DeGPR: Deep Guided Posterior Regularization for Multi-Class Cell Detection and Counting

Aayush Kumar Tyagi[1*], Chirag Mohapatra[1*], Prasenjit Das[3], Govind Makharia[3], Lalita Mehta[3], Prathosh AP[2], Mausam[1]
[1]IIT Delhi [2]IISc, Bangalore [3]AIIMS, New Delhi
{tyagiaayushkumar, chirag131020, prasenaiims, govindmakharia, mehralalita9910, prathoshap}@gmail.com, mausam@cse.iitd.ac.in

1. Dataset details

Table 1 provides the annotation number for each class of the MuCeD dataset.

Table 1. Annotation numbers

S.No	Tissue Name	Annotated
1	Intra-Epithelial Lymphocyte	2090
2	Epithelial Nuclei	6518

Fig 1 provides the details of how whole slide image (WSI) is sliced to get the image sub-slices. Here, it is important to note that we mask out region apart from the annotated epithelial area. Epithelial area is shown in fig 1 as bold boundary of the villi. We select good villi based on continuity of epithelial layer for our analysis. Further, image is sliced into 9 sub-images, where each dimension is 640 $\times$ 640.

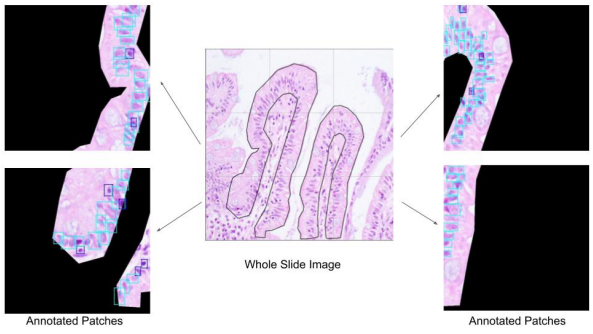

Figure 1. Fig shows sample image of celiac dataset. Each WSI image is sliced into 9 parts and further rescaled to improve the magnification of cells. IELs are marked in violet bounding boxes. and Epithelial nuclei are marked in cyan bounding boxes

[*]Equal contribution

2. Training Object detection model

We tested DeGPR on multiple object detection models to verify if it is object agnostic. Yolov5 is from the class of single stage object detectors. Yolo slices the images into multiple grids and corresponding to each grid, it predicts the bounding boxes with confidence score and class probability map. Faster-RCNN introduced a region proposal network (RPN) to make computation of region proposals cost effective. RPN can be trained end-to-end to generate high quality proposals. EfficientDet introduced Bi-directional feature pyramid network (BiFPN) and compound scaling to uniformly scale depth, width and resolution.
For experimentation, we use official implementation of Yolov5 [*] and Faster-RCNN using are in pytorch. We use pytorch implementation of EfficientDet [*] with pre-trained model efficientdet d0.
We believe that EfficientDet has more parameters and easily overfits on small datasets, hence the performance of EfficientDet is comparitively lower than Yolo and Faster-RCNN. We pre-trained model with Kaggle cell segmentation dataset. However, with CoNSeP and MoNuSac, we did not see improvements with pre-training, since they both have much more training data than MuCeD. DeGPR creates a computation overhead for training GMM $G(\theta)$ for mini-batch. This results in increased training time upto 1.5 hour in addition to baseline training time. Yolov5 takes 3-4 hour to complete the training process depending on early stop. Faster-RCNN takes 6-7 hours with DeGPR and EfficientDet takes 8-10 hours to complete the training process. All experiments were preformed on NVIDIA-RTX 5000. We found p values of 0.008 and 0.007 for Precision and mAP with MuCeD using student's paired t-test. While performing ablation study, we did experiments with only size (mAP 020 0.773) and only intensity (mAP 0.770), and observed that combined size and intensity performs better (mAP 0.779).

[*]https://github.com/ultralytics/yolov5
[*]https://github.com/rwightman/efficientdet-pytorch

Table 2. Ablation for EncoderE_ϕ

Model	Acc	Precision	Recall	F1
ResNet18	0.867	0.938	0.740	0.827
ResNet34	0.8564	0.931	0.6455	0.762
ResNet50	0.8604	0.946	0.661	0.778
ResNet101	0.859	0.929	0.73	0.817

3. Training Contrastive encoder

Fig 2 provide the t-SNE plot of the trained Encoder for one of the fold of the MuCeD dataset. As we know encoder $E(\theta)$ is trained with supervised constrastive loss (L_{SupCon}). Contrastive encoding enforces similar data points to be close. We can see from the fig 2, IEL's (red) and EN (blue) are represented by separate clusters. Hence, encoder $E(\theta)$ is able to learn the discriminative features between the cell types.

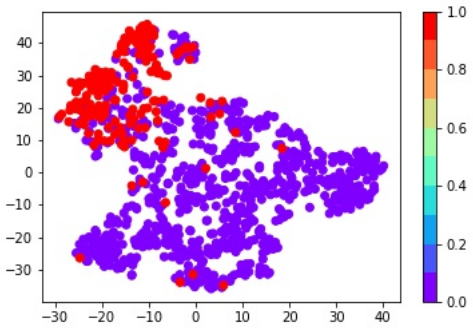

Figure 2. t-SNE plot of embeddings learnt for IEL's (red) and EN (blue). We can see from the embeddings that two clusters are formed which meant that model is able to map similar embeddings close to each other.

Tab 2 provide comparative analysis for different encoder backbones like ResNet 18, 34, 50, 101. We observed that larger models does not provide any additional performance gain hence we used ResNet18 for our experimentation purpose.

4. Counting via detection

Table 3 compares counting via detection with density map based methods for MoNuSAC dataset. We observe from tab 3 Yolov5 with DEGPR is performing well for most of the cell types.

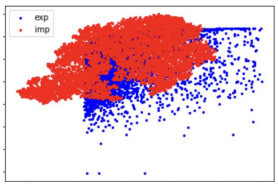

Figure 3. Implicit and explicit features

Table 3. Counting vs Localization (MoNuSAC)

Model	MAE Epith	MAE Lympho	MAE Neutro	MAE Macro	MAE Avg
UNet	57.64	33.91	1.31	2.55	23.85
FCRN-A	64.48	60.71	**0.17**	1.52	31.72
SAU-Net	60.42	54.42	1.49	2.72	29.76
Yolov5 (DeGPR)	**12.01**	**10.69**	0.81	**2.32**	**6.46**

For the density map based methods, we trained separate models for each class type. We can see that Yolov5 with DeGPR has the best performance.

5. Visualization

Fig 5, 6 and 7 provide the qualitative analysis of the model performance on MuCeD, CoNSeP and MoNuSAC respectively. The first column shows the raw images, the second column shows the images with ground truth bounding boxes, the third column shows the performance of the baseline Yolov5 model while the final column shows the performance of the Yolov5 model with DEGPR. We can see that DEGPR helps in solving problems like extra detections, missed detections and misclassifications.

Fig 3 is a t-SNE plot for implicit vs explicit features. We can see from the fig 3 implicit and explict features do not overlap completely and are actually capture complementary information.

6. Convergence of losses

Fig 4 provide loss convergence for detection loss ($\mathcal{L}_{det}$), classification loss ($\mathcal{L}_{cls}$) and DeGPR ($\mathcal{L}_{reg}$). We can observe from the fig 4 that all loss converge over the epochs.

7. SOTA comparision

We compared our method with MCSpatNet. MCSpatNet is a dot-annotation method, so we were only able to add DEGPR with intensity feature. On ConSeP, stomal cell predictions gained 7.7 F1 points, due to intensity differences captures by DEGPR Tab 5. Because MCSpatNet uses spatial information to cluster cells, it fails to perform well on MuCeD (0.549 mean F1, compared 078 to 0.742

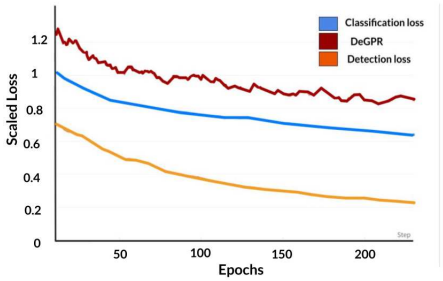

Figure 4. Convergence of losses.

Table 4. F score of individual classes for MuCeD dataset

Model	IEL	Epith
MCSpatNet	0.545	0.554
DEGPR	**0.725**	**0.759**

Table 5. F score of individual classes for CoNSeP dataset

Model	Infl.	Epi.	Sto.
MCSpatNet	0.724	0.695	0.682
DEGPR	0.731	0.632	0.538
MCSpatNet(DEGPR)	**0.736**	**0.698**	**0.759**

Table 6. Comparative analysis with HoverNet for CoNSeP

Model	Dice	AJI	DQ	SQ	PQ
HoverNet	0.853	0.531	0.702	0.778	0.547
Reprod.	0.838	0.534	0.652	0.764	0.499
DEGPR	0.838	0.533	0.655	0.764	0.502

for Yolo+DEGPR), because here, IELs and ENs are interspersed across the region as shown in Tab 4. Another method HoverNet uses segmentation based annotation. We experimented with CoNSeP Tab 6, and found that DEGPR yields negligible improvements (but doesn't hurt performance), probably because vital information of shape is already captured by the base model.

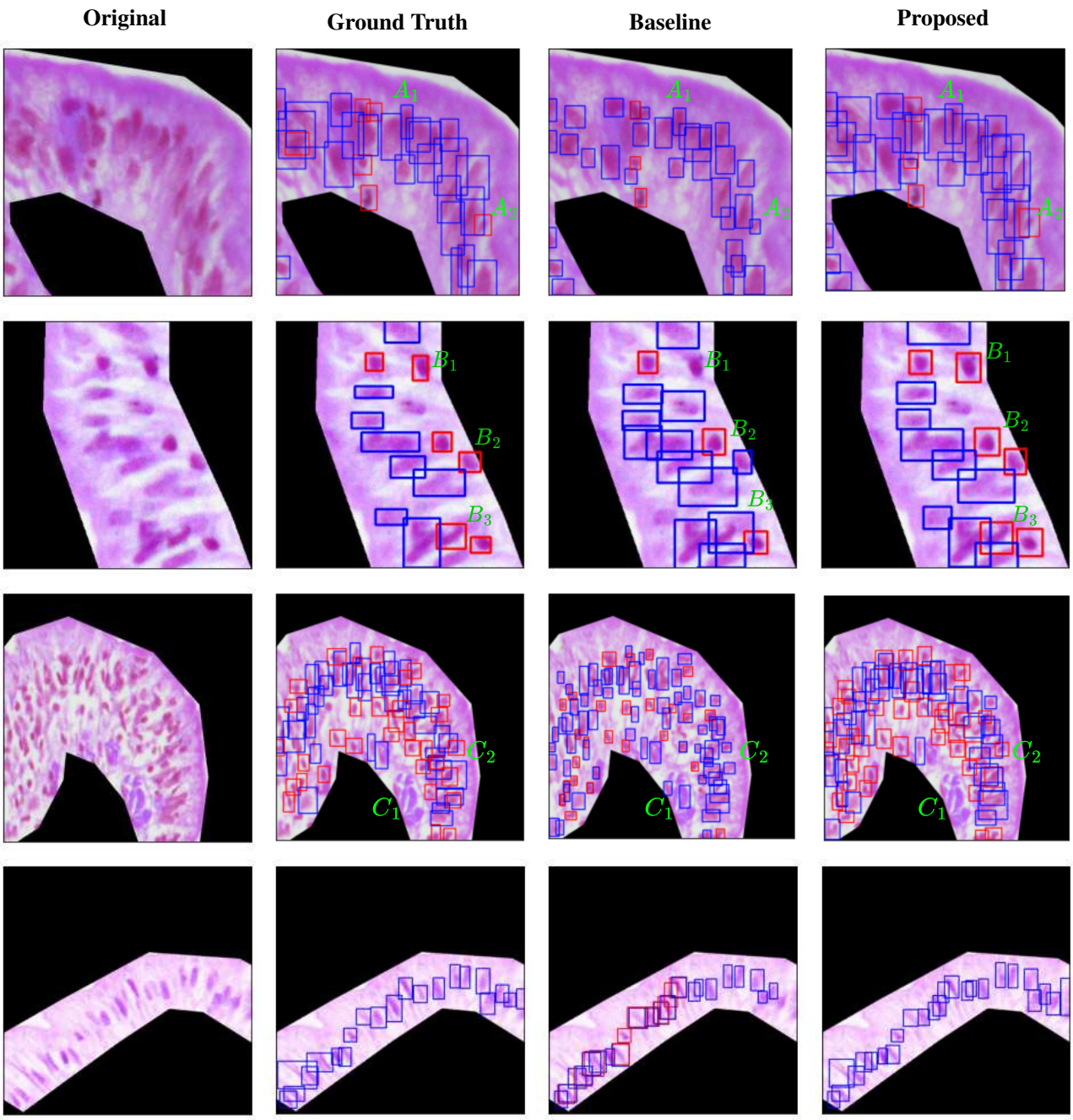

Figure 5. Qualitative performance of DEGPR for MuCeD. The cells are marked as Blue (EN) and Red(IEL). In the first row, in the region A_1, we can see that the baseline model misses multiple Epithelial Nuclei which are overlapping. The model with DEGPR performs better here. In region A_2, the baseline model misclassifies an IEL as an EN. In the second row, the regions B_1, B_2, B_3 show reduction in misclassfication errors as well as cells which were missed. In the third row, in region C_1, the baseline model makes extra predictions which the model with DEGPR has solved. In region C_2, there is an IEL which is inside an EN(in 3-d, it is on top of the EN). The model with DEGPR is successfully able to detect the IEL. Finally, the fourth row has cells of very light intensity. While the baseline model makes misclassifications, the model with DEGPR is successful in detecting all cells correctly.

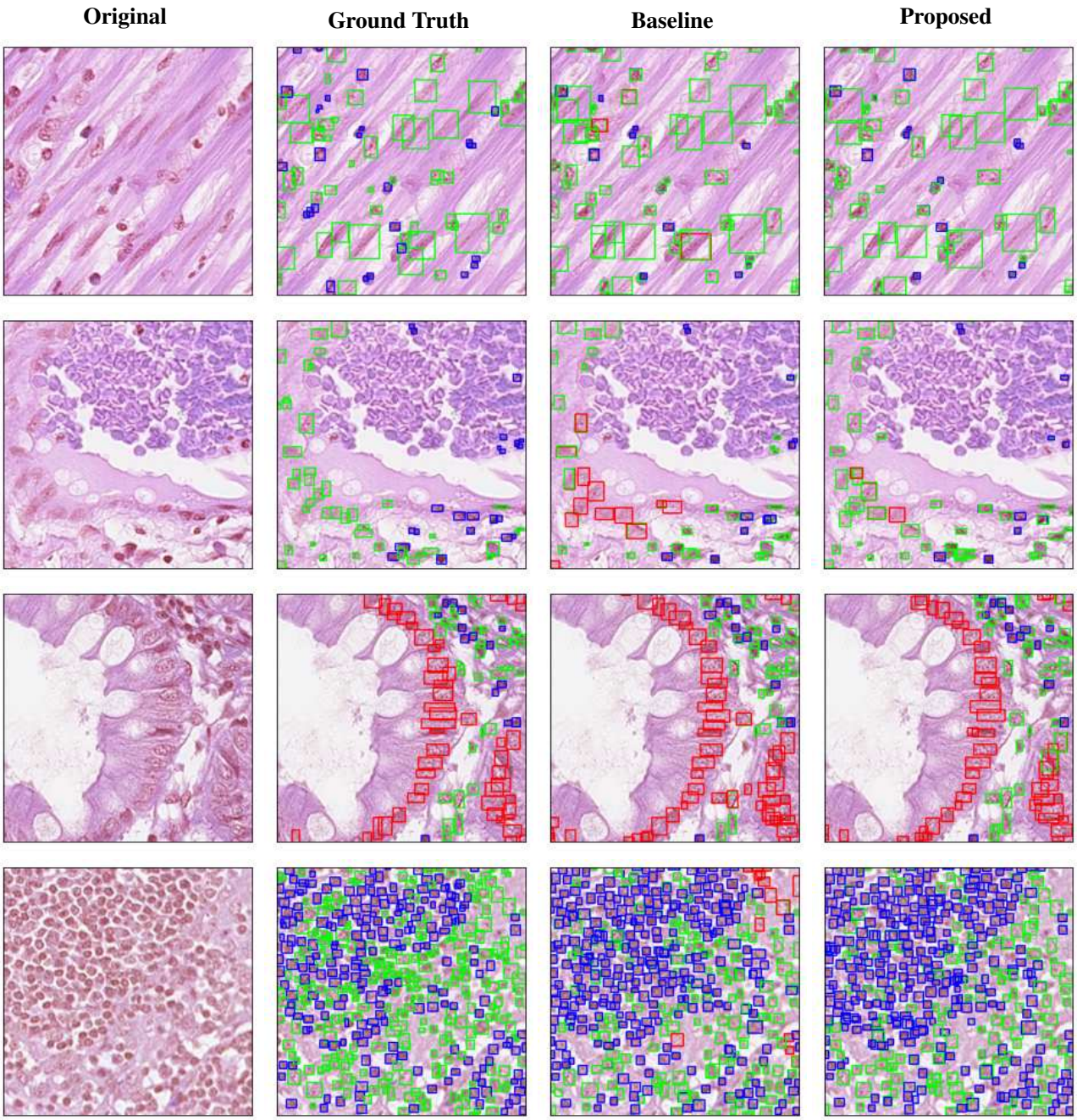

Figure 6. Qualitative performance of DEGPR for CoNSeP. The cells are marked as Blue (Inflammatory), Red (Epithelial) and Green (Spindle). We can see that the baseline model often misclassifies spindle cells as inflammatory cells. This is probably because of their high structural similarity. The model with DEGPR is successfully able to capture the differences between spindle and inflammatory cells. However, it is still not perfect as can be seen in row 4 where there are multiple misclassifications of spindle cells as epithelial cells. We believe one of the main reasons this happened is because of the high density of cells in the image and which dampens the effect of DEGPR since it considers the average feature difference between two classes

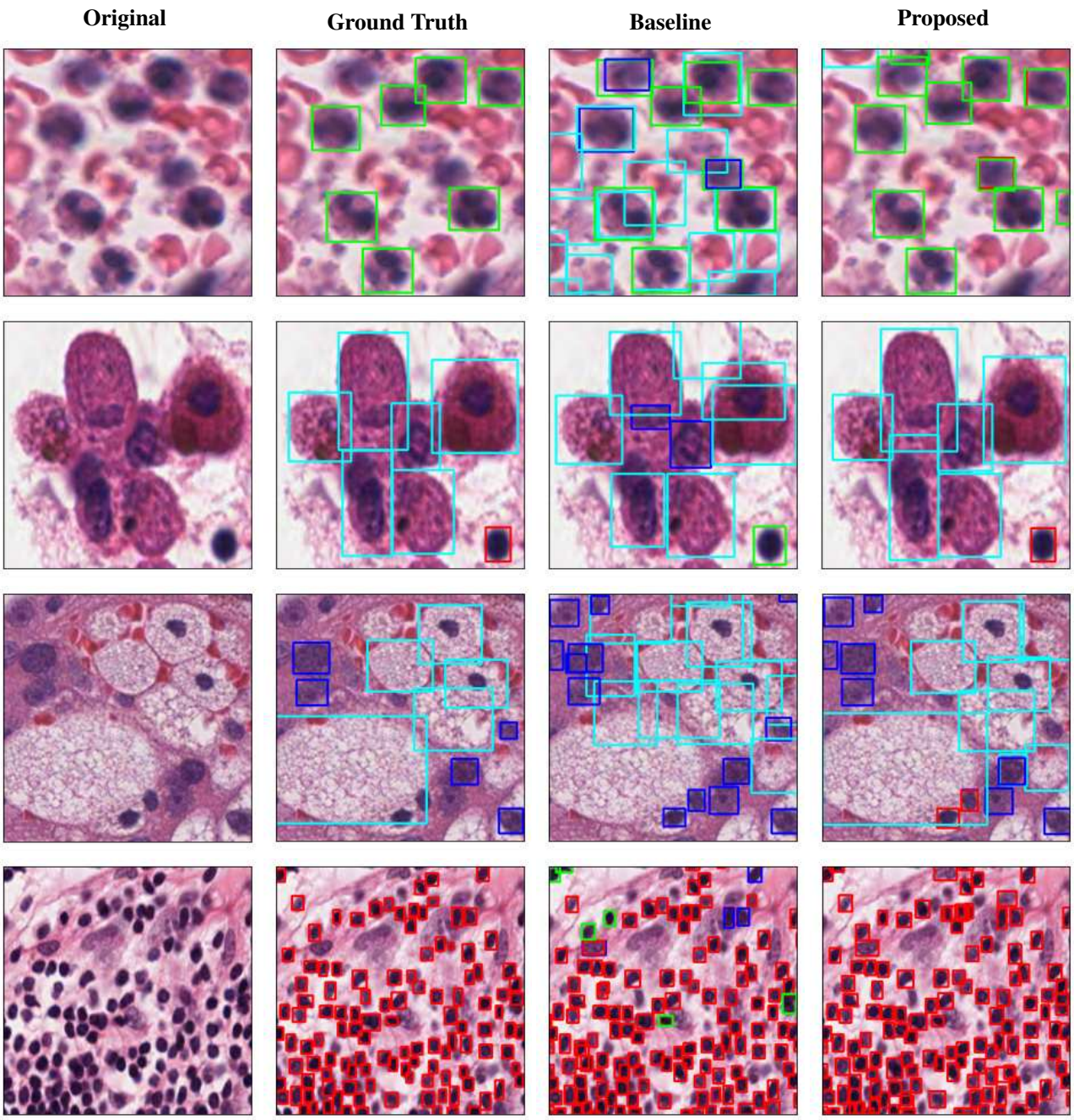

Figure 7. Qualitative performance of DEGPR for MoNuSAC. The cells are marked as Blue (Epithelial), Red (Lymphocyte), Green (Neutrophil) and Cyan (Macrophage). The baseline model often makes extra predictions for the macrophage class as is evident from the first three rows. The model with DEGPR is able to successfully solve this issue. In row 3, we can also see how the model with DEGPR is able to detect the large macrophage which was missed by the baseline model. This is a perfect example where size as an explicit feature is helping the model. In row 2 and row 4, the baseline model misclassifies lymphocytes as epithelial or neutrophil cells. The model with DEGPR is successfully able to solve this issue. At the same time, there are some errors like extra predictions in rows 1 and 3. These extra predictions probably arise since the detected cells are structurally similar to our required cells.