

CONCH: Implementation Report and Challenges (ver-2)

Dataset Description:

Dataset: <https://www.kaggle.com/datasets/mahmoodyousaf2018/patched-dataset-wsi-for-kat>

The dataset contains patches of WSI patches sorted by patient ID, where the first 12 characters of each directory refers to the patient ID (eg. TCGA-3L-AA1B). A csv file effectively corresponds the actual class (msi/mss) of each slide to patient ID, this helps us sort our data into msi and mss classes.

Structure:

```
patched-dataset-wsi-for-kat/  
└──  
    TCGA-3L-AA1B-01Z-00-DX1.8923A151-A690-40B7-9E5A-FCBE  
    DFC2394F/  
        ├── Small.txt  
        ├── Overview.jpg  
        ├── Overview.txt  
        ├── Small/  
        └── Overview/
```

To make the dataset intuitive/user-friendly, we sorted it into the following structure:

Structure:

```
patched-dataset-wsi-for-kat/  
├── train  
│   ├── msi  
│   └── mss  
└── test  
    ├── msi  
    └── mss
```

The files inside of mss/msi directories contained the contents of the Overview folder of each patient directory, the patient ID of the respective patches was appended to the filenames of each patch for bag level classification.

Tasks Assigned:

1. Share text prompts used previously in zero-shot classification, along with a brief explanation of its significance to the task at hand. (completed)
2. Calculate bag level metrics on fine tuned CONCH. (attempted)
3. Explore applications of CONCH including (**tissue segmentation**, captioning, image-to-text retrieval). (pending)

Note: In approaching task 2, tissue segmentation was assigned as the primary focus by our team lead/ supervisor (Sir Aamir).

Stepwise breakdown (Activities):

1. Fine Tuned model:
 1. Froze pre-trained layers to retain learned features.
 2. Added a new classification head
 - a. Fully connected layer (`nn.Linear(visual_output_dim, 256)`) reduced dimensionality to 256.
 - b. Activated by ReLU.
 - c. Dropout layer ($p=0.5$) after the first linear layer.
 - d. Fully connected layer (`nn.Linear(256, 128)`) further reduced dimensionality to 128.
 - e. Activated by ReLU.
 - f. Dropout layer ($p=0.5$) after the second linear layer.
 - g. Final layer (`nn.Linear(128, num_classes)`) outputs predictions for `num_classes` classes
 3. Unfroze and fine-tuned only the classification head.
 4. Used cross-entropy loss and Adam optimizer for training.

Summary:

Input dimension: 512

Layers:

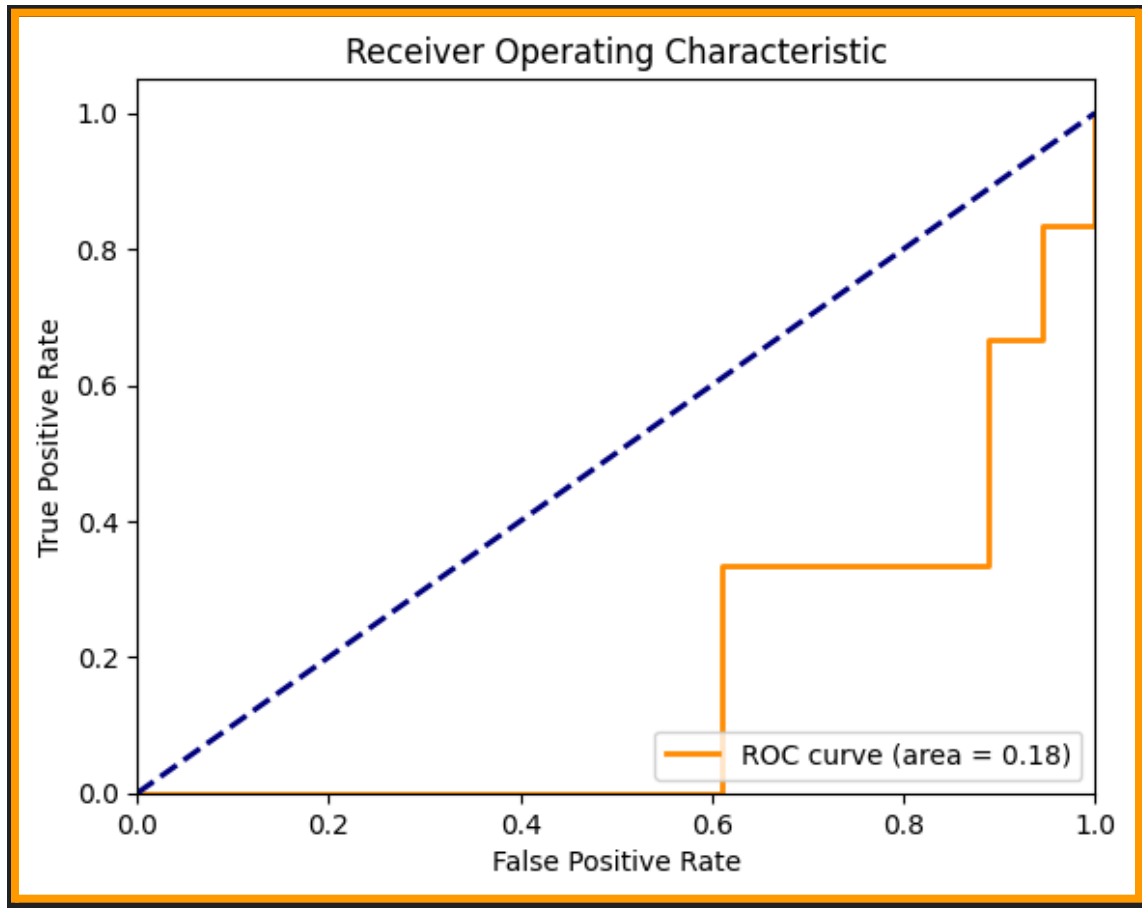
- Fully connected layer (512 -> 256) with ReLU activation and 50% dropout
- Fully connected layer (256 -> 128) with ReLU activation and 50% dropout
- Final layer (128 -> 2) for binary classification

Results:

- Calculated bag level auc and f1 score:

Patient-level AUC: 0.176

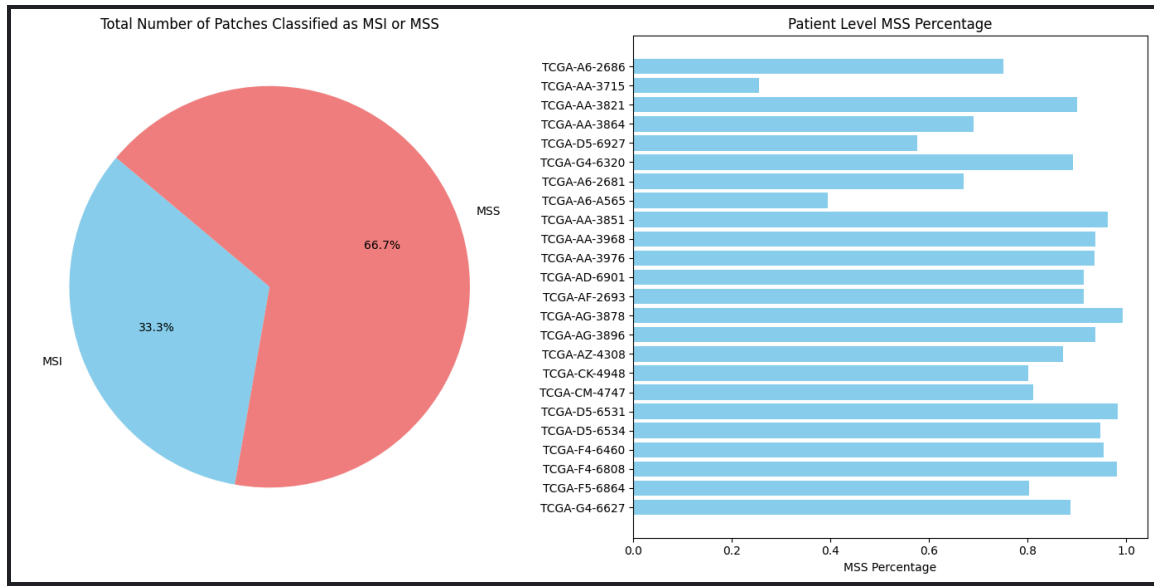
Patient-level F1 score: 0.357



- Calculated mss percentage (number of patches in a slide predicted as mss) of each bag (patient slide):

- TCGA-A6-2686 mss percentage: 0.7508833922261484
- TCGA-AA-3715 mss percentage: 0.2551622418879056
- TCGA-AA-3821 mss percentage: 0.9005328596802842
- TCGA-AA-3864 mss percentage: 0.6898638426626323
- TCGA-D5-6927 mss percentage: 0.5754716981132075
- TCGA-G4-6320 mss percentage: 0.8928571428571429
- TCGA-A6-2681 mss percentage: 0.6698564593301436
- TCGA-A6-A565 mss percentage: 0.3939393939393939
- TCGA-AA-3851 mss percentage: 0.9633333333333334
- TCGA-AA-3968 mss percentage: 0.937888198757764
- TCGA-AA-3976 mss percentage: 0.9356223175965666
- TCGA-AD-6901 mss percentage: 0.9142857142857143
- TCGA-AF-2693 mss percentage: 0.9148936170212766
- TCGA-AG-3878 mss percentage: 0.9936305732484076

- TCGA-AG-3896 mss percentage: 0.937984496124031
- TCGA-AZ-4308 mss percentage: 0.8713550600343053
- TCGA-CK-4948 mss percentage: 0.8019257221458047
- TCGA-CM-4747 mss percentage: 0.8112118713932399
- TCGA-D5-6531 mss percentage: 0.9831649831649831
- TCGA-D5-6534 mss percentage: 0.947856315179606
- TCGA-F4-6460 mss percentage: 0.9536354056902002
- TCGA-F4-6808 mss percentage: 0.9815837937384899
- TCGA-F5-6864 mss percentage: 0.8032454361054767
- TCGA-G4-6627 mss percentage: 0.8875793291024479



Training Parameters:

- Patience for Early Stopping: 10 epochs
- Minimum Delta for Early Stopping: 0.001

Environment Settings:

Dataset Transformation:

- Applied transformations:
 - Convert to tensor
 - Normalize with mean=[0.485, 0.456, 0.406] and std=[0.229, 0.224, 0.225]
- Dataset details:
 - Number of training samples: 72652
 - Number of validation samples: 15633

Model Architecture:

- **Pretrained Model:**
 - Used a pretrained model (specified model Conch_ViT-B-16) for feature extraction.
- **Custom Classification Head:**
 - Method mentioned above in detail activity 1.

Training Parameters:

- **Optimizer:**
 - Adam optimizer with learning rate: 1e-4
- **Loss Function:**
 - CrossEntropyLoss

Training Setup:

- **Batch Size:** 32
- **Number of Epochs:** 100
- **Stopped at:** ~(11-16 epochs)
- **Device:** GPU P100

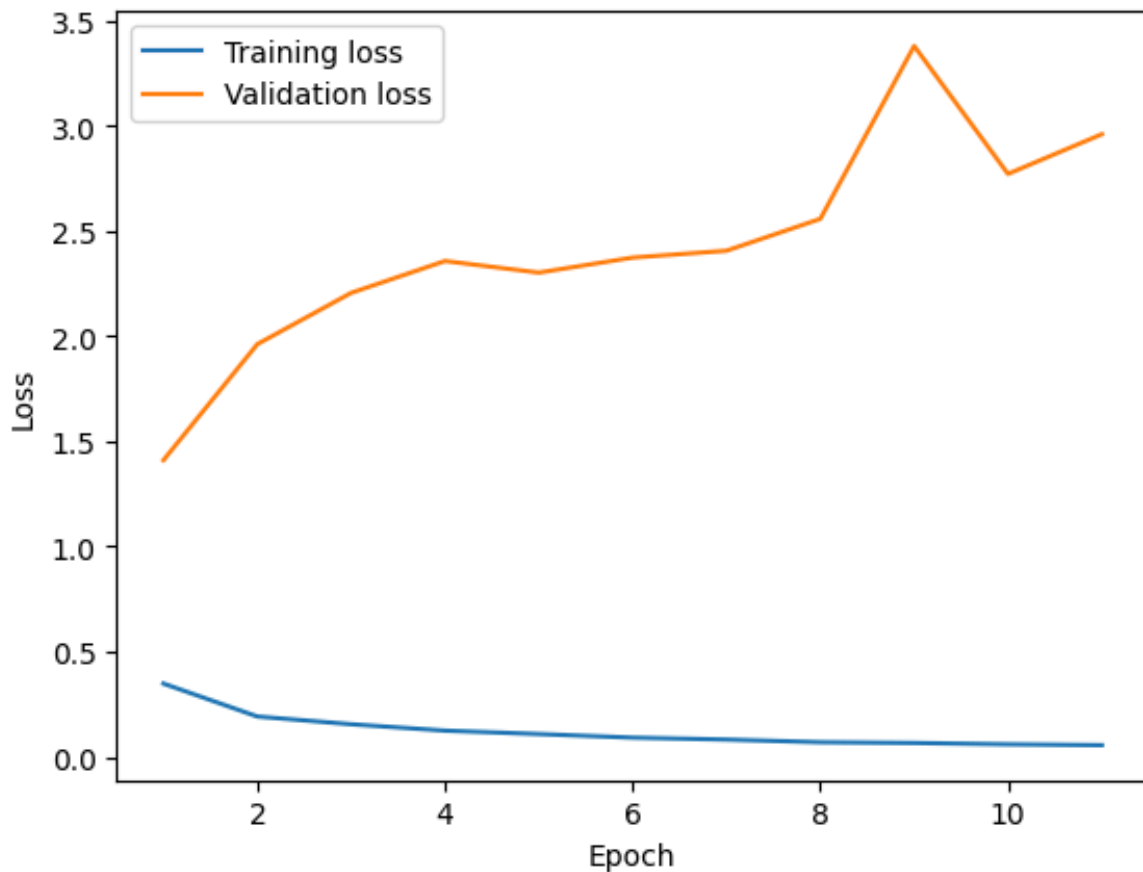
Challenges:

1. In our initial attempts, we were faced with extreme overfitting, upon analyzing data, we were brought to the conclusion that this was primarily due to a data imbalance caused by a greater number of mss samples than msi ones:

Total MSI paths: 28

Total MSS paths: 88

We believe this caused overfitting, causing our loss curves to look similar to this:



To potentially resolve this challenge, we limited the number of mss samples (which were clearly abundant) to match the total number of available msi samples. This approach helped improve our results slightly, however the overfitting remained:

Total MSI paths: 28

Total MSS paths: 28

Total MSI patches: 23641

Total MSS patches: 19517

Train MSI patches: 19449

Test MSI patches: 4192

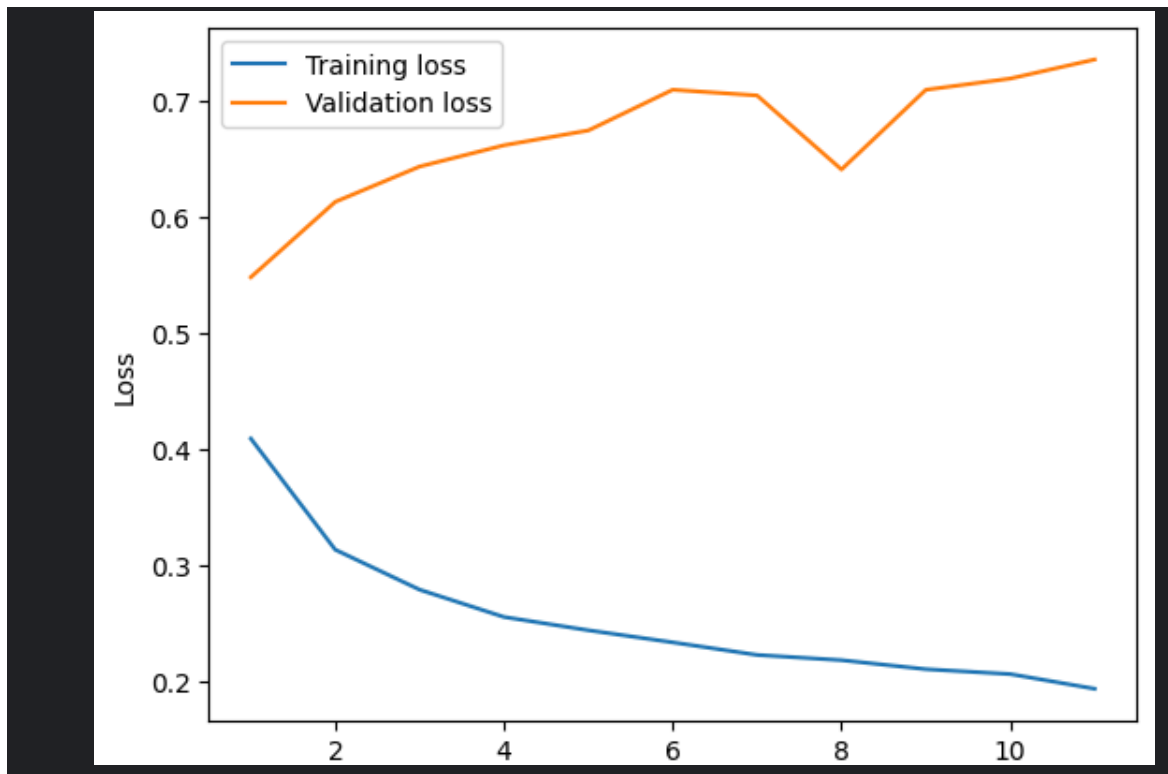
Train MSS patches: 16615

Test MSS patches: 2902

{0: 'msi', 1: 'mss'}

Number of training samples: 36064

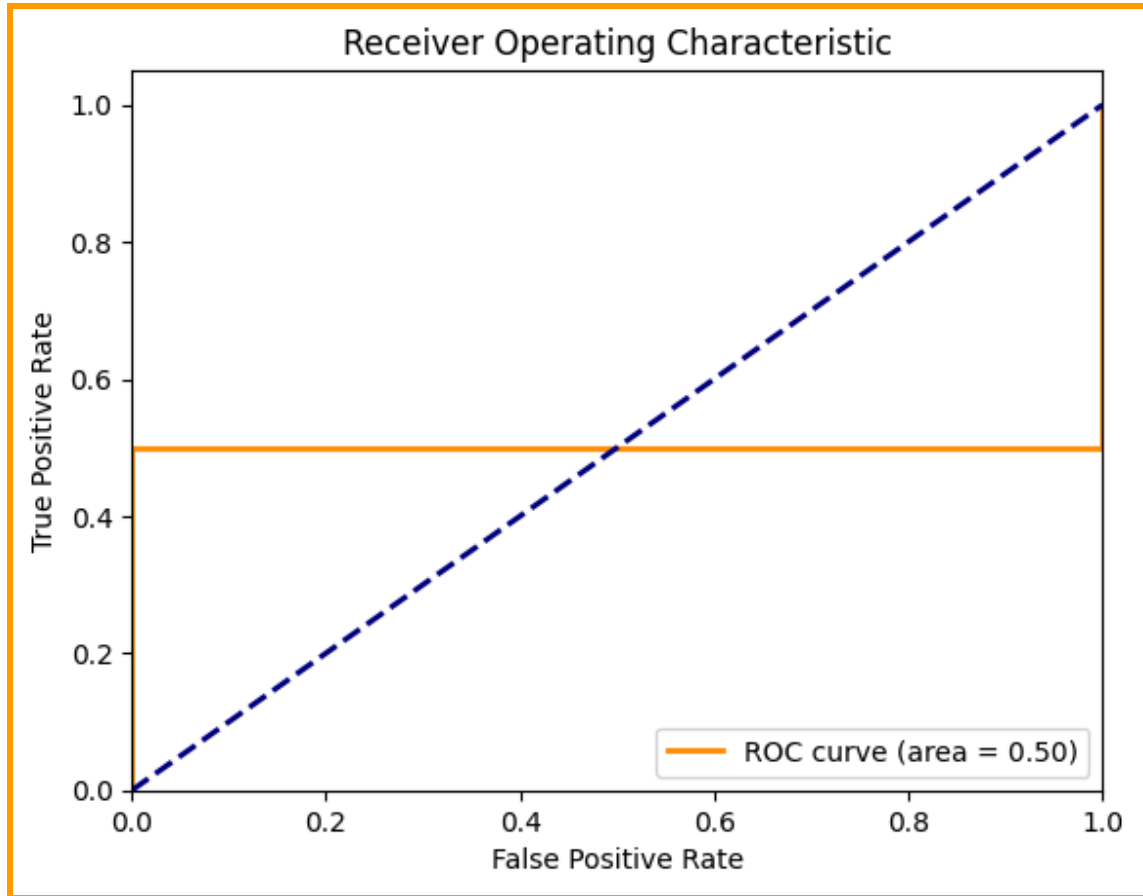
Number of validation samples: 7094



Our results after implementing this approach:

Patient-level AUC: 0.500

Patient-level F1 score: 0.667



2. Task 2, which was tissue segmentation, is highly reliant on the task of image classification. The methodology of image segmentation, for clarity is summarized below:
 1. Divide WSI into tiles (patches).
 2. Classify tiles using classification.
 3. Assign predicted class label to all pixels in each tile (patch).
 4. Tile WSIs with 75% overlap.
 5. Average prediction scores in overlapping regions.

Without a well-performing classification model, we are at a stalemate and cannot effectively proceed with segmentation and other related tasks.

Questions:

1. No data for generating captions as in technical biological data for tumor and tissue descriptions, morphology (like
2. Ground truth labels for patches not present (we have very few annotated images, not sufficient) , without those we cannot do segmentation, unless our classification model is capable of producing ground truth (100% accuracy) which is impossible.