PIXEL-WISE SEGMENTATION OF MICROSCOPY IMAGES WITH DEEP LEARNING

Filip Sawicki, Karl Ulbæk* and Teun Huijben**

*Danish Technical University, Lyngby, Denmark

ABSTRACT

Current biomedical and industrial research depends heavily on doing optical microscopy, for diagnosing diseases, understanding fundamental cellular processes, and drug discovery. Image segmentation is vital to extract quantified measurements from microscopy images and is often heavily dependent on the user. To make the process of image segmentation faster, reliable, general, and more automated, we will implement and validate multiple deep learning approaches.

Index Terms— Deep Learning, tissue images, segmentation, U-net, Mask R-CNN

1. INTRODUCTION

In this project, we will develop a tissue segmentation algorithm based on the principles of deep learning. The advantage of using deep learning approaches over standard threshold segmentation methods is that the nature of the images can be very different between samples and experimental conditions. Deep learning, if trained correctly, will provide a general classification method that can successfully classify the heterogeneous set of tissue images.

We will investigate two different approaches to perform the segmentation task, a convolutional neural network (CNN) and a mask region based convolutional neural network (Mask R-CNN). For the first approach, we have implemented the popular U-net architecture [2], consisting of the characteristic encoder-decoder structure. U-net has proven itself capable for a myriad of segmentation tasks in a wealth of contests. For the second approach, we have implemented the Mask R-CNN architecture, which is an improvement to a CNN by first predicting the regions of interest and subsequent segmentation of each region.
We will start by explaining the data that is used for training the different networks and the metrics we calculated to compare the segmentations of the different methods. Subsequently, we will illustrate the two approaches in more detail, describe the details and discuss the results.

2. DATA

The data used for training and validating the neural networks is taken from the Gland Segmentation contest in 2015 [1]. It contains 165 images, divided into three sets: a training set of 85 images and two test sets of 60 and 20 images, respectively. All images are converted to square images by zero-padding and resized to 512x512. The images are histological tissue samples of glandular tissue, stained with hematoxylin and eosin, where hematoxylin stains the cell nuclei blue and eosin the extracellular matrix pink. For all images, the glandular tissue is manually segmented (an example image, together with the segmentation, is shown in

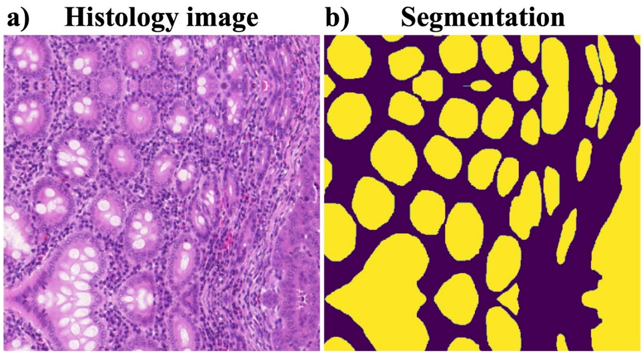


Figure 1. Example Image with segmentation. *a) Example image from the dataset showing the microscopy image of a thin slice of stained glandular tissue, accompanied with the manual segmentation of the glands (b). In the segmentation, yellow represents the foreground (gland) and purple represents the background (non-gland).*

Normally, deep learning approaches require large datasets to train the network. However, the dataset used in this project only contains 165 images, which is not sufficient to overcome overtraining. To increase the number of images, we applied multiple augmentation techniques (Figure 2). Each image in the dataset undergoes a multitude of augmentation techniques, where each one is chosen with a certain probability:

- cropping to 75% of original size (p=0.50)
- flipping horizontal (p=0.50)
- flipping vertical (p=0.50)
- rotation, with a random angle between -90 and 90 degrees (p=0.50)
- elastic deformation, with a random distortion field strength between 5 and 10, and a standard deviation of 3 (p=1)
- adding Gaussian noise with zero mean and standard deviation of 0.01-0.03 (p=0.20)
- blurring by taking the median of neighbouring pixels and sharpening by alpha-blending the result with the original image (p=0.20)

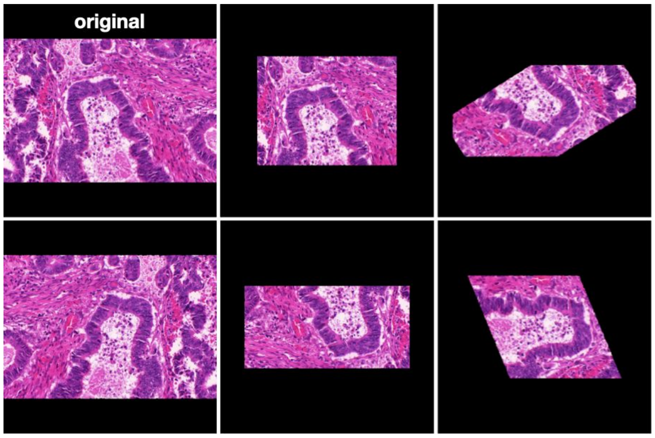
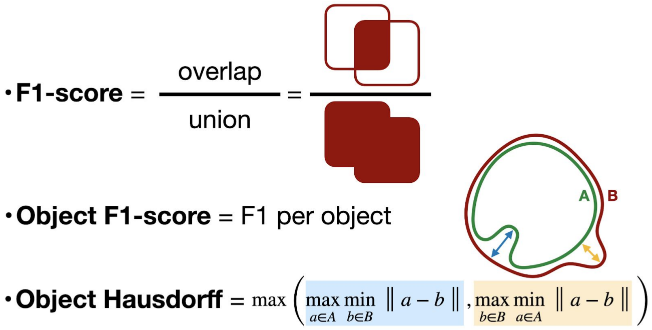


Figure 2. Image augmentation. *The original images (top-left) together with 5 random augmentations of the original image.*

3. METRICS

To compare results of the segmentation approaches, we will calculate and compare three different metrics (Figure 3). The first metric is the F1-score, which calculates the ratio between the overlap and the total area between the predicted segmentation and the ground truth segmentation. When the segmentation is correctly predicted, the overlap will be equal to the union, which results in an F1-score close to one, where a completely wrong prediction results in a score of zero. The second metric is the object-F1, which is simply the F1 score but calculated per object. The third metric is the object Hausdorff distance and is defined as the longest distance of all the points on one contour to the closest point on the other contour. In the visual representation in Figure 3, the blue arrow indicates the longest distance between a position on the green contour and the closest position on the red contour.



- **F1-score** = $\frac{\text{overlap}}{\text{union}}$ =
- **Object F1-score** = F1 per object
- **Object Hausdorff** = $\max\left(\max_{a\in A}\min_{b\in B}\| a-b \|, \max_{b\in B}\min_{a\in A}\| a-b \|\right)$

Figure 3. ***Metrics****. The three metrics used to evaluate and compare the different segmentation approaches are the F1-score, the object F1-score and the object Hausdorff distance.*

4. CONVOLUTIONAL NEURAL-NETWORK

4.1. Unet

Introduction

The first of 2 networks that will be implemented and applied to the segmentation problem is U-net [2]. A state of the art model from 2015, which back then beat its competition with a remarkable margin and especially excelled at problems with little available training data.

General architecture and intuition.

U-net is a convolutional neural network with a total of 23 convolutional layers (Figure 4). U-net consists of a contraction, expansion (encoder decoder) structure. Doing the contraction step, the image is spatially contracted by gradually reducing its height and width while increasing the number of channels, following the classic intuition of reducing the "where" and increasing the "what". The goal is not to classify but to create a high-resolution segmentation map, therefore the image has to be upscaled back to its original dimension. This is done in the expansion step, where transposed convolutions are used to upscale the image resolution. In order to maintain spatial information (doing the expansion), high resolution feature maps from the contraction path are concatenated onto the corresponding steps in the upscaling path. This concatenation step is often referred to as skip connections.

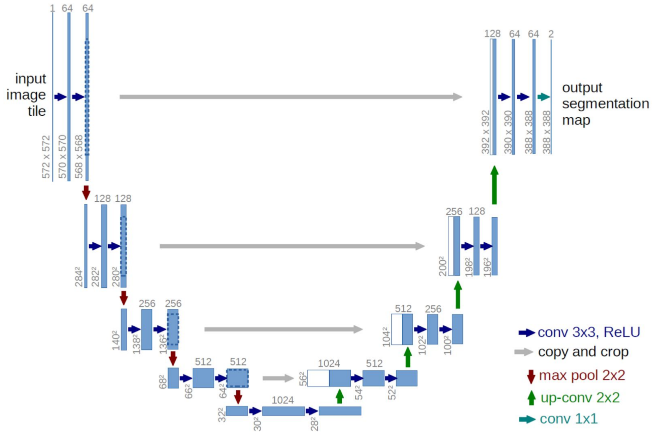


Figure 4. U-net architecture. *The full size image of the U-net architecture can be found in Figure A1 of the appendix.*

Contraction

Doing the contraction step the image is propagated in the following repeating manner: First, the number of channels is doubled by a 3 by 3 convolutional layer, followed by a ReLu activation function. Subsequently, a second convolutional layer is applied (this time without changing the number of channels), followed by another ReLu activation and a 2 by 2 max-pooling operation. It should be noted that the very first convolutional layer of the network does not merely double the number of channels but takes it from 1 channel to 64 channels.

This cycle is repeated 5 times, with the slight alteration that the final cycle does not contain the max-pooling layer. After the 5 cycles, the expansion step begins.

Expansion

A similar repeating pattern is found doing the expansion path. Firstly, a 2 by 2 transposed convolution is applied, increasing the resolution 2-fold while cutting the number of channels in half. Then, the image is concatenated by its corresponding feature map from the contracting path (the long horizontal gray arrows in Figure 4). Afterwards, two 3 by 3 convolutional layers each followed by applying a ReLu activation function. It should be noted that the first of the two aforementioned convolutional layers reduce the dimensionality by a factor of 2, while the second does not alter the dimensions.

This expansion cycle is repeated a total of 4 times, after which one final convolutional layer is applied taking the number of channels from 64 to 2, while retaining the resolution.

Implementations and differences

Our implementation of U-net was done from the ground-up in python using PyTorch, following the original paper [2], with a few minor changes. We used RGB images rather than grayscale images, meaning that the input image had 3 channels instead of one. The implemented channel dimensionality throughout the rest of the network was identical to that of the original paper, with the second exception that the output image was changed to having 1 channel rather than 2. Furthermore, zero padding was employed in this project in order to maintain the same input and output resolution. Lastly, in the original implementation of U-net, the authors used a pixelwise cross-entropy loss as well as a precalculated cell-edge weight-map. This weight-map penalizes the loss more significantly along the edges of two adjacent cells, thus enforcing the model to better distinguish two neighboring cells. This cell-edge weight-map was not employed in our implementation of U-net.

Hyper parameters and training

The network was trained using the binary cross-entropy loss, thus only distinguishing between foreground and background i.e. tissue and not tissue. The training was carried out in batches of 10 images using the Adam optimizer with a learning rate of 0.001. A 90/10% train-validation split was carried out in order to estimate the optimal number of epochs before the model started to overfit. From the loss graph (Figure 5a) can be concluded that the optimal number of epochs before the model started to overfit was around 24. Therefore, a final model was trained with 100% of the training data for 24 epochs.

4.2. Results

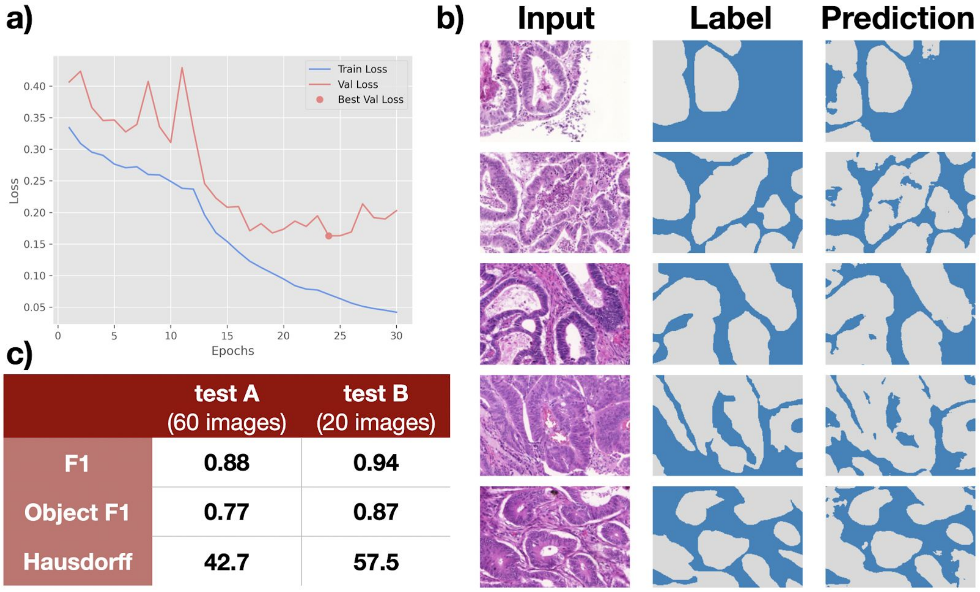
We test the trained network by using it to infer the segmentation on 80 unseen test images, divided over two test sets (Set A of 60 images and set B of 20 images). Visual comparison shows a high similarity between the predicted and ground-truth segmentation (Figure 5b), which indicates that the trained U-net network is capable of correctly segmenting unseen images. To further quantify the performance of the network, three metrics are calculated (Figure 5c).

5. MASK R-CNN

5.1. Concept

Mask R-CNN is a region-based convolutional neural network architecture (R-CNN) which specializes in semantic segmentation tasks. It extends Faster R-CNN, a state-of-the-art architecture for instance segmentation, by including a third parallel branch for predicting segmentation masks on each region of interest (Figure 6). The training process of Mask R-CNN is specified as a multi-task loss on each branch per sampled region of interest.

Faster R-CNN is a multi-stage R-CNN consisting of a convolutional "backbone" network used for feature extraction from an input image, an attention mechanism called region proposal network (RPN), which generates candidate bounding boxes and two parallel networks for bounding box offset regression and classification.



	test A (60 images)	test B (20 images)
F1	0.88	0.94
Object F1	0.77	0.87
Hausdorff	42.7	57.5

Figure 5. Results of U-net implementation. a) *Training and validation loss for 30 training epochs for 90-10% training-validation split.* ***b)*** *Five example test images together with the ground-truth and predicted segmentation.* ***c)*** *Quantified metrics of the trained model on the two different test sets, containing 60 and 20 images, respectively.*

"RoIPool" layer combines features from the backbone network with its corresponding regions of interest, which are then used as feature maps for box regression and classification. It divides ROIs into sections of the same dimensions, finds maximum valued pixels in each section and forwards them to the next layer. "RoIAlign" on the other hand uses bilinear interpolation to avoid quantization of region of interest boundaries leading to great improvement in semantic segmentation. It fixes an issue of pixel-to-pixel misalignment in Mask R-CNN by replacement of RoIPool with RoIAlign layer in the feature extraction part.

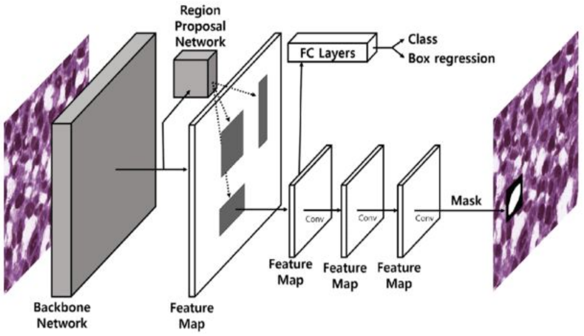


Figure 6. Mask R-CNN architecture *with all stages and data flow indicated by arrows.*

5.2. Implementation and Training

For the implementation of Mask R-CNN we have used PyTorch version 1.7.0+cu101 and trained it on GPUs provided by Google Colab with the same hardware and software configuration as for UNet.

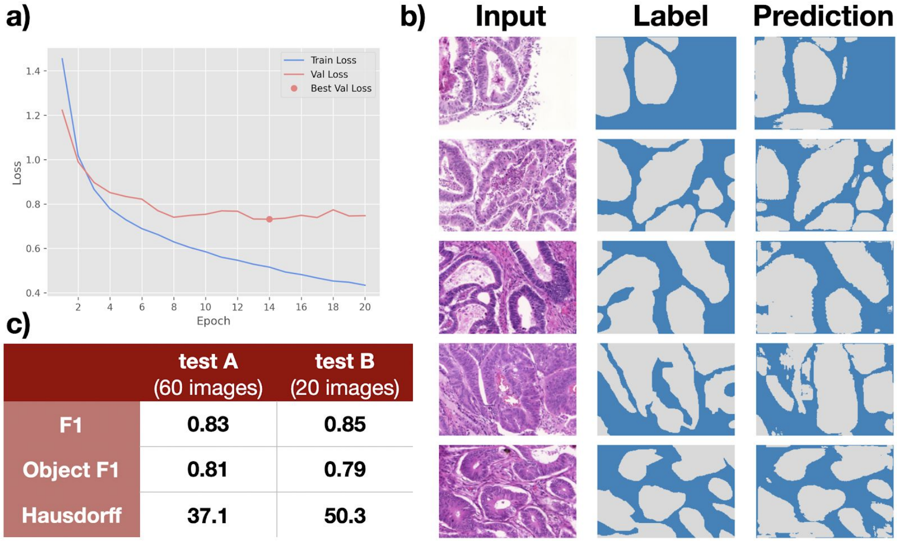
Mask R-CNN consists of a resnet50 backbone, pretrained on a COCO dataset, for feature extraction and an untrained RPN and mask predictor with 256 hidden layers.

Training is performed using the SGD optimizer with a momentum of 0.9 and weight decay of 0.0003. Additionally we have set a step-wise learning rate with gamma of 0.1 starting at 0.0001 and decreasing every 15 epochs. The loss function used by optimizer is the sum of mask, bounding box and classifier losses.

Due to limited VRAM resources on the free version of Google Collab we had to limit the batch size to two images.

Finally, to receive valid output masks we have implemented a custom function which combines all masks produced by our network. It removes small and overlapping bounding boxes with less than 0.4 probability of having a cell inside using a non-maximum suppression algorithm, then all pixels with probability less than 0.3 are set to 0 and corresponding pixels in selected masks are summed and thresholded at 0.9.

A 90/10% train-validation split was carried out in order to estimate the optimal number of epochs before the model started to overfit. From the loss graph (Figure 6a) can be concluded that the optimal number of epochs before the model started to overfit was around 14.



	test A (60 images)	test B (20 images)
F1	0.83	0.85
Object F1	0.81	0.79
Hausdorff	37.1	50.3

Figure 7. Results of the Mask R-CNN implementation. a) *Training and validation loss for 20 training epochs for 90-10% training- validation split.* ***b)*** *Five example test images together with the ground-truth and predicted segmentation.* ***c)*** *Quantified metrics of the trained model on the two different test sets, containing 60 and 20 images, respectively.*

5.3. Results

We hypothesised that MaskRCNN would perform better than U-Net. However based on the calculated metrics, we find that the performances are very comparable with each other (Figure 7c).

Visual inspection of predictions showed us that Mask R-CNN struggles with correct segmentation of large tissue patches, especially those with ring-like shapes or any cells having holes inside (Figure 7b).. However, Mask R-CNN visually outperforms U-Net for many small and similarly sized/shaped tissue patches.

5. CONCLUSION AND DISCUSSION

We can conclude that both our U-net and Mask R-CNN implementations are able to correctly segment the glandular tissue structures in histological microscopy images. We hypothesized that Mask R-CNN would outperform U-net, however, U-net has a slightly better performance compared to Mask R-CNN. This can probably be improved by a more elaborate choice of hyperparameters. Next to that, we concluded from visual inspection of the predicted segmentations, that U-net performs better in the case of large tissue patches, where Mask R-CNN performs better for many equal-sizes smaller tissue patches.

Further improvements that will result in an increase in segmentation performance could be; test- time augmentations [4], where the predicted segmentation is determined by averaging the results of multiple augmented test images. Other approaches could be to make an ensemble method of U-net and Mask R-CNN [5], better controlling the anchor point locations of RPN, or using a modified loss function that penalizes overlapping borders [2].

6. CODE AND DATA AVAILABILITY

The used dataset and Python code are publicly available on GitHub:
https://github.com/fpsawicki/Pixel-wise-Segmentation-of-Microscopy-Images

7. REFERENCES

[1] K. Sirinukunwattana, et al. "A Stochastic Polygons Model for Glandular Structures in Colon Histology Images," in IEEE Transactions on Medical Imaging, 2015

[2] Ronneberger, Olaf, *et al*. "U-net: Convolutional networks for biomedical image segmentation." *International Conference on Medical image computing and computer-assisted intervention*. Springer, Cham, 2015.

[3] He, Kaiming, *et al*. "Mask r-cnn." *Proceedings of the IEEE international conference on computer vision*. 2017.

[4] Moshkov, N., *et al.* Test-time augmentation for deep learning-based cell segmentation on microscopy images. *Scientific reports*. 2020.

[5] Vuola, A. O., et al. Mask-RCNN and U-net ensembled for nuclei segmentation. *IEEE 16th International Symposium on Biomedical Imaging*. 2019

Appendix

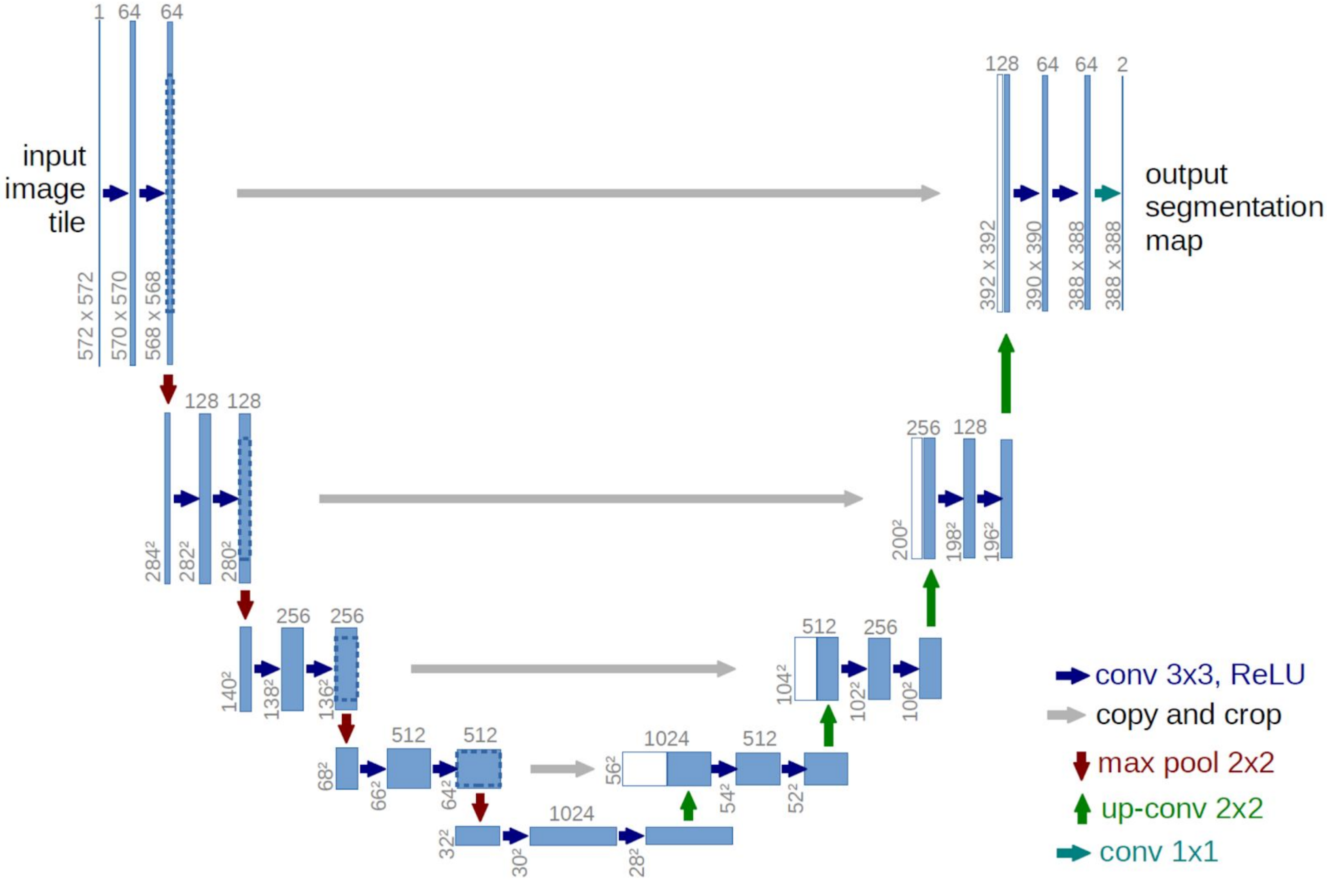


Figure A1. *Enlarged image showing the U-net architecture*