
UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Carlos Henrique Caloete Pena

# An Ensemble Learning Method for Segmentation Fusion

Recife

2022

Carlos Henrique Caloete Pena

# An Ensemble Learning Method for Segmentation Fusion

Trabalho apresentado ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.

**Área de Concentração**: Inteligência Computacional.

**Orientador**: Tsang Ing Ren

Recife

2022

Catalogação na fonte
Bibliotecária Monick Raquel Silvestre da S. Portes, CRB4-1217

P397e Pena, Carlos Henrique Caloete
*An ensemble learning method for segmentation fusion* / Carlos Henrique Caloete Pena. – 2022.
55 f.: il., fig., tab.

Orientador: Tsang Ing Ren.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2022

Inclui referências.

1. Inteligência computacional. 2. Segmentação de imagens. 3. Aprendizagem profunda I. Ren, Tsang Ing (orientador). II. Título.

006.31 CDD (23. ed.) UFPE - CCEN 2022-168

**Carlos Henrique Caloete Pena**

**“An Ensemble Learning Method for Segmentation Fusion”**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação. Área de Concentração: Inteligência Computacional.

Aprovado em: 25 de agosto de 2022.

**BANCA EXAMINADORA**

_____

Prof. Dr. Carlos Alexandre Barros de Mello
Centro de Informática / UFPE

_____

Prof. Dr. Luiz Filipe Alves Pereira
Departamento de Ciência da Computação / UFAPE

_____

Prof. Dr. Tsang Ing Ren
Centro de Informática / UFPE
(**Orientador**)

## ACKNOWLEDGEMENTS

I would like to thank my family for providing all the necessary support. To the Professors and Doctors of COSE, Tsang Ren, Pedro Diamel, Fidel Pena, and Alexandre Cunha. To my friends who helped in the revision of this dissertation: Marcus Vinicius, Mariana Barros, Gustavo Mascarenhas, Gabriel Bandeira, Heitor Rapela, Natalia Nascimento, Douglas Vasconcelos, Eduarda Pontual. To pets, Bela, Fliper, Ariel, Pucca, Bob, Arya, Plush, and Vic. Finally, I thank the groups E.S.T.U.F.A, RobôCIn team, and FCX Labs.

## ABSTRACT

The segmentation of cells present in microscope images is an essential step to automate many tasks, including cell counting, analysis of the cell-division cycle, determining protein concentration, and analysis of gene expression per cell. In single-cell genomics studies, cell segmentations are vital to assess the genetic makeup of individual cells and their relative spatial location. Deep learning models are currently the most promising approaches among the various techniques and tools that have been developed to provide robust segmentation. We propose a learning ensemble strategy that aggregates many independent candidate segmentations of the same image to produce a single consensus segmentation as an alternative to developing another cell segmentation targeted model. We are particularly interested in learning how to ensemble crowdsourced image segmentations created by experts and non-experts in laboratories and data houses. Hence, these image segmentations are subject to high potential annotation errors created on purpose or by chance. We compare our trained ensemble model with other fusion methods adopted by the biomedical community, such as SIMPLE and STAPLE, and assess the robustness of the results on three aspects: fusion with outliers, missing data, and synthetic deformations. Our approach outperforms these methods in efficiency and quality, especially when there is a high disagreement among candidate segmentations of the same image.



**Keywords**: image segmentation fusion; image segmentation; deep neural networks; computer vision.

## RESUMO

A segmentação de células realizadas em imagens microscópicas é uma etapa essencial para automatizar multiplas tarefas, incluindo a contagem de células, a aferição da concentração de proteínas e a análise da expressão gênica das células. Em estudos de genômica, a segmentação das células é vital para avaliar a composição genética de células individualmente e a sua localização espacial relativa. Vários métodos e ferramentas foram desenvolvidos para oferecer uma segmentação robusta, sendo, atualmente, os modelos de *deep learning* as soluções mais promissoras. Como alternativa ao desenvolvimento de outro modelo direcionado a segmentação de imagens microscópicas, propomos, nesta dissertação, uma estratégia de aprendizado de fusão que agrega diversas segmentações candidatas independentes provindas de uma mesma imagem para produzir uma única segmentação de consenso. Estamos particularmente interessados em aprender como agrupar segmentações de imagens provindas de ferramentas *crowdsourcing*, podendo ser criadas por especialistas e não especialistas em laboratórios e *data centers*. Assim, comparamos nosso modelo de fusão com outros métodos adotados pela comunidade biomédica, tal como SIMPLE e STAPLE, e avaliamos a robustez dos resultados em três aspectos: fusão com *outliers*, segmentação parcial e deformações sintéticas. Nossa abordagem supera os métodos em eficiência e qualidade, especialmente, quando há uma grande discordância entre as segmentações candidatas da mesma imagem.



**Palavras-chaves**: fusão de segmentações de imagens; segmentação de imagens; redes de aprendizagem profunda; visão computacional.

## LIST OF FIGURES

## LIST OF TABLES

## LIST OF ABBREVIATIONS AND ACRONYMS

| | |
|---|---|
| **CE** | Cross Entropy |
| **COSE** | Collaborative Segmentation |
| **DT** | Distance Transform |
| **DTM** | Distance Transform Merge |
| **EM** | expectation–maximization |
| **FCN** | Fully Convolutional Network |
| **IOU** | Intersection Over Union |
| **J-REG** | Youden's J statistic regularization |
| **MAP** | Maximum a Posteriori |
| **PQ** | Panoptic Quality |
| **ReLU** | Rectified Linear Unit |
| **RM** | Reliability Map |
| **SIMPLE** | Selective and Iterative Method for Performance Level Estimation |
| **STAPLE** | Simultaneous truth and performance level estimation |

## LIST OF SYMBOLS

| | |
|---|---|
| $\phi$ | Distance Transform |
| $C$ | Cost Function |
| $\mathcal{L}$ | Loss symbol |
| $\mathcal{J}$ | Younden Regularization |

## CONTENTS

# 1 INTRODUCTION

Image segmentation is the partitioning of a given image into several objects, facilitating the analysis by focusing on each object. These objects varies according to the segmentation objective. For instance, in a park monitoring system, it is interesting to have each person as a distinct object; on the other hand, in a driver drowsiness detection, it is interesting that the objects are the driver's eyes.

An example of image segmentation employed to facilitate analysis is the automatic verification of compliance with ISO international standards for facial images in documents. One of the criteria defined by the International Civil Aviation Organization (ICAO) is to verify whether objects are close to the face (ICAO 30). It can be simplified when dividing the image into regions of homogeneous coloring and performing the score inversely proportional to the number of regions in the background, as is done in (FERRARA et al., 2012).

There are several methods to perform image segmentation, where the best method will depend on the defined problem. For instance: thresholds algorithms such as the Otsu (OTSU, 1979), the Li (LI; TAM, 1998), and the Ridler-Calvard (ISODATA) (RIDLER; CALVARD et al., 1978) methods; a marker-controlled watershed (MEYER, 1992); an edge detector such as the Canny (CANNY, 1986), and diverse types of deep neural networks approaches.

## 1.1 MOTIVATION

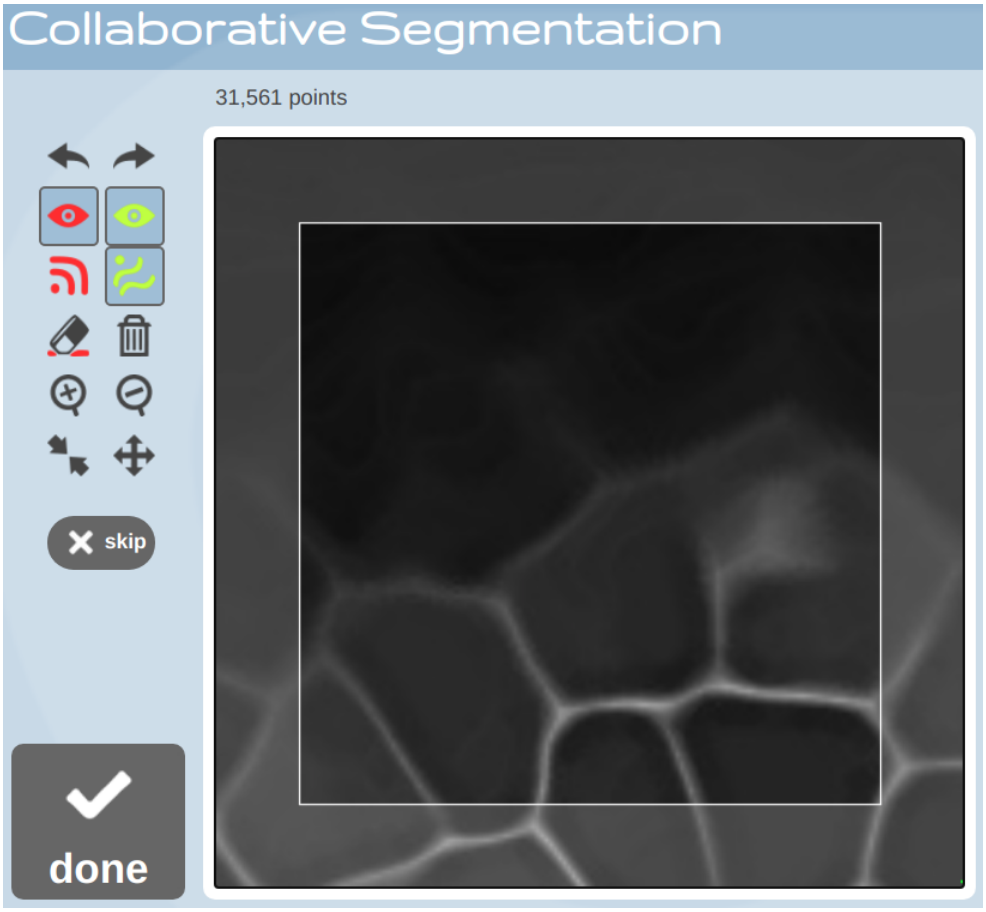
An essential benefit of image segmentation is in microscopy images of tissue, cells segmentation is crucial for many biological analyses, e.g., cell count, smoothness, reproduction rate over time, and visualization of cell behavior. For instance, a classic diagnosis system used in Wisconsin Hospitals (STREET; WOLBERG; MANGASARIAN, 1993) to predict whether the cancer is malignant or benign performs an image segmentation to extract the boundaries of the shape of the nuclei of cells. After the segmentation, for each cell is calculated several features, such as radius, perimeter, area, smoothness, concavity, symmetry, texture, and many others.

However, the image segmentation task is very costly (time and financially) to have a dedicated professional take note. One of the possible methods to mitigate the annotation problem is via a crowdsourcing system, in which non-expert people can annotate at a much lower cost. Although this approach significantly increases the speed and quantity of annotated

images, it has the disadvantage of losing segmentation quality.

An example of a web crowdsourcing platform is the COSE (BARBOSA; REN, 2014)[1] that allows general collaborators to perform image segmentation with a user-friendly interface as can be seen in Figure 1. The platform has two operation modes. The first one is manual, in which the collaborator needs to draw every cell outline. The second one is semi-automatic, where the collaborator needs to choose one of three implemented algorithms (Live Wire, Live-vessel, or Watershed) and interactively input seeds corresponding to the foreground region to adjust the chosen algorithm's response. However, image segmentation is still a complex task; some factors can confuse the user, such as noise in the acquisition process, low-resolution image, cell overlapping, user fatigue, and lack of specialized knowledge. An example of an easy and challenging one is shown in Figure 2.

Figure 1 – The COSE interface, the user can perform manual annotation using the manual mode (red contour) or the smart mode (green contour).




**Source:** http://cose.cambia.caltech.edu/ (2022)

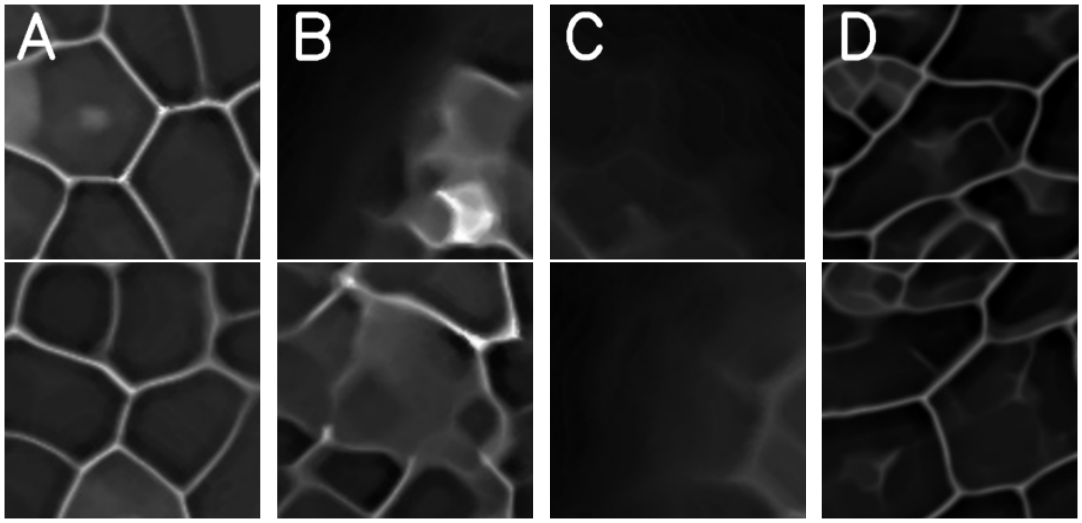
Since crowding source platforms allow non-experts to annotate images, the platforms must

[1] The 2D segmentation web interface is available at http://cose.cambia.caltech.edu/

be prepared for a high annotation error rate, which may be even caused by malicious users. One of the strategies is to allow several users to annotate the same image, thus creating a set of image segmentations, then using a segmentation fusion algorithm to transform this set of segmentations into a single image segmentation with a lower error rate.

Currently, the solutions used for segmentation fusion are based on classic learning methods, such as the popular segmentation fusion algorithms: Majority Vote, STAPLE (WARFIELD; ZOU; WELLS, 2004), SIMPLE (LANGERAK et al., 2010), and GEMS (CUNHA, 2019). However, despite the algorithms reducing the average error rate, it is common for errors to occur mainly when there is a high degree of disparity in the segmentation set. Inspired by the results of works using deep learning in biomedical image segmentation with weakly supervised data (GUERRERO-PEÑA et al., 2019), (CAICEDO et al., 2019), we propose to use deep learning for the task of fusion image segmentations. To the best of our knowledge, this type of learning technique was not used in the context of fusion biological image segmentation as presented in this work.

Figure 2 – Comparison between images considered easy to segment (A), and images considering hard to segment due to blur (B), low-level luminance (C), and tiring due to presence of small regions with no precise edges (D).




**Source:** The author (2022)

## 1.2 OBJECTIVES

Our objective is to enhance fusion of biological image segmentation originated from crowd-source annotation tools with high disagreement. Our proposal is a new framework based on a deep convolutional network that relies on heavy data augmentation for high disagreement

annotations found in crowdsource tools. We compare the robustness of the proposed method with well-known algorithms such as SIMPLE (LANGERAK et al., 2010) and STAPLE (WARFIELD; ZOU; WELLS, 2004) in three different tasks: varying the number of outliers, missing data, and deformations.

In order to propose a robust fusion model, we train three models with different loss functions: Cross Entropy (LONG; SHELHAMER; DARRELL, 2015), Dice (MILLETARI; NAVAB; AHMADI, 2016) and J Regularization (PEÑA et al., 2020). These functions were selected because of the well-proven results for the panoptic segmentation task (KIRILLOV et al., 2019) in a high imbalanced class problem with weak annotations.

## 1.3 CONTRIBUTIONS

- An adaptation of the U-Net neural network for segmentation fusion, focusing on high disagreement and noise environment.
- An analysis of several well-known fusion algorithms in three tasks: outliers, missing data, and deformations, that are based on the problems encountered in crowdsource segmentation platforms.

The work written in this dissertation has been published below:

- **Carlos H. C. Pena**, Tsang Ing Ren, Pedro D. Marrero Fernandez, Fidel A. Guerrero-Peña and Alexandre Cunha. "An Ensemble Learning Method for Segmentation Fusion." The 2022 International Joint Conference on Neural Networks (IJCNN). IEEE, 2022.

## 1.4 STRUCTURE

The remaining sections of this dissertation are structured as follows:

- Chapter 2: We introduce and discuss the concepts and terminology that underlie the development of this work such as the concepts of image segmentation, basic morphological operations and evaluation metrics.
- Chapter 3: The basic concepts of well-known segmentation fusion algorithms used to compare the proposed work.

- Chapter 4: This chapter describes and discusses the proposed solution to the problem of fusion weakly supervised segments, the changes made, and the robustness tasks used to verify the quality of the results.
- Chapter 5: This chapter presents the final considerations on the main topics covered in this dissertation, including contributions and indications for future work.

# 2 THEORETICAL BACKGROUND

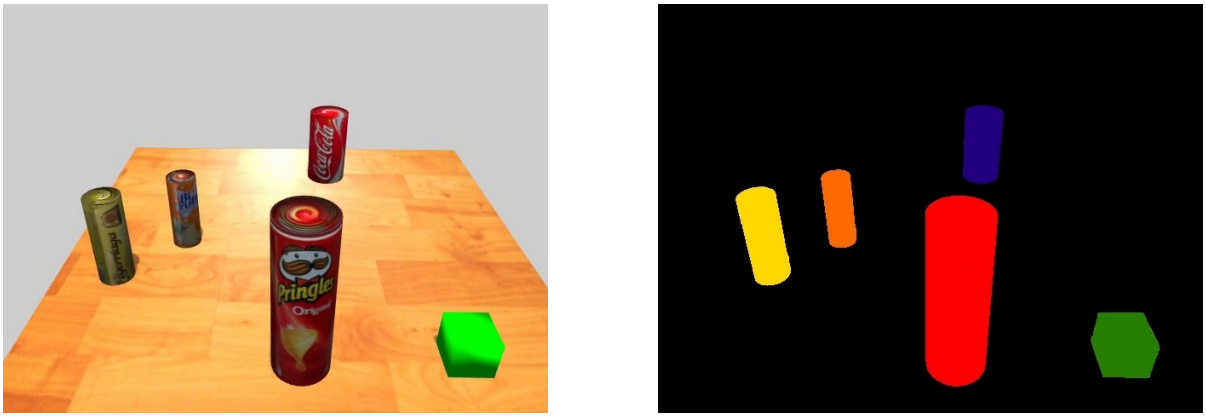
This chapter presents the concepts of image segmentation, the morphology operations commonly used as segmentation post-processing, supervised evaluation methods, a quick view of convolution neural networks, and recommended data augmentation used during training.

## 2.1 SEMANTIC SEGMENTATION

In academic and industrial fields, an image is often taken to describe objects or sub-objects with a technical description, for instance, a robot that takes an image to count the number of objects inside an environment, locates the position of those objects, and then, extracts the dimensions of the objects. Sometimes, the direct extraction of these features is a difficult task, and to walk through this procedure, it is often used an image segmentation that converts those objects into a visualization where each object is described with a unique label, commonly this label is represented as a unique color, as shown in Figure 3, where the pixels of the same color represent the same object. With this segmented view, the object counting is as easily as count the number of unique non-background pixels values, the object position could be estimated using the mass center of each label, and the object width[1] subtracting the right-most pixel position by the left-most pixel position.

Figure 3 – An image of simulated environment with five objects 3a, and its image segmentation 3b.

(a) (b)




**Source:** The author (2022)

There are several methods to perform an image segmentation. In general, they aim to

[1] Converting the size of the object in the image to the actual size involves techniques that are outside the scope of this project.

analyze the similarity or the discontinuity of the pixels level intensity. The similarity (region-based) algorithms splitting images into regions based on their intensity value, that could be a threshold value (median, percentile, Otsu (OTSU, 1979)), a region growing (watershed with markers based) (BEUCHER et al., 1992), a cluster based (K-Means(MACQUEEN et al., 1967)), and many others. On the other hand, the discontinuity (boundary-based) algorithms search for local abrupt change on the image, in other words, points, lines, and edges.

**Notation.** Let a two-dimensional single-channel image be defined as $x : \Omega \to \mathbb{R}$, where every element $p \subset \Omega$ is called pixels. Then, the semantic segmentation task can be expressed as a function $y = H(x)$, that assigns a categorical label $y : \Omega \to 0, ..., C$ for every pixel $p$, where $C$ is the number of unique classes in the given problem. The transformation $H(x)$ can be done in different ways, such as manually, by an algorithm such as Otsu thresholding, seeded Watershed(COELHO; SHARIFF; MURPHY, 2009), or even a Deep Neural Network (BADRINARAYANAN; KENDALL; CIPOLLA, 2017)(WANG et al., 2018)(RONNEBERGER; FISCHER; BROX, 2015).

An example of image segmentation is cell nucleus segmentation in microscopy images, where it is necessary to distinguish between cell nucleus (foreground) and non-nuclear objects (background). Given a microscopy image, as shown in Figure 4a, the segmentation generates a binary map as in Figure 4b, in which the foreground and the background are typically represented with a white and a black label, respectively.

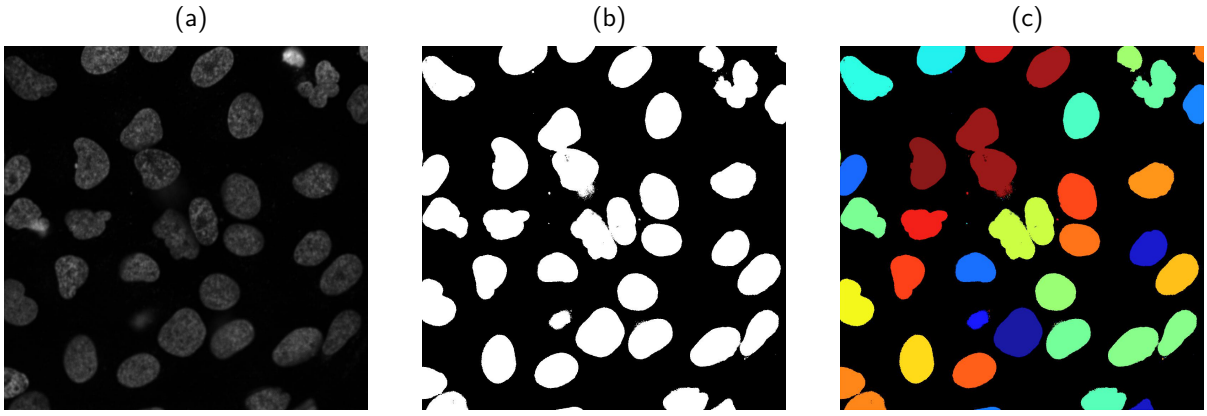
There are several ways to perform this image segmentation, one of the ways is using the thresholding technique, in which, contiguous regions above the threshold value are defined as foreground and the remaining as background. As an example, let the threshold value $T_h$ be the average pixel value of the original image as in Equation 2.1, the function $H(x)$ is defined ,in this scenario, as in Equation 2.2.

Although in this example, it is possible to define a function $H(x)$ in terms of the $x$ pixels of the original image, this threshold method is prone to a high error rate. For instance, in some datasets, the edges of the images have a high light intensity, therefore, raising the $T_h$ value. For other problems, such as the segmentation of individuals, animals, and objects, the precise definition of this function is still unclear, and in these scenarios, deep learning methods are the most promising to solve them.

$$T_h = \frac{1}{|\Omega|} \sum_{p \in \Omega} p \tag{2.1}$$

$$H(x) = \begin{cases} 1 & \text{if } x \geq T_h \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

Figure 4 – An microscopy image of U2OS cells (a), Its segmentation using the average pixel value as a threshold (b), and its panoptic segmentation (c).




**Source:** (a) (COELHO; SHARIFF; MURPHY, 2009) (b-c), the author (2022)

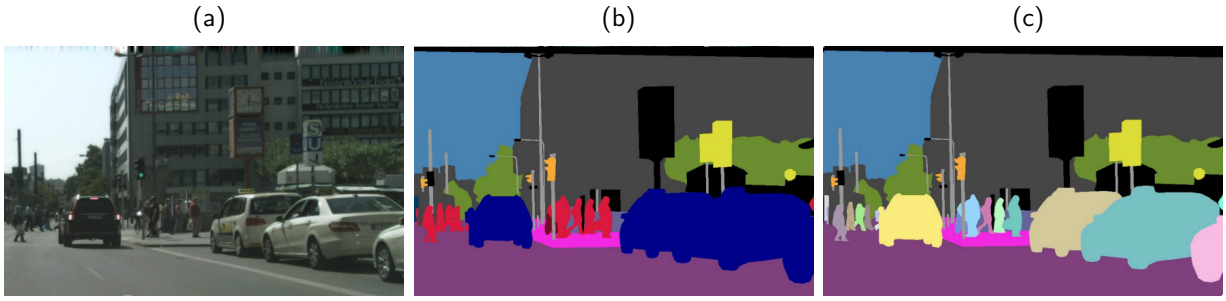
## 2.2 PANOPTIC SEGMENTATION

Kirillov *et al.* Introduced the Panoptic Segmentation in 2019 (KIRILLOV et al., 2019). The task consists of assign to each pixel of an image a semantic label (*e.g.*, car, pedestrian, table) and a unique ID value (*e.g.*, 0, 1, 2). In which it is not possible to have overlapping between the instances.

Using as an instance the problem of an autonomous vehicle, the Figure 5a was recorded. Also, its semantic and panoptic segmentation was calculated, as shown in Figure 5b and 5c respectively. Depending on the project requirements, the kind of segmentation is more suitable to solve the problem, for example, if the car just needs to navigate the city, respect traffic laws, and avoid pedestrians. In this case, it is essential to know where the pedestrians are but not necessarily how many there are. However, in a risk reduction system case, is essential to forecast if a pedestrian is heading to cross the street on a collision course. Therefore, it is necessary to account for each object individually. Thus, panoptic segmentation is more suitable.

However, when analyzing cell images, it is crucial to have an individual distinction of each object so that its features: area, radius, symmetry, and others, are correctly accounted for.

Figure 5 – A sample image (a) followed by its semantic segmentation (b) and its panoptic segmentation (c).




**Source:** (KIRILLOV et al., 2019)

In situations like this, the Panoptic Segmentation is more suitable. It is possible to transform from semantic segmentation to panoptic segmentation, if the objects of the same class are not juxtaposed to each other, by assigning a unique ID to each connected component of a foreground class. As a case in point, Figure 4c shows the semantic instance segmentation of Figure 4b. This panoptic segmentation presents different cells with the same ID due to juxtaposed cells. In these cases, a pre-processing function can be used to separate the wrong joint cells, such as the morphological Opening operation, which will be discussed in the next section.

## 2.3 MORPHOLOGICAL OPERATORS

Morphological operations are tools used for image analysis, widely used for pattern search, text enhancement, region thinning, and many others. In image segmentation, dilate, erode, opening, and closing operations are widely used as post-processing, which we describe below. These four operations change the original image $A$ according to a matrix $s$, called a structuring element. The matrix $s$ needs an central point (commonly the mass center), and it is common for $s$ to be much smaller than image $A$, such as a circle with a radius of 3 pixels or a 5x5 square.

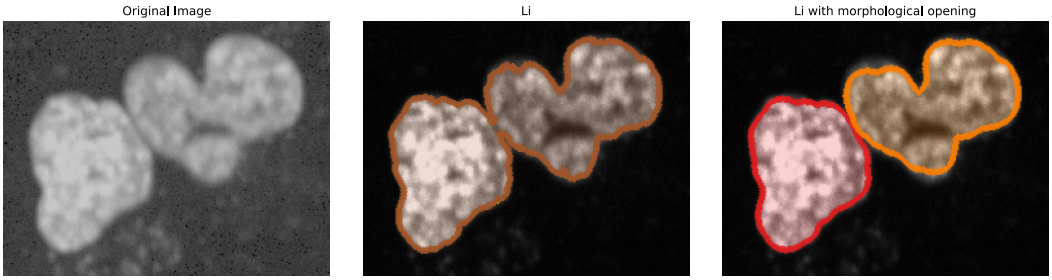
**Dilation**: is an operation of thickening the image defined as a set of all translations of $s$ with points that have at least one non-zero element in common with $A$. Formally, $A \oplus s = \{z|(\hat{s}_z \cap A \neq \emptyset)\}$, where $\hat{s}$ is the reflection of the structural element and $s_z$ is the translation of $s$ over the pixel $z \in \Omega$. This operator can be used to combine two close components that are slightly disconnected.

**Erode**: is the operator that thins elements from $A$, defined as a junction of all pixels $z$

that the translations of $s$ by $z$ are contained in the image $A$. Formally $A \ominus s = \{z|(s_z \subseteq A)\}$ (GONZALEZ; WOODS, 2006). This operator removes fine or small elements, such as salt noise.

**Opening**: defined as an erosion followed by a dilation with the same structuring element, $A \circ s = (A \ominus s) \oplus s$, the operation has the effect of smoothing the contours, and removing tight connections, fine or small elements. As in Figure 6, in which, two juxtaposed cells can be wrongly segmented into just one cell by a few pixels connecting both. The morphological opening operation is used to split the segmentation.

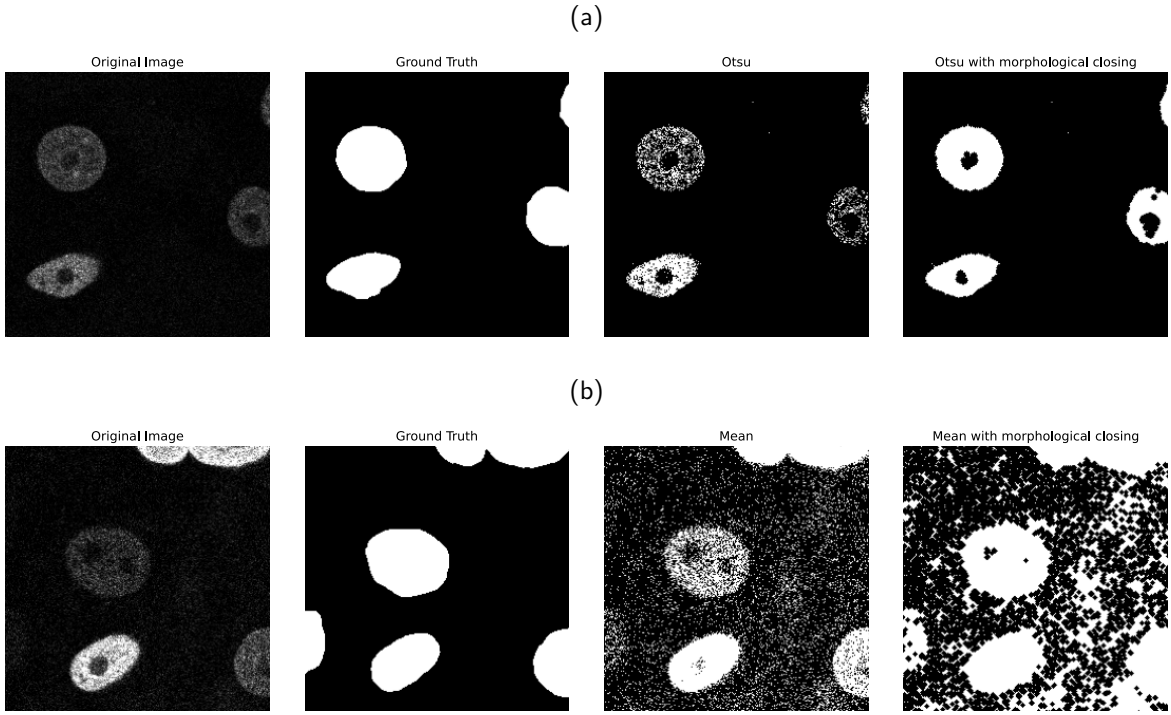
Figure 6 – Sample image with two juxtaposed cells segmented into one cell when using Li's threshold method, and the segmentation with morphological opening post-processing.




**Source:** The author (2022)

**Closing**: defined as dilation follow by an erosion with the same structuring element, $A \bullet s = (A \oplus s) \ominus s$, the operation has the effect of joining discontinuities and reducing holes. Figure 7a shows an image segmentation using the Otsu algorithm, in this scenario in which most of the errors shown are found in the foreground areas. It is possible to process the Otsu segmentation to form a single connected set that represents the cell, note that if the morphological closing is applied to a segmentation similar to the one shown in Figure 7b it can further degrade the quality of segmentation.

Figure 7 – Example of the morphological closing operation in a binary segmentation using the (a) Otsu threshold and (b) mean threshold.




**Source:** The author (2022)

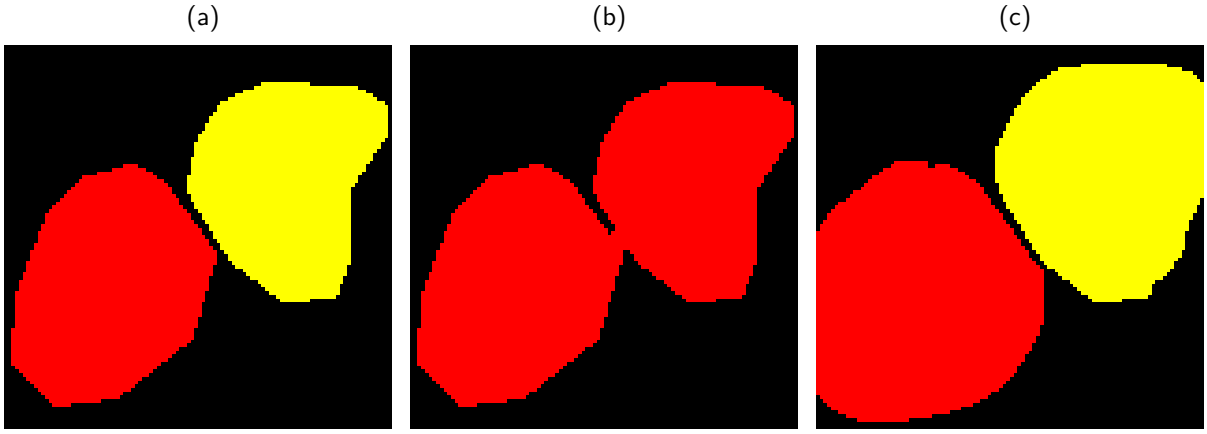
## 2.4 EVALUATION MEASURES

There are several ways to measure the quality of a segmentation, some of these metrics are derived from a pixel-wise comparison from the proposed segmentation $y$ and its ground truth $g$ in which every pixel is classified into one of the following terms: True Positive (TP) if it is correctly classified as a cell, True Negative (TN) if it is correctly classified as a background, False Positive (FP) if it is classified as a cell although it is a background and finally False Negative (FN) if it is classified as a background pixel, however, it should be a cell pixel.

One of the most used and straightforward metric is the accuracy, which is the sum of True pixels $(TP + TN)$ over the total number of pixels. However, in cell segmentation, the accuracy could be misleading, because one of the key requirements in task is to separating cells correctly. An example of this behavior can be seen in Figure 8, in which the segmentation 8b wrongly merges two cells. Although only a few pixels are wrongly classified, this error directly impacts all biological analyses cell count, radius of the cells, symmetry, and numerous others. Nevertheless it achieves an accuracy of around 99%. Because of the similarity between the intensity level of the touching edges and the inner part of the cell, the problem of wrongly

merging cells is frequent when using automatic segmentation.

Figure 8 – A toy segmentation problem where pixels of the same color (red or yellow) denote an element of the same cells, with (a) ground truth with two cells, and two proposed segmentation: (b) that contains only one cell however with borders similar to ground truth and (c) with two cells, however, these cells do not have correctly marked edges.




**Source:** The author (2022)

Another key point for the evaluation process is considering the class imbalance frequently found in microscopic images. For instance, the image shown in Figure 9 is from a Mouse stem cells dataset of the Cell Tracking Challenge[2], and it has just one tiny cell present that represents less than 0.0005% of the whole image. In other words, a dummy segmentation method that just returns the background class has over 99.9% of accuracy in this image.

The F-Measure (F1) is the harmonic mean with equal importance to precision (positive prediction rate) and recall (true positive rate) and is widely used in scenarios in which the True Negative is not relevant for the task. As in the example in Figure 9, considering the True Negative pixels, the result is biased and misleading.

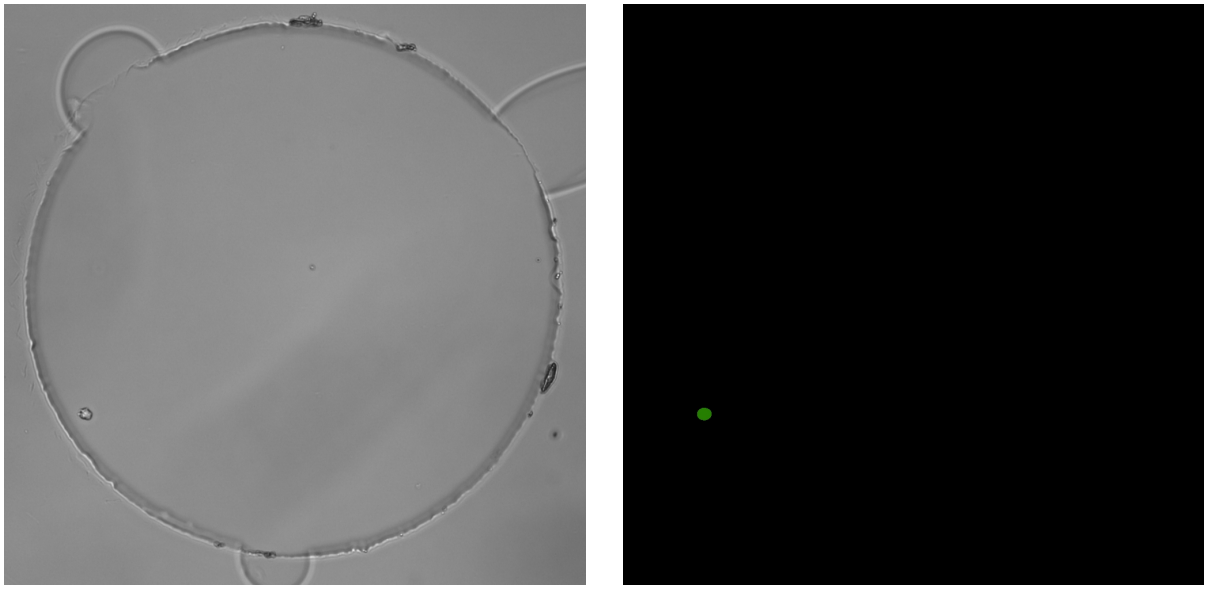
Another measure function is the Intersection Over Union (IOU) (also known as Jaccard index or Jaccard similarity coefficient), widely used in detection and segmentation tasks. Given a predicted mask P and its correct segmentation G, the measure calculates the intersection area of P and G divided over the union areas of P and G.

Finally, the Panoptic Quality (PQ)(KIRILLOV et al., 2019) metric was explicitly created for the Panoptic Segmentation problem. This problem has a particular property that, given a ground truth mask, it can have at the most one respectively predicted mask with IOU greater than 0.5. The metric can be summarized as the average IOU of matched segments (IOU greater than a threshold value) with a penalization for unmatched segments.

[2] http://celltrackingchallenge.net/

Figure 9 – A sample image of mouse hematopoietic stem cells in hydrogel microwells (left) which contains a small number of pixels considered in the foreground (green pixels on the right image).

(a) (b)



**Source:**(a) http://celltrackingchallenge.net/2d-datasets/ (2022) (b) The author (2022)

As in recent works of microscopic image segmentation, we use the PQ as the primary metric for this project. This metric is more consistent with the problem of cell segmentation. Back to our problem of Figure 8 the first proposed segmentation 8b achieves an accuracy of 0.99 and a PQ of 0.35; and 8c with an accuracy of 0.86 and PQ of 0.77, which is much closer to what would be defined as good for a specialist. Finally, the Equations from 2.3 to 2.8 summarize the metrics discussed in this section.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.3}$$

$$precision = \frac{TP}{TP + FP} \tag{2.4}$$

$$recall = \frac{TP}{TP + FN} \tag{2.5}$$

$$F\_measure = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{2.6}$$

$$PQ = SQ \times RQ \tag{2.7}$$

$$PQ = \frac{\sum_{(p,q) \in TP} IoU(p,q)}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|} \tag{2.8}$$

## 2.5 SEGMENTATION WITH DEEP LEARNING

When the segmentation task requires a low error rate, as in the example of the Cell Tracking Challenge[3], or when it is necessary to segment several classes, such as Cityscapes (CORDTS et al., 2016) and Microsoft Common Objects in Context (COCO) (LIN et al., 2014) datasets, a manual segmentation algorithm will hardly achieve the desired performance. (CAICEDO et al., 2019) reported on the 2018 Data Science Bowl, competition on the Kaggle platform over cell segmentation. The dataset for this competition is quite challenging due to the variety of cell types, capture procedures, and few images. The article reports the three best solutions, all of them based on a Fully Convolutional Network (FCN), with the first being an ensemble of 32 FCN with eight different architectures.

In biomedical image segmentation, there has been a significant advance using Deep Learning, specially, since the creation of U-NET (RONNEBERGER; FISCHER; BROX, 2015). The U-NET was a revolutionary model that made it possible to train a deep network with few annotated images with a relatively quick evaluation time. The model relies on solid data argumentation and its U-shaped encoder-decoder architecture with skip connection, as illustrated in Figure 10.

Like the other models of convolutional networks, U-Net architecture is based on the following operations:

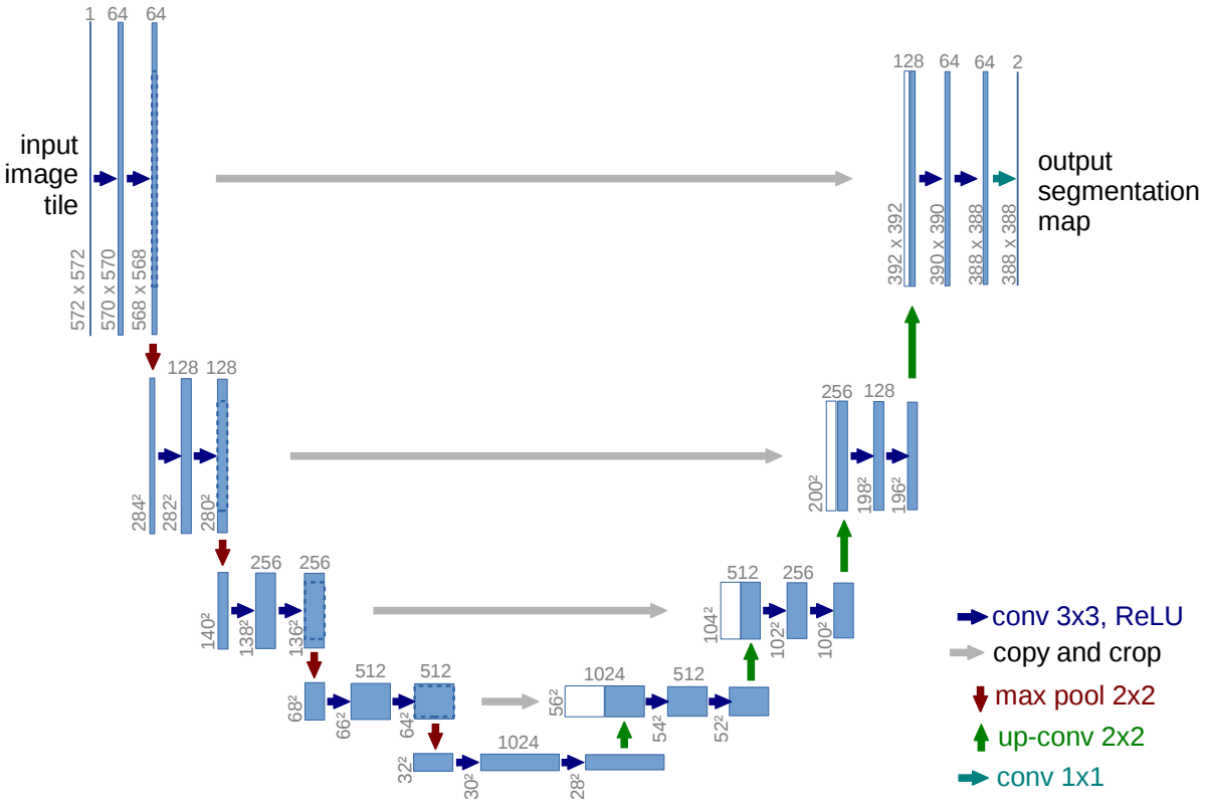
- **Convolution**: Convolution is the primary operation of this kind of network. We can visualize it as a trainable pattern detector. An oversimplified example is shown in Figure 11, in which a 6x6 kernel for detecting the letter "A" is convoluted over the letters "A", "B", "C", and "D". The result of the convolution of the image by the kernel gives us a score that varies according to the similarity with the kernel, in this scenario, the letter "A".

  In the case of neural networks, several kernels are convolved[4] in the image at each layer. More specifically, in the first layer of the U-Net network, 64 kernels 3x3 are convoluted to the original image, creating a multichannel feature map in which each pixel value of the i-th channel results from the convolution of $kernel_i$ in a region of the image. In the subsequent layers, the feature maps from the previous layer are used as input,

[3] http://celltrackingchallenge.net/

[4] Despite the name 'convolution', as the weights are trained, the libraries use the correlation operation to avoid the computational cost of reversing the input.

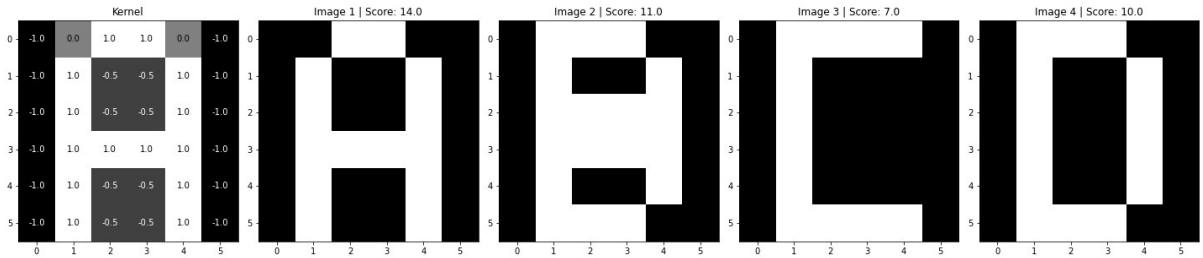
Figure 10 – The enconder-decoder U-Net architecture.




**Source:** U-Net: Convolutional Networks for Biomedical Image Segmentation (RONNEBERGER; FISCHER; BROX, 2015)

thus creating new feature maps, following Equation 2.9. Where $C_{IN}$ and $C_{OUT}$ are the numbers of input and output channels, respectively, and $*$ is the correlation operation. Finally, a bias is add to weight the feature maps individually.

Figure 11 – Example of a kernel for detecting the letter "A" and the four letters "A, B, C, D" with their respective convolution scores.




**Source:** The author (2022)

$$feature\_map(C_{out}) = \left( \sum_{k=0}^{C_{in}-1} weight(C_{out}, k) * input(k) \right) + bias(C_{out}) \quad (2.9)$$

- **Activation Function**: prevents the output from being modeled as a combination of linear functions $(a \cdot x + b)$; therefore, also prevents the final solution to also be linear. Usually, the action functions of neural network models use one of the following:
  - Hyperbolic Tangent $= \frac{e^x - e^{-x}}{e^x + e^{-x}}$
  - Sigmoid $= \frac{1}{1+e^{-x}}$
  - Rectified Linear Unit (ReLU) $= max(0, x)$

  The ReLU is the most used among the three, the function avoids the gradient vanishing problem that would occur in deep learning models with Hyperbolic Tangent or Sigmoid since the ReLU derivative has codomain one for each $x$ greater than zero. Furthermore, for making the network achieve better results faster(KRIZHEVSKY; SUTSKEVER; HINTON, 2012). However, a disadvantage of the ReLU function is that for negative x values, its result is zero; consequently, the calculated gradient is also zero, thus making optimization algorithms difficult. To smooth this problem, (MAAS et al., 2013) proposed a modification of ReLU, called Leaky ReLu, where the hardcoded zero is replaced by $0.01x$, so regions of negative values of $x$, have a non-zero gradient.

- **Pooling**: The pooling layer (also known as the subsampling layer) is used to reduce the dimensions of the feature maps, in which a single value replaces over an N x M grid, consequently reducing the number of features thus, the number of mathematical operations. The most commonly used pooling functions are the maximum pooling (as in the U-Net) and the average pooling.

In general, segmentation networks are based on the encoder-decode architecture using a building block from a well-known classifier network on the encoder path. An example of a building block is the ResNet(HE et al., 2016) and the VGG16(SIMONYAN; ZISSERMAN, 2014) used in LinkNet(SHVETS et al., 2018) and TernausNet(IGLOVIKOV; SHVETS, 2018), respectively.

## 2.6 DATA AUGMENTATION

A good learning model must learn from the data and generalize the data to get good results with new data. One of the methods to improve generalization is to use synthetic transformations in the data to increase their diversity. Especially when dealing with microscope images, the images and their annotations are scarce.

In this scenario, the main operations performed randomly are rotations, flips, resize, crops, blur, change in brightness, contrast and gamma, and mainly elastic distortions (SIMARD et al., 2003). As said in (RONNEBERGER; FISCHER; BROX, 2015) the elastic deformations is essential for training with few images, this operation consists of creating a Gaussian displacement field and performing a pixel-to-pixel shift in the original image according to the vector in the same position on the displacement field.

## 2.7 SEGMENTATION FUSION

While the goal of image segmentation is to create a new representation $y$ of the original image $x$, the goal of the segmentation fusion is to merge *N* already segmented images $Y = \{y_1, y_2, ..., y_n\}$ into a unique segmentation $\hat{y}$. These segmentations could be created from any automatic method or be handmade segmented. Nevertheless, as in any ensemble method, the final result is expected to be as good as the data are precise and diverse. As well as the segmentation the output is a that assigns a categorical label $\hat{y} : \Omega \rightarrow 0, ..., C$ for every pixel $p$, so we can use the same evaluation methods described in the 2.4 section, as well as the neural networks and data augmentation.

# 3 RELATED WORKS

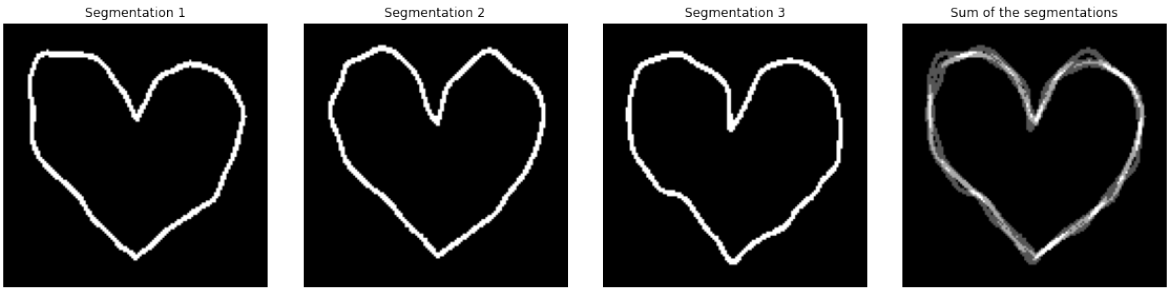
In this chapter, we explain the basic concepts of well-known segmentation fusion algorithms used to compare the proposed work. All the following algorithms require solely the pool of segmentations for the fusion task, not requiring the original image during this process.

## 3.1 FUSION ALGORITHMS

Let $Y = \{y_1, y_2, ..., y_n\}$ be the segmentation pool composed of $n$ 2D binary segmentation $y$, the goal of the following methods is to fusion it into a unique segmentation $\hat{y}$. For this dissertation, we use the 0 value for background and 1 value for the foreground (segmentation border). To exemplify the concepts of the techniques explained below, we use a toy dataset composed of three hand-drawn hearts $\{y_1, y_2, y_3\}$, as shown in the Figure 12, considering that all pixels, whether background or foreground, have the same confidence level in the segmentations.

Depending on the problem, it is common for some operations to be applied to the method output $\hat{y}$. In this scenario, it might be interesting to use an algorithm to merge near contours, (e.g., Morphological Closing), and a method to uniform the thickness of the contour, such as morphological skeleton or Zhang-Suen Thinning (ZHANG; SUEN, 1984).

Figure 12 – Three freehand hearts segmentations, and the sum of these segmentations (rescaled for visual proposes).

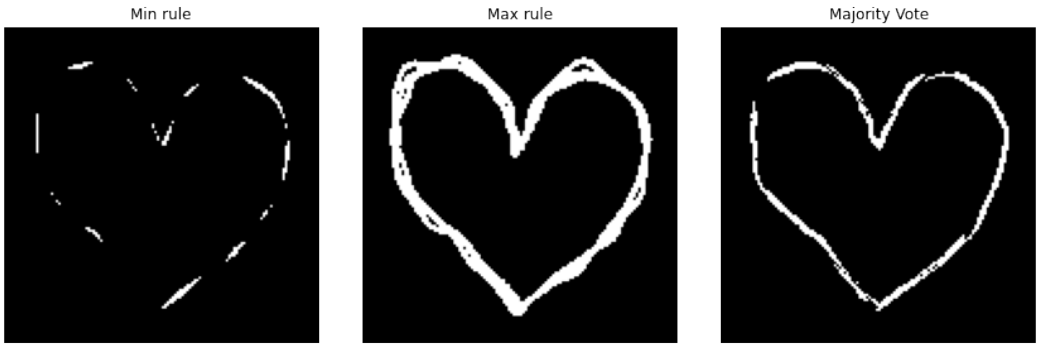



**Source:** The author (2022)

### 3.1.1 Fixed Rules Algorithms

Several non-trainable rules algorithms such as the Average Rule and Majority Voting (MV) are widely used to aggregate binary inputs (DUIN et al., 1998; DUIN, 2002; TAN et al., 2019; THAMBAWITA et al., 2021). Although they are fairly straightforward algorithms, they are likely to produce unsatisfactory results without regard to spatial dependence when applied in image segmentation tasks. An example of them is seen in Figure 13, where the Minimum, Maximum, and Majority Voting are performed in the segmentations of the toy problem of Figure 12. In this scenario, the operation described is performed on all pixels with the same position $(x, y)$ along the segmentation pool $Y$.

Figure 13 – The segmentation output of the Figure 12 produced by the fixed rules algorithms: min, max and majority vote.




**Source:** The author (2022)

### 3.1.2 Simultaneous Truth and Performance Level Estimation (STAPLE)

Warfield et al (WARFIELD; ZOU; WELLS, 2004) shows the Simultaneous truth and performance level estimation (STAPLE), an expectation–maximization (EM) weighted voting technique for aggregate image segmentation. The algorithm iteratively estimates every segmentation's sensitivity and specificity and predicts a ground truth segmentation based on the previous score estimation. Despite being one of the most famous classical algorithms for image fusion, this approach's principal drawback is the computational cost required for the EM convergence.

Let $N$ be the number of pixels in a segmentation, and $R$ the size of segmentation pool, also the column vectors $\mathbf{p} = (p_1, p_2, ..., p_R)^T$, $\mathbf{q} = (q_1, q_2, ..., q_R)^T$ in which $p_i$ and $q_i$ be the True Positive Rate and True Negative Rate of the i-th segmentation. Let $\mathbf{D}$ be an $N \times R$

matrix showing a ravel view of the segmentation pool, in which, every segmentation is viewed as a row, as shown in Figure 14. Finally, let $\mathbf{T}$ be a vector representing the binary ground truth. Given the probability function $f(\mathbf{D}, \mathbf{T}|\mathbf{p}, \mathbf{q})$ the STAPLE estimates the vectors $(\mathbf{p}, \mathbf{q})$ using the Equation 3.1. Since we do not have *a priori* the vectors **T**, **p** and **q**, the EM method is used for the iterative calculation of the non-observable variables.

$$(\hat{\mathbf{p}}, \hat{\mathbf{q}}) = \arg\max_{\mathbf{p},\mathbf{q}} f(\mathbf{D}, \mathbf{T}|\mathbf{p}, \mathbf{q}) \tag{3.1}$$

Figure 14 – The $\mathbf{D}^T$ matrix of the toy dataset 12. We transpose the $\mathbf{D}$ matrix and resize the segmentations to 8x8 pixels (N=64) for visual purposes.


**Source:** The author (2022)

### 3.1.3 Selective and Iterative Method For Performance Level Estimation (SIMPLE)

In (LANGERAK et al., 2010) Langerak et al. introduce the iterative algorithm Selective and Iterative Method for Performance Level Estimation (SIMPLE) for the multi-atlas based segmentation problem. This algorithm stands out for having a result comparable to STAPLE, decreasing the implementation complexity, and improving the response time.

Let $\chi_n$ be a score of each segmentation of the set $\{y_n\}_{n=1}^N$, and let $\hat{y}$ be a ground-truth estimation calculated with the Weight Majority Vote rule of the segmentations $\{y_n\}_{n=1}^N$ using $\chi_n$ as a weight. The SIMPLE algorithm starts by setting the array $\{\chi_n\}_{n=1}^N$ as unit value and iteratively updates the estimation $\hat{y}$, and consecutively the $\chi_n$ score using the Normalized Mutual Information (NMI) metric between each segmentation $y_n$ and the ground-truth estimation $\hat{y}$. After the $i$-th iteration, the segmentations whose scores are lower than a threshold value $\theta$ are removed from the set. This threshold value $\theta$ is given by Equation 3.2, where $\overline{\chi^i}$, $\sigma^i$ are the average and the standard deviation of $chi$ on the $i$-th iteration, respectively, moreover $\alpha$ is a parameter that controls the algorithm convergence speed.

$$\theta^i = \overline{\chi^i} - \alpha \cdot \sigma^i \tag{3.2}$$

### 3.1.4 Distance Transform Merge (DTM)

The DTM (BARBOSA; REN, 2014) is a fast and effective algorithm, initially proposed for the COSE system (BARBOSA; REN, 2014), based on the Distance Transform (DT) algorithm. The DT creates a heatmap where the value of each pixel $p$ represents the euclidean distance between $p$ and $q$, the nearest foreground pixel. Formally $\phi_t(p) = \min \|p - q\|_2^2$. This heatmap is used to build an intermediate probability region near the foreground pixels. The DTM creates an Reliability Map (RM) based on the Equation 3.3, where $L$ is the number of segmentations, and $\phi_{i'}$ is the DT of the complementary of the $i$-th segmentation, the final segmentation is calculated using the Otsu thresholding over the Reliability Map. The DT, RM, and DTM results of the toy dataset is shown in Figure 15.

$$RM(x, y) = \sum_{i=1}^{L} \frac{1}{1 - \phi_{i'}(x, y)} \tag{3.3}$$

Figure 15 – An example of the DT of the complementary segmentation 1 in Figure 12, the RM of the toy dataset of 12, and the final DTM result.


**Source:** The author (2022)

### 3.1.5 Geometric Median Shapes (GEMS)

The work of Cunha (CUNHA, 2019) proposes a fast and simple extension to the median rule for high dimensions inputs. The algorithm aims to capture the central tendency of the segmentations (in the article is referred to as curves), and is focused on being robust to high levels of outliers. Initially, the algorithm deals with the problem of finding the best curve as an optimization problem for finding the optimal curve that minimizes its distance to the set of

curves. In the experiments the author showed that the GEMS algorithm are still able to find a curve that captures the central tendency even with $50\%$ contamination.

Let $x$ be a point and $\Gamma$ be a curve, the distance between $x$ and $\Gamma$ is defined as the Euclidean distance between $x$ and the nearest point $y$ on the $\Gamma$ curve as in Equation 3.4. Moreover, this definition is expanded to the distance between two curves $\Gamma$ and $\Psi$ as the average of the distance between every point $x \in \Gamma$ and the curve $\Psi$ as in Equation 3.5, which $d_\Psi$ is given by Equation 3.4.

$$d_\Gamma(x) = \min_{y \in \Gamma} \|x - y\| \tag{3.4}$$

$$d(\Gamma, \Psi) = \frac{1}{|\Gamma|} \int_\Gamma d_\Psi(x) ds \tag{3.5}$$

$$\Gamma^* = \arg\min_\Gamma \sum_{i=1}^{n} d(\Gamma, \Gamma_i) \tag{3.6}$$

Equation 3.6 defines an optimal closed curve $\Gamma^*$ as the curve with the smallest accumulated distance between $\Gamma^*$ and a given set of contours $\Gamma_i \in \mathbf{\Gamma}$. Once the objective of finding the $\Gamma^*$ curve has been defined, the author uses the watershed (BEUCHER; MEYER, 2018) method to solve $\Gamma^*$ as an optimization problem, thereby is created a grid $\phi$, in which each element $x$ represents the cumulative distance between $x$ and every curve in $\mathbf{\Gamma}$ using the DT algorithm. Finally, the markers (required for the watershed algorithm) are calculated using the basins in $\hat{\phi} = \text{invert}(\phi)$ with the complement of the convex hull of $\mathbf{\Gamma}$. The pseudocode of the GEMS method is illustrated in Algorithm 1, and a visual comparison of the method is shown in Figure 16.

**Algorithm 1:** Geometric Median Shapes

**Input:** contour images $\mathbf{\Gamma}$

minima height $h$

**Result:** optimal closed curve $\Gamma^*$

$\phi = 0$

**for** $\Gamma_i \in \mathbf{\Gamma}$ **do**

  $\phi = \phi +$ DistanceTransform$(\Gamma_i)$

$\hat{\phi} =$ invert$(\phi)$

$B =$ WatershedBasins$(\hat{\phi}, h)$

$H =$ ConvexHull$(\mathbf{\Gamma})$

$markers = B \cup \{\Omega \setminus H\}$

$\Gamma^* =$ Watershed$(\hat{\phi}, markers)$

Return $\Gamma^*$

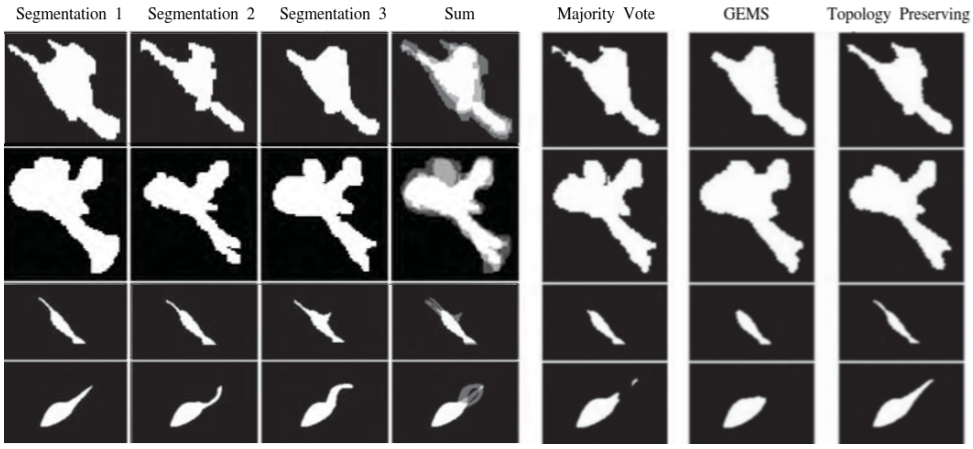
**Source:** (CUNHA, 2019) (adapted-2022)

### 3.1.6 Topology preserving segmentation fusion for cells with complex shapes

Most recently, in the (MELNIKOVA; MATULA, 2021), Melnikova and Matula propose a fusion segmentation mask algorithm focusing on preserving the topology of the inputs segmentations. Unlike the other algorithms discussed in this chapter, this algorithm focuses on simply connected segments.

The algorithm averages the protrusions regions with only the segmentations that contain the protrusions. After the calculation, the regions of protrusions are merged using spline interpolation, which creates the final segmentation. The algorithm achieved good results compared to several well-known algorithms, mainly for not producing artifacts on the borderline region, as shown in Figure 16.

Figure 16 – Example in which each line represents a set of three segmentations, followed by their sum (rescaled for visual proposes) and their segmentation fusion using Majority Vote, GEMS, and Topology Preserving algorithms.




**Source:** (MELNIKOVA; MATULA, 2021) (adapted)

# 4 IMAGE SEGMENTATION FUSION FOR CROWDSOURCING DATA

In this chapter, we focus on analyzing and mitigate the problem of crowdsourcing image segmentation platforms, as is the example of the COSE platform. In this platform, given an image, it is collected N manual segmentations made by non-experts, and it is required to create a unique segmentation for the initial image. The following experiments were implemented in Python, in which the main libraries used were: OpenCV/Scikit-image (image processing), PyTorch (Deep Neural networks) and PyTVision (package of datasets, models, and image transformations for PyTorch).

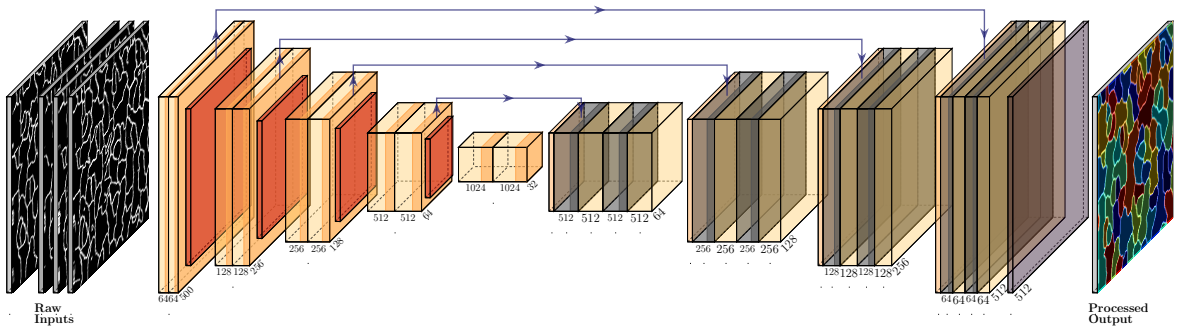
## 4.1 METHODOLOGY

Let a two-dimensional single-channel image be defined as $x : \Omega \rightarrow \mathbb{R}$, where every element $p \subset \Omega$ are called pixels. Then, the semantic segmentation task can be expressed as a function $y = H(x)$, that assigns a categorical label $y : \Omega \rightarrow 0, ..., (C-1)$ for every pixel $p$, where $C$ is the number of unique classes in a given problem. The transformation $H(x)$ can be done in different ways, such as manually, by an algorithm such as Otsu thresholding(OTSU, 1979), seeded Watershed(COELHO; SHARIFF; MURPHY, 2009), or even a Deep Neural Network (BADRINARAYANAN; KENDALL; CIPOLLA, 2017; WANG et al., 2018; RONNEBERGER; FISCHER; BROX, 2015).

Moreover, given a pool of $n$ segmentations $y_1, ..., y_n$, the goal of segmentation fusion is to merge this set into a unique segmentation $\hat{y} = F(y_1, ..., y_n)$. It is the desire that such fused segmentation must be, on average, more accurate than every individual segmentation in the pool. Nevertheless, as in any ensemble method, the final result is expected to be as good as the data are precise and diverse (CRUZ; SABOURIN; CAVALCANTI, 2018).

Similar to the image segmentation task, for learning to solve the segmentation fusion task, a training dataset $S = \{(y_{11}, ..., y_{1n}, g_1), ..., (y_{m1}, ..., y_{mn}, g_m)\}$ it is used, where for each image $x$, its segmentations $y_1, ..., y_n$ and ground truth $g$ are known, being $g$ defined as $g : \Omega \rightarrow 0, ..., C$. The function $F(.)$ can be done with different techniques, such as the well-known SIMPLE or STAPLE. However, in this experiment, we focus on using a parametric function $F_\theta$ to solve the segmentation fusion task. In other words, modeling as a neural network as illustrated in Figure 17. Therefore, finding a feasible fusion function is equivalent to solve

Figure 17 – The overall architecture, where the deep learning architecture is the U-Net(RONNEBERGER; FISCHER; BROX, 2015).




**Source:** https://github.com/HarisIqbal88/PlotNeuralNet (adapted-2022)

the optimization problem $\theta^* = \arg min_{\theta \subset \Theta} C(\theta)$, where $C(\theta)$ is the cost function for a generic loss function $\mathcal{L}(g, \hat{y})$ described in Equation 4.1.

$$C(\theta) = \frac{1}{|S|} \sum_{(y_{i1},...,y_{in},g_i) \in S} \mathcal{L}(g_i, F_\theta(y_{i1}, ..., y_{in})) \tag{4.1}$$

We evaluated three loss functions, the Cross Entropy (CE)(LONG; SHELHAMER; DARRELL, 2015), Smooth Dice (Dice)(MILLETARI; NAVAB; AHMADI, 2016), and Youden's J statistic regularization (J-REG)(PEÑA et al., 2020). The CE is an information theory-based loss function described in Equation 4.2, where $g$ and $z$ are the ground truth in one hot representation and the probability likelihood, respectively.

$$\mathcal{L}_{CE}(g, \hat{y}) = -\frac{1}{\Omega} \sum_{l=0}^{C-1} \sum_{p \subset \Omega} g_l(p) \cdot log \hat{y}_l(p) \tag{4.2}$$

Moreover, the region based loss function Smooth Dice is shown on Equation 4.3 where, in this work, we use the smooth parameter $\alpha$ equal to one, for better stability during training time. To address the highly imbalanced classes, we add weight for each class inversely proportional to the number of pixels in each class.

$$\mathcal{L}_{DICE}(g, \hat{y}) = 1 - \frac{\alpha + 2 \sum_{i=1}^{C} \hat{y}_i \cdot g_i}{\alpha + \sum_{i=1}^{C} \hat{y}_i^2 + \sum_{i=1}^{C} g_i^2} \tag{4.3}$$

Also, for the same reason, the J Regularization was evaluated. The regularization consists of the cross entropy loss with Younden Regularization as in Equation $\mathcal{L}_J = \mathcal{L}_{CE}(y, z) + \mathcal{J}(y, z)$, where the $\mathcal{J}(y, z)$ is described in Equation 4.4.

$$\mathcal{J}(g,z) = -\sum_{i=0}^{C}\sum_{k=0}^{C}\lambda_{i,k}\log\left[\frac{1}{2} + \sum_{p\subset\Omega} z_i(p)\cdot\left(\frac{g_i(p)}{2n_i} - \frac{g_k(p)}{2n_k}\right)\right] \tag{4.4}$$

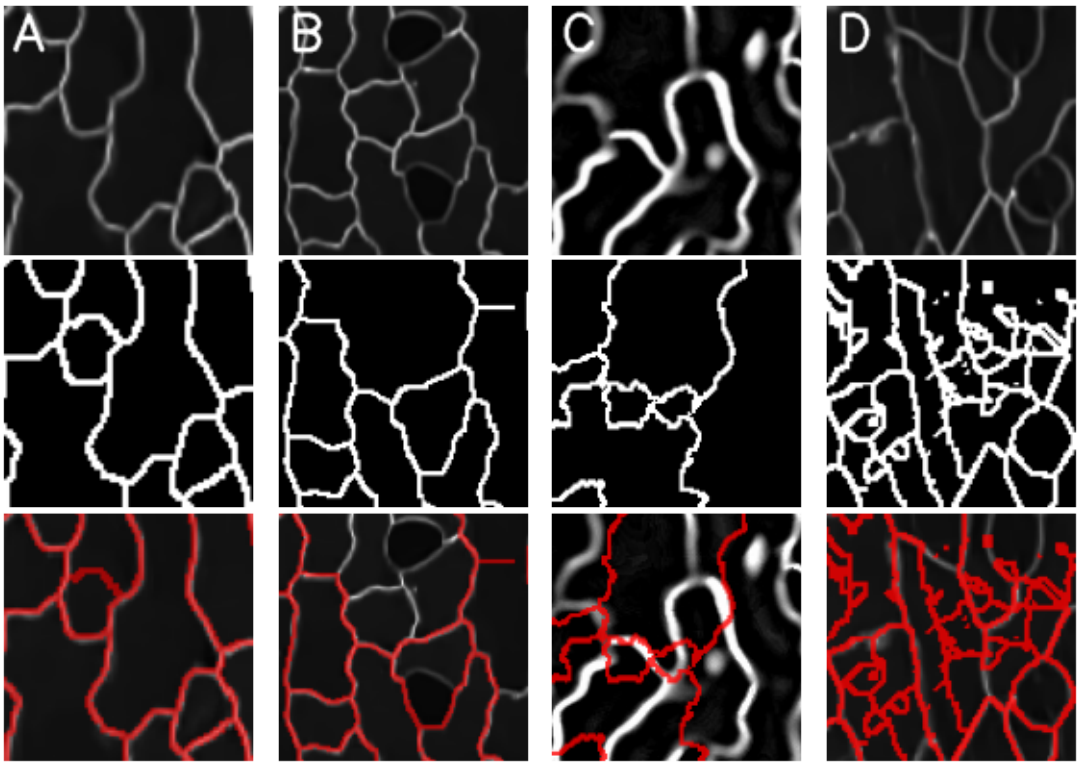
In the proposed method called UFUSION, we decided to use a modified version of the convolutional encoder-decoder U-Net (RONNEBERGER; FISCHER; BROX, 2015) architecture for the task of image segmentation fusion. the U-Net was selected due to fast and consistent results with few annotated training samples in the biomedical image segmentation task. One-pixel padding was used for every convolution operations to preserve the spatial resolution. During the training phase, random transformations such as rotating, mirroring, warping, and elastic deforming augmentations were applied. In early experiments, we also evaluate changing the input space by applying the same transformations in all input segmentation sets $\{y_{i1}, ..., y_{im}\}$. The operations are skeletonization, Gaussian blur, and distance transform. Nevertheless, in this scenario, this modification deteriorates the model performance. Another unsuccessful test was when using pre-trained Generative Adversarial Networks (GANs) as data augmentation. We believe this preprocessing can be favorable, though we leave it for future work.

## 4.2 ROBUSTNESS TASKS

To evaluate the quality of the final segmentation, we defined three robustness tasks: the presence of outliers, missing data, and deformations. These tests were determined based on annotation problems observed in the COSE software. Figure 18 shows the original image, a manual segmentation, and the overlap between the original image and the manual segmentation in four scenarios. The first scenario in Figure 18 (A) is an example of good segmentation in which all marks were performed entirely and accurately; Figure 18 (B) shows a segmentation in which several borders were not marked, although the existing markings were annotated precisely; Moreover, in Figure 18 (C), the markings shown are unrelated to the original image and thus increase the fusion task difficulty; Finally, in Figure 18 (D), a noise over-segment sample, despite having some correlation with the original image due to the high noise, is also a bad segmentation example.

**Outliers Test**. This test evaluates the robustness of the algorithms with a percentage of unrelated data. This unrelated data comes from a random selection of three sources: the evaluated dataset, a different cell segmentation dataset, and hand-drawn geometric figures, as shown in Figure 20a. In real scenarios, this unexpected segmentation is inserted into the

Figure 18 – Sample of data collected in COSE system, in which the top row is the original image, the middle row is a manual segmentation, and the bottom row overlaps the top row and its manual segmentation. Here are good segmentation (A), under segmentation (B), outlier segmentation (C), and over-segmentation in the presence of salt noise (D).



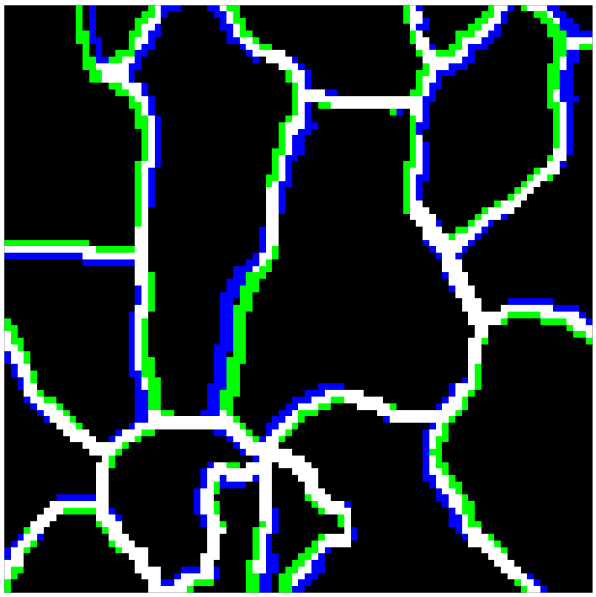

**Source:** The author (2022)

system by users who do not have proper knowledge of the tool or by malicious users, as can be seen in Figure 18 (C).

**Missing Data**. In some segmentations, the result may come with a lack of some markings as in Figure 20b. Some factors that can increase the incidence may be the presence of dark regions, lack of specific knowledge in some areas (e.g., tiny cells), and even fatigue and inattention of the annotator; an example of missing data can be seen in Figure 18 (B).

**Deformations**. Even though two segmentations have a pixel-level disagreement, both may be good results, as shown in Figure 19. This behavior may happen in regions where the border is too thick or due to precision problems of the annotator. To simulate this behavior, we use elastic distortion(SIMARD et al., 2003) for every segmentation $y$, in which, Figure 20a shows an example of two segmentations with a low and high level of elastic distortion.

Figure 19 – An example of two 'good' segmentation (green and blue) overlapped, where the white pixels denote the agreement of both.

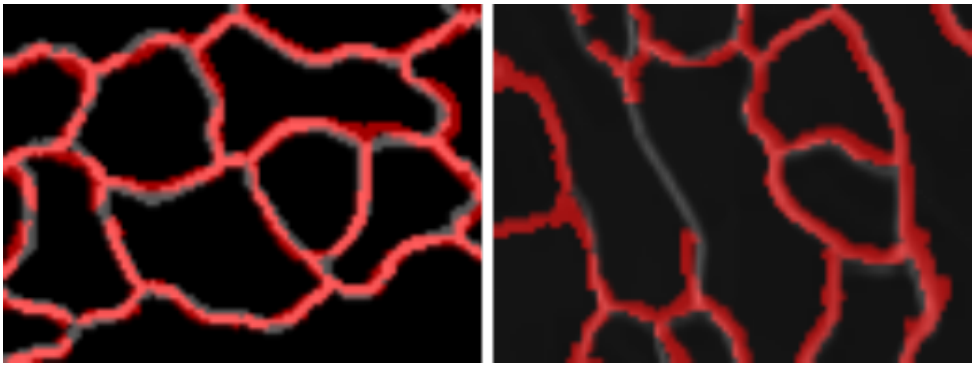


**Source:** The author (2022)

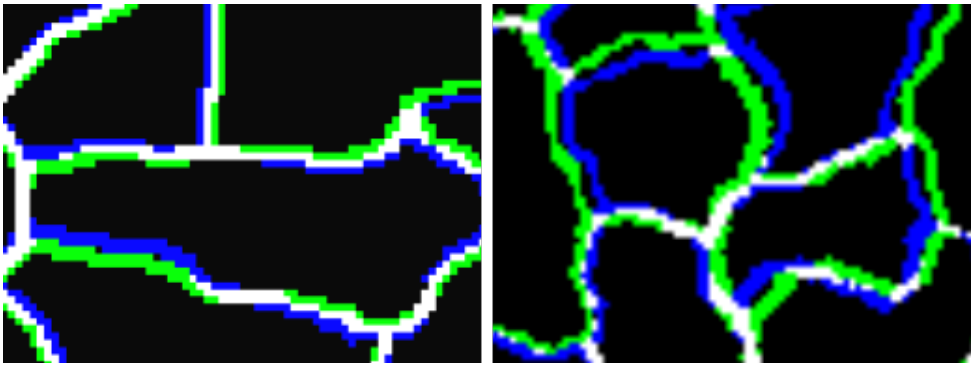
Figure 20 – Sample images of the robustness task.

(a) Outliers test: Unrelated segmentation.

(b) Missing data. The segmentation is represented in red, in which some parts are erased.



(c) Deformations test. Visual comparison between a segmentation (green) and the elastic distortion of it (blue) where the white areas denote unaltered pixels.

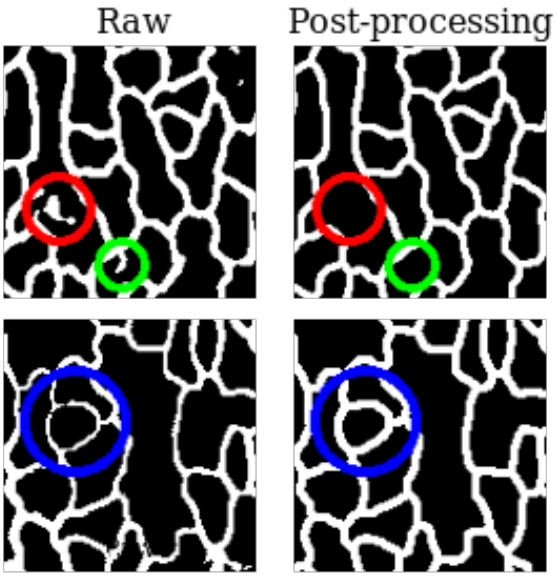


**Source:** The author (2022)

## 4.3 POST PROCESSING

For each algorithm, we applied the following procedure: first, if the output segmentation is multichannel, it is used the Maximum a Posteriori (MAP), where the given raw segmentation $y$ with $c$ channels, and an output segmentation $y'$ is calculated with $y' = \arg\max_c y(c)$. Figure 21 illustrates the subsequent post-processing operations, in which, the red, green, and blue circles exemplify: the removal of small regions with a morphological opening operation; the removal of unclosed touch region; and finally, the width of the edges is ensured.

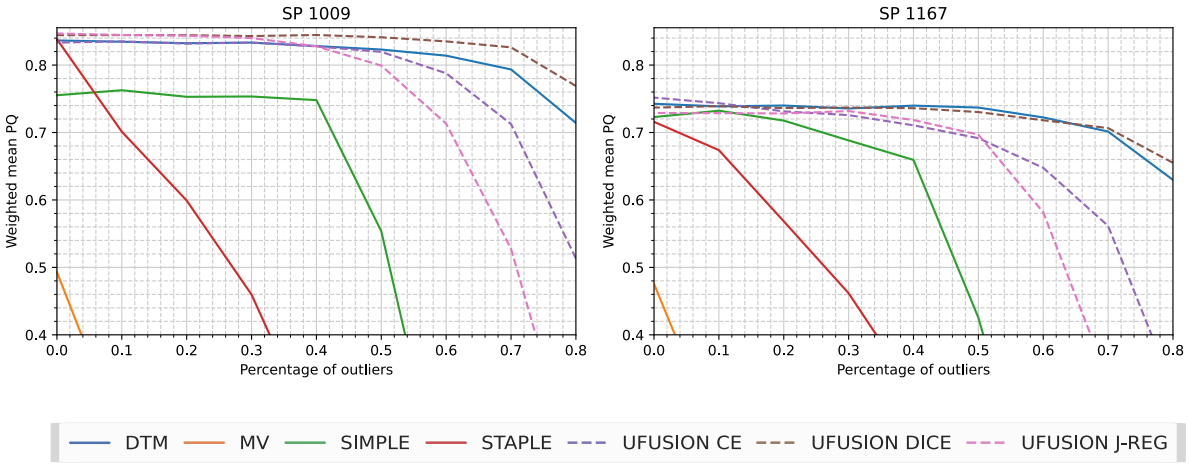
Figure 21 – Example of the post-processing function, where the red, green, and blue regions show: removed unconnected touch region, removed unclosed touch region, and the standardization of edges width, respectively.



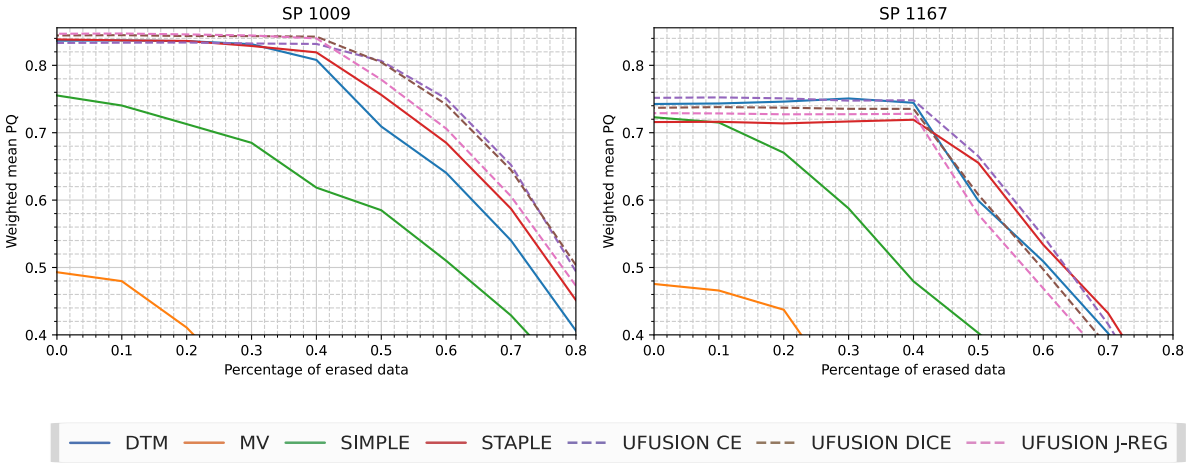

**Source:** The author (2022)

Figure 22 – Results for the three tests, presence of outliers, missing data, and deformations for two segmentation problems (SP) 1009 and 1167, in which dashed lines represent the proposed method. In all three cases, the further to the right on the x axis, the more difficult it is to obtain the correct segmentation fusion.

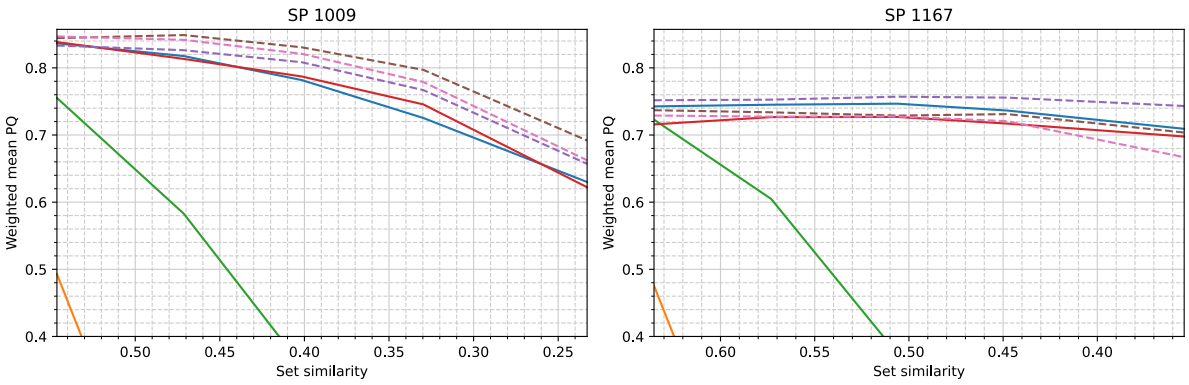
(a) Outliers test




(b) Missing data




(c) Deformations test




**Source:** The author (2022)

## 4.4 RESULTS

We compare the proposed method, UFUSION, with four algorithms: Majority Vote (MV), STAPLE(WARFIELD; ZOU; WELLS, 2004)[1], SIMPLE(LANGERAK et al., 2010), and DTM(BARBOSA; REN, 2014) with the same post-processing described in 4.3. All networks are guaranteed to begin with the same random set of weights, and the parameters of the Table 1. Random data augmentation is also guaranteed to be the same during training time. The Table 2 shows the data augmentation operations performed as well as their parameters. In the initial experiments, it was detected that the results of all tests were degraded when reducing the elastic distortion. In addition, we tested operations such as random foreground erasing, and adding random lines, but despite having better results in a given test, the others results were drastically impacted.

To measure the final instance segmentation quality, we adopted the metric PQ (KIRILLOV et al., 2019) weighted by the number of cells in the given image. The metric consists of multiplication between the average Intersection over Union (IoU) of the matched segments, called Segmentation Quality (SQ), and the F1 score of the detection, called by Recognition Quality (RQ). For the deformation test, we used as the similarity distance the average PQ between all segmentations in the input set.

Table 1 – Parameters used in this experiment for training neural networks

| Option | Value |
|---|---|
| Architecture | U-Net |
| Learning Rate | $10^{-5}$ |
| Optimizer | Adam |
| Scheduler | Fixed |
| Epochs | 200 |
| Post Processing | MAP |

**Source:** The author (2022)

Two sepal cells datasets, the Segmentation Problem 1009 and the Segmentation Problem 1167, were collected to evaluate the performance of the algorithms. They were imaged on a Zeiss LSM 880 Confocal laser scanning with a 20x magnification lens resulting in a pixel

[1] Initially, we evaluate two STAPLE implementations, https://pypi.org/project/staple and https://simpleitk.org, although the first implementation requires high computational cost, in our scenario, it produces better results, and due to that, we use this implementation in the results section.

Table 2 – Transformations used in neural network training and their respective parameters, where $\alpha$ is a scaling factor and $\sigma$ is the Gaussian of standard deviation of the Elastic Distortion algorithm (SIMARD et al., 2003).

| Random Operation | Parameters |
|---|---|
| Scale | max=30% |
| Flip | prob=50% |
| Rotate | prob=50%, max_angle=45 |
| Translate | prob=50%, max=20% |
| Warp Perspective | prob=50%, max=2% |
| Elastic Distortions | prob=50%, $\alpha$=32, $\sigma$=12 |

**Source:** The author (2022)

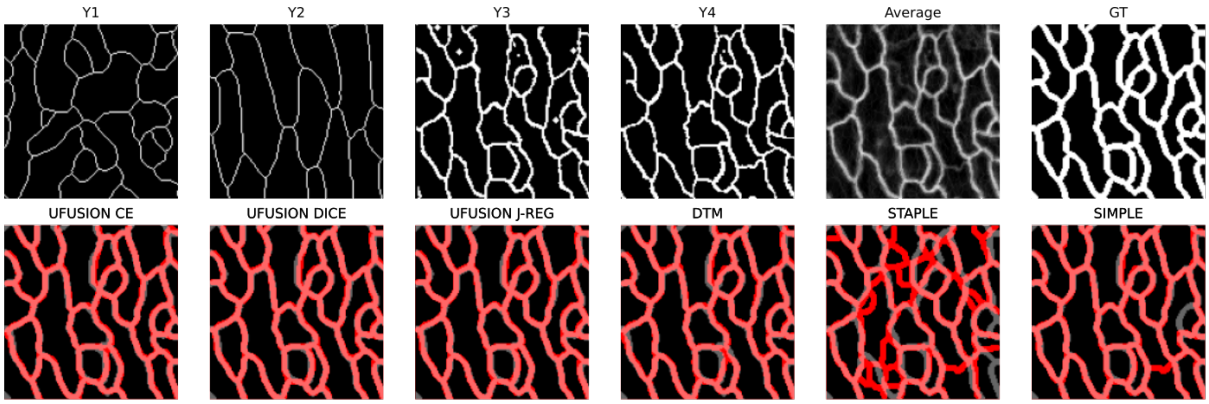
size of 0.621 x 0.621 microns. Furthermore, both datasets were artificially modified (noise, deformations) to simulate more challenging scenarios.

Figure 22 shows the results of the segmentation fusion methods. As can be observed, all approaches except majority vote and SIMPLE achieved close results in low noise situations (leftmost part of the graphs). The STAPLE algorithm showed comparable results for missing data and deformation test scenarios, showing significant differences only for higher deformation values. However, differently from our proposal, the STAPLE algorithm suffered a severe degradation of the performance for the outlier test. This drop of performance happens due to the wrong fusion of both the correct segmentation set and the outlier set, forming an image with many False Positives (foreground pixels from every input segmentation), which results in segmentation with many small regions. Moreover, the SIMPLE and the Majority Vote indicated a high decay along with the given task. And finally, the proposed method, UFUSION, and the DTM achieved the best results in this evaluation with an advantage to UFUSION with Smooth DICE loss function. Note that, since there are different sources of outliers, the data central tendency is still maintained even with more than 50% of outliers.

The Figure 23, 24 and 25 illustrates four samples $Y_1, Y_2, Y_3, Y_4$ of the input set, the average of all inputs, the ground-truth (GT), and the segmentation fusion output. Note that even in this high noise scenario the proposed method makes fewer mistakes. It avoids some segmentations error, such as cells inside other cells, which is not present in this dataset.

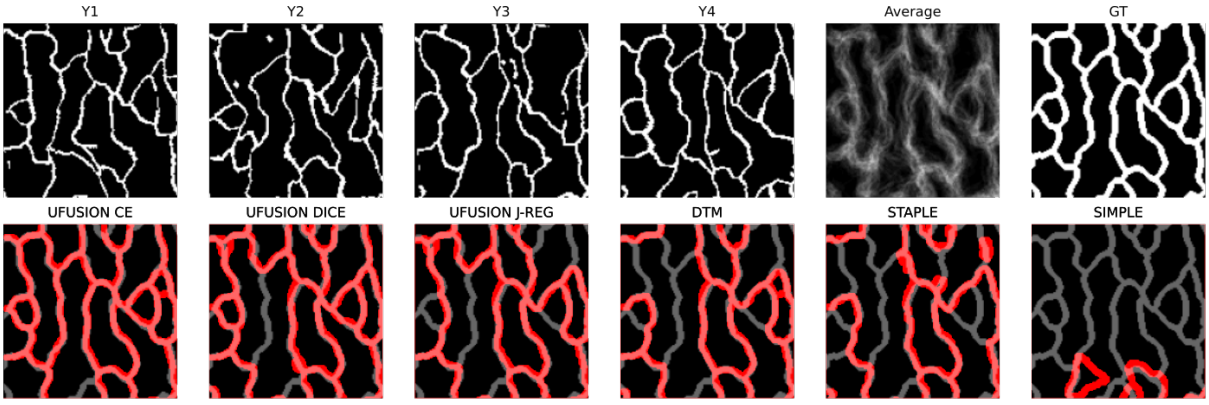
Figure 25 shows a case of random erasing, in which the probability of a pixel being erased is proportional to its coordinates, where the upper right pixel has the lowest probability of being erased and the lower left pixel the highest probability. In addition to having a more precise

Figure 23 – Example of the outlier test, whose top row $Y_1, ..., Y_4$ are samples of inputs segmentation, followed by their average, the ground truth (GT), note that, the $Y_1$ and $Y_2$ represent outliers segmentations. The bottom row is shows the post processed output of the input segmentation set $S$, in which UFUSION is the proposed model.




**Source:** The author (2022)

Figure 24 – Example of the deformation test, whose top row $Y_1, ..., Y_4$ are samples of inputs segmentation $S$ with elastic distortion, followed by their average and ground truth segmentation GT. Furthermore, bottom row shows the segmentation performed by the evaluated algorithms, in which UFUSION is the proposed model.

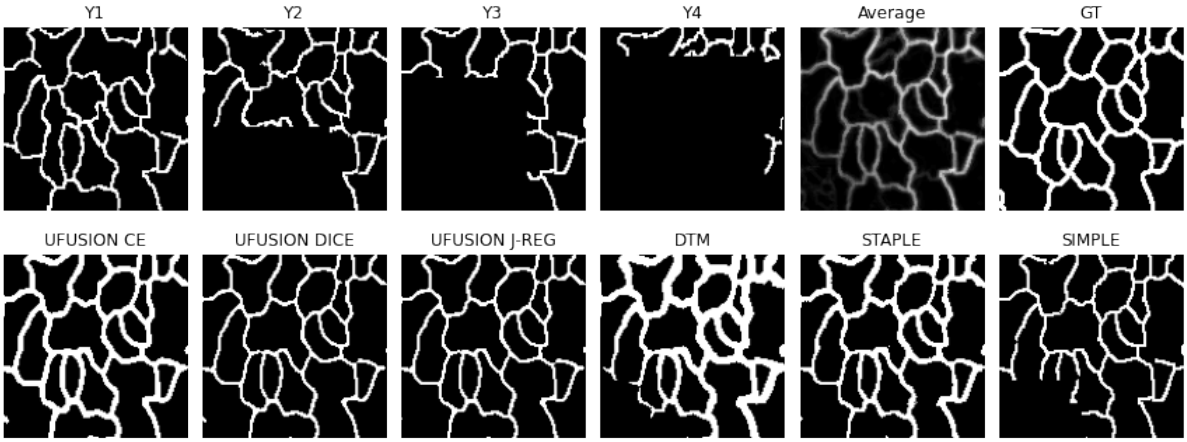



**Source:** The author (2022)

segmentation throughout the whole image, the proposed methods did not change the edge thickness along the image, indicating a more stable algorithm for this test.

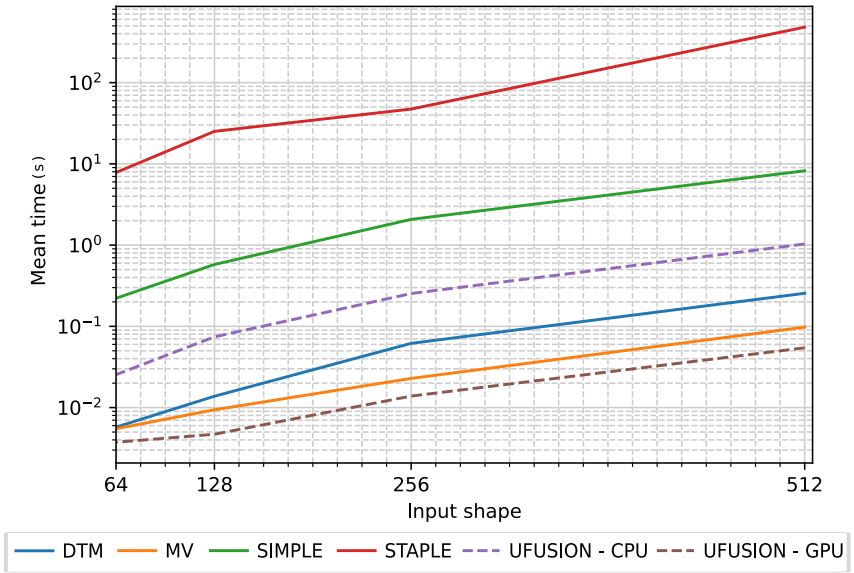
Finally, the evaluation time is shown in Figure 26, as the evaluation time of the neural networks are variants to the architecture but not the loss function, the proposed method is shown as a unique label UFUSION. The DTM, MV, SIMPLE, STAPLE, and UFUSION-CPU were measured using an AMD Ryzen 5 5600X, and for the UFUSION - GPU an NVIDIA RTX 3060 TI, the GPU time accounts the copy process from RAM memory to GPU memory, and vice versa. Although real-time performance is not mandatory, SIMPLE and STAPLE require

Figure 25 – Example of the missing data, whose top row $Y_1, ..., Y_4$ are samples of inputs segmentation $S$ where each segmentation has a probability of being partially erased. followed by their average and ground truth segmentation GT. Furthermore, bottom row shows the raw segmentation performed by the evaluated algorithms, in which UFUSION is the proposed model.




**Source:** The author (2022)

Figure 26 – The average processing time of the fusion methods along with the input segmentation shape.




**Source:** The author (2022)

more than 80x, 4000x time of the majority vote, respectively, which is not convenient for this task.

# 5 CONCLUSION

In this work, we adapt the deep learning U-Net architecture for segmentation fusion, focusing on high disagreement and noise environments. To evaluate the behavior of the adapted network, we defined three robustness tasks based on real problems found in the COSE crowdsourcing platform, which are the robustness of outliers segmentation, missing data, and deformations.

According to the explored scenarios, in high disagreement environments, the most promising method was using the proposed deep learning architecture with the smooth dice loss function. In cases of low disagreement, despite the proposed method obtaining the best results, due to the training process required by the neural network, we believe that the DTM method obtained the best tradeoff.

## 5.1 CONTRIBUTIONS

- An adaptation of the U-Net neural network for segmentation fusion, focusing on high disagreement and noise environment. The proposed model aims at replacing an existing segmentation fusion method of the COSE crowdsourcing platform for having better results in situations of high diversity that happen on crowdsourcing platforms.
- Analyzing several well-known fusion algorithms in three tasks: outliers, missing data, and deformations, that are based on the problems encountered in crowdsource segmentation platforms.
- The published work: **Carlos H. C. Pena**, Tsang Ing Ren, Pedro D. Marrero Fernandez, Fidel A. Guerrero-Peña and Alexandre Cunha. "An Ensemble Learning Method for Segmentation Fusion." The 2022 International Joint Conference on Neural Networks (IJCNN). IEEE, 2022.

## 5.2 FUTURE WORKS

To analyze the results of this work, we found that the segmentation pool cannot always provide a good intuition about optimal segmentation. We often resort to looking at the original image to obtain a clearer idealization of the segmentation. For this reason, we plan to

concatenate more information into the network input besides the segmentation pool, in which the original image is one of the possible sources of information. However, we expected that with the original image, the network would improve the overall result at the cost of greater dependence on the training dataset. We also aim to attach a regression neural network to the beginning of the proposed method to weight each segmentation individually and compare it with an attention-based architecture.

Furthermore, we are also interested in the unsupervised learning process. (GUERRERO-PEÑA et al., 2019) showed that deep neural networks could have excellent results even when trained with weakly supervised data. We want to extend their work with fully unsupervised data.

# REFERENCES


BADRINARAYANAN, V.; KENDALL, A.; CIPOLLA, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 39, n. 12, p. 2481–2495, 2017.

BARBOSA, P. A. L.; REN, T. I. *COSE: sistema de segmentação de imagens colaborativo*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2014.

BEUCHER, S.; MEYER, F. The morphological approach to segmentation: the watershed transformation. In: *Mathematical morphology in image processing*. [S.l.]: CRC Press, 2018. p. 433–481.

BEUCHER, S. et al. The watershed transformation applied to image segmentation. *SCANNING MICROSCOPY-SUPPLEMENT-*, Scanning Microscopy International, p. 299–299, 1992.

CAICEDO, J. C.; GOODMAN, A.; KARHOHS, K. W.; CIMINI, B. A.; ACKERMAN, J.; HAGHIGHI, M.; HENG, C.; BECKER, T.; DOAN, M.; MCQUIN, C. et al. Nucleus segmentation across imaging experiments: the 2018 data science bowl. *Nature methods*, Nature Publishing Group, p. 1–7, 2019.

CANNY, J. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, Ieee, n. 6, p. 679–698, 1986.

COELHO, L. P.; SHARIFF, A.; MURPHY, R. F. Nuclear segmentation in microscope cell images: a hand-segmented dataset and comparison of algorithms. In: IEEE. *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. [S.l.], 2009. p. 518–521.

CORDTS, M.; OMRAN, M.; RAMOS, S.; REHFELD, T.; ENZWEILER, M.; BENENSON, R.; FRANKE, U.; ROTH, S.; SCHIELE, B. The cityscapes dataset for semantic urban scene understanding. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 3213–3223.

CRUZ, R. M.; SABOURIN, R.; CAVALCANTI, G. D. Dynamic classifier selection: Recent advances and perspectives. *Information Fusion*, Elsevier, v. 41, p. 195–216, 2018.

CUNHA, A. Gems-geometric median shapes. In: IEEE. *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*. [S.l.], 2019. p. 1492–1496.

DUIN, R.; KITTLER, J.; HATEF, M.; MATAS, J. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 20, n. 3, p. 226–239, 1998.

DUIN, R. P. The combining classifier: to train or not to train? In: IEEE. *Object recognition supported by user interaction for service robots*. [S.l.], 2002. v. 2, p. 765–770.

FERRARA, M.; FRANCO, A.; MAIO, D.; MALTONI, D. Face image conformance to iso/icao standards in machine readable travel documents. *IEEE Transactions on Information Forensics and Security*, IEEE, v. 7, n. 4, p. 1204–1213, 2012.

GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006. ISBN 013168728X.

GUERRERO-PEÑA, F. A.; FERNANDEZ, P. D. M.; REN, T. I.; CUNHA, A. A weakly supervised method for instance segmentation of biological cells. In: *Domain Adaptation and Representation Transfer and Medical Image Learning with Less Labels and Imperfect Data*. [S.l.]: Springer, 2019. p. 216–224.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 770–778.

IGLOVIKOV, V.; SHVETS, A. Ternausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation. *arXiv preprint arXiv:1801.05746*, 2018.

KIRILLOV, A.; HE, K.; GIRSHICK, R.; ROTHER, C.; DOLLÁR, P. Panoptic segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2019. p. 9404–9413.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, v. 25, 2012.

LANGERAK, T. R.; HEIDE, U. A. van der; KOTTE, A. N.; VIERGEVER, M. A.; VULPEN, M. V.; PLUIM, J. P. Label fusion in atlas-based segmentation using a selective and iterative method for performance level estimation (simple). *IEEE transactions on medical imaging*, IEEE, v. 29, n. 12, p. 2000–2008, 2010.

LI, C.; TAM, P. K.-S. An iterative algorithm for minimum cross entropy thresholding. *Pattern recognition letters*, Elsevier, v. 19, n. 8, p. 771–776, 1998.

LIN, T.-Y.; MAIRE, M.; BELONGIE, S.; HAYS, J.; PERONA, P.; RAMANAN, D.; DOLLÁR, P.; ZITNICK, C. L. Microsoft coco: Common objects in context. In: SPRINGER. *European conference on computer vision*. [S.l.], 2014. p. 740–755.

LONG, J.; SHELHAMER, E.; DARRELL, T. Fully convolutional networks for semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2015. p. 3431–3440.

MAAS, A. L.; HANNUN, A. Y.; NG, A. Y. et al. Rectifier nonlinearities improve neural network acoustic models. In: ATLANTA, GEORGIA, USA. *Proc. icml*. [S.l.], 2013. v. 30, n. 1, p. 3.

MACQUEEN, J. et al. Some methods for classification and analysis of multivariate observations. In: OAKLAND, CA, USA. *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. [S.l.], 1967. v. 1, n. 14, p. 281–297.

MELNIKOVA, A.; MATULA, P. Topology preserving segmentation fusion for cells with complex shapes. In: IEEE. *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*. [S.l.], 2021. p. 204–207.

MEYER, F. Color image segmentation. In: IET. *1992 international conference on image processing and its applications*. [S.l.], 1992. p. 303–306.

MILLETARI, F.; NAVAB, N.; AHMADI, S.-A. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: IEEE. *2016 fourth international conference on 3D vision (3DV)*. [S.l.], 2016. p. 565–571.

OTSU, N. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, IEEE, v. 9, n. 1, p. 62–66, 1979.

PEÑA, F. A. G.; FERNANDEZ, P. D. M.; TARR, P. T.; REN, T. I.; MEYEROWITZ, E. M.; CUNHA, A. J regularization improves imbalanced multiclass segmentation. In: IEEE. *2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI)*. [S.l.], 2020. p. 1–5.

RIDLER, T.; CALVARD, S. et al. Picture thresholding using an iterative selection method. *IEEE trans syst Man Cybern*, v. 8, n. 8, p. 630–632, 1978.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: SPRINGER. *International Conference on Medical image computing and computer-assisted intervention*. [S.l.], 2015. p. 234–241.

SHVETS, A. A.; RAKHLIN, A.; KALININ, A. A.; IGLOVIKOV, V. I. Automatic instrument segmentation in robot-assisted surgery using deep learning. In: IEEE. *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. [S.l.], 2018. p. 624–628.

SIMARD, P. Y.; STEINKRAUS, D.; PLATT, J. C. et al. Best practices for convolutional neural networks applied to visual document analysis. In: *Icdar*. [S.l.: s.n.], 2003. v. 3, n. 2003.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

STREET, W. N.; WOLBERG, W. H.; MANGASARIAN, O. L. Nuclear feature extraction for breast tumor diagnosis. In: SPIE. *Biomedical image processing and biomedical visualization*. [S.l.], 1993. v. 1905, p. 861–870.

TAN, T. Y.; ZHANG, L.; LIM, C. P.; FIELDING, B.; YU, Y.; ANDERSON, E. Evolving ensemble models for image segmentation using enhanced particle swarm optimization. *IEEE access*, IEEE, v. 7, p. 34004–34019, 2019.

THAMBAWITA, V.; HICKS, S. A.; HALVORSEN, P.; RIEGLER, M. A. Divergentnets: Medical image segmentation by network ensemble. *arXiv preprint arXiv:2107.00283*, 2021.

WANG, S.; HUA, Y.; CAO, Y.; SONG, T.; XUE, Z.; GONG, X.; WANG, G.; MA, R.; GUAN, H. Deep learning based fetal middle cerebral artery segmentation in large-scale ultrasound images. In: IEEE. *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. [S.l.], 2018. p. 532–539.

WARFIELD, S. K.; ZOU, K. H.; WELLS, W. M. Simultaneous truth and performance level estimation (staple): an algorithm for the validation of image segmentation. *IEEE transactions on medical imaging*, IEEE, v. 23, n. 7, p. 903–921, 2004.

ZHANG, T. Y.; SUEN, C. Y. A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, ACM New York, NY, USA, v. 27, n. 3, p. 236–239, 1984.