Mitosis Detection

Ashna Chalil Abraham
23116870
ashna.abraham@fau.de

Abstract—**In this project report, we delve into mitosis detection through the training of a detection model specifically designed to identify mitotic figures. Our approach involves fine-tuning a pre-trained RetinaNet model tailored to our unique use case. Subsequently, we present a comprehensive analysis of the performance exhibited by these models.**

Index Terms—**mitosis detection, object detection, deep learning**

I. Introduction

Mitotic cell density, often quantified as the mitotic count, serves as a crucial indicator of tumor behavior and prognosis. Mitosis, the process by which cells divide to form new cells, is typically heightened in rapidly proliferating tissues, such as tumors. Identifying mitotic figures is critical for tumor grading, prognostication and treatment selection.

In this report, we share our insights from training a mitosis detection model with images collected from diverse scanners. Sections II and III delve into distinct aspects concerning dataset composition and implementation details outlined in the programming tasks. Section IV assesses the model performance across diverse hyperparameter configurations on our designated test set. Following this, Section V investigates the influence of domain shift on model performance. Finally, Section VI consolidates our findings and conclusions.

II. Task1

Task 1.1

What are the essential characteristics of the data and the labels?

The dataset has 200 cropped images of human breast cancer from whole slide images (WSI) with 50 images each from four different scanners. The data set has two class labels: the positive (mitotic figures) and negative (non-mitotic figures) cells. The bounding box coordinates are given for the corresponding labels.

Task 1.2

Note that there is something wrong with how the test and training set are being used. What is the problem in this case and why is it a problem? Correct the problem to be able to train the networks with the dataset at hand. Why is the training set provided like this by the MIDOG challenge?

The 50 images from the fourth scanner are provided without annotations. This means if we utilize these samples for training and testing our training algorithm might see misclassified patches (e.g., we take a random patch from sampling with a mitotic figure but since there is no annotation, we mark it as non-mitotic). This will feed our model with wrong information. To counter it, we filter out the samples without annotation before training and testing.

The challenge organizers explain that they provide this non-annotated set for unsupervised domain adaptation approaches.

Task 1.3

What are the important parameters for this network and why are they set this way? What would be further interesting parameters to adapt?

This network generates a RetinaNet model with

- a ResNet-50-FPN backbone and feature pyramid network (FPN) for feature extraction.
- Downstream classification head to predict class probabilities and bounding boxes.

Both of these parts are re-trainable.

We initialize the model with weights pretrained on the COCO dataset [1], leveraging its comprehensive object detection, segmentation, and captioning annotations. The parameter `trainable_backbone_layers` is utilized to fine-tune feature extraction, crucial for adapting the network to our specific task.

Key network parameters include the optimization method, initial learning rate, learning rate scheduler, batch size, and number of training epochs.

Additionally, parameters such as patch size for training and inference, and the number of patches per slide in training and validation data, could be further adjusted to optimize model performance.

III. Task 3

What would be a good way to uniquely associate patches with a continuous index in Test Dataset?

We can split the entire Region of Interest (ROI) into multiple overlapping patches, depending on a specified threshold for overlap. Subsequently, we can store the coordinates of these patches as a list. Upon accessing a patch by its index, we retrieve the corresponding coordinates, crop the patch from the entire image, and return it alongside its corresponding label.

IV. Task 4

Train the network architecture for mitosis detection and compare different hyperparameters. Export the results in a suitable format and report the final results on the test set.

If not stated otherwise, all models undergo training under the following configuration:

- Threshold score for accepting mitotic figures: 0.3

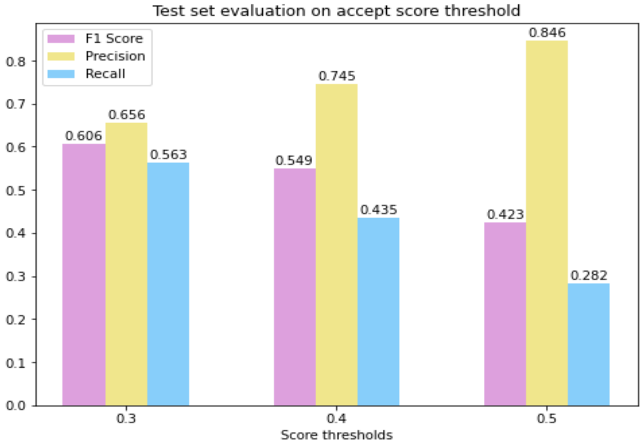

Fig. 1: Test set evaluation on accept score threshold

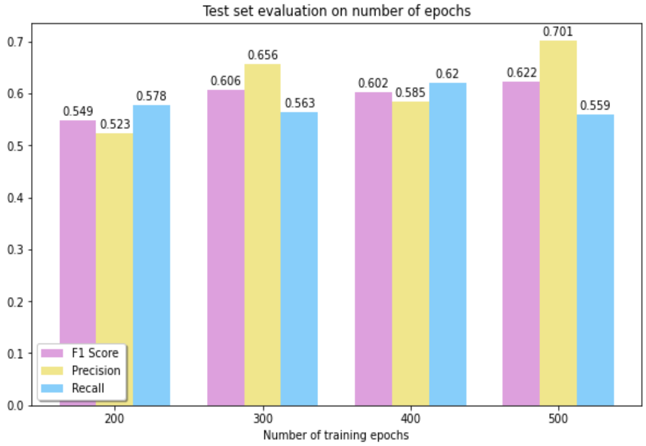

Fig. 2: Test set evaluation on number of epochs

- Patch size for training and testing: 512
- Training batch size: 8
- Training duration in epochs: 300
- Number of patches per slide for both training and validation data: 10

A. Threshold Score for identifying mitotic figures

This denotes the minimum probability score assigned by the model for identifying patches containing mitotic figures. Our training was conducted using three threshold values, with results depicted in Figure 1.

As the threshold increases, precision improves while recall decreases. This phenomenon is attributed to the highly unbalanced nature of the data, where mitotic patches are outnumbered by impostor patches. Consequently, we proceeded with further experiments using a lower threshold of 0.3.

B. Number of training epochs

We conducted model training across four epochs while maintaining other parameters constant, as depicted in Figure 2.

Evidently, our model exhibits signs of underfitting at 200 epochs. As we extend the training duration to 300 epochs, all performance metrics exhibit improvement. However, at 400 epochs, although recall experiences a slight enhancement, precision values start to decline. Notably, in the case of 500 epochs, there is a shift in behavior, with precision showing improvement while recall decreases.

C. Patches per slide

"Patches per slide" refers to the quantity of patches extracted from each whole slide image during training and validation. Our experimentation involved three settings: 10, 15 and 20 patches per slide, with results illustrated in Figure 3.

Augmenting the number of patches yielded enhanced model performance. Notably, with 20 patches per slide, precision showed improvement, but with a decline in recall.

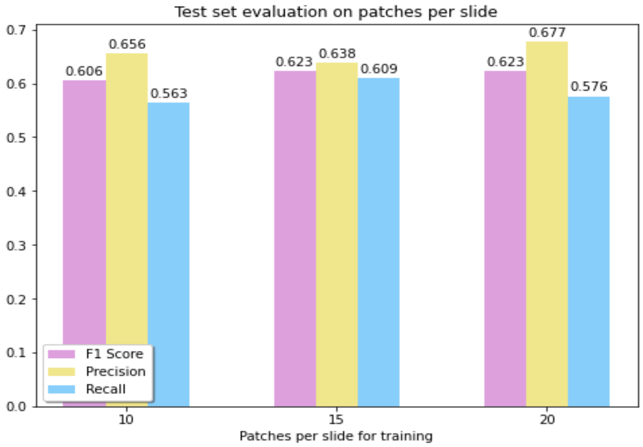

Fig. 3: Test set evaluation on patches per slide

D. Patches size

This refers to the patch dimension on which we train our model. We use square patches of sizes 512*512 and 256*256. The evaluation results of both setups can be referred to in figure 4.

When reducing the patch size to 256, a huge decline in precision value is observed. This could be attributed to fewer mitotic figures encountered by our model owing to the smaller window size. To address this, increasing the number of patches per slide might prove effective.

V. Domain shift

This programming exercise is closely tied to my seminar topic on domain shift and generalization. The task highlights domain shift through variations in visual aspects observed in images from different scanners. To address this challenge, we employ various geometric augmentation techniques aimed at mitigating domain shift effects. However, there exist numerous sophisticated methods specifically designed to tackle this issue, which we can explore further to enhance the performance of our model.

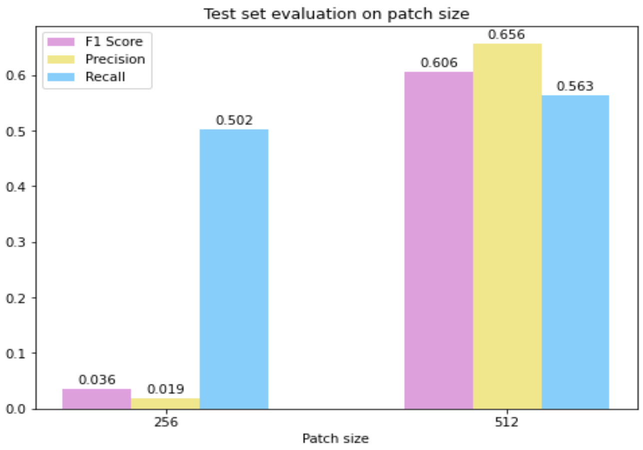

Fig. 4: Test set evaluation on patches per slide

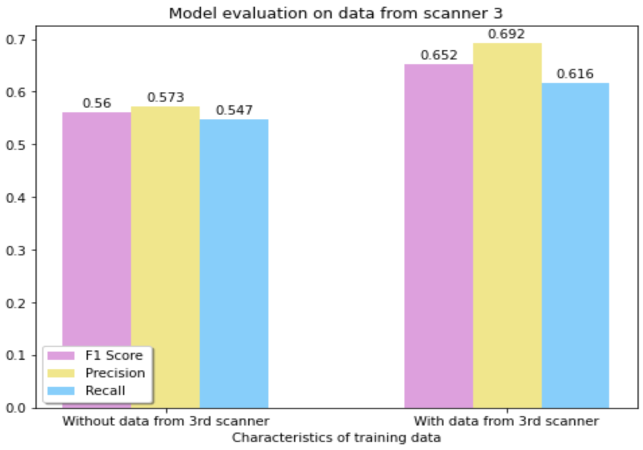

Fig. 5: Evaluation of the effect of domain shift

To demonstrate the impact of domain shift, we conducted an experiment where we excluded data from the third scanner. Subsequently, we trained our model using the complete dataset from the first two scanners and evaluated its performance on images from the third scanner. The results of this experiment are depicted in Figure 5.

We've noticed a significant drop in model performance when excluding a specific scanner from training, indicating a domain shift. To mitigate this issue, we can implement domain generalization methods to enable generalization across various domains.

VI. Conclusion

Exploring various hyperparameters underscored the significance of addressing class imbalance within our training dataset. Rather than simply adjusting thresholds, optimizing our sampling function to provide our model with an increased number of patches containing mitotic figures during training could prove more effective. Additionally, domain shift emerges as a pivotal concern in digital pathology image analysis. Thus, integrating domain generalization methods becomes imperative to mitigate its adverse effects.

References

[1] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft coco: Common objects in context," 2015.