0

Cover Letter

This is the project report of the Semantic Segmentation Project for CSDS600 Fall 2022.

Title: **Semantic Segmentation of Prostate Gland from MR Images across Different Loss Functions**

Author: **Cerag Oguztuzun** ***cxo147@case.edu***.

Submitted on 26/10/2022

Semantic Segmentation of Prostate Gland from MR Images across Different Loss Functions

Çerağ Oğuztüzün
Department of Computer and Data Sciences
Case Western Reserve University, Cleveland, OH 44106, USA
E-mail: cxo147@case.edu

The growing use of medical image analysis has given rise to the segmentation of organs, diseases, and abnormalities within medical images. In recent years, deep neural models have shown considerable promise for segmenting medical images and helping monitor tumor growth and administering medicine. A growing part of magnetic resonance imaging is image analysis. Semantic segmentation allows regions of interest in an image to be described, categorize, and visualized. It is possible to segment a large number of images with an enormous variety using semantic image segmentation. This is done by assigning a class of objects to every pixel within an image. The prostate gland is segmented from magnetic resonance images using deep learning techniques using semantic segmentation. Moreover, Tversky loss functions were analyzed and compared to obtain the highest Dice and F1 scores.[1] Binary Cross Entropy with Logit Loss is used as a baseline loss function.[2] Experiments with different Tversky Loss functions are performed by weighting recall and precision values. Results are compared in terms of accuracy and performance.

1. Introduction

Medical image analysis has become increasingly sophisticated and the segmentation of organs, diseases, or abnormalities in medical images has become increasingly relevant. Medical image segmentation can be challenging due to the various artifacts present in the images. It can also be used to monitor the growth of diseases like tumors and control medication dosages. Deep neural models have recently demonstrated their application to various image segmentation tasks.[3] Due to the high performance and achievements of deep learning strategies, this significant growth has been achieved.

An image data set that is obtained by Magnetic Resonance Imaging (MRI) or Computed Tomography (CT) must be segmented in order to identify areas of interest. With semantic segmentation, each pixel of an image is assigned a specific class, and then the class is assigned a specific class. The classification is output as a mask image.

With this study, we train a U-net model to perform semantic segmentation of prostate glands in MR images using a deep learning approach.[4] The model aims to segment prostate gland contours accurately from MR scans of patients. For the highest Dice score and the highest F1 score, our approach includes analysis and comparison of relevant loss functions. The Binary Cross Entropy with Logit Loss function is employed as a baseline loss function. The experiments are carried out with different weights for false negatives and false positives to compute Tversky Loss functions.[1] A comparison is made between the findings based on

performance and accuracy.

According to our hypothesis, the model with α=0.25 will perform the best segmentation. False Negatives will be penalized more than False Positives. For False Positives, the weight is directly proportional to α, while for False Negatives, the weight is directly proportional to $1 - \alpha$. It is intuitive that False Negatives in medical imaging pose a greater risk since they may cause the reader not to recognize a patient's potential threat. Segmenting not only the periphery but the contours of the prostate gland is important when segmenting tasks because the mask must fit into the contours of the prostate gland. The results of the model will also have higher recall and a lower precision value.

2. Materials and Methods

We used U-net to segment the prostate gland from an MRI scan. We experimented with 8 loss functions to characterize the association between loss functions and model success. Furthermore, we characterize the association between penalizing different components of the confusion matrix for classification and the success of the model as measured by precision and recall.

In the following subsections, we first describe our dataset and then present how we address these experiments.

2.1. *Description of Data*

The large set of externally annotated PROSTATEx Gland Segmentations is used as the dataset.[5] This dataset is consisted of manual segmentations of prostate glands completed by Cuocolo et al.[6] using the PROSTATEx dataset. It is comprised of mask images for 204 patients created using axial T2 weighted MRI scans.[7] These scans were acquired using a turbo spin echo sequence and had a resolution of around 0.5 mm in plane and a slice thickness of 3.6 mm.[6]

2.2. *Data Pre-Processing*

The files are downloaded in compressed form in NIFTI format. Each of 204 patients had a number of MRI scans which in total yielded 4166 images in total. The images are decompressed and converted to png format. The conversion to dicom to png is achieved using an external library *dicom2png*[8] for all images. The dataset provides a mask image for each prostate gland MRI scan image. Hence we have 4166 masks for 4166 prostate gland MRI scan images. Each mask image and grayscale prostate gland scan has 384x384 dimension.

Using a 80 to 20 ratio for partitioning the dataset into training and validation sets, 3191 pairs of masks and prostate images are used for the training set and 975 pairs are used for the validation set.

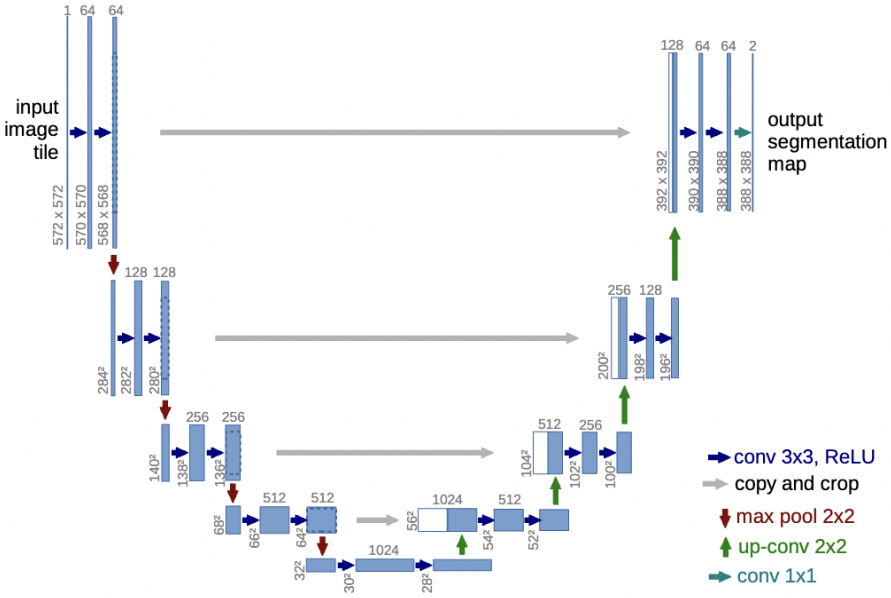


Fig. 1: U-Net Model Architecture[4]

2.3. *U-net Model*

In any semantic segmentation task today, the U-net is one of the most frequently used approaches.[4] In our study, we chose U-net because it is a standard, baseline model in semantic segmentation and is designed to learn from fewer training samples. It is a fully convolutional neural network. It is easy to customize.

The downsampling/encoding path consists of convolutions, ReLU, max pooling, and upsampling, convolutions, and ReLU, while the upsampling/decoding path consists of convolutions and ReLU. In addition, the U-net model worked well with small datasets, as demonstrated in the paper that initially proposed it, and it got favorable results.[4]

The U-shaped architecture of the model consists of:

(1) As the model learns what is in the image, but loses spatial information as it contracts, the downsampling path appears to be creating something similar to a scale space.

(2) In this expansive path, the model recovers the information lost in downsampling by propagating these coarse, high-level features into each original pixel.

(3) In the horizontal path from a downward level to an upward level, the lost details are re-injected. These are called skip connections.

2.3.1. *Data Loading Pipeline*

To input the data into the model, the images are converted to pixel arrays. A pixel of 255 indicates a white pixel, while a pixel of 0 indicates a black pixel. Using the sigmoid activation as our last activation, we normalized the pixel values between 0 and 1, where 1 denotes the probability that the pixel is white.

Using the Albumentations library, data augmentation is implemented to reduce overfitting and prevent data scarcity in order to improve model prediction accuracy.[9] For the train set, resizing, rotating, flipping horizontally and vertically, and normalizing pixel values are part of the data augmentation. For the test set, resizing and normalizing of pixel values are included in the data augmentation.

If the skip connection shape does not match the input shape in the skip connections, resizing is implemented. The size of the input image is resized to the shape of a skip connection. The intuition behind this is that, since in the downsampling path the images are reduced in size, the reduction in size will not change the performance of the model when used to match the dimensions in the skip connections part of the model.

2.3.2. *Model Specifications*

All experiments are executed on the NVIDIA GeForce RTX 2080 Ti GPU on the CWRU HPC cluster.[10]

For all models, learning rate of 0.0001 is used as It achieved good convergence in every loss function we used. The Adam optimizer is used. The training is done in batches of size 16. The maximum number of epochs are 100 but early stopping is implemented where the training is stopped if model accuracy does not improve by 0.3 for 3 epochs. The hyperparameter tuning is done by experimenting with a variety of values. The model implementation is inspired by an open source repository .[11]

2.4. *Loss Functions*

When a model is being trained (using already-labeled data), the loss function informs it how well it is approaching optimal parameters. In the context of image segmentation, it guides the model in finding the "ideal" approximation of the input data to the output data. Each training batch's residuals are used to calculate the overall error in neural network models. Loss functions are used to calculate the error. In this section, we will introduce the eight loss functions that are implemented, as their choice affects how the models adjust their internal weights when performing backpropagation.

2.4.1. *Binary Cross Entropy with Logits Loss*

Binary Cross-Entropy (BCE) is the default loss function used in segmentation and binary classification, which is why we used it as our baseline loss function.[12] Logits Loss combines a Sigmoid layer and a BCE Loss in one single class to create a numerically stable version. Prior to loss computation, we convert all real values between 0 and 1 using the sigmoid activation function on the final output. The Binary Cross Entropy Loss can be formulated as follows:

$$BCE = -\sum_{i=1}^{C'=2} t_i log(s_i) = -t_1 log(s_1) - (1 - t_1) log(t - s_1) \tag{1}$$

In this formulation C' refers the to number of classes where for binary cross entropy the

value of C' is 2, pixels can be 0 or 1. t_i refers to the target for class i, s_i refers to the score for class i.

2.4.2. *Tversky Loss*

Observing the target image set, we can see that the distribution of observations in the target class is very uneven. We are using very imbalanced data. It is undesirable to train with unbalanced data, especially in medical applications, where false negatives are much less tolerable than false positives. This results in predictions with high precision but low recall. Salehi et al. (2017) proposed the Tversky Loss function as a solution to the challenge of training networks on highly imbalanced data.[1]

$$Tversky = \frac{TP}{(TP + \alpha FP + \beta FN)} \tag{2}$$

α is the weight of penalty to give for false positives where β is the weight of penalty to give for false negatives where $\alpha + \beta = 1$. Higher β (lower α) must lead to higher recall and lower specificity.

Note that, in the case of $\alpha = \beta = 0.5$ the loss simplifies to be equivalent to Dice loss, which is also equivalent to the F1 score.

The Dice Coefficient or Tversky metrics are commonly used as metrics by competitors to measure model performance. Our loss functions are similarly derived from these metrics - usually in form 1 - f(x) where f(x) is the metric used.

2.5. *Model Performance*

The performance of our models are evaluated by the Dice Similarity Coefficient and the F1 Score.

2.5.1. *Dice Similarity Coefficient*

For the evaluation of medical image segmentation, the Dice is a popular performance metric. It is a measure of the overlap between a segmentation result and its ground truth. The value of Dice ranges from 0 to 1, with 1 representing the most similarity between the prediction and the truth.

$$Dice(A, B) = \frac{2 \times |A \cap B|}{(A + B)} \tag{3}$$

A refers to the pixel values of predicted segmentation result images and B refers the the pixel values of the target images.

2.5.2. *F1 Score*

The F1 Score combines precision and recall into a single measure that captures both characteristics. We are experimenting with precision-recall tradeoffs, so the harmonic mean of precision

and recall is an effective metric to combine precision and recall values. Furthermore, the F1 score is also useful when the data is imbalanced. F1 Scores range from 0 to 1, and 1 is the perfect F1 Score.

$$F1Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4}$$

3. Results

Table 1: **Performance metrics results** on validation set for models trained using different loss functions are recorded. The best values for each metric have been highlighted in bold.

Loss Function	Dice Score	Precision	Recall	F1 Score
BCE Loss	0.85551	0.81207	0.91116	0.85876
Tversky Loss $\alpha = 0.25$ $\beta = 0.75$	0.85253	0.81798	0.92828	0.86965
Tversky Loss $\alpha = 0.35$ $\beta = 0.65$	0.85776	**0.91000**	0.92189	**0.91590**
Tversky Loss $\alpha = 0.45$ $\beta = 0.55$	0.83326	0.43057	**0.97267**	0.59691
Dice Loss (Tversky Loss $\alpha = 0.50$ $\beta = 0.50$)	0.86958	0.76809	0.94913	0.84907
Tversky Loss $\alpha = 0.65$ $\beta = 0.35$	0.86581	0.70920	0.52906	0.60603
Tversky Loss $\alpha = 0.75$ $\beta = 0.25$	**0.89506**	0.76617	0.81707	0.79081
Tversky Loss $\alpha = 0.85$ $\beta = 0.15$	0.88112	0.64821	0.91519	0.75890

In order to observe the effect of using different loss functions, especially to observe how the Tversky loss function can yield different effects by hyperparameter tuning, we recorded the evaluation metrics for each loss function in Table 1. Generally, as the α value increases, and as the β value decreases, the precision-recall tradeoff should be balanced, but we cannot observe this. As expected, higher β values provided higher recall values, however, did not provide lower precision values. Counter-intuitively, the precision value decreases as α value decreases. Also, comparing the two models which used $\alpha = 0.45$ and $\alpha = 0.50$, the difference in precision is almost twice, however, the difference in α is only 0.5. The highest performing model is trained by Tversky Loss $\alpha = 0.75$ $\beta = 0.25$.

Higher β were hypothesized to be yield higher recall values, in Figure 3 we are able to observe an almost identical trend in precision and recall values. Also we are not able to observe a determined trend between models compared to Dice Loss where α and β are 0.5.

The F1 score can give us a summarized information on precision and recall values for model. In Figure 4, we can observe two bell curves to the left and right of Dice Loss, in the F1 Score column. The Dice Score also follows a rising trend as the α penalty weight gets higher. However, we cannot see a similarity between Dice Score and F1 Score.

4. Discussion

From Figure 5 when comparing the contours of masks with the ground truth(a), as the α values increase in loss functions (c to i) the boundaries get more conserved. As α is a penalty

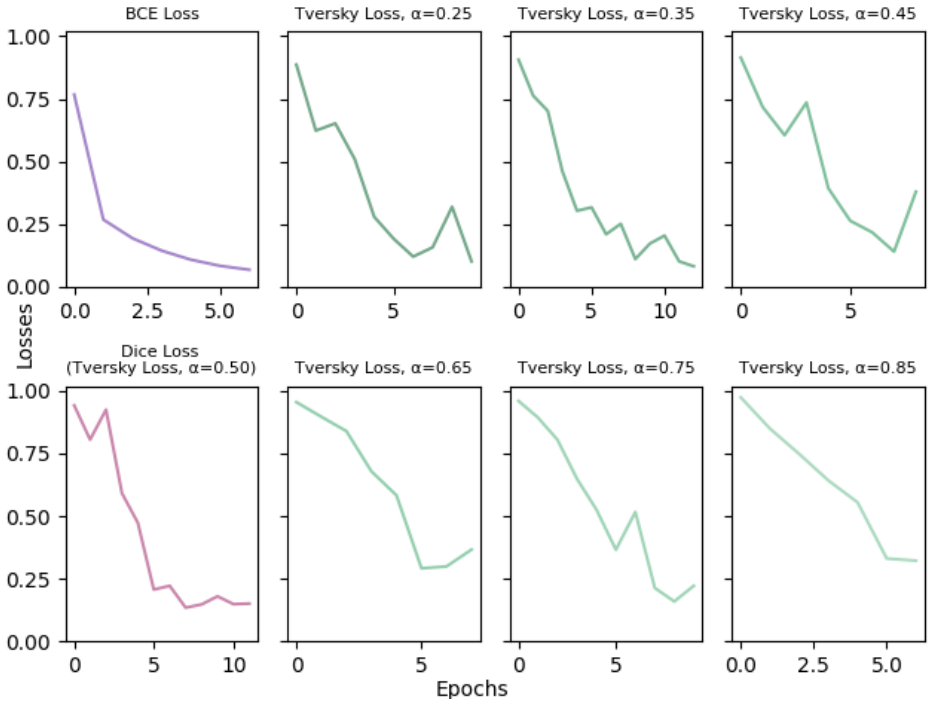


Fig. 2: **Losses** plotted for each model per epoch. Due to early stopping not all models trained for the same time, but all models reached to a convergence.

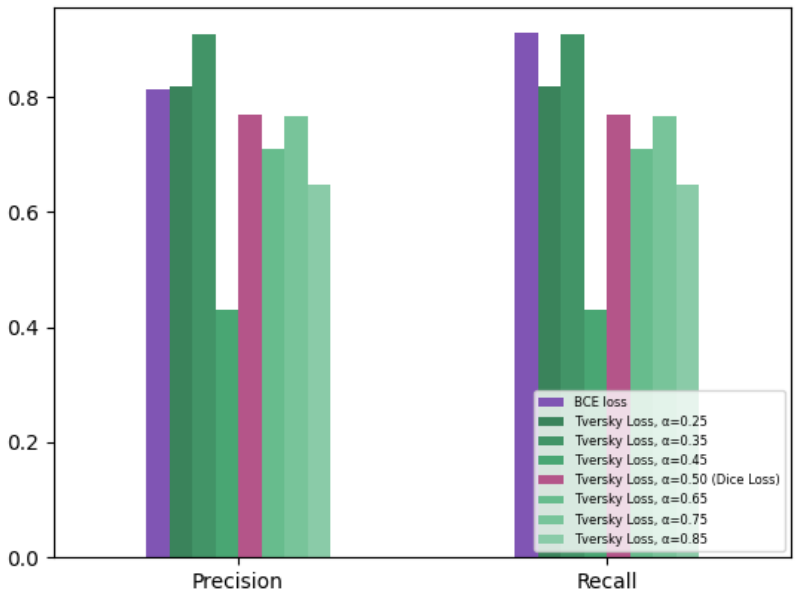


Fig. 3: **Precision and Recall** plotted for each model. Lighter color is given to models with higher α values, dark red color is used to indicate equal α and β values.

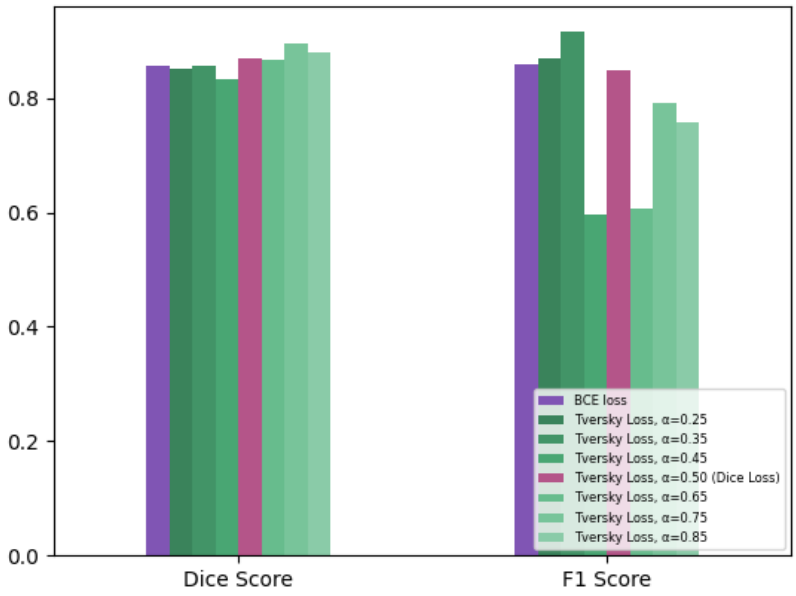
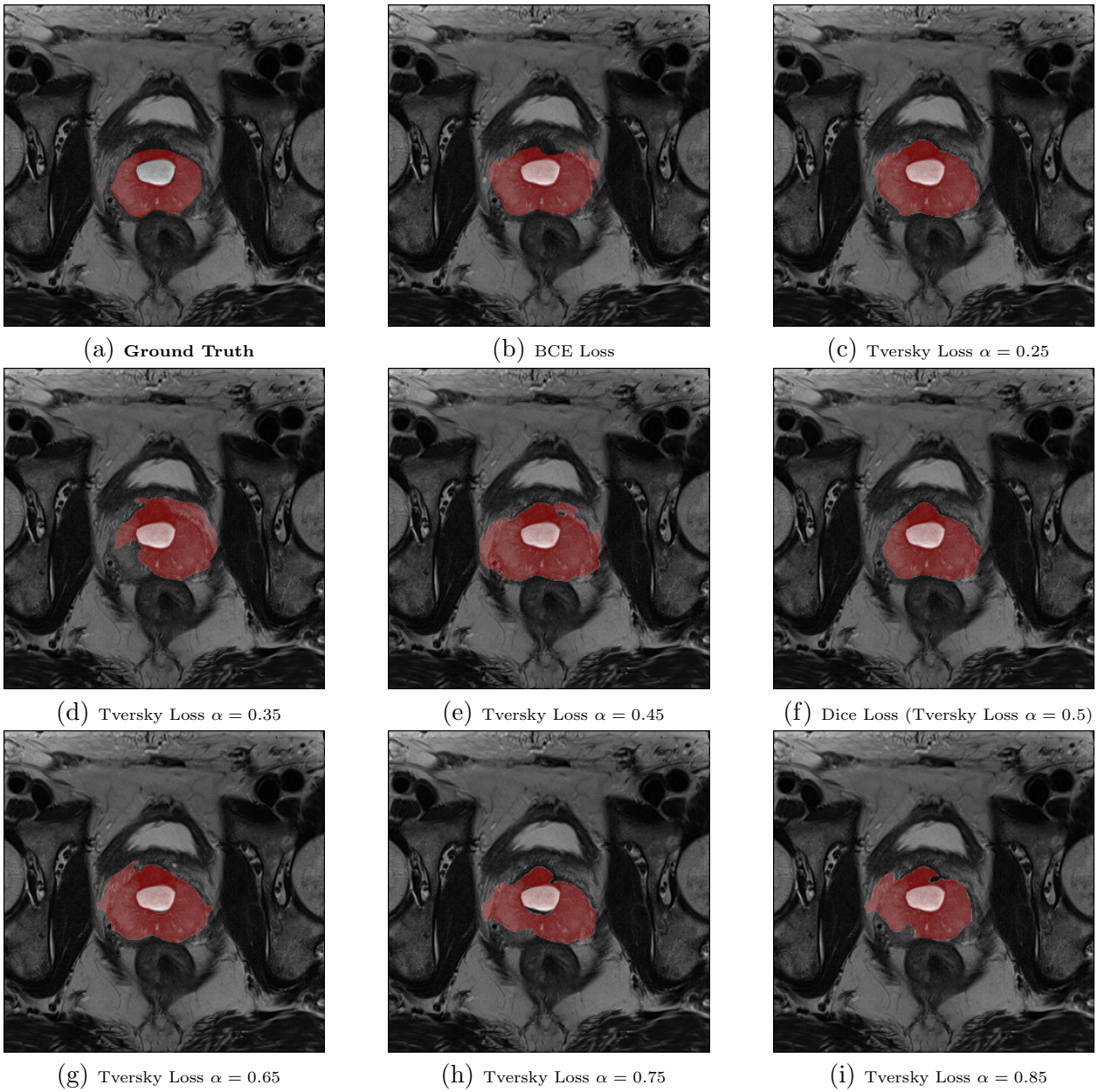


Fig. 4: **Dice and F1 Scores** plotted for each model. Lighter color is given to models with higher α values, dark red color is used to indicate equal α and β values.

value to False Positives, extra pixels outside the boundary of the ground truth image gets more penalized and the model learns to keep the boundaries more conserved as the model gets penalized by False Positive pixels. Similarly, as the β values increase (observe from i to b), the boundaries get more spread out. As False Negatives get penalized more, the model learns to spread the boundary of the segmentation and does not have and indented contours that go within the mask, the mask gets more convex and gets penalized more if It gets concave due to high False Negative penalty. These observations fit the intuition and theoretical basis of the Tversky Loss Function.

However in Table 1, the precision and recall values does not fit the intuition and theoretical basis of the Tversky Loss Function. The precision-recall tradeoff would be maintained at a balance as the α value increases and the β value decreases. However, this is not observed. It was expected that higher β values produced higher recall values, however not lower precision values. This observation does not fit the results we are able to derive from Figure 5, hence this is believed to be caused by the imbalanced dataset. In our hypothesis we thought that the model which uses the Tversky Loss function with α=0.25 will perform the best segmentation, because penalizing False Negatives more heavily could enable the model to learn to keep the boundaries of the segmentation more restricted. A mask with larger size than the original size of the prostate gland can have bad implications in medicine If the doctors are trying to see the progression of a disease.

However, the most successful model in terms of Dice Score was the model trained with



(a) **Ground Truth** (b) BCE Loss (c) Tversky Loss $\alpha = 0.25$

(d) Tversky Loss $\alpha = 0.35$ (e) Tversky Loss $\alpha = 0.45$ (f) Dice Loss (Tversky Loss $\alpha = 0.5$)

(g) Tversky Loss $\alpha = 0.65$ (h) Tversky Loss $\alpha = 0.75$ (i) Tversky Loss $\alpha = 0.85$

Fig. 5: **Scan Images and Segmentation Overlays** For each prostate gland MR scan image, the segmentation predictions made by the models trained using different loss functions. This particular MR scan has the Patient ID 0200, the sequence name z013.

Tversky Loss $\alpha = 0.75$ $\beta = 0.25$. In Table 1 we can see that this model has the highest Dice Score of 0.89506 where 1 indicates a complete overlap between the ground truth and the image. Also, It is important to observe that, the only model that was able to capture the inner negative space that is not a part of the mask, is again the model with the best Dice score.

In terms of F1 score, the model trained with Tversky Loss $\alpha = 0.35$ $\beta = 0.65$ has the highest F1 score of 0.91590, this is because both the precision and recall values for this model

are very high and balanced. We thought we would see a equivalence between Dice Score and F1 Score since intuitively, union is equal to the sum of True Positive, False Negative and False positive values where intersection is equal to the True Positive values. In our results in Figure 3 we were not able to observe a direct proportion. This might be caused by the data imbalance which limited us to observe a clear precision recall trade off in our study. Given this, we believe the Dice Score is a more reliable performance metric than F1 Score, as It is more robust to the precision recall trade off and data imbalance.

In Salehi et al,[1] It was proposed that the Tversky Loss Function would help with imbalanced data. Although we can see improvements to Dice Score in Figure 4, when compared to the BCE Loss, the improvements evaluated with the Dice Score seems more reliable and robust than the F1 Score. More statistical analyses can be made in the future to be able to quantify and communicate these observations.

5. Conclusion

The model that penalized False Positives more heavily (Tversky Loss with $\alpha = 0.75$) performed the best and yielded the highest Dice Score. In our experiments with Tversky Loss function with different hyperparameters, we were not able to observe a precision-recall trade-off, which is believed to be caused by imbalance of the dataset. This imbalance caused us to not being able to observe a direct correlation between Dice Score and F1 Score. This made us conclude that Dice Score is a more robust performance metric than F1 Score. Although, by using Tversky Loss we were able to see improvement in the models compared to our baseline model which is trained using BCE Loss.

The work on semantic segmentation carries great importance and the trade off between penalizing False Positive and False Negative pixels carries great importance because It changes the finding's size and contour in a way that can affect the progress control of the patient's disease, which can have implications in the patient's side. To sum up, more robust evaluation metrics that use trade-off of concepts with great detail and attention must be used in practice.

References

1. S. S. Salehi, D. Erdogmus and A. Gholipour, Tversky loss function for image segmentation using 3d fully convolutional deep networks, *Machine Learning in Medical Imaging* , p. 379–387 (2017).
2. U. R. Dr.A, Binary cross entropy with deep learning technique for image classification, *International Journal of Advanced Trends in Computer Science and Engineering* **9**, p. 5393–5397 (2020).
3. R. Wang, T. Lei, R. Cui, B. Zhang, H. Meng and A. K. Nandi, Medical image segmentation using deep learning: A survey, *IET Image Processing* **16**, p. 1243–1267 (2022).
4. O. Ronneberger, Invited talk: U-net convolutional networks for biomedical image segmentation, *Informatik aktuell* , p. 3–3 (2017).
5. Spie-aapm-nci prostatex challenges (prostatex) the cancer imaging archive (tcia) public access - cancer imaging archive wiki.
6. R. Cuocolo, A. Stanzione, A. Castaldo, D. R. De Lucia and M. Imbriaco, Quality control and

whole-gland, zonal and lesion annotations for the prostatex challenge public dataset, *European Journal of Radiology* **138**, p. 109647 (2021).
7. Rcuocolo, Rcuocolo/prostatex_masks: Lesion and prostate masks for the prostatex training dataset, after a lesion-by-lesion quality check.
8. Dicom2jpg.
9. Albumentations.
10. Cwru hpc cluster.
11. Aladdinpersson, Aladdinpersson/machine-learning-collection: A resource for learning about machine learning amp; deep learning.
12. Bce with logits loss, pytorch 1.12 documentation.