

Gated-Shape Convolutional Neural Networks (GSCNN) for Semantic Nuclei Segmentation In Digital Pathology Images

Justin T. Pontalba

Faculty of Biomedical Engineering

Yeats School of Graduate Studies, Ryerson University

Toronto, Canada

jpontalb@ryerson.ca

Abstract—Segmenting overlapping nuclei continues to be a barrier for developing robust computational pathology tools. This paper addresses this barrier by implementing a Gated Shape CNN (GSCNN) which leverages multiple streams for simultaneously segmenting semantic classes and edge classes. By using two streams, the GSCNN attempts to separate the processing of high level abstractions such as objects, and low level features such as colour and edges. Instead of a ResNet, which was used previously to process the high level abstractions, a UNET is proposed. As a result of this alteration, improved performance was observed for both nuclei and boundary segmentation for the GSCNN-UNET architectures.

Index Terms—digital pathology, image processing, convolutional neural networks, gradients

I. INTRODUCTION

Object segmentation continues to be a highly researched topic in computational pathology tasks. Specifically, individual nuclei segmentation is needed to automate the task of analyzing the morphological features of hematoxylin and eosin (H & E) stained tissue biopsies. Conventionally, histological analysis is performed by a Pathologist to determine the diagnosis or severity of tissue under investigation. However, this task can be extensive and the workload can be overwhelming. By automating this task through computational pathology tools, the goal is to improve diagnostic accuracy, improve workflow for pathologists, and ultimately improve patient outcomes.

Deep convolutional neural networks (DCNNs) have been the standard method for performing object segmentation. In both natural images and medical images, DCNNs have demonstrated improved accuracy compared to traditional methods, easy off-the-shelf implementation, and less time consuming when designing pre or post processing methods. For DCNNs, more time and effort have been dedicated to designing new architectures or re-designing architectures to improve model performance. Despite the variations of CNN architectures that exist, segmenting overlapping nuclei continues to be a barrier for developing robust computational pathology tools.

In this work, the *Gated-Shape CNN (GSCNN)* architecture is implemented for semantic nuclei segmentation. The GSCNN used in this paper is adapted from [1] and utilizes two streams for learning, 1) the *Regular Stream* and 2) the *Shape Stream*.

Because of the separate streams multiple loss functions and regularizers are used for learning. The goal of utilizing the GSCNN architecture is to improve the segmentation of overlapping nuclei. By using the two stream methodology, shape and boundary information should be learned more effectively resulting in improve segmentation performance.

To evaluate performance, variations of the GSCNN model are evaluated and compared to a baseline UNET architecture. The variations of the GSCNN architecture include the use of regularizers as well as different CNN architectures for the *Regular Stream* which are RESNET and UNET. To quantify performance the *dice similarity coefficient* is used.

This paper is organized as follows. *Materials and Methods* are described in Section II, *Results* are presented in Section III, and *Discussions and Conclusions* are given in the last section, Section IV.

A. Related Works

1) *Semantic Segmentation*: State-of-the-art approaches for semantic nuclei segmentation are predominantly based on CNNs. An architecture that has been utilized and redesigned to many variations is the UNET architecture by [2]. The UNET architecture utilizes a encoding stream which learns dense feature representations, and a decoding stream which localizes the objects and performs convolution 2D-transpose to resize the image back to the original image space. Other works such as [3], [4], and [5] have adapted the UNET architecture to improve performance. [3] introduces the *Inception U-Net* architecture which combines the UNET structure with *inception layers* of the GoogLe-Net. The inception layer is known to be flexible in automating the selection of layer types to use in DCNNs [3] [6]. In a similar way, [4] alters the UNET architecture by using *residual inception blocks* in the encoding stream, and by adding additional streams; 1) for decoding the nuclei predictions and 2) for decoding the nuclei-boundary predictions. The boundary stream uses the standard decoding blocks while the nuclei stream uses channel-attention blocks [4]. Lastly, in [1], the work in which this paper is based on, uses two streams as well. The first stream, *RegularStream*, uses traditional fully convolutional architectures such as RESNET.

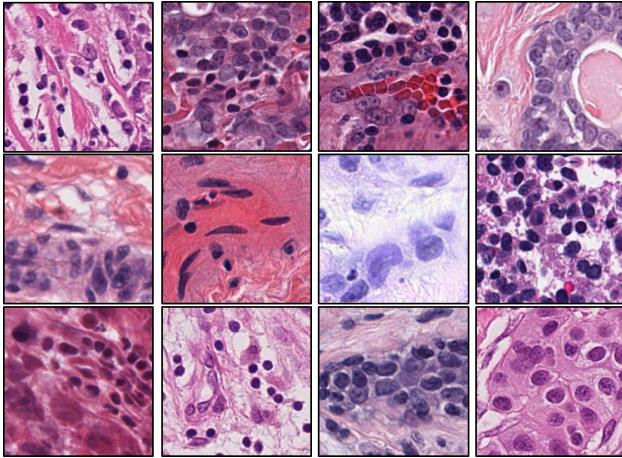


Fig. 1. Example images of dataset

The second stream, *ShapeStream*, uses gated convolutional layers with image gradients and edge labels as inputs. The outputs of both streams are then combined using atrous spatial pyramid pooling (ASPP) followed by the final prediction layer. In this paper, the RegularStream is evaluated using 1) a RESNET + ASPP and 2) using a UNET architecture. The ShapeStream is kept constant between these evaluations. In addition, the use of regularization is evaluated between both methodologies. By carrying out the experiment in this way, the effects of using ASPP for decoding and the effects of regularization are observed.

II. MATERIALS AND METHODS

A. Data

The data used in this paper was adapted from [7] and directly accessed from *Genomic Data Commons (GDC) Data Portal*. From both sources, only breast tissue is used where the disease type of the tissue is ductal and lobular neoplasms. The original whole slide images, (WSI) from which the regions of interest (ROI) images were cropped, were scanned at 40X magnification. The semantic annotations were completed by an undergraduate research assistant where nuclei, boundary, and background classes were identified. [7] provided ROIs of size 1000 x 1000 pixels, whereas the ROIs cropped from GDC cases are of size 512 x 512 pixels. Both image sets were subsequently cropped to a size of 256 x 256 pixels as the CNN architecture used in this work takes such image size as input. In total, 296 256 x 256 ROIs and associated ground truths are used. From the image set, gradient images are produced which will be addressed in a later section. **Figure 1** below depicts examples of patches used in the experimentation. The dataset demonstrates the variation in the appearance of the ROIs as well as the variation in number of nuclei.

B. Architecture

Figure 2 depicts a general overview of the GSCNN implementation. As previously stated, the architecture consists of two streams, the *RegularStream* and the *ShapeStream*. The

goal of the RegularStream is to process the input images and labels normally using any fully convolutional neural network. The ShapeStream on the other hand consists of depthwise, residual, and gated convolutional layers and is to process the gradients of the input images. The inputs for the RegularStream are the RGB images and semantic labels, whereas, the inputs to the ShapeStream are gradients of the RGB images and edge labels. The outputs of each stream are subsequently combined and passed to the output layer which contains the semantic predictions.

1) *Regular Stream*: In the original implementation, the authors compared the performance of ResNet and VGG architectures as the RegularStream. In this paper, the baseline implemented uses the ResNet with a decoding ASPP module as the RegularStream. Furthermore, to compare the performance of other architectures as the RegularStream, the UNET CNN is proposed. The RegularStream can be denoted as $R(I)$, takes images $I \in R^{256x256x3}$, and outputs $R_{l_n}(I)$ and $R_f(I)$ where l_n denotes outputs at various layers and f denotes the final semantic output. **Figure 2** and **Figure 3** depict the intermediate outputs of $R(I)$ and their flow into the *ShapeStream*, and the structure of the ResNet and UNET architectures respectively.

a) *ResNet and Atrous Spatial Pyramid Pooling*: The ResNet architecture only contains an encoding stream where inputs are encoded down to dense feature representations. The residual blocks used in this work are depicted in **Figure 3a**). The blocks consist of alternating layers of batch normalization, convolution, ReLu activation, and conclude with max pooling. In addition, the inputs of the blocks are added to the outputs which promote *residual learning*. For instance, if the inputs to the ResBlocks are considered x and the residual block is considered $f(x)$, then the sum at the output is $h(x)$. This operation is written as:

$$h(x) = f(x) + x, \quad (1)$$

or in terms of the residual block

$$f(x) = h(x) - x. \quad (2)$$

This operation is a result of the *skip connection* which forces the residual block to model **equation 2**, the residual mapping. Residual learning is known to address the problem of vanishing gradients, and allows information to flow more easily from one layer to the next [8].

In order to transform the dense features back to the original image space, instead of using decoders, atrous spatial pyramid pooling (ASPP) and bilinear interpolation is used. In other works [9], these methods form the decoding component of the architecture. [9] stipulates that the pooling, convolution, and stride operations of conventional decoding streams, as in UNET, decimates learned information related to object boundaries. ASPP and bilinear transformation in natural image applications maintain boundaries, which is a goal in semantic nuclei segmentation. Furthermore, these methods have been proven to capture multi-resolution features which mirror that

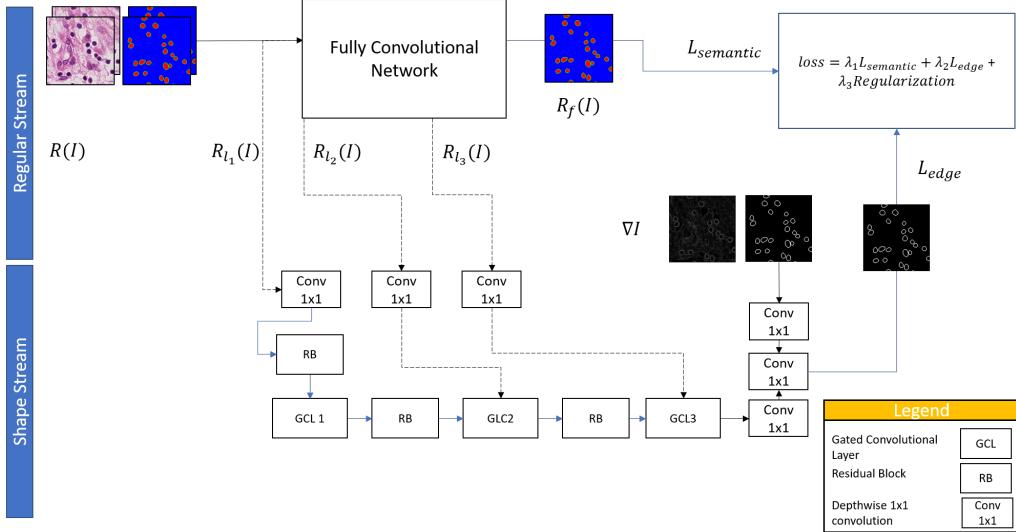


Fig. 2. Overview of Architecture

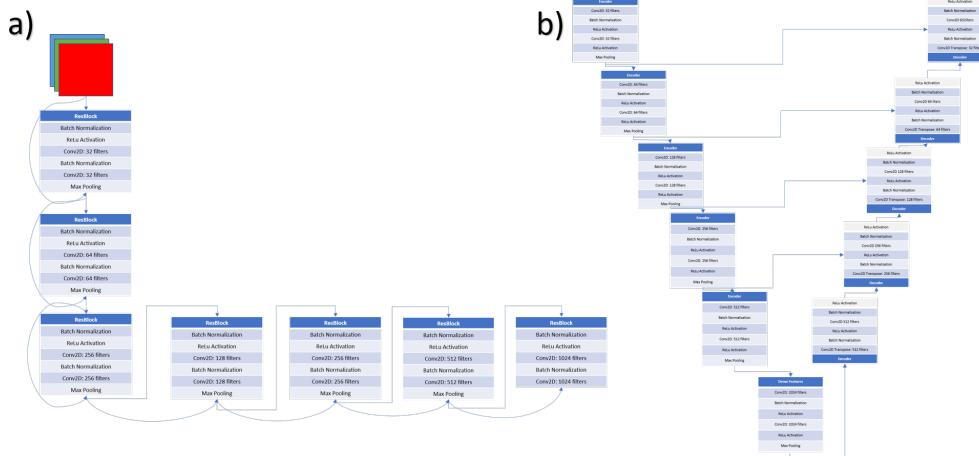


Fig. 3. ResNet and UNET Architecture

of the UNET. This makes ASPP suitable for the RegularStream. **Figure 4** presents the ASPP methodology applied to the deep feature representation. Multi-resolution information is captured using dilated convolutions which operate similarly to conventional convolution, but instead use a larger receptive field to capture neighbourhoods. For instance, rather than using neighbourhoods where pixel neighbours are adjacent to one another, the neighbourhood is dilated where the samples are captured further apart. Depending on the dilation rate, contextual or more localized features will be captured. From **Figure 4**, dilation rates of 6, 12, and 18 are concatenated with the dense feature representation followed by depthwise convolution. This feature mapping is used concatenated with the ShapeStream output where semantic prediction will be conducted. Final semantic prediction will be explained in a later section.

b) UNET: **Figure 3b** presents the UNET architecture. The UNET is used as the RegularStream without the ASPP

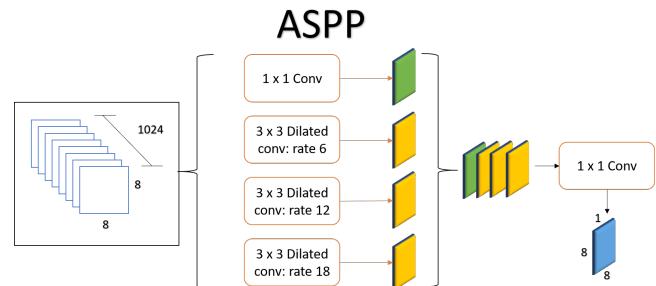


Fig. 4. ASPP Module

module because a decoding stream is already a component of the architecture. Furthermore, in another experiment the UNET without a ShapeStream is used to compare performance to observe if the ShapeStream impacts the overall segmentation performance.

The encoding blocks of the UNET are comprised of convolution, batch normalization, and ReLu activation layers followed by max pooling. In addition, skip connections are concatenated to the decoding stream to localize features in the output that were learned from the encoding stream. Furthermore, to incorporate information learned in the ShapeStream, the output of the last layer of the UNET is concatenated with the outputs of the ShapeStream.

2) *Shape Stream*: **Figure 2** depicts the ShapeStream of the GSCNN architecture. This components contains depthwise convolution layers, residual blocks, and the actual *gated* components of the GSCNN. The ShapeStream architecture remains constant despite the interchangeability of the RegularStream. The inputs to the ShapeStream are gradients of the gray-scale versions of the RGB images, ∇I , and the corresponding edge labels.

a) *Image Gradients*: To obtain ∇I , the gradient magnitude is formed from the Sobel horizontal and vertical components. These components are found through the convolution of the Sobel horizontal and vertical kernels with gray-scale input images. The resulting images are normalized between [0,255].

b) *Depth-wise Convolution*: Depthwise or 1×1 convolution is used to reduce the intermediate feature representations, R_{l_n} coming from the RegularStream. Intermediate representations from the third and fourth encoders, or R_{l_3} and R_{l_4} , are reduced to a single channel by depthwise convolution. In depthwise convolution the features are added along the depth dimension.

c) *Gated Convolution Layer*: The goal of the gated convolution layer (GCL) is to facilitate the flow of information from the RegularStream to the ShapeStream and to help the ShapeStream to only process relevant information by supressing the rest [1]. In general, the GCL deactivates irrelevant features of the ShapeStream by using features from the RegularStream through using *attention maps*. Let $\alpha_l \in R^{H \times W}$ be an attention map formed by concatenating processed R_{l_n} with a representation of the ShapeStream, $s \in R^{256 \times 256}$. After concatenation, α_l is passed through batch normalization, convolution, ReLu activation, convolution, batch normalization layers concluded with sigmoid activation. In the following equation, the gate output is represented by g with input α , sigmoid activation σ , and intermediate operations as z

$$g = \sigma(z(R_{l_n} || s)) \quad (3)$$

g is then passed through residual blocks and subsequently passed to the next GCL, where it is again combined with another intermediate feature representation from the RegularStream. At the end of the ShapeStream, outputs from the gates are combined with the gradients using concatenation followed by depthwise convolution and sigmoid activation. This output is then fused with the output of the RegularStream through concatenation, followed by final semantic prediction using sigmoid activation. The sigmoid activation applied in the GCL is the component which suppresses particular features.

Since only relevant features are wanted, features which do not have high values will be attenuated by the sigmoid and features which are relevant will be passed. It is predicted that edge information will be captured by the gates from the intermediate representations of the RegularStream.

3) Multitask Learning:

a) *Loss Functions*: To reiterate, the outputs of the RegularStream are the semantic predictions of nuclei, boundary, and background labels, whereas, the outputs of the ShapeStream are predictions of the boundary class. The RegularStream is trained using RGB images, whereas the ShapeStream is trained on gradients of the grays-scale version of the RGB images. The streams are trained simultaneously using two components; 1) the *binary cross entropy (BCE) loss* comprised of the semantic and edge predictions, and 2) the dual task regularizer. The BCE loss is presented in the equation below:

$$L = \lambda_1 L_{BCE}(s, \hat{s}) + \lambda_2 L_{BCE}(\hat{y}, y) + L_{reg} \quad (4)$$

where \hat{s} represents the edge labels, and \hat{y} represents the semantic labels. There s and y represent the edge and semantic predictions respectively. According to [1], λ_1 is calculated as β , the ratio of the total number of edge pixels to the total number of pixels in an image (equation 5). In contrast, λ_2 is calculated by **equation 6**.

$$\lambda_1 = \beta = \frac{e^+}{totalpixels} \quad (5)$$

$$\lambda_2 = 1 - \beta \quad (6)$$

By observing **equation 5** and **equation 6**, it is clear that the total loss is weighted by the presence of edge pixels in the image during training. This type of weighting demonstrated success in balancing pixel classes in natural image semantic segmentation [10].

b) *Dual Task Regularizer*: L_{reg} is a component of the total loss and acts as a regularizer to penalize the loss when boundaries predicted the semantic prediction differ from the prediction of the ShapeStream. The equation for L_{reg} is presented in **equation 7**:

$$L_{reg} = \lambda_3 * \sum |\hat{y}_{edge} - y_{edge}| + \lambda_4 \sum \hat{y} \log(y * s), \quad (7)$$

where \hat{y}_{edge} and y_{edge} represent the semantic edge label and semantic edge prediction respectively, and y and s represent the semantic prediction and shape edge prediction. In this paper, λ_3 and λ_4 have equal weights of 1 [1].

In this paper, experimentation is carried out with and without regularization to observe if regularization has an impact on overall performance.

III. EXPERIMENTAL RESULTS

In this section, the results of the experimentation is presented. To reiterate, semantic segmentation performance between versions of the GSCNN model are compared. Versions

of the GSCNN include 1) a ResNet RegularStream with ASPP 2) the proposed model with a UNET RegularStream 3) a UNET architecture without the ShapeStream, and 4) architectures 1) and 2) without the use of regularization. The overall goal is to observe if semantic segmentation improved with the help of the ShapeStream, and to observe if regularization has an effect on architectures that use it. The UNET model without the ShapeStream is included in the evaluation to observe the GSCNNs performance in comparison to a basic DCNN architecture.

A. Data Preparation and Training Protocol

To develop the DCNN models used in this paper, the TCGA-breast data from [10] and our own GDC ROIs were used. A total of 296 256 x 256 ROIs were randomly split into ~ 72% training, ~ 8% validation, and ~ 20% testing. These proportions were chosen such that the training set contained a large variation of breast tissue images. Furthermore, the testing set should be representative of data not seen during training. This is done to observe how the models perceive new instances. The ground truth images for the semantic stream were one-hot-encoded, where each batch was of size b x H x W x K. In the experiments this translates to b, a batch size of 3, H x W, image dimensions of 256 x 256, and K = 3, for three classes (nuclei, boundary, and background). Similarly, the ground truth batches for the ShapeStream were of size 3 x 256 x 256 x 1, for a batch size of 3, image dimensions of 256 x 256, and because the classes are binary, a channel size of K=1 was used. A batch size of three was chosen due to GPU memory limitations.

Each model was trained using the *Adam optimizer* for 100 epochs. λ_1 and λ_2 are adaptive weights based on **equation 5** and **equation 6**, while λ_3 and λ_4 were set to a value of 1. A value of 1 was used because the regularizer adapts depending on the quality of the predictions between the RegularStream and ShapeStream. As such, the weight of the regularizer was maintained at 1 to ensure the effects of the regularizer are not exaggerated (λ_3 and $\lambda_4 \geq 1$), or suppressed (λ_3 and $\lambda_4 \leq 1$).

The models were trained using a system with a Intel Core i7-8750H CPU at 2.20 GHz, 16 GB of RAM, and 256 GB SSD. Training duration for each model was approximately 50 minutes.

B. Metrics for Evaluation

Semantic segmentation performance is evaluated using both qualitative and quantitative observations. Qualitatively, a positive performance would demonstrate clear separation between nuclei, boundary, and background classes, continuous edges, segmented nuclei without holes, and high probability for predicted classes. To quantify model performance, the *dice similarity coefficient is used*.

1) *Dice Similarity Coefficient*: The *Dice Similarity Coefficient (DSC)* is an overlapping metric used to quantify how much overlap there is between a prediction and label. In general, it is a ratio of 2x the number of true positives, TP, to

the sum of 2x TP, false positives, FP, and false negatives, FN. The equation for DSC is presented below.

$$DSC = \frac{2 * TP}{2 * TP + FP + FN} \quad (8)$$

Lastly, the loss functions are analyzed to observe how the use of regularization effects the overall performance of the model in relation to DSC.

C. Nuclei Segmentation Model Performance

1) *Loss Functions*: The baseline method used in this paper is the original GSCNN implementation by [1], which uses a ResNet and ASPP module, whereas the proposed model uses a UNET in the RegularStream. Furthermore, the performance of these models are compared when regularization is used or omitted. Lastly, all the models are compared to a basic UNET which demonstrated positive segmentation performance in previous works [11] [4] [9] [2]. **Figure 5** presents the training and testing results with respect to the loss functions. Column a) depicts the semantic loss and the edge loss, column b) depicts the semantic DSC and edge DSC, and COlumn c) depicts the testing semantic loss and DSC. Each row of **Figure 5** denotes the model which corresponds to the training results.

Rows 3 and 4, which depict the training results of original GSCNN model, show that regularization decreases the variability observed in the testing loss and DSC values over each step. This observation is obvious when comparing Figure 5c)-3 and 5c)-4. Because the regularization technique penalizes when there is error between the semantic prediction and the shape prediction, it can be said that regularization is helping to keep consistency between the predictions of both streams. However, despite the consistency, the edge DSC seems to be quite low compared to the semantic DSC when observing the training functions.

Rows 1 and 2 depicts the training and testing functions of the GSCNN when the UNET was used as the RegularStream with and without regularization. Contrary to the original GSCNN model, regularization seems to negatively impact the GSCNN-UNET testing loss more when compared the GSCNN-UNET without regularization. Both the training and testing loss values are higher in the GSCNN-UNET model that uses regularization. By these observations, perhaps there is contradiction in the semantic and shape predictions. Because regularization attempts to keep these consistent, it is likely that for this particular architecture regularization might negatively impact segmentation accuracy.

Perhaps analyzing the intermediate representations from the gates can give more insight to what the ShapeStream is learning, and how it may impact semantic segmentation. **Figure 5** illustrates the intermediate representation taken at the third gated convolution layer. It is interesting to observe that the GSCNN-UNET based models extract edge information compared to the GSCNN-OG which has a ResNet RegularStream. The UNET and ResNet encoders share the same encoding structure so it is interesting to observe different

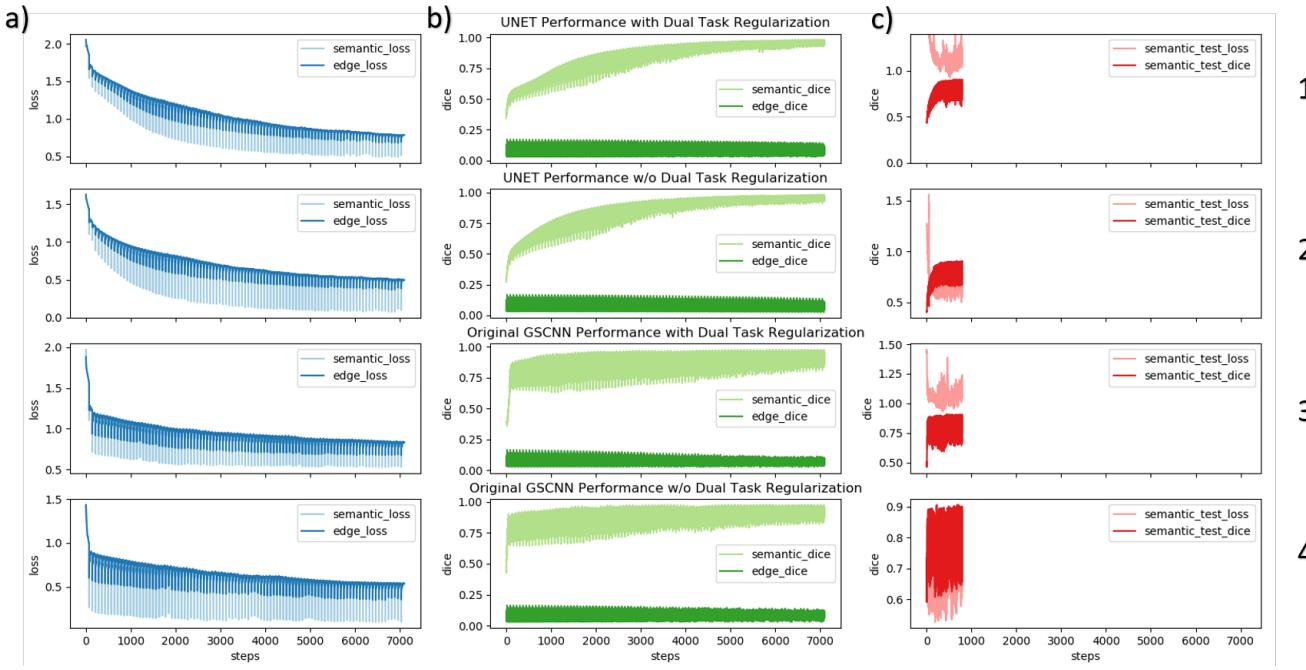


Fig. 5. Comparing GSCNN Model Training: Column a) depicts the semantic loss and the edge loss, Column b) depicts the semantic dsc and edge dsc, and Column c) depicts the testing semantic loss and dsc

representations at the same gates. Furthermore, the GSCNN-UNET based models depict bright pixels at nuclei locations whereas the GSCNN-OG models depict dark pixels at the nuclei locations. This could be an indication that for the GSCNN-OG, the ShapeStream has a inverse relationship for semantic prediction. For the GSCNN-UNET based model, perhaps edge information is actually learned as evidenced by the presence of edge information at the gates. This may be an indication of the consistency enforced by the regularizer which could be more effective with the GSCNN-UNET compared to the GSCNN-OG.

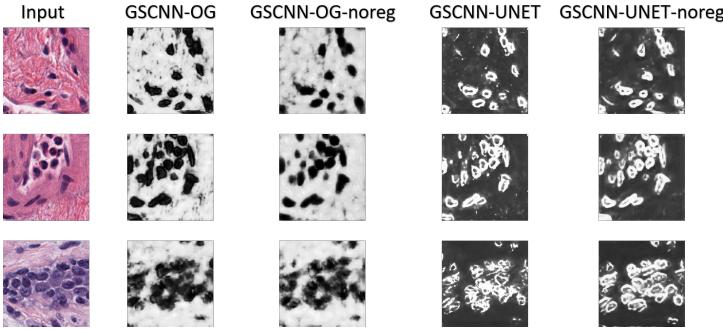


Fig. 6. Gate 3 outputs taken at the 100th epoch

2) *Predictions:* Figure 6 presents the distribution of the nuclei DSC over the testing set. The line central to each plot defines the median DSC, the box defines the interquartile range, and the points laying outside the extended lines define the outliers. Comparing each GSCNN variation, the GSCNN-

UNET without regularization (green) achieves the highest median DSC. Table 1 below summarizes the segmentation results.

TABLE I
SEGMENTATION PERFORMANCE

Method	$\mu_{median} DSC$	σDSC
GSCNN-OG	0.292	0.186
GSCNN-OG-noreg	0.415	0.194
GSCNN-UNET	0.419	0.201
GSCNN-UNET-noreg	0.537	0.198
UNET	0.710	0.150

From Figure 6 and Table 1 it is evident that among the GSCNN models, the GSCNN-UNET-noreg demonstrates the highest median nuclei DSC, followed by GSCNN-UNET, GSCNN-OG-noreg, and GSCNN-OG. Furthermore, for the boundary segmentation results the GSCNN-UNET-noreg demonstrated the highest median boundary DSC followed by GSCNN-OG-reg, GSCNN-OG-noreg, GSCNN-UNET-reg. Overall, using the UNET as the RegularStream without regularization demonstrated a performance increase compared to the vanilla UNET model. However, in comparison to the vanilla UNET model, the GSCNN-UNET models perform subpar.

3) *Segmentation Prediction:* The predictions in Figure 8 are formed from applying a final convolution layer with sigmoid activation to the joined representation of the RegularStream and ShapeStream (Figure 9). Despite the vanilla UNET demonstrating the top DSC, among the GSCNN vari-

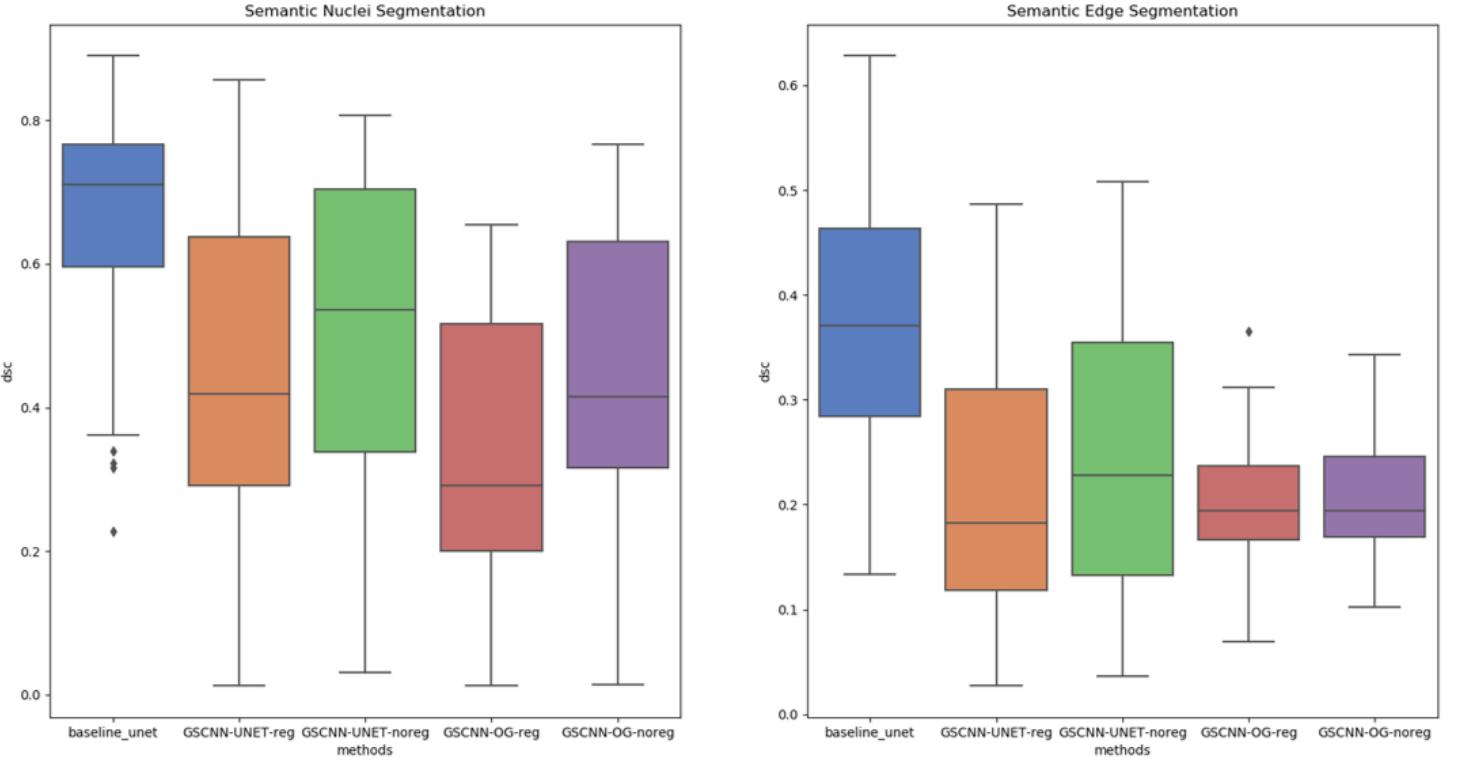


Fig. 7. Segmentation Performance of each model

tions the GSCNN-UNET models demonstrated edge preservation in the predicted images. These results demonstrate that the UNET as the RegularStream proved to be advantageous for segmenting edges. This would mean that edge preservation is likely a result of the decoding structure in the UNET architecture.

In the examples presented in **Figure 8**, certain instances actually demonstrate better edge preservation in the GSCNN-UNET predictions compared to the regular UNET predictions. For instance, in both the second and fourth examples, the edges are more prevalent. The fourth example of the UNET prediction does not contain much edge content. This is a positive observation as the goal of the ShapeStream is to improve the quality of boundary segmentation. Furthermore, since the GSCNN model which demonstrated promising boundary results is the GSCNN-UNET-noreg, it is evident that the boundary improvement is a result of the ShapeStream, and that the simple binary cross entropy loss for the edges should be suffice for training.

IV. DISCUSSIONS AND CONCLUSIONS

In this paper, the performance of several GSCNN architectures were compared for the task of nuclei segmentation. The goal of utilizing this particular architecture was to improve boundary segmentation. The original implementation boasts that the use of a second stream, a *dual-task loss*, and regularization will boost the performance of models to segment objects and their boundaries. These concepts were applied in

the task of nuclei segmentation. By segmenting boundaries effectively, nuclei will be better separated and downstream processes such as nuclei feature extraction can be improved. To evaluate the performance of the GSCNN architecture, the RegularStream was interchanged with a ResNet and the UNET models. Furthermore, experiments were carried out using the dual task loss and regularization techniques proposed by [1]. From the experiments, the GSCNN variation which demonstrated the best performance was the GSCNN-UNET without regularization. The success of the architecture is likely attributed to the decoding stream which was able to localize and segment nuclei better compared to the other GSCNN variations. The original implementation, GSCNN-OG, demonstrated improved performance in natural images (self driving cars) compared to its performance for nuclei segmentation. This could be a result of the *gross* nature of natural images. Nuclei edges, texture, and boundaries are finer, and contain great class imbalance. Furthermore, the bi-linear transform method for up-sampling the dense features may be too aggressive, and may not allow the network to learn the fine edges for the output. This is demonstrated by the lack of edge information in the GSCNN-OG predictions. The GSCNN-UNET, however, performs up-sampling gradually through the multiple decoder blocks. Furthermore, the skip connections from the encoding stream allow intermediate abstractions to be included in the output. From Figure 8, neither of the GSCNN-OG models have prominent edges in the prediction, whereas the GSCNN-UNET variations do.

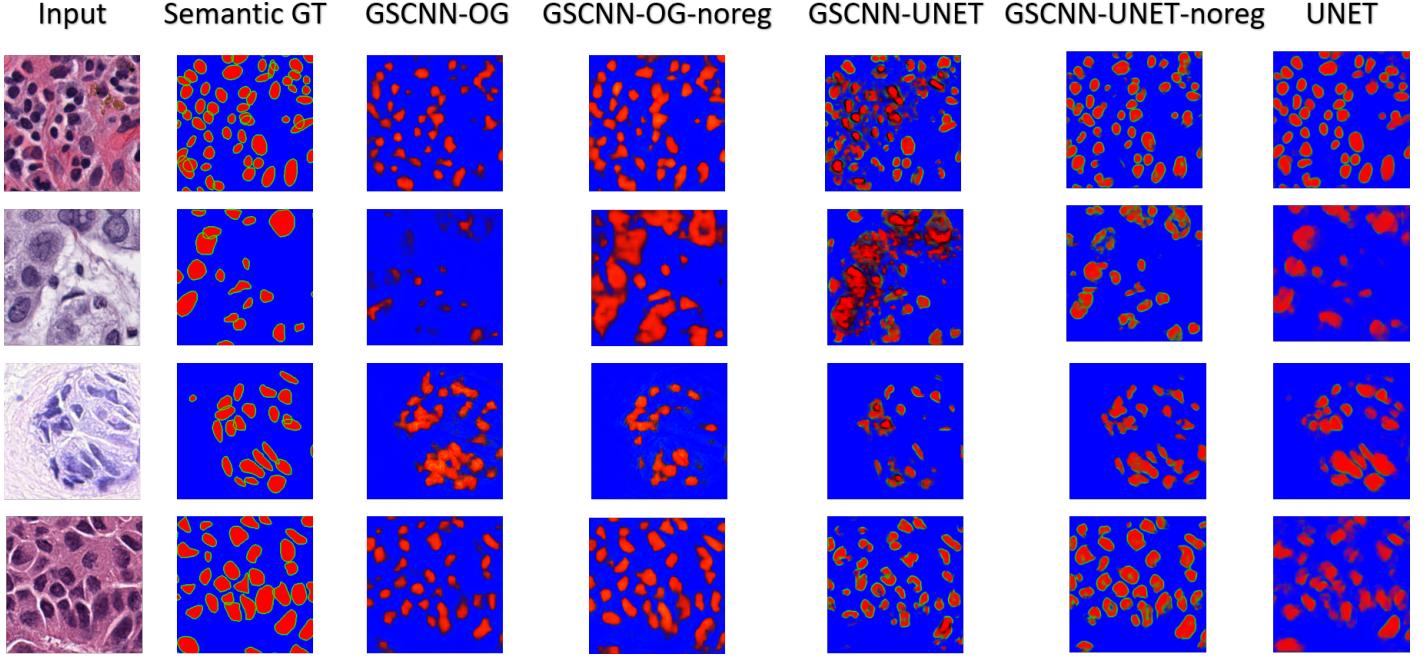


Fig. 8. Qualitative Segmentation Results - Semantic Prediction

From the experiments carried out in this paper, regularization seemed to negatively impact performance, rather than benefit it. In both cases of the GSCNN-OG and the GSCNN-UNET, the models with regularization demonstrated sub-par performance. Because the regularization technique attempts to keep the predictions of the RegularStream and ShapeStream consistent, the decrease in performance could be a result of contradicting predictions which affect the loss and how the gradients update. As demonstrated by the loss functions, while the regularization improved the test loss for the GSCNN-OG model, regularization negatively impacted the learning of the GSCNN-UNET. Despite these observations, **Figure 9**, which depicts predictions by the ShapeStream, demonstrate that regularization for the GSCNN-UNET generates decent gradient predictions. Compared to the GSCNN-UNET, the other methods do not predict the boundaries. Instead, it seems as though whole nuclei are segmented. Further research is needed to understand if regularization is the definite cause of these observations.

For future works, more research is needed to tune the parameters of the ShapeStream and weights of the loss function. Number of filters and number of intermediate representations are examples of parameters that could be optimized. In addition, further research can be done on building a similar architecture where edge maps guide the final semantic segmentation. While the proposed architecture does not out perform the UNET baseline, a greater understanding of architecture design was attained. Lastly, new insights of how traditional image processing methods (image gradients) can be leveraged in deep learning were achieved.

REFERENCES

- [1] Towaki Takikawa, David Acuna, Varun Jampani, and Sanja Fidler. Gated-sconv: Gated shape cnns for semantic segmentation. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.
- [2] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.
- [3] Narinder Singh Punn and Sonali Agarwal. Inception u-net architecture for semantic segmentation to identify nuclei in microscopy cell images. *ACM Trans. Multimedia Comput. Commun. Appl.*, 16(1), February 2020.
- [4] Z. Zeng, W. Xie, Y. Zhang, and Y. Lu. Ric-unet: An improved neural network based on unet for nuclei segmentation in histology images. *IEEE Access*, 7:21420–21428, 2019.
- [5] Faisal Mahmood, Daniel Borders, Richard J. Chen, Gregory N. McKay, Kevan J. Salimian, Alexander S. Baras, and Nicholas J. Durr. Deep adversarial training for multi-organ nuclei segmentation in histopathology images. *CoRR*, abs/1810.00236, 2018.
- [6] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.
- [7] N. Kumar, R. Verma, S. Sharma, S. Bhargava, A. Vahadane, and A. Sethi. A dataset and a technique for generalized nuclear segmentation for computational pathology. *IEEE Transactions on Medical Imaging*, 36(7):1550–1560, 2017.
- [8] Aurlien Gron. *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O'Reilly Media, Inc., 1st edition, 2017.
- [9] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *CoRR*, abs/1802.02611, 2018.
- [10] Zhiding Yu, Chen Feng, Ming-Yu Liu, and Srikanth Ramalingam. Casenet: Deep category-aware semantic edge detection. *CoRR*, abs/1705.09759, 2017.
- [11] Justin Tyler Pontalba, Thomas Gwynne-Timothy, Ephraim David, Kiran Jakate, Dimitrios Androutsos, and April Khademi. Assessing the impact of color normalization in convolutional neural network-based nuclei segmentation frameworks. *Frontiers in Bioengineering and Biotechnology*, 7:300, 2019.

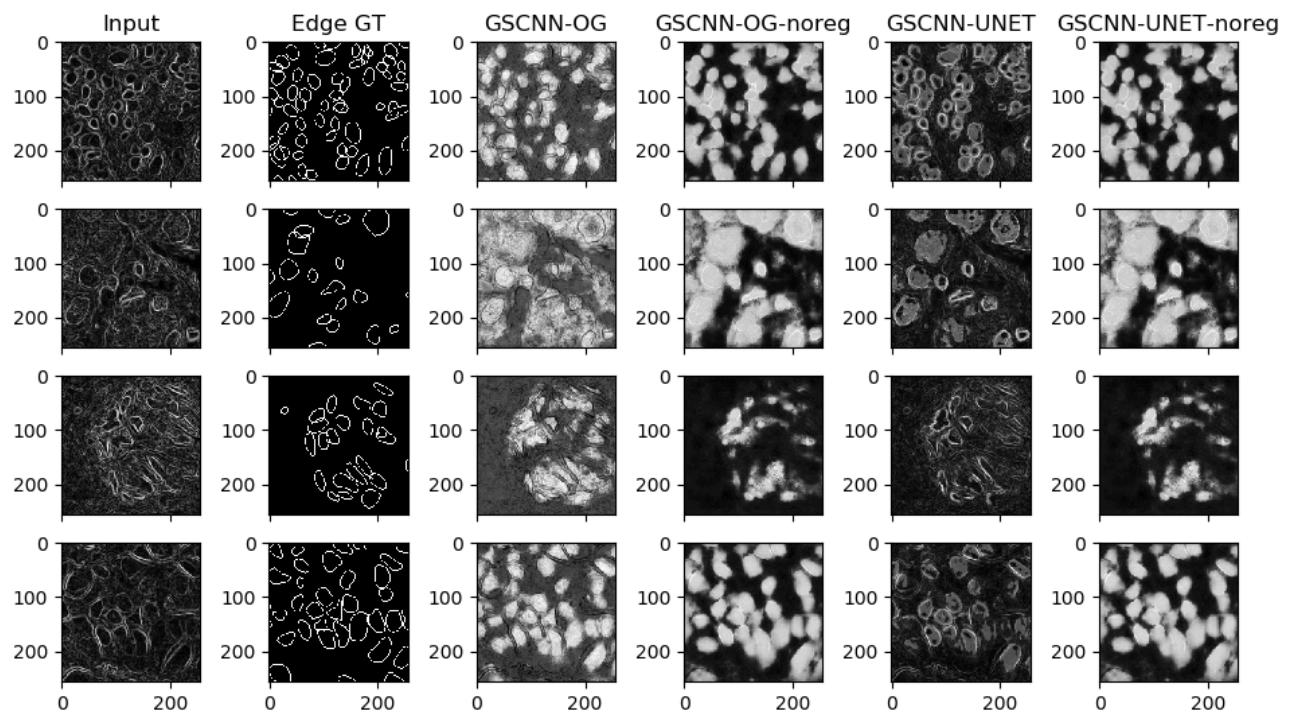


Fig. 9. Qualitative Segmentation Results - Edge Prediction