CS5242 Project Report
Terence Kong
E1287893

Background

Nuclei segmentation, classification and detection are an important but challenging task for the biomedical industry. Morphological changes in the cell nucleus are an important signal for detecting many diseases, especially cancers. The conventional task requires manual inspection from pathologists. However, the task is laborious and underscores the need to develop automated image analysis methods as an alternative means to addressing the issue.

Convolutional Neural Networks have been the dominant technique for visual analysis and have achieved great success in object detection and segmentation in medical images. U-Net is a widely used architecture based on contracting encoder layers and expansive decoder layers connected by skip connections to generate a segmentation map. Its primary purpose was to address the challenge of limited annotated data in the medical field.

Cell staining is used to visualize cells and cell components which are then captured and used to train models. However, due to the complexity of nuclei shape, differences and imperfections in staining procedures, overlapping nuclei and scanning artifacts, nuclei instance segmentation can still prove challenging.

To overcome this, we look at how fine tuning techniques can be used to improve adaptation to localized procedures, improve performance and overcome the challenges of limited data.

Methodology

Architecture

As mentioned, the primary focus of this paper is to experiment with using U-Net to do segmentation and classification and to explore how U-Net performs with additional information. We chose U-Net as it is able to function with very few annotated images and has a very reasonable training time on weaker GPUs.

Pre Training

Since U-Net is a segmentation based architecture and requires labels, we only train on the CAM16 dataset for pretraining. The provided dataset only has a small portion of the dataset come with annotated masks, but since we are using U-Net, we are still able to get decent results.

Initially, work was done to attempt to generate more segmentation masks using conventional image processing programs such as ImageJ. However, it was difficult to come up with accurate segmentation masks and the use of suboptimal masks only led to worse results.

Hyperparameters

The following hyperparameters were used for this project:

- Adam optimizer with a learning rate of 0.0001
- Binary Cross Entropy
- Batch Size = 4

Since we are performing an image segmentation task, accuracy was not a very good representation of the effectiveness of the model as images tend to be very imbalanced (e.g. normal lymph nodes were classified as all 0s and a model could get good accuracy by just predicting 0s)

As such, we explored various metrics in order to better determine the effectiveness of our model. The following metrics were tested:

- F1
- Dice
- Jaccard (IoU)

Data Augmentation & Regularisation

Since we are only using a small portion of the dataset, we need to rely on data augmentations to prevent overfitting and allow our model to generalize better.

The following augmentations were experimented with (PyTorch):

- Horizontal and vertical flips
- 90 degree rotations
- Affine transformations (shifting, sheers, scaling)
- Normalisation

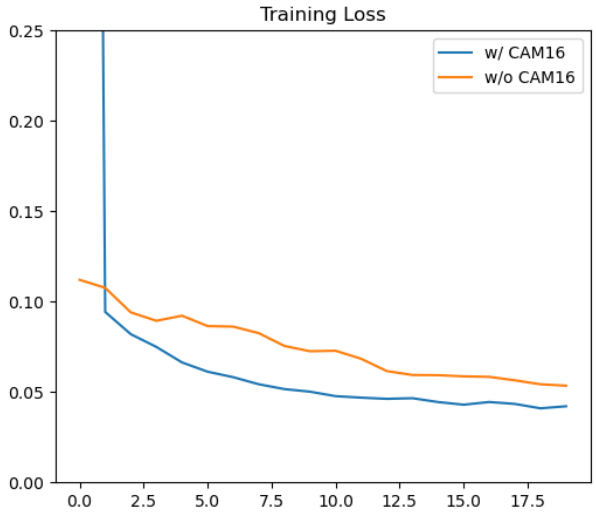
Fine Tuning

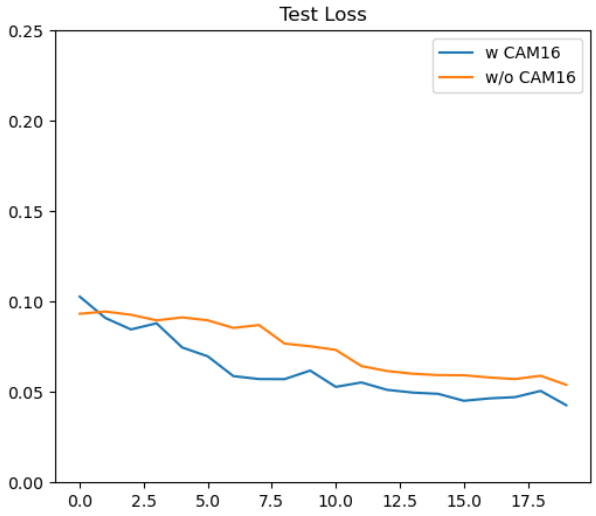
For fine tuning, we initialize our model with the pre-training weights from CAM16 and then proceed to retrain the entire model on the new dataset. The only modifications that were made to the pre-training architecture was changing the output layer from a binary classification to a multilabel classification (5 channel output).

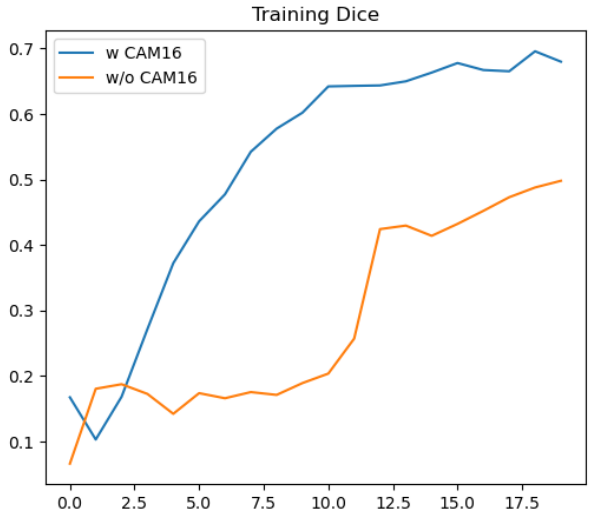
Initially, we explored freezing layers from the pre-training model, but were not able to get good results.

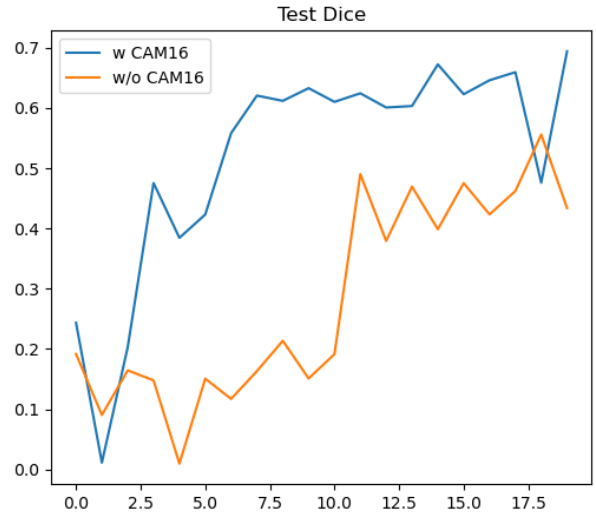
Results

WBC 100

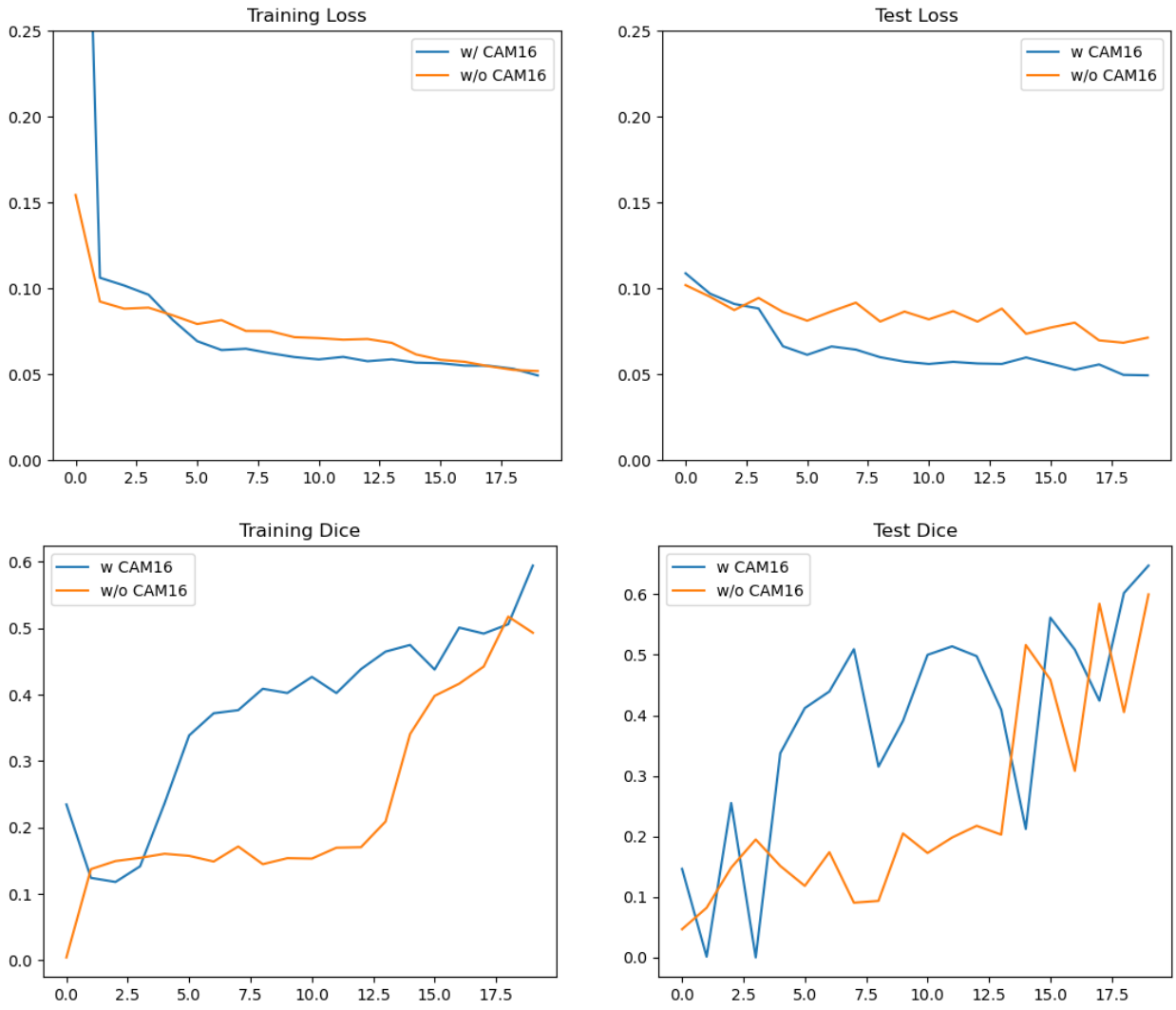




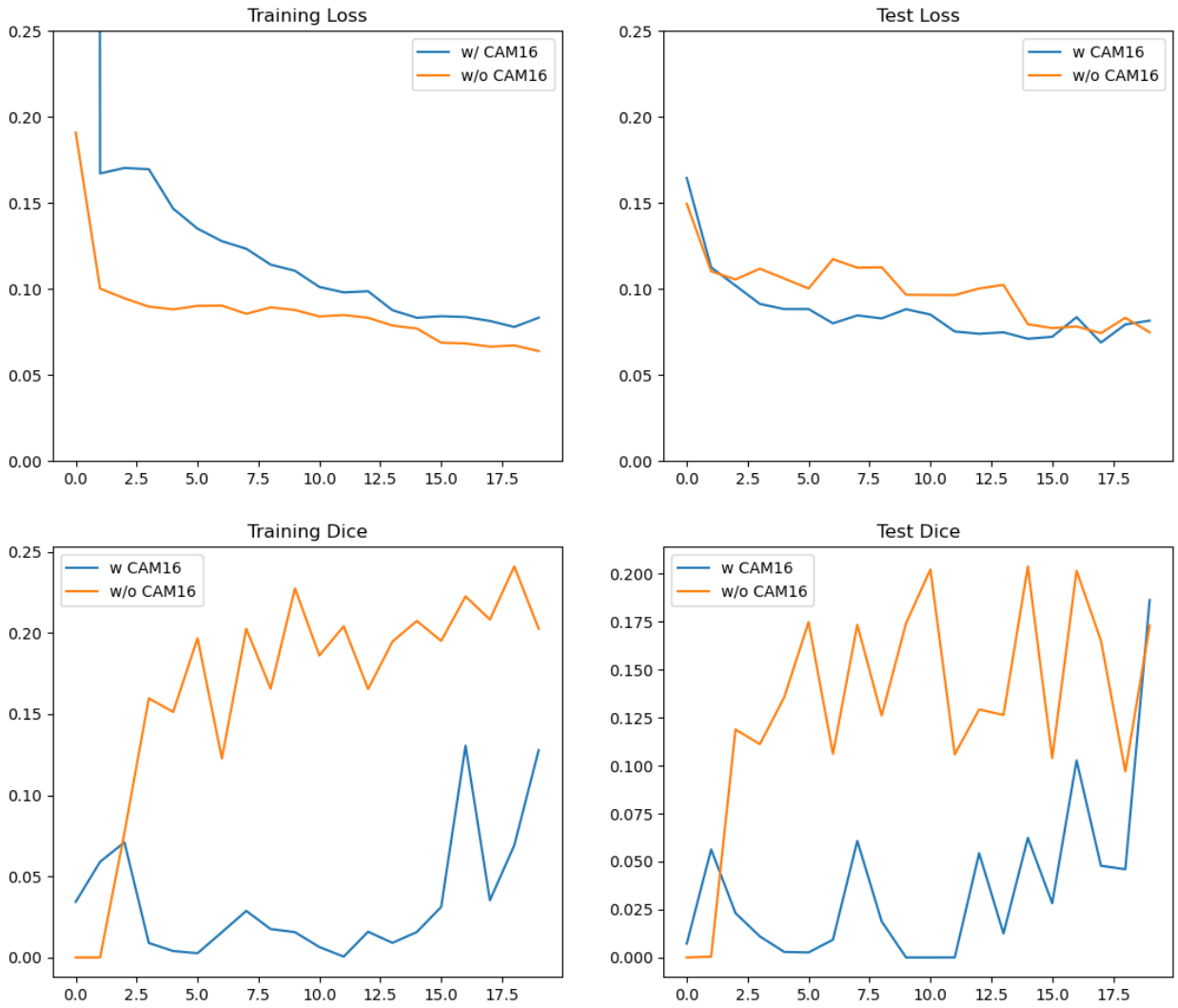




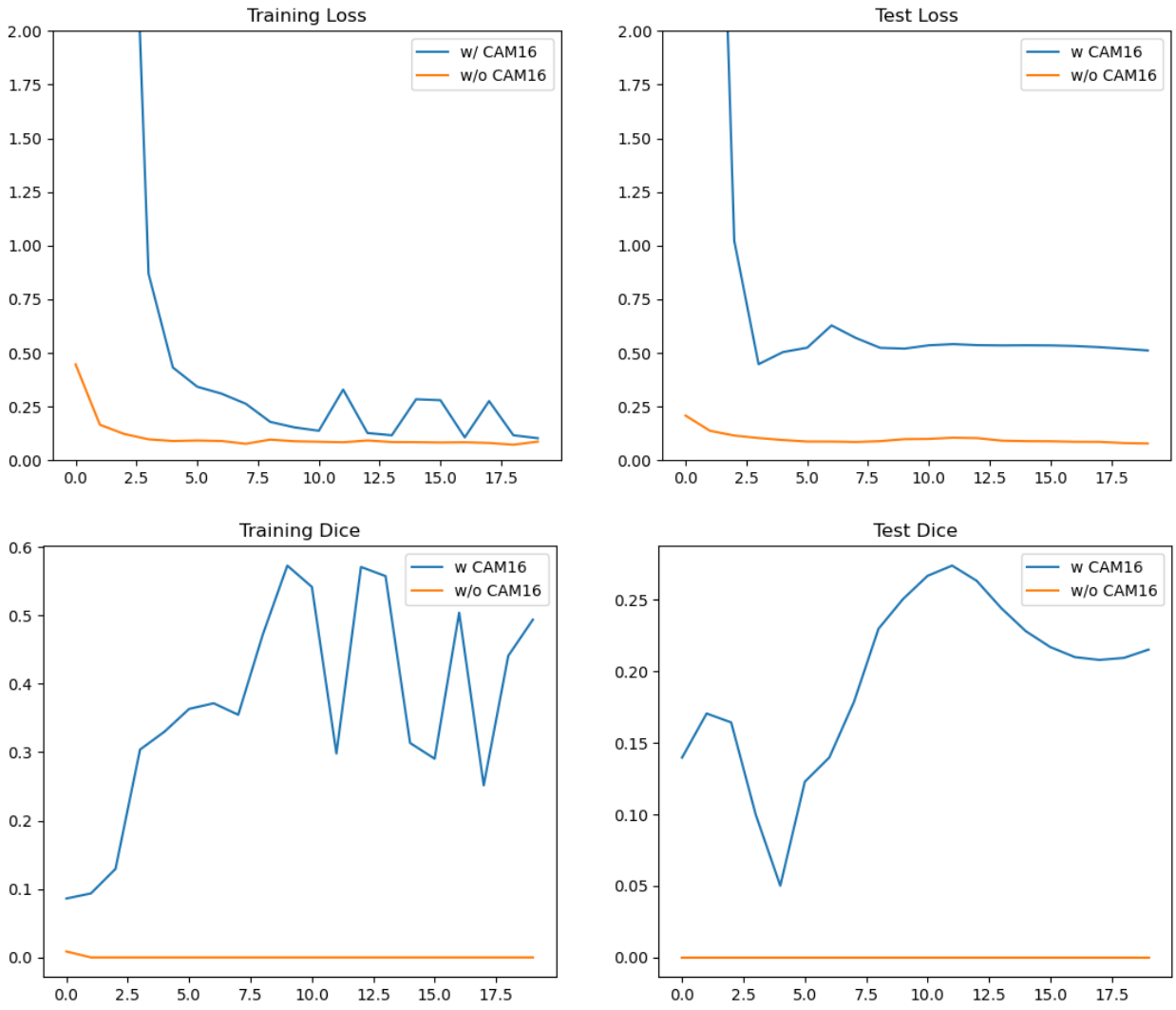
WBC 50


Training Loss
w/ CAM16
w/o CAM16
Test Loss
w CAM16
w/o CAM16
Training Dice
w CAM16
w/o CAM16
Test Dice
w CAM16
w/o CAM16

WBC 10


Training Loss
w/ CAM16
w/o CAM16
Test Loss
w CAM16
w/o CAM16
Training Dice
w CAM16
w/o CAM16
Test Dice
w CAM16
w/o CAM16

WBC 1



Conclusion

In general, the models seem to perform better with the fine tuned data across the board (WBC 1-100). WBC 1 in particular showed significantly better performance on the dice metric without the additional information, despite having a lesser loss. This seems to imply that when the dataset is sufficiently large, fine tuning with additional information might provide a marginal benefit, but fine tuning is especially useful when the dataset is extremely small. Additionally, we also find that training accuracy is generally a poor fit as a metric for evaluating a model's performance and that metrics like the dice coefficient are more robust to imbalanced classes, as it will not be overly penalized for failing to detect a small region.

References

1. J. C. Caicedo, A. Goodman, K. W. Karhohs, B. A. Cimini, J. Ackerman, M. Haghighi, C. Heng, T. Becker, M. Doan, C. McQuin, M. Rohban, S. Singh and A. E. Carpenter, “Nucleus segmentation across imaging experiments: the 2018 Data Science Bowl,” in *Nature Methods*, vol. 16, no. 12, December 2019. [Online]. Available: https://www.nature.com/articles/s41592-019-0612-7
2. Kaggle, ‘2018 Data Science Bowl,’ Kaggle, 2018. [Online]. Available: https://www.kaggle.com/c/data-science-bowl-2018/discussion/54741
3. DigitalSreeni, “208 - Multiclass semantic segmentation using U-Net,” YouTube, 2019. [Online]. Available: https://www.youtube.com/watch?v=XyX5HNuv-xE&t=3s

Appendix