CS482/682 Final Project Report (Group 9)

Deep Learning for Non-Invasive Blood Cytometry

By: Shuhao Lai (slai16), Luojie Huang (lhuang48), Yuan Zhou (yzhou143)

1. Introduction

Background Current Complete Blood Count (CBC) is now achieved in an invasive way. Now an Oblique Back-illumination (OBM) capillaroscopy from Dr. Nicholas Durr's Lab makes noninvasive CBC possible. In this project, our goal is to use deep learning to analyze videos depicting cells in the human bloodstream. We want to count the number of unique cells present in the video. Our results can be used to determine the velocity of the blood stream.

Related Work In the past decade, many deep learning models were applied to cytometry to detect, classify, or segment images of cells. CNNs, especially Region CNN (R-CNN), are one of the most prevalent models used in segmentation and classification because they can generate rich feature maps for accurate bounding boxes of cells [1-3]. 2D and 3D U-Net also perform well in cell segmentation with a small dataset from electron microscopy (EM) images [4-5]. Several works focused on studying cell image sequences by using Recurrent Neural Network (RNN) to take advantage of temporal information to detect cells [6-7].

Due to the novelty and difficulty of the tracking cells in a video, we do not have a baseline. We used U-Net to obtain a preliminary result for further studies. The mean dice score is only 0.6148. Therefore, U-Net may not be robust enough for our tracking and counting tasks.

2. Methods

Dataset The raw data is obtained using noninvasive Oblique Back-illumination capillaroscopy from Dr. Nicolas Durr's lab. There are 200 annotated frames from 2 real-time blood flow videos. The annotations specify the id and cell mask for each cell.

Preprocessing In experiments, stabilizing the videos gave us better results. Our video stabilizing algorithm estimates and smooths the optical flow of feature points between frames to reduce camera jitter.

Mask R-CNN + Counting For our first proposed method, Mask R-CNN [8] was used to generate a bounding box and mask for each cell in a frame. We calculated a fixed Region of Interest (ROI) across the targeted vessel based on motion estimation then counted the number of cells passing through it. To track a cell across frames, we first estimate its position in the upcoming frame using its trajectory. We can then calculate the dice score between the predicted cell and cells in the current frame. This tracking prevents double counting and gives a more accurate result. A stabilization strategy is required before the counting step.

Our Mask R-CNN model is trained for 25 epochs using an Adam optimizer and weight decay of 1e-5.

Mask-Track R-CNN Mask-Track R-CNN adds another head to the Mask R-CNN model, which is responsible for learning new representations of identified cells and using them to compute a similarity metric between cells in adjacent frames.

The original Mask-Track R-CNN [9] uses dense layers and a dot product to compute the similarities between frames, but replacing the dense layers with convolutions significantly improved accuracy (~25% increase). Further, the original model remembers many frames from the past, which we discovered can be shortened since only a small

number of past frames are useful for tracking in our tasks. Moreover, the original model is inapplicable to large shift and deformation and bad at solving competitive matching. Therefore, we used an iterative matching method to find an optimal match set for all cells in a certain frame at one time.

CNN + RNN This model uses VGG16 with batch normalization to extract a feature vector from each frame to feed to a RNN with two LSTM cells. Two fully connected layers at the end make the predictions. (Fig. 1). The intuition is for the RNN to keep track of cells across time/frames to prevent double counting.

The model uses color jittering and crops to augment the training data and uses an Adam optimizer. The model uses an adaptive learning rate starting at 0.01 with a patience of 10 epoches and multiplication factor of 0.2. Early stopping with a patience of 15 epochs and L2 regularization with weight decay = 0.00001 were also used. Due to GPU memory issues, the 100 frame video was split into independent smaller videos and passed into the model.

Evaluation All three models were evaluated using percent error between the ground truth and prediction, which were rounded if necessary.

3. Results

For cell detection, Mask R-CNN performs within a margin of error for reasonable cell tracking and counting (Fig. 2). Table.1 shows the counting results for our three proposed methods. CNN + RNN was not able to converge (Fig. 3) and had the lowest performance among the three models. A possible reason for the poor performance is that the features extracted by RNN are not valuable (see Discussion). The best performing model overall is the Mask R-CNN + Counting. Fig. 4 is the output for Mask Track R-CNN. Because Mask-Track R-CNN was built for large video datasets [9], so there is a reasonable possibility for it to outperform the other two models when given more data.

Approaches	Count Number						BFV (cells/s)
	50fs			100fs			
	GT	Pred	Error (%)	GT	Pred	Error (%)	
Mask RCNN (ResNet-101)	**33**	**35**	**6.06**	**59**	**57**	**3.39**	91.2
Mask-track RCNN (ResNet-101)	**33**	**35**	**6.06**	59	64	8.47	102.4
CNN + RNN (VGG)	33	44	33.3	59	85	44.07	136.0

Table. 1 Results of all three models

It is important to note that making predictions for more frames is more difficult because errors accumulate.

4. Discussion

Limitations The prediction accuracy of Mask RCNN is sensitive to the density of cells. Further, the Mask-Track algorithm is sensitive to cell deformation, so cells traveling through a non-linear path will result in poor results. Lastly, the CNN + RNN feature extractor was not fine-tuned due to the lack of data. Since the RNN heavily relies on the feature extractions and VGG was trained on objects reasonably different from our domain, we believe a more appropriate feature extractor will greatly improve performance. Lastly, the lack of data may prevent our models from generalizing well.

Future Improvements Exploring more augmentation approaches for our data and collecting more annotated data will likely improve all three models. Further, using a better performing object detector, such as EfficientDet, will likely improve all three models as well.

Appendix

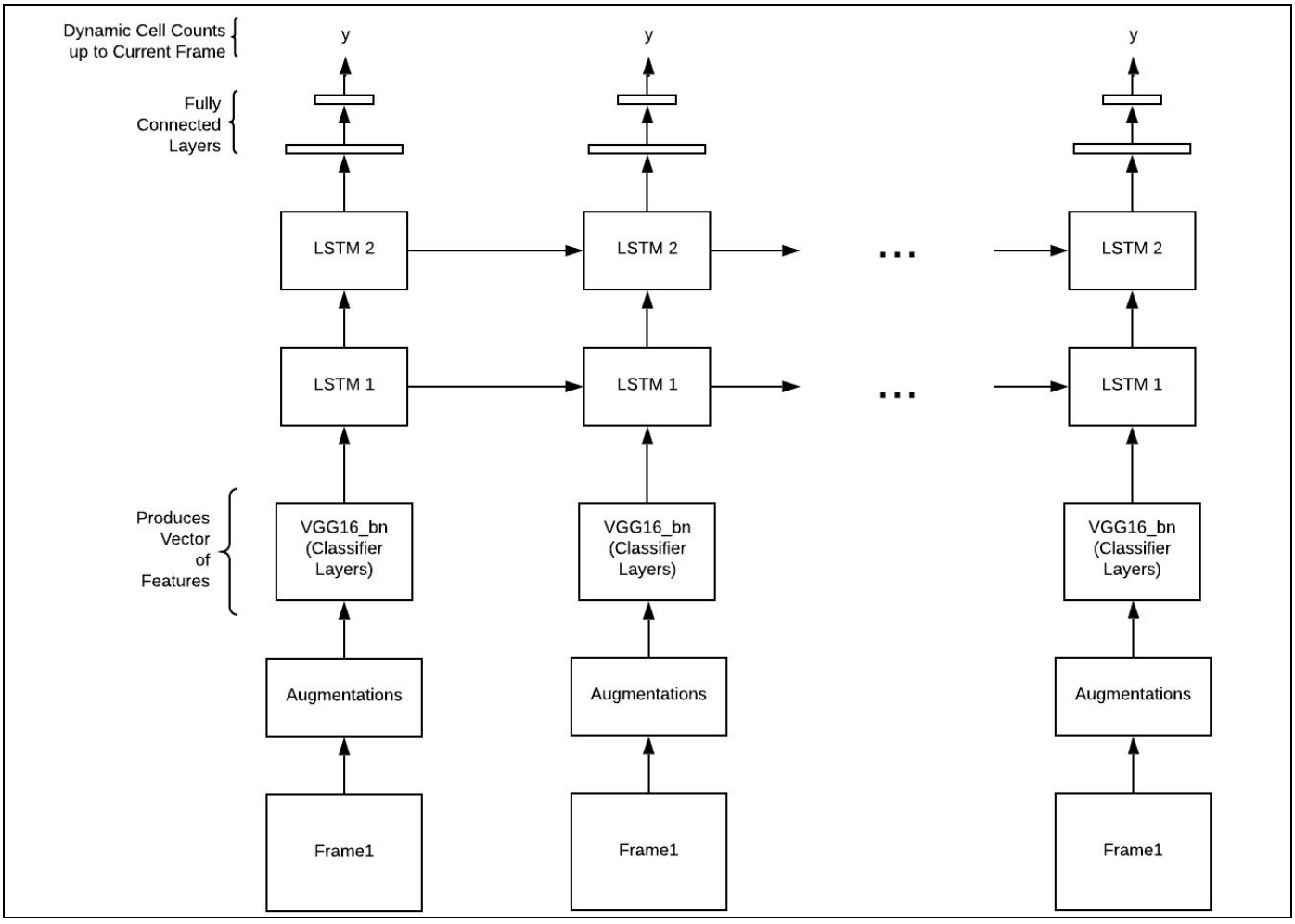

Fig. 1 CNN + RNN Architecture

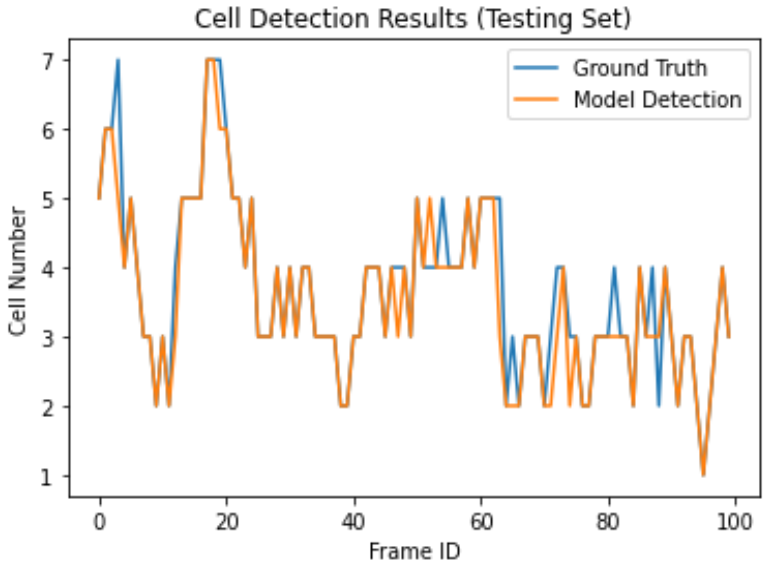

Fig. 2 Cell Detection Result of Testing Data. The result from our model (the orange line) is close to the ground truth (the blue line), which is good enough for further tracking and counting steps.

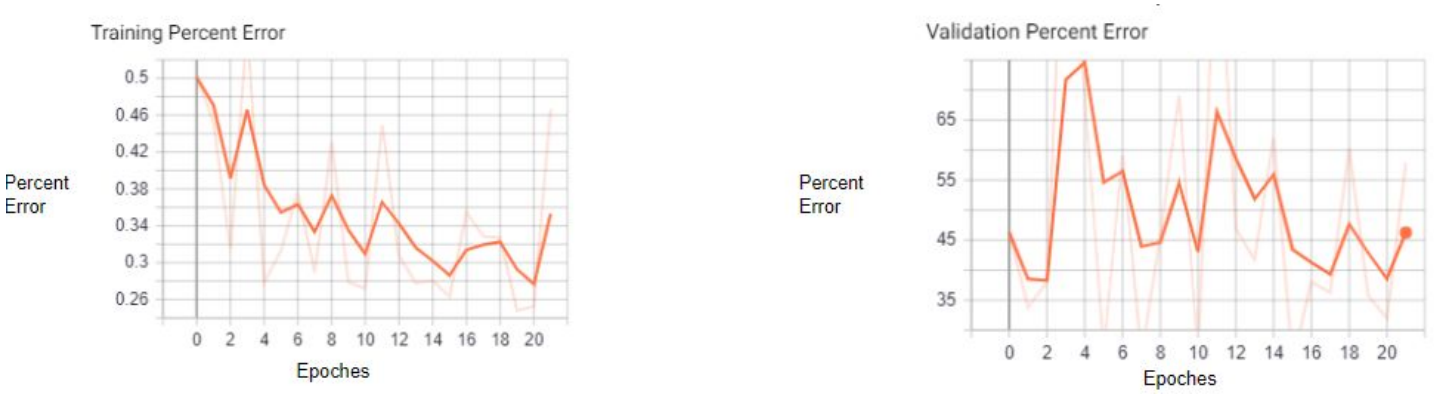

Fig. 3 Plots for CNN + RNN. These plots suggest that the model is memorizing the training data but not learning specific patterns important for this task.

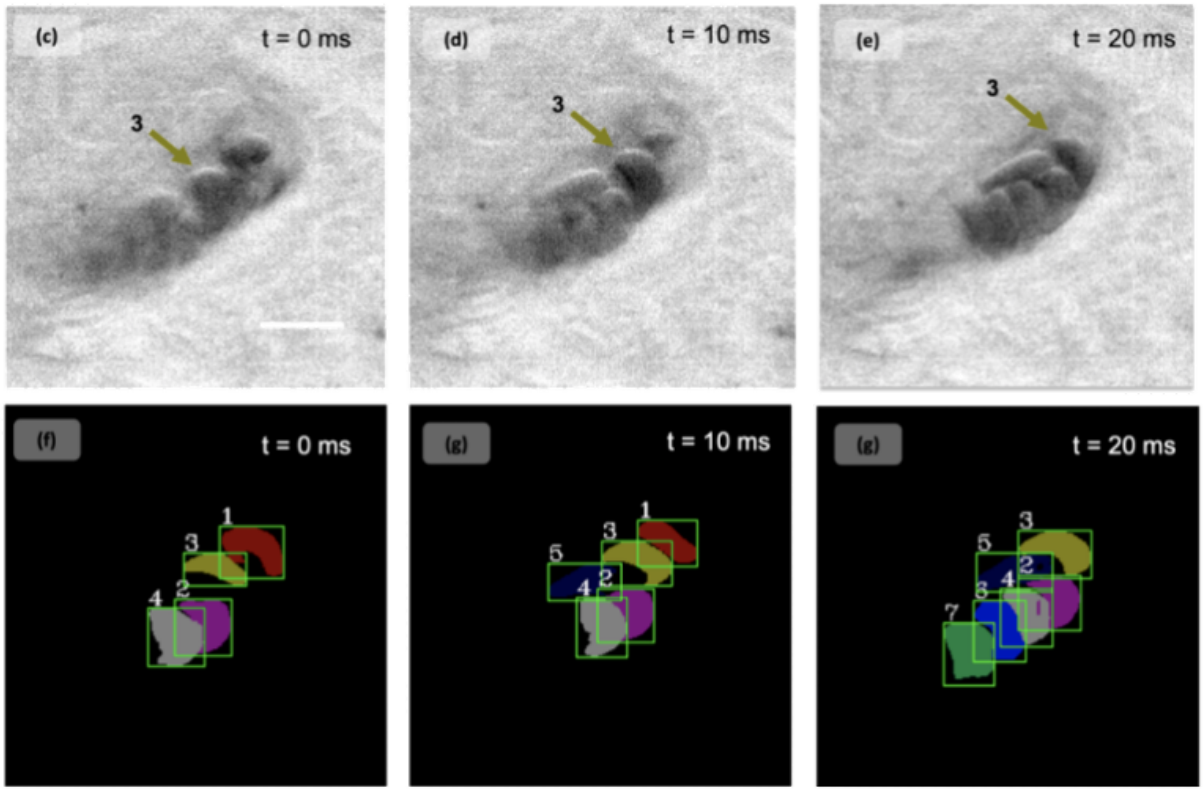

Fig. 4 Example output for Mask-Track R-CNN where the numbers are Ids for a cell.

References

1. Ciresan D, Giusti A, Gambardella LM, Schmidhuber J. Deep neural networks segment neuronal membranes in electron microscopy images. Advances in neural information processing systems, 2012. pp. 2843-2851.
2. Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. I.E. Conference on ComputerVision and Pattern Recognition. 2014.
3. Ren S, He K, Girshick R, Sun J. Faster R-CNN: Toward real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 2017;39(6):1137–1149.
4. Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. Medical Image Computing and Computer-AssistedIntervention (MICCAI). Springer International Publishing, 2015. pp. 234–41.
5. Çiçek Ö, Abdulkadir A, Lienkamp SS, Brox T, Ronneberger O. 3D U-Net: LearningDense Volumetric Segmentation From Sparse Annotation. Lecture Notes in Computer Science. New York: Springer International Publishing, 2016. pp. 424–432.
6. Phan HT, Kumar A, Feng D, Fulham M, Kim J. An unsupervised long short-term memory neural network for event detection in cell videos. 2017.
7. Villa AG, Salazar A, Stefanini I. Counting cells in time-lapse microscopy using deep neural networks. 2018.
8. He, Gkioxari, Piotr, Girshick, & Ross. Mask R-CNN. 2018, January 24.
9. Yang, Linjie, Xu, Ning. Video Instance Segmentation. In ICCV, 2019.