A Survey of Loss Functions for Semantic Segmentation

Shruti Jadon

August 19, 2020

A survey of loss functions for semantic segmentation

Shruti Jadon
IEEE Member, UMass Amherst
CA, USA
sjadon@umass.edu

***Abstract*—Image Segmentation has been an active field of research, as it has the potential to fix loopholes in healthcare, and help the mass. In the past 5 years, various papers came up with different objective loss functions used in different cases such as biased data, sparse segmentation, etc. In this paper, we have summarized some of the well-known loss functions widely used for Image segmentation and listed out the cases where their usage can help in fast and better convergence of a Model. Furthermore, We have also introduced a new log-cosh dice loss function and compared its performance on NBFS skull-stripping with widely used loss functions. We showcased that certain loss functions perform well across all datasets and can be taken as a good choice in unknown-distribution datasets.**

***Index Terms*—Computer Vision, Image Segmentation, Medi Image, Loss Function, Optimization, Healthcare, Skull Strippi Deep Learning**

I. Introduction

Deep learning has revolutionized various industries rangi from software to manufacturing. Medical community h also benefitted in large from deep learning, innovations frc disease classification, tumor segmentation using U-Net, canc detection using SegNet, CapsNet has saved many hours physicians and helped reduce costs by millions of dolla Among this, Image segmentation is one of the crucial co.. tribution of deep learning community to medical fields, as apart from telling that some disease exists it also showcase where exactly it exists, which has drastically helped in creating automated softwares to detect lesions etc in CT scans.

Image Segmentation can be defined as classification task on pixel level; an image consists of various pixels, and these pixels grouped together define different elements in image, therefore a method of classifying these pixels into the a elements is called semantic image segmentation. While designing such complex image segmentation based Deep learning architectures we come across a crucial choice, which loss/objective function to choose, as they instigate the learning process of algorithm. The choice of loss function is very crucial for any architecture to learn proper objective, and therefore since 2012 various researchers have came across to design domain specific loss function to obtain better results for their datasets. In this paper we have summarized 15 such segmentation based loss functions that has been proven to provide state of art results in different domains. These loss function can be widely categorized into 4 categories: . Distribution-based, Region-based, Boundary-based, and Compounded (Refer I). We have also discussed the conditions to determine which objective/loss function might be useful in a scenario. Apart from this, we have also proposed a new log-cosh dice loss function for semantic segmentation. To showcase its efficiency, we have also compared the performance of all loss functions on NBFS Skull-stripping dataset and shared the outcomes in form of Dice Coefficient, Sensitivity, and Specificity. The code implementation is available at GitHub: https://github.com/shruti-jadon/Semantic-Segmentation-Loss-Functions.

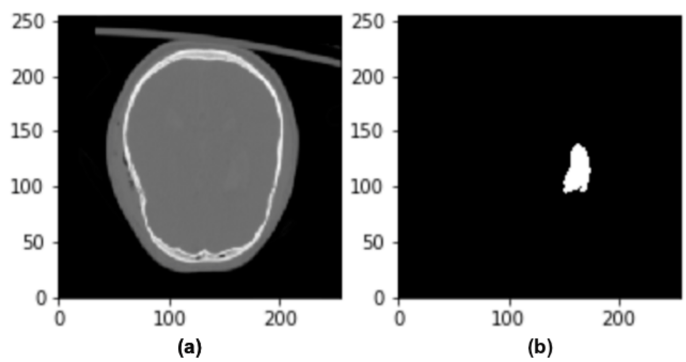

Fig. 1. Sample Brain Lesion Segmentation CT Scan [1]. In this segmentation mask you can see, that number of pixels of white area(targeted lesion) is ¡ number of black pixels.

TABLE I
Types of Semantic Segmentation Loss Functions

Type	Loss Function
Distribution-based Loss	Binary Cross-Entropy Weighted Cross-Entropy Balanced Cross-Entropy Focal Loss Distance map derived loss penalty term
Region-based Loss	Dice Loss Sensitivity-Specificity Loss Tversky Loss Focal Tversky Loss **Log-Cosh Dice Loss**(ours)
Boundary-based Loss	Hausdorff Distance loss Shape aware loss
Compounded Loss	Combo Loss Exponential Logarithmic Loss

II. Loss Functions

Deep Learning Algorithms uses stochastic gradient descent approach to optimize and learn the objective. To learn an objective accurately and faster, we need to ensure that our mathematical representation of objectives, also known as loss functions are able to cover even the edge cases. The introduction of loss functions have roots in traditional machine learning, where these loss functions were derived on basis of distribution of labels. For example, Binary Cross Entropy is derived from Bernoulli distribution and Categorical Cross-Entropy from Multinoulli distribution. In this paper, we have focused on Semantic segmentation instead of Instance Segmentation, therefore the number of classes at pixel level is restricted to 2. Here, we will go over 15 widely used loss functions and understand their use-case scenarios.

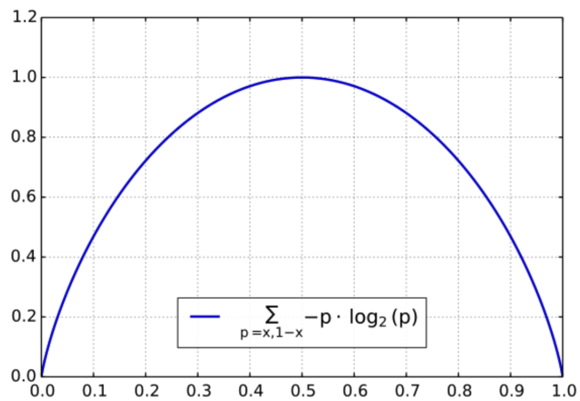

Fig. 2. Graph of Binary Cross Entropy Loss Function. Here, Entropy is defined on Y-axis and Probability of event is on X-axis.

A. Binary Cross-Entropy

Cross-entropy [2] is defined as a measure of the difference between two probability distributions for a given random variable or set of events. It is widely used for classification objective, and as segmentation is pixel level classification it works well. Binary Cross-Entropy is defined as:

$L_{BCE}(y,\hat{y}) = -(ylog(\hat{y}) + (1-y)log(1-\hat(y)))$

Here, $\hat{y}$ is the predicted value by the prediction model.

B. Weighted Binary Cross-Entropy

Weighted Binary cross entropy (WCE) [3] is a variant of binary cross entropy variant. In this the positive examples get weighted by some coefficient. It is widely used in case of skewed data [4] as shown in figure 1. Weighted Cross Entropy can be defined as:

$L_{W-BCE}(y,\hat{y}) = -(y\beta log(\hat{y}) + (1-y)log(1-\hat(y)))$

Note: β's value can be used to tune false negatives and false positives. E.g; If you want to reduce the number of false negatives then set $\beta > 1$, similarly to decrease the number of false positives, set $\beta < 1$.

C. Balanced Cross-Entropy

Balanced cross entropy (BCE) [5] is similar to Weighted Cross Entropy. The only difference is that in this apart from just positive examples [6], we also weight also the negative examples. Balanced Cross-Entropy can be defined as follows:

$L_{BCE}(y,\hat{y}) = -(\beta * ylog(\hat{y}) + (1-\beta) * (1-y)log(1-\hat(y)))$

Here, β is defined as $1 - \frac{y}{H*W}$

D. Focal Loss

Focal loss (FL) [7] can also be seen as variation of Binary Cross-Entropy. It down-weights the contribution of easy examples and enable model to focus learning more on hard examples. It works well for highly imbalanced class scenario, as shown in fig 1. Lets look at how this focal loss is designed. We will first look at binary cross entropy loss and learn how Focal loss is derived from cross-entropy.

$$CE = \begin{cases} -log(p), & \text{if } y = 1 \\ -log(1-p), & \text{otherwise} \end{cases} \tag{1}$$

To make convenient notation, Focal Loss define the estimated probability of class as:

$$p_t = \begin{cases} p, & \text{if } y = 1 \\ 1-p, & \text{otherwise} \end{cases} \tag{2}$$

Therefore, Now Cross-Entropy can be written as,

$CE(p,y) = CE(p_t) = -log(p_t)$

Focal Loss propose to down-weight easy examples and focus training on hard negatives using a modulating factor, $(1-p)t)^{\gamma}$ as shown below:

$FL(p_t) = -\alpha_t(1-p_t)^{\gamma}log(p_t)$

Here, $\gamma > 0$ and when $\gamma = 1$ Focal Loss works like Cross-Entropy loss function. Similarly, α generally range from [0,1], It can be set by inverse class frequency or treated as a hyper-parameter.

E. Dice Loss

The Dice coefficient is widely used metric in computer vision community to calculate the similarity between two images. Later in 2016, it has also been adapted as loss function known as Dice Loss [8].

$DL(y,\hat{p}) = 1 - \frac{2y\hat{p}+1}{y+\hat{p}+1}$

Here, 1 is added in numerator and denominator to ensure that the function is not undefined in edge case scenarios such as when $y = \hat{p} = 0$.

F. Tversky Loss

Tversky index (TI) [9] can also be seen as an generalization of Dice's coefficient. It adds a weight to FP (false positives) and FN (false negatives) with the help of β coefficient.

$TI(p,\hat{p}) = \frac{p\hat{p}}{p\hat{p}+\beta(1-p)\hat{p}+(1-\beta)p(1-\hat{p})}$

Here, when $\beta = 1/2$, It can be solved into regular Dice coefficient. Similar to Dice Loss, Tversky loss can also be defined as:

$TL(p,\hat{p}) = 1 - \frac{1+p\hat{p}}{1+p\hat{p}+\beta(1-p)\hat{p}+(1-\beta)p(1-\hat{p})}$

G. Focal Tversky Loss

Similar to Focal Loss, which focuses on hard example by down-weighting easy/common ones. Focal Tversky loss [10] also attempts to learn hard-examples such as with small ROIs(region of interest) with the help of γ coefficient as shown below:

$FTL = \sum_c (1 - TI_c)^{\gamma}$

here, TI indicates tversky index, and γ can range from [1,3].

H. Sensitivity Specificity Loss

Similar to Dice Coefficient, Sensitivity and Specificity are widely used metrics to evaluate the segmentation predictions. In this loss function, we can tackle class imbalance problem using w parameter. The loss [11] is defined as:

$SSL = w * sensitivity + (1 - w) * specificity$, where, $sensitivity = \frac{TP}{TP+FN}$ and $sensitivity = \frac{TN}{TN+FP}$

I. Shape-aware Loss

Shape-aware loss [12] as the name suggests takes shape into account. Generally, all loss functions work at pixel level, Shape-aware loss calculates the average point to curve Euclidean distance among points around curve of predicted segmentation to the ground truth and use it as coefficient to cross-entropy loss function.It is defined as follows:

$E_i = D(\hat{C}, C_{GT})$

$L_{shape-aware} = -\sum_i CE(y, \hat{y}) - \sum i E_i CE(y, \hat{y})$

Using E_i a network learns to produce a prediction masks similar to the training shapes.

J. Combo Loss

Combo loss [13] is defined as a weighted sum of Dice loss and a modified cross entropy. It attempts to leverage the flexibility of dice loss of class imbalance and at same time use cross-entropy for curve smoothing. It's defined as:

$L_{m-bce} = -\frac{1}{N} \sum_i \beta(y - log(\hat{y})) + (1-\beta)(1-y)log(1-\hat{y})$

$CL(y, \hat{y}) = \alpha L_{m-bce} - (1 - \alpha) DL(y, \hat{y})$

Here DL is Dice Loss.

K. Exponential Logarithmic Loss

Exponential Logarithmic Loss [14] function focuses on less accurately predicted structures using combined formulation of Dice Loss and Cross Entropy loss. [] propose to make exponential and logarithmic transforms to both Dice loss an cross entropy loss, to incorporate benefits of finer decision boundaries and accurate data distribution. It is defined as:

$L_{Exp} = w_{Dice} L_{Dice} + w_{cross} L_{cross}$, where

$L_{Dice} = E(-ln(DC)^{\gamma_{Dice}})$

$L_{cross} = E(w_l(-ln(p_l))^{\gamma_{cross}}))$.

In this paper wong et. al. [14] have used $\gamma cross = \gamma_{Dice}$ for simplicity.

L. Distance map derived loss penalty term

Distance Maps can be defined as distance(euclidean,absolute, etc) between the ground truth and the predicted map. There are 2 ways to incorporate distance maps, either create neural network architecture, where there's a reconstruction head along with segmentation, or induce it into loss function. Following same theory, [15] have used distance maps, derived from ground truth masks and created a custom penalty based loss function. Using this approach, its easy to guide the network's focus towards hard-to-segment boundary regions. The loss function is defined as:

$L(y, p) = \frac{1}{N} \sum_{i=1}^{N} (1 + \phi)(\odot) L_{CE}(y, p)$

Here, ϕ are generated distance maps

Note Here, constant 1 is added to avoid vanishing gradient problem in U-Net and V-Net architectures.

M. Hausdorff Distance Loss

Hausdorff Distance (HD) is a metric used by segmentation approaches to track the performance of a model. It is defined as:

$d(X, Y) = max_{x \epsilon X} min_{y \epsilon Y} ||x - y||_2$

The objective of any segmentation model is to maximize the Hausdorff Distance [16], but due to its non-convex nature, its not widely used as loss function. [17] has proposed 3 variants of Hausdorff Distance based loss functions, which both incorporate the metric use case as well as ensure that the loss function is tractable.

N. Correlation Maximized Structural Similarity Loss

A lot of semantic based segmentation loss mainly focus on classification error at pixel level, while disregarding the pixel level structural information. Some other loss functions [18] have attempted to add information using structural priors such as CRF, GANs, etc. In this loss functions, authors have introduced a structural similarity loss (SSL) to achieve a high positive linear correlation between the ground truth map and the predicted map. Its divided into 3 steps: Structure Comparison, Cross-Entropy weight coefficient determination, and mini-batch loss definition.

As part of Structure comparison, authors have calculated e-coefficient, which can measure the degree of linear correlation between ground truth and prediction:

e = $\left| \frac{y - \mu_y + C_4}{\sigma_y + C_4} - \frac{p - \mu_p + C_4}{\sigma_p + C_4} \right|$

Here, C_4 is stability factor set to be 0.01 as an empirical observed value. μ_y and σ_y is local mean and standard deviation of the ground truth y respectively. y locates at the center of the local region and p is the predicted probability.

After calculating the degree of correlation, authors have used it as coefficient for cross entropy loss function, defined as:

Using this formula, the loss function will abandon those predictions which have e¡somevalue. In simpler terms, Loss function will automatically abandon those pixel level predictions, which doesn't show correlation in terms of structure.

$f_{n,c} = 1 * e_{n,c} > \beta e_{max}$

Using this coefficient function, we can define SSL loss as:
$Loss_{ssl}(y_{n,c}, p_{n,c}) = e_{n,c} f_{n,c} L_{CE}(y_{n,c}, p_{n,c})$
and finally for mini-batch loss calculation, The SSL can be defined as:
$L_{ssl} = \frac{1}{M} \sum_{n=1}^{N} \sum_{c=1}^{C} L_{ssl}(y_{n,c}, p_{n,c})$ where, M is $\sum_{n=1}^{N} \sum_{c=1}^{C} f_{n,c}$

O. Log-Cosh Dice Loss

Dice Coefficient is a widely used metric to evaluate the segmentation output. It has also been modified to use as loss function, as it fulfill the mathematical representation of segmentation objective. But due to its non-convex nature, various times it fails to achieve the optimal results. Lovász-softmax loss aimed to tackle the problem of non-convex loss function by adding the smoothing using Lovász extension. Log-Cosh approach has been widely used in regression based problem to smoothen the curve.

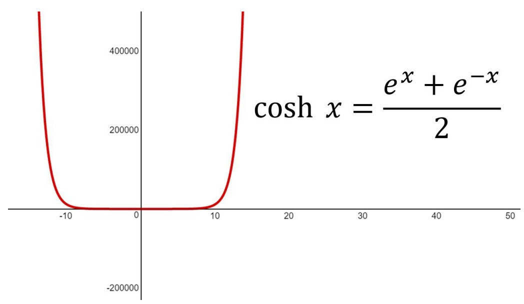

Fig. 3. Cosh(x) function is the average of e^x and e^-x

Hyperbolic functions have been used by deep learning community in terms of non-linearities such as tanh layer. They are tractable as well as easily differentiable. Cosh(x) is defined as:
$coshx = \frac{e^x + e^- x}{2}$, and $cosh'x = \frac{e^x - e^- x}{2} = sinhx$. but, at present $coshx$ range can go up to infinity. So, to capture it in range, log space is used, making the log-cosh function to be: $L(x) = log(coshx)$ and using chain rule and
$L'(x) = \frac{1}{coshx} sinhx$ Therefore, $L'(x) = tanhx$

TABLE II
COMPARISON OF SOME ABOVE MENTIONED LOSS FUNCTIONS ON BASIS OF DICE SCORES, SENSITIVITY AND SPECIFICITY FOR SKULL SEGMENTATION

Loss Functions	**Evaluation Metrics**		
	Dice Coefficient	***Sensitivity***	***Specificity***
Binary Cross-Entropy	0.968	0.976	0.998
Focal Loss	0.936	0.952	0.999
Dice Loss	0.970	0.981	0.998
Tversky Loss	0.965	0.979	0.996
Focal Tversky Loss	0.977	0.990	0.997
Log Cosh Dice Loss	0.975	0.975	0.997

Using this proof of concept that our loss will be continuous and in a defined range. We are proposing Log-Cosh Dice Loss function for its tractable nature, while encapsulating the features of dice coefficient. It can defined as:
$L_{lc-dce} = log(cosh(DiceLoss))$

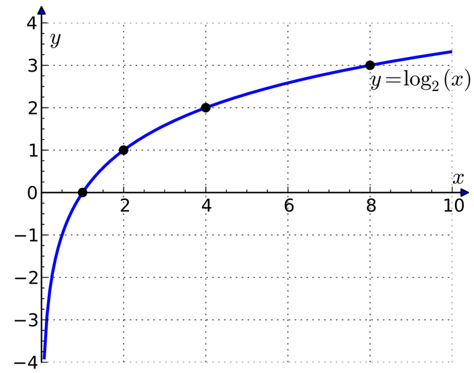

Fig. 4. Log(x) function with range from $(-\infty, C)$

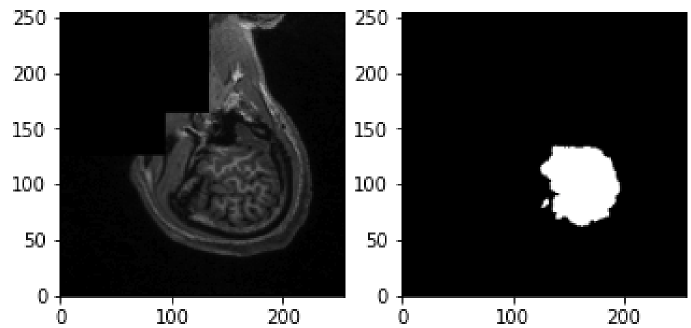

Fig. 5. Sample CT scan image from NBFS Skull Stripping Dataset

III. EXPERIMENTS

For experiments, we have implemented simple 2D U-Net model [1] architecture for segmentation with 10 convolution encoded layers and 8 decoded convolutional transpose layers. We have used NBFS Skull-stripping dataset, which consists of 125 2-D skull CT scans, and each scan consists of 100 slices (refer figure 5). For training, we have used batch size of 32 and adam optimizer with learning rate 0.001 and learning rate reduction up to 10^{-8}. After training the model for different loss function, we have evaluated them on basis of well known metrics: Dice Coefficient, Sensitivity, and Specificity (Ref table II).

IV. CONCLUSION

Loss functions plays an important role in determining the model performance. For complex objectives such as segmentation, it's not possible to determine a universal loss function. Majority of the times, it depends on the properties of data-set used for training, such as distribution, skewedness, boundaries, etc. It's not possible to conclude on a universal use-case loss function. However, we can say, that highly imbalanced segmentation works better with focus based loss functions. Similarly, binary-cross entropy works best with balanced data-sets, whereas mildly skewed data-sets can work around smoothed or generalized dice coefficient. In this paper, we have summarized 14 well known loss functions for semantic

TABLE III
TABULAR SUMMARY OF SEMANTIC SEGMENTATION LOSS FUNCTIONS

Loss Function	**Use cases**
Binary Cross-Entropy	Works best in equal data distribution among classes scenarios Bernoulli distribution based loss function
Weighted Cross-Entropy	Widely used with skewed dataset Weighs positive examples by β coefficient
Balanced Cross-Entropy	Similar to weighted-cross entropy, used widely with skewed dataset weighs both positive as well as negative examples by β and $1-\beta$ respectively
Focal Loss	works best with highly-imbalanced dataset down-weight the contribution of easy examples, enabling model to learn hard examples
Distance map derived loss penalty term	Variant of Cross-Entropy Used for hard-to-segment boundaries
Dice Loss	Inspired from Dice Coefficient, a metric to evaluate segmentation results. As Dice Coefficient is non-convex in nature, it has been modified to make it more tractable.
Sensitivity-Specificity Loss	Inspired from Sensitivity and Specificity metrics Used for cases where there is more focus on True Positives.
Tversky Loss	Variant of Dice Coefficient Add weight to False positives and False negatives.
Focal Tversky Loss	Variant of Tversky loss with focus on hard examples
Log-Cosh Dice Loss(ours)	Variant of Dice Loss and inspired regression log-cosh approach for smoothing Variations can be used for skewed dataset
Hausdorff Distance loss	Inspired by Hausdorff Distance metric used for evaluation of segmentation Loss tackle the non-convex nature of Distance metric by adding some variations
Shape aware loss	Variation of cross-entropy loss by adding a shape based coefficient used in cases of hard-to-segment boundaries.
Combo Loss	Combination of Dice Loss and Binary Cross-Entropy used for lightly class imbalanced by leveraging benefits of BCE and Dice Loss
Exponential Logarithmic Loss	Combined function of Dice Loss and Binary Cross-Entropy Focuses on less accurately predicted cases
Correlation Maximized Structural Similarity Loss	Focuses on Segmentation Structure. Used in cases of structural importance such as medical images.

segmentation and proposed a tractable variant of dice loss function for better and accurate optimization. In future, we will use this work as a baseline implementation for few-shot segmentation experiments.

REFERENCES

[1] Shruti Jadon, Owen P. Leary, Ian Pan, Tyler J. Harder, David W. Wright, Lisa H. Merck, and Derek L. Merck. A comparative study of 2D image segmentation algorithms for traumatic brain lesions using CT data from the ProTECTIII multicenter clinical trial. In Po-Hao Chen and Thomas M. Deserno, editors, *Medical Imaging 2020: Imaging Informatics for Healthcare, Research, and Applications*, volume 11318, pages 195 – 203. International Society for Optics and Photonics, SPIE, 2020.

[2] Ma Yi-de, Liu Qing, and Qian Zhi-Bai. Automated image segmentation using improved pcnn model based on cross-entropy. In *Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, 2004.*, pages 743–746. IEEE, 2004.

[3] Vasyl Pihur, Susmita Datta, and Somnath Datta. Weighted rank aggregation of cluster validation measures: a monte carlo cross-entropy approach. *Bioinformatics*, 23(13):1607–1615, 2007.

[4] Yaoshiang Ho and Samuel Wookey. The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling. *IEEE Access*, 8:4806–4813, 2019.

[5] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015.

[6] Shiwen Pan, Wei Zhang, Wanjun Zhang, Liang Xu, Guohua Fan, Jianping Gong, Bo Zhang, and Haibo Gu. Diagnostic model of coronary microvascular disease combined with full convolution deep network with balanced cross-entropy cost function. *IEEE Access*, 7:177997–178006, 2019.

[7] TY Lin, P Goyal, R Girshick, K He, and P Dollár. Focal loss for dense object detection. arxiv 2017. *arXiv preprint arXiv:1708.02002*, 2002.

[8] Carole H Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pages 240–248. Springer, 2017.

[9] Seyed Sadegh Mohseni Salehi, Deniz Erdogmus, and Ali Gholipour. Tversky loss function for image segmentation using 3d fully convolutional deep networks. In *International Workshop on Machine Learning in Medical Imaging*, pages 379–387. Springer, 2017.

[10] Nabila Abraham and Naimul Mefraz Khan. A novel focal tversky loss function with improved attention u-net for lesion segmentation. In *2019 IEEE 16th International Symposium on Biomedical Imaging (ISBI 2019)*, pages 683–687. IEEE, 2019.

[11] Seyed Raein Hashemi, Seyed Sadegh Mohseni Salehi, Deniz Erdogmus, Sanjay P Prabhu, Simon K Warfield, and Ali Gholipour. Asymmetric loss functions and deep densely-connected networks for highly-imbalanced medical image segmentation: Application to multiple sclerosis lesion detection. *IEEE Access*, 7:1721–1735, 2018.

[12] Zeeshan Hayder, Xuming He, and Mathieu Salzmann. Shape-aware instance segmentation. *arXiv preprint arXiv:1612.03129*, 2(5):7, 2016.

[13] Saeid Asgari Taghanaki, Yefeng Zheng, S Kevin Zhou, Bogdan Georgescu, Puneet Sharma, Daguang Xu, Dorin Comaniciu, and Ghassan Hamarneh. Combo loss: Handling input and output imbalance in multi-organ segmentation. *Computerized Medical Imaging and Graphics*, 75:24–33, 2019.

[14] Ken CL Wong, Mehdi Moradi, Hui Tang, and Tanveer Syeda-Mahmood. 3d segmentation with exponential logarithmic loss for highly unbalanced object sizes. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 612–619. Springer, 2018.

[15] Francesco Caliva, Claudia Iriondo, Alejandro Morales Martinez, Sharmila Majumdar, and Valentina Pedoia. Distance map loss penalty term for semantic segmentation. *arXiv preprint arXiv:1908.03679*, 2019.

[16] Javier Ribera, David Güera, Yuhao Chen, and Edward J. Delp. Weighted hausdorff distance: A loss function for object localization. *ArXiv*, abs/1806.07564, 2018.

[17] Davood Karimi and Septimiu E Salcudean. Reducing the hausdorff

distance in medical image segmentation with convolutional neural networks. *IEEE Transactions on medical imaging*, 39(2):499–513, 2019.

[18] Shuai Zhao, Boxi Wu, Wenqing Chu, Yao Hu, and Deng Cai. Correlation maximized structural similarity loss for semantic segmentation. *arXiv preprint arXiv:1910.08711*, 2019.