Bioinformatics project: segmentation on human renal tissue

Emanuele Fasce
Politecnico di Torino
S277983
emanuele.fasce@studenti.polito.it

Abstract

In this project several neural architectures for segmentation are implemented and evaluated on a dataset composed of 200 images of cancerous cells depicting two types of cancer, Clear cell renal cell carcinoma and Papillary renal cell carcinoma. First, the U-Net network is implemented, fine-tuned on the dataset and used as a baseline. Secondly, Linknet, U-Net++ and MANet are evaluated. The goal of this project is to find out which network performs best on this task and to evaluate the effect of different transformations and hyperparameters. Lastly, the best performing model is then used to build another dataset composed of its predictions: a classification model that uses the predictions as input images predicts the type of cancer surprisingly well.

1. Introduction

A segmentation task is a pixel-wise classification task. Most segmentation models are based on supervised learning approaches that require labeled images which are obtained through manual segmentation, a very time-consuming task. Another challenge to consider with the automated segmentation of medical images is the large variations in shape, size, texture and colour between patients and poor contrast between regions. Noise or lack of consistency in source data acquisition may also result in wide variations in the source image data which is often the case in real applications. In the past human feature engineering was extensively used, even though it is time-consuming and fails to handle natural data in their raw form. The most recent state of the art models in computer vision, instead, all rely on deep learning approaches, which are capable of processing natural data in their raw form. They usually take advantage of the Convolutional Neural Network (CNN). These deep model approaches have been successfully used also in biomedical image segmentation. Among the CNNs architectures proposed for image segmentation, a very important branch is composed of FCN derived networks, like the ones used in this project. They are only made of locally connected layers, such as convolution, pooling and upsampling, without fully connected layers. These networks consist of a downsampling path made of pooling and convolutions, used to extract and interpret the context, and an up-sampling path which allows for localization. FCNs also employ skip connections to recover the fine-grained spatial information lost in the downsampling path. All the networks used for segmentation come from the FCN model.

2. Dataset

The dataset is composed of 200 images of size of human renal tissue depicting two different classes of tumour: clear cell renal cell carcinoma and papillary renal cell carcinoma. 100 images show the renal cell carcinoma and images show the papillary cell carcinoma. Original images with their segmentation ground truth are provided. This is an unbalanced dataset since only 20% of the pixels are labeled as positive, requiring suitable loss functions. The original dataset is split into training (160 images), validation (40 images) and test set (10 images). The model that performs best on the segmentation task makes predictions for the images of the whole dataset, creating a new dataset (for the classification task) whose images contain only two values. Therefore, the prediction of an image belonging to training set in the first dataset belongs to the training set in the second dataset.

3. Preprocessing

As usual when dealing with computer vision problems, images are resized and normalized before being processed by the network. In this case images are resized to 224x224 and normalized with two different set of values: both the mean and variance values of Imagenet and the values of this dataset are used in the experiments. When the used type of network allows it, in a few experiments images are resized to 448x448. For what concerns the transformations, some of them are proven to be more effective in the biomedical field. However, since there is not a general consensus, three different types of transformations are used: affine, elastic and pixel-wise. The affine transformation (Figure 2) is composed of a random flip function and a random shift-scale-rotate function. The pixel wise transformation (Figure 4) consists in randomly applying gaussian noise and changing hue, saturation, brightness contrast, and gamma values. The elastic transformation is shown in Figure 3.

4. Metrics

The metric that has been chosen to compare the results in the segmentation task is the IOU coefficient.

$$IOU = \frac{A \cap B}{A \bigcup B}$$

Even though the dice coefficient is more used in segmentation tasks, I chose to use the dice coefficient as a loss function because of its differentiability, and to use the IOU as the main metric. For what concerns the classification task, the f1 score is chosen.

$$W\_BCE = -\frac{1}{N}\sum_{i=1}^{N}(y_i log(p(y_i)) + (1-y_i) log(1-p(y_i)))$$

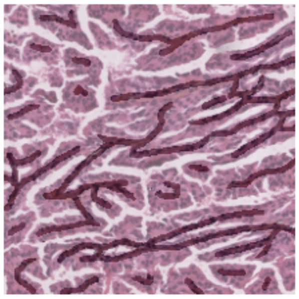


Figure 1. Original image

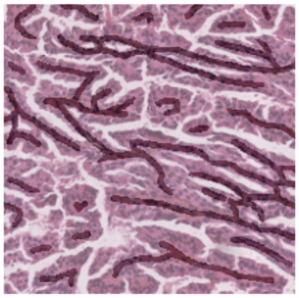


Figure 2. Affine transformation

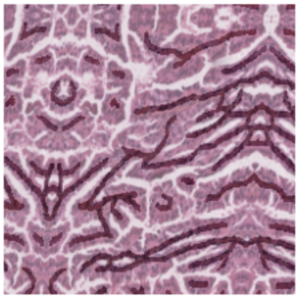


Figure 3. Elastic transformation

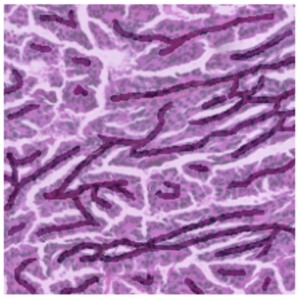


Figure 4. Pixel-wise transformation

5. Loss and activation functions

In the beginning of the experiments, the sigmoid function was used as activation function and a binary crossentropy loss as the loss function. However, this loss did not obtain satisfying results due to the class unbalance issue. Better results were obtained using a weighted binary crossentropy loss and the dice coefficient, which are used commonly in class unbalanced segmentation problems. Their formulas are provided below.

$$W\_BCE = -\frac{1}{N}\sum_{i=1}^{N}(y_i log(p(y_i)) + (1-y_i) log(1-p(y_i)))$$

$$Dice = -\frac{2|A \cap B|}{(|A| + |B|)}$$

6. Network architectures

A very short sum-up of the difference between the architectures is provided below.

6.1. U-Net

The U-Net architecture is different from the most basic FCN architecture in that the high resolution features coming from the contracting path are combined with the up-samples output through a concatenating operation: a successive convolutional layer learns to assemble a more precise output. The FCN instead upsamples only once and through a bilinear interpolation. Since the up-sampling path is as long as the down-sampling path, this network is called U-Net.

6.2. LinkNet

This architecture is very similar to the U-net architecture. The main difference comes from the usage of a simple addition in order to combine the contracting and up-sampling path.

6.3. U-Net++

What distinguishes U-Net++ from U-Net is the redesigned skip pathways that connect the two subnetworks, aiming at reducing the semantic gap between the feature maps of the encoder and decoder sub-networks, leaving an easier optimization task to the optimizer. In U-Net, the feature maps of the encoder are directly received in the decoder: in the U-Net++, they undergo a dense convolution block whose number of convolution layers depends on the pyramid level.

6.4. MANet

The authors of the Multi Attention Network paper identify the direct concatenation without refinement of the feature maps coming from the contracting path with the ones from the up-sampling path and the insufficient exploration of long-range dependencies as the main drawbacks of the U-net architecture. They introduce an efficient dot product attention mechanism suitable for large input images, which proves to be very effective when dealing with remote sensing tasks.

7. Experimental results for the segmentation task

7.1. Followed methodology

All the experiments are shown in table 1, which reports the size of the images after the resizing, the model type, the backbone, the loss function and the transformations used, as well as the loss value and the IOU reached. Since there could have been so many hyper-parameters to try, I decided to try a lot of them when using the U-Net model and then use that knowledge to find the best hyper-parameters on the others. This is the reason why 16 runs are performed with the U-Net and a fewer runs with the other models. It is also worthwhile to notice that some runs with the U-Net are not even included since the results were poor (40% of IOU on average), due to loss functions that did not account for the class imbalance. After having performed the experiments, one could argue that the chosen architectures give very similar results, probably because they are all derived from the U-Net model. However, some interesting patterns can be observed.

7.2. Insights

For what concerns the transformations, using them always benefits the training with respect of not using them. From the results, it seems that the affine and elastic transformations are more useful than the pixel-wise transformations. Indeed, the highest IOU values are reached when using together these two types of transformations. This is not surprising since the affine transformation involves flipping and rotating, which create very similar pictures. By observing the pixel-wise transformation (Figure 4), it can be observed that colors are very different from the original ones, which may have harmed the results. The parameters of the transformations could have been modified in order to get more realistic results, however this would have required some prior knowledge and extensive experimentation. The backbone that performs best is usually the Resnest network, a variance of Resnet that enables attention across featuremap. Also other backbones were considered but were not included in the experiments table since performing worse than the default backbone Resnet50. Deeper backbones work better than more shallow ones with this dataset, indeed the Resnest101 gives consistently better results with all the models. The loss functions which have been used in the experiments are standard in the imbalanced segmentation tasks: the weighted binary cross entropy loss and the dice loss. It can be observed that the weighted binary cross entropy loss leads to higher IOU more consistently. Images have always been resized to 224x224 apart from two experiments in which they have been resized to 448x448, leading to the best performing model when using the U-Net network. This is possible since the U-Net is a fully convolutional network without fully connected layers and can accept tensors with different size. Among all the trained models, the model that achieves the best IOU on the validation set (65.86) is tested on the test set, reaching a IOU of 65.10.

8. Visualizations

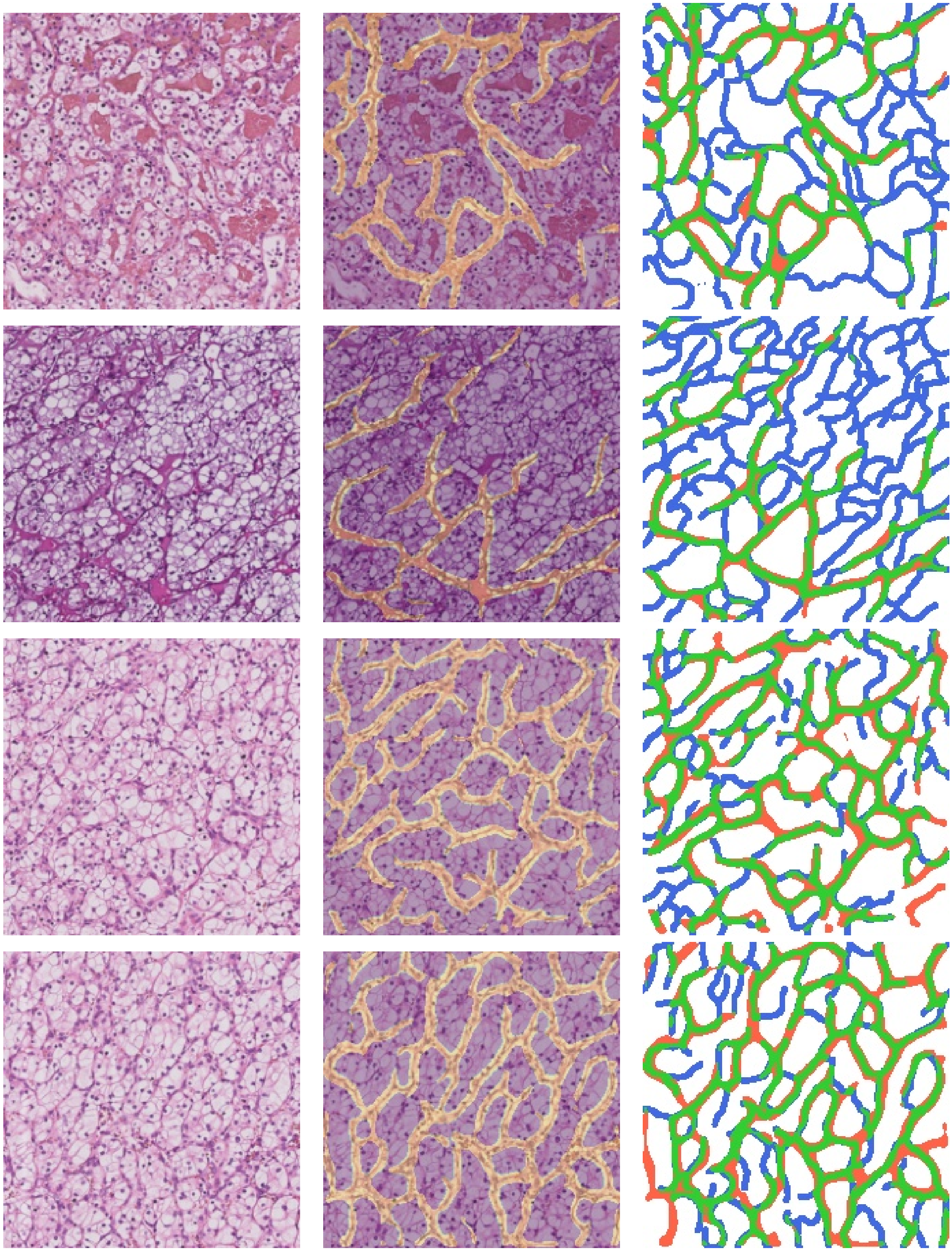
The predictions of the best performing model on the test set are visualized in Figure 5. Two visualization techniques are introduced. In the first one, the target image and the predictions are merged and displayed together on the center column. This could be an useful insight to a SME when analyzing where the model tends to fail more. In the second one shown in the right column the true negative, false negative, true positive and false positive are directly shown on the image using different colors, mimicking a contingency table. Green is used for true positive, white for true negative, blue for false negative and red for false positive.

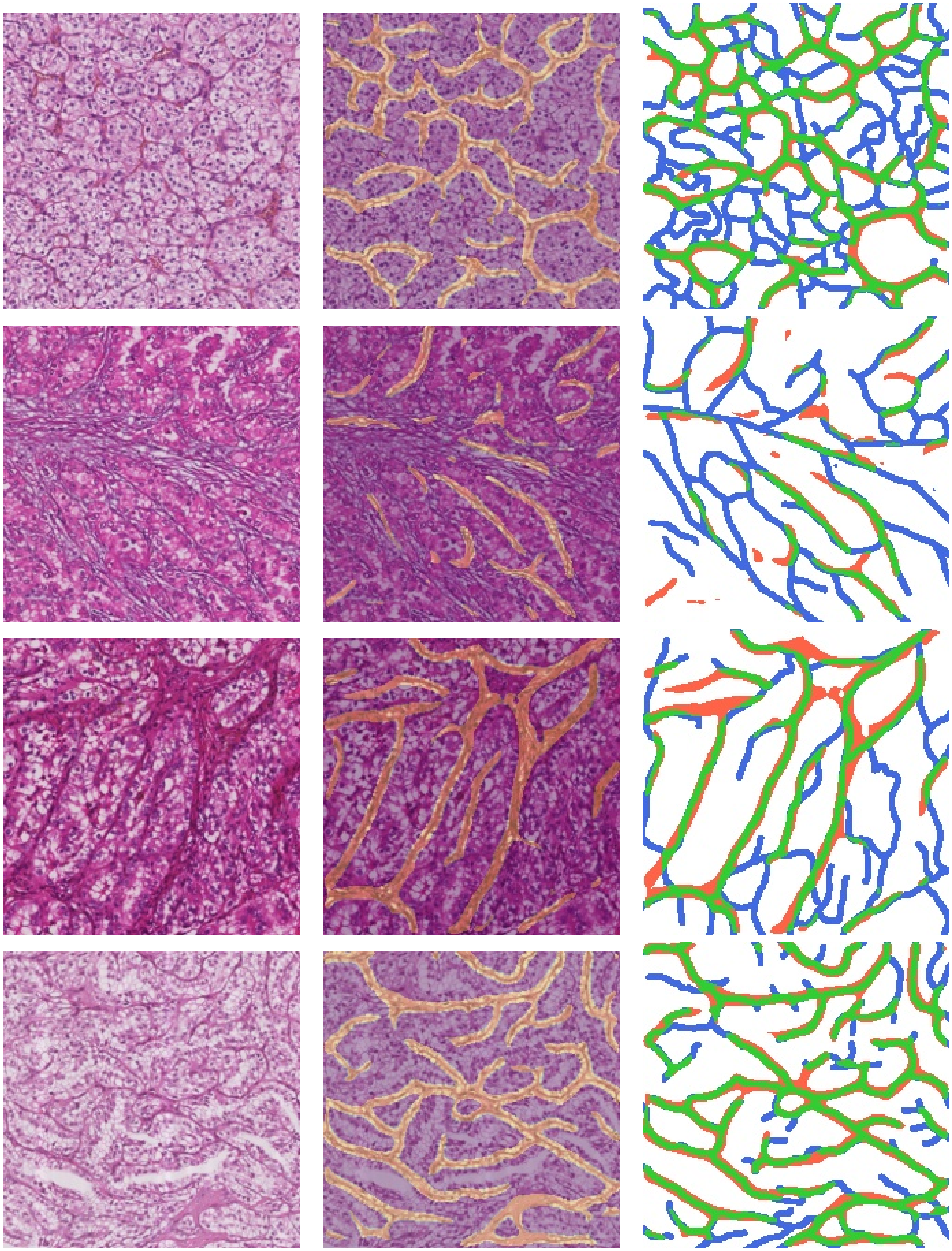
9. Experimental results for the classification task

The model that performs best on the segmentation task (U-Net++ with Binary BCE, Resnest101 and Affine+Elastic transformations) makes predictions for the images of the whole dataset, creating a new dataset whose tensors contain only two values. A simple classification model (Resnet50) which predicts the type of cancer is trained on this new dataset and only after 10 epochs is able to reach 100% accuracy on the training, validation and test set.

Size	Model	Loss function	Backbone	Transformations	Training loss	Validation loss	Validation IOU
224	U-Net	Dice	Resnet50	None	0.013	0.0233	60.73
224	U-Net	Dice	Resnet18	None	0.015	0.024	60.01
224	U-Net	Dice	Resnest26	None	0.012	0.023	61.61
224	U-Net	Dice	Resnest50	Affine	0.017	0.021	63.92
224	U-Net	Dice	Resnest50	Elastic	0.018	0.022	63.19
224	U-Net	Dice	Resnest50	Pixel-wise	0.015	0.022	63.01
224	U-Net	Dice	Resnest50	All	0.019	0.021	63.83
224	U-Net	Dice	Resnet101	All	0.029	0.034	62.22
224	U-Net	Dice	Resnest101	All	0.027	0.028	65.31
224	U-Net	Binary CE	Resnet50	None	0.162	0.192	62.29
224	U-Net	Binary CE	Resnet50	All	0.169	0.201	62.22
224	U-Net	Binary CE	Resnest101	All	0.154	0.191	62.67
224	U-Net	Binary CE	Resnest50	Affine+Elastic	0.180	0.173	64.62
224	U-Net	Binary CE	Resnest50	Affine+Elastic	0.180	0.173	64.62
224	U-Net	Binary CE	Resnest101	Affine+Elastic	0.177	0.172	65.59
448	U-Net	Binary CE	Resnest101	Affine+Elastic	0.147	0.151	65.86
224	MA-Net	Binary CE	Resnet50	Affine	0.166	0.200	61.33
224	MA-Net	Binary CE	Resnest50	All	0.189	0.177	64.38
224	MA-Net	Binary CE	Resnest101	All	0.185	0.173	65.06
224	MA-Net	Dice loss	Resnet18	No	0.020	0.034	60.26
224	MA-Net	Dice loss	Resnet50	No	0.026	0.031	61.98
224	MA-Net	Dice loss	Resnest50	No	0.026	0.031	62.39
224	MA-Net	Dice loss	Resnest50	All	0.027	0.031	61.88
224	MA-Net	Dice loss	Efficient-net	All	0.034	0.030	60.94
224	LinkNet	Dice loss	Resnet50	All	0.029	0.030	61.93
224	LinkNet	Dice loss	Resnet101	All	0.032	0.035	58.60
224	LinkNet	Dice loss	Resnest50	All	0.027	0.029	64.56
224	LinkNet	Binary BCE	Resnest50	All	0.028	0.030	64.02
224	LinkNet	Binary BCE	Resnest101	Affine+Elastic	0.176	0.172	65.20
448	Linknet	Binary BCE	Resnest101	Affine+Elastic	0.374	0.411	61.92
224	U-Net++	Dice loss	Resnet50	All	0.033	0.030	61.51
224	U-Net++	Dice loss	Resnest101	All	0.028	0.027	65.38
224	U-Net++	Binary BCE	Resnest101	Affine+Elastic	0.172	0.173	65.67

Table 1. Segmentation experiments





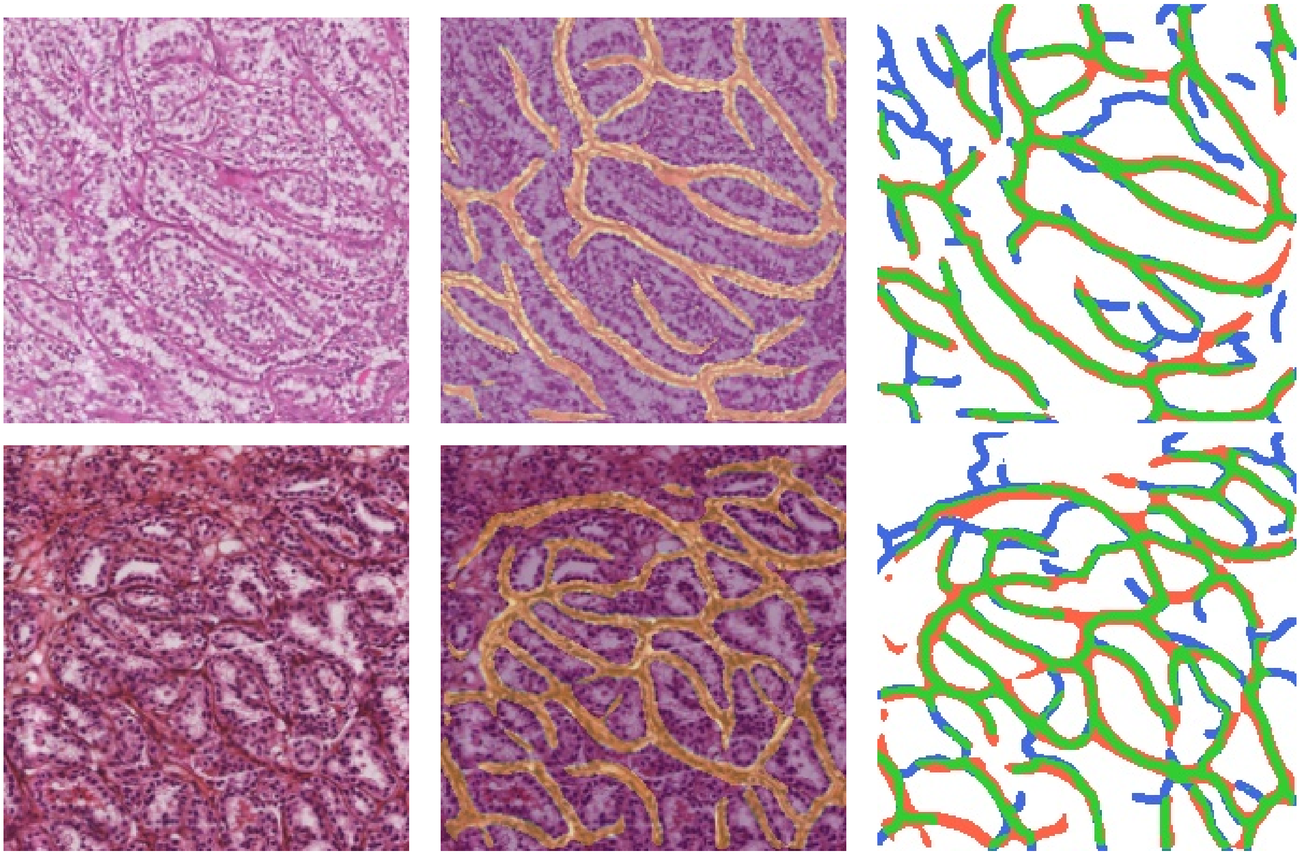


Figure 5. Images from the test set are shown in three different ways: original images (left), original images with predictions of the model (center), contingency table (right)