ALL YOU NEED IS CELL ATTENTION: A CELL ANNOTATION TOOL FOR SINGLE-CELL MORPHOLOGY DATA

Qiang Li
RWTH Aachen
qiang.li@rwth-aachen.de

Lily Xu
Vertex Pharmaceuticals
lily_xu@vrtx.com

Otesteanu Corin
ETH Zurich
corino@ethz.ch

ABSTRACT

The purpose of this paper [1] is to invent a unifying approach capable of imaging single-cell morphology of thousands of peripheral blood cells and data-driven learning of characteristic morphology indicative of the presence of the disease. We introduce a lightweight novel family of deep hierarchical network architectures, called AttentionNet. Currently, most methods take manual strategies to annotate cell types for single-cell image processing. Such processes are labor-intensive and heavily rely on user expertise, which may lead to inconsistent results. AttentionNet aims to combine lighter-weight layers, K Means++ techniques in preprocessing, and GBCIOU multi-object segmentation to achieve a nearly-semantic segmentation for cells with lower computational cost and complexity. Its goal is to eliminate artifacts on the sampled cell images due to different experimental conditions, such as lighting conditions, various empirical objects and noise deviations, ensuring more advanced classification.

1 RELATED WORK

Early diagnosis of cancer is a crucial determinant of patient outcome. However, current existing state-of-the-art approaches on cancer diagnosis are only of limited use in deriving a morphological signature in a diagnostic trial, since they often require a cell type annotation for every single-cell image. Labeling for large dataset in actual cancer detection is very time-consuming and resource-intensive. Unsupervised learning or weakly supervised learning methods are often hard to be applied on clinical medical cancer detection because of insufficient accuracy. However, recent developments in neural network architecture design and training strategies have enabled researchers to solve previously intractable learning tasks.

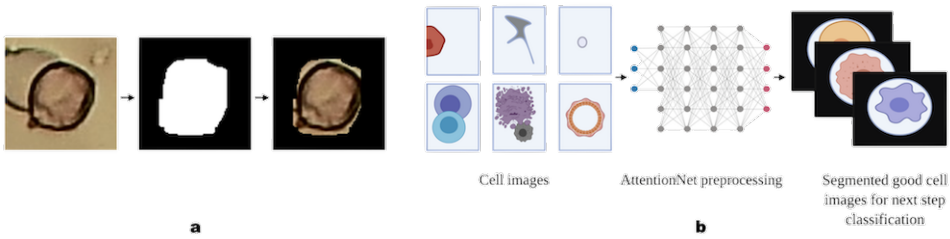

Figure 1: **Manual annotation and AttentionNet labeling.** (a) Typical cell annotation by aerobiology experts, cell images from the largest dataset of microscope pollen grain (Battiato et al., 2021), ICPR 2020 Pollen Grain Classification Challenge. More than 13,000 objects have been detected and hardly labeled by aerobiology experts. (b) Our method for cell artifact elimination and annotation.

Deep learning-based approaches have become very successful in addressing a wide range of biomedical image analysis tasks such as detection of skin cancers from photographic images (Esteva et al., 2017), detection of pneumonia on chest X-rays (Rajpurkar et al., 2017), detection of breast cancer metastases in histopathology images and many others (Angermueller et al., 2016). Another widely

[1] Work performed at ETH Zurich support by IDEA League Research Grant.

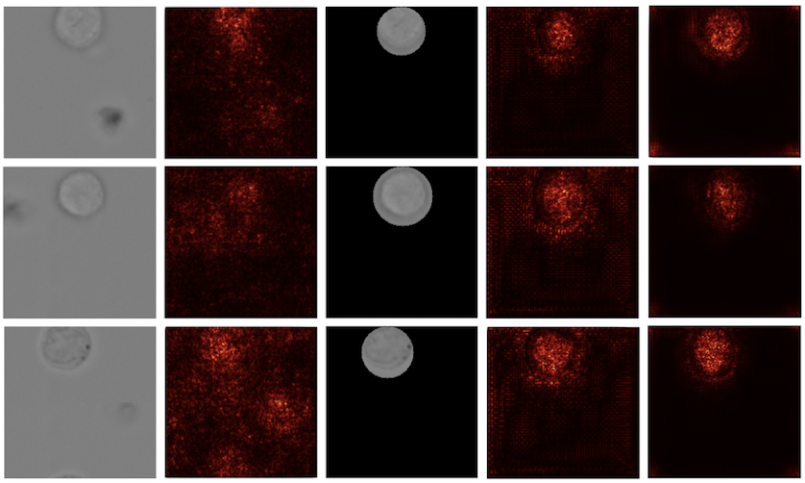

Figure 2: **The illustration of the necessity of Attentionnet segmentation for Sezary Syndrome Dataset.** First column: original cell image. Second column: saliency map of ResNet18 (He et al., 2016) on the original image. Third column: original cell image after taken the AttentionNet segmentation. Fourth and fifth columns: saliency map of ResNet18 and VGGNet (Simonyan & Zisserman, 2015) after AttentionNet segmentation. Leading SOTA models are more focused on non-cell features and the debris feature. After segmentation and image pre-processing, the noise information was removed and only keep the cell's morphological characteristics. It decreases the computational cost and improves reliability and accuracy.

used approach in biomedical image segmentation is U-Net (Ronneberger et al., 2015). They proposed a net and training strategy that relies on the strong use of data augmentation in order to utilize the less available annotated samples more efficiently (Ronneberger et al., 2015).

Furthermore, single-cell analysis might require thousands or even millions of cells analyzed by researchers and clinicians. For example, we received nearly millions of sezary sample images, as in Figure 2. However, characterizing thousands and millions of cell types and cellular states from a complex cell noisy data set is a considerable challenge. These leads to numerous labor-intensive work and expert supervision. SingleR (Aran et al., 2019) infers the cell type for each of the single cells using a novel hierarchical clustering method based on similarity. Similarly, scMatch citepHou annotates single cells by identifying their closest match in gene expression profiles of a large reference dataset. However, such approaches require ideally under the same experimental design using the same platform, which is often not available (Cao et al., 2020).

2 METHOD

For biomedical image processing, different experimental conditions, such as lighting conditions and various empirical objects, noise deviations are likely to appear on the sampled cell images (Guenova et al., 2015). That noise and variability in the background would be confounding variables. When applying AttentionNet, we explicitly learn features focusing on the morphology structure of the cell. We mainly design the unifying approach for sezary cell diagnosis, an aggressive cutaneous T cell lymphoma characterized by tumor T cells with abnormal nucleus morphology in the peripheral blood (Guenova et al., 2015). Hence, we assume that after AttentionNet segmentation, convolutions are more likely to learn feature representations, particularly for cell objects, as only the cell objects are preserved.

2.1 ATTENTIONNET

AttentionNet discards the Darknet (Redmon & Farhadi, 2018) part of the original YOLO (Redmon et al., 2016), i.e., a multi-convolutional stacked layer, and relies on only two YOLO output layers. The original YOLOv3 (Redmon & Farhadi, 2018) used the Darknet front-end feature extraction module, but the detection performance on the Sezary Syndrom dataset is unsatisfied. On the con-

trary, AttentionNet with only 13×13, 26×26 YOLO scale output tensors adopts multi-scale fusion, and K means++ clustering 1 techniques, outperforms TF-Yolo (He et al., 2019) and YOLOv3 (Redmon & Farhadi, 2018), which occupied 13 × 13, 26 × 26, 52 × 52 YOLO scale output tensor. In order to train a suitable segment, it is recommended to choose corresponding scale tensors that refer to different data sets. Another novelty of our method is that we performed K-means++ Clustering in pre-processing. Instead of using the prior nine boxes given by YOLOv3 (Redmon & Farhadi, 2018) trained on the COCO dataset, for our customer dataset, it is more important to give the network the prior knowledge of ground truth box. Utilizing a small subset of the manually annotated cell, we can improve GIOU (ground truth box position, prediction box location) score. It will further achieve higher accuracy.

Then, the GBCIOU and Circle segmentation algorithm (See. Alg 3) will essentially be applied to convert the bounding box to nearly semantic cell prediction while guaranteeing user-defined character requirements of the cell structure. Using the cvfillPoly function we can easily classify the cell image into unaffected polygonal areas and achieve high-speed segmentation once we obtain the output of bounding box detection, namely (x_1, y_1, x_2, y_2). However, it has undeniable shortcomings that the shape of segmentation does not perfectly approximate the ground truth of the cell. To overcome this problem, we proposed an HSV space mask threshold method. It quietly converted to HSV space and use a threshold to eliminate the outside part of the central circle of box detection. For the common challenge of the YOLO original version (Redmon et al., 2016): when multiple objects are standing in the same area or overlapping on the central point, it will become problematic to draw the correct prediction and always leads to wrong labeling. If multiple cells occur and one cell has a more competitive confidence score than another during circle detection, that will lead to either non-labeled or partly labeled problems (See Figure 4).

We proposed the GBCIOU (See. Alg 3) to ideally find a general intersected box center when multiple box predictions occur in the same image, in addition to the GIOU (See. Alg 2) computes the deviation between ground truth and the prediction. The attributes of self-detection and labeling, of nearly real-time resolve, and general resource requirements mainly characterize the AttentionNet.

3 Experiments

To compare our methods on the Sezary Syndrom dataset, we adopted most of those evaluation schemes from the original YOLOv3 (Redmon & Farhadi, 2018), such as Confusion Matrix Precision and Recall, F1 Score, mAP, the IOU, and Classification Loss (See Figure: 3). Moreover, combining Area Under the PR curve leads to conclusive quality metrics paying attention to various aspects (See Figure: 7). We further acknowledge the resource limitation and time cost for a software application in real scenarios usages. We then analyzed the time cost in different GPU availability and computation FLOPs (See Figure: 5).

3.1 AttentionNet on Sezary Sydrome Dataset

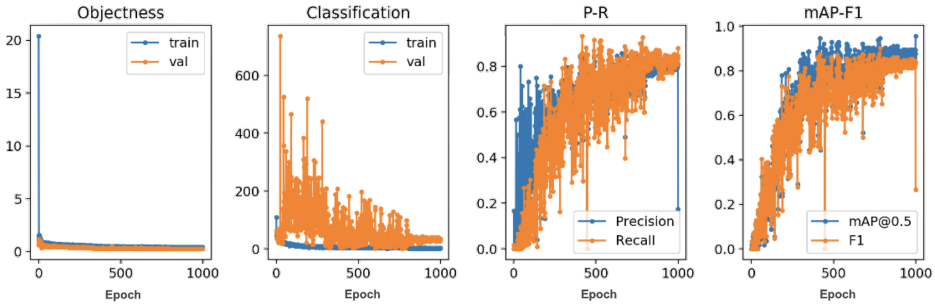

Figure 3: **The train performance of AttentionNet based on manually labeled 1500 cell images**. Here, we try to calculate the recall, precision, objectiveness loss, and classification loss in the binary classification (sezary cell/noise) scenario after each epoch. The P-value is nearly 85%, and the mAP@0.5 reaches almost 88%.

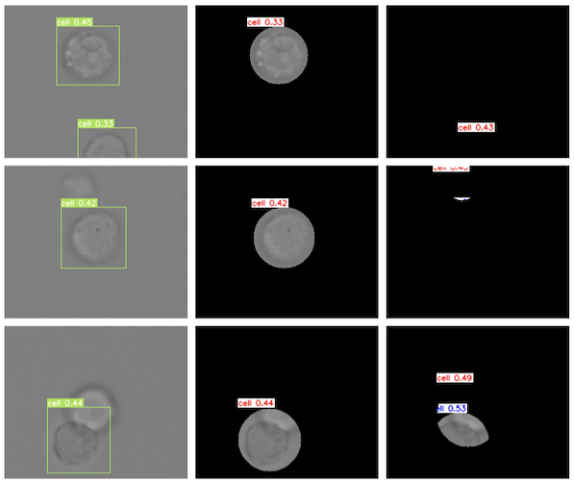

Figure 4: **Experiments on GBCIOU for multiple predictions occur in same cell image.** The first left column stands for original cell bounding box detection by YOLO based network such as (Redmon & Farhadi, 2018) (Redmon et al., 2016), especially multiple cells overlapping or present in the same frame. The second column represents with the help of GBCIOU circle segmentation comparing to without GBCIOU in the third column. With the help of GBCIOU, it obtains the best segmentation performance on cells.

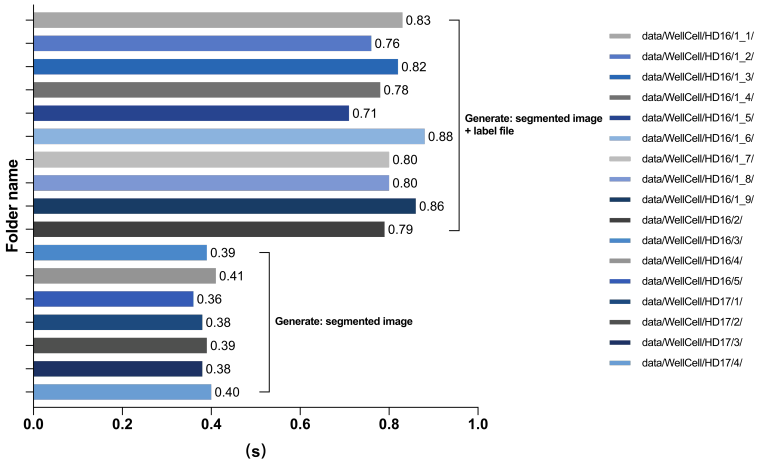

Figure 5: **Time cost of AttentionNet cell segmentation per image on Google Colab**. Overall the average time cost for the AttentionNet segmentation on validation set is 0.80 seconds per image (generated both segmented image and annotation file of cell location). In comparison, the manual method costs an experienced lab assistant around one week to annotate 1500 images. We performed this experiment on Google Colab Tesla V100 16GB.

4 Conclusion

Currently, for cell sample data, manual annotation is unrealistic and consumes more energy and resources. There are often even more noise spots on cell images due to the clutter of data. Our YOLO-based cell annotation and segmentation tool can quickly annotate images while eliminating the noise, thus improving cell classification reliability. The experiments conducted on benchmarks illustrate that an AttentionNet method is a plug-and-play tool for SOTA module for data pre-processing tasks with remarkable speed as shown in Table: 2 and Figure: 5. However, when facing even more complicated scenarios such as multiples cell morphology data or irregular shape cell data is still quite problematic for us. It is open question for us to move forward. Here, we also provide our software, which has won second place in the Deecamp2020 Medical Track Competition. Software code.

REFERENCES

Christof Angermueller, Tanel Pärnamaa, Leopold Parts, and Oliver Stegle. Deep learning for computational biology:. *Molecular Systems Biology*, 12:878, 07 2016. doi: 10.15252/msb.20156651.

Dvir Aran, Agnieszka Looney, Leqian Liu, Esther Wu, Valerie Fong, Austin Hsu, Suzanna Chak, Ram Naikawadi, Paul Wolters, Adam Abate, Atul Butte, and Mallar Bhattacharya. Reference-based analysis of lung single-cell sequencing reveals a transitional profibrotic macrophage. *Nature Immunology*, 20, 02 2019. doi: 10.1038/s41590-018-0276-y.

Sebastiano Battiato, Francesco Guarnera, Alessandro Ortis, Francesca Trenta, Lorenzo Ascari, Consolata Siniscalco, Tommaso De Gregorio, and Eloy Suárez. Pollen grain classification challenge 2020. Springer, Cham, 2021. accepted.

Yinghao Cao, Xiaoyue Wang, and Gongxin Peng. Scsa: A cell type annotation tool for single-cell rna-seq data. *Frontiers in Genetics*, 11:490, 05 2020. doi: 10.3389/fgene.2020.00490.

Andre Esteva, Brett Kuprel, Roberto Novoa, Justin Ko, Susan Swetter, Helen Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542: 115–118, 01 2017. doi: 10.1038/nature21056.

Emmanuella Guenova, Desislava Ignatova, Yun-Tsan Chang, Emmanuel Contassot, Tarun Mehra, Ieva Saulite, Alexander Navarini, Reinhard Dummer, Dmitry Kazakov, Lars French, Wolfram Hoetzenecker, and Antonio Cozzio. Expression of cd164 on malignant t cells in sézary syndrome. *Acta dermato-venereologica*, 96:464–467, 11 2015. doi: 10.2340/00015555-2264.

He, Chang-Wei Huang, Liqing Wei, Lingling Li, and Guo Anfu. Tf-yolo: An improved incremental network for real-time object detection. *Applied Sciences*, 9:3225, 08 2019. doi: 10.3390/app9163225.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. pp. 770–778, 06 2016. doi: 10.1109/CVPR.2016.90.

Yunchen Pu, Zhe Gan, Ricardo Henao, Xin Yuan, Chunyuan Li, Andrew Stevens, and Lawrence Carin. Variational autoencoder for deep learning of images, labels and captions. *Advances in neural information processing systems*, pp. 2352–2360, 09 2016.

Pranav Rajpurkar, Jeremy Irvin, Kaylie Zhu, Brandon Yang, Hershel Mehta, Tony Duan, Daisy Ding, Aarti Bagul, Curtis Langlotz, Katie Shpanskaya, Matthew Lungren, and Andrew Ng. Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning. 11 2017.

Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. 04 2018.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. pp. 779–788, 06 2016. doi: 10.1109/CVPR.2016.91.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi (eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pp. 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. URL `http://arxiv.org/abs/1409.1556`.

A APPENDIX

Algorithm 1 K-means++ Clustering in ground truth boxes.

Require: manually labelled a series of ground truth B^i bounding boxes coordinates $B^i = \left(x_1^i, y_1^i, x_2^i, y_2^i\right), i \in n$. And clusters number K.

Ensure: optimal prediction box size (w, h)

1: For each box B^i, calculated the width and height by:
$w^i = (x_2^i - x_1^i), h^i = (y_2^i - y_1^i)$.
2: Choose an initial center t_1 uniformly at random from the dataset $X = (w^i, h^i), i \in n$.
3: **while** $d\,(x)$ shortest distance and K cluster not reached **do**
4: Choose the next center t_i , selecting $t_i = x' \in X$ with probability $\frac{d(x')^2}{\sum_{x \in X} d(x)^2}$ where $d\,(x)$ is the distance from a data point x to the closest cluster center.
5: For $i \in \{1, 2, ...K\}$, set the cluster T_i to be the set of points in X that are closer to t_i than they are to t_j for all $i \neq j$.
return $T_i = (w^i, h^i), i \in K$.

Algorithm 2 Generalized Intersection over Union(GIoU) as Bounding Box loss.

Require: Predicted B^p and ground truth B^g bounding box coordinates
$B^p = \left(x_1^p, y_1^p, x_2^p, y_2^p\right), B^g = \left(x_1^g, y_1^g, x_2^g, y_2^g\right)$.

Ensure: GIoU loss L_{GIoU}

1: For the predicted box B^p, ensuring $x_2^p > x_1^p$ and $y_2^p > y_1^p$:
$\hat{x}_1^p = min(x_1^p, x_2^p)$, $\hat{y}_1^p = min(y_1^p, y_2^p)$, $\hat{x}_2^p = max(x_1^p, x_2^p)$, $\hat{y}_2^p = max(y_1^p, y_2^p)$.
2: Calculating the area of B^g: $A^g = (x_2^g - x_1^g) \times (y_2^g - y_1^g)$
3: Calculating the area of B^p: $A^p = (\hat{x}_2^p - \hat{x}_1^g) \times (\hat{y}_2^p - \hat{y}_1^p)$
4: Calculating the intersection area I between $B^g$and B^p:
$x_1^I = max(\hat{x}_1^p, x_1^g)$, $y_1^I = max(\hat{y}_1^p, y_1^g)$, $x_2^I = min(\hat{x}_2^p, x_2^g)$, $y_2^I = min(\hat{y}_2^p, y_2^g)$.
5: **if** $x_1^I < x_2^I, y_1^I < y_2^I$ **then**
6: $I = (x_2^I - x_1^I) \times (y_2^I - y_1^I)$
7: **else**
8: $I \leftarrow 0$
9: **end if**
10: Finding the coordinate of the smallest enclosing convex object C:
$C = (min(\hat{x}_1^p, x_1^g), max(\hat{x}_2^p, x_2^g), min(\hat{y}_1^p, y_1^g), max(\hat{y}_2^p, y_2^g))$
11: Calculating area of the smallest enclosing convex object S^C
12: $IoU = I/(A^g + A^p - I)$
13: $GIoU = IoU - \frac{S^C - A^g - A^p + I}{S^C}$
return $L_{GIoU} = 1 - GIoU$

Algorithm 3 GBCIOU for objects overlapping

Require: Two cell object Prediction A and B bounding box coordinates in the same flame $B = \left(x_1^b, y_1^b, x_2^b, y_2^b\right), A = \left(x_1^a, y_1^a, x_2^a, y_2^a\right)$. And confidence Score of each object $S^b > S^a$.

Ensure: Optimal prediction box P.

1: For the box A and B, ensuring: $x_2^b > x_1^b$, $y_2^b > y_1^b$, $x_2^a > x_1^a$, $y_2^a > y_1^a$.
2: Calculate the intersection area:
$I = (min(x_2^a, x_2^b) - max(x_1^a, x_1^b)) \times (min(y_2^a, y_2^b) - max(y_1^a, y_1^b))$
3: Calculating the Union area, contrarily to the original Union we add small $1e-16$ to balance the integral side effect of A and B bounding box coordinates, but it will not effect the $GIoU$:
$Union = (x_2^b - x_1^b) \times (y_2^b - y_1^b) + 1e - 16 + (x_2^a - x_1^a) \times (y_2^a - y_1^a) - I$
4: Finding the coordinate of the smallest enclosing convex object $C_{w,h}$:
$w, h = max(x_2^a, x_2^b) - max(x_1^a, x_1^b), max(y_2^a, y_2^b) - max(y_1^a, y_1^b)$
5: Calculating enclose area E: $E = w \times h + 1e - 16$
6: Calculating $GIoU$: $GIoU = \frac{E - Union}{E}$
7: **if** $GIoU < threshold$ **then**
8: $P \leftarrow (min(x_1^a, x_1^b), min(y_1^a, y_1^b), max(x_2^a, x_2^b), max(y_2^a, y_2^b))$
9: **else**
10: $P \leftarrow (x_1^b, y_1^b, x_2^b, y_2^b)$
11: **end if**
return P

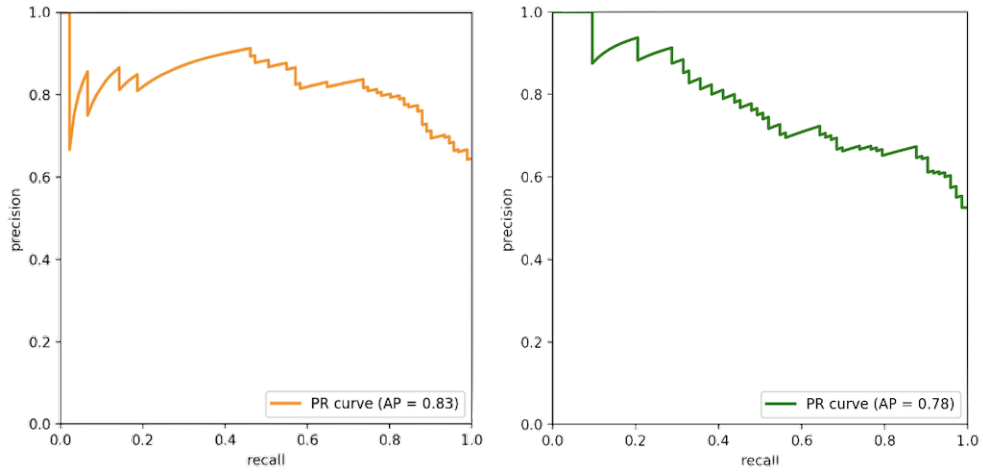

Figure 6: **Precision-Recall curve on val data set of Sezary Syndrom dataset in terms of different scale YOLO output layers.** Both measured on val dataset: 1k training image, 726 test images. The left image illustrates AttentionNet with only 13×13, 26×26 YOLO scale output tensors. The right image illustrates TF-Yolo (He et al., 2019) and YOLOv3 (Redmon & Farhadi, 2018), which occupied 13×13, 26×26, 52×52 YOLO scale output tensor. PR metrics show that 13×13, 26×26 we adopted are more suitable for Sezary Syndromes cell detection.

Table 1: **AttentionNet network structure**. We followed the same symmetrical encoder-decoder architecture (Pu et al., 2016) with additional skip-connections interconnected in the same hierarchy level. At each hierarchy level, we consecutively performed several point-wise convolutions. It is worth noting that the final Yolo output layer should equal to $3 \times (classes + 5)$, as stated in Yolov3.

layer	Type	Filters	Size/Stride	Input	Output
0	Convolutional	16	$3 \times 3/1$	$416 \times 416 \times 3$	$416 \times 416 \times 16$
1	Maxpool		$2 \times 2/2$	$416 \times 416 \times 16$	$208 \times 208 \times 16$
2	Convolutional	32	$3 \times 3/1$	$208 \times 208 \times 16$	$208 \times 208 \times 32$
3	Maxpool		$2 \times 2/2$	$208 \times 208 \times 32$	$104 \times 104 \times 32$
4	Convolutional	64	$3 \times 3/1$	$104 \times 104 \times 32$	$104 \times 104 \times 64$
5	Maxpool		$2 \times 2/2$	$104 \times 104 \times 64$	$52 \times 52 \times 64$
6	Convolutional	128	$3 \times 3/1$	$52 \times 52 \times 64$	$52 \times 52 \times 128$
7	Maxpool		$2 \times 2/2$	$52 \times 52 \times 128$	$26 \times 26 \times 128$
8	Convolutional	256	$3 \times 3/1$	$26 \times 26 \times 128$	$26 \times 26 \times 256$
9	Maxpool		$2 \times 2/2$	$26 \times 26 \times 256$	$13 \times 13 \times 256$
10	Convolutional	512	$3 \times 3/1$	$13 \times 13 \times 256$	$13 \times 13 \times 512$
11	Maxpool		$2 \times 2/1$	$13 \times 13 \times 512$	$13 \times 13 \times 512$
12	Convolutional	1024	$3 \times 3/1$	$13 \times 13 \times 512$	$13 \times 13 \times 1024$
13	Convolutional	256	$1 \times 1/1$	$13 \times 13 \times 1024$	$13 \times 13 \times 256$
14	Convolutional	512	$3 \times 3/1$	$13 \times 13 \times 256$	$13 \times 13 \times 512$
15	Convolutional	**18**	$1 \times 1/1$	$13 \times 13 \times 512$	$13 \times 13 \times 18$
16	YOLO				
17	**Rout13**				
18	Convolutional	128	$1 \times 1/1$	$13 \times 13 \times 256$	$13 \times 13 \times 128$
19	Upsampling		$2 \times 2/2$	$13 \times 13 \times 256$	$26 \times 26 \times 128$
20	**Route 19, 8**				
21	Convolutional	256	$3 \times 3/1$	$26 \times 26 \times 384$	$26 \times 26 \times 256$
22	Convolutional	**18**	$1 \times 1/1$	$26 \times 26 \times 256$	$26 \times 26 \times 18$
23	YOLO				

Table 2: **Comparison in terms of detection/segmentation accuracy with Yolo-based methods in(Redmon & Farhadi, 2018) (He et al., 2019).** Here we selected 850 representative images for training from the sezary syndrome dataset, consist of noise images and typical cell images (manually labeled HD cell image and SS cell image). In the evaluation stage, we utilized 723 images (308 HD cell images, 306 SS cell images, and 109 noises images). We tried to simulate the actual cell data distribution, as noise image less than cell image in the real sezary dataset. It is worth noting that by applying AttentionNet*, we mean adopting a bunch of algorithms mentioned above together, including GBCIOU segmentation, KMean++ Clustering in pro-processing, and 13×13, 26×26 output Yolo layers, compared to original Yolo widely used in only detection or object localization scenario without segmentation. TP means cell detected as cell, FP implicit stands for noise detected as cell, and TN refers to noise image correctly labeled. mAP here refers mean Average Precision.

Method	TP	FP	TN	Image No Label	mAP
Yolov3-tiny (Redmon & Farhadi, 2018)	63.19%	0.91%	87.16%	33.05%	0.55
AttentionNet* Solution	96.25%	11%	80.73%	1.93%	0.88
TF-Yolo (He et al., 2019) with Kmean++ Clustering	91.20%	9.17%	66.05%	11.20%	0.73

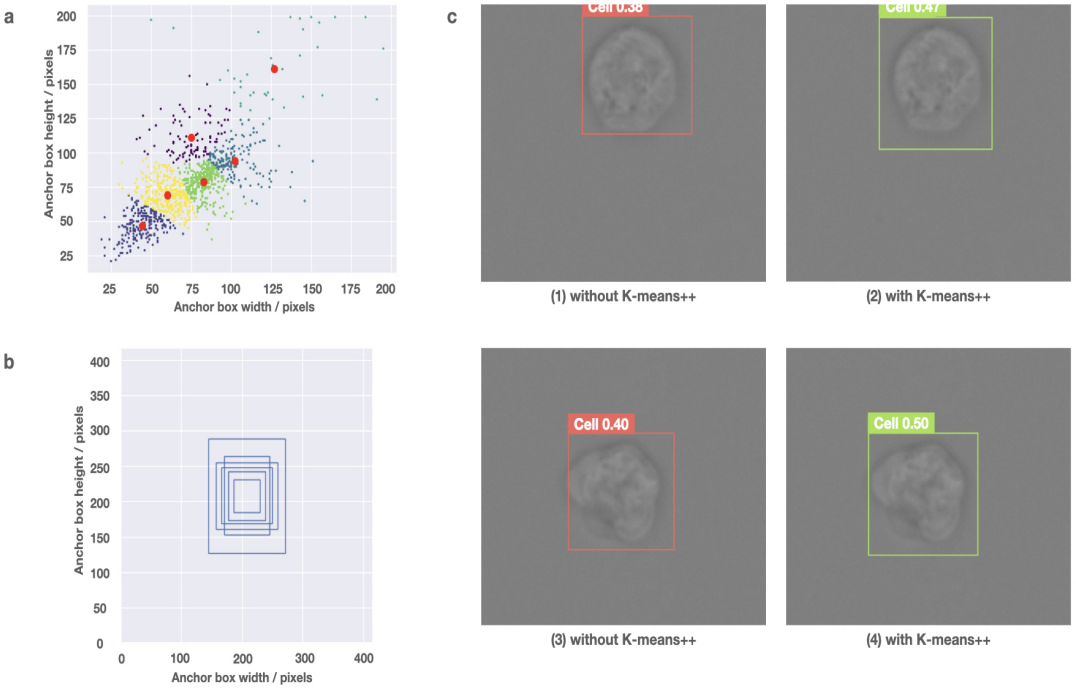

Figure 7: **K-means++ in clustering posterior boxes.** K-means++ clustering in ground-truth anchor boxes of real cell could provide quantitative guidance for 6 types of fitting anchor box of YOLO output layer. As real cell sizes various in left (a), with help of K-means++ clustering in (b) , the final fitting anchor box could better fit cell and further improve the cell confidence score, as shown in (c2), (c4) with K-means++, (c1) and (c3) without K-means++ clustering only use anchor box size provided by original YOLOv3 pre-trained on COCO dataset.