EDNFC-NET: CONVOLUTIONAL NEURAL NETWORK WITH NESTED FEATURE CONCATENATION FOR NUCLEI-INSTANCE SEGMENTATION

Shiv Gehlot[1†] *Anubha Gupta*[1] *Ritu Gupta*[2†].

[1] SBILab, Department of ECE, IIIT-Delhi, New Delhi, 110020, India
[2] Laboratory Oncology Unit, Dr. B.R.A.IRCH, AIIMS, New Delhi 110029, India

ABSTRACT

Accurate nuclei identification is an important step in diagnosis of several diseases. The problem is complex due to heterogeneity in structure, color, and texture among the different categories of cells. The problem is further complicated due to overlapped/clustered nuclei. To address these challenges, we propose an Encoder-Decoder based Convolutional Neural Network (CNN) with Nested-Feature Concatenation (EDNFC-Net) for automatic nuclei segmentation. The feature concatenation cell (FCC) of the EDNFC-Net is made up of two stacks of convolutional filters combined with non-linearity, followed by a concatenation of features. Apart from intra-FCC feature concatenation, a mechanism is also provided for inter-FCC feature concatenation. This arrangement leads to better feature flow and feature-reusability. Similarly, direct feature flow is provided between the encoder and decoder module that preserves the context information. A new loss function with better-penalizing capability is also proposed that helps in the better background and foreground separation. Qualitative and quantitative results are provided on two datasets to validate the proposed architecture and loss function.

***Index Terms*—** Nuclei segmentation, Deep Learning, CNN, Cell detection

1. INTRODUCTION

Nuclei detection, which aids in cell detection, is a starting point in the diagnosis of diseases like cancer. The quick and accurate identification of nuclei under varied conditions can expedite the diagnosis that is necessary for a better survival outcome. Nuclei detection/segmentation is also the preceding step in computer-aided diagnosis, where the aim might be to classify the nuclei as healthy or cancerous. Such automatic tools should not only be resource optimized but also be easily deployable. From all these perspectives, accurate nuclei segmentation is a crucial stage.

Researchers are working on the development of automatic segmentation tools because manual segmentation, apart from being time-intensive, is also vulnerable to human errors. Initial automated segmentation methods were based on clustering, thresholding, and region growing [1]. However, these methods have limited capability in handling complex structures like color/texture variations or clustered cells. Machine learning-based approaches utilizing hand-crafted features have also been applied to this task, but their performance is limited due to low-level features. Deep learning-based methods perform better than the traditional machine learning-based methods as the former learn the task-specific features from the data itself [2].

The U-Net architecture [3] is a convolutional neural network (CNN) based encoder-decoder type architecture that has been successfully used in cell segmentation. The architecture, unlike the fully convolutional network (FCN) [4], involves the concatenation of feature maps from the encoder layers to decoder layers, which helps in context information preservation and generation of smoother segmentation masks. The U-Net, however, finds the segregation of the clustered nuclei a challenging task. This problem is solved in [5] through the additional prediction of the contour of the region of interest and in [6], through the formulation of segmentation as a three-class classification, i.e., contour, foreground, and background.

Some approaches use bounding box proposal networks to perform nuclei-instance segmentation [7, 8]. These networks perform better in separating the overlapped cells but have relatively poor performance in the generation of accurate segmented masks. There have also been to attempts the model segmentation as a regression problem rather than the classification problem [9]. Some approaches combine different methodologies to obtain a more robust pipeline. For example, authors in [10, 11] have utilized CNN in the first step to obtain the approximate segmented masks followed by morphological operations (like watershed) to refine the obtained masks. Similarly, authors in [12, 13] have combined encoder-decoder type architecture with detection networks for better segmentation performance.

In this paper, we propose a single encoder-decoder based architecture, namely, EDNFC-Net that is shown to have ro-

† Corresponding authors. emails: shivg@iiitd.ac.in and dr-ritugupta@gmail.com. Shiv Gehlot would like to thank University Grant Commission (UGC) for Junior Research Fellowship (JRF). Authors would also like to thank the Department of Science and Technology (DST), Govt. of India, for research grant funding (Grant No.: EMR/2016/006183). We also acknowledge the Infosys Center for Artificial Intelligence, IIIT-Delhi for our research work.

bust performance in segmentation mask generation and cluster segregation with datasets of different modalities. A new loss function is also proposed that induces better pixel classification capability.

Henceforth, cell and nucleus have been used interchangeably in the text.

2. PROPOSED EDNFC-NET ARCHITECTURE

The network is designed by arranging the Feature Concatenation Cells (FCCs) in an encoder-decoder type architecture as shown in Fig.1.

Feature Concatenation Cell (FCC): The focus of the FCC is to facilitate the learning of better discriminative features and to induce efficient feature flow. The FCC is composed of two stacks of $k \times 3 \times 3$ convolutional filters, where k represents the number of filters. The dropout and batch normalization is introduced after each convolutional layer. The batch normalization, apart from assisting in faster convergence of training, is also helpful to counter the color variation effects of the input images, which are prevalent due to staining in the microscopic images. Dropout is introduced to check overfitting due to a lack of required training data. Finally, the concatenation of the outputs of both convolutional layers is passed to the next FCC. Such feature concatenation leads to feature reusability and also helps in controlling the vanishing gradient problem through the introduction of the shorter connections [14].

Encoder Module: Encoder module is composed of five FCCs, and each FCC is followed by a downsampling layer, except for the last one. The number of filters consistently increases in a multiple of two, with first FCC and last FCC having 64 and 1024 filters, respectively. This set up leads to a progressive reduction in the feature map size but an increase in the number of feature maps, which helps to capture the intricate structure in the data.

Decoder Module: The decoder module is composed of four FCCs, four upsampling modules, and one output layer. The upsampling module performs upsampling by two on the input features and concatenates the same size feature maps of the encoder module with obtained upsampled feature maps. The direct connections between encoder and decoder are helpful in context information preservation. Upsampling modules progressively increase the feature maps size, which is finally the same as the input image size. The final FCC is followed by an output layer having $c \times 1 \times 1$ filters, with c representing the number of classes. In our experiments, we predict background, and edges along with the foreground, and hence, c is set to three.

Inter-FCC Features Concatenation: Apart from the intra-FCC features concatenation, the network also uses inter-FCC features concatenation. The features of an FCC in the encoder/decoder module are provided to all the following FCCs in the respective modules. However, there is a spatial size discrepancy between the features of the different FCCs, as the following FCC has feature maps of half spatial size than the previous FCC. To resolve this, the features of the FCCs are passed through suitable downsampling followed by convolutional layer with size $16 \times 1 \times 1$, as shown in Fig.1. This convolutional layer limits the number of feature maps being passed directly to the following FCCs. This concatenation mechanism leads to a direct flow of features within the whole encoder/decoder module leading to improved segmentation performance.

2.1. Proposed Loss Function

Consider an input image $\mathbf{A}$. We label the foreground with symbol $\mathbf{F}$, background with symbol $\mathbf{B}$, and edges with symbol $\mathbf{S}$. For a given $\mathbf{A}$, we predict $\hat{\mathbf{A}}$ with $\hat{\mathbf{F}}$, $\hat{\mathbf{B}}$ and $\hat{\mathbf{S}}$. The weighted sum of dice loss and binary cross-entropy (BCE) loss is a commonly used loss function in the segmentation problem. Inspired from the metric introduced in [15] we replace the dice loss with the following balanced-loss ($\mathcal{L}_{BL}$) term:

$$\mathcal{L}_{BL} = 1 - \left(1 - \frac{|\mathbf{B} \cap \hat{\mathbf{F}}|}{|\mathbf{F} \cap \hat{\mathbf{F}}| + |\mathbf{F} \cap \hat{\mathbf{B}}|}\right)\left(1 - \frac{|\mathbf{F} \cap \hat{\mathbf{B}}|}{|\mathbf{F} \cap \hat{\mathbf{F}}| + |\mathbf{F} \cap \hat{\mathbf{B}}|}\right) \tag{1}$$

The last two terms consists of the ratio of $|\mathbf{B} \cap \hat{\mathbf{F}}|$ (false positive) and $|\mathbf{F} \cap \hat{\mathbf{B}}|$ (false negative) with the foreground pixels ($|\mathbf{F} \cap \hat{\mathbf{F}}| + |\mathbf{F} \cap \hat{\mathbf{B}}|$), respectively. The convergence of this loss function demands both of these ratios to be less than one, which in turns will force the minimization of both, false positives (FP) and false negatives (FN). Compared to dice loss given as,

$$\mathcal{L}_{dice} = 1 - 2 * \frac{|\mathbf{A} \cap \hat{\mathbf{A}}| + \delta}{|\mathbf{A}| + |\hat{\mathbf{A}}| + \delta}, \tag{2}$$

where δ mitigates the effect of zero denominator, balanced loss introduces a stronger penalization as shown in Fig. 2. The curves are shown for the varying values of FP and FN. The balanced loss is always higher than dice loss for FP, FN $\in [0, 1]$, particularly in the mid-range.

We use final loss function as a weighted combination of BCE and balanced loss:

$$\mathcal{L} = w_1 \mathcal{L}_{BCE} + w_2 \mathcal{L}_{BL} \tag{3}$$

where $\mathcal{L}_{BCE} = BCE(\mathbf{A}, \hat{\mathbf{A}})$. Due to a stronger penalization, this loss function induces better separation of background and foreground.

2.2. Training and Testing

We have used Keras library for the implementation and Adam [16] optimizer to perform the training. The training process converged in 75 epochs with an initial learning rate of 0.0001 that is commutatively dropped by a factor of 10 after 60 and 70 epochs, respectively. Also, the batch size of 10 is used, and the NVIDIA RTX2080 GPU is utilized to faster the training process. For the weighted combination in the loss functions,

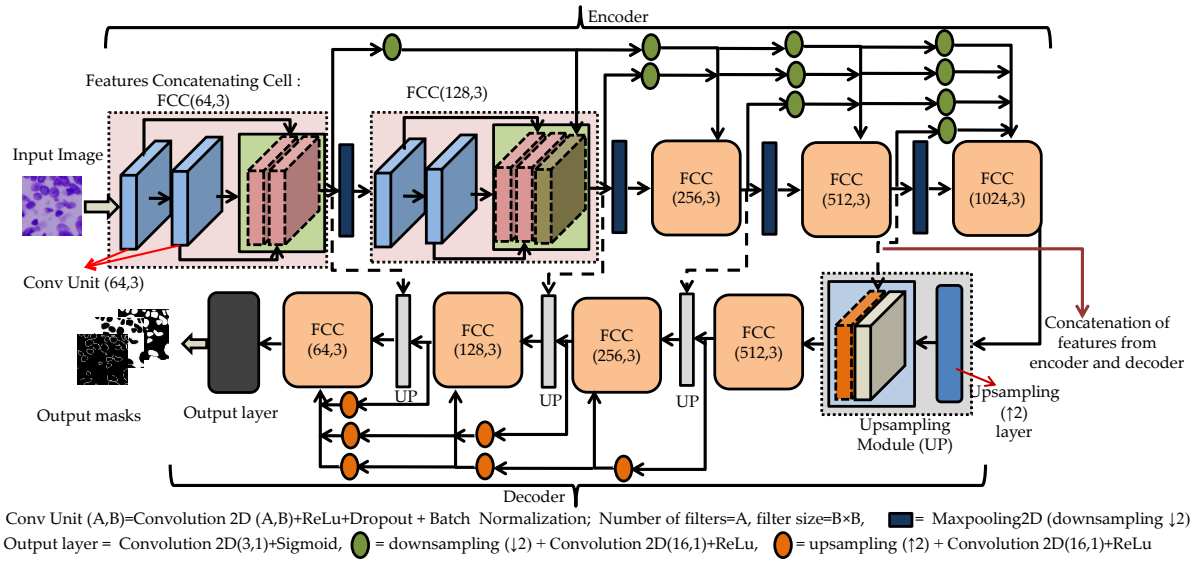


Fig. 1: EDNFC-Net: Encoder-Decoder based Convolutional Neural Network (CNN) with Nested Feature Concatenation

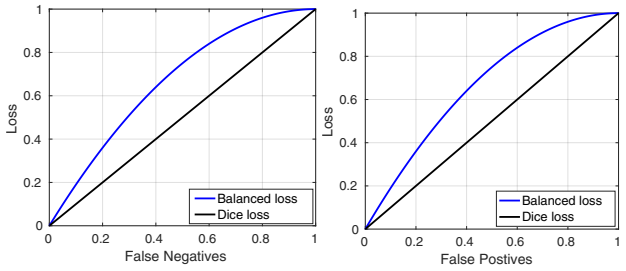


Fig. 2: Variation of balanced loss and dice loss with false negatives (left) and false positives (right).

the value of w_1 and w_2 is set to 0.6 and 0.4 respectively. However, it is not explored if training the EDNFC-Net using other commonly used optimizers like SGD or RMSprop [17] will achieve better performance. This analysis can be carried out in the future work.

3. DATASET DESCRIPTION

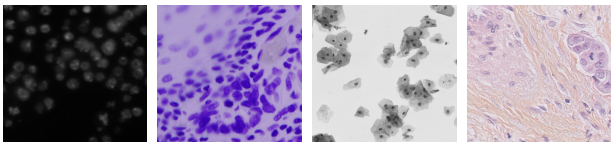
Two datasets are used to evaluate the performance of the EDNFC-Net. The first dataset is the 2018 Data Science Bowl challenge (DSBC18) dataset, which has images of three different modalities: i) histology, ii) fluorescence, and iii) brightfield [18]. The second dataset consists of 50 Histology images of breast cancer (BC19) [9]. We have also performed experiments by mixing these two datasets, which leads to a more complex dataset. The sample images from these datasets are shown in Fig-3. A detailed description of training and test sets is presented in Table-1. The 10% of the total images are used for testing, while the training is performed using small-sized patches (refer Table-1) of the remaining 90% images, with a 10% subsection used as the validation set.

4. RESULTS AND DISCUSSION

To evaluate the quantitative performance we have used dice similarity score (DSC) or F_1 score and intersection over union

Table 1: Dataset description. The training and testing set is obtained by extracting the patches of a given size from the images.

Dataset	Total Images	Patch Size	Training	Test
DSBC18 [18]	660	128×128	10000	2503
BC19 [9]	50	128×128	2080	297



(a) (b) (c) (d)

Fig. 3: Some samples from the dataset used. (a) DSBC18 (Flourescence) [18] (b) DSBC18 (Histology) [18] (c) DSBC18 (Brightfield) [18] (d) BC19 [9]

(IoU) or Jaccard similarity index. DSC and IoU are the frequently used evaluation metrics for medical image segmentation [19]. For an image $\mathbf{I}$ with ground truth segmentation mask $\mathbf{F}$, and predicted segmentation mask $\hat{\mathbf{F}}$, the DSC and IoU are given as:

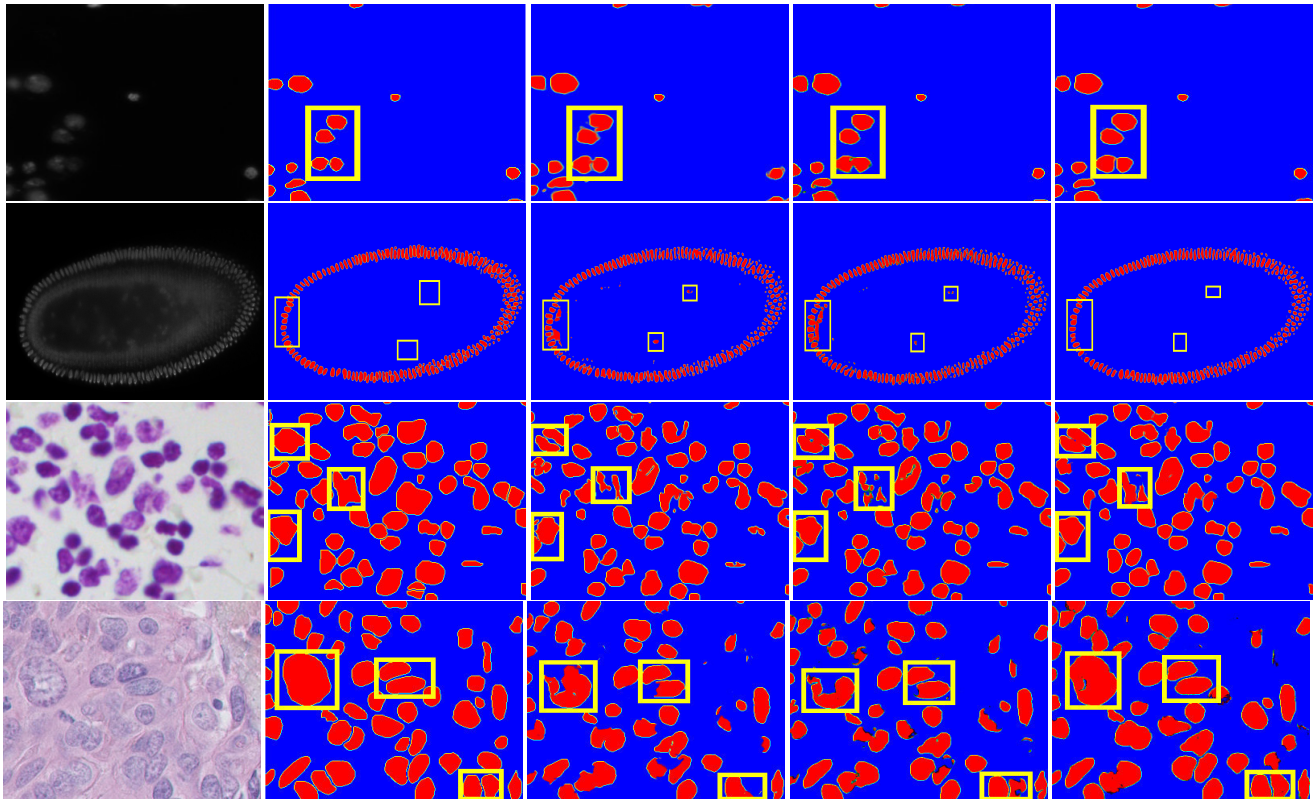
$$\text{DSC} = 2 * \frac{|\mathbf{F} \cap \hat{\mathbf{F}}|}{|\mathbf{F}| + |\hat{\mathbf{F}}|}, \text{ and IoU} = \frac{|\mathbf{F} \cap \hat{\mathbf{F}}|}{|\mathbf{F} \cup \hat{\mathbf{F}}|} \tag{4}$$

Both of these metrics have values in the range $[0, 1]$ with higher value denoting the more similarity between prediction and ground truth. Also, the better performing model will have higher values of the DSC and IoU.

We have performed the qualitative and quantitative analysis to evaluate the performance of EDNFC-Net and the proposed loss function (Eq. 3). The performance of EDNFC-Net is compared with U-Net trained with a weighted sum of dice loss and BCE loss, and U-Net$_{\text{BL}}$, the U-Net trained using the proposed loss (Eq. 3). Table-2 shows the quantitative performance comparison in terms of DSC and IoU. Since the

Table 2: Comparison of U-Net, U-Net$_{BL}$ and EDNFC-Net on images of different modalities of DSBC18 [18], on BC19 [9] and on the mixture of DSBC18 and BC19

	Fluorescence (DSBC18)			Histology (DSBC18)			Brightfield (DSBC18)			BC19 [9]			Mixed (DSBC18 and BC19)		
	U-Net	U-Net$_{BL}$	EDNFC-Net	U-Net	U-Net$_{BL}$	EDNFC-Net	U-Net	U-Net$_{BL}$	EDNFC-Net	U-Net	U-Net$_{BL}$	EDNFC-Net	U-Net	U-Net$_{BL}$	EDNFC-Net
DSC	.901	.904	**.919**	.838	.845	**.858**	.850	.866	**.890**	.762	.742	**.766**	.858	.860	**.874**
IoU	.835	.839	**.858**	.736	.744	**.769**	.760	.782	**.816**	.640	.618	**.648**	.768	.772	**.792**



(a) Original (b) Ground Truth (c) U-Net (d) U-Net$_{BL}$ (e) EDNFC-Net

Fig. 4: Rows 1 to 3: Results on DSBC18 dataset [18] and Row 4: Results on BC19 dataset [9]. Results are shown with U-Net, U-Net$_{BL}$, and EDNFC-Net. Results of Row 4 on BC19 are shown on the model trained on the mixture of the dataset DSBC18 and BC19. Pixels in the yellow boxes highlight the performance difference between the different models. These highlighted pixels contrast the difference in cluster segregation and segmentation mask generation with different architectures.

DSBC18 dataset consists of images of three different modalities, we have shown the performance for each modality separately. The EDNFC-Net is performing better for all three types of modalities. Maximum improvement is observed for brightfield modality with 4% and 5.5% gain in DSC and IoU, respectively. Similarly, U-Net$_{BL}$ is performing better than U-Net, which proves the hypothesis regarding the proposed loss function. Similar trends are also seen for the BC19 dataset with EDNFC-Net leading the remaining methods. However, U-Net$_{BL}$ is lagging with respect to U-Net in this case. To further evaluate the performance, we mixed the images of both the datasets (DSBC18 and BC19), to form a more challenging dataset. For this dataset also, EDNFC-Net is the leading method, and U-Net$_{BL}$ also has better performance than U-Net.

The qualitative results for all three methods are shown in Fig. 4. From the highlighted pixels in row 1, it is observed that EDNFC-Net has better performance in cluster segregation. Similarly, from row 2-row 4, it can be seen that EDNFC-Net is more immune to false positives and false negatives as compared to the other two methods. Also, it can be inferred from Fig. 4 that U-Net$_{BL}$ shows better qualitative results with respect to U-Net, which again proves the contribution of the proposed loss function.

5. CONCLUSION

In this paper, we have addressed the problem of nuclei-instance segmentation problem on two cancer imaging dataset. We have made two contributions: 1) we have proposed a new EDNFC-Net architecture with intra and multiscale inter-feature concatenation in encoder and decode, 2) EDNFC-Net is trained with a new loss function consisting of a weighted combination of BCE and proposed balanced-loss. On both the dataset, the proposed architecture trained with the proposed performs much better compared to a) U-Net trained with weighted combination of BCE and dice loss, and b) U-Net trained with the proposed loss function.

6. REFERENCES

[1] E. Meijering, "Cell segmentation: 50 years down the road [life sciences]," *IEEE Signal Processing Magazine*, vol. 29, no. 5, pp. 140–145, 2012.

[2] Juan C. et al. Caicedo, "Evaluation of deep learning strategies for nucleus segmentation in fluorescence images," *Cytometry Part A*, vol. 95, no. 9, pp. 952–965, 2019.

[3] O.Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer Assisted Intervention*, 2015, pp. 234–241.

[4] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 3431–3440.

[5] H. Chen et al., "DCAN: Deep contour-aware networks for object instance segmentation from histology images," *Medical Image Analysis*, vol. 36, pp. 135 – 146, 2017.

[6] N. Kumar et al., "A dataset and a technique for generalized nuclear segmentation for computational pathology," *IEEE Transactions on Medical Imaging*, vol. 36, no. 7, pp. 1550–1560, 2017.

[7] S.U. Akram et al., "Cell segmentation proposal network for microscopy image analysis," in *Deep Learning and Data Labeling for Medical Applications*, 2016, pp. 21–29.

[8] J. W. Johnson, "Adapting Mask-RCNN for automatic nucleus segmentation," *arXiv:1805.00500*, May 2018.

[9] P. Naylor et al., "Segmentation of nuclei in histopathology images by deep regression of the distance map," *IEEE Transactions on Medical Imaging*, vol. 38, no. 2, pp. 448–459, Feb 2019.

[10] P. Naylor, M. La, F. Reyal, and T. Walter, "Nuclei segmentation in histopathology images using deep neural networks," in *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, April 2017, pp. 933–936.

[11] W. Wang et al., "Learn to segment single cells with deep distance estimator and deep cell detector," *Computers in Biology and Medicine*, vol. 108, pp. 133 – 141, 2019.

[12] A. O. Vuola, S. U. Akram, and J. Kannala, "Mask-RCNN and U-Net ensembled for nuclei segmentation," in *IEEE International Symposium on Biomedical Imaging*, April 2019, pp. 208–212.

[13] Z. Xu et al., "US-Net for robust and efficient nuclei instance segmentation," in *IEEE International Symposium on Biomedical Imaging*, April 2019, pp. 44–47.

[14] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 2261–2269.

[15] F. A. M. Cappabianco et al., "A general and balanced region-based metric for evaluating medical image segmentation algorithms," in *2019 IEEE International Conference on Image Processing (ICIP)*, Sep. 2019, pp. 1525–1529.

[16] J. Ba D. P. Kingma, "Adam: A Method for Stochastic Optimization," *arXiv:1412.6980*, Dec 2014.

[17] T. Tieleman and G. Hinton, "Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude," COURSERA: Neural Networks for Machine Learning, 2012.

[18] "2018 data science bowl," `https://www.kaggle.com/c/data-science-bowl-2018`, Accessed: 2019-10-02.

[19] J. Bertels et al., "Optimizing the dice score and jaccard index for medical image segmentation: Theory and practice," in *International Conference on Medical Image Computing and Computer Assisted Intervention*, 2019, pp. 92–100.