Bachelor's Degree Thesis

Degree in Data Science and Engineering

Cell detection and classification of breast cancer histology images using a deep learning approach based on the U-Net architecture

Júlia Sala Prat

Supervised by Philippe Salembier and Montse Pardàs

Signal Theory and Communications Department

June, 2021

Facultat d'Informàtica de Barcelona (FIB)
Facultat de Matemàtiques i Estadística (FME)
Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona (ETSETB)

Acknowledgments

The great work that DigiPATICS carries out could help all the laboratory scientists who spend hours counting and classifying cells visually and above all, help breast cancer patients receive the most appropriate treatment for their case. As a woman, working in investigation in a field such as breast cancer has made me feel very good about myself. It has been a great honor to work in DigiPATICS: the acquired knowledge, the experience working with other students and teachers, discussing solutions in a cooperative way and explaining new discoveries, all to achieve the same goal. I would like to thank the whole group and, especially, my director and co-director, who have guided me throughout the whole process.

Abstract

This project deals with the creation of an algorithm capable of detecting and classifying cell nuclei in histology images from breast cancer patients. These images, provided by the Catalan Institute of Health (ICS), were stained to detect the Ki67 and ER (estrogen receptor) antigens that allow the pathologists to diagnose patient disease gravity and also specify the proper treatment. The algorithm has been developed using the Ki67 antigen database and it consists of 3 different blocks. Block A uses a U-Net architecture to develop a semantic segmentation algorithm that detects and classifies sets of cells pixel-by-pixel. Block B also uses a U-Net architecture but this time, to develop a regression algorithm that detects the cell centers. A final step, block C, combines both results to identify each individual cell using a watershed algorithm to split the connected components from A in as many centers as block B predicts. The limited number of images in the database, the lack of balance in the cell classes and the far from perfect quality of the ground truth images make the task of training a deep learning algorithm complex. However, an exhaustive optimization of the three blocks is performed and the final algorithm reaches a cell based weighted F-score of 0.7811. The Ki67 score that gives pathologists the notion of how serious the disease is, is predicted with a mean absolute error of 3.68%. When generalizing to the ER database, with a slight modification in the hyperparameters, the algorithm reached a weighted F-score of 0.8183 with 1.72% of error in ER score.

Resum

Aquest projecte consisteix en la creació d'un algoritme capaç de detectar i classificar nuclis cel·lulars en imatges d'histologia de pacients de càncer de mama. Aquestes imatges, proporcionades per l'Institut Català de la Salut (ICS), s'han tenyit per tal de detectar les proteïnes Ki67 i RE (receptor d'estròGens) que permeten, als patòlegs, diagnosticar la gravetat de la malaltia de la pacient i especificar el tractament més adequat. L'algoritme es desenvolupa utilitzant la base de dades de l'antigen Ki67 i consisteix en 3 blocs. El bloc A utilitza l'arquitectura U-Net per desenvolupar un algoritme de segmentació semàntica que detecta i classifica conjunts de cèl·lules píxel a píxel. El bloc B també utilitza l'arquitectura U-Net però, aquesta vegada, per desenvolupar un algoritme de regressió que detecta els centres de les cèl·lules. Un últim pas, el bloc C, combina els dos resultats anteriors per identificar cada cèl·lula de forma individual utilitzant un algoritme de watershed (conca hidrogràfica) per partir les cèl·lules que s'han unit en una sola component connexa del bloc A en tants centres com hagi predit el bloc B. El nombre limitat d'imatges de la base de dades, la falta d'equilibri en les classes de cèl·lules i la qualitat, lluny de ser perfecta, de les imatges del ground truth fan complexa la tasca d'entrenar un algoritme d'aprenentatge profund. Tot i això, es realitza una optimització exhaustiva dels tres blocs i l'algoritme final aconsegueix un F-score ponderat basat en cèl·lules de 0.7811. La puntuació Ki67 que dóna als patòlegs la noció de la gravetat de la malaltita es prediu amb un error absolut mitjà del 3.68%. Al generalitzar a la base de dades de RE, amb una lleugera modificació en els hiperparàmetres, l'algoritme arriba a un F-score ponderat de 0.8183 amb un 1.72% d'error en la puntuació RE.

Resumen

Este proyecto consiste en la creación de un algoritmo capaz de detectar y clasificar núcleos celulares en imágenes histológicas de pacientes con cáncer de mama. Estas imágenes, proporcionadas por el Instituto Catalán de la Salud (ICS), se han teñido para detectar las proteínas Ki67 y RE (receptor de estrógenos) que permiten a los patólogos diagnosticar la gravedad de la enfermedad de la paciente y especificar el tratamiento más adecuado. El algoritmo se ha desarrollado utilizando la base de datos del antígeno Ki67 y consta de 3 bloques diferentes. El bloque A utiliza una arquitectura U-Net para desarrollar un algoritmo de segmentación semántica que detecta y clasifica conjuntos de células píxel a píxel. El bloque B también utiliza una arquitectura U-Net pero, esta vez, para desarrollar un algoritmo de regresión que detecta los centros de las células. Un último paso, el bloque C, combina ambos resultados para identificar cada célula individualmente utilizando un algoritmo de watershed (cuenca hidrográfica) para dividir las células que se han unido en una sola componente conexa del bloque A en tantos centros como haya predicho el bloque B. El limitado número de imágenes en la base de datos, la falta de equilibrio en las clases de células y la calidad nada perfecta de las imágenes del ground truth hacen compleja la tarea de entrenar un algoritmo de aprendizaje profundo. Sin embargo, se realiza una optimización exhaustiva de los tres bloques y el algoritmo final alcanza un F-score ponderado basado en células de 0.7811. La puntuación Ki67 que da a los patólogos la noción de la gravedad de la enfermedad se predice con un error absoluto medio del 3.68%. Al generalizar a la base de datos de RE, con una ligera modificación en los hiperparámetros, el algoritmo alcanza un F-score ponderado de 0.8183 con un 1.72% de error en la puntuación de RE.

Index

1. Introduction

This project is based on the creation of an efficient and accurate algorithm to detect and classify tumor cell nuclei on breast cancer histology images to reduce the tiresome task of visual classification executed by pathologists and minimize the risk of misdiagnosis and, consequently, its mistreatment

In order to make the objectives of the project more understandable, an explanation of the medical context in which it is situated is necessary. The objectives of the project will then be discussed. To end the introduction, the working environment of the project is contextualized, i.e. the group in which we have participated during the whole period of this project.

1.1 Medical background

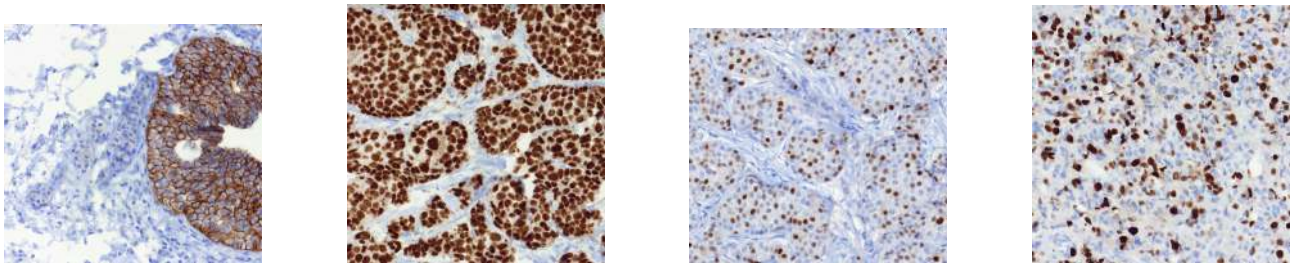
Breast cancer is a malignant proliferation of the epithelial cells surounding the breast ducts or lobules. It is a clonal disease; where an individual cell resulting from a series of mutations acquires the ability to divide without control or order, causing it to reproduce until it forms a tumor. The resulting tumor, which begins as a mild abnormality, becomes severe, invades neighboring tissues and eventually spreads to other parts of the body.
In 2020, 34.088 new cases of breast cancer were diagnosed in Spain, according to the European Cancer Information System (ECIS). It is the most frequent type of tumor among women in the country and the most diagnosed in the world, surpassing lung cancer according to data from the International Agency for Research on Cancer (IARC) in 2021.

The diagnosis of breast cancer is confirmed by analyzing histological tissues from the suspicious area. A biopsy of the area is taken, stored in paraffin so that thin sections can be made and placed on slides. Once the samples are in the slides the staining process can start [9].

The most typical stains for the tissues are hematoxylin and eosin (H&E) and immunohistochemistry (IHC). H&E reveals a considerable amount of microscopic anatomy and can be used to diagnose a wide range of histopathologic conditions. However, it does not always provide enough contrast to differentiate all tissues, cellular structures, or the distribution of chemical substances.
IHC is the process of antigen identification in tissue cells by taking advantage of the fact that antibodies bind to antigens in biological tissues. Specifically, the IHC staining process starts with an incubation of the sample with concrete antibodies to detect a single protein. These antibodies may be associated with a color-giving enzyme or may be detected by a secondary antibody that is associated with a color-giving enzyme. The nuclei containing more antigens bind more antibodies and thus, the color given by te enzyme will be different. This is not only useful to detect the antigen but also the amount of antigen the nuclei has.
The main biomarkers for breast cancer in IHC are HER2, the estrogen receptor (ER), progesterone receptor (PR) and Ki67 antigens. In Figure 1, an image for each kind of protein can be observed.

- HER2 (from human epidermal growth factor receptor 2): A protein located on the surface of breast cells that promotes growth. Normally, HER2 receptors help control the way a healthy breast cell grows, divides and repairs itself. But in 10% to 20% of breast cancer cases, the HER2 gene does not work properly and makes many copies of itself (this is known as HER2 gene amplification) [8].

- ER: Proteins found inside the cell that are activated by the estrogen molecule. The binding of estrogen to the membrane stimulates the proliferation of breast cells and results in increased cell division and DNA replication that can lead to mutations and thus give rise to a cancerous cell.
- PR: Protein found inside the cells that is activated by the hormone progesterone to help regulate several normal cellular functions, including cell proliferation. In 60-70% of breast cancer cases, this biomarker is overexpressed [6].
- Ki67: A cancer antigen considered a good marker of proliferation because it is found in growing and dividing cell nuclei but absent when cells are not growing. Tumors with higher levels of Ki67 may have worse prognosis but may respond particularly well to chemotherapy since it attacks all rapidly growing cells.



(a) HER2 (b) ER (c) PR (d) Ki67

Figure 1: Stained images of breast cancer tissues

Numerous procedures in biology and medicine require this kind of cell counting and detection which, being normally performed by visual estimation, is an extremely tedious and time-consuming task. However, it can be the key point to help doctors determine whether the cancer has spread beyond the breast and, in turn, determine which treatements are most likely to be effective for each patient.

1.2 Project goals

The purpose of this project is to help pathologists create a statistical analysis by precisely detecting and classifying each cell nucleus given histology images in the context of breast cancer.

The project main goals are:

1. Development of a PyTorch platform to train a model able to detect and classify cell nuclei in histology images.
2. Creation of an algorithm capable of identifying the individual cell nucleus' shape, position and class.
3. Creation of an algorithm to detect and classify tumor zones for three different markers: the Ki67 antigen, the estrogen receptor (ER) and the progesterone receptor (PR). Unfortunately, the ground truth images for the progesterone receptor biomarker could not be achieved during the course of this project. Therefore, the objective was reduced to the Ki67 and the ER biomarkers.
4. Quantification of the classification results.

1.3 DigiPATICS

The project is carried out in the Signal Theory and Communications department, specifically as a part of the DigiPATICS project, created to help with breast cancer diagnosis in the Catalan Institute of Health (ICS)'s network of hospitals. It deals with image digitalization and computer vision algorithms to optimize the diagnosis time and the treatment applied to each patient.
The UPC Image and Video Processing Group works on the development of computer vision algorithms, which is a key point in the support of the pathologist's diagnostic. The efforts were divided depending on the type of protein used for breast cancer prognosis.
On one hand, there is the HER2, which is membranous and requires different approaches when compared to typical machine learning algorithms that deal with histology images.
On the other hand, there are the ER, PR and Ki67 proteins, which stain the nuclei giving rise to more traditional models.
The DigiPATICS goal is to deal with immunohistochemistry (IHC) images, commonly used in the ICS hospitals.

2. State-of-the-art

2.1 Semantic segmentation

Computer Vision is a scientific field that faces the goal of making computers understand what a digital image or video is showing as easily as the human visual system can. Depending on the level of granularity of computer understanding, different problems can be defined. For example, image classification aims to return a discrete label of which object is presenting a given image. In classification and localization, the computer should also return a bounding box defining where the object is located in the image. Object detection is similar to the previous technique but has more than one object in an image.
This project deals with the semantic segmentation problem, where the goal is to label each pixel of an image with the class of what it is representing. In particular, the algorithm should return an image of the same size as the original one, where each pixel has been assigned one of the classes of the dataset. Due to this classification, cell nuclei can be defined as the connected components having the same label.

Semantic Segmentation has a lot of applications like autonomous driving, medical image diagnosis, handwriting recognition or face recognition. Before deep learning emerged, classical machine learning techniques like SVM, Random Forest or K-means Clustering were used to solve this kind of problem. However, deep learning has works considerably better and has supposed an exponential advance in Computer Vision in the last decade. One of the most used techniques to solve image segmentation problems is the U-Net network, which is the one used during this project.

2.2 U-Net

The U-Net is a convolutional neural network developed for biomedical image segmentation. It is based on the so-called "fully convolutional network" [7] and was created by Olaf Ronneberger, Philipp Fischer, Thomas Brox in 2015 in the paper "U-Net: Convolutional Networks for Biomedical Image Segmentation" [5]. Its architecture was created by modifying and extending the fully convolutional network architecture to work with fewer training images and to produce more precise segmentations. The U-Net consists of a contracting path and an expansive path, which are more or less symetric and give it the U-shaped architecture.
The contracting path is a typical convolutional network with repeated application of 3x3 convolutions, each followed by a rectified linear unit (ReLU) and a 2x2 max pooling operation with stride 2 for downsampling. At each downsampling step the number of feature channels is doubled. The expansive path combines an upsampling of the feature map followed by a 2x2 convolution that halves the number of feature channels, a concatenation with the correspondingly cropped feature map from the contracting path and two 3x3 convolutions, each followed by a ReLU. Finally, a 1x1 convolution is used to map each 64-component feature vector to the desired number of classes. In total the architecture has 23 convolutional layers. See figure 2.

2.3 Previous achievements

The information recieved before starting the project were different papers about object counting and object detection [4], [3]; about the network that was planned to be used, the U-Net [5], and finally, two articles related to the same topic and the same environment [1], [2] which helped to understand the

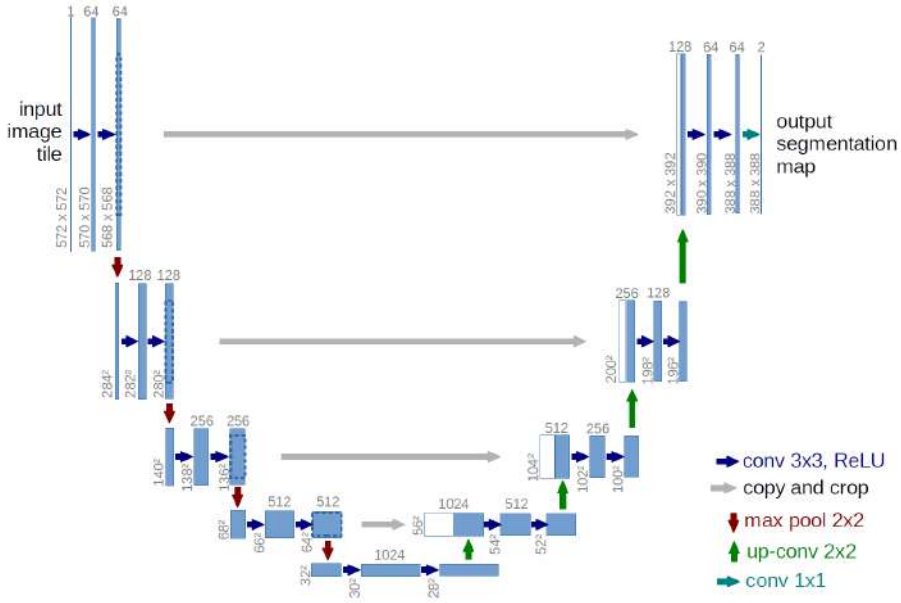


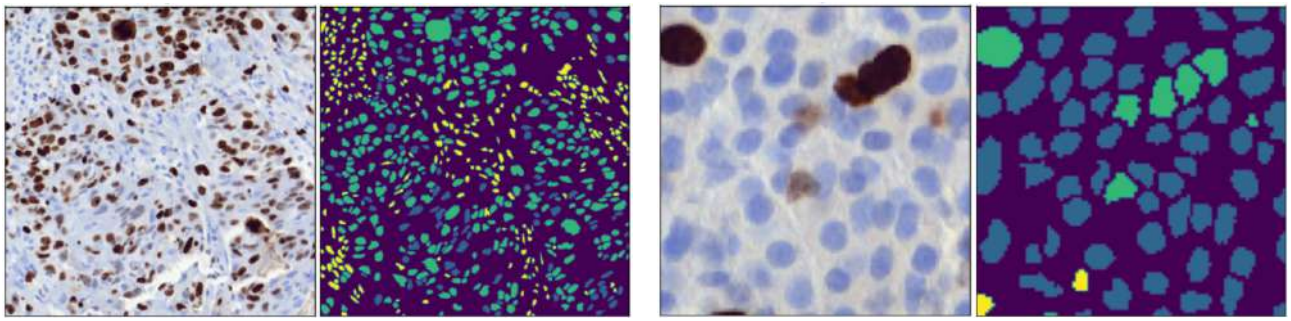
Figure 2: Architecture of the U-Net model [5].

objectives and context of the DigiPATICS project.
Besides the theoric information, two algorithms, already developed by DigiPATICS, were provided:

- **Cell Segmentation Algorithm**: Given the original images and their ground truth, this algorithm computes a semantic segmentation model using a U-Net architecture that learns to predict a value for each pixel in order to later define the shape and position of each cell nucleus. The ground truth images are binary or multiclass masks where each pixel is assigned to a given class. See figure 3a. Note that in the original images, close cells do not have a clear contour that splits them. Since the algorithm only treats the information of each original pixel, it cannot take into account the shape of the cells to have more context and know whether the pixel is located inside a cell or between two cells. Therefore, when in the original image two cells are touching, the algorithm is not capable of distinguishing them and thus it merges them in a single connected component. In 3b there is a large dark cell but the ground truth image shows that this component should be split into 3 different cells. This algorithm would return a big connected component merging these three cells.
  Initially, the metrics used to optimize the model were F-score, precision and recall, the loss function was the Dice loss. The model used the Adam optimizer with a learning rate of 0.00068, 4 images per batch in the training process and 100 epochs. No results can be shown since the database used was a different one.

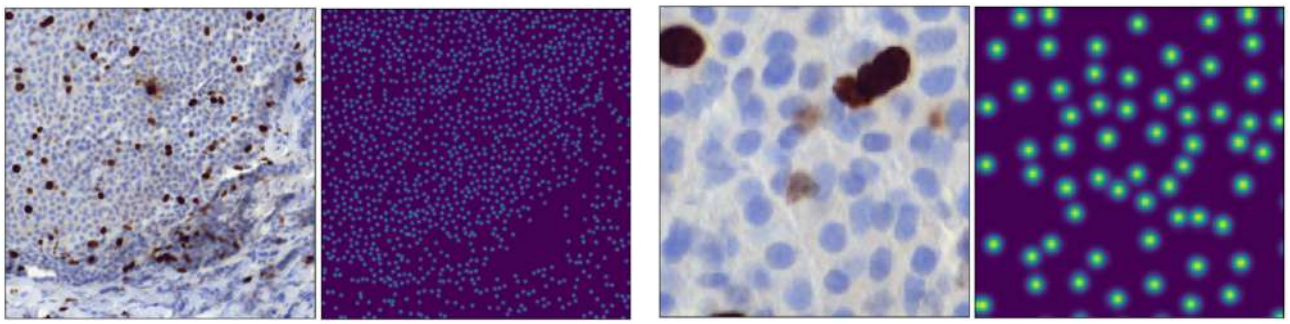
- **Cell Count Algorithm**: In [3] and [4] two different algorithms were created: one was based on the estimation of an algorithm that infers fully convolutional networks able to count and detect microscopy cells and the other was created to estimate the number of cells in a microscopic image avoiding the hard task of learning to detect and localize those objects. Both shared the same idea, to estimate an image density whose integral over any image region had the count of objects within that region. Those density maps returned gaussian-like functions for each detected cell whose center would be easily detected by selecting the maximum values.

These algorithms gave the idea to create a similar approach to solve the problem of the cell segmentation algorithm and thus achieve a more accurate prediction of the real number of cell nuclei in an image. The ground truth images from the cell segmentation algorithm were modified in order to obtain the center of mass of each individual cell and a gaussian filter was applied to obtain similar images as the density maps. Therefore, the position of each cell nucleous center was represented by the maximum of a gaussian function. These new images became the ground truth images to train the cell count model. See figure 4b.
Initially, no metrics could be used for this model since there was no method to compare the prediction with the ground truth images. The loss function used was the MSE loss and the model used the Adam optimizer with a learning rate of 0.00068, 4 images per batch in the training process and 200 epochs.



(a) (b)

Figure 3: In (a), original image and its multiclass ground truth. In (b), a region of an original image where cells could be merged into a unique connected component and its multiclass ground truth mask.



(a) (b)

Figure 4: In (a), original image and its ground truth using sigma=5. In (b), a region of the same image to see how the centers do not overlap even if several cells could create a connected component.

To make sure that the model does not overfit, a data augmentation method is applied to the training data shifting, rotating, scaling and/or flipping each image before its use.

Both algorithms use an important public library to train the desired model, which is "segmentation_models_pytorch" [13]. Using this package many types of models can be trained just by choosing the network, changing the parameters, the loss function, the metrics to be obtained, among other options.

2.4 Databases

There are two different databases, one for each breast cancer protein (ER and Ki67). A biopsy is performed on each breast cancer patient from the Vall d'Hebron Hospital in Barcelona. After the required staining and digitalization processes, IHC-stained samples are obtained called "whole slides" with an original size of, approximately, 200,000x90,000 pixels. Afterwards, analyzing the interesting zones of these images, 1,500x1,500 pixel images are extracted. However, the images used for the project are a resized version to 512x512 of these images because of its big size.

For each database, there exists a set of ground truth images, the ones for the cell segmentation algorithm. The model needs a ground truth with the shape, position and class of each cell nucleus (each cell has a unique identifier related to its class). The creation of these ground truth images was developed by the DigiPATICS group using the original images and a set of techniques different for each database. Afterwards, using the previous ground truth images, the center of each individual nucleus was obtained to create the set of images for the cell count algorithm.
As different databases classify the cells in different numbers of classes, the creation of the algorithm was developed firstly with a single database, Ki67, and finally generalizing to the other one, ER.

Ki67 database

The Ki67 dataset consists of 66 images from 20 different patients. In the original images the cells inside tumoral zones are classified into a class: positive, negative or stroma. The principal characteristic of each class in this zones is its color and shape: the positive malignant nuclei (PMN) are brown, the negative malignant nuclei (NMN) are blue and round-shaped and, finally, the stroma nuclei are also blue and thus easy to be confused with the negative class. However, the stroma does not have a regular shape and is a bit smaller than the other classes. The score needed for the pathologists is:

$$Ki67\ score = \frac{\#PMN}{\#PMN + \#NMN}$$

The ground truth images for the cell segmentation algorithm were created through classical mathematical morphology algorithms applied to the original images. The resulting images were given to expert pathologists, who modified the detected errors and accepted the resulting images. However, there can be a lot of variability between pathologists labeling the same class or even, human errors. Afterwards, using these images the center of mass of each connected component is computed, which will represent the ground truth for the cell count algorithm. This set of 66 images, called version 1, had problems and thus, a subset of the database was created.

Some images seemed to have suspicious annotations, that could be caused by the fact that pathologists did not take into account whether the image belonged to a tumoral zone or that the stroma and negative class are difficult to distinguish between them. Those images were removed from the dataset. As a result, there remain 42 images. This dataset called version 2 is the one used during the optimization of the networks.

During the course of the project the three classes are going to be represented by three colors: negative class in blue, positive class in brown and stroma in yellow. See figure 5.

Figure 5: Colors used in the algorithm to represent the classes of the Ki67 database.

ER database

The ER dataset has 105 images from 12 different patients. In this case the cell nuclei in tumoral zones can be associated to one of the 5 following classes: negative malignant nuclei which are blue, positive malignant nuclei type 3 which are represented in dark brown, positive malignant nuclei type 2 represented in mid brown, positive malignant nuclei type 1 represented in light brown and stroma.
The 2010 guideline [12] recommended that invasive breast cancers can be considered ER-positive if at least 1% of cancer nuclei stain are positive and that patients with such cancers be considered for endocrine therapy, while such therapy should be withheld from patients with cancers with less than 1% staining. It was also noted that it is reasonable for oncologists to discuss the pros and cons of endocrine therapy with patients whose cancers contain low levels of ER by immunohistochemistry (cases with weak stain intensity or less than 10% of cells staining) and to make a decision based on the totality of information about the individual case [10]. That is why an interesting ER-score would take into account the intensity of the stain (type 3, type 2 or type 1) and the percentage of stained nuclei over the total number of tumoral nuclei (positive and negative, not stroma) in the image.

$$0 \leqslant \%PMN_{type1} * 1 + \%PMN_{type2} * 2 + \%PMN_{type3} * 3 \leqslant 300$$

The ground truth images for the algorithms were also created by the DigiPATICS group defining a morphological algorithm, as in the previous database but this time no verification by the pathologists was performed.

During the project the classes of the ER database are expressed in 4 different colors: negative in blue, positive type 3 in brown, positive type 2 in red, positive type 1 in pink and stroma in yellow. See figure 6.

Figure 6: Colors used in the algorithm to represent the classes of the ER database.

3. Developed algorithm and optimizations

3.1 Final algorithm structure

The final algorithm consists of a block A where a semantic segmentation algorithm is applied using a U-Net network that classifies each pixel into one of the three classes of the Ki67 dataset given the original color images of size 512x512x3 and their multiclass ground truth masks. The resulting pixel class determines the shape, position and class of each predicted cell. The metrics computed pixel by pixel for each class are F-Score, precision and recall. However, as mentioned before, this algorithm does not behave accurately when several cells are close because it merges them as a single and bigger cell. See block A from figure 7 in the output image at the top right there are several cells forming a big connected component.

Hence, the necessity of a block B, able to run in parallel, where the same original images are used but the training is done with a different set of ground truth images. This time, it is necessary to find the center of each individual cell in order to distinguish them when they belong to a single connected component. Therefore, the ground truth images are dot annotations, where each dot corresponds to one cell nucleus. For training, a gaussian filter is applied to the dot annotations. The task is to regress the density surface created by the superposition of these gaussians given the original image. A U-Net model is trained using the mean square error as the loss function for regression to achieve a heat map predicting the density map. See block B of figure 7. To achieve the predicted centers and compute the F-Score, precision and recall metrics, the maximum of each gaussian function is needed. To do so, the local maxima of each gaussian function that exceeds a certain contrast are extracted. The metrics are computed through an algorithm created by DigiPATICS able to match a predicted pixel with the closest ground truth pixel and mark it as a true positive even if the location was not exactly the same.

Finally, a last block, C, merges the results of both blocks. A watershed is applied on each predicted connected component from block A to split them into as many cells as centers block B predicted. See in figure 7 block C how the big connected component has several centers and how after the watershed the component has been split. The semantic segmentation (A) assigns a class to each individual pixel and, after the cell shape has been defined, there is the need to assign a single class to each individual cell, which is computed through majority voting of the pixel class. Then, F-score, precision and recall are computed using the position of the predicted centers and the class of the cell. In the majority of the cases, if block B predicted a center, block A predicted a cell around it. However, in some cases there were cells without center or vice versa (centers without cell) and thus, the question on how to handle these types of situations was raised. After some experimental evidence, the best way to follow was to only take the intersection of both predictions, i.e., the cells that contain at least one center.
In this case, the ground truth images are single pixels at the center of each nucleus with the value of its class. The final result is a label image with an identifier for each pixel belonging to the cell nucleus and a document with the corresponding class of each cell.

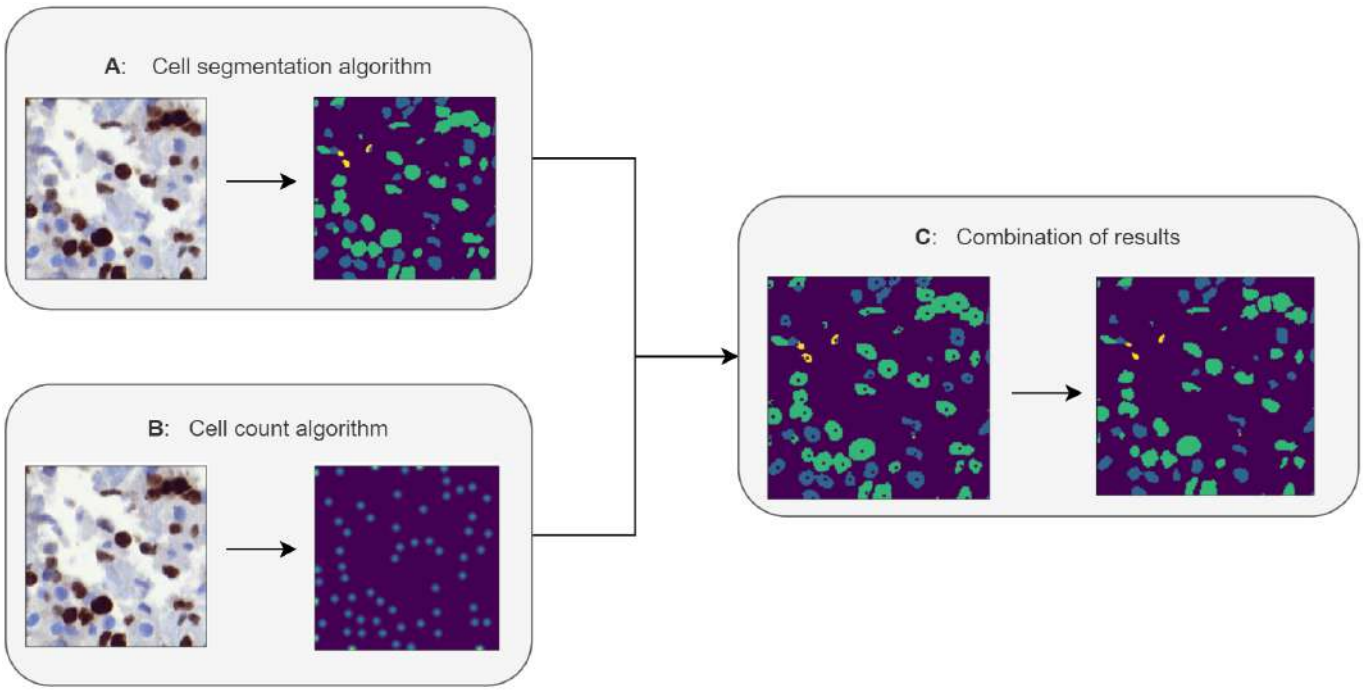


Figure 7: Structure of the final algorithm and example of the input and output of each block. In block A each pixel is classified, in B gaussian functions are predicted at the center of each cell and a postprocessing is applied to retrieve each maximum value (here a gaussian filter is applied to see the detected centers). In block C, the combination of results is made to split big components in different cells.

3.2 Networks optimization

The project started optimizing each algorithm with the Ki67 database. When the final results for this database were achieved, a process of generalization into the ER database was performed. Before starting developing and optimizing each algorithm, initial decisions had to be made.

As in every machine learning model, the database needs to be split into training, validation and test subsets. Since the databases do not contain a very large number of images and the models need a lot of them to learn and avoid overfitting, the decision was to split all the images into training and validation. The consequent decision was to choose the kind of split to do so: (1) a random split where the 20% of images was saved for the validation set and the rest to the training set or (2) saving one image per patient for the validation set and the rest to train.

The model hyperparameters to optimize were also another decision to make. Different parameters of the model were proposed to be optimized in order to achieve the convergence of the loss, precision, recall and F-score: the optimizer, its learning rate and the batch size of the training step.

A first optimization of these hyperparameters and the type of split was done using the cell segmentation algorithm with binary ground truth images to have an initial idea of the values to study during the optimization of the different algorithms. The results in the appendix A.1 show that the kind of split does not have much influence in the results and thus, the random split was chosen.

In both algorithms, a data augmentation step is executed: every time an image is taken it is shifted with a shift factor range of (-0.2, 0.2), scaled with scale factor range of (-0.1, 0.1) and rotated with a rotation range of (-10, 10) and with probability 0.5, flipped. Initially, some experiments were carried out to assess if the application of this process was relevant to the algorithm and it resulted in an aid for the model to avoid overfitting. That is why the application of the data augmentation will be executed during the whole project.

A BLOCK: Cell Segmentation Algorithm

The implementation of this algorithm aims to classify the individual pixels of the image into one of the three classes of the Ki67 database version 2. From this classification the cell nuclei shape, position and class are going to be defined.

The first aspect to check in a classification algorithm is whether the classes are balanced in order to better understand the results. In the Ki67 dataset the classes are unbalanced leading the stroma class to be underrepresented. Table 1 shows that, taking into account the number of cells, the negative tumoral class is the most represented one and stroma and positive class are almost equally represented. However, the positive cell nuclei are always bigger than de stroma cell nuclei and, since the model learns pixel by pixel, the values to look at should be the number of pixels for each class. In that way, the least represented class is the stroma with a 13%, 25% less than the positive class. There exists an unbalance, not very biassed, but enough to be taken into account during the optimization and the results interpretation.

To achieve a final value that takes into account the results for all the classes in a proportional way, a weight is applied to the value of F-score, precision and recall related to the number of pixels of the class in the image. Therefore, the final pixel-by-pixel metrics for the whole algorithm are going to be represented by the weighted F-score, weighted precision, weighted recall and the average F-score of each one of the classes.

	Negative class	**Positive class**	**Stroma class**
Mean nº of cells (%)	49.53	26.55	23.92
Mean nº of pixels (%)	48.02	38.71	13.27

Table 1: Number of cells and number of pixels for each class in the version 2 Ki67 dataset.

Hyperparameters optimization

Since this algorithm does not require any other experiments than the optimization of the model hyperparameters a new loss function was proposed. In the original implementation of the segmentation algorithm, the loss function used was the Dice loss, a function based on the Dice coefficient which is a widely used metric in the computer vision community to calculate the similarity between two binary images. However, the Focal loss is able to down-weight the contribution of correctly classified examples and enables the model to focus more on learning hard examples. Moreover, it works well for highly imbalanced class scenarios [11]. The Ki67 database is not a case of highly imbalance but it could be useful to deal with it. Therefore, both functions were evaluated.

The results using the first version of the Ki67 dataset showed that the convergence was slower when using the Focal loss, and thus, the learning rate had to be reduced and the number of epochs increased

to achieve the convergence of the models. Final configurations tested the influence of the number of images for each batch for the validation dataset (batch size) and the optimizer used. In table 2, the multiple configurations of these hyperparameters and the weighted metrics and average scores per class can be seen.

mod	loss	opt	lr	ep	bs	w_fsc	w_prec	w_rec	F-score per class
1	dice	adam	0.005	200	2	0.7829	0.7776	0.7987	1: 0.7298 2: 0.8362 3: 0.311
2	dice	adam	0.0005	200	2	0.777	0.7937	0.7713	1: 0.717 2: 0.8316 3: 0.34
3	dice	adam	0.00005	200	2	0.7573	0.7809	0.7462	1: 0.6982 2: 0.8206 3: 0.2783
4	focal	adam	0.005	100	2	0.5868	0.8608	0.4674	1: 0.4925 2: 0.7435 3: 0.0
5	focal	adam	0.0005	400	2	0.7131	0.82	0.642	1: 0.6709 2: 0.8151 3: 0.0
6	focal	adam	0.00005	400	2	0.6291	0.6355	0.6475	1: 0.5529 2: 0.7124 3: 0.0008
7	dice	adam	0.005	200	2	0.7735	0.7551	0.8101	1: 0.7179 2: 0.8345 3: 0.2609
8	dice	adam	0.005	200	4	0.7505	0.7527	0.7536	1: 0.7208 2: 0.8382 3: 0.0
9	dice	adam	0.005	200	6	0.7789	0.782	0.7934	1: 0.732 2: 0.8327 3: 0.2563
10	dice	adam	0.005	200	8	0.7444	0.7416	0.7554	1: 0.6984 2: 0.8319 3: 0.0
11	dice	sgd	0.005	200	2	0.6913	0.6788	0.7289	1: 0.6648 2: 0.7768 3: 0.0

Table 2: Optimization of the hyperparameters to achieve the convergence and the highest metrics. In the table, **mod** is the number of the model, **loss** is the loss function, **opt** is the optimizer used, **lr** is the learning rate, **ep** the number of epochs trained, **bs** the number of images for each batch in the training dataset, **w_fsc** is the weighted fscore, **w_prec** is the weighted precision, **w_rec** is the weighted recall and **F-score per class** is the F-score averaged for each class (1: negative, 2: positive, 3: stroma).

Even when reducing the learning rate to 0.00005 and reaching up to 400 epochs (model number 6), the focal loss seemed to hinder the model learning. In figure 8 it can be seen that with 200 epochs the model tends to be omitting a lot of objects since the recall is very far from the precision.

From model 7 to 11, the batch size and optimizer effects were evaluated. Model 7 has exactly the same values as model number 1 but the numerical values of the metrics are not the same and the reason is the random initialization of the weights of the network at the beginning of the training process. This slight change is comparable to the one produced by the change of batch size, and thus, it is concluded that the batch size is not as significant with this kind of database.
The SGD optimizer in many cases has been tested and always seemed a bad optimizer for the algorithms. It was used but fastly discarded because of its slow training and strange behaviour in the learning curve.

Therefore, the best configuration could be the one of the model number 1, with Dice loss, Adam optimizer with a learning rate of 0.005 until 200 epochs, which gave the best weighted F-score and F-scores per class (with the exception of the stroma class). However, when looking at the curves created by the metrics it seems that there is a lot of variability in the metrics of the validation dataset between two consecutive epochs, see figure 9. That is why the next best configuration of the table optimization was studied and the figure created by the metrics was now more stable, which could lead to a better generalization to other images. The model number 2 has the learning rate decreased from 0.005 to 0.0005 and it seems to be the best, see figure 10. The slight change in the metrics is not significant since a first random initialization can produce similar changes. Therefore, this is considered the best model of this optimization with a weighted F-score of 0.777.

In conclusion the best configuration of hyperparameters for this algorithm is:

Dice loss, optimizer = Adam, learning rate = 0.0005, 200 epochs, batch size = 2

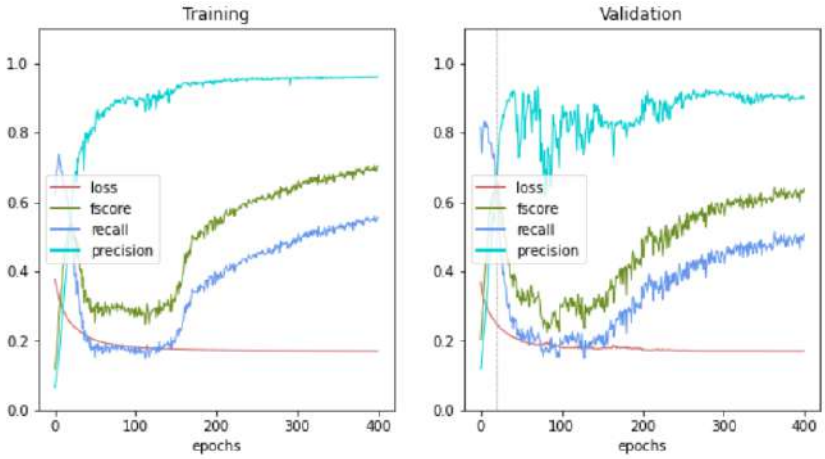


Figure 8: Learning curves of the model number 6: Focal loss, Adam optimizer with learning rate 0.00005 and batch size 2. In the left, the training metrics, in the right, the validation metrics at each epoch. The best result is in epoch 19.

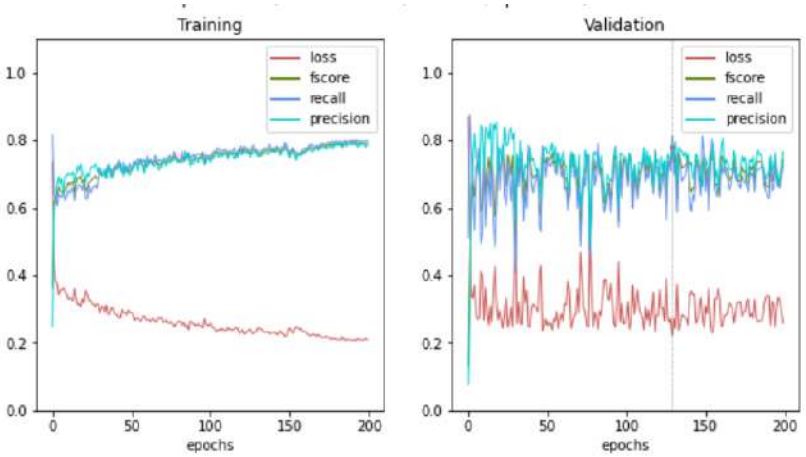


Figure 9: Learning curves of the model number 1: Dice loss, Adam optimizer with learning rate 0.005 and batch size 2. In the left, the training metrics, in the right, the validation metrics at each epoch. The best result is in epoch 129.

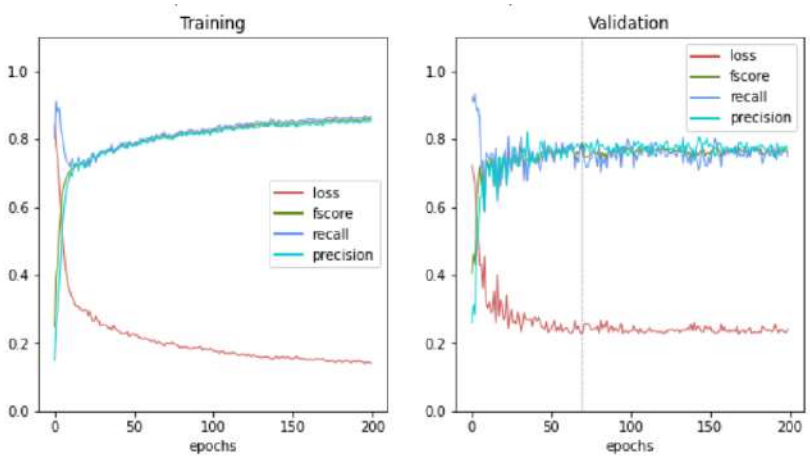


Figure 10: Learning curves of the model number 2: Dice loss, Adam optimizer with learning rate 0.0005 and batch size 2. In the left, the training metrics, in the right, the validation metrics at each epoch. The best result is in epoch 69.

B BLOCK: Cell Count Algorithm

This algorithm aims to detect the center of each cell nucleus in Ki67 database version 2 in order to have a better approximation of the real number of cells in the image. As said previously, the dot annotations representing the center of the cell nuclei are transformed to gaussian functions for the training process. Therefore, the algorithm executes a regression model that tries to approximate the created density map. However, the goal is to be able to extract the maximum of each predicted gaussian function and use this detection as a cell nucleus center to compute the metrics with the original ground truth images with single pixels.
This gaussian filter creates wide or narrow gaussians depending on the sigma parameter. If a larger number is used, the gaussians could overlap and add false information to the image, otherwise, the model could not have enough information. That is why this sigma parameter will be later optimized.

Before starting the optimization of hyperparameters, two main problems emerged: how to transform the predicted images to compare them with the ground truth and how to compute metrics with these ground truth images.

Cell based metrics

The metrics of interest are F-score, precision and recall. However, as the algorithm implements a regression model some changes had to be made. The pixel based metrics are not adequate in this case since they compare if each ground truth pixel is equal to its corresponding predicted pixel. That is why a new calculus of the metrics had to be implemented, which was carried out by the DigiPATICS group. The idea was to create cell based metrics that could take into account if the cell was correctly predicted. There are two steps: matching predicted pixels with ground truth pixels and checking the classes. The method steps are: given each center of the prediction the closest center in the ground truth is searched and vice

versa (for each center of the ground truth the closest center in the prediction is searched), discards the correspondences with higher distance than a certain threshold, discards the correspondences that are not 1 to 1 and calculates the cell based metrics comparing the pixel class. Using the cell based metric it can be ensured that if a center is predicted a few pixels off with respect to the ground truth it will be taken as if it had been predicted correctly.

Postprocessing method optimization

As the prediction returned by the model consists of gaussian-like shapes, a postprocessing was needed to extract the center of these gaussians and to use them as an estimate of the cell centers. This postprocessing could be done in two ways: binarizing the predicted image so that the highest values represent the center of the cell nuclei or focusing on the local maxima with a certain contrast. Both methods returned a set of connected components from which the center of mass was extracted and used as the center of the gaussian function.

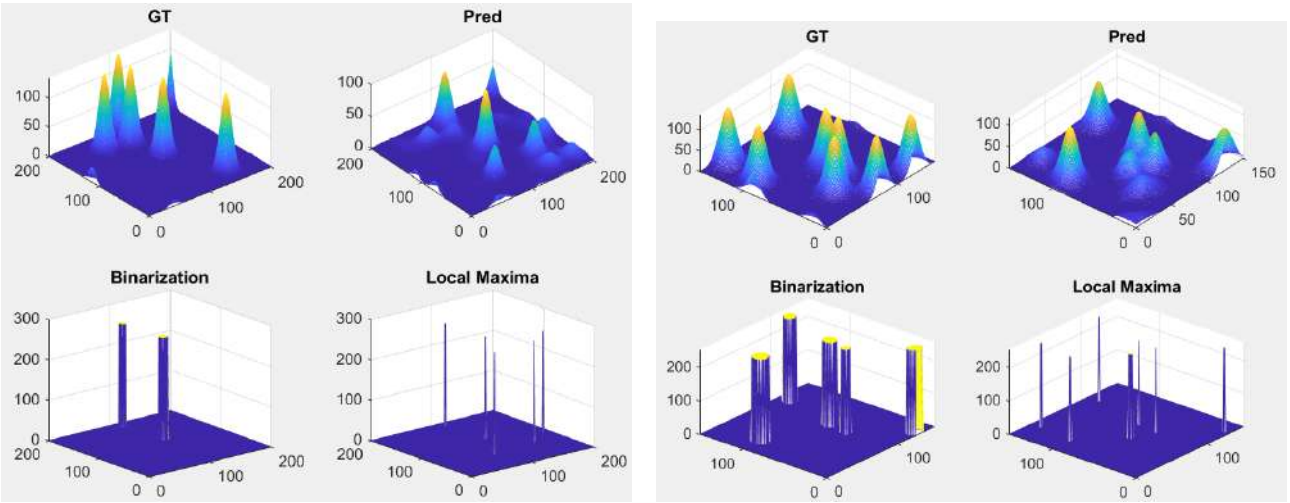
The first optimization experiment was to choose one of the two methods to extract the centers of each predicted gaussian. The process consisted in training a model with a fixed value for the sigma parameter, set provisionally to 7, which gave enough information to train and the gaussians did not overlap. The prediction of the validation images was postprocessed with each method and a value for the F-score was given. The best method could be later optimized with the sigma parameter to retrieve the best combination.

- **Binarization method**: consisted in optimizing a threshold value to binarize the predicted image so that the highest values represented the cell nuclei and its center of mass, the center of the cell nuclei. The predicted images had a grey level range from 0 to 255 and thus the threshold value was optimized between 50 and 200. The best score was obtained with the threshold set to 63, i.e, only the pixels with a lower value than 63 were set to 0. Afterwards, the center of each connected component was computed and used as the centers to compare with the ground truth. The optimal F-score obtained using the cell based metrics was 0.7213, computed as the average F-score between all the validation images.
  Note that this method could discard lower values that do represent cell nuclei and keep high values that do not represent them.

- **Local Maxima method**: the idea was to detect all the local maxima in the image and retrieve them as centers of the gaussian functions. However, as the prediction has noise, many local maxima could be present, and thus, a certain threshold is needed. The local maxima are connected sets of pixels with equal gray level strictly greater than the gray level of all pixels in the neighborhood. Then, as a threshold value, a certain contrast h could be established to avoid selecting local maxima whose path to another equal or higher local maximum does not decrease more than h. Therefore, the h value had to be optimized.
  This method was able to distinguish two local maxima that overlap but have a contrast greater than h. The contrast value was optimized between 0.05, 0.15 and 0.25. The optimal F-score was 0.7477 with h=0.15.

In figure 11 both methods are illustrated: the binarization method cuts the gaussian functions from the prediction and a few gaussians are retrieved whereas the local maxima method is able to select the gaussians that stand out the most even if they do not reach high values. Moreover, the prediction could

not be precise and two high gaussian functions could be predicted, which the local maxima could identify as a single gaussian function and the binarization method could not. The conclusion of this experiment was to use the local maxima method, that besides reaching a higher average F-score, it preserves important information that the other method could discard and it is able to distinguish ambiguous cases.



(a) 200x200 square section of an image. (b) 150x150 square section of an image.

Figure 11: The ground truth with a gaussian filter of sigma=7 applied, the prediction of the cell count algorithm, the location of the gaussian maxima for the binarization method and the location of the gaussian maxima for the local maxima method.

Sigma and contrast optimization

An important optimization was the sigma parameter which widens the gaussian functions located at the center of the cell nuclei. It was done in conjunction with the contrast value of the local maxima method. After some tests the values studied for the sigma parameter were set from 3 to 7 since more than 7 lead to overlapping gaussian functions and alteration of the information. With less than 3 the model could not predict as accurately as wanted and a lot of noise was added to the predictions. The contrast values studied were 0.05, 0.15 and 0.25 since more than that caused two sufficiently contrasted local maxima to be considered as one.
After some initial tests, given the mild difference between sigma-contrast configurations, the optimization of both parameters was postponed for later steps, where more definitive conclusions could be obtained. The results are shown in 3.3 and the best configuration is with sigma 6 and contrast 0.15.

Final hyperparameter optimization

In order to have an optimal configuration of hyperparameters, a final selection of the best optimizer, learning rate and batch size was needed. The loss function was always set to MSE loss since it is the most common loss function used in regression algorithms.

In table 3 different configurations testing every hyperparameter can be observed. The SGD optimizer is always difficult to make the model converge and the number for the batch size does not have much influence in the metrics. If the model does converge with 100 epochs, then with 200 epochs could achieve slightly better results but doubling the time. Then, the learning behaviour of the metrics from the model with the best F-score are observed to know if there is the need to train during more epochs. In figure 12 the horizontal lines formed by the metrics and the loss can be seen and it is concluded that there is no need to check the model with more epochs since the model has already converged. Therefore the best hyperparameter configuration for the cell count algorithm is the following with a maximum F-score of 0.8817:

MSE loss, optimizer = Adam, learning rate = 0.005, 100 epochs, batch size = 2

mod	opt	lr	ep	bs	fscore	precision	recall
20	adam	0.05	100	2	0.8796	0.8861	0.8775
21	adam	0.005	100	2	0.8817	0.8915	0.8760
22	adam	0.0005	200	2	0.8755	0.8826	0.8729
23	adam	0.00005	200	2	0.8727	0.8816	0.8690
24	adam	0.0005	100	2	0.8744	0.8828	0.8724
25	adam	0.0005	100	6	0.8640	0.8608	0.8742
26	sgd	0.0005	100	2	0.2770	0.1689	0.8792

Table 3: Final hyperparameter optimization to achieve convergence and high metrics. In the table, **mod** is the number of the model, **opt** the optimizer, **lr** its learning rate, **ep** the number of epochs and **bs** the batch size for the training dataset. The metrics obtained are the average F-score, average precision and average recall for all the validation images.

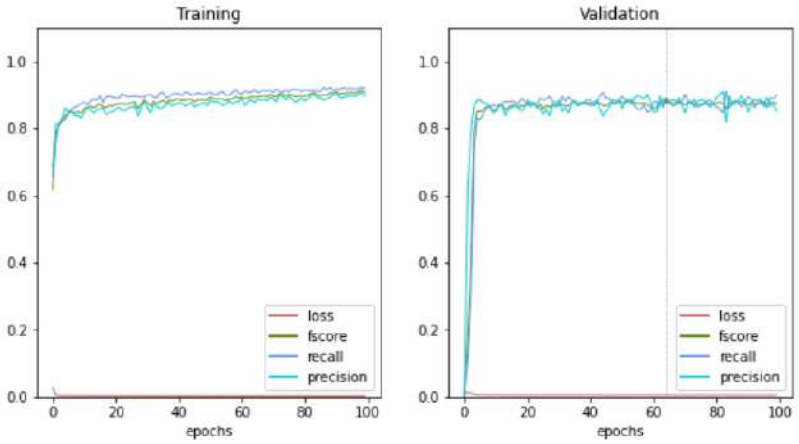


Figure 12: Learning curves of the model number 21: MSE loss, Adam optimizer, learning rate 0.005 and batch size 2. In the left, the training metrics, in the right, the validation metrics at each epoch. The best result is in epoch 64.

3.3 Combination of prediction results

The last block, C, is necessary to combine the results of both algorithms into a single, improved prediction. From block A, there is the prediction of the cell nuclei given by the model number 2 whose main problem is the merging of neighboring cells into a single connected component: the classification step assigns the most probable class to each pixel but, to deliver the results homogeneously, each individual cell should contain a single class. With this algorithm, individual cells are represented by the connected components which, in many cases, do not represent a single cell nucleus but a set of them. The results from block B will help to divide them into as many centers as the model predicts.

The first questions that came up were: which algorithm is more reliable? Since there are predicted cells that do not have a predicted center or centers that do not have a predicted cell, how many centers does each connected component have? Will the split process increase the F-score value significantly? Some initial experiments were made to understand the algorithm results, the proper combination and try to answer some of the previous questions.
First of all, when overlapping the gaussians prediction and the cell segmentation prediction there could be seen large connected components with many centers, connected components without centers inside and centers without a connected component, see figure 13.

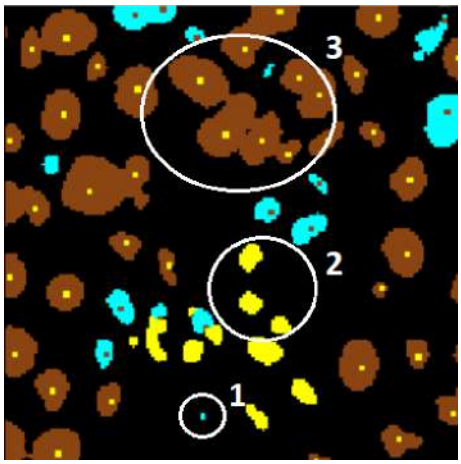


Figure 13: Overlap of the cell segmentation prediction and the dilation of the cell count prediction. Number 1 is a center predicted from the cell count algorithm without a connected component from the segmentation algorithm. Number 2 is the example of predicted components from the cell segmentation algorithm without a center. Number 3 is a large connected component with many centers.

The number of connected components coinciding or not with a center was an interesting aspect to look at since it could give information about the kind of improvement the watershed could suppose. In the validation set (with 9 images) there were in average:

- 102.44 connected components without a center associated,
- 44.67 centers without a connected component associated,
- 551.78 connected components with a single center,

- 65.33 connected components with 2 associated centers,
- 12 connected components with 3 associated centers,
- 4.25 connected components with 4 centers,
- 2.2 connected components with 5 centers and
- 1 connected component with 6, 7 and 8 centers.

There are a lot more connected components without center that centers without connected components, meaning that the cell segmentation either is capable of detecting cells that the cell count is not or the cells it detects do not have to be considered cells. Note that the 70% of connected components coincide with a single center, i.e., both algorithms coincide with the prediction. If no splitting is made there would be 86.78 connected components with more than 1 center inside, but when splitting is applied it would retrieve 219.66 individual cells. Thus, the splitting process would significantly improve the results.

At this point, 4 logical approaches emerged:

- Approach 1: take into account only the predicted connected components that have one or more predicted centers inside. Then, the centers that the count algorithm predicted but the segmentation did not mark as a cell would be discarded and viceversa. Therefore, the intersection of predictions was taken.
- Approach 2: take into account the intersection of predictions but add the predicted centers that the cell count algorithm found and the cell segmentation did not mark. Therefore, the cell count algorithm would be considered the most reliable.
- Approach 3: take into account the intersection of predictions but now, adding the connected components that the cell segmentation algorithm predicted and discarding the centers only predicted by the second algorithm. The center of mass of each additional cell is computed to represent them. Therefore, the cell segmentation algorithm would be seen as the most reliable.
- Approach 4: take into account all the predictions from both algorithms. All the predicted centers and all the predicted connected components.

Optimization of the approaches

The decision of which approach to follow was studied during the optimization of the contrast and sigma from block B since these hyperparameters have an important influence on the number of centers predicted by the algorithm. As said, sigma was optimized between 3 and 7 and contrast was optimized between 0.05, 0.15 and 0.25. It results in 15 different models.
The four approaches were tested using the centers predicted by each one of the models and the detection of the best cell segmentation model (model 2), regardless of the class predicted. The metrics computed were the cell based F-score, precision and recall.

The results can be seen in appendix A.2. The first approach reaches a maximum F-score of 0.879 tied with the second approach. However, when adding the center of mass of each additional connected component in approach 3 the metrics start to fall slightly reaching 0.871 and approach 4 is clearly not the optimal solution with any of the sigma-contrast combinations since the highest value is 0.866. Between the first and second approaches there are such slight changes that one approach could not be considered significantly better than the other. These tables give information about the best value for sigma and contrast. Observing the first two approaches, the changes between configurations are minimal. Despite this, it seems that contrast=0.15 leads to the best F-scores in both cases. Even if the sigma parameter from 5 to 7 always gives an F-score of 0.879, the value 6 is chosen as the best since the precision and recall are the ones most balanced.

In conclusion, the best sigma-contrast configuration is 6 and 0.15 and the optimal approach needs to be studied more deeply.

The semantic segmentation network output gives for each pixel the probability to belong to each class. When the algorithm does not assign a class to a pixel it means that none of the three probabilities exceeded 0.5. Once this is explained, the fact that approach 2 has some additional centers located in a zone not detected by the segmentation algorithm also means that the probabilities of belonging to one of the three classes were not high enough. Hence, a threshold could be useful to decide whether to accept or discard those additional centers: take the three probabilities of the pixel, select the maximum one and accept the center if the value exceeds the threshold.
This probability threshold was optimized from 0 to 0.5: with 0.5 or greater, any center could not be added to the prediction and with 0, every center could be added. Table 4 shows the different values for the probability threshold and the metrics obtained. When adding all of them it represented a 6% of the centers but the weighted F-score worsened with respect to the rest. Besides the weighted F-score, the reason why the value taken to continue with the project was 0.5 or greater was that the numerical changes were not significant taking into account the random initialization of both algorithms and thus, the simplest option was chosen. Therefore, the chosen approach is the first one, when both algorithms coincide in detecting a cell and a center.

In conclusion, the best approach depends on the database and the algorithms results, but in this project the intersection of prediction results is chosen.

prob thresh	w_fscore	fscore per class	added centers (%)
0.5-1	0.7374	1: 0.7520, 2: 0.8395, 3: 0.3436	0
0.1	0.737	1: 0.7512, 2: 0.8386, 3: 0.3466	0.8068
0.01	0.7353	1: 0.7494, 2: 0.8367, 3: 0.3459	1.6292
0.001	0.7349	1: 0.7487, 2: 0.8371, 3: 0.3445	2.0904
0.0001	0.7333	1: 0.7474, 2: 0.8359, 3: 0.3404	2.8448
0.00001	0.7314	1: 0.7450, 2: 0.8352, 3: 0.3375	3.4694
0	0.7283	1: 0.7438, 2: 0.8331, 3: 0.3212	6.128

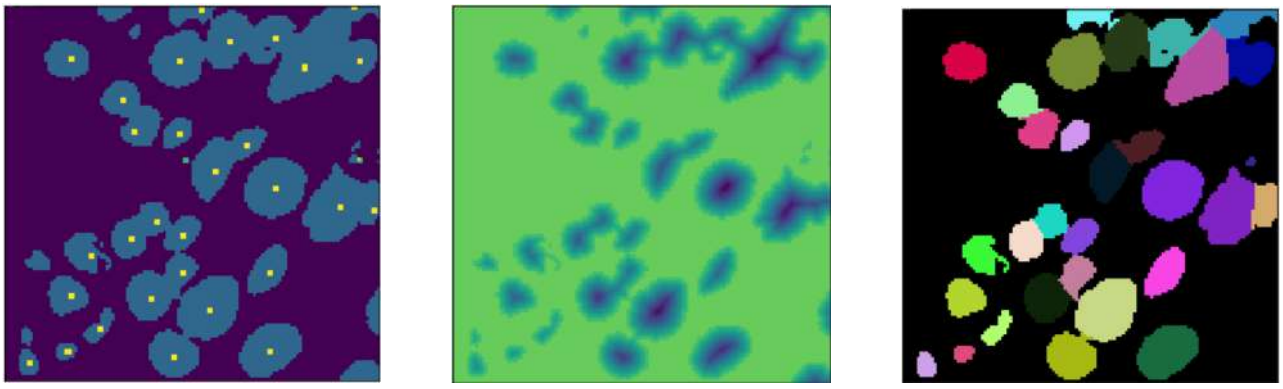
Table 4: Optimization of the probability threshold to accept or discard the additional centers of the second approach. Metrics: the weighted F-score, the average F-score per class (1 negative, 2 positive, 3 stroma) and the percentage of added centers with respect to the real number of centers.

Splitting connected components: watershed

Afterwards, the process of splitting the connected components into individual cells with the information from the centers started. The idea was to use a watershed algorithm, which treats the pixel values as an elevation and floods basins from each marker (defined by the user) until basins from different markers meet. The lines created by these meetings become the watershed lines which differentiate each individual cell from its neighbours.

In this case, the predicted centers would be the markers that identify the starting point of each basin that watershed needs to flood. The basins are defined by the inverse distance transform of the predicted components from the cell segmentation algorithm in binary. The distance transform shows the distance from each foreground pixel to its closest background pixel. Thus, the inverse distance transform has higher values in the borders and lower values in the centers of the cells therefore creating basins at each cell. The predicted components are also useful to define the zones that the flooding cannot exceed, called the mask. This method is able to split big connected components in as many markers as it contains because every marker will start a flood if a basin is defined.

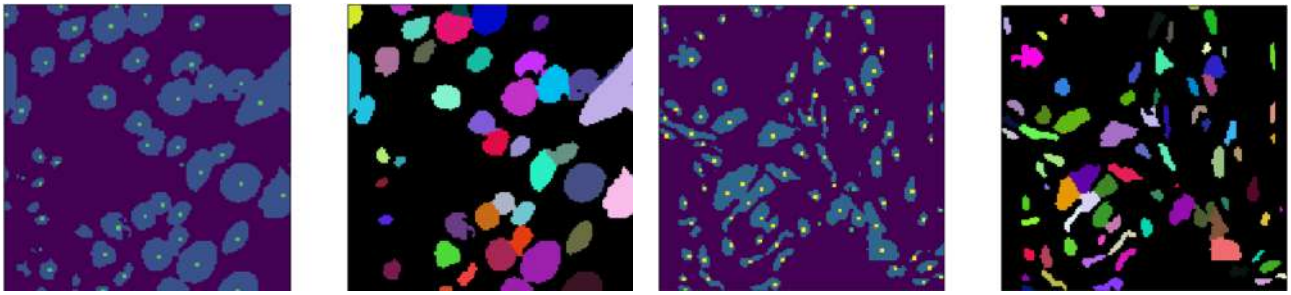
In figure 14a, the markers and the mask can be seen overlapped. Note that there is an instance of centers without component in the center of the image (identified with a green pixel) and some instances of components without center. As the decision was to keep only the intersection of algorithms (approach 1), these objects should not be present, but the watershed itself is able to remove them during the process. Figure 14b presents the inverse distance transform of the mask. If no basin is defined around the marker no flooding will start and thus, no label will be created (the green pixel is erased). In a similar way, if there is a basin but no marker shows the process to start flooding it, no label will be retrieved (the cells without yellow pixels will be erased). Therefore, the approach 1 conditions are accomplished: only the intersection of predictions is taken into account.
In a big connected component with many centers like the top right one, the markers define the initial locations of the flooding and the labeling starts. When the floods meet, a watershed line is traced. The outcome can be seen in 14c in the change of colors between different labels. As a result, there are as many labels as markers (placed inside a cell) and each color represents a single cell.



(a) Markers and mask (b) Distance transform (c) Watershed labels

Figure 14: Watershed process. In (a) the markers showing the cells that have to be retrieved and the mask, the zones that need to be labeled, in (b) the inverse distance transform of the segmentation prediction (in green high values, in blue low values) and in (c) the watershed labels with a different color for each individual cell.

Some examples of other images before and after the watershed can be seen in 15. The watershed lines are not always placed in the same location where the visual human system would do but the differences are slight. These resulting images are useful for the pathologists to have an accurate visualization of the results showing each predicted cell with the most precise shape, position and class. If they could take into account that the split between cells is not as accurate as a human could do, the global perspective of the patient's disease would not change.
Furthermore, the slight errors in watershed lines do not have a numeric influence since the final metrics obtained will be cell based, and thus, will only take into account the position of the center and the predicted class. Therefore, the class needs to be as precise as possible.



(a) Example 1 (b) Example 2

Figure 15: 2 examples of the watershed process. In the left the markers and mask coinciding with the prediction of both algorithms and in the right the labels obtained after watershed.

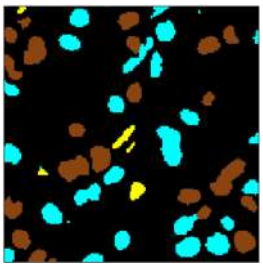
Homogenization of individual cells

Block A predicts the class of the cells given the information of a single pixel, i.e., a cell could have pixels of all three classes. That is why a final homogenization is needed to assign a single class to each cell. This step could not have been done at the end of block A since a big connected component including 3 different-class cells would be classified as the class of the biggest cell and would lead to major errors.
The idea is to match the value of all pixels to one of the three classes. For each one of the output labels from watershed, the median of the values from the multiclass prediction is computed and assigned to each pixel. Thus, each cell will only have one associated class.

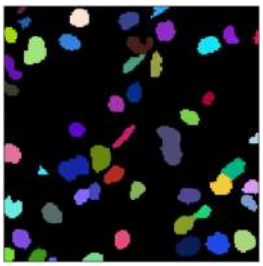
Figure 16 shows the multiclass prediction from the segmentation algorithm which classifies each pixel into one of the three classes. At the bottom right corner there appears an infinity-shaped cell with two classes. If the cell is not split it would take the class after majority voting (since the bigger cell is the blue one, the whole cell would take the blue color, see figure 16c). However, watershed does split this cell and therefore, there can be two individual cells with its corresponding class each, see figure 16d.
In the Y-shaped top center cell the watershed did not hit the right watershed line and thus the homogenization does not work properly.

Despite this, the quantity of brown and blue cells is maintained. In average for all the images in the validation set (9 images) the number of cells retrieved before watershed was 634.67, but after the watershed is applied it reaches 743.45 which represents a 17% increase. To evaluate this increase, the classes should be taken into account using the Ki67 score (the positives with respect to the positives and negatives). The mean absolute error between the Ki67 score for the ground truth images and the prediction before the

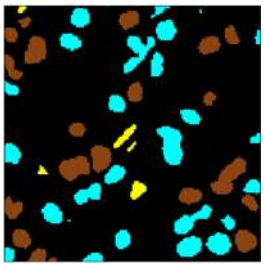
watershed is 4.65% but after the watershed reaches 3.68%. Therefore, the watershed does help classify more cells correctly.



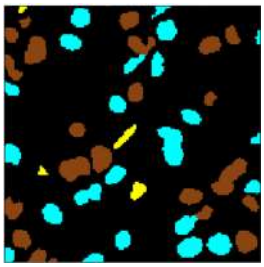
(a) Multiclass prediction.



(b) Labels obtained from the watershed.



(c) Homogenization before watershed.

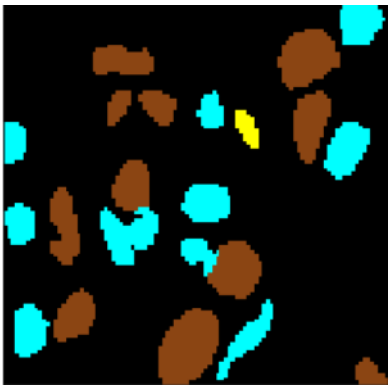


(d) Homogenization after watershed.

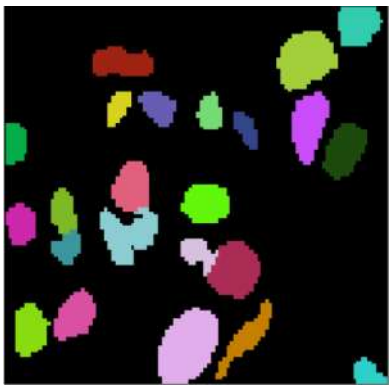
Figure 16: Comparison between the homogenization before or after the watershed is applied.

Final delivery

Finally, the delivered output is an image with a unique identifier for each cell, which was the one obtained from the watershed, and a CSV document matching each unique identifier to its class. Thus, the pathologists will be able to discover the number of cells of each class only by reading the CSV document and printing the image will know the shape, position and volume of each cell.
Figure 17 illustrates an example of a final multiclass classification in (a), an identifier for each individual cell in (b) and for each identifier, the predicted class. For example, the first cell is the top right one where the prediction was negative and thus the CSV file contains "1,1". The second cell, the one below the first, was predicted positive and thus the "2,2" in the file.



(a) Final multiclass classification.



(b) A label for each individual cell.

1,1	2,2	3,2
4,2	5,2	6,1
7,2	8,3	9,1
10,1	11,2	12,1
13,2	14,1	15,1
16,2	17,1	18,2
19,2	20,1	21,1
22,2	23,2	

(c) CSV file: first value is the identification of the cell, second value the number of the class.

Figure 17: In (a) the image resulting from the watershed and homogenization, in (b) the image delivered to the pathologists in (c) the delivered csv file.

4. Final results

The whole algorithm was trained and optimized using the Ki67 database version 2, which was split between training and validation randomly. As said previously, there was no test subset because of the low number of images in the database. That is a risk, knowing that a whole set of new images could use this algorithm and the results should be similar but it must be assumed since a deep learning algorithm needs a lot of data to train and no other images are currently available.
The metrics in the following sections are cell based considering cells of each class in the whole validation dataset.

4.1 Ki67

Version 2

The dataset used for the detection and classification of the Ki67 antigen contains 42 images, that after an optimization, were split randomly (80-20) for the training and validation datasets. When the best hyperparameters are optimized and the final algorithm is obtained, the results for this second version can be shown.

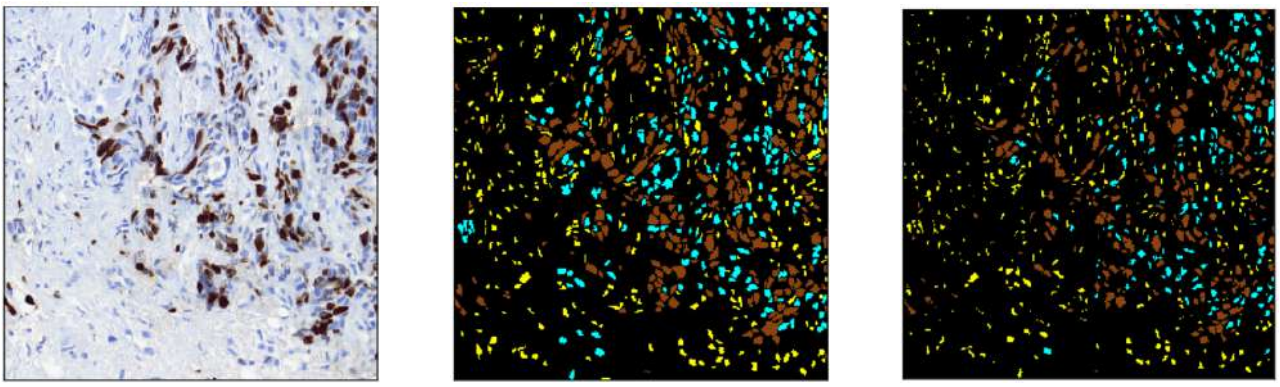
The algorithm gives a final cell based weighted F-score of 0.7811 with high F-scores per class in the case of negative and positive classes, reaching around 0.8 in both cases. The stroma class, though, does not achieve 0.6 but, as it does not take part in the calculation of the Ki67 score, the predicted Ki67 score is very similar to the ground truth one. The final mean absolute error between the ground truth and the predicted Ki67 score is 0.0368, which taking into account that the score covers from 0 to 1, it represents a 3.68% of error, a very good result. See table 5.

	Negative	Positive	Stroma	Weighted
F-score	0.7957	0.8231	0.5733	0.7811
Precision	0.8488	0.854	0.5934	0.8236
Recall	0.7488	0.7943	0.5546	0.7429

MAE Ki67 score
3.68%

Table 5: Metrics for the version 2 dataset: F-score, precision and recall for each class and its weighted value. The mean absolute error of the Ki67 score.

To visually see the results, some examples are shown. In figure 18, the original and ground truth images of a complex image in the validation dataset are seen. It has a lot of small cells with strange shapes and many stroma cells. The main problem in this image is the detection of the negative class, which is highly confused with stroma. The stroma class is detected with a F-score of 0.7248 and the positive class is the one most accurate, with 0.7964. See table 6. The weighted F-score reaches 0.7131 which is one of the lowest weighted F-score in all the validation images. Despite this, the absolute error of the Ki67 score is 1.3%, thus the algorithm predicts very accurately the score for this image.



(a) Original image (b) Ground truth mask (c) Final prediction

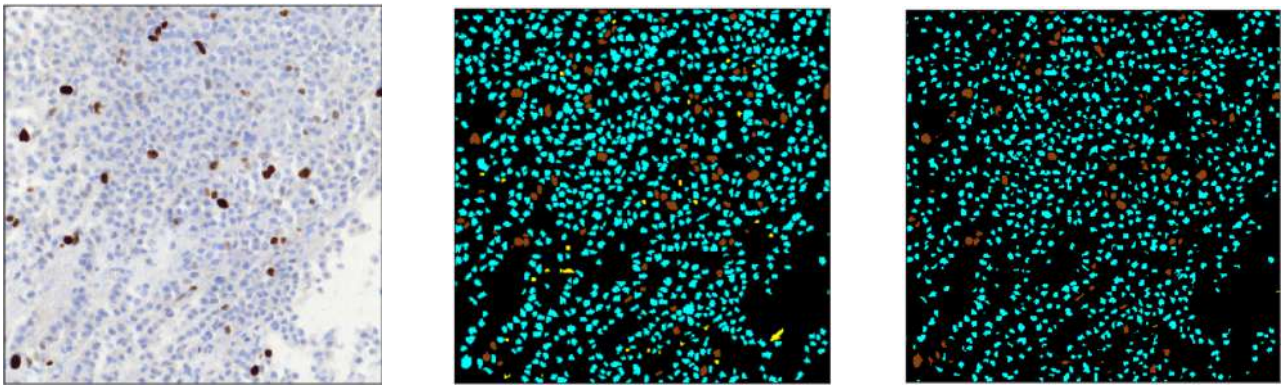
Figure 18: Example of original, ground truth and predicted images from a complex image.

	Negative	Positive	Stroma	Weighted
F-score	0.5649	0.7964	0.7248	0.7131
Precision	0.6622	0.905	0.6718	0.755
Recall	0.4925	0.711	0.7868	0.6893

Ki67 score	
GT	0.6075
Pred	0.6205
AE	1.3%

Table 6: Metrics of a complex image: F-score, precision and recall for each class and its weighted value. Ki67 score for the ground truth image, the predicted and the absolute error in percentage.

To contrast the low negative F-score in the previous example, the image in figure 19 basically contains negative cells and gives the most precise prediction between the validation images with a weighted F-score of 0.8865 and very balanced precision and recall (see table 7). There are not many stroma cells and the algorithm is not capable of detecting them in the majority of cases. Hence, the stroma F-score being so low and the negative so high. In this case the image has a low Ki67 score and the algorithm is able to predict it very accurately.



(a) Original image (b) Ground truth mask (c) Final prediction

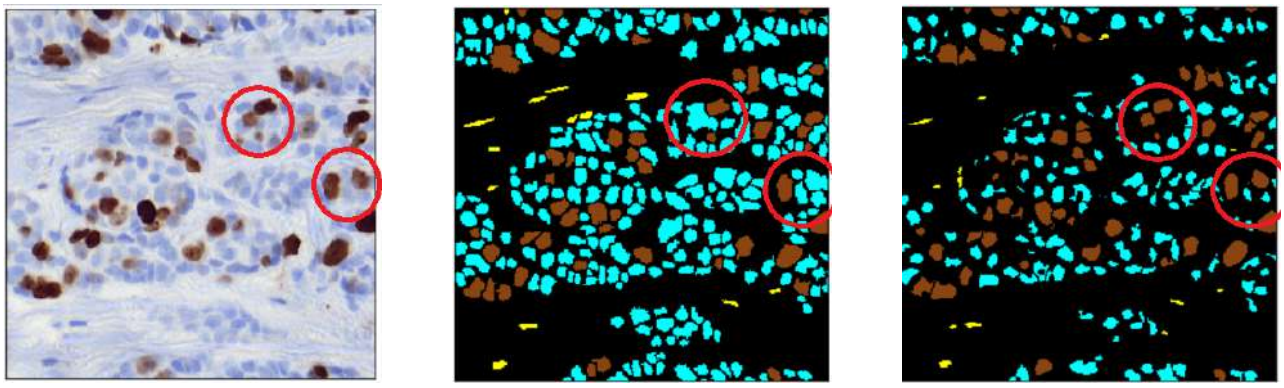
Figure 19: Example of original, ground truth and prediction of an image with many negative cells.

	Negative	Positive	Stroma	Weighted
F-score	0.914	0.8392	0.0645	0.8865
Precision	0.9135	0.8108	0.25	0.889
Recall	0.9145	0.8696	0.037	0.8882

Ki67 score	
GT	0.0695
Pred	0.0741
AE	0.56%

Table 7: Metrics of an image with many negative cells: F-score, precision and recall for each class and its weighted value. Ki67 score for the ground truth image, the predicted and the absolute error in percentage.

Despite these good results, there are cases where the ground truth labeling is questionable. The red circles in figure 20 illustrate some cells with positive class characteristics that the ground truth has classified as negative. In cases like this, the algorithm predicts the expected classes but the numerical results fall. The whole image has the lowest positive F-score between all the validation images, with a 0.7603.



(a) Original image (b) Ground truth mask (c) Final prediction

Figure 20: Example of a 250x250 region of an image where some labeling errors can be seen.

In general, the algorithm predicts the classes of the cells accurately, with the exception of the stroma class which is more difficult. By adding more stroma images to the training set the F-score could increase. What may be also improved is the visualization of the predicted images since the cells do not have a rounded-shape as the majority in the ground truth images. This is due to the semantic segmentation algorithm which predicts the classes pixel by pixel and cannot take into account the shape and other context information. However, the prediction of the Ki67 score is very precise, which is the important numerical value the pathologists need to determine the diagnosis and treatment.

Version 3

After a deeper revision of the ground truth images carried out by the DigiPATICS group, many inconsistencies and wrong detections and classifications were found due to the fact that, in non-tumoral zones, cells with the same characteristics as the positive and negative classes must be labeled as stroma. If the ground truth images do not take into account the morphology of the area, the model will not be able to differentiate these zones and will get confused. Furthermore, after this second revision, some images with a lot of stroma and negative cells that were previously removed but did not have a lot of labeling errors

were added. Finally, this redistribution resulted in a third version dataset with, 42 images from 18 different patients.
In this case, the training-validation split was reconsidered: the stroma class main characteristics are not the color and shape, it basically depends on the morphology of the area which changes depending on the patient. Therefore, a new partition splitting the training and validation subsets taking into account the patient was created. In other words, all the images from 12 patients were used to train and the rest, to validate. It resulted in 28 images for the training set and 14 for the validation set. Whereas 28 training images contain a lot of different cells, using only 12 patients for the training represents a very low variability. So one cannot expect the algorithm to really generalize well to unseen patients. In the long term, the algorithm is expected to be trained with many more patients and its generalization capability should improve.

The final algorithm was trained with this dataset without optimizing any hyperparameter. It was an interesting experiment to evaluate how the results were affected by the change of partition. In general, the weighted F-score decreased from 0.7811 to 0.727 which does not seem a negative result taking into account that the algorithm cannot generalize well when dealing with unseen patients. Negative and stroma F-scores have slightly decreased, as it was expected, but the positive F-score, precision and recall have not changed significantly. The mean absolute error of the Ki67 score has increased until 8.54%, which is nonetheless a good result taking into account the low variability of patients. See table 8.
However, taking a look at the individual values, two outliers could be seen, one with 64% of error and the other with 38%. Both images belong to the same patient and are two of the most complex images to predict in this database since there appear non-tumoral zones where many cells with negative class characteristics should be classified as stroma but the model classifies them as negative. If both images were omitted, the Ki67 score error would be reduced to 1.4%.

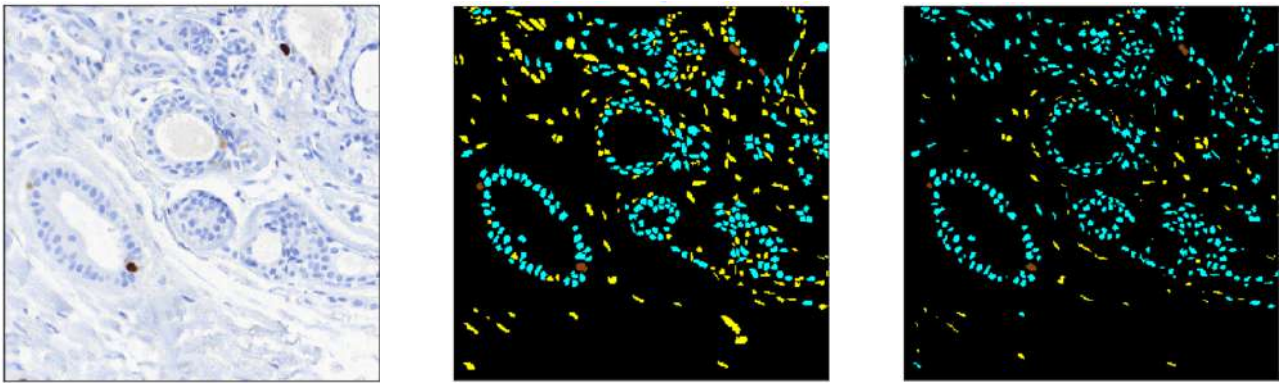
	Negative	**Positive**	**Stroma**	**Weighted**
F-score	0.7795	0.8213	0.5338	0.727
Precision	0.7813	0.8515	0.6736	0.7672
Recall	0.7778	0.7932	0.4421	0.6988

MAE Ki67 score
8.54%

Table 8: Metrics for the version 3 dataset: F-score, precision and recall for each class and its weighted value. The mean absolute error of the Ki67 score.

There is a group of 3 images belonging to the same patient that has achieved low values of weighted F-score. The three images have similar appearance: almost no positive cells, negative cells creating circles and stroma cells around them. Their weighted F-score does not exceed 0.64 while the rest of images move from 0.7 till 0.82. There is an image with very doubtful labelings whose F-score is 0.5 and it is considered an outlier.
One of the images in the group is illustrated in figure 21. It is a complex image since the main objective is to distinguish the negative and stroma classes, which is the most difficult task for the algorithm. Moreover, there are stroma cells detected on the ground truth that the model is not able to detect. For example, the bottom right corner contains two large stroma cells (in yellow) that the final prediction does not have, but when looking at the original image it seems that no cells should be detected.
In this case, the positive F-score is high since almost all the few positive cells are correctly detected, the negative metric is also high because of the quantity of cells in the class but the stroma F-score is too low.

However, the Ki67 score is predicted with an absolute error of 0.83%. See table 9.



(a) Original image (b) Ground truth mask (c) Final prediction

Figure 21: An example of a group of images from a particular patient that have almost no positive cells and achieve a low weighted F-score.

	Negative	Positive	Stroma	Weighted
F-score	0.718	0.8571	0.4247	0.5823
Precision	0.6029	1	0.7209	0.6617
Recall	0.8874	0.75	0.301	0.6122

Ki67 score	
GT	0.017
Pred	0.0087
AE	0.83%

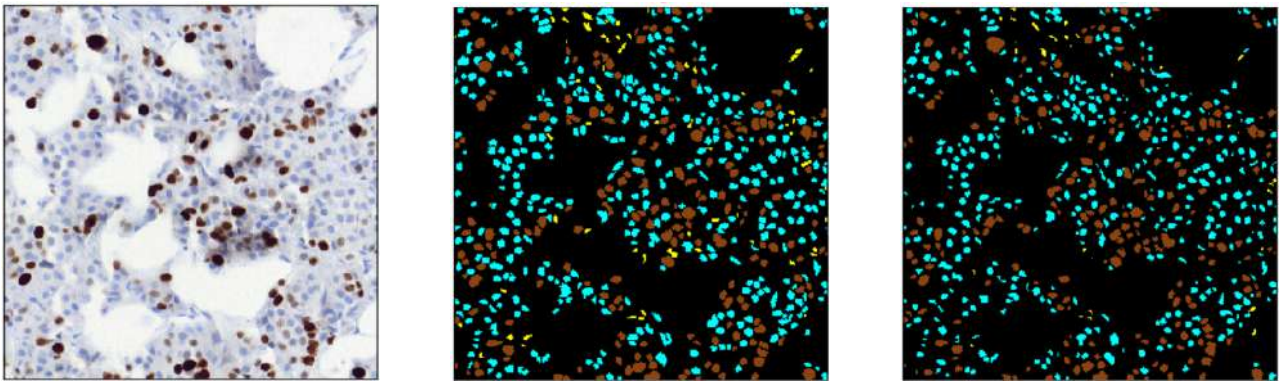
Table 9: Metrics of an image from a particular patient with many negative cells and low weighted F-score: F-score, precision and recall for each class and its weighted value. Ki67 score for the ground truth image, the predicted and the absolute error in percentage.

A completely different example could be the one in figure 22. In this case the weighted F-score reaches 0.8245. See table 10. The algorithm is able to detect the few cells that are clearly stroma but other small cells are missed and that is the cause of the low F-score for this class. The negative and positive classes are accurately predicted with 0.85 and 0.89 F-scores, respectively. Taking a look at the original and ground truth images, at the top left corner there is a big positive round cell omitted in the ground truth image, which is clearly an error, but it is detected by the algorithm and correctly classified. The Ki67 score is still correctly predicted with a 0.7% of error.

	Negative	Positive	Stroma	Weighted
F-score	0.8528	0.889	0.4222	0.8245
Precision	0.8396	0.878	0.6129	0.8248
Recall	0.8773	0.9	0.322	0.8333

Ki67 score	
GT	0.3431
Pred	0.3361
AE	0.7%

Table 10: Metrics of an image with more positive cells: F-score, precision and recall for each class and its weighted value. Ki67 score for the ground truth image, the predicted and the absolute error in percentage.



(a) Original image (b) Ground truth mask (c) Final prediction

Figure 22: Example of an image with more positive cells.

In general, the algorithm has behaved slightly worse with the new redistribution of the database, but taking into account the possible errors in the database, the low variability of patients and the detected outliers, it maintains good metrics. As said, in the future more patient images could be added in the dataset and the generalization for this kind of split would potentially improve.

4.2 ER

One of the objectives of the project was to create an algorithm to detect and classify tumor zones for Ki67 and ER biomarkers. The approach was to create the algorithm for the Ki67 database, which contains cell nuclei of 3 different classes, and after the best algorithm was obtained, see if it could generalize to other kinds of biomarkers. The ER database also contains images with negative nuclei and stroma but the positive class is now divided into 3 different classes, not easily differentiable at first sight. In Ki67 the negative class was the most represented one unlike the ER database, which is highly underrepresented. See table 11.

	Negative	**Positive type3**	**Positive type2**	**Positive type1**	**Stroma**
Mean nº of cells (%)	6.06	60.97	15.88	10.56	6.53
Mean nº of pixels (%)	9.46	65.38	13.02	7.65	4.49

Table 11: Number of cells and number of pixels for each class in the ER dataset (type3 is dark positive, type2 is medium positive, type1 is light positive).

The best hyperparameters of each block were used to train this new database. No modifications were proposed since the optimizations of the algorithm when training the Ki67 database showed robust results. The training for ER resulted in a weighted F-score of 0.7583, not bad results taking into account the differences in the database exposed previously. However, looking at the predicted images, all the stroma cells were omitted due to the approach 1 decision: the cell count algorithm (B) detected the stroma cells but the cell segmentation (A) did not. Therefore, during block C, those centers were removed. This showed that the cell count algorithm generalized better to this database than the cell segmentation. Both the selection of approach 2 to treat all points and the attempt to improve the cell segmentation algorithm could probably affect the final results positively.
The decision taken was to improve the cell segmentation algorithm with a small change to achieve the detection of stroma cells and thus recover them at the final prediction. The learning rate was lowered from 0.0005 to 0.00005, which was proved to help the algorithm detect the stroma cells during the cell segmentation optimization, and the number of epochs was enlarged from 200 to 300 to ensure the convergence.

With this slight modification the results improved significantly achieving a final weighted F-score of 0.8183 and an ER score mean absolute error of 5.162, which taking into account that this score ranges from 0 to 300, represents a 1.72% of error, an extremely good result. The dark positive (type 3) class achieves the highest F-score, 0.9127 with balanced precision and recall. The negative class almost reaches 0.8 and the rest of positives and the stroma classes move around 0.66. See table 12.

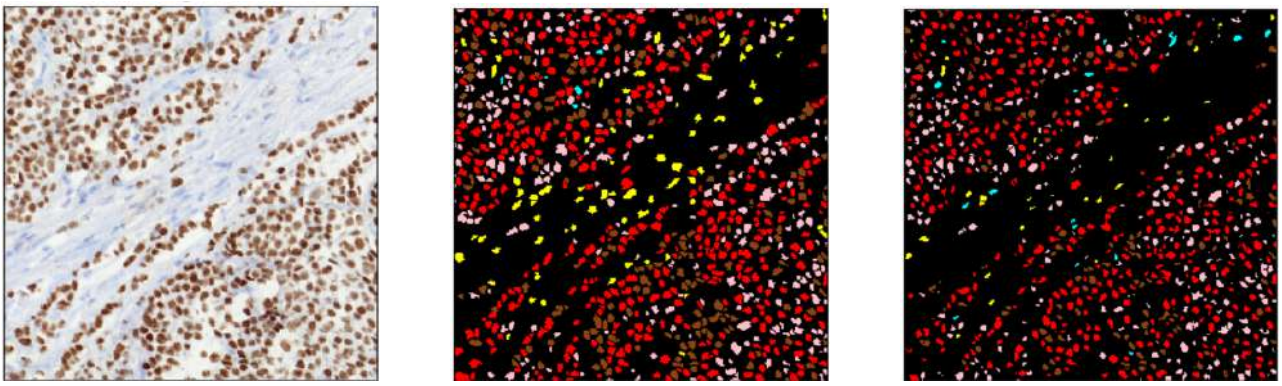
	Negative	**Positive type3**	**Positive type2**	**Positive type1**	**Stroma**	**Weighted**
F-score	0.7935	0.9127	0.6755	0.6581	0.672	0.8183
Precision	0.8185	0.9237	0.707	0.784	0.7283	0.8509
Recall	0.7699	0.902	0.6467	0.5671	0.6237	0.7911

MAE ER score
1.72%

Table 12: Metrics for the ER dataset: F-score, precision and recall for each class and its weighted value (type3 is dark positive, type2 is medium positive, type1 is light positive). The mean absolute error of the ER score.

It is clear that the dark positive cells (type3) are predicted almost perfectly and those images with many cells of this class are going to have high weighted F-score. In particular, the images with the highest F-scores, around 0.89, contain basically dark positive cells. Thus, other images are going to be illustrated to see the difficulties in distinguishing the 3 kinds of positive cells and the stroma and negative classes.

An example of how the model struggles distinguishing positive cells is in figure 23, where many positive cells of the three classes are observed. In table 13, the F-scores per class show that the most accurate one is the positive type 2 (medium) closely followed by the positive type 1 (light). However, the negative class is observed and predicted but not at the correct locations and thus, it gives a 0 F-score. The ER score is difficult to predict for this kind of images because of the hard distinction between positive types but even so, the error does not reach the 5%.



(a) Original image (b) Ground truth mask (c) Final prediction

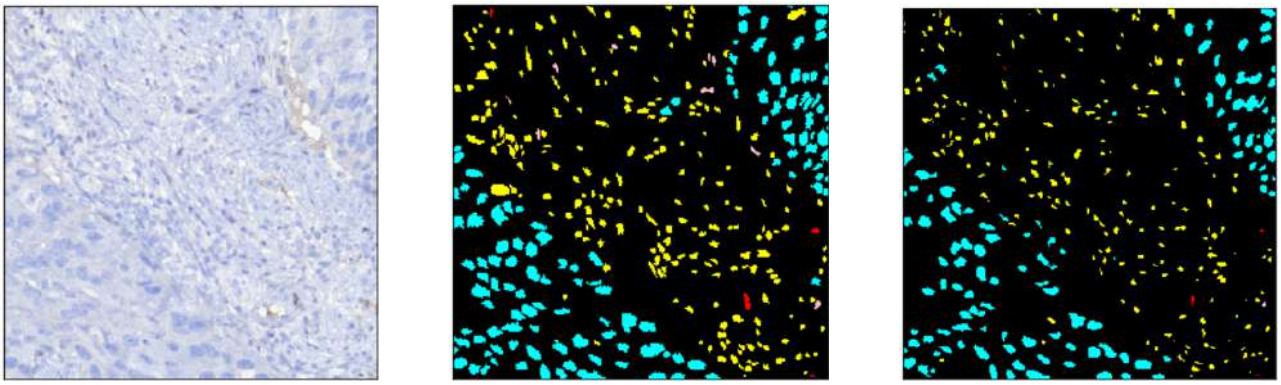
Figure 23: Example of an image with the three positive types.

	Negative	Positive type3	Positive type2	Positive type1	Stroma	Weighted
F-score	0	0.7031	0.779	0.7632	0.549	0.738
Precision	0	0.8957	0.7743	0.7329	0.875	0.7898
Recall	0	0.5787	0.7838	0.7961	0.4	0.7137

ER score	
GT	189.7
Pred	175.7
AE	4.67%

Table 13: Metrics of an image with more positive cells: F-score, precision and recall for each class and its weighted value. ER score for the ground truth image, the predicted and the absolute error in percentage.

An example of the hard distinction between stroma and negative cells can be seen in figure 24. It reaches a weighted F-score of 0.73091, one of the lowest in the validation set. The algorithm confuses some stroma cells with negative cells and misses other cells. However, looking at the original image, it is a very hard task for the human visual system and still the algorithm is able to compute the ER score with a 1.61% of error. See table 14.



(a) Original image (b) Ground truth mask (c) Final prediction

Figure 24: Example of an image with negative and stroma cells.

	Negative	Positive type3	Positive type2	Positive type1	Stroma	Weighted
F-score	0.7918	-	0.75	0.2	0.7303	0.7391
Precision	0.822	-	0.75	1	0.7537	0.7843
Recall	0.7638	-	0.75	0.1111	0.7083	0.7135

ER score	
GT	12.14
Pred	7.32
AE	1.61%

Table 14: Metrics of an image with negative and stroma cells: F-score, precision and recall for each class and its weighted value. ER score for the ground truth image, the predicted and the absolute error in percentage.

As a final conclusion, with a few adjustments on the hyperparameters the algorithm has reached a very high final weighted F-score and an extremely low error in ER score. Therefore, the algorithm has generalized well with the new database despite the number of classes and its characteristics being different. The most important metric for pathologists to determine the diagnosis is achieved with 1.72% of error and the visualization of the results, even though is not as accurate as desired, gives a similar global sight as the ground truth images.

5. Conclusions

The initial goals of the project were to develop a PyTorch platform to create and train an algorithm capable of detecting and classifying cells in histology images in the context of breast cancer. In particular, detect tumor zones for three different biomarkers: Ki67, ER and PR proteins. However, until the last few weeks of the study the only available database was Ki67, which contains less images than a deep learning algorithm typically uses. Afterwards, ER ground truth images were created and some experiments could be made, but the PR database never arrived. As a consequence, the objectives had to be modified. Furthermore, the Ki67 database was studied by the DigiPATICS group and errors were discovered in many ground truth images when classifying and detecting the cells. There was inconsistency when distinguishing stroma cells because of the morphology of each area. In addition, since the pathologists' task of evaluating this type of images is tedious and slightly subjective, variability appeared in the labelings, which made the objective of the project more complex.

Detecting and classifying cells could have been done only with the semantic segmentation approach from block A and the resulting images would be quite similar to the ground truth ones. However, pathologists need the score that takes into account the number of cells per class and the intensity of the class. Therefore, a semantic segmentation model with which a connected component could include more than one cell could not be the final algorithm. Consequently, another goal of the project was to identify the cell nuclei individually, which were obtained thanks to the centers predicted from block B and the combination of predictions in block C. The regression from block B predicts very accurately the center of all kinds of cells regardless of the database used. The watershed algorithm splits precisely the connected components with more than one center, which allows the final homogenization process to assign a class to each individual cell even if they are part of the same connected component. Otherwise, the algorithm would not have reached such a low mean absolute error neither in Ki67 nor in ER scores.

A challenge for this project was to create an algorithm optimized for a certain Ki67 database that was able to achieve similar results for ER, as well. The final experiment that trained the ER database shows that the algorithm has been able to successfully generalize a new set of images even with different number of cell classes and distinguishing intensities of positive cells, which in some cases is a really hard and subjective task for humans. Furthermore, during the whole development of the project, robustness in the results and the stability in the models convergence was observed while changing certain hyperparameters. Thus, the generalization to a database similar to the Ki67 one, in case a test set is available, might be confirmed.

As future work, the first aspect to improve would be the databases. The datasets should be expanded with images from more patients, coming from other hospitals and analyzed by different pathologists since no meaningful generalization will be achieved if no variability is present in the databases. However, the ground truth images should be correctly checked because training a deep learning algorithm with errors in the database could make the model learn with those errors. In addition, the expansion of the databases could be useful to create a test dataset and achieve final metrics taking images that were never used before. When the PR database will be available, one should of course check if the algorithm can also generalize as in ER and perhaps, optimize the hyperparameters.
Taking into account the final algorithm, block A could be improved by trying to deal better with the unbalanced classes of the datasets to achieve a higher F-score and then perhaps all classes would be detected when generalizing to other databases without the need of changing the models hyperparameters.

Moreover, the detection of the cell shape would probably improve. The block C main decision was to select the common predictions from cell segmentation and cell count (approach 1), but another approach could be better for a different database. For example, with ER keeping all the centers predicted from the cell count algorithm seemed to be a good solution to help the algorithm detect the stroma cells. Therefore, other approaches should be tested when new datasets will be available.

Despite the low number of images, the possible errors in the databases, the time spent optimizing the models and the errors from the development itself, the algorithm is able to distinguish quite accurately the different classes of the database and the score that pathologists need to determine the diagnosis is predicted with a very low error. When taking a look at images checked by more than one pathologist, much more disagreement between human experts can be observed and therefore, from a non-expert point of view, it seems that the algorithm is providing a high precision or at least can be one of the basis to consider for the final diagnosis. However, only pathologists themselves can determine if it is accurate enough to be implemented in practice and this type of real test will be done in the near future in the context of the DigiPATICS project.

References

[1] Pau Magariño (2021). Cancerous nuclei segmentation using object detection. UPC.

[2] Cristina Aguilera (2021). Cell nuclei detection and segmentation in histological images based on a deep semantic segmentation network. UPC.

[3] Weidi Xie, J. Alison Noble & Andrew Zisserman (2018). Microscopy cell counting and detection with fully convolutional regression networks, Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization, 6:3, 283-292, DOI: 10.1080/21681163.2016.1149104.

[4] Victor Lempitsky, Andrew Zisserman (2010). Learning to count objects in images. Visual Geometry Group, University of Oxford.

[5] Olaf Ronnegergen, Philipp Fischer and Thomas Brox (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. Computer Vision and Pattern Recognition, Cornell University.

[6] Fernando Candanedo-González, Eduardo Pérez-Salazar (2012). El papel de la progesterona en cáncer de mama. Gaceta Mexicana de Oncología, Vol. 11. Núm. 3, 182-188.

[7] Long, J.; Shelhamer, E.; Darrell, T. (2014). Fully convolutional networks for semantic segmentation. Computer Vision and Pattern Recognition, Cornell University.

[8] BreastCancer.org. HER2 Status. `https://www.breastcancer.org/es/sintomas/diagnostico/her2`

[9] Carolina Martinez Ciarpaglini (2021). Proyecto de Innovación docente. Universidad de Valencia. `https://www.youtube.com/watch?v=TpnN9oGTwo0`

[10] Kimberly H. Allison, M. Elizabeth H. Hammond, Mitchell Dowsett, Shannon E. McKernin, Lisa A. Carey, Patrick L. Fitzgibbons, Daniel F. Hayes, Sunil R. Lakhani, Mariana Chavez-MacGregor, Jane Perlmutter, Charles M. Perou, Meredith M. Regan, David L. Rimm, W. Fraser Symmans, Emina E. Torlakovic, Leticia Varella, Giuseppe Viale, Tracey F. Weisberg, Lisa M. McShane, Antonio C. Wolff. Estrogen and Progesterone Receptor Testing in Breast Cancer: ASCO/CAP Guideline, Update. J Clin Oncol. 2020 Apr 20;38(12):1346-1366. doi: 10.1200/JCO.19.02309. Epub 2020 Jan 13. PMID: 31928404.

[11] Shruti Jadon (2020). A survey of loss functions for semantic segmentation. Image and Video Processing, Cornell University.

[12] Hammond ME, Hayes DF, Dowsett M, Allred DC, Hagerty KL, Badve S, Fitzgibbons PL, Francis G, Goldstein NS, Hayes M, Hicks DG, Lester S, Love R, Mangu PB, McShane L, Miller K, Osborne CK, Paik S, Perlmutter J, Rhodes A, Sasano H, Schwartz JN, Sweep FC, Taube S, Torlakovic EE, Valenstein P, Viale G, Visscher D, Wheeler T, Williams RB, Wittliff JL, Wolff AC; American Society of Clinical Oncology; College of American Pathologists (2010, Jul). American Society of Clinical Oncology/College of American Pathologists guideline recommendations for immunohistochemical testing of estrogen and progesterone receptors in breast cancer (unabridged version). Arch Pathol Lab Med. 134(7):e48-72. doi: 10.5858/134.7.e48. PMID: 20586616.

[13] Pavel Yakubovskiy (2020). Segmentation Models Pytorch. Published on GitHub repository `https://github.com/qubvel/segmentation_models.pytorch`

A. Appendix

A.1 Binary cell segmentation algorithm: model hyperparameters & split optimization

The cell segmentation algorithm aims to detect and classify every pixel given and define a set of cells for the Ki67 dataset. The first approach to this algorithm was to optimize the hyperparameters of the U-Net network to achieve a first initial idea of the values to study during the optimization of the different algorithms. However, since the multiclass classification was a bit more complex, only the shape and position of the cells were used, the classes were not taken into account. Therefore, the ground truth images were binary with 1 where the original image has a cell and 0 otherwise.

The kind of separation between the training and validation images was a first hyperparameter (**TrainValid**). Related to the model, the optimizer, its learning rate, the number of epochs and the batch size were important parameters to take into account since they have an important role to play in the model convergence. The optimizer (**opt**) used in the algorithm originally was Adam with a learning rate (**lr**) of 0.00068 but SGD and a higher value (0.005) were added to the optimization. The convergence was achieved with 100 **epochs** but the changing of the parameters could lead to a slower learning so, perhaps, 200 epochs could be necessary. Finally, the number of images taken at each minibatch for the training step was 4, so 2 and 6 were added to the optimization (**batch_size**).

Therefore, a first grid search was developed retrieving the pixel by pixel metric F-score for each hyperparameter configuration.

The first thing to notice in figure 25, is that the kind of split between the training and validation sets does not have much influence on the results since the maximum F-score is similar when only changing the **TrainValid** hyperparameter. Therefore, the random split was chosen.
With respect to the model hyperparameters, the batch_size=6 does not improve the results in any case and the value of 2 is the one that leads to the best results. Besides taking much longer time, learning during 200 epochs does not provide much improvement in the results since in most cases the convergence is achieved before 100 epochs. The SGD optimizer does not converge as fast as Adam optimizer, but highering the learning rate to 0.005 the results improved significantly. However, the Adam optimizer reaches better results more easily and faster.

To sum up, the optimal configuration of this optimization are: the 80-20 split, Adam optimizer with a learning rate of 0.005, during 100 epochs with 2 training images per batch. The learning curve in figure 26 shows that the convergence is already achieved with 20 epochs and overfitting is not a problem now.

TrainValid	lr	opt	epochs	batch_size	Max Fscore	in epoch
80-20	0,00068	adam	100	**2**	0,86296	42
80-20	0,00068	adam	100	**4**	0,86116	99
80-20	0,00068	adam	100	**6**	0,8593	92
80-20	0,00068	adam	200	2	0,86519	80
80-20	0,00068	adam	200	4	0,86062	102
80-20	0,00068	adam	200	6	0,85918	119
80-20	0,00068	sgd	100	2	0,73667	97
80-20	0,00068	sgd	100	4	0,65871	96
80-20	0,00068	sgd	100	6	0,61268	99
80-20	0,005	adam	100	2	0,86789	97
80-20	0,005	adam	100	4	0,86741	89
80-20	0,005	adam	100	6	0,86637	94
80-20	0,005	sgd	100	2	0,8174	98
80-20	0,005	sgd	100	4	0,79571	99
80-20	0,005	sgd	100	6	0,78528	99
1perPatient	0,00068	adam	100	2	0,85707	55
1perPatient	0,00068	adam	100	4	0,85344	37
1perPatient	0,00068	adam	100	6	0,85164	78
1perPatient	0,00068	adam	200	2	0,85586	61
1perPatient	0,00068	adam	200	4	0,85375	41
1perPatient	0,00068	adam	200	6	0,85341	199
1perPatient	0,00068	sgd	100	2	0,7442	95
1perPatient	0,00068	sgd	100	4	0,62964	96
1perPatient	0,00068	sgd	100	6	0,49032	99
1perPatient	0,005	adam	100	2	0,86441	98
1perPatient	0,005	adam	100	4	0,86277	41
1perPatient	0,005	adam	100	6	0,86275	70
1perPatient	0,005	sgd	100	2	0,81415	98
1perPatient	0,005	sgd	100	4	0,79462	99
1perPatient	0,005	sgd	100	6	0,77869	99

Figure 25: Optimization of the split, optimizer, learning rate, epochs and batch size for the binary cell segmentation algorithm. Maxim F-score and its epoch for each configuration of hyperparameters.

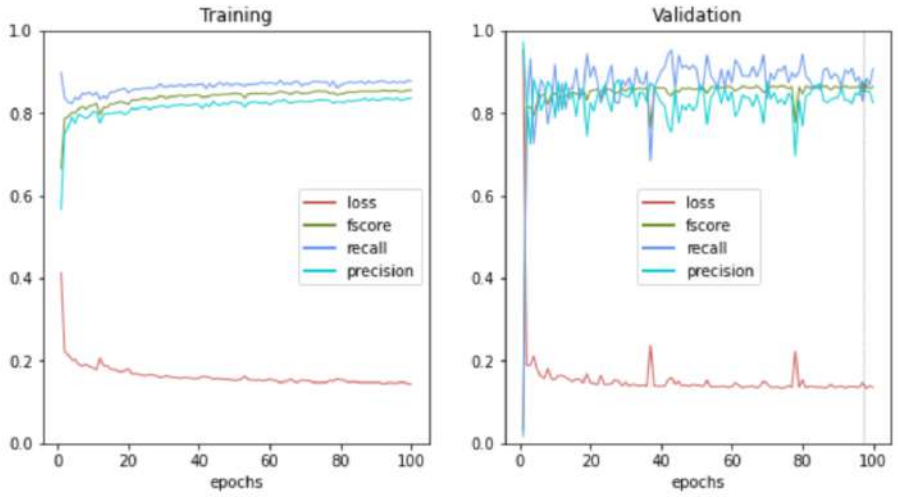


Figure 26: Best configuration. Maximum validation F-score of 0.86789 reached in epoch 97.

A.2 Optimization of the sigma and contrast parameters with each approach

On the one hand, during the optimization of the cell count algorithm two parameters needed to be checked: the sigma which widens the gaussian functions applied to each center of the ground truth images and the contrast value that helps a postprocessing method to achieve the maximum values of these gaussian functions.
On the other hand, the combination of the results lead to 4 different approaches depending on the centers and components to take into account. Since both sides influence each other, a single optimization was developed.

As said, sigma was optimized between 3 and 7 and contrast was optimized between 0.05, 0.15 and 0.25. Therefore, 15 cell count models were tested for each approach with the detection of the best cell segmentation model independently of the class predicted. The metrics computed are cell based F-score, precision and recall. Tables 15, 16, 17 and 18 show the final results.

Approach 1	**contrast=0.05**	**contrast=0.15**	**contrast=0.25**
sigma=3	fscore=0.877, precision=0.899, recall=0.861	fscore=0.875, precision=0.898, recall=0.858	fscore=0.871, precision=0.915, recall=0.837
sigma=4	fscore=0.874, precision=0.88, recall=0.872	fscore=0.874, precision=0.909, recall=0.847	fscore=0.872, precision=0.915, recall=0.838
sigma=5	fscore=0.874, precision=0.879, recall=0.873	fscore=0.877, precision=0.904, recall=0.856	fscore=0.875, precision=0.914, recall=0.843
sigma=6	fscore=0.877, precision=0.879, recall=0.878	fscore=0.878, precision=0.904, recall=0.857	fscore=0.871, precision=0.913, recall=0.836
sigma=7	fscore=0.874, precision=0.88, recall=0.872	**fscore=0.879, precision=0.895, recall=0.865**	fscore=0.864, precision=0.877, recall=0.853

Table 15: Optimization of sigma-contrast values for Approach 1: the intersection of both algorithms is taken into account. Metrics are cell based F-score, precision and recall. In bold, the metrics of the configuration with the highest F-score and more balanced precision and recall.

Approach 2	h=0.05	h=0.15	h=0.25
sigma=3	fscore=0.873, precision=0.866, recall=0.887	fscore=0.871, precision=0.866, recall=0.884	fscore=0.871, precision=0.896, recall=0.855
sigma=4	fscore=0.863, precision=0.831, recall=0.904	fscore=0.876, precision=0.891, recall=0.867	fscore=0.873, precision=0.897, recall=0.857
sigma=5	fscore=0.863, precision=0.833, recall=0.902	fscore=0.879, precision=0.883, recall=0.879	fscore=0.877, precision=0.894, recall=0.865
sigma=6	fscore=0.866, precision=0.83, recall=0.912	**fscore=0.879, precision=0.88, recall=0.882**	fscore=0.873, precision=0.903, recall=0.849
sigma=7	fscore=0.868, precision=0.843, recall=0.901	fscore=0.876, precision=0.863, recall=0.892	fscore=0.864, precision=0.857, recall=0.874

Table 16: Optimization of sigma-contrast values for Approach 2: all the centers from the cell count algorithm are taken into account. Metrics are cell based F-score, precision and recall. In bold, the metrics of the configuration with the highest F-score and more balanced precision and recall.

Approach 3	h=0.05	h=0.15	h=0.25
sigma=3	fscore=0.87, precision=0.856, recall=0.888	fscore=0.867, precision=0.855, recall=0.885	fscore=0.867, precision=0.863, recall=0.876
sigma=4	fscore=0.866, precision=0.845, recall=0.894	fscore=0.866, precision=0.857, recall=0.881	fscore=0.865, precision=0.861, recall=0.874
sigma=5	fscore=0.865, precision=0.844, recall=0.893	fscore=0.87, precision=0.859, recall=0.885	fscore=0.866, precision=0.861, recall=0.875
sigma=6	fscore=0.868, precision=0.843, recall=0.898	**fscore=0.871, precision=0.859, recall=0.886**	fscore=0.865, precision=0.854, recall=0.879
sigma=7	fscore=0.867, precision=0.841, recall=0.898	fscore=0.87, precision=0.847, recall=0.896	fscore=0.853, precision=0.82, recall=0.891

Table 17: Optimization of sigma-contrast values for Approach 3: all the connected components from the cell segmentation algorithm are taken into account. The center of mass of the components without a center are computed and used as an approximation of the center. Metrics are cell based F-score, precision and recall. In bold, the metrics of the configuration with the highest F-score and more balanced precision and recall.

Approach 4	**h=0.05**	**h=0.15**	**h=0.25**
sigma=3	fscore=0.861, precision=0.826, recall=0.906	fscore=0.859, precision=0.825, recall=0.903	fscore=0.864, precision=0.847, recall=0.887
sigma=4	fscore=0.848, precision=0.797, recall=0.914	fscore=0.863, precision=0.84, recall=0.893	fscore=0.861, precision=0.842, recall=0.885
sigma=5	fscore=0.85, precision=0.801, recall=0.913	**fscore=0.866, precision=0.838, recall=0.9**	fscore=0.862, precision=0.84, recall=0.888
sigma=6	fscore=0.852, precision=0.798, recall=0.92	fscore=0.866, precision=0.836, recall=0.902	fscore=0.864, precision=0.844, recall=0.887
sigma=7	fscore=0.855, precision=0.805, recall=0.916	fscore=0.86, precision=0.816, recall=0.911	fscore=0.849, precision=0.801, recall=0.904

Table 18: Optimization of sigma-contrast values for Approach 4: all centers and all components from both algorithms are taken into account. Metrics are cell based F-score, precision and recall. In bold, the metrics of the configuration with the highest F-score and more balanced precision and recall.