Carlos Henrique Caloete Pena

Attention-based Image Segmentation

Universidade Federal de Pernambuco
graduacao@cin.ufpe.br
www.cin.ufpe.br/~graduacao

Recife
2019
Carlos Henrique Caloete Pena

Attention-based Image Segmentation

A B.Sc. Dissertation presented to the Centro de Informática of Universidade Federal de Pernambuco in partial fulfillment of the requirements for the degree of Bachelor in Computer Engineering.

Concentration Area: Computational Intelligence
Advisor: Tsang Ing Ren

Recife
2019
ACKNOWLEDGEMENTS

Not only in this work, but throughout my undergraduate degree, I needed a lot of motivation and a lot of teaching to get through each step. Therefore, I want to thank everyone who helped me in these moments. I thank my family for providing all the necessary support. To the Professors and Doctors, Tsang Ren, Pedro Diamel, Fidel Pena, Alexandre Cunha, and George Darmiton, for the many lessons. To my project teammates, Lucas Cavalcanti, Heitor Rapela, Cristiano Santos, Roberto Fernades, To my friends who helped in the revision of this thesis: Marcus Vinicius, Mariana Barros, Leonardo Alves, Víctor Sabino, Matheus Branco, and Amanda Lasserre. To pets, Bela, Fliper, Romeu, Juju, Olivia, Belinha, Kiara, Marie, Miau, Ariel and Chico. Finally, I thank the groups of the CIn, E.S.T.U.F.A and, especially, the entire RobôCIn team, especially its teachers: Edna Barros, Hansenclever Bassani, Tsang Ren, Paulo Salgado, and Pedro Braga. And to the OKI Recife laboratory.
“And then for your soul, for your life
I will pray every night until life carries on.”

–Terra Prima
In the image processing field, image segmentation is one of the most complex scenarios, widely explored over the years. There are several applications to this field; one of them is the segmentation of medical images, more specifically, in microscopic images. Despite efforts, the results have not been successful yet. On the other hand, ensemble learning is widely used mainly in problems involving classification, and regression tasks when individual classifiers or regressors do not achieve the desired results. However, it still needs some research to use Ensemble Learning with image segmentation, especially in nuclear panoptic segmentation, where the main objective is to segment each cell nucleus in a different instance without overlapping. This project proposes a framework to use multiple segmentation networks with a deep neural attention network to improve results on nuclei panoptic segmentation tasks. To evaluate the proposed framework, we compare the results in seven metrics, with five monolithic models and with the non-trainable ensemble of those five models using seven aggregation functions. The experiments show that the proposed framework overcomes the results of a set of individuals segmentation network and a set of non-trainable ensemble using different aggregation functions.

**Keywords:** Machine Learning. Image Segmentation. Panoptic Segmentation.
RESUMO

No campo de processamento de imagens, a segmentação de imagens é uma das tarefas mais complexas e amplamente exploradas ao longo dos anos. Existem várias aplicações para esse campo; uma delas é a segmentação de imagens médicas, mais especificamente, imagens microscópicas, em que apesar dos esforços, os resultados ainda não foram bem-sucedidos. Por outro lado, a aprendizagem por ensemble é amplamente usada, principalmente em problemas que envolvem classificação e tarefas de regressão, quando classificadores ou regressores monolíticos não obtêm os resultados desejados. No entanto, ainda é necessária alguma pesquisa para usar o ensemble learning com segmentação de imagem, especialmente na segmentação panóptica de núcleos celulares, onde o objetivo principal é segmentar cada núcleo celular em uma instância diferente, sem sobreposição. Este projeto propõe uma estrutura para usar várias redes de segmentação com uma rede neural profunda de atenção para melhorar os resultados nas tarefas de segmentação panóptica de núcleos celulares. Para avaliar a estrutura proposta, compararamos os resultados em sete métricas, com cinco modelos monolíticos e com o ensemble das cinco redes monolíticas usando sete funções de agregação estática. Os resultados mostram que a estrutura proposta supera os resultados de um conjunto de redes de segmentação monolíticas e de um conjunto de ensemble não treináveis utilizando diferentes funções de agregação.

Palavras-chave: Aprendizagem de máquina. Segmentação de imagens. Segmentação panóptica
LIST OF FIGURES

Figure 1 – Comparison between cancerous cells and normal cells. Source: Lynne Eldridge (2019) ................................. 12

Figure 2 – Example of microscopic biological images. Source Coelho et al. (2009) 12

Figure 3 – A copper segmentation example where (a) is the grayscale original image, and two segmentation using threshold values (b) from the mean pixel (value of 161.1) (c) from Otsu’s method (value of 127.0) ................. 14

Figure 4 – Segmentation of Figure 3a with the K-Means algorithm where 4a, 4b, 4c have K value equal to 2, 3, 4 reciprocally. ......................... 15

Figure 5 – The U-Net architecture. A convolutional encoder-decoder, where the encoder or contracting path is at the left side, and the decoder or expansive path at the right side. Source Ronneberger et al. (2015) ................. 16

Figure 6 – A building block for ResNet101. In this block, every other three convolutional layers have a skip connection. Source He et al. (2016) ................. 16

Figure 7 – The original image and example of data augmentation ............................. 18

Figure 8 – Project pipeline ................................................. 20

Figure 9 – Example of output F function, using weight arithmetic mean ................. 21

Figure 10 – Division of dataset 2DNuclei ................................................. 22

Figure 11 – The network input (a), and desired output (b). ................................................. 24

Figure 12 – Toy problem, with (a) ground truth and two proposed segmentation: (b) that archives an accuracy of 0.998 and a PQ of 0.351; and (c) with an accuracy of 0.864 and PQ of 0.770. ................................. 26

Figure 13 – Segmentation done by every monolithic model inside PoolNet, where black and white is the correct segmentation, and the colors is the cells predictions. ................................................. 29

Figure 14 – Segmentation done by static aggregation, where black and white is the correct segmentation, and the colors is the cells predictions. ......................... 30

Figure 15 – Segmentation done by the hypothetical models, Oracle and Best Model, where black and white is the correct segmentation, and the colors is the cells predictions. ................................................. 31

Figure 16 – Segmentation done by the proposed framework, where black and white is the correct segmentation, and the colors is the cells predictions. ......................... 32
# LIST OF TABLES

Table 1  –  Results with a single model in the test set with ten images.  

Table 2  –  Metrics using monolithic models, classic aggregation, Oracle and the proposed framework on test set with ten images.
<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN</td>
<td>Convolutions Neural Networks</td>
</tr>
<tr>
<td>FCN</td>
<td>Fully Connected Network</td>
</tr>
<tr>
<td>IOU</td>
<td>Intersection Over Union</td>
</tr>
<tr>
<td>IOUT</td>
<td>Mean Average Precision at Different Intersection Over Union Thresholds</td>
</tr>
<tr>
<td>PQ</td>
<td>Panotic Quality</td>
</tr>
<tr>
<td>PSO</td>
<td>Particle Swarm Optimization</td>
</tr>
</tbody>
</table>
LIST OF ALGORITHMS

Algorithm 1  –  K-Means for image segmentation . . . . . . . . . . . . . . . . . . . 15
Algorithm 2  –  Image data augmentation with contrast, and brightness . . . . . . . 17
Algorithm 3  –  Image data augmentation with gamma . . . . . . . . . . . . . . . . . . . 17
## CONTENTS

1 INTRODUCTION .................................................. 11
1.1 OBJECTIVE .................................................. 12
1.2 ORGANIZATION OF THE MONOGRAPHY ...................... 13

2 LITERATURE REVIEW ............................................. 14
2.1 IMAGE SEGMENTATION .......................................... 14
2.2 SEGMENTATION WITH DEEP LEARNING ......................... 15
2.3 DATA AUGMENTATION ........................................... 17
2.4 PANOPTIC SEGMENTATION ...................................... 18
2.5 ENSEMBLE WITH SEGMENTATION ................................ 19
2.6 ORACLE AND BEST MODEL ..................................... 19

3 METODOLOGY ..................................................... 20
3.1 FRAMEWORK .................................................. 20
3.1.1 PoolNet ................................................... 20
3.1.2 Expert ..................................................... 21
3.1.3 The F function ............................................... 21
3.1.3.1 Non-Trainable ........................................ 21
3.1.3.2 Trainable ............................................... 22
3.2 DATASET DIVISION ............................................ 22

4 EXPERIMENTS ................................................... 24
4.1 METRICS ..................................................... 24
4.2 SEGMENTATION WITH MONOLITHIC NETWORK ............. 25
4.3 AGGREGATION WITH NON-TRAINABLE SELECTION ........ 27
4.4 ORACLE AND BEST MODEL ..................................... 27
4.5 PROPOSED FRAMEWORK ...................................... 27

5 CONCLUSION AND FUTURE WORKS ............................... 33
REFERENCES .......................................................... 34
INTRODUCTION

Image segmentation is a way to subdivide an image into regions or objects that compose it (Gonzalez & Woods, 2006), where the pixels that describe the same features have the same label. The regions or the objects are a meaningful simplification of the image (Kaur & Kaur, 2014) that could be used, for instance, to reduce the search area for an object detection algorithm.

There are several approaches for image segmentation; these approaches focus on analyzing two factors of the pixel intensity: the discontinuity and the similarity (Gonzalez & Woods, 2006); for instance, the threshold methods, such as mean value and otsu’s (Otsu, 1979) method, that can produce great results for a simple two-classes segmentation task with low computational cost. Another way to segment is by using a clustering algorithm such as K-Means (MacQueen et al., 1967) or use a watershed-based algorithm (Beucher et al., 1992). Nevertheless, later results such as Kaggle (Data Science Bowl, 2018a) show us that for a sophisticated segmentation task, deep neural networks, such as U-Net (Ronneberger et al., 2015) and Mask-RCNN (He et al., 2017), outperform the classical methods (Caicedo et al., 2019).

Since the creation of U-Net (Ronneberger et al., 2015), a fully convolutional network, the field of image segmentation had a significant advance, especially in microscopic images. In microscopic images, there are two problems: first, a lack of images and then a lack of handmade ground truth done by a specialist. The U-Net addresses this issue using data augmentation, such as elastic transformation, and with the proposed u-shaped architecture with long skip connections (Drozdzal et al., 2016).

The image segmentation is an essential step for many automatized tasks; for instance, nuclear segmentation is helpful for cell counting, analysis of the cell-division cycle, and analysis of the growth of cancer cells. As Lynne Eldridge (2019) shows, a cancer cell has a different size, shape, and boundary in comparison to non-cancerous cells; moreover, a cancer cell tends to continue growing and reproducing along the time. These characteristics could be extracted from an image or a sequence of images with a good segmentation algorithm; an illustrated comparison is shown in Figure 1.

In this work, we focus on microscopic biological images (Coelho et al., 2009), as shown in Figure 2. This dataset is a collection of 48 images of U2OS cells, these cells are present on the osteosarcoma cancer that occurs on a human bone. In those images, there are two main
Figure 1: Comparison between cancerous cells and normal cells. Source: Lynne Eldridge (2019)

tasks: first, distinguish between cells and not cells, and then split one cell from the other without overlapping. Furthermore, there are three obstacles: first, the presence of multiple cells stuck together, the presence of noise caused by residual luminescence substance used to highlight the cells and annotation errors.

Figure 2: Example of microscopic biological images. Source Coelho et al. (2009)

1.1 OBJECTIVE

We propose a new framework of image segmentation fusion to outperform monolithic models and ensemble models with static aggregation. This framework is divided into three parts: a set of segmentation models such as U-Net based models, the PoolNet; an attention model, the Expert; and an aggregation function, F. We exploit two variants for the F function, a non-trainable and trainable variant.

To evaluate the proposed framework, we compare with the results of five monolithic segmentation network and with the non-trainable ensemble of those five models using seven
aggregation functions.

### 1.2 ORGANIZATION OF THE MONOGRAPHY

This rest of this monography is arranged as the following. Chapter 2 shows an overview of image segmentation and ensemble methods applied to image segmentation. The description of the proposed pipeline and each component of the framework is shown in Chapter 3. Chapter 4 presents the results and compares the framework with single models and static aggregation functions. Finally, Chapter 5 summarizes the results of the monography and highlights the principal contributions.
LITERATURE REVIEW

2.1 IMAGE SEGMENTATION

Initially, algorithms of image segmentation are based on a threshold method, for instance, the threshold value could be a static value from an image, such as the mean pixel value, or calculated from a rule that minimizes the intra-class variance as the Otsu’s method (Otsu, 1979). In these scenarios, values below and above the threshold value are assigned to the class one or two reciprocally. An essential step is to use a post-processing algorithm to filter small noise from the segmentation, such as removing from class two all objects with an area below a low limit. An example of segmentation using the mean pixel value and Otsu’s method is shown in Figure 3, where given an image of two types of coin, it is necessary to segment all the copper coins in one category, represented as white pixels. As expected, both algorithms perform a visually acceptable segmentation, even that there are many pixels incorrectly segmented, as can be seen in the borders of the silver coin.

![Figure 3: A copper segmentation example where (a) is the grayscale original image, and two segmentation using threshold values (b) from the mean pixel (value of 161.1) (c) from Otsu’s method (value of 127.0)](image)

Another way to segment images is to use a clustering algorithm such as K-Means, where every cluster represents a region. Therefore, pixels assigned to the same cluster represent the
same region; a result produced by the Algorithm 1, described below, can be seen in Figure 4. A drawback of this method is that the programmer needs to know a priori the optimal K value. Figure 4a to 4c shows a comparison of segmentation of the same input image with different K values. An important point of this K-Means algorithm is that the K-Means does not consider the spatial position of the pixel for the segmentation. To improve the K-Means algorithm, Ng et al. (2006) uses a combination of K-Means segmentation with a watershed algorithm.

Algorithm 1: K-Means for image segmentation

**Input:** Src // A gray scale image
K // Number of clusters

**Result:** Dst // A segmented image

1. Assign at random initial values for the clusters; \( m_1, m_2, \ldots, m_K \), from 0 to 255

2. **while** Not converged **do**

3. **for** each pixel \( I \) in Src **do**

4. calculate the distance from the \( I \) value to all clusters

5. Assign the pixel \( I \) to the cluster with small distance

6. **for** each cluster \( C \) **do**

7. calculate the mean of every pixel assigned to \( C \)

8. Assign to \( C \) the mean previously calculated as the new cluster value

9. **for** each pixel \( I \) in Src **do**

10. calculate the distance from the \( I \) value to all clusters

11. Assign the cluster value that has the small distance from \( I \) for the pixel in the same location in the Dst

Figure 4: Segmentation of Figure 3a with the K-Means algorithm where 4a, 4b, 4c have K value equal to 2, 3, 4 reciprocally.

2.2 SEGMENTATION WITH DEEP LEARNING

In general, segmentation models for biological images, such as U-net (Wang & Zhang, 2015) and Albunet (Wang et al., 2018), or even general proposed models such as SegNet
(Badrinarayanan et al., 2017), work similarly. The main idea is to construct a convolutional encoder-decoder architecture, as can be seen in Figure 5. In the first step, the encoder, the whole image is compressed, using operations such as convolution and pooling into a high-dimensional feature vector. After, in the decoder, this feature representation is expanded to create a likelihood map, usually with the same dimensions as the input image.

![Figure 5: The U-Net architecture. A convolutional encoder-decoder, where the encoder or contracting path is at the left side, and the decoder or expansive path at the right side. Source Ronneberger et al. (2015)](image)

On the encoder, it is common to use some well-known architectures such as ResNet (He et al., 2016) and VGG (Simonyan & Zisserman, 2014). For instance, the network UnetResNet101 is a U-Net architecture with encoder depth of 101 residual layers.

![Figure 6: A building block for ResNet101. In this block, every other three convectional layers have a skip connection. Source He et al. (2016)](image)
2.3 DATA AUGMENTATION

To a deep neural network achieve robustness is necessary a large and diverse dataset, although, in some application such as biological images is not possible to build a large dataset due to factors such as time and cost to a specialist create the ground truth. Data augmentation is used to solve this problem; in other words, apply transformations on the original data to generate synthetic data. Some examples of data augmentation in an image segmentation task are to apply at random, one or more of the following operations at the original image: crop, resize, vertical and horizontal flips, modify the image brightness, contrast, and gamma and finally elastic distortion (Simard et al., 2003) as can be seen in Figure 7. The crop and resize operation are often used to reduce the necessary memory to train and evaluate a deep neural network; also, the vertical and horizontal flips operations help the segmentation network to obtain rotational invariance features.

Unlike the others, the brightness, contrast, and gamma transformation should be only used in the original image, and not in the ground truth, the algorithm for these operations can be found at Algorithm 2 and 3 where, commonly, for each image the parameters $\alpha, \beta, and \gamma$ are randomly chosen.

**Algorithm 2:** Image data augmentation with contrast, and brightness

**Input:** Src // The original image  
$\alpha$ // Contrast factor  
$\beta$ // Brightness factor  

**Result:** Dst // An augmented image

1. for each pixel $I$ in Dst do  
2. Let $J$ be the pixel in the same location of $I$ in Src  
3. $I = J \times \alpha + \beta$

**Algorithm 3:** Image data augmentation with gamma

**Input:** Src // The original image  
$\gamma$ // Gamma factor  

**Result:** Dst // An augmented image

1. Normalize Src in the range 0.0 to 1.0  
2. for each pixel $I$ in Dst do  
3. Let $J$ be the pixel in the same location of $I$ in Src  
4. $I = J^{\gamma}$  
5. Normalize Dst in the range 0 to 255
2.4 PANOPTIC SEGMENTATION

The panoptic segmentation is a segmentation task introduced by Kirillov et al. (2019); it is a task where each pixel on the segmentation output has one and just one semantic label (for instance, a car, a tree or a bicycle), and one and just one instance ID. The representation of an object is formed by every pixel that contains the same label and ID values; as a result of the definition is not possible to have overlapping between objects.

As panoptic segmentation needs a pair (a semantic label, and an instance ID), some networks such as U-Net do not provide a panoptic segmentation output as default. In this case, it is necessary to convert the U-Net segmentation in the post-processing. In general, for the semantic label it is used the same semantic label as proposed by the network; for the instance ID, it is necessary to decompose the segmentation in connected components, where every connected component with the same segmentation label has an unique ID.
2.5 ENSEMBLE WITH SEGMENTATION

Wang & Zhang (2015) uses a different way to ensemble segmentations, called Cluster Ensemble using two clustering techniques; K-means and Nyström Spectral Clustering; they focus on a high-performance algorithm with a low cost of memory.

Moreover, Tan et al. (2019) introduces two types of ensemble: first, an ensemble hybrid clustering model, and second, an ensemble deep segmentation network for skin lesion segmentation. In the ensemble hybrid clustering model, the authors trained three Fuzzy C-Means and aggregated them with a pixelwise majority vote. In the ensemble deep segmentation network, the same procedure was used to train three Convolutions Neural Networks (CNN) and aggregate them. To make the centroids of Fuzzy C-Means more robust to noise and local optimum, and to reduce computational cost on finding the CNN hyper-parameters such as the learning rate and the weight decal, the authors use a Particle Swarm Optimization (PSO) (Kennedy & Eberhart, 1995) based algorithm. This PSO based algorithm uses a mixture of optimization algorithms such as Simulated Annealing (Kirkpatrick et al., 1983), Differential Evolution (Storn & Price, 1997), PSO, and Levy search.

2.6 ORACLE AND BEST MODEL

Originally in Mult-Classifier Systems literature, the Oracle is a hypothetical model that always predicts the correct label if one or more classifier predicts correctly. We adjust this definition to the segmentation field. The Oracle is a hypothetical model that segments every pixel correctly if one or more model segment the same pixel correctly. Also, we defined another hypothetical model, the Best Model, as a model that, for every image, predicts a segmentation as good as the best segmentation on the pool. The Best Model can be seen as an Oracle that, instead of predicting pixelwise, predicts for the whole picture.
3

METODOLOGY

First, Section 3.1 presents the proposed framework and its three key components: the PoolNet 3.1.1; the Expert network 3.1.2; and exploit the F variants 3.1.3. Later, Section 3.2 shows the dataset division.

3.1 FRAMEWORK

We defined a framework to combine results from a group of segmentation models into a new and better segmentation. Figure 8 shows the top flowchart of the project. First, given an image \( x \), from the ImageData set, it is passed through every network inside PoolNet and it is obtained its likelihood map for each network. The F function \( R^{N \times H \times W} \rightarrow R^{H \times W} \) combined those likelihood maps and weights that were obtained from an attention network Expert \( E_\theta \); this combination is the new likelihood map proposed in this monography.

![Figure 8: Project pipeline](image)

3.1.1 PoolNet

The PoolNet is a pool of segmentation models; for this monography, we use five different networks, that are UNetresNet, UNetvgg, AlbuNet, SegNet, and SimplestNet, where SimplestNet is a fully convolutional network with five convolutional layers. Every network is trained with the same subset of microscope cell images.
3.1.2 Expert

The expert network, UnetResNet, is trained to highlight regions where there is the highest probability to be correct for each segmentation available on PoolNet, resulting in a segmentation map with the same size of PoolNet’s likelihood map.

3.1.3 The F function

The F function receives two pieces of information: first, several likelihood maps from PoolNet, and second, an attention map from the Expert network; The F function combines that information into a single likelihood map. The combination could be trainable or non-trainable. The following section shows several variants to combine that information.

3.1.3.1 Non-Trainable

A Non-trainable function could be any static differentiable function that could transform $R^{N \times H \times W}$ into $R^{H \times W}$, where $N$ is the number of likelihood map, $H$ is the image height and $W$ the image width. For example, summation function, weight arithmetic mean and median.

An example is the weighted arithmetic mean from the likelihood map from PoolNet, as shown in Equation 3.1, where the weight $E'_{\theta}$ is represented as the normalized output of the Expert network $E_{\theta}$, as can be seen in Equation 3.2. In our experiments, other normalization methods such as softmax or sigmoid function to preprocessing the weights degraded the performance.

$$F(S_p, E_{\theta}) = \sum_{n} (S_p[n] \times E'_{\theta}[n]) \quad \text{(3.1)}$$

$$E'_{\theta} = \frac{E_{\theta}}{\sum_{n} E_{\theta}[n]} \quad \text{(3.2)}$$

Figure 9: Example of output F function, using weight arithmetic mean
3.1.3.2 Trainable

Instead of defining a traditional math function, on this variant, the Gate method, we decided to use a small neural network, $F_{net}$, to perform this function. For that, we first concatenate $S_p$ and $E_\theta$ and pass forward to this network. In this method, the final likelihood map is the network output.

$$F(S_p, E_\theta) = F_{net}(cat(S_p, E_\theta))$$ \hspace{1cm} (3.3)

Similar to Gate, GateX uses the same network, $F_{Net}$, but also adds to the concatenation function the input image $X$, as can be seen in Equation 3.4.

$$F(S_p, E_\theta) = F_{net}(cat(S_p, E_\theta, X'))$$ \hspace{1cm} (3.4)

3.2 DATASET DIVISION

The dataset of 2D nuclear cell images segmentation in a microscope (Coelho et al., 2009) was divided into three subsets: the train, validation, and test set; both train and validation with 19 original images and the test set with 10 original images. Figure 10 illustrates the dataset division. After that, there are applied some image transformations like resizing, random crop, mirroring, and elastic distortion. Furthermore, there are applied other operations, like: change with the mean value of brightness, contrast, and gamma of the input image, as shown in Section 2.3, resulting in 10,000 images in both sets. For the test set, it was applied a fixed resize and central crop to preserve the same spatial relation on the training and validation set.

Figure 10: Division of dataset 2DNuclei
Every model in PoolNet is trained with the training set and uses the weight values that performed better on the validation set. After that, all metrics are calculated with the test set. On the other hand, the Expert was trained with a different subset of the nuclear image, the validation set, in comparison to the PoolNet models, to prevent overfitting.
EXPERIMENTS

This chapter is presented in the following way. First, Section 4.1 explains the metrics used to evaluate those segmentation models. Section 4.2 shows the results with a single model. Section 4.3 presents a comparison with classical non-trainable aggregation techniques, and Section 4.4 shows the results with Oracle and Best Model that will be used as an approximation of upper bounding value. Moreover, in Section 4.5, we expose results with this proposed framework. To compare all the experiments, we choose seven metrics, as can be seen in Table 2.

Figure 11 shows the input image and the correct segmentation. In this chapter, this background image is used as a visual comparison of all models presented in this project.

![Fluorescence microscopy image of cells DNA nucleus](image1)

![Handmade ground truth of (a)](image2)

Figure 11: The network input (a), and desired output (b).

4.1 METRICS

In this experiment, seven metrics were used to evaluate each segmentation: Accuracy (Equation 4.1), Precision (Equation 4.2), Recall (Equation 4.3), F-score (Equation 4.4), Intersection Over Union (IOU) (Equation 4.5), Mean Average Precision at Different Intersection Over Union Thresholds (IOUT) (Data Science Bowl, 2018b) (Equation 4.6), and Panotic Quality (PQ) (Kirillov et al., 2019) (Equation 4.7). Where TP is True Positive, FP is False Positive, TN is
True Negative, and FN is False Negative. Each metric gives a score that varies from zero to one, where one is the best case.

\[
\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}
\]

\[
\text{precision} = \frac{TP}{TP + FP} \tag{4.2}
\]

\[
\text{recall} = \frac{TP}{TP + FN} \tag{4.3}
\]

\[
\text{F-score} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{4.4}
\]

IOU is the intersection of the predicted and real segmentation over the union of both, as can be seen in Equation 4.5, where A and B are two arbitrary segmentations. The \( \cap \) denotes the intersection operator and \( \cup \), the union operator.

The IOU is the Mean Average Precision, where the segmentation is just considered correct if the segmentation has an IOU higher than a threshold value, as shown in Equation 4.6. This threshold value, \( t \), ranges from 0.5 to 0.9 with steps of 0.05.

\[
\text{IOU}(A,B) = \frac{|A \cap B|}{|A \cup B|} \tag{4.5}
\]

\[
\frac{1}{|\text{thresholds}|} \sum_{t} \frac{TP(t)}{TP(t) + FP(t) + FN(t)} \tag{4.6}
\]

Moreover, PQ can be summarized as the IOU mean of match segments penalized by non-match segments, as can be seen in Equation 4.7. We decide to use the metric PQ as the primary metric for discussion.

\[
PQ = \frac{\sum_{(A,B) \in TP} \text{IOU}(A,B)}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|} \tag{4.7}
\]

An example of PQ can be found in Figure 12, where it shows two hypothetical segmentation and its metrics. In the first proposal segmentation, Figure 12b, there is an incorrect connection between the two cells, where the PQ metric highly penalizes it. Moreover, in the second segmentation, Figure 12c the hypothetical model predicts a cell larger than it should be. Even though the first proposal makes fewer mistakes compared to the second, it is heavily penalized for connecting two cells.

### 4.2 SEGMENTATION WITH MONOLITHIC NETWORK

Initially, we tested eight different deep neural networks, using the varying parameters, such as loss function, learning rate, and optimizer. After training, we calculated several metrics from the test set, as can be seen in Table 1.
Figure 12: Toy problem, with (a) ground truth and two proposed segmentation: (b) that archives an accuracy of 0.998 and a PQ of 0.351; and (c) with an accuracy of 0.864 and PQ of 0.770.

Table 1: Results with a single model in the test set with ten images.

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>IOU</th>
<th>IOUT</th>
<th>Precision</th>
<th>Recall</th>
<th>F score</th>
<th>PQ</th>
</tr>
</thead>
<tbody>
<tr>
<td>UnetResnet101</td>
<td>0.987 ±</td>
<td>0.860 ±</td>
<td>0.735 ±</td>
<td>0.968 ±</td>
<td>0.980 ±</td>
<td>0.974 ±</td>
<td>0.837 ±</td>
</tr>
<tr>
<td></td>
<td>0.003</td>
<td>0.062</td>
<td>0.092</td>
<td>0.014</td>
<td>0.010</td>
<td>0.004</td>
<td>0.063</td>
</tr>
<tr>
<td>UnetVgg</td>
<td>0.963 ±</td>
<td>0.735 ±</td>
<td>0.292 ±</td>
<td>0.866 ±</td>
<td>0.999 ±</td>
<td>0.928 ±</td>
<td>0.470 ±</td>
</tr>
<tr>
<td></td>
<td>0.010</td>
<td>0.110</td>
<td>0.092</td>
<td>0.025</td>
<td>0.001</td>
<td>0.014</td>
<td>0.111</td>
</tr>
<tr>
<td>Albunet</td>
<td>0.986 ±</td>
<td>0.791 ±</td>
<td>0.635 ±</td>
<td>0.957 ±</td>
<td>0.988 ±</td>
<td>0.972 ±</td>
<td>0.767 ±</td>
</tr>
<tr>
<td></td>
<td>0.005</td>
<td>0.109</td>
<td>0.193</td>
<td>0.019</td>
<td>0.007</td>
<td>0.009</td>
<td>0.129</td>
</tr>
<tr>
<td>Segnet</td>
<td>0.906 ±</td>
<td>0.471 ±</td>
<td>0.192 ±</td>
<td>0.725 ±</td>
<td>0.984 ±</td>
<td>0.835 ±</td>
<td>0.389 ±</td>
</tr>
<tr>
<td></td>
<td>0.030</td>
<td>0.146</td>
<td>0.104</td>
<td>0.023</td>
<td>0.013</td>
<td>0.017</td>
<td>0.152</td>
</tr>
<tr>
<td>SSectnet</td>
<td>0.908 ±</td>
<td>0.596 ±</td>
<td>0.035 ±</td>
<td>0.965 ±</td>
<td>0.643 ±</td>
<td>0.771 ±</td>
<td>0.108 ±</td>
</tr>
<tr>
<td></td>
<td>0.027</td>
<td>0.031</td>
<td>0.032</td>
<td>0.018</td>
<td>0.044</td>
<td>0.031</td>
<td>0.065</td>
</tr>
<tr>
<td>UnetResnet34</td>
<td>0.987 ±</td>
<td>0.820 ±</td>
<td>0.670 ±</td>
<td>0.972 ±</td>
<td>0.976 ±</td>
<td>0.974 ±</td>
<td>0.745 ±</td>
</tr>
<tr>
<td></td>
<td>0.005</td>
<td>0.088</td>
<td>0.162</td>
<td>0.018</td>
<td>0.012</td>
<td>0.006</td>
<td>0.137</td>
</tr>
<tr>
<td>LinkNet34</td>
<td>0.225 ±</td>
<td>0.172 ±</td>
<td>0.001 ±</td>
<td>0.224 ±</td>
<td>0.887 ±</td>
<td>0.351 ±</td>
<td>0.001 ±</td>
</tr>
<tr>
<td></td>
<td>0.069</td>
<td>0.083</td>
<td>0.001</td>
<td>0.072</td>
<td>0.034</td>
<td>0.090</td>
<td>0.001</td>
</tr>
<tr>
<td>FNC34</td>
<td>0.490 ±</td>
<td>0.017 ±</td>
<td>0.003 ±</td>
<td>0.283 ±</td>
<td>0.762 ±</td>
<td>0.410 ±</td>
<td>0.007 ±</td>
</tr>
<tr>
<td></td>
<td>0.059</td>
<td>0.013</td>
<td>0.008</td>
<td>0.067</td>
<td>0.077</td>
<td>0.079</td>
<td>0.017</td>
</tr>
</tbody>
</table>

In this initial test, two networks, Fully Connected Network (FCN) and LinkNet did not obtained an accuracy greater than 0.9, and therefore, they were removed on the next steps. Also, based on visual results, we decided to remove the UnetResnet34 for the next steps due to similar results to UnetResnet101.

Table 2, inside monolithic model type, shows the metrics of the five selected networks. A visual comparison is presented in Figure 13.
4.3 AGGREGATION WITH NON-TRAINABLE SELECTION

Seven different non-trainable aggregation functions were chosen. In these experiments, Majority Vote outperforms the other functions in 5 out of 7 metrics, including the primary metric PQ, which achieves 0.843.

4.4 ORACLE AND BEST MODEL

As explained in Section 2.6, the definition of Best Model is: given an image $x$, if we look at all segmentations from models inside PoolNet, it will always choose the network that predicts more similar to the real segmentation. If we change this definition to a pixelwise perspective, it is the definition of Oracle. The results using Oracle are used as an upper bounding approximation value. In fact, in this scenario it is possible, due to a combination of values, to overcome this value, but as shown on Table 2, inside Oracle type, on PQ metric the Oracle could predict with 0.931, so there is no much room for improvements. Figure 15 shows these visual results.

4.5 PROPOSED FRAMEWORK

This Section shows the results using our framework and its several variants, as described in Section 3.1.3. There are eight variants, among which five are non-trainable (Multiplication, Minimum, Maximum, Mean, Median) and three trainable (Gate using SimplestNet and UnetPad, GateX using SimplestNet).

All variants of the proposed model, whether trainable or non-trainable, achieved superior results in IOU, IOUT, and PQ metrics compared to monolithic models; however, the best method for the Panoptic Segmentation task is Median, which achieves 0.908 at the PQ metric; in fact, the Gate using UnetPad achieves similar results, but since the Median aggregation requires less computational power, we choose this approach as the best method for this dataset. The result of 0.908 at PQ achieves highest values compared to Majority Vote and Best Model, as shown in Table 2. In this experiment, none of the segmentation models achieves results superior to the Oracle.
Table 2: Metrics using monolithic models, classic aggregation, Oracle and the proposed framework on test set with ten images

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy</th>
<th>IOU</th>
<th>OIU</th>
<th>Precision</th>
<th>Recall</th>
<th>F Score</th>
<th>PQ</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Monolithic Model</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>UnetResnet</td>
<td>0.987 ±</td>
<td>0.860 ±</td>
<td>0.735 ±</td>
<td>0.968 ±</td>
<td>0.980 ±</td>
<td>0.974 ±</td>
<td>0.837 ±</td>
</tr>
<tr>
<td></td>
<td>0.003</td>
<td>0.062</td>
<td>0.092</td>
<td>0.014</td>
<td>0.010</td>
<td>0.004</td>
<td>0.063</td>
</tr>
<tr>
<td>UnetVgg</td>
<td>0.965 ±</td>
<td>0.735 ±</td>
<td>0.292 ±</td>
<td>0.866 ±</td>
<td>0.999 ±</td>
<td>0.928 ±</td>
<td>0.470 ±</td>
</tr>
<tr>
<td></td>
<td>0.010</td>
<td>0.110</td>
<td>0.092</td>
<td>0.025</td>
<td>0.001</td>
<td>0.014</td>
<td>0.111</td>
</tr>
<tr>
<td>Albunet</td>
<td>0.986 ±</td>
<td>0.791 ±</td>
<td>0.635 ±</td>
<td>0.957 ±</td>
<td>0.988 ±</td>
<td>0.972 ±</td>
<td>0.767 ±</td>
</tr>
<tr>
<td></td>
<td>0.005</td>
<td>0.109</td>
<td>0.193</td>
<td>0.019</td>
<td>0.007</td>
<td>0.009</td>
<td>0.129</td>
</tr>
<tr>
<td>Segnet</td>
<td>0.906 ±</td>
<td>0.471 ±</td>
<td>0.192 ±</td>
<td>0.725 ±</td>
<td>0.984 ±</td>
<td>0.835 ±</td>
<td>0.389 ±</td>
</tr>
<tr>
<td></td>
<td>0.030</td>
<td>0.146</td>
<td>0.104</td>
<td>0.023</td>
<td>0.013</td>
<td>0.017</td>
<td>0.152</td>
</tr>
<tr>
<td>SimplestNet</td>
<td>0.908 ±</td>
<td>0.596 ±</td>
<td>0.035 ±</td>
<td>0.965 ±</td>
<td>0.643 ±</td>
<td>0.771 ±</td>
<td>0.108 ±</td>
</tr>
<tr>
<td></td>
<td>0.027</td>
<td>0.031</td>
<td>0.032</td>
<td>0.018</td>
<td>0.044</td>
<td>0.031</td>
<td>0.065</td>
</tr>
<tr>
<td><strong>Classical Aggregation</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mean Rule</td>
<td>0.985 ±</td>
<td>0.881 ±</td>
<td>0.601 ±</td>
<td>0.949 ±</td>
<td>0.991 ±</td>
<td>0.969 ±</td>
<td>0.744 ±</td>
</tr>
<tr>
<td></td>
<td>0.005</td>
<td>0.049</td>
<td>0.089</td>
<td>0.021</td>
<td>0.005</td>
<td>0.010</td>
<td>0.069</td>
</tr>
<tr>
<td>Median Rule</td>
<td>0.984 ±</td>
<td>0.876 ±</td>
<td>0.444 ±</td>
<td>0.947 ±</td>
<td>0.990 ±</td>
<td>0.968 ±</td>
<td>0.606 ±</td>
</tr>
<tr>
<td></td>
<td>0.005</td>
<td>0.052</td>
<td>0.119</td>
<td>0.022</td>
<td>0.006</td>
<td>0.010</td>
<td>0.119</td>
</tr>
<tr>
<td>Minimum Rule</td>
<td>0.911 ±</td>
<td>0.590 ±</td>
<td>0.046 ±</td>
<td>0.995 ±</td>
<td>0.634 ±</td>
<td>0.774 ±</td>
<td>0.143 ±</td>
</tr>
<tr>
<td></td>
<td>0.028</td>
<td>0.036</td>
<td>0.039</td>
<td>0.003</td>
<td>0.043</td>
<td>0.031</td>
<td>0.078</td>
</tr>
<tr>
<td>Geom Mean</td>
<td>0.984 ±</td>
<td>0.883 ±</td>
<td>0.710 ±</td>
<td>0.974 ±</td>
<td>0.960 ±</td>
<td>0.967 ±</td>
<td>0.830 ±</td>
</tr>
<tr>
<td></td>
<td>0.006</td>
<td>0.040</td>
<td>0.076</td>
<td>0.013</td>
<td>0.014</td>
<td>0.005</td>
<td>0.045</td>
</tr>
<tr>
<td>Harmonic Mean</td>
<td>0.968 ±</td>
<td>0.821 ±</td>
<td>0.375 ±</td>
<td>0.987 ±</td>
<td>0.882 ±</td>
<td>0.932 ±</td>
<td>0.567 ±</td>
</tr>
<tr>
<td></td>
<td>0.011</td>
<td>0.030</td>
<td>0.076</td>
<td>0.008</td>
<td>0.028</td>
<td>0.013</td>
<td>0.079</td>
</tr>
<tr>
<td>Maximum Rule</td>
<td>0.905 ±</td>
<td>0.465 ±</td>
<td>0.105 ±</td>
<td>0.715 ±</td>
<td>0.999 ±</td>
<td>0.834 ±</td>
<td>0.240 ±</td>
</tr>
<tr>
<td></td>
<td>0.028</td>
<td>0.152</td>
<td>0.074</td>
<td>0.022</td>
<td>0.001</td>
<td>0.015</td>
<td>0.136</td>
</tr>
<tr>
<td>Majority Vote</td>
<td>0.988 ±</td>
<td>0.895 ±</td>
<td>0.749 ±</td>
<td>0.964 ±</td>
<td>0.985 ±</td>
<td>0.974 ±</td>
<td>0.843 ±</td>
</tr>
<tr>
<td></td>
<td>0.004</td>
<td>0.044</td>
<td>0.071</td>
<td>0.017</td>
<td>0.007</td>
<td>0.007</td>
<td>0.049</td>
</tr>
<tr>
<td><strong>Oracle</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Best Model</td>
<td>0.988 ±</td>
<td>0.835 ±</td>
<td>0.706 ±</td>
<td>0.965 ±</td>
<td>0.985 ±</td>
<td>0.975 ±</td>
<td>0.815 ±</td>
</tr>
<tr>
<td></td>
<td>0.004</td>
<td>0.074</td>
<td>0.141</td>
<td>0.012</td>
<td>0.007</td>
<td>0.005</td>
<td>0.093</td>
</tr>
<tr>
<td>PoolNet Oracle</td>
<td>0.999 ±</td>
<td>0.974 ±</td>
<td>0.876 ±</td>
<td>0.997 ±</td>
<td>0.999 ±</td>
<td>0.998 ±</td>
<td>0.931 ±</td>
</tr>
<tr>
<td></td>
<td>0.001</td>
<td>0.024</td>
<td>0.097</td>
<td>0.002</td>
<td>0.001</td>
<td>0.001</td>
<td>0.055</td>
</tr>
<tr>
<td><strong>Framework</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multiplication</td>
<td>0.990 ±</td>
<td>0.897 ±</td>
<td>0.847 ±</td>
<td>0.978 ±</td>
<td>0.982 ±</td>
<td>0.980 ±</td>
<td>0.901 ±</td>
</tr>
<tr>
<td></td>
<td>0.003</td>
<td>0.054</td>
<td>0.101</td>
<td>0.018</td>
<td>0.008</td>
<td>0.007</td>
<td>0.063</td>
</tr>
<tr>
<td>Minimum</td>
<td>0.990 ±</td>
<td>0.899 ±</td>
<td>0.852 ±</td>
<td>0.976 ±</td>
<td>0.983 ±</td>
<td>0.979 ±</td>
<td>0.880 ±</td>
</tr>
<tr>
<td></td>
<td>0.003</td>
<td>0.051</td>
<td>0.074</td>
<td>0.014</td>
<td>0.010</td>
<td>0.005</td>
<td>0.033</td>
</tr>
<tr>
<td>Maximum</td>
<td>0.989 ±</td>
<td>0.883 ±</td>
<td>0.800 ±</td>
<td>0.972 ±</td>
<td>0.981 ±</td>
<td>0.977 ±</td>
<td>0.856 ±</td>
</tr>
<tr>
<td></td>
<td>0.003</td>
<td>0.056</td>
<td>0.083</td>
<td>0.014</td>
<td>0.007</td>
<td>0.005</td>
<td>0.073</td>
</tr>
<tr>
<td>Mean</td>
<td>0.990 ±</td>
<td>0.899 ±</td>
<td>0.868 ±</td>
<td>0.981 ±</td>
<td>0.978 ±</td>
<td>0.980 ±</td>
<td>0.902 ±</td>
</tr>
<tr>
<td></td>
<td>0.003</td>
<td>0.048</td>
<td>0.081</td>
<td>0.014</td>
<td>0.012</td>
<td>0.006</td>
<td>0.053</td>
</tr>
<tr>
<td>Median</td>
<td>0.990 ±</td>
<td>0.900 ±</td>
<td>0.857 ±</td>
<td>0.982 ±</td>
<td>0.977 ±</td>
<td>0.979 ±</td>
<td>0.908 ±</td>
</tr>
<tr>
<td></td>
<td>0.003</td>
<td>0.051</td>
<td>0.089</td>
<td>0.015</td>
<td>0.011</td>
<td>0.005</td>
<td>0.048</td>
</tr>
<tr>
<td>Gate - SimplestNet</td>
<td>0.989 ±</td>
<td>0.901 ±</td>
<td>0.844 ±</td>
<td>0.984 ±</td>
<td>0.971 ±</td>
<td>0.977 ±</td>
<td>0.903 ±</td>
</tr>
<tr>
<td></td>
<td>0.003</td>
<td>0.041</td>
<td>0.088</td>
<td>0.013</td>
<td>0.011</td>
<td>0.005</td>
<td>0.044</td>
</tr>
<tr>
<td>Gate - UnetPad</td>
<td>0.988 ±</td>
<td>0.909 ±</td>
<td>0.872 ±</td>
<td>0.988 ±</td>
<td>0.963 ±</td>
<td>0.976 ±</td>
<td>0.908 ±</td>
</tr>
<tr>
<td></td>
<td>0.004</td>
<td>0.040</td>
<td>0.073</td>
<td>0.012</td>
<td>0.014</td>
<td>0.005</td>
<td>0.038</td>
</tr>
<tr>
<td>GateX - SimplestNet</td>
<td>0.990 ±</td>
<td>0.891 ±</td>
<td>0.846 ±</td>
<td>0.985 ±</td>
<td>0.975 ±</td>
<td>0.980 ±</td>
<td>0.884 ±</td>
</tr>
<tr>
<td></td>
<td>0.003</td>
<td>0.053</td>
<td>0.093</td>
<td>0.015</td>
<td>0.009</td>
<td>0.005</td>
<td>0.061</td>
</tr>
</tbody>
</table>
Figure 13: Segmentation done by every monolithic model inside PoolNet, where black and white is the correct segmentation, and the colors is the cells predictions.
Figure 14: Segmentation done by static aggregation, where black and white is the correct segmentation, and the colors is the cells predictions.
Figure 15: Segmentation done by the hypothetical models, Oracle and Best Model, where black and white is the correct segmentation, and the colors is the cells predictions.
Figure 16: Segmentation done by the proposed framework, where black and white is the correct segmentation, and the colors is the cells predictions.
CONCLUSION AND FUTURE WORKS

Nuclear segmentation is a necessary step to automatize many tasks in the biologic field. Still, due to some adversarial conditions such as lack of ground truth image and high noise presence, nuclear segmentation turns to be a hard problem for the segmentation algorithms.

We present a new framework for image segmentation based on the use of multi segmentations, an attention network, and an aggregation function with seven variants. We also adapt the concept of Oracle to the image segmentation field. Finally, compare the results with five monolithic models and with an ensemble segmentation using static aggregation function. The proposed framework outperforms both.

In future works, we aim to use the same framework for a three-channel image: background, foreground, and touch channel, as used in Guerrero-Pena et al. (2018), improve the diversity of the ensemble with a non-deep learning segmentation method, and with neural networks trained in a different dataset and use a grid search strategy to fine tune the neural networks present in the framework.
REFERENCES


