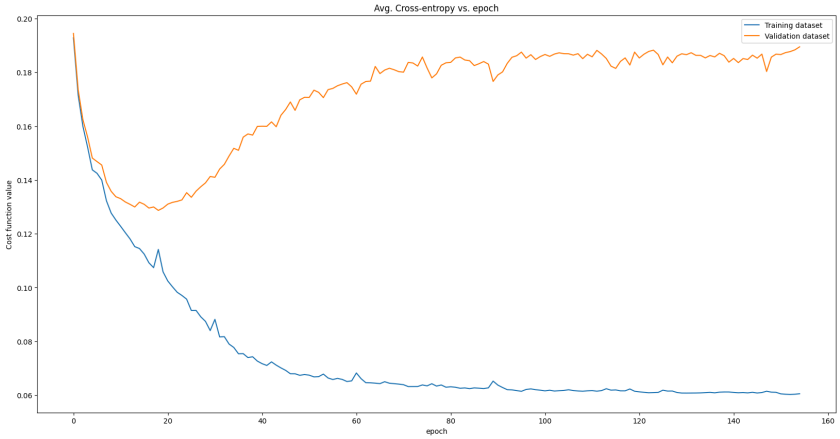
Ailin Wu
May 16, 2024

# Final Project Report: Counting Cells in Microscopy Images

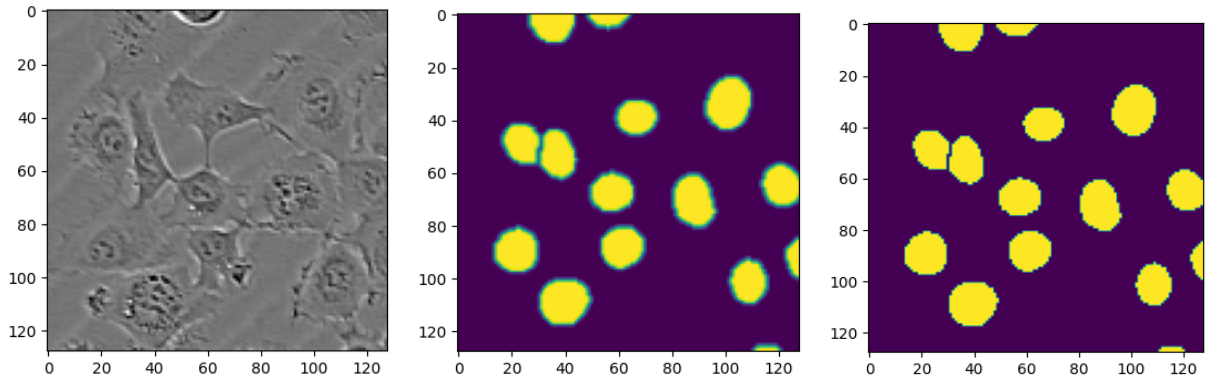
In the Kaggle contest, the training data consists of 2000 microscopy images of cells, stored in a numpy array X, and corresponding segmentation masks stored in a numpy array y. Each image (and each corresponding mask) has shape (128, 128). The segmentation mask y[i] for image X[i] contains "soft labels" (that is, numbers between 0 and 1) which tell you which pixels in the image X[i] belong to a cell nucleus.

In this project, I aimed to count the number of cells by answering the question “How many cells appear in a given microscopy image?”. I start off by creating a separate dataset class for the labeled data and the test data because the getitem method will work differently in each class. Then I have created the dataset and dataset loader objects. I implemented the U-Net architecture with 23 convolutional layers in PyTorch. The first 10 layers are applied by max pooling and the next 13 layers are applying upsampling and convolutional neural networks. The U-Net architecture trains the training data to yield more precise segmentation tasks. Afterwards, I have created a model and trained it for 155 epochs with a learning rate of 0.0003. In the training process, I have computed the Average Cross-Entropy Loss Function for both the training dataset and the validation dataset. The plot below shows the convergence of the Average Cross-Entropy Loss Function for both training dataset and validation dataset over 155 epochs.

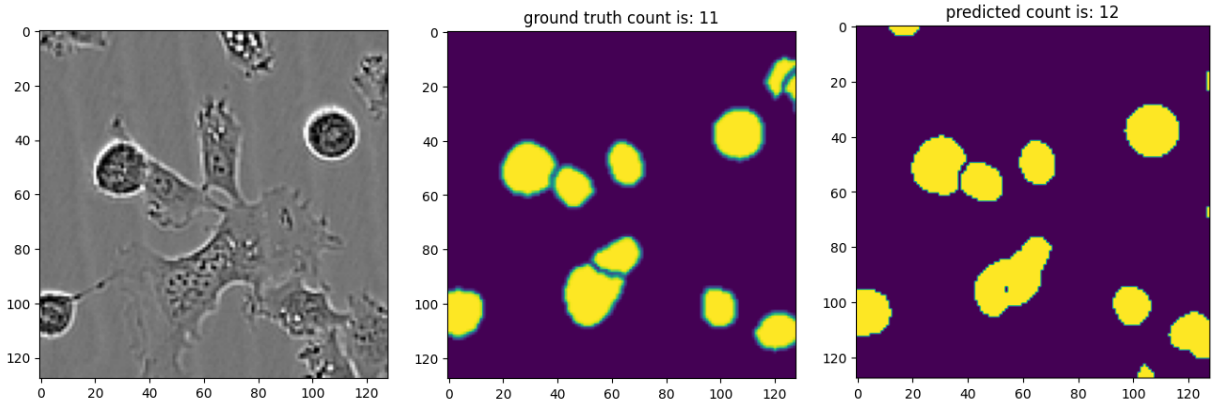



The training dataset’s Average Cross-Entropy Loss Function decreases significantly, while the validation dataset’s Average Cross-Entropy Loss Function starts increasing after a certain point, indicating potential overfitting.

Furtherly, I grabbed a batch of data and displayed a randomly selected image and the predicted label from that batch. The second image is the ground truth and the third image is the hard truth shown below.



Another batch of data shown below, when the cells are adjacent to each other, it might be connected and count as one.




Initially, I faced a challenge in lowering the Kaggle score below 9 and through debugging, I discovered an error in reading the test file as part of the training data. After correcting this error, my Kaggle score improved significantly with the model's performance. I have retrieved a final Mean Absolute Error (MAE) on the validation dataset of 1.4875.

By implementing a U-Net architecture for this project of cell counting in microscopy images yielded promising results. Despite the challenges, the model achieved a MAE of 1.4875 on the validation dataset after debugging and fine-tuning. The convergence plot and segmented images examples illustrated the model's effectiveness. In the future, I would focus more on furthering to reduce the overfitting and exploring data augmentation techniques to enhance it more.

In conclusion, this final project report reflects a methodical approach to tackling the cell counting problem and demonstrates the successful application of deep learning techniques in biomedical image analysis.