Comparative Analysis of Prostate MRI Image Segmentation: Watershed, U-Net, SegNet and SegFormer Approaches

Ivan Krstev
2071991

Joan Orellana Rios
2081188

Esteban Ortega Dominguez
2084248

Abstract

This study provides a comparative analysis of prostate MRI image semantic segmentation for distinguishing the central gland and peripheral zone using multiple models such as U-Net, SegNet, SegFormer and, experimentally, the Watershed algorithm. The results highlights SegFormer's high performance, particularly for minority classes, with a Dice coefficient of 0.82 for the central gland compared to U-Net (0.53) and SegNet (0.56).

Link to repository

1. Introduction

Semantic segmentation is a computer vision problem that has a wide range of applications. It is relevant to image classification since it generates per-pixel category predictions rather than image-level predictions. This per-pixel classification, often known as the 'mask,' provides precise information about the spatial distribution of objects within an image. Given the close relationship between classification and semantic segmentation, many commonly used techniques are variations on classification models.

Prostate MRI image segmentation is important in modern medical diagnostics because it allows for exact identification and analysis of prostate structures. The prostate is divided into two separate regions: the central gland and the peripheral zone, each of which plays a specific role in prostate function. Accurate segmentation of these structures is required for a variety of therapeutic applications, including treatment planning and illness detection.

As medical imaging technology grows, the choice of segmentation models becomes more important. In this study, we compare four different approaches—Watershed, U-Net, SegNet, and SegFormer. Watershed is recognized for its simple and intuitive boundary demarcation. In contrast, deep learning models such as U-Net, SegNet, and SegFormer have grown in popularity, demonstrating outstanding accuracy in image segmentation tasks.

The primary goal of this project is to evaluate and compare the performance of the Watershed, U-Net, SegNet, and SegFormer models in the context of prostate MRI image segmentation. By evaluating their effectiveness in determining the central gland and peripheral zone of the prostate, we want to give useful insights to the field, assisting in the selection of ideal segmentation methodologies for enhanced medical diagnostics.

2. Related Work

The evolution of prostate segmentation extends beyond the advent of deep learning and practical multi-image processing. A notable precursor is the work presented in [4], which explores the application of the Watershed algorithm for segmenting multiple organs in prostate scans. This early study provides a historical perspective on the quest for accurate segmentation techniques before the widespread adoption of deep learning methods.

In [2] a prostate segmentation task for adaptive radiotherapy using three deep-learning approaches: UNet, ENet, and ERFNet is being addressed. Their study compared the accuracy and efficiency of these models, with ENet emerging as a fast and accurate choice, especially in scenarios where GPU availability is limited. The findings suggest the potential application of ENet for prostate delineation, even in situations with small training datasets, showcasing its suitability for personalized patient management.

In [6] it is proved that in recent years, artificial intelligence (AI) approaches for auto-contouring in treatment planning have shown promising results in improving efficiency and consistency. They proposed an auto-contouring strategy utilizing a U-Net/VAE model with an outlier management (OM) technique. The OM strategy, designed to handle interobserver uncertainty, demonstrated enhanced accuracy by mitigating the impact of outliers in the training dataset. This approach leverages transfer learning and probabilistic hierarchical U-Net with VAE, showcasing improved segmentation results for prostate contouring.

An extensive analysis of segmentation models for prostate gland segmentation is being conducted in [7], introducing an object detection step using a YOLO-v4 model for cropping MRI volumes around the prostate. The study emphasizes the effectiveness of nnU-Net in full volume segmentation tasks, outperforming other models, with additional insights on the impact of model selection on segmentation performance. The introduction of an object detector for image cropping shows promise for further exploration in leveraging learned local representations to enhance segmentation tasks.

In the rest of this paper, we will try to follow the time frame of the image segmentation advancements, starting from a model as simple as watershed, up to a transformer architecture for segmentation.

3. Dataset

The dataset for our study, obtained from The Cancer Imaging Archive (TCIA) through a collaborative challenge by the National Cancer Institute's (NCI's) Cancer Imaging Program and the International Society for Biomedical Imaging (ISBI), includes 3D MRIs from 60 patients.

The data collection process involved the use of two MRI scanners, a 1.5T (Philips Achieva) and a 3T (Siemens TIM). Doctors from RUNMC University in the Netherlands marked individual slice central gland (CG) and peripheral zone (PZ) outlines for each case.

In the dataset, these MRIs are classified into three subsets: a training set consisting of 43 cases, a validation set with 5 cases, and a test set containing 12 cases. Each case includes approximately 20 cross-sections for 1.5T MRIs and about 30 cross-sections for 3T MRIs. Each MRI is segmented into three distinct classes: the background, the central gland, and the peripheral zone of the prostate. An example of this segmentation can be seen in Figure 1. The white part corresponds to the central gland and the gray part corresponds to the peripheral zone of the prostate.

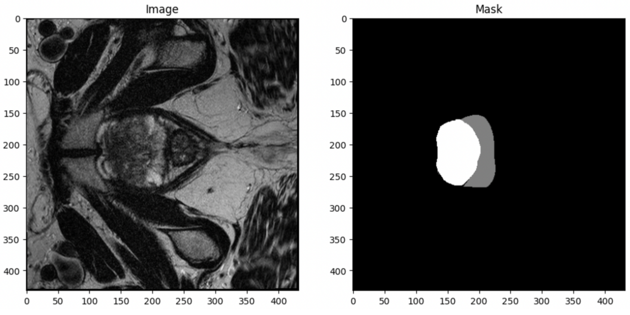


Figure 1: Example of MRI scan cross-section with corresponding segmentation

To simplify data retrieval, we used the NBIA Data Retriever software, which is used to download medical imagery. Subsequently, image loading into our code was performed using the MedPy library. Given the different sizes of images produced by 1.5T and 3T scanners, rescaling was necessary. While rescaling for images was straightforward (we opted to rescale images so they matched the size of the first mask), segmentation masks required the use of interpolation to maintain accuracy. We chose the ”nearest” interpolation mode. As a preprocessing step, we flatten the images across the patient/case dimension, aiming to ensure equal treatment for every image, regardless of the patient to whom it belongs. Additionally, we applied one-hot encoding to the masks for future processing.

4. Segmentation models

4.1. Watershed

The watershed algorithm is inspired by the natural flow of water over a terrain and is particularly effective in delineating object boundaries in complex and textured images. In this method, the grayscale intensity of the image is treated as a topographic surface, where regions of interest are considered as basins. By iteratively flooding the basins from markers (user-defined/automatic seeds) and allowing the water to reach equilibrium, the algorithm identifies boundaries where the flooding fronts meet. This process results in segmentation maps that can accurately outline objects and regions within an image, making watershed segmentation a valuable tool in various applications such as medical image analysis, object recognition, and scene understanding, especially speaking about the period prior to the advent of deep neural networks. The biggest bottleneck in the approach is the sensitivity to even the smallest noise in the data, as noted in [9].

4.2. U-Net

U-Net [8] is a convolutional neural network architecture designed for semantic segmentation tasks, featuring a contracting path for encoding contextual information and an expansive path for precise localization.

Figure 2 shows the architecture of the network. It is made up of an expanding path on the right side and a shrinking path on the left. The convolutional network's standard architecture is followed by the contracting path. Two 3x3 convolutions (unpadded convolutions) are applied repeatedly, each of which is followed by a rectified linear unit (ReLU) and a 2x2 max pooling operation with stride 2 for downsampling. Every downsampling step results in twice as many feature channels. Each step in the expanding path is an upsampled feature map, followed by a concatenation with the correspondingly cropped feature map from the contracting path, a 2x2 convolution (also called a ”up-convolution”) that halves the number of feature channels, and two 3x3 convolutions, each of which is followed by a ReLU. Every convolution results in the loss of boundary pixels, which makes cropping inevitable. A 1x1 convolution is employed at the last layer to transfer every 64-component feature vector to the required number of classes.

There are 23 convolutional layers in the network overall.

The architecture we employ follows the original U-Net architecture proposed in [8] with a Cross-entropy objective, and RMSprop optimizer.

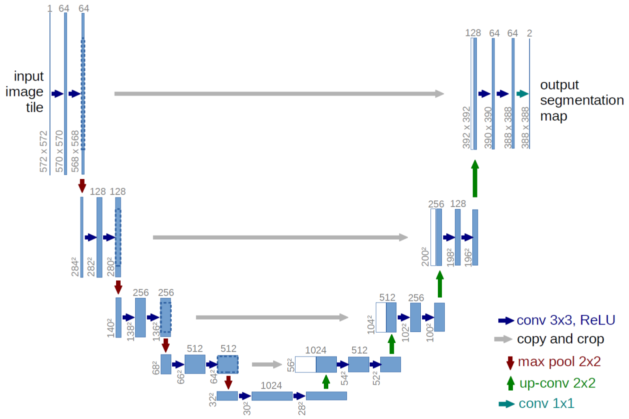


Figure 2: Architecture of the U-Net Model.

4.3. SegNet

SegNet is a deep fully convolutional neural network architecture designed for semantic pixel-wise segmentation. It consists of an encoder network, a corresponding decoder network, and a pixel-wise classification layer. The novelty of SegNet lies in the manner in which the decoder upsamples its lower resolution input feature maps using pooling indices computed in the max-pooling step of the corresponding encoder to perform non-linear upsampling. This eliminates the need for learning to upsample and allows for efficient memory and computation during inference. [1]

Segnet differs from U-Net; in Segnet only the pooling indices are transferred to the expansion path from the compression path, using less memory. Whereas in U-Net, entire feature maps are transferred from compression path to expansion path making, using a lot of memory. [5]

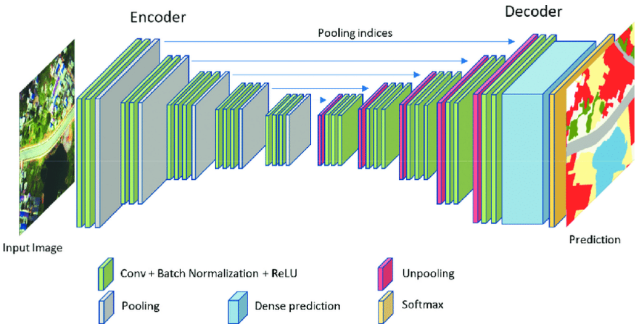


Figure 3: SegNet Architecture

Here's how the implementation we employed works:

Initialization: The network takes the MRIs as input and produces segmentation maps with a specified number of output channels, which corresponds to the number of classes. Batch Normalization momentum is a parameter controlling the trade-off between the current mini-batch statistics and the running statistics.

Encoding Layers: The encoding part consists of 5 stages. Each stage includes convolutional layers, batch normalization, and max pooling. Max pooling is used for down-sampling, and the pooling indices are stored for later use in the decoding layers. The number of filters in the convolutional layers increases with each stage.

Decoding Layers: The decoding part also consists of 5 stages, mirroring the encoding stages. Each stage includes upsampling (using max unpooling with stored indices), convolutional layers, and batch normalization. The goal of the decoding layers is to reconstruct the segmented output from the features obtained in the encoding layers. The number of filters in the convolutional layers decreases with each stage.

Because the cross-entropy loss we use already incorporates the softmax function on the output, we decided not to apply an additional softmax activation function to the output inside the SegNet architecture.

4.4. Visual transformers: SegFormer

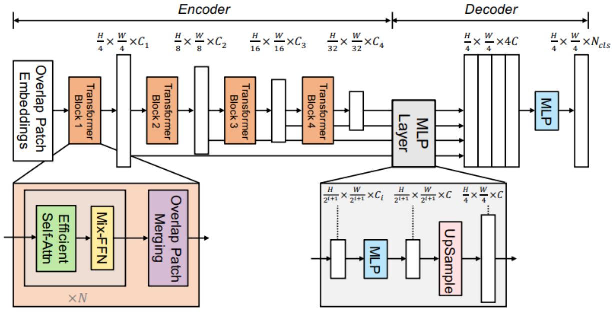
Given the great success in natural language processing (NLP) there has been an increase in interest to introduce the Transformers architecture to vision tasks.

Previous to Dosovitskiy work the attention mechanism was mainly applied either in conjunction with convolutions networks or used to replace certain CNNs while keeping their overall structure in place. Dosovitskiy showed that it's not necessary to rely on CNNs and a pure transformer applied directly to a sequences of images can perform very well on image classification tasks. The architecture of the Vision Transformer (ViT) proposed splits an image into multiple linearly embedded patches and feeds them into a standard Transformer with positional embedding as an input. Image patches are treated the same way as tokes in an NLP application. The model is trained on image classification in supervised fashion. The breakthrough in the direct application of the Transformer architecture came when the training was done on a large dataset of images (14M-300M), attaining excellent results. [3]

Following the successful implementation of Transformers to vision classification tasks, different approaches were tried for semantic segmentation, SegFormer was introduced as a framework for semantic segmentation that jointly considers efficiency, accuracy and robustness. Redesigning both the encoder and decoder, the primary novelties being [11]:

- A positional-encoding-free and hierarchical Transformer encoder: avoids interpolating positional codes when performing inference on images with resolutions different from the training one.

- A efficient All-MLP decoder design: aggregates information from different layers, combining local and global attention.
- Results: improved state of the art performance and efficiency on ADE20K, Cityscapes, and COCO-Stuff datasets



Encoder
Decoder
$\frac{H}{4}\times\frac{W}{4}\times C_1$
$\frac{H}{8}\times\frac{W}{8}\times C_2$
$\frac{H}{16}\times\frac{W}{16}\times C_3$
$\frac{H}{32}\times\frac{W}{32}\times C_4$
$\frac{H}{4}\times\frac{W}{4}\times 4C$
$\frac{H}{4}\times\frac{W}{4}\times N_{cls}$
Overlap Patch Embeddings
Transformer Block 1
Transformer Block 2
Transformer Block 3
Transformer Block 4
MLP Layer
MLP
Efficient Self-Attn
Mix-FFN
Overlap Patch Merging
$\times N$
$\frac{H}{2^{i+1}}\times\frac{W}{2^{i+1}}\times C_i$
$\frac{H}{2^{i+1}}\times\frac{W}{2^{i+1}}\times C$
$\frac{H}{4}\times\frac{W}{4}\times C$
MLP
UpSample

Figure 4: SegFormer consists of two main modules: A hierarchical Transformer encoder to extract coarse and fine features; and a All-MLP decoder to directly fuse these multilevel features and predict the semantic segmentation mask

The final configuration of the SegFormer pre-trained model from Huggingface is given in Table 1.

Model variant	MiT-b0
Depth	[2,2,2,2]
Hidden sizes	[32, 64, 160, 256]
Decoder hidden size	256
Params (M)	3.7
ImageNet-1k Top 1	70.5

Table 1: SegFormer MiT-b0 pre-trained

5. Methods

5.1. Data Preprocessing

5.1.1 Histogram Equalisation

In our image processing pipeline, we implemented histogram equalization as a crucial step to enhance the contrast and improve the overall visual quality of images within our dataset. The process involved redistributing pixel intensities in each image to achieve a more balanced and stretched histogram. Specifically, we computed the cumulative distribution function (CDF) of the original pixel intensities and mapped them to a new set of values based on a desired intensity distribution. This transformation effectively expanded the dynamic range of the images, mitigating the impact of uneven lighting conditions and enhancing the visibility of important features, as shown in Figure 5.

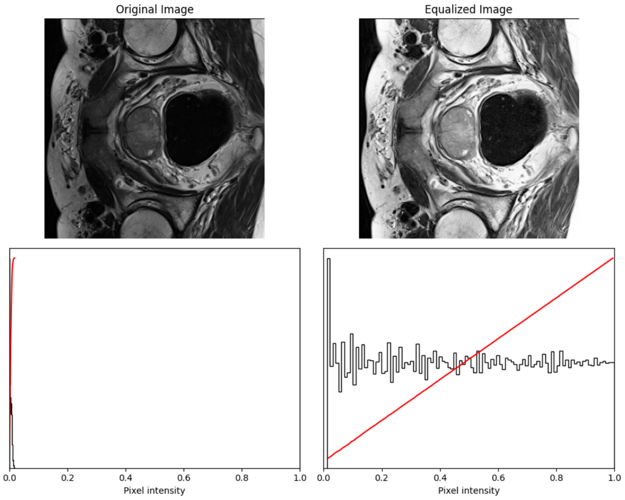


Figure 5: Example of a histogram equalization of a prostate MRI slice.

5.1.2 Data Augmentation

In our data augmentation strategy for the training set, we employ a combination of geometric and pixel-level transformations to enhance the variability and robustness of our deep learning model. For each slice in the training set, we generate two augmentations using a set of diverse functions. Firstly, we introduce rotation by a minor angle, randomly selected between -10 and 10 degrees, to simulate variations in image orientation. Secondly, we implement a scaling and cropping operation, adjusting the volume size by a random scale factor (between 1 and 1.2) and cropping around the center. Additionally, we incorporate grayscale variation by introducing a random global increment in the gray-level value of the volume. Furthermore, elastic deformation, involving the application of distortion to the volume, is applied with customizable parameters such as alpha and sigma. These augmentations collectively contribute to a more comprehensive and diverse training dataset, fostering the model's adaptability to real-world variations and improving its generalization capabilities.

5.2. Training

We conducted the experiments using Google Colab in a GPU environment for accelerated model training. The codebase was developed in Python 3, utilizing libraries for deep learning and medical image processing, such as

PyTorch for Segnet and U-net and HuggingFace for SegFormer. We trained each model for 250 Epochs or until convergence. We selected a batch size of 16 for all models. We chose the Cross-entropy loss as our loss function, which is well-suited for multi-class segmentation problems, acting as a measure of dissimilarity between the predicted segmentation and the ground truth masks. We used RMSprop as an optimizer with momentum. However, in the case of SegFormer, the decoupled weight normalized Adam optimization algorithm was used (AdamW). The learning rate was set to 1e-6 initially, and weight decay was applied to prevent overfitting.

5.3. Testing

During the training phase, metrics were calculated at each epoch for the validation sets, taking into account each label in the dataset. After the training procedure was completed, the resulting model was carefully evaluated on the selected test dataset, with multiple performance metrics computed. 2

5.3.1 Performance Metrics

Sørensen–Dice coefficient: The Sørensen-Dice coefficient measures the similarity between two samples. It is widely used in semantic segmentation tasks to calculate the ratio of the intersection of two sets to the sum of their individual cardinalities.

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|}$$

Intersection over Union (IoU): IoU, often known as the Jaccard Index, is a key statistic used to determine the similarity and dissimilarity of sample sets. In the field of image processing, IoU represents the ratio of image overlap to union.

$$IoU(X, Y) = \frac{|X \cap Y|}{|X| \cup |Y|}$$

Recall: Recall assesses the ability of a model to correctly identify and capture all instances of a specific class within an image:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives + False Negatives}}$$

These performance metrics provide a comprehensive evaluation of the models' segmentation accuracy, highlighting their effectiveness in capturing specific classes within the prostate MRI images.

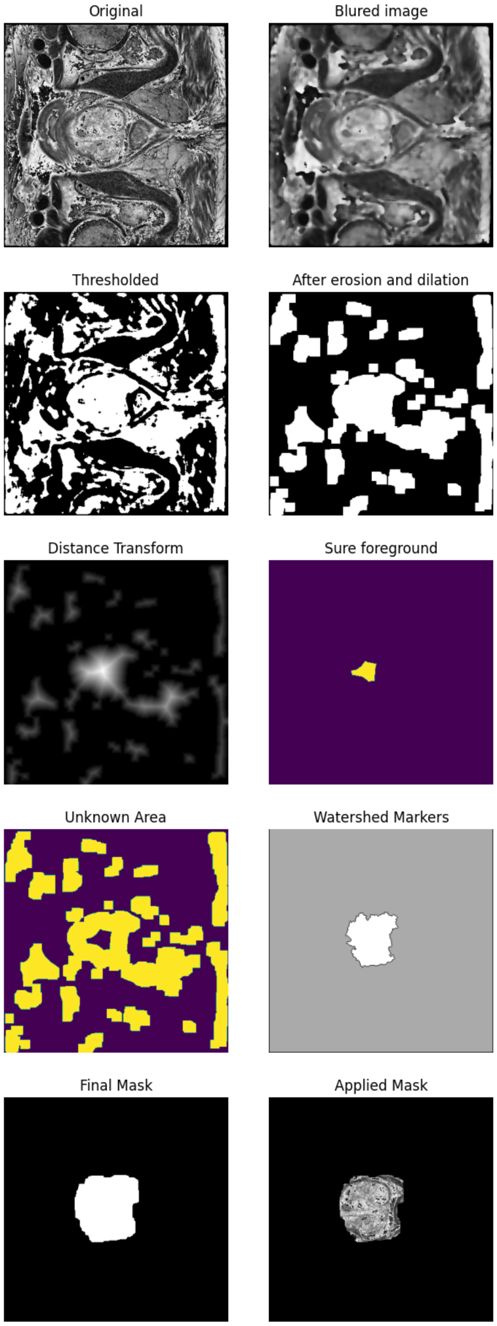


Figure 6: Watershed segmentation of the prostate (peripheral zone and central gland together) using automatic seed detection.

6. Results

In this study, we focused on applying the Watershed method to a single sample while reducing manual intervention. Figure 6 shows how several picture alterations can be used to automatically detect markers. It's worth noting that the algorithm can only detect one class at a time. Moreover, it is critical to keep in mind that this method was experimental, with no measures to assess its effectiveness.

Model	Label	IoU	Dice	Recall
U-Net	All Classes	0.987	0.994	0.991
	Background	0.996	0.998	0.999
	Peripheral Zone	0.143	0.228	0.145
	Central Gland	0.406	0.527	0.424
SegNet	All Classes	0.983	0.991	0.996
	Background	0.992	0.996	0.999
	Peripheral Zone	0.300	0.424	0.486
	Central Gland	0.441	0.556	0.618
SegFormer	All Classes	0.971	0.988	0.985
	Background	0.989	0.995	0.997
	Peripheral Zone	0.611	0.758	0.603
	Central Gland	0.696	0.820	0.899

Table 2: Model results

U-Net's segmentation of the Peripheral Zone and Central Gland might be improved, as demonstrated by lower IoU, Dice coefficient, and Recall scores for these classes. The major causes for this behavior could be a low amount of training data and, more specifically, a massive data imbalance, which caused the two classes of interest to compete with one another, and as measurements for the central gland increased, those for the peripheral zone decreased.

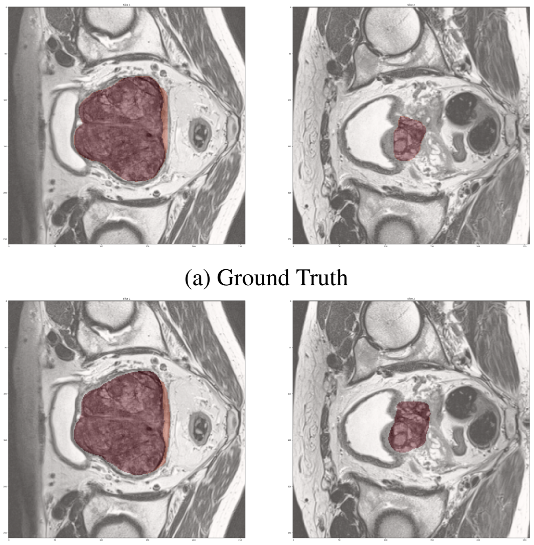
A possible solution would be to augment only the slices in the training where we have evidence of either of the two classes of interest or adopt a 3D U-Net model where instead of training on the slices independently, we train on a 3D representation of the MRI image.

SegNet, on the other hand, demonstrates good segmentation performance across all classes. It not only performs well but also outperforms U-Net in the Peripheral Zone and Central Gland, showing improvements in IoU, Dice coefficient, and Recall for these regions.

SegFormer performs very well in segmentation, as seen in Table 2, particularly for the Central Gland class. It achieves high IoU, Dice coefficient, and Recall for these areas, indicating robust segmentation. Overall, SegFormer appears to be the most effective model among the three, demonstrating superior performance across different classes. This comes as no surprise given that the Huggingface model is pre-trained on a significant quantity of data, whereas the other two models needed to be trained from scratch. Figure 1 illustrates the prediction's accuracy.

7. Conclusions

Semantic segmentation application for medical images, especially in tasks such as cancer detection is an important field of study. The challenges of semantic segmentation in the prostate has drawn much attention. Through this study we have seen that the transformer architecture applied to visual tasks, especially initializing it with pre-trained weights, such as SegFormer, can perform very well, getting close to the state of the art dice coefficients for segmentation of the central gland, which stands at around 0.85 [10].



(a) Ground Truth

(b) Predictions

Figure 7: SegFormer: Ground Truth vs Predictions

References

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (Oct. 2016).

[2] Albert Comelli et al. "Deep learning-based methods for prostate segmentation in magnetic resonance imaging". In: *Applied Sciences* 11.2 (2021), p. 782.

[3] Alexey Dosovitskiy et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020).

[4] Zhe Huang et al. "Automatic multi-organ segmentation of prostate magnetic resonance images using watershed and nonsubsampled contourlet transform". In: *Biomedical Signal Processing and Control* 25 (2016), pp. 53–61.

[5] Abhishek Kumar. *Semantic Segmentation — SegNet*. June 2020. URL: https://medium.com/@abhishekkakiak/semantic-segmentation-segnet-a54af19b6d6.

[6] Xin Li et al. “An uncertainty-aware deep learning architecture with outlier mitigation for prostate gland segmentation in radiotherapy treatment planning”. In: *Medical physics* 50.1 (2023), pp. 311–322.

[7] Nuno M Rodrigues et al. “A comparative study of automated deep learning segmentation models for prostate mri”. In: *Cancers* 15.5 (2023), p. 1467.

[8] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*. Springer. 2015, pp. 234–241.

[9] Arindrajit Seal, Arunava Das, and Prasad Sen. “Watershed: an image segmentation approach”. In: *International Journal of Computer Science and Information Technologies* 6.3 (2015), pp. 2295–2297.

[10] Carine Wu et al. “Automatic segmentation of prostate zonal anatomy on MRI: a systematic review of the literature”. In: *Insights into Imaging* 13.1 (2022), pp. 1–17.

[11] Enze Xie et al. “SegFormer: Simple and efficient design for semantic segmentation with transformers”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 12077–12090.