1):

```
curl -s https://get.nextflow.io | bash \
&& chmod +x nextflow \
&& mv nextflow /usr/local/bin/
```

- Clone the GitHub repository:

```
git clone https://github.com/elirn98/cancer-subtyping-nextflow-pipeline
```

- Download the Docker image:

```
docker pull elirn98/cancer_subtyping_nextflow_pipeline:v1
```

- Navigate to the application directory:

```
cd /app
```

- Run the pipeline:

```
nextflow run pipeline.nf -with-docker cancer-subtyping:v1.0
```

Workflow Steps

1. **Training Phase:**
   - **Input:**
     - `MODEL` - Model name.
     - `params.SAVE_PATH` - Directory to save model checkpoints and statistics.
     - `params.DATA_PATH` - Path to the training dataset.
   - **Action:** Trains the DNN model using the provided dataset.
   - **Output:**
     - Trained model checkpoint.
     - Training statistics (accuracy and loss).
2. **Testing Phase:**
   - **Input:**
     - `hoptimus0_model` - Path to the trained model.
     - `params.SAVE_PATH` - Directory to save testing results.
     - `params.DATA_PATH` - Path to the testing dataset.
   - **Action:** Tests the trained model on the test dataset.
   - **Output:**
     - Testing results (accuracy and loss).
3. **Visualization Phase:**
   - **Input:**
     - `model_statistics` - Training statistics.
     - `test_results` - Testing statistics.
     - `params.PLOT_PATH` - Path to save plots.
   - **Action:** Generates visualizations such as accuracy vs. epoch, loss vs. epoch, and ROC curve.
   - **Output:** Plots saved to the specified directory.

Input

- **Training Phase:** Model name, training dataset path, and checkpoint save path.
- **Testing Phase:** Trained model path, test dataset path, and results save path.
- **Visualization Phase:** Training and testing statistics files, and output directory for plots.

Output

- **Training Phase:** Trained model checkpoint and training statistics.
- **Testing Phase:** Testing results (accuracy and loss).
- **Visualization Phase:** Generated plots (Precision Recall Curve, Confusion Matrix).

Expected Results

Note that having 1 slide per subtype is not enough for training such a model, that's why some of the results are not as expected. This is due to limitations in memory, gpu, and run time.

5 Class Dataset Training:

- Dataset: 5 slides for training (1,000 patches) and 1 slide for testing (200 patches).
- Epochs: 1.
- Outputs:
  - Training Accuracy = 90.2
  - Training Loss = 0.396
  - Test Accuracy = 78.5
  - Test Loss = 0.568

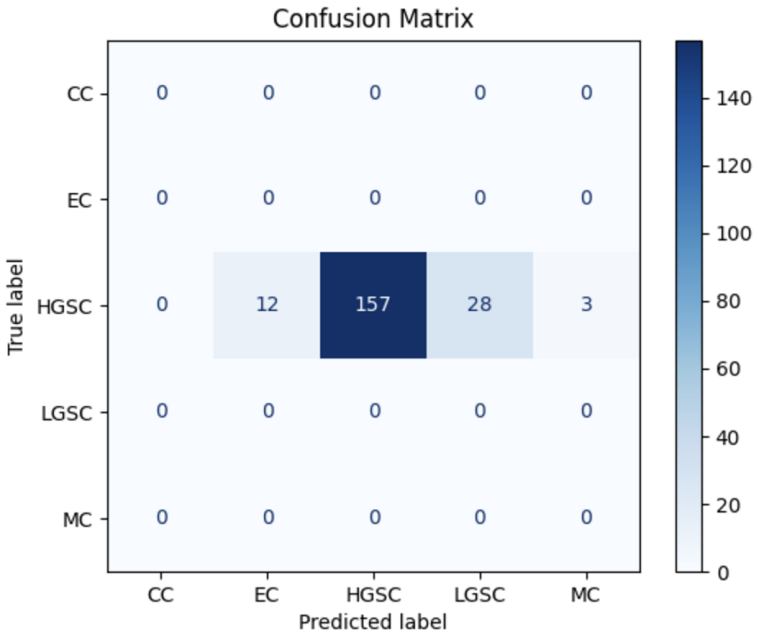

Figure 1: Confusion Matrix

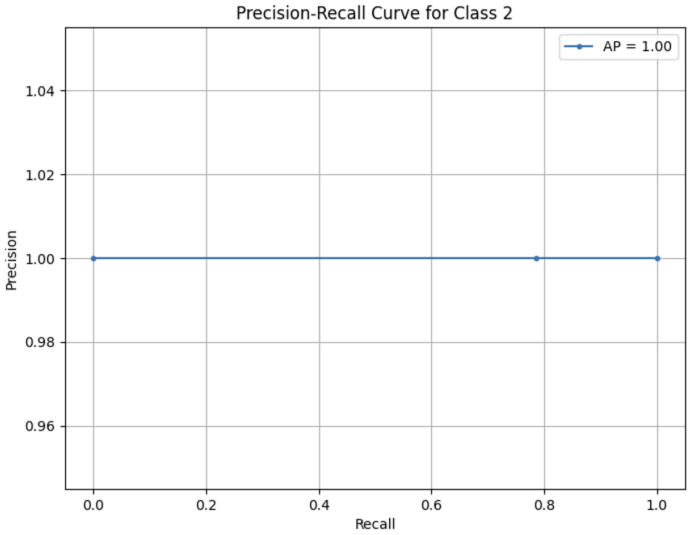

Figure 2: Precision Recall Curve

Quick Test (Reduced Dataset):

Running this pipeline takes about 20 minutes on cpu.

- Dataset: 1 slide (200 patches) for training and 1 slide (200 patches) for testing.
- Epochs: 1.
- Outputs:
  - Training Accuracy = 90.5
  - Training Loss = 0.414
  - Test Accuracy = 100
  - Test Loss = 0.0466 - Test Accuracy = 100

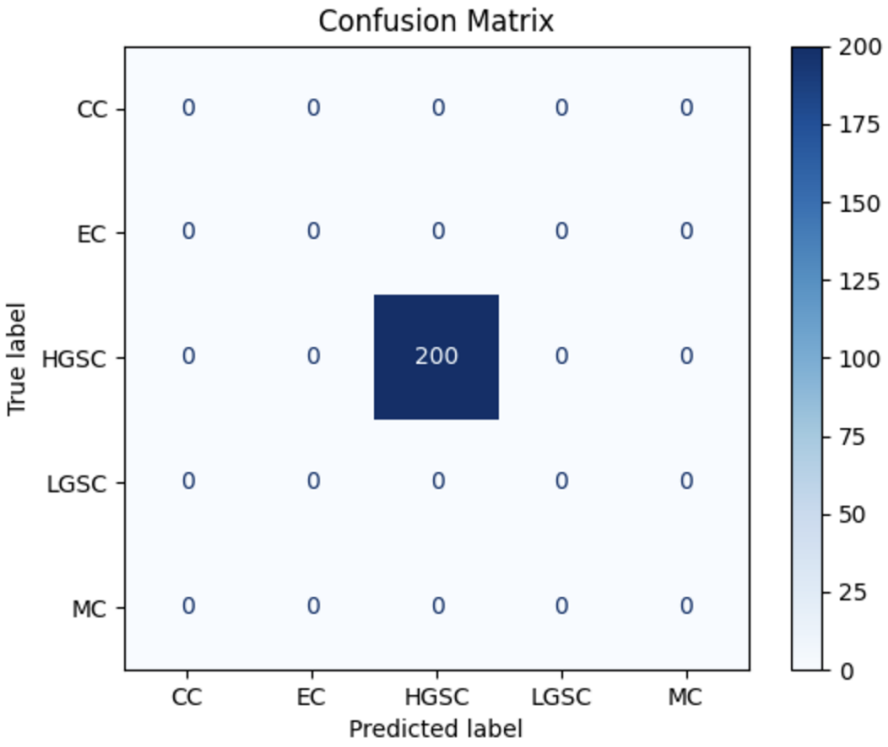

Figure 3: Confusion Matrix

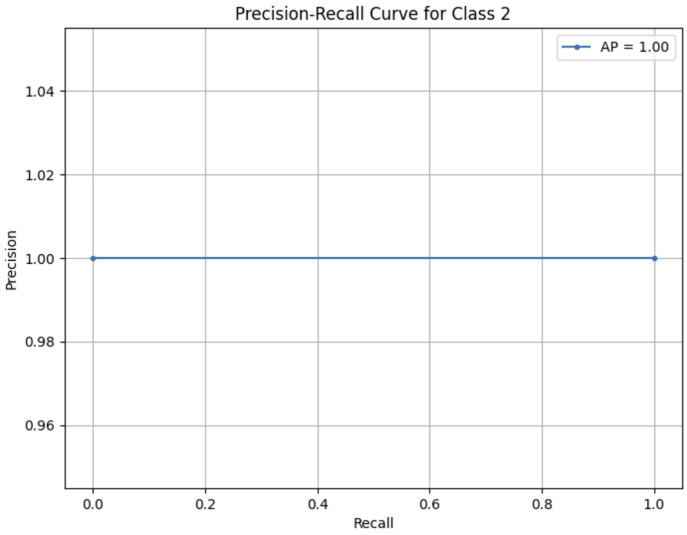

Figure 4: Precision Recall Curve