SEGMENTATION OF IMAGE CELLS USING MASK-RCNN AND U-NET WITH CELLARI

Jonathan Gundorph (S153114) & Johanne Thorkilsdatter (S175385)

Technical University of Denmark

ABSTRACT

The amount of image material and high-quality cameras have grown exponentially in the last decade. This has led to computers being able to be able to recognise objects in images much better, faster and more accurately than the human eye can. In this project, two different methods will be used to identify cells in images by doing pixel-wise segmentation. The first approach is Mask R-CNN where the results will be compared to the results of a slightly more simple model, U-Net. Both are pre-defined network, and have then been customized by having their own unique parameters. Both models yield fairly good results in most cases, but some images have proven to be more difficult than others to segment, due to the different patterns found in the cell images.

***Index Terms*—** Mask-RCNN, U-Net, Image Segmentation, Medical Image, Deep Learning

1. INTRODUCTION

Clinicians and researchers spend an enormous amount of time and resources on repetitive, manual tasks, such as identifying and counting objects in images. The process of identifying objects can be highly subjective, which induces reproducibility issues. Segmentation of gland is challenging due to the great variation of glandular morphology. By using deep learning to aid clinicians and researchers in identifying objects in images, the repetitive and complex tasks can be reduced, while minimizing scientific reproducibility issues.

2. REVOLUTION OF CONVOLUTIONAL NEURAL NETWORKS

Since 2012 Convolutional Neural Networks (CNN) has been the golden standard for image classification. As the model has gotten more complex it has become possible to solve more refined image tasks. Previously, the most basic image tasks was usually solved by simple *classification*, where the output is the names of the classes in the image. The next step could then be to do *object detection/localization*, where bounding box coordinates are output for each object in the image. More advanced, a data scientist could choose to carry out *semantic segmentation* or pixel segmentation where the model assigns each pixel to a category/class. Or, if each individual instance of an object was of interest, the *instance segmentation* method, where the model assigns an individual object label to each pixel as a way of ex. counting the number of cells or number of people in an image. Here the label could be cell1, cell2, cell3, etc. An illustration of these four tasks can be seen in image 1.

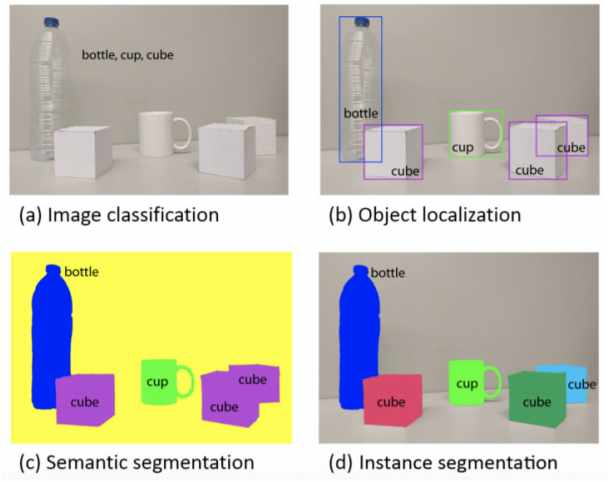

Fig. 1. Different visual perception problems

U-Net is a convolutional neural network to do semantic segmentation that was developed in 2015. It is a fully convolutional network and is called U-Net, due to the architectural structure of the model. The initial context of the model was biomedical images but it has shown great results on natural images as well. U-Net consists of a contracting path to detect the low-level features; the local intensity variations such as edges. Then an expansive path to detect the high-level features, which is the potential objects. This method works well for semantic segmentation but cannot do instance segmentation, only identify cell vs. background pixels and not separate individual cells.

The year before that, in 2014 Region based CNN (R-CNN) was developed, which identifies the main objects in

Thanks to Peter Jensen from Cellari for good supervision

an image with bounding boxes and labels. Here category-independent region proposals, called regions of interest (ROI) is created through selective search and these are passed through a modified version of the CNN, AlexNet; that made the breakthrough in 2012. Each ROI's output features are then passed through a Support Vector Machine (SVM) that classifies whether or not it is an object and if so, which object class it belongs to.

In 2015 the R-CNN was re-written to Fast R-CNN by simplifying the algorithm and speeding it up. This was done by using Region of Interest Pooling (RoI Pool) to be able to only run CNN once per image, instead of one time per proposed region. The RoI Pool shares the forward pass of a CNN across an image's subregions. Moreover the SVM classifier is replaced with a softmax layer on top of CNN to output a classification, and parallel to this is a linear regression layer to output bounding box coordinates for each object in the image. In this way, all the needed outputs come from one single network.
But the process to generate potential regions of interest with selective search still slowed down the network. In the middle of 2015, they realised that the features of the image were already being calculated in the CNN's forward pass and this could be reused instead of running a separate selective search algorithm. This made the runtime shorter and they therefore called this modified version Faster R-CNN.

In 2017 the Faster R-CNN was extended to also do pixel level segmentation, which is outputted as masks over the objects in the image, and therefore the architecture is known as Mask R-CNN. Mask R-CNN combines object detection and semantic segmentation. A branch that outputs a binary mask is added to the Faster R-CNN. This branch looks at each pixel in a bounding box and determines whether or not the pixel is a part of an object. To make this possible, the RoI Pool needed to be replaced with RoI Align to avoid misalignment.

For this project Mask R-CNN and U-Net will be used to solve pixel-wise segmentation task of cells. The main focus area being will be Mask R-CNN. The results from U-Net will be used for comparison and to evaluate the results of the Mask R-CNN.

3. DATASET

The dataset used for this project, the Warwick-QU dataset, was generated for *The GlaS Challenge Contest* in 2015 about gland segmentation on images of Hematoxylin and Eosin (HE) stained slides, together with ground truth annotations by expert pathologists. The dataset contains 165 annotated images of benign and malignant colorectal adenocaricinoma.

3.1. Preprocessing

To load the image information into the Mask R-CNN, a custom PyTorch class for loading and creating the cell image dataset is created. This class contains methods that are to be used in the PyTorch Dataloader.
In the dataset creation the following is done:

1. Defines paths for images, transforms and masks
2. Loads in the images and masks
3. Get unique pixel values for each of the two classes (background/foreground or cell)
4. Define masks and bounding boxes around each mask.

After the dataset is created, it is loaded into the PyTorch Dataloader, which is used to feed data into the Mask R-CNN. For training, a batch size of two is used, and for testing, a batch size of one is used.

4. MODEL SPECIFICATION: MASK R-CNN IN-DEPTH

As stated earlier, the Mask R-CNN is just an extension of the Faster R-CNN. The structure of the Mask R-CNN network can be seen illustrated in figure 2

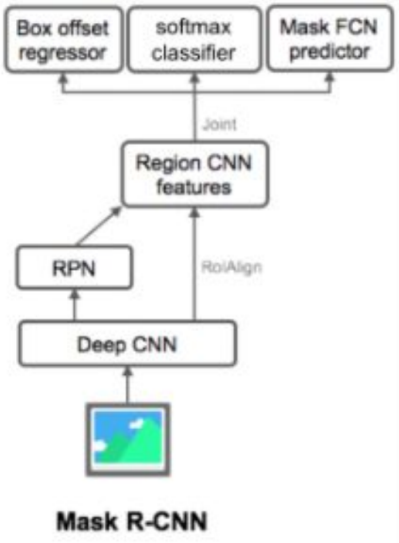

Fig. 2. Structure of the Mask-RCNN

Each step, after the image has been fed to the model, will be explained below.

1. First, the Mask R-CNN uses a Deep CNN to extract feature maps from the given images. This ConvNet is known as the backbone of the model.

2. The obtained feature maps are then passed through a Region Proposal Network (RPN), returning the candidate bounding boxes.

3. Following this, a RoI align layer is applied to these candidate bounding boxes, which resizes all candidates to the same size.

4. Then, the proposals are passed to a Fully Connected Layer (FC), which classifies and outputs the bounding boxes for various objects.

5. Finally, a segmentation mask is created for every region containing an object. This is done by adding a mask branch to the architecture.

The backbone of choice for the Mask R-CNN is the ResNet-50 network (Residual Network 50), which introduces residual learning (the usage of skip connections to jump over some layers) and consists of 50 layers. The general ResNet architecture is a type of network that revolutionized the research community in 2015, due to its introduction of "identity shortcut connections" which effectively solved the vanishing gradient issue. The vanishing gradient issue is that as the gradient is back-propagated to previous layers, the repeated multiplications that takes place can make the gradient infinitely small. The solution is to let the stacked layers of a network fit a residual mapping, instead of allowing them to directly fit to the underlying mapping, as shown in figure 3 [1, 2]. While ResNet-50 was chosen for this project, there exist several ResNet networks, such as the deeper ResNet-101 and ResNet-152.

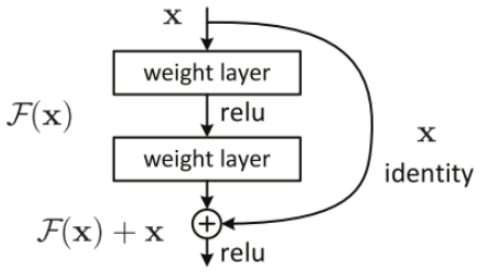

Fig. 3. A residual block

It has been proven that training networks this way is easier than doing it the conventional way, and it also resolves the issue of saturating and degrading accuracy, as the network goes deeper.

The backbone is initialized with weights of a model trained on the COCO dataset, which is a large dataset made for the purpose of object detection, segmentation and captioning. Having pre-trained weights from the COCO dataset gives a better starting point and thereby makes the training quicker and more accurate.

The optimizer used is the Stochastic Gradient Descent (SGD) with an initial learning rate of 0.005, a momentum of 0.9 and a weigh decay of 0.0005. The model starts with some "training warm-up steps" which means that the learning rate for the first epoch is very low. Afterwards the initial learning rate is used, but with a staircase-type learning rate decay scheduler, which decreases the learning rate in discrete steps by a factor of 10 after every third epoch. This allows the model to better converge toward the minimum, as smaller steps are taken as it approaches convergence. Apart from the Stochastic Gradient Descent, other optimizers were also attempted with different parameters, such as the *AdamW* optimizer, which features an adaptive learning rate. The *AdamW* optimizer is an improved version of the well known Adam optimizer, which should yield a better training loss and generalization, and also has a feature known as *amsgrad*, which seeks to fix convergence issues seen in the standard *Adam* optimizer. In theory, it should perform better or at least on par with SGD with momentum, but in practice the SGD still outperformed the AdamW for the results of this project.

5. MODEL SPECIFICATION: U-NET

U-Net is a deep convolutional neural network that is often used for biomedical image segmentation. The network was first published in the International Conference on Medical Image Computing and Computer-Assisted Intervention in 2015 [3]. It is a state of the art network that is modified to be able to work with a relatively small set of training images, and still yield precise segmentations. The reason for its name, U-Net, is due to the expansive/encoding part of the network being symmetric to the contracting/decoding part, thus forming a U-shape, as illustrated in figure 4.

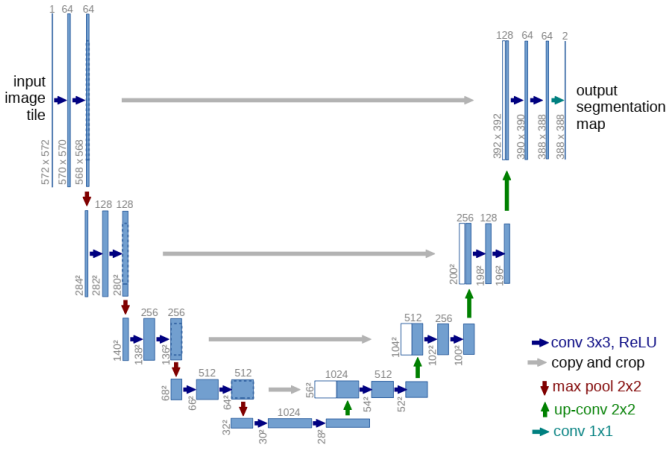

Fig. 4. U-Net Architecture

U-Net won the ISBI cell tracking challenge in 2015 by

a large margin. This was in large part due to the speed of the network, being capable of segmenting a single 512x512 image in less than a second, on a modern GPU. U-Net will, for the metrics where the results are comparable, be used as a baseline for comparison with the Mask R-CNN results.
Since the output is binary (cell/background), the optimizer used for U-Net is the *Adam* optimizer with a learning rate of 0.0001 and Cross Entropy Loss as the loss function.

6. PERFORMANCE EVALUATION METRICS

After training, the Mask R-CNN is evaluated using a number of different metrics: F1 score, F1 object score, IoU (Intersection over Union) score, IoU object score and Hausdorff object distance.

6.1. F1 score

The F1 score that measures the incorrectly classified cases. It is based on a ratio of Precision (PRC) and Recall (REC), which in turn is based on the number of correct and incorrect classified elements. A minimum area overlap of 50% between the true mask and the predicted mask is required to consider a detection a true positive (TP). Otherwise it is considered a false positive (FP). A false negative (FN) is a detection whose ground truth object has no corresponding segmented object, or which has less than 50% area overlap with the segmented object.
The PRC and REC is defined as

$$PRC = \frac{TP}{TP + FP} \qquad REC = \frac{TP}{TP + FN} \tag{1}$$

And the F1 score is defined as

$$F1 = \frac{2 \times PRC \times REC}{PRC + REC} \tag{2}$$

The more correct and fewer incorrect classifications there are, the lower the F1 score will be.
Another related metric is the Object F1, which is the same as F1 score but for individual objects. It is also called the object-level Dice score.

6.2. IoU score

IoU stands for Intersection over Union, and is also known as the Jaccard index. It is defined as the overlap between the predicted bounding box and the ground truth bounding box. As stated earlier, an overlap larger than a threshold of 50% is usually considered "good", or as a TP, but the threshold may be set higher as well. If IoU is 0 it means that there is no overlap and if it is 1 it means that the boxes are completely overlapping.

6.3. The shape similarity

The object-level Hausdorff distance measures the shape similarity. It is based on the Hausdorff distance, which is defined as

$$H(G,S) = \max\left\{\sup_{x\in G}\inf_{y\in S}||x-y||, \sup_{y\in S}\inf_{x\in G}||x-y||\right\} \tag{3}$$

where it takes the maximum distance from a point in a set of pixels belonging to the predicted object (S), to the closest point in a set of pixels annotated as belonging to the ground truth object (G). Based on this distance, the more advanced object-level Hausdorff distance is computed, which is the shape similarity between all segmented objects in S and all ground-truth objects in G.

For further elaboration on this and the other evaluation metrics, as well as an explanation of the IoU-object and object F1 scores (the object-level Dice index), please refer to [4].

7. RESULTS

7.1. Mask-RCNN results

The Mask-RCNN has large fluctuations in its performance, based on the "difficulty" of the image. This can be seen when comparing figure 5 to figure 6.

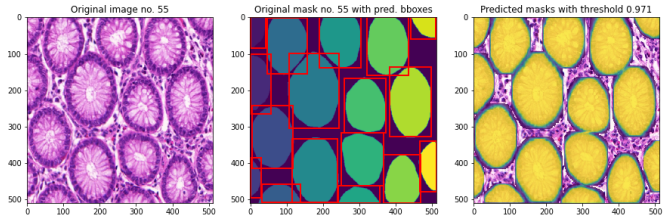

Fig. 5. Good prediction from Mask R-CNN

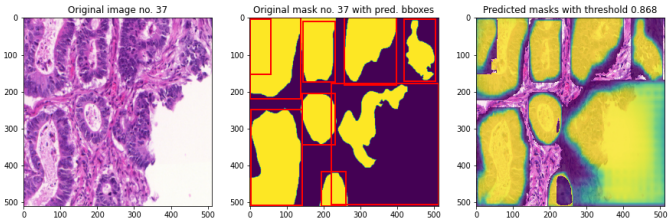

Fig. 6. Bad prediction from Mask R-CNN

The subfigures in figure 5 and figure 6 shows the output of the model. The upper left figure is the original image. The upper right is the original ground truth mask, but with the Mask R-CNN's predicted bounding boxes on top. The lower left image is the maximum values of the "allowed" masks. The way this is computed, is that each predicted mask in the image comes with a "score" value, a so-called confidence level

in how certain the algorithm is that it made a right prediction. There is only an interest in picking the masks above a certain threshold, to avoid introducing noise into the prediction, when all the masks are added together. Hence, a threshold value of 0.77 is chosen, except for the case that there are less than 5 predicted masks above this threshold value. If this is the case, the first 5 masks is chosen instead, even if some of them have a lower score-value, to be able to represent the whole image. Otherwise, large sections of the predicted mask will be missing. This is a user-chosen value, of course, and can be changed. Finally, the lower right image is the predicted masks from the algorithm. In addition, the metric values for this particular image's segmentation is written above the predicted mask figure.

There are in total 6 metric values used to determine the quality of the prediction. As elaborated upon in section 6, they assess the quality of very different things, so it is hard to say which one is the "most telling" metric. However, the IoU-score is generally a good metric to evaluate the quality of the Mask-RCNN, as it looks at how well the predicted bounding box fits the ground truth. As illustrated in figure 7, the IoU value also fluctuates greatly across all 80 test images, but is on average above the 0.5 value (the green line), which is the threshold for a "good result". Overall, the mean metric results looks good, as shown in table 1.

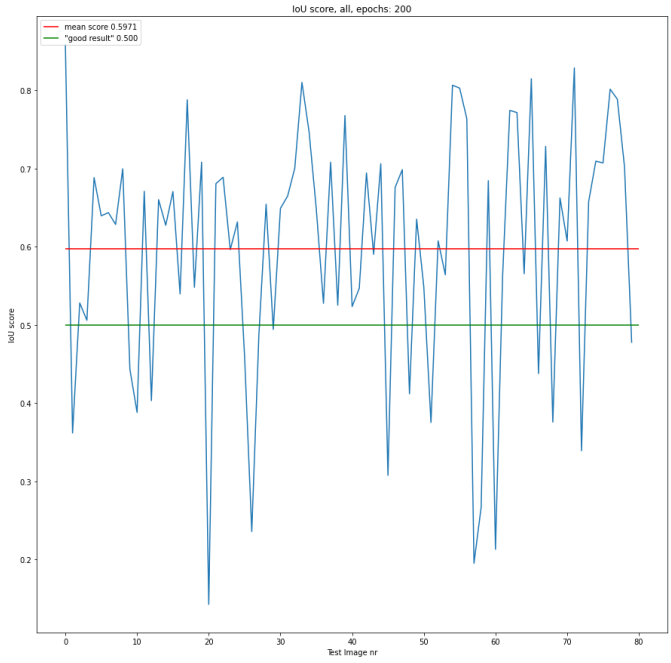

Fig. 7. All IoU scores

Mask R-CNN metrics	
Count-score	0.81
F1-score	0.79
F1-object score	0.75
IoU score	0.66
IoU-object score	0.68
Hausdorff-distance	153

Table 1. Mean Mask R-CNN metric results

8. DISCUSSION

8.1. Baseline comparison with U-Net

In order to study the performance of the Mask R-CNN it is compared with U-Net results on the same dataset. When comparing the results of the Mask R-CNN with U-Net, it is important to keep in mind that they are two different models, and accomplishes different things. The Mask R-CNN computes instance segmentation and bounding boxes of segmented cells, while U-Net computes pixel-wise classification. Hence, various evaluation metrics used for the Mask R-CNN are incompatible with U-Net, such as the count score and the IoU score, since there are no bounding boxes. But in terms of being able to segment objects of interest, the metrics such as the Hausdorff Object Score and the F1-score are still feasible to use, as elaborated upon in section 7.
The results are the mean values across all images, from running U-Net for 15 epochs and the Mask R-CNN for 200 epochs.

Algorithm	F1 Score	Hausdorff Distance
U-Net	0.49	79
Mask R-CNN	0.79	153

Table 2. Mean evaluation results

As seen in the results of table 2, U-Net outperforms the Mask R-CNN in terms of the Hausdorff distance, while the Mask R-CNN vastly outperforms U-Net in terms of the F1-score. To better interpret the implications of this, the image shown in figure 8 has been segmented by both models. The predicted segmentations output by each model can be seen illustrated in figure 9

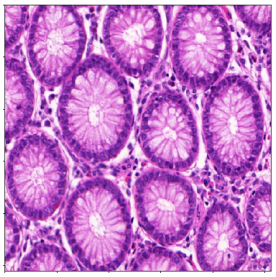

Fig. 8. GLAND image No. 55

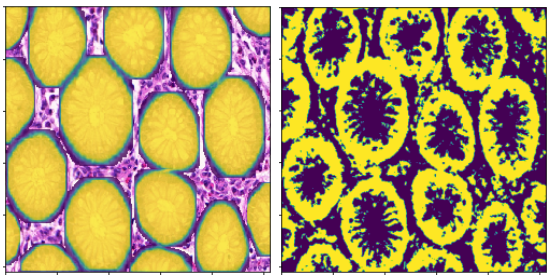

Fig. 9. Mask R-CNN (left) and U-Net (right) predictions
Mask R-CNN: F1: 0.893, Hausdorff: 173
U-Net: F1: 0.54, Hausdorff: 148

From aforementioned results, one can see that the U-Net manages to capture the intricate details within each cell, as each pixel is treated separately, while the Mask R-CNN segments whole objects. So which model is better? It depends on the objective and desires of interest. Both models are great in its own way, and so it is up to the researcher to chose which model is the most relevant.

8.2. Future work

While the implementation of the Mask R-CNN works very well, there are still large room for improvement, as is almost always the case in Deep Learning. These improvements range from choosing a different pre-trained CNN for the backbone of the model, fine-tuning and changing the hyperparameters of the training loop, or looking into using a different optimizer. The AdamW optimizer had already been looked into, as mentioned previously in section 4, but there are also many other potential optimizers, such as Adam, RMSprop and AdaGrad, for example. It should be noted, however, that all aforementioned optimizers are adaptive optimizers. In the Data Science Community some data scientists have begun to argue that these generalize worse (often significantly worse) than SGD, even though they may have a better training performance. See the following paper for reference [5].

As for the backbone, there are many pre-trained CNN's to pick from. ResNet-50 is the one that was used in this project, but ResNet-101 or even ResNet-151 could also be used, and may lead to a higher accuracy. Apart from these, the VGG ConvNet, MobileNet and ResNext have also seen widespread use in Mask R-CNN applications.

In terms of the hyperparameters, one could choose different values for the learning rate, batch size, the weight decay, the learning rate decay rate, etc. These values have been 'played around with' extensively already for this project, but more thorough research into the parameters, based on a deeper theoretical understanding of the implications of raising or lowering the parameter values, may yield a better final result.

Finally, a better data augmentation strategy, by including data augmentations such as random cropping, color shifting, random rotations, shearing and warping etc., should reduce overfitting and provide a larger and better dataset, thus providing a higher accuracy for the model.

9. CONCLUSION

Throughout this report, the Mask R-CNN and U-Net has been explained, and the results and output of both models discussed and compared. Mask R-CNN seems to be the state-of-the-art algorithm for segmenting objects, such as cells, in the current industry, while U-Net seems to be one of the best available models out there, for the sake of localization, i.e. denoting each pixel to a separate corresponding class. Compared to the Mask R-CNN, U-Net is a fairly simple network but it can, without any pre-trained weights and a limited number of epochs, give a fairly accurate prediction of the pixels belonging to cells in the images.
The Mask R-CNN really shows the results of the collective effort and the sharing of ideas (and code). The pre-trained weights makes the model faster and more precise and PyTorch makes the network available without much pre-coding needed. The output from the Mask R-CNN are more accessible to work with which makes the visualization and explanation task easier when presenting the results to non-coders.
Both networks have their advantages and dis-advantages and it would not make sense to declare any of them as the all time best solution.

10. GITHUB REPOSITORY

The Github Repository, containing the poster, final report, and all project code used to carry out this project, can be found in the link below:

`https://github.com/JonathanGundorph/DeepLearningProject_Mask_R-CNN_U_Net`

11. REFERENCES

[1] Vincent Fung, "An overview of resnet and its variants," 2017.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[3] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[4] Dep. of computer science University of Warwick, "Evaluation," 2019, https://warwick.ac.uk/fac/sci/dcs/research/tia/glascontest/evaluation.

[5] Ashia C. Wilson, Rebecca Roelofs, Mitchell Stern, Nathan Srebro, and Benjamin Recht, "The marginal value of adaptive gradient methods in machine learning," 2018.