CSE 253 Final Project
Find the Nuclei: Biomedical Image Segmentation via Convolutional Neural Networks

Chih-Hui Ho, Chun-Han Yao, Po-Ya Hsu, Yao-Yuan Yang, Hsin-Yang Chen, Ying-Chuan Liao
Department of Computer Science and Engineering
University of California, San Diego
{chh279, chy235, p8hsu, yay005, hsc061, yil697}@eng.ucsd.edu

Abstract

Identifying the nuclei of cells is a crucial starting point for most medical analyses, which, however, is time-consuming for researchers to label manually. In this project we aim to automate the process of finding and segmenting nuclei in divergent medical images. Our dataset is provided by Kaggle 2018 Data Science Bowl [1], including 670 training images with 29462 segmentation masks and 69 testing images with no ground truth. We propose to combine conventional computer vision methods and convolutional neural networks (CNN) for instance segmentation. As our experimental result, **we achieve mean average precision (mAP) of 0.413, which is a top 12% performance on current Kaggle's leaderboard**. Potentially, a thorough performance comparison of existing methods will help devise a novel framework for nuclei segmentation.

1 Goal

Our goal is to advance medical discovery by automating the identification of nuclei in diverse images. Specifically, it is a task of instance segmentation, where the inputs are single images and the outputs are the masks for each nucleus instance in the cells. An example of input-output pair is shown in Fig.1, where the output masks are stacked into a single image.

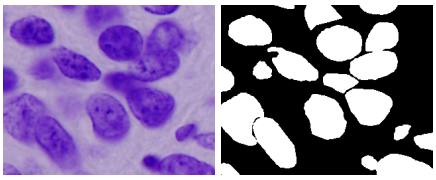

Figure 1: An example of input image and its segmentation mask

2 Motivation

Identifying the nuclei of cells is the starting point for most medical analyses because most of the human body's 30 trillion cells contain a nucleus full of DNA, the genetic code that programs each cell. Detecting nuclei allows researchers to locate each individual cell in a sample, and by measuring how cells react to various treatments, the researchers can understand the underlying biological processes at work.

While it costs considerable time and effort for human researchers to label the cells, the task is rather simple for computers inherently. By automating the process of identifying nuclei, we will allow for more efficient drug testing as well as other medical analyses.

With a compact problem formulation, we believe that CNNs are specifically suitable for producing high-level feature representation to aid pixel-wise classification. Conventional computer vision methods can further validate the output mask and refine the instance clustering.

3 Dataset

Our dataset consists of 739 cell images from Kaggle 2018 Data Science Bowl [1], including 670 training images with 29462 segmentation masks and 69 testing images with no ground truth. Each ground truth mask corresponds to one nuclei, and does not overlap with other masks. Some examples are plotted in Fig.2 and 3. Since the size of training set is rather small, we rely on heavy data augmentation to train the CNNs.

Observing from the dataset, the training and testing samples can be categorized into three types. Different types of images have distinct color distribution and nucleus appearance. Generally, the background is plain and the nuclei have distinctively different pixel intensity. However, the size and shape of nuclei varies drastically between image types, and they may overlap with each other when densely populated.

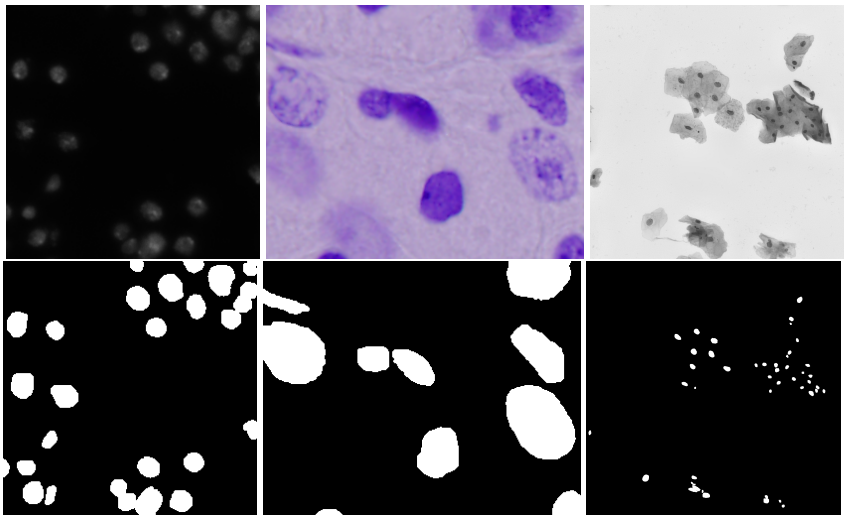

Figure 2: Three types of training samples and their ground truth segmentation masks.

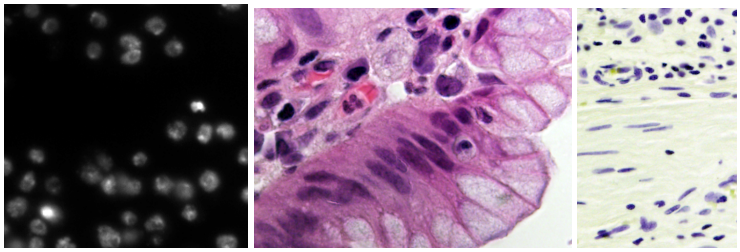

Figure 3: Three types of testing samples

4 Approaches

In this section we briefly introduce the approaches used in our project. First, the training data is increased via data augmentation, and pre-processing is employed to normalize the luminance distribution. Convolutional neural networks or superpixel with SVM then segment the images into binary masks. To further separate the nuclei with adjacent boundary, clustering methods such as K-means and Gaussian mixture model are investigated. Finally, we experiment with model ensemble and post-processing for performance boosting. An overview of the proposed framework is illustrated in Fig.4.

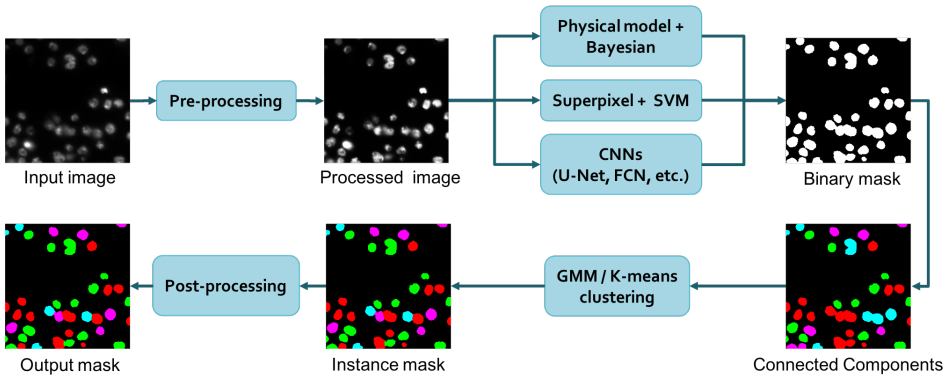

Figure 4: Framework overview.

4.1 Data Augmentation

With a limited amount of training data, we rely heavily on data augmentation to train the neural networks. In addition to typical transformations such as random scaling, cropping, flipping, rotation, and color manipulation, we also perform random deformation since the shape of nuclei is deformable.

4.2 Pre-processing

Noticing that the luminance distribution of the images varies considerably, we employ histogram equalization to normalize the luminance distribution. The effects of input size and representation are also investigated. In particular, we experiment on RGB, HSV, and gray scale images, with size of 128×128, 256×256, and 512×512.

4.3 Convolutional Neural Networks

For segmentation prediction we experiment on different convolutional networks. **U-Net** [2] proposed by Ronneberger *et al.* is designed for biomedical image segmentation specifically, where the precision of mask boundary is critical. The network architecture is shown in Fig.5. To maintain high resolution of the output mask, the feature maps in the down-sampling and up-sampling parts of the network at each scale are connected, and larger penalty is imposed to the boundary between adjacent instances. **Mask-RCNN** [3] and **SDS** [4] are instance segmentation methods that perform pixel-wise segmentation on the regions proposed by detection models. While they generally perform well on various datasets, they are harder to train and require more training data. Semantic segmentation models such as **FCN** [5] are also capable of producing accurate nuclei-background segmentation, from which clustering methods can be applied to further separate neighboring instances. The architecture of FCN is illustrated in Fig.6. We also experienced **DeepMask** [11] and **SharpMask** [12] to perform the same task. They both focus on predictions the two subjects, whether the input image contains an object and the mask of the object if so. SharpMask outperforms DeepMask since it can perform pixel-wise mask prediction while DeepMask have only a low resolution mask as outputs. The structures of the two are shown in Fig.7 and Fig.8.

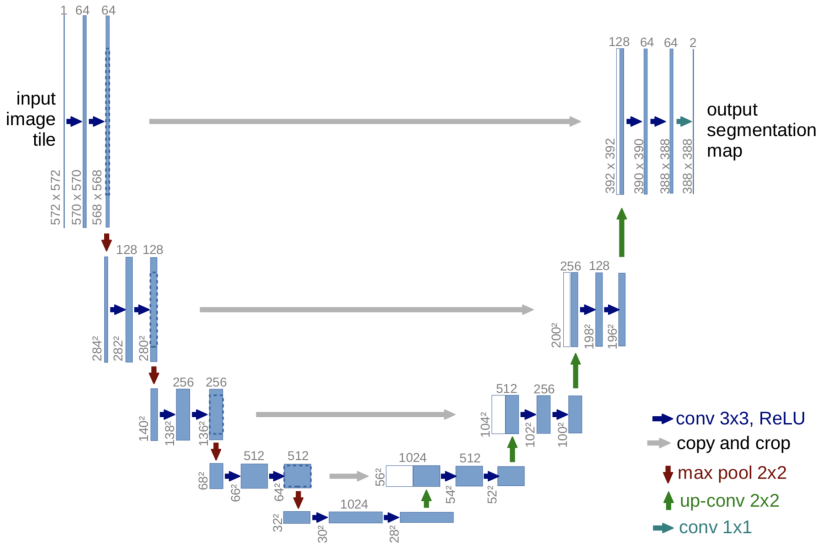

Figure 5: U-Net architecture.

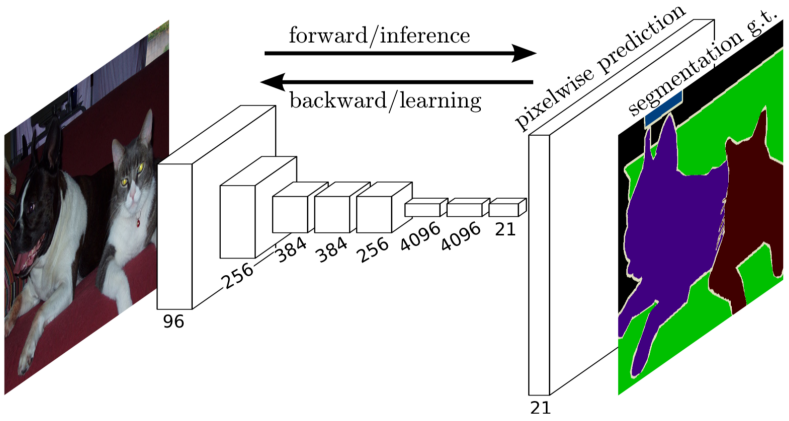

Figure 6: FCN architecture.

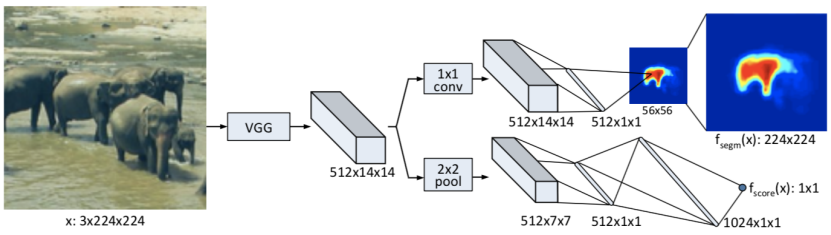

Figure 7: DeepMask architecture

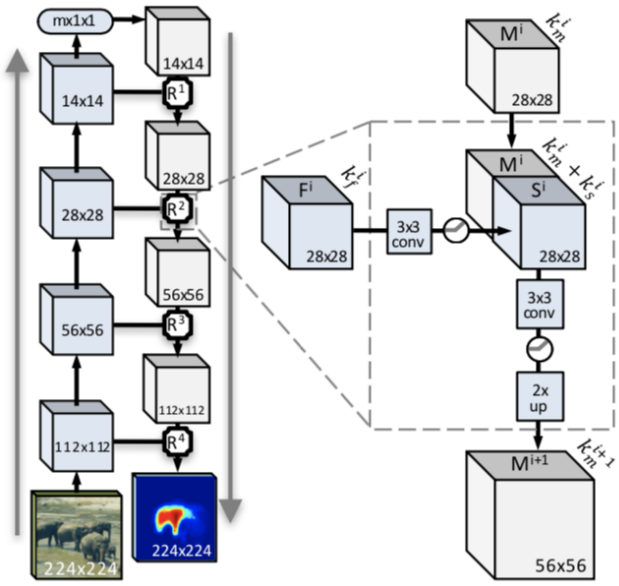

Figure 8: SharpMask architecture

4.4 Conventional Computer Vision Methods

As the baselines for performance comparison, conventional computer vision approaches are also implemented. For instance, with the patches overly segmented by **SLIC superpixel** [6], we classify the them as nuclei or background using **SVM [7] on color histogram**. Another typical physical approach is to combine the lighting features and **Bayesian probability model** [8], where the intensity and gradient of the pixels are utilized to capture each cell in the given image.

4.5 Instance Clustering

Given the binary mask predicted by either CNNs or conventional computer vision approaches, we rely on clustering methods such as **K-means clustering [9]** and **Gaussian mixture model (GMM) [10]** to separate the overlapping or adjacent nucleus instances. Since typical nuclei are in circular or elliptic shapes, the assumption of Gaussian distribution holds. However, we find it hard to determine the number of clusters, since the negative log likelihood decreases as the number of clusters increases. As illustrated in Fig.9, we use the "elbow method" to choose an optimal number of cluster.

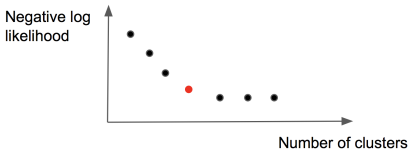

Figure 9: To determine the number of clusters for each connected component in the predicted binary masks, we use the "elbow method" where the red dot indicates the number of cluster that have the largest second-order difference of negative log likelihood.

4.6 Post-processing

To further boost the performance, model ensemble is done by pixel-wise voting, *i.e*, each pixel is classified as nuclei if more than half of the models predict so. Finally, we perform post-processing including smoothing the mask boundary, filling the holes in each predicted nucleus, and removing the outliers with irregular size.

5 Evaluation Metric

This Kaggle competition is evaluated on the mean average precision (mAP) at different intersection over union (IOU) thresholds. The IOU of predicted mask A and ground truth mask B can calculated as IOU(A,B) = $\frac{A \cap B}{A \cup B}$. With IOU threshold t, a prediction is considered correct when the IOU $\geq t$. The metric takes the average precision over different IOU thresholds, which range from 0.5 to 0.95 with a step size of 0.05: (0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95).

At each threshold value t, the precision value is calculated as $\frac{TP(t)}{TP(t)+FP(t)+FN(t)}$, where TP, FN, FP denote true positives, false negatives, and false positives, respectively. These values can be calculated by matching the predicted masks to the ground truth masks. A true positive (TP) is counted when a single predicted instance matches a ground truth instance with an IOU above the threshold. A false positive indicates a predicted instance has no associated ground truth instance. A false negative indicates a ground truth instance has no associated predicted instance. The average precision of a single image is then calculated as the mean of the precision values at each IOU threshold:

$$\frac{1}{T} \sum_{t} \frac{TP(t)}{TP(t)TP(t) + FP(t) + FN(t)} \tag{1}$$

Lastly, the score is averaged over the testing images.

6 Experimental Results

In our experiments on the training and testing images, we achieve similar mAP above 0.3 with variations of U-Net and FCN, which is significantly higher than that of conventional computer vision approaches. With instance clustering by Gaussian mixture model, the performance of CNN models can be further improved significantly. Detail mAP comparison is shown in Table 1, and qualitative results are demonstrated in Fig.10, 11, and 12.

From Fig.11 we observe that CNN models produce accurate binary mask but can not distinguish adjacent nuclei from one another. Weighting the penalty on the boundary pixels as suggested in U-Net does not work well, either. To remedy this, GMM is able to group the superpixels shown in Fig.10 or cluster the pixels as shown in Fig.12. The number of clusters is determined carefully by the "elbow method", otherwise it tends to over-segment or under-segment the pixels.

The results of DeepMask/SharpMask network is shown in Fig.14. We can see that it did well on some images while for particular kind of images, especially images with purple cells and bright backgrounds, our model cannot recognize any cell at all. This might due to the insufficient dataset we generated from training images. DeepMask/SharpMask requires a clearly separated mask for different objects which are large and sparsely located, such as those in the COCO dataset. As long as we are unable to process desirable training dataset, there will always be a bottleneck for the DeepMask/SharpMask architecture.

Approaches	mAP
FCN	0.315
U-Net	0.340
U-Net + GMM	**0.392**
U-Net + K-means	0.307
Superpixel + SVM	0.237
Physical model + Bayesian	0.234
DeepMask/SharpMask	0.170
Model ensemble	0.392
Model ensemble + GMM	**0.413**

Table 1: Performance comparison on the testing images in terms of mAP

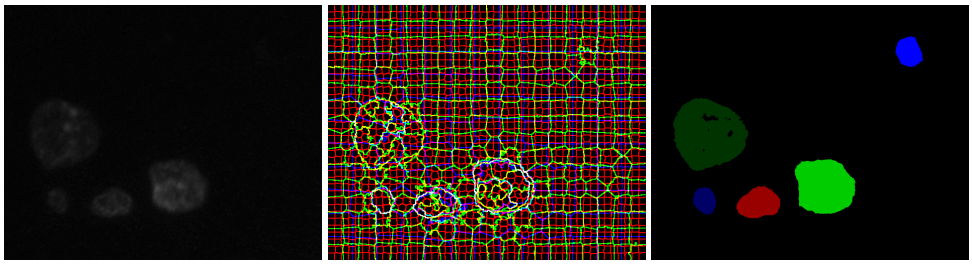

Figure 10: Qualitative result of superpixel segmentation with SVM classification and GMM clustering. The input image, superpixel segmentation at multiple scale, and the predicted mask are shown from left to right.

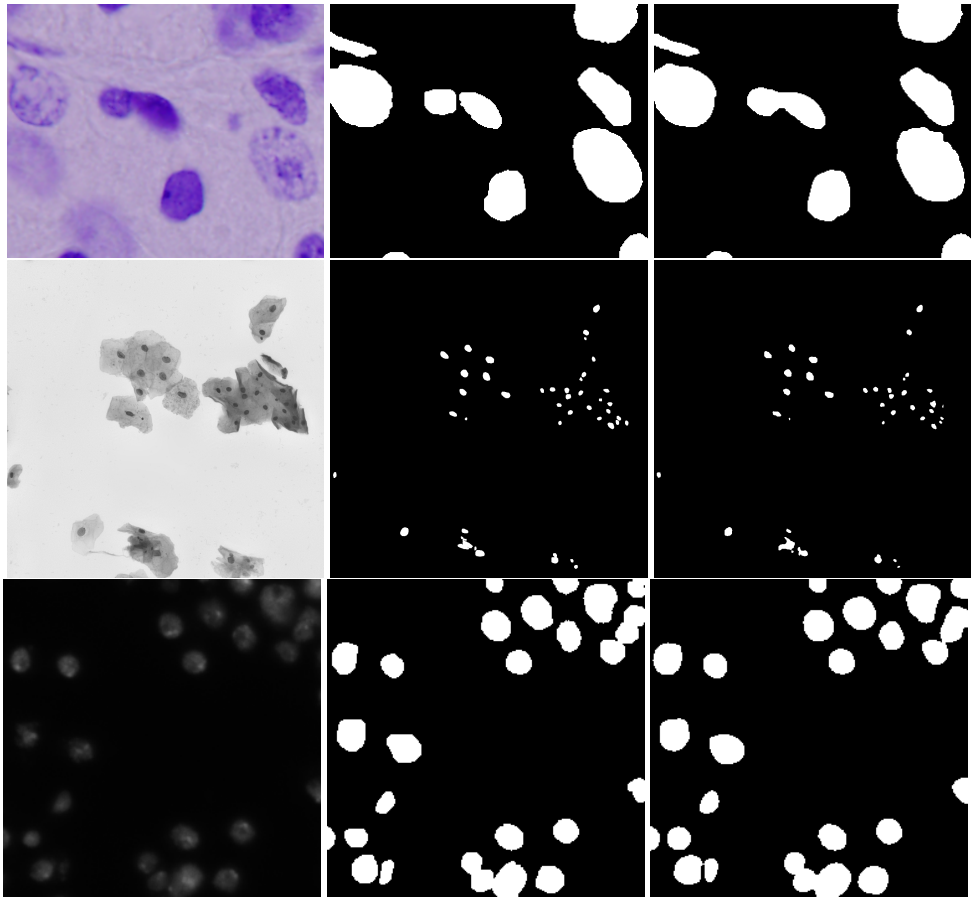

Figure 11: Qualitative results on training data. The input images, ground truth mask, and results of U-Net are shown in the left, middle, and right column, respectively.

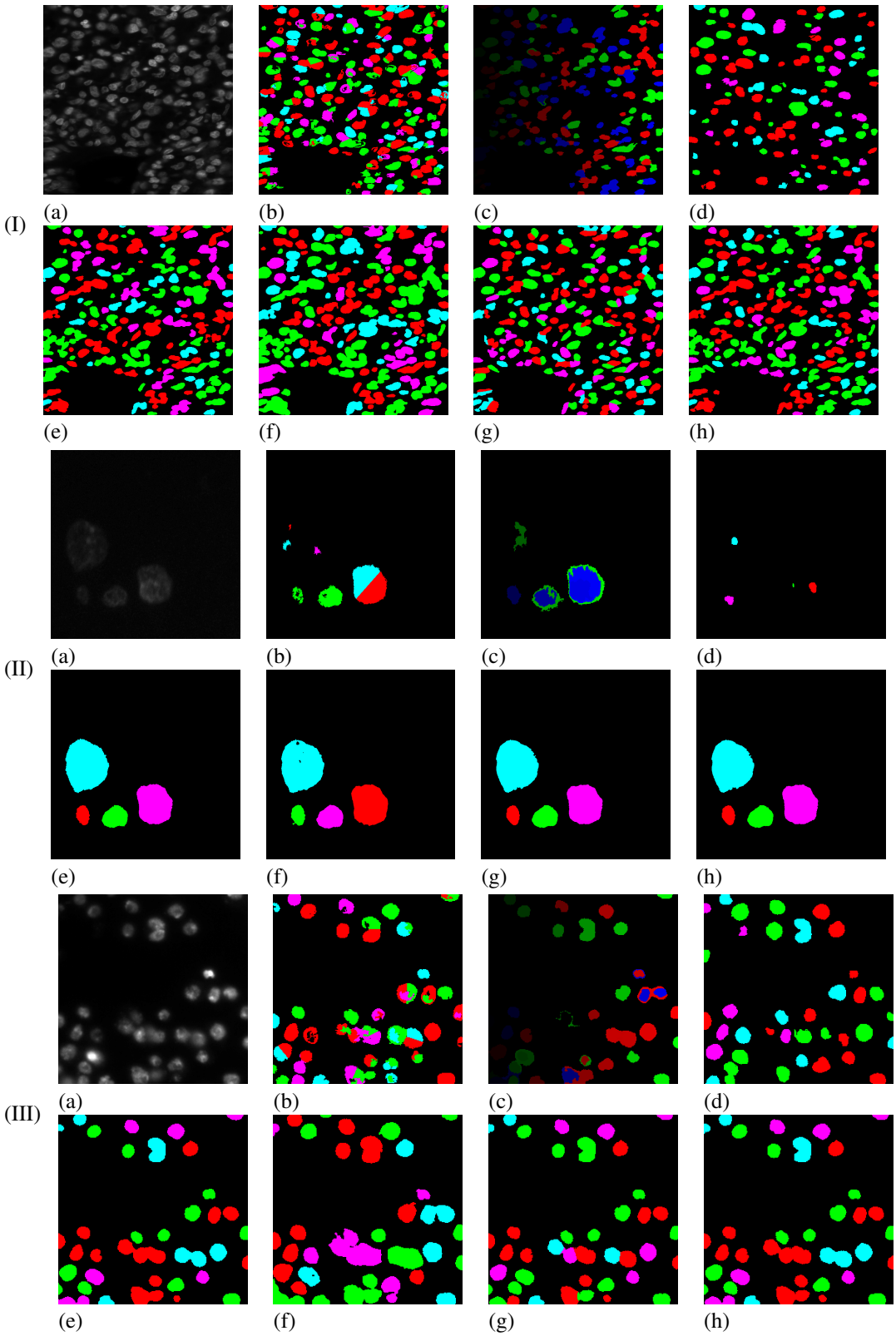

Figure 12: Qualitative comparison on 3 different testing data (I), (II) and (III). (a) Test image (b) Physical model with baysian (c) Super-pixel (d) DeepMask (e) FCN (f) U-net (g) U-net + GMM (h) Ensemble

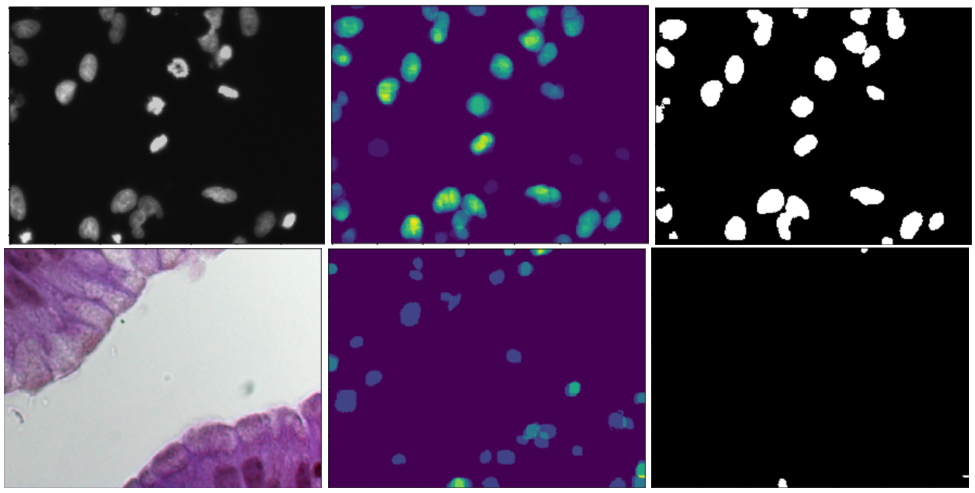

Figure 13: Different results for two type of images. The original images, heatmaps generated by DeepMask/SharpMask, and the final mask results are plotted in the left, middle, and right column, respectively.

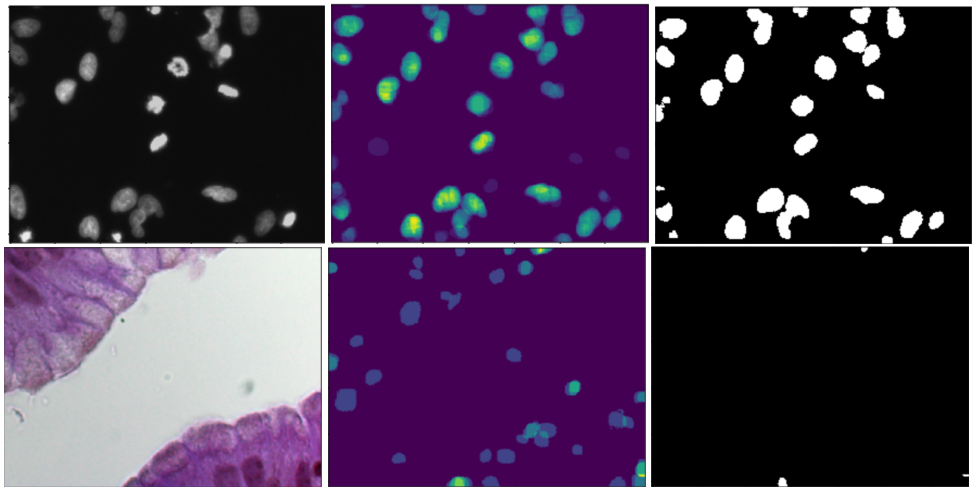

Figure 14: Different results for two type of images. The original images, heatmaps generated by DeepMask/SharpMask, and the final mask results are plotted in the left, middle, and right column, respectively.

7 Conclusion

CNNs are able to learn powerful semantic representation of diverse images, and carefully-designed architectures such as U-Net and FCN can produce fine-grain segmentation masks at full resolution. However, to distinguish overlapping or neighboring instances, weighting the penalty on the boundary pixels does not work well in our experiments. We resort to Gaussian mixture model for clustering the predicted pixels, which is shown to be surprisingly effective. Lastly, since the proposed models are built independently by different group members, model ensemble significantly boosts the performance. As our result, **we achieve mean average precision (mAP) of 0.413, which is a top 12% performance on current Kaggle's leaderboard**.

For the future work, detection-based segmentation models such as Mask-RCNN and SDS could be investigated. It would also be interesting if a novel CNN architecture can capture specific shapes of object like Gaussian mixture model does. In this way, the pixel-wise classification and instance clustering can be jointly optimized by end-to-end training.

8 Contribution

8.1 Chih-Hui Ho

- Implement Unet architecture and incorporate some concepts borrowed from SegNet, FCN, GoogleNet with keras.
- Experiment with different activation functions (relu, elu, selu) and different architecture, such as replacing pooling layer with convolution layers

8.2 Chun-Han Yao

- Implement superpixel segmentation and SVM classification on color histogram
- Implement post-processing including Gaussian mixture model and K-means clustering.
- Perform model ensemble.
- Draft report.

8.3 Po-Ya Hsu

- Implement distribution detection / model selection at pre-processing step.
- Implement physical features extraction method.
- Implement Bayesian / probabilistic sampling method.
- Implement denoise at post-processing step.

8.4 Yao-Yuan Yang

- Implement FCN using RBG and HSV as feature.
- Test out clustering algorithms for post-processing.
  prediction.

8.5 Hsin-Yang Chen

- Implement DeepMask and SharpMask for both Pytorch and Keras.
- Generate dataset for DeepMask/SharpMask training.
- Improve DeepMask/SharpMask structure by adding additional layers.

8.6 Ying-Chuan Liao

- Experiment of different training and inferencing parameters for DeepMask/SharpMask.
- Generate final DeepMask/SharpMask results.

9 References

[1] https://www.kaggle.com/c/data-science-bowl-2018

[2] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015.

[3] He, Kaiming, et al. "Mask r-cnn." Computer Vision (ICCV), 2017 IEEE International Conference on. IEEE, 2017.

[4] Hariharan, Bharath, et al. "Simultaneous detection and segmentation." European Conference on Computer Vision. Springer, Cham, 2014.

[5] Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[6] Achanta, Radhakrishna, et al. Slic superpixels. No. EPFL-REPORT-149300. 2010.

[7] Suykens, Johan AK, and Joos Vandewalle. "Least squares support vector machine classifiers." Neural processing letters 9.3 (1999): 293-300.

[8] Geman, Stuart, and Donald Geman. "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images." Readings in Computer Vision. 1987. 564-584.

[9] Hartigan, John A., and Manchek A. Wong. "Algorithm AS 136: A k-means clustering algorithm." Journal of the Royal Statistical Society. Series C (Applied Statistics) 28.1 (1979): 100-108.

[10] Zivkovic, Zoran. "Improved adaptive Gaussian mixture model for background subtraction." Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on. Vol. 2. IEEE, 2004.

[11] Pinheiro, Collobert, and Dollar. "Learning to Segment Object Candidates." Computer Science - Computer Vision and Pattern Recognition. 2015

[12] Pinheiro, Lin, Collobert, and Dollàr. "Learning to Refine Object Segments." 2016.