



Universidade Federal de Pernambuco
Centro de Informática
Programa de pós-graduação em Ciência da Computação

Fidel Alejandro Guerrero Peña

**Loss Function Modeling for Deep Neural Networks
Applied to Pixel-level Tasks**

Recife
2019

Fidel Alejandro Guerrero Peña

**Loss Function Modeling for Deep Neural Networks
Applied to Pixel-level Tasks**

Tese de Doutorado apresentada ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para obtenção do título de Doutor em Ciência da Computação.

Área de Concentração: Inteligência Computacional

Orientador: Prof. Dr. Tsang Ing Ren

Coorientador: Prof. Dr. Germano Crispim Vasconcelos

Recife

2019

Catálogo na fonte
Bibliotecária Mariana de Souza Alves CRB4-2105

G934l Guerrero Peña, Fidel Alejandro
Loss Function Modeling for Deep Neural Networks Applied to
Pixel-level Tasks – 2019.
111 f., fig., tab.

Orientador: Tsang Ing Ren.
Tese (Doutorado) – Universidade Federal de Pernambuco. CIn,
Ciência da Computação. Recife, 2019.
Inclui referências.

1. Inteligência Computacional. 2. Redes Convolucionais
Profundas. 3. Função de Perda. 4. Segmentação de Instâncias. I.
Ren, Tsang Ing (orientador). II. Título.

006.31

CDD (22. ed.)

UFPE-MEI 2019-170

Fidel Alejandro Guerrero Peña

**“Loss Function Modeling for Deep Neural Networks
Applied to Pixel-level Tasks”**

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Doutor em Ciência da Computação.

Aprovado em: 27/11/2019.

Orientador: Prof. Dr. Tsang Ing Ren

BANCA EXAMINADORA

Profa. Dra. Teresa Bernarda Ludermir
Centro de Informática / UFPE

Prof. Dr. George Darmiton da Cunha Cavalcanti
Centro de Informática / UFPE

Prof. Dr. Carlos Alexandre Barros de Mello
Centro de Informática / UFPE

Prof. Dr. Francisco Madeiro Bernardino Junior
Escola Politécnica de Pernambuco / UPE

Prof. Dr. Jorge de Jesus Gomes Leandro
Departamento de Pesquisa e Desenvolvimento / Motorola Mobility LLC

ACKNOWLEDGEMENTS

This work was possible thanks to my lovely wife Sara, who always encouraged me and kindly participated in some of the experiments. Thanks to my sister Elaine and parents Elina and Fidel for valuable discussions, advice, infinite love, and life in general. Thanks to my in-laws Alina and Carlos for all their support during this time. I want to acknowledge my advisor, Prof. Tsang Ing Ren, for giving me the opportunity to work with him on all the projects, for his guidance, and for teaching me how to be a better professional. Thanks to my advisor, Prof. Germano Crispim Vasconcelos, for accepting me as his student, for teaching me how to research and publish the results, his understanding, and encouragement. Thanks to Dr. Alexandre Cunha, Director of Cambia Lab at Caltech, for letting me learn from him and his help at all times. I would like to thanks Prof. Elliot Meyerowitz, Prof. Mary Yui, Prof. Ellen Rothenberg, and all the staff of the California Institute of Technology for allowing me to be part of their wonderful community. Not least, I would like to thanks the professors and staff of the Center for Informatics (UFPE) for welcoming me as one of their own. A special thanks to Daniel, Socorro, and Lilia for their help since my arrival in Brazil. Thanks to my co-authors for their comments, valuable discussion, and assistance throughout the project. Thanks to all my friends. This work was partially or fully financed by the Brazilian funding agency FACEPE, the research cooperation project between Motorola Mobility LLC (a Lenovo Company) and the Center for Informatics (UFPE), and Cambia Lab (CALTECH).

ABSTRACT

In recent years, deep convolutional neural networks has overcome several challenges in the field of Computer Vision and Image Processing. Particularly, pixel-level tasks such as image segmentation, restoration, generation, enhancement, and inpainting have shown significant improvements thanks to the advances of the technique. In general, training a neural network is similar to solving a complex optimization problem where the unknowns are the parameters of the model, and the goal is to transform vectors from the input domain to the output set. This optimization process can be seen as a directed search through an error surface where the optimal set of weights is the one that gives a minimal error over a data sample. Since reaching the global minimum is very difficult, the task is simplified to find an acceptable solution for the task. However, because of the high dimensionality of the solution space, the non-convexity of the error surface, and the presence of many flat regions and saddle points in the surface, the process of training a neural network is generally addressed by carefully tuning the hyperparameters of the model and annotating a vast training dataset. The three core components of the cost function used for supervised training are the architecture, the data, and the loss function. Despite the emergence of many new architectures, finding better networks to solve a task is difficult. The modeling of new loss functions is a more feasible approach to improve the optimization and, therefore, find better-performed models. This work proposes to use a given network, and concentrates on the designing of loss functions for pixel-level regression and pixel-level classification problems, namely, image segmentation, to improve results. The rationale behind proposed loss functions is that the incorporation of priors in the form of regularization terms helps to distinguish between similarly-performed models, like the ones found in flat regions. New pre-processing and post-processing techniques are also introduced in each case to assist in solving real-life problems. The applicability of pixel-level classification loss functions in instance segmentation task with full and weak supervision was studied using challenging biological image datasets with isolated and clustered cells for both 2D and 3D. A pixel-level regression loss function was applied to the multi-focus image fusion problem. Experimental results for instance segmentation and image restoration tasks suggest an improvement of the performance when compared to other competitive loss functions. 3D segmentation and multi-focus image fusion approaches showed low execution time.

Keywords: Deep Convolutional Neural Networks. Loss Function. Instance Segmentation. Multi-focus Image Fusion.

RESUMO

Nos últimos anos, os métodos baseados em redes convolucionais profundas superaram vários desafios no campo de Visão Computacional e Processamento de Imagens. Particularmente, tarefas em nível de pixel, como segmentação, restauração, geração, *inpainting*, e recuperação de informação em imagens, mostraram melhorias significativas graças ao avanço das redes neurais profundas. No geral, o treinamento de uma rede neural é o mesmo que resolver um problema de otimização complexo, onde as incógnitas são os parâmetros do modelo, e o objetivo é transformar vetores do domínio de entrada para o conjunto de saída. Esse processo de otimização pode ser visto como uma busca direcionada em uma superfície de erro, em que o conjunto ideal de pesos é aquele que gera um erro mínimo em uma amostra de dados. Dado que chegar ao mínimo global é muito difícil, a tarefa é simplificada a encontrar uma solução aceitável para uma tarefa dada. No entanto, devido à alta dimensionalidade do espaço da solução, a não-convexidade da superfície de erro, e a presença de muitas planícies, o processo de treinamento de uma rede neural é geralmente tratado por meio do ajuste cuidadoso dos hiperparâmetros do modelo e criando anotações de um amplo conjunto de dados de treinamento. As três componentes principais da função de custo usada no treinamento supervisionado são a arquitetura, os dados, e a função de perda. Apesar do surgimento de muitas novas arquiteturas, encontrar modelos com desempenho aceitável é muito difícil. A modelagem de funções de perda é uma abordagem mais efetiva para melhorar o processo de otimização e, por consequência, achar modelos com melhor desempenho. Este trabalho propõe-se a usar uma rede dada e concentra-se na proposição de funções de perda para problemas de regressão e classificação em nível de pixel, também conhecida como segmentação de imagem, visando a melhorar o desempenho. A lógica por trás das funções de perda propostas é que a incorporação de *priors* em forma de regularização ajuda a diferenciar modelos com desempenho semelhante. Novas técnicas de pré-processamento e pós-processamento também são propostas em cada caso para ajudar na solução de problemas reais. A aplicabilidade das funções de perda de classificação em nível de pixel na tarefa de segmentação de instância com supervisão completa e fraca foi estudada usando conjuntos de dados desafiadores de imagem biológica com células isoladas e agrupadas para 2D e 3D. A função de perda de regressão em nível de pixel foi aplicada ao problema de fusão de imagem com múltiplos focos. Os resultados da experimentação em tarefas de segmentação de instâncias e restauração de imagens sugerem uma melhoria do desempenho quando comparado com funções de perda semelhantes. Nas propostas de segmentação 3D e fusão de imagens com múltiplos focos, foi observado um baixo tempo de execução.

Palavras-chave: Redes Convolucionais Profundas. Função de Perda. Segmentação de Instâncias. Fusão de Imagens com Múltiplos Focos.

LIST OF FIGURES

Figure 1	– Computer Vision tasks taxonomy according to the dimension and number set of the output vector y . An example of CNN structure is given for image-level, pixel-level and region-level tasks.	19
Figure 2	– Thesis structure diagram. Contributions for pixel-level classification are divided into three chapters that increasingly address clustered cells, weakly supervision, and imbalanced data. Contributions for pixel-level regression are shown in Chapter 6.	24
Figure 3	– Example of (A) convex-like error surfaces and (B) loss landscape of a neural network with an initial θ^0 and optimal θ^* solution.	26
Figure 4	– Error surfaces for six different elements along with expected value of the loss.	28
Figure 5	– Overall scheme of the encoder-decoder type of network U-Net.	29
Figure 6	– Example of (A) a mask t and its (B) convex hull c_t , (C) skeleton s_t , and (D) distance transform ϕ_t . The mask contour and skeleton are shown in red in (B) and (C) for better visualization.	35
Figure 7	– Example of (A) a binary image h along with two initial markers represented in color red and blue respectively, (B) the topological surface to be flooded, and (C) the final segmentation segmentation.	36
Figure 8	– Example of cells marked by the <i>mTomato</i> fluorophore are shown in (A). Their corresponding signal of interest, <i>CD25</i> , which changes over time, is expressed in some cells (B). The goal is to segment individual cells, as shown in (C), and colocalize <i>CD25</i> to measure its concentration within each cell (D) and consequently count how many cells are active at any given time. In this illustration, the top two cells are fully active as reported by their high <i>CD25</i> content. Colored masks in (C) are for illustration purpose only. A typical cluttering of T-cells is presented on panel E.	40
Figure 9	– Example of distinct intensity and structural signatures of the three predominant regions: background (A), cell interior (B), in-between cells (C). The combined histogram curves for comparison is show in (D). This distinction led us to adopt a multiclass learning approach which helped resolve the narrow bright boundaries separating touching cells, as seen in (C).	41
Figure 10	– Example of (A) two classes ground truth, (B) cluster of cells after morphological closing, (C) touching region and (D) its morphological dilation, and (E) final three classes label augmentation.	42

Figure 11 – Example of computation of each term in Equation 3.2 for going from the semantic segmentation ground truth to the final DWM weight map. Color code is normalized to maximum weight value with red representing higher weights and blue small weights.	43
Figure 12 – Example of (A) cell contour Γ_t and concave complement contour Γ_r with (B) its respective skeletons s_t and s_r . The contour points importance $\omega_\tau^c(p)$, for $p \in \Gamma$, is also shown in (B). Finally, (C) copy padding and Gaussian smoothing is applied and (D) sum to the class balance weight $\omega^B + 1$. Color code is normalized to maximum weight value with red representing higher weights and blue small weights.	44
Figure 13 – An example of clustered cells is shown in (A). The weight maps, from left to right, are the (B) class balancing weight map ω^B , the (C) proposed distance transform based weight map ω^{DWM} , and the (D) proposed shape aware weight map ω^{SAW} . Color code is normalized to maximum weight value with red representing higher weights and blue small weights.	45
Figure 14 – Overall segmentation scheme with touching pixels assignments and thresholded maps approach.	46
Figure 15 – F1 scores for radii $\iota \in [1, 7]$ in (A) 1X, (B) 1.1X, (C) 2X, and (D) 4X field of view size for each model. F1 values were consistently better for SAW and DWM in most of the cases.	48
Figure 16 – Class weighted accuracy (blue) and Weighted Cross Entropy (red) of SAW network for every epoch are shown in (A). In (B) it is observed the models accuracy during training with outperforming rates of proposed DWM and SAW.	49
Figure 17 – Examples of segmentation contour obtained with UNET2 ($\gamma_1 = 0.50, \gamma_2 = 0.06$), UNET3 ($\gamma_1 = 0.40, \gamma_2 = 0.06$), FL ($\gamma_1 = 0.50, \gamma_2 = 0.16$), DWM ($\gamma_1 = 0.45, \gamma_2 = 0.06$), and SAW ($\gamma_1 = 0.50, \gamma_2 = 0.11$) and ground truth delineations for eight regions of two images. Results are for the best combination of γ_1 and γ_2 thresholds. Contour colors are merely used to illustrate the separation of individually segmented regions. . . .	50
Figure 18 – Example of image with its corresponding annotation and obtained probability map and segmentation using DWM. Probability map is show as an RGB image whitth background (red), cell (green) and touching (blue) classes.	50
Figure 19 – Instance segmentation of clustered cells of four images. Instances colors and green contour are merely used to illustrate the separation of individually segmented regions.	51

Figure 20 – Examples of (A) incomplete and (B) inaccurate annotations of training images, pointed by arrows above. The goal of weakly supervised methods is to be able to segment well under uncertainty and limited data as shown in the examples of a missing cell and slim part, respectively, in the right panels of (A) and (B).	54
Figure 21 – Example of transformation from instance g to semantic h segmentation ground truth.	55
Figure 22 – Example of contrast modulation around touching regions. In this figure, higher, $a = -1.0, -0.5$, and lower, $a = 0.5, 1.0$, contrast examples are shown. $a = 0$ gives the original image.	56
Figure 23 – Example of semantic classes and weights values for a row (black line) in h and $\omega_{\beta,\nu,\sigma}$ with $\beta = 30, \nu = 10$, and $\sigma = 3$. A semantic ground truth and its corresponding weight map are shown in top left and bottom left images. Points P and Q denotes background tip and touching region respectively.	57
Figure 24 – Overall segmentation scheme of the proposed Watershed-based approach.	58
Figure 25 – Example of wrong cell separation using Maximum A Posteriori post-processing because of the confusion in the probability map. Probability maps are showed as RGB images where the channels correspond with background (B), cell (C) and touching (T) classes respectively.	60
Figure 26 – Panoptic Quality (PQ) metric during training for Lovász-Softmax (LSMAX), Weighted Cross Entropy with class Balance Weight Map (BWM), UNET weight map, and Triplex Weight Map (W^3) methods using (A) Maximum A Posteriori (MAP), (B) Thresholded Maps (TH), and (C) Watershed Transform (WT) post-processing.	61
Figure 27 – Example of segmentation results with Lovász-Softmax (LSMAX), Mask R-CNN (MRCNN), Weighted Cross Entropy with class Balance Weight Map (BWM), UNET weight map, Triplex Weight Map (W^3) and average combination (COMB). Instances colors and green contour are merely used to illustrate the separation of individually segmented regions.	61
Figure 28 – Zero-shot panoptic segmentation for meristem and sepal images with W^3 approach trained for T-cells images. Instances colors and green contour are merely used to illustrate the separation of individually segmented regions.	62

Figure 29 – Example of ideal probability map with its corresponding annotation and obtained probability maps and segmentation by SAW and proposed W^3 loss function. Instances colors and green contour are merely used to illustrate the separation of individually segmented regions. Probability map is show as an RGB image with background (red), cell (green) and touching (blue) classes.	63
Figure 30 – Example of confusion reduction with W^3 when compared with LSMAX, UNET, and BWM. Probability map is show as an RGB image with background (red), cell (green) and touching (blue) classes.	63
Figure 31 – Weakly supervised biomedical image instance segmentation of four images.	64
Figure 32 – Example of (A) near touching cells image with its respective weak annotation, and the obtained (B) three and (C) four classes output probability map and instance segmentation. The exclusion of dimmed, unseen cells during annotation is not intentional. Image is enhanced only for better visualization.	68
Figure 33 – Performance of classifiers C1 and C3 (BOUGHORBEL; JARRAY; EL-ANBARI, 2017) measured by Youden (J), Matthews Correlation Coefficient (MCC), Jaccard, F1, Tversky, and Accuracy scores for different imbalance ratios π . Youden and MCC are the only ones invariant to all imbalance ratios.	69
Figure 34 – Correlation between values of MCC and J for different imbalance ratios π . The linear correlation was measured using Pearson Correlation Coefficient with values of 0.92 for $\pi = 0.01$, 0.99 for $\pi = 0.25$, and approximately 1.0 for $\pi = 0.5$	70
Figure 35 – Simulations performed for analyzing the behavior of Cross Entropy and Youden index-based loss functions. During the first iterations, the segmentation is shrunk until it fits the ground truth, with a slow increase of the probabilities for each class. After this point, the probabilities are gradually increased for all classes at each time step resulting in CE values near to zero but higher values of J	72

Figure 36 – Segmentation results for Hela cells (A), Hela <i>nuclei</i> (B*), T-cells (C), meristem cells (a YZ-slice of the 3D segmented stack is shown) (D), Drosophila cells (E*), and sepal cells (z projection) (F*) images using networks trained with $J3$ and $J4$ loss functions. Probability maps are shown as RGB images with background (red), cell (green), and touching (blue) classes. For $J4$, the proximity prediction is shown in white. Asterisks (*) indicate zero-shot instance segmentations with networks trained exclusively over T-cells (C). Colors are to show cell separation. Original images were enhanced to help visualization. Whites arrows and circles are used to indicate some differences between $J3$ and $J4$. . .	73
Figure 37 – Loss landscape visualization around a fixed optimizer of Weighted Cross Entropy with class balance \mathcal{L}_{WCE} , proposed Triplex weight map \mathcal{L}_{W^3} , and introduced Cross Entropy with Youden-based regularization $\mathcal{L}_{CE} + \mathcal{L}_J$	74
Figure 38 – Example of 3D segmentation using the proposed $J4$ loss function. Original and enhanced versions (left column) of a meristem portion image stack and their respective segmentations (two views on the middle and right columns). Instances colors and black contour are merely used to illustrate the separation of individually segmented regions. Colors are randomly assigned for every cell.	77
Figure 39 – Examples of 2D weakly supervised biomedical image instance segmentation with $J3$, $J4$, and Dice regularizations. Instances colors and green contour are merely used to illustrate the separation of individually segmented regions.	78
Figure 40 – 3D weakly supervised instance segmentation with $J4$ over two views of the meristem volumen. Instances colors and black contour are merely used to illustrate the separation of individually segmented regions. Colors are randomly assigned for every cell.	79
Figure 41 – Example of different focus source images and the all-in-focus resulting image. The sources A and B represent the same image in different focal planes.	82
Figure 42 – Overall method scheme for a multi-focus fusion of an input burst. The images within the burst are incrementally fused through the n -th functional power f^n	83
Figure 43 – Example of synthetic tuple \mathbf{x} created by applying the MFIF dataset using MS COCO image y and its segmentation mask g . The focus map g^b was created using two classes as background and the other three objects as foreground. The blurred image \bar{y} and resulting sources (x_A, x_B) are shown in the second row.	85

Figure 44 – Multiple sources hourglass networks for multi-focus image fusion. Sources are showed separated in the figures but the input block is 6 channels depth map. For HF-Reg the output layer corresponds to the all-in-focus regressed image. In the case of HF-Seg, the output layer is a 2-channel feature map, and values $z_i(p)$ represents the probability of selecting pixel p from input source i	86
Figure 45 – Distances mapping for L1, L2, NPS6, NPS8 and NPS10 dissimilarity functions.	88
Figure 46 – Example of fusion results for tuples with normal and reversed order. In the first row are shown the frames within the tuples. In second and third row are shown the fusion results with the regression and segmentation networks respectively for both normal and reverse order evaluations. . .	90
Figure 47 – Example of the values of the fusion metrics for HF-Reg without and with Near post-processing and two dummy methods that returns the first (Dummy A) and second (Dummy B) image of the tuple as a result for the fusion.	91
Figure 48 – Example of the multi-focus image fusion obtained with intermediate L1, L2 and HF-Reg networks during the training.	92
Figure 49 – Logarithm of the error over the synthetic multi-focus test dataset. . . .	93
Figure 50 – Box plot for SSIM reference metric over the synthetic multi-focus test dataset.	93
Figure 51 – Example of synthetic test example, most methods have (B) higher values in objective assessment metrics. However, with a visual inspection (A) it can be observed that the proposed methods show a better quality fusion.	94
Figure 52 – Example of fusion results with HF-Reg and HF-Seg over the Lytro two sources real dataset.	95
Figure 53 – Example of fusion results with different literature methods and the proposed HF-Reg and HF-Seg methods over the "golf image" of the Lytro 2 dataset.	96
Figure 54 – Example of fusion results over the Lytro three sources real dataset. . .	98
Figure 55 – Example of multi-focus image fusion and filtering of noisy sources. . . .	99

LIST OF TABLES

Table 1 – Output tensor size, convolution kernels shapes and trainable parameters for each layer in U-Net network assuming an RGB input image of 1024×1024	30
Table 2 – F1 scores for different contour uncertainty <i>radii</i> . The method SAW performed better than others, with DWM the second best on training data.	47
Table 3 – Detection metrics for Jaccard Index above 0.5 is much pronounced for SAW meaning it can detect more cells than the other methods.	49
Table 4 – Results for different post-processing methods where TH and WT represent the best thresholds combination for Threshold and Watershed post-processing respectively.	59
Table 5 – Best thresholds combinations for TH and WT in each method.	60
Table 6 – Performance comparison of U-Net trained over T-cell dataset using Weighted Cross Entropy with class Balance (BWM), Cross Entropy with Dice regularization (DSC), Weighted Cross Entropy with Triplex weight map (W^3), and Youden based regularization over three ($J3$) and four ($J4$) classes.	75
Table 7 – Results obtained over different datasets show the benefits of using the additional gap class. In all cases a higher PQ value is obtained for $J4$. A (*) indicates zero-shot segmentation.	76
Table 8 – Mean and standard deviation of the objective assessment over the synthetic multi-focus test dataset.	94
Table 9 – Mean and standard deviation of the objective assessment over the Lytro multi-focus two sources dataset.	95
Table 10 – Execution time for each mfif method with three different image size. Time unit is second.	99

LIST OF ABBREVIATIONS AND ACRONYMS

AI	Artificial Intelligence
ANN	Artificial Neural Networks
BCE	Binary Cross Entropy
BWM	Balance Weight Map
CE	Cross Entropy
CNN	Convolutional Neural Networks
COMB	Average Combination
CV	Computer Vision
DCT	Discrete Cosine Transform
DCT+CV	Discrete Cosine Transform with Consistency Verification
DL	Deep Learning
DSC	Dice regularization
DWM	Distance transform based Weight Map
FCN	Fully Convolutional Network
FL	Focal Loss function
FNR	False Negative Rates
FPR	False Positive Rates
GFF	Guided Filtering Fusion
HF-Reg	Hourglass Fusion Regression network
HF-Seg	Hourglass Fusion Segmentation network
IM	Image Matting
IoU	Intersection over Union
L1	L1 norm
L2	L2 norm
LSMAX	Lovász-Softmax loss function
MAP	Maximum A Posteriori
MCC	Matthews Correlation Coefficient
MFIF	Multi-focus Image Fusion
MRCNN	Mask R-CNN
NPS	Normalized Positive Sigmoid

P05	Detection Precision of instances with Jaccard index above 0.5
PQ	Panoptic Quality
RQ	Recognition Quality
SAW	Shape aware Weight Map
SGD	Stochastic Gradient Descent
SQ	Segmentation Quality
SSIM	Structural Similarity
TH	Thresholded maps post-processing
TNR	True Negatives Rates
TPR	True Positives Rates
U-NET	U shaped Network
UNET	Near objects weight map proposed in U-Net paper
UNET2	Binary Cross Entropy loss function with UNET weight map
UNET3	Weighted Cross Entropy loss function with UNET weight map over three classes
VGG	Visual Geometry Group network
W³	Triplex weight map
WCE	Weighted Cross Entropy
WT	Watershed Transform post-processing

CONTENTS

1	INTRODUCTION	18
1.1	MOTIVATION	20
1.2	OBJECTIVES	21
1.3	CONTRIBUTIONS	22
1.4	THESIS STRUCTURE	23
2	BACKGROUND	25
2.1	DEEP NEURAL NETWORKS TRAINING	25
2.2	PIXEL-LEVEL ARCHITECTURE	28
2.3	LOSS FUNCTIONS	31
2.4	MATHEMATICAL MORPHOLOGY	35
2.5	INSTANCE SEGMENTATION	36
2.6	MULTI-FOCUS IMAGE FUSION	37
3	MULTICLASS WEIGHTED LOSS FOR INSTANCE SEG- MENTATION OF CLUSTERED CELLS	39
3.1	INTRODUCTION	39
3.2	MULTICLASS SHAPE-BASED WEIGHTED CROSS ENTROPY LOSS FUNCTIONS	40
3.2.1	Class Augmentation	41
3.2.2	Focus Weights Map	42
3.2.3	Assignment of Touching Pixels	45
3.3	EXPERIMENTS AND RESULTS	46
3.4	CONCLUSIONS	52
4	A WEAKLY SUPERVISED METHOD FOR INSTANCE SEG- MENTATION OF BIOLOGICAL CELLS	53
4.1	INTRODUCTION	53
4.2	MULTICLASS SHAPE-BASED WEAKLY SUPERVISED LOSS FUNC- TION	54
4.2.1	From Instance to Semantic Ground Truth	54
4.2.2	Touching Region Augmentation	55
4.2.3	Robust Weight Maps for Weak Annotations	55
4.2.4	From Semantic to Instance Segmentation	57
4.3	EXPERIMENTS AND RESULTS	58
4.4	CONCLUSIONS	65

5	J REGULARIZATION IMPROVES IMBALANCED MULTI-CLASS SEGMENTATION	66
5.1	INTRODUCTION	66
5.2	IMBALANCED MULTICLASS WEAKLY SUPERVISED LOSS FUNCTION	67
5.2.1	Gap Class	67
5.2.2	J Regularization	68
5.2.3	Gap Output Assignment	71
5.3	EXPERIMENTS AND RESULTS	72
5.3.1	Loss Landscape Visualization	74
5.3.2	Instances Segmentation Performance	75
5.4	CONCLUSIONS	80
6	A MULTIPLE SOURCE HOURGLASS DEEP NETWORK FOR MULTI-FOCUS IMAGE FUSION	81
6.1	INTRODUCTION	81
6.2	MULTI-FOCUS IMAGE FUSION LEARNING	82
6.2.1	Multi-focus Image Fusion Dataset	83
6.2.2	Multiple Sources Hourglass Network	84
6.2.3	Implementation Details	87
6.3	EXPERIMENTS AND RESULTS	88
6.3.1	Commutativity	89
6.3.2	Multi-focus Image Fusion Metrics	89
6.3.3	Loss Function - L1 vs L2 vs NPS	91
6.3.4	Two Source Synthetic Dataset	93
6.3.5	Two Sources Real Dataset	94
6.3.6	Three Sources Real Dataset	97
6.3.7	Execution Time	97
6.3.8	Applications of HF-Reg	97
6.4	CONCLUSIONS	99
7	CONCLUSIONS	100
7.1	LIMITATIONS	101
7.2	FUTURE WORKS	102
	REFERENCES	103

1 INTRODUCTION

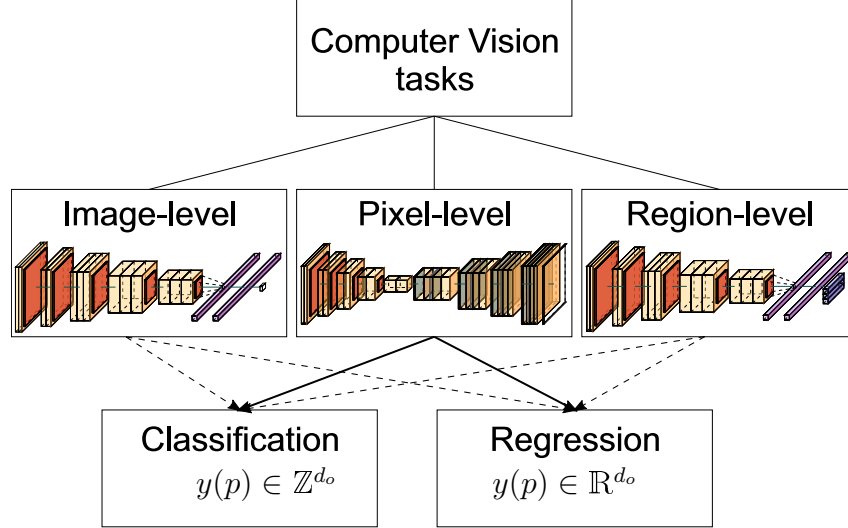
Artificial Intelligence (AI) is an area of Computer Science that has as a goal the development of techniques for solving tasks simulating human intelligence. The term was proposed by John McCarthy in 1955 (MCCARTHY et al., 2006) and it includes several topics such as Natural Language Processing, Computer Vision, and decision making. Different kinds of disciplines can be found within AI solutions, such as Rule-based systems, Machine Learning, Representation Learning, and Deep Learning (DL) that is a sub-field of Artificial Neural Networks (ANN) area (GOODFELLOW; BENGIO; COURVILLE, 2016). This last discipline, in particular, has proven capable of solving complex real-world tasks that were considered very difficult for computers few years before. The technique is a subset of Machine Learning that uses ANN to learn to transform objects from an input domain to an output set.

Among the different fields of research under the Deep Learning category, Computer Vision (CV) has seen great improvements with the recent advances in a particular type of ANN known as Convolutional Neural Networks (CNN). In this field, the input element for the method is an image or sequence of images. An image can be formally expressed as a mapping $x: \Omega \rightarrow \mathbb{R}^{d_c}$, where $\Omega \subset \mathbb{R}^{d_i}$ is a regular grid. The elements $p \in \Omega$ are called pixels in the case $d_i = 2$ and voxels when $d_i = 3$. For d_c taking value of 1, the image is said to be in gray-levels. Color images are represented by using a color model where d_c could take values of 3 or 4.

According to the shape of the output space, CV tasks can be grouped into three sub-categories: image-level, pixel-level, and region-level (SUN et al., 2018). Examples of image-level and region-level problems are image classification and object detection, respectively. However, because the number of tasks in Computer Vision is so immense, this work focus only on pixel-level category, where input and output mappings are defined over the same grid Ω . This means that the width and height of the expected output is the same as the input image, but the number of channels might be different. According to the number set in which the output mapping is defined, a task can be further grouped as classification or regression, *e.g.*, integer or real-valued outputs (Figure 1). The pixel-level category includes tasks as image segmentation, restoration, generation, enhancement, and inpainting, among others. This study pivots its attention in two problems within pixel-level classification and regression: instance segmentation and multi-focus image fusion.

For solving such kind of tasks, Deep Learning-based models learn to transform inputs to outputs given a training dataset of examples, $S = \{(x, y) | x \in \mathbb{I} \text{ and } y \in \mathbb{O}\}$. In general, the neural networks used for learning the mapping are just complex function $f_\theta: \mathbb{I} \rightarrow \mathbb{O}$, consisting of millions of parameters θ . Here, the notation $f_\theta(x)$, also written as $f(x; \theta)$, is used to refer to a specific function within the family of functions $\{f_{\theta_i}\}$ for a given

Figure 1 – Computer Vision tasks taxonomy according to the dimension and number set of the output vector y . An example of CNN structure is given for image-level, pixel-level and region-level tasks.



Source: The author (2019)

representation of the target function f . A specific function $f_\theta \in \{f_{\theta_i}\}$ is known as a model. The sets \mathbb{I} and \mathbb{O} refer to the domain and image of the function and varies according to the task to be solved. In terms of optimization, training a network means finding a good combination of parameters θ^* such that the predictions for inputs x within the training set match the expected output. The accuracy of a model to solve a task directly depends on the values of the parameters vector.

Finding appropriate values for θ relies on the specification of a cost function. The cost of a model in supervised learning is commonly defined as

$$\mathcal{C}(\theta) = \frac{1}{|\mathfrak{M}|} \sum_{(x_i, y_i) \in \mathfrak{M}} \mathcal{L}(y_i, f_\theta(x_i)), \quad (1.1)$$

where $\mathfrak{M} \subseteq S$ (GOODFELLOW; BENGIO; COURVILLE, 2016). The optimal value for θ is then obtained by solving the optimization problem $\theta^* = \arg \min_{\theta \in \Theta} \mathcal{C}(\theta)$, which tries to minimize the value of the cost. In the end, the entire optimization can be seen as a guided search over an error surface where the height of the landscape corresponds with the value of the cost \mathcal{C} . As can be seen in the equation above, the morphology of the error surface depends implicitly on three key components: the data \mathfrak{M} , the representation of the target function f , and the metric \mathcal{L} . The last one is known as the loss function $\mathcal{L}(y, \hat{y})$ and is the one in charge of measuring how well the model predicts the output of a given data sample.

Because of the extremely high dimension of the parameter space and the non-convex nature of the error surface (DAUPHIN et al., 2014; CHOROMANSKA et al., 2015), the global optima θ^* is not reached after the training finishes. Then, the ultimate goal when training a deep neural network is simplified to find an acceptable solution. Obtaining such

good models is very related to the proper definition of each component in Equation 1.1. Modifications of the data, for a given task, is not always possible because this usually corresponds to real-acquired inputs. Nevertheless, obtaining alternative output representations \bar{y} can still lead to the discovery of better-performed models (CHEN et al., 2016). On the other hand, manual design of new architectures is one of the most common practices in current DL research (MINAR; NAHER, 2018). However, because of the hierarchical shape and complexity of existing architectures, changes in network topology has unknown effects over the error surface. This makes the entire process of manually looking for an architecture that leads to improvements in the performance very challenging and mostly based on extensive trials and errors. In this regard, community opinion is divided about the scientific practice and rigor of the technique, sometimes referring to the approach as a modern form of alchemy (HUDSON, 2018).

1.1 MOTIVATION

The difficulties of manually proposing new architectures usually lead to deepest and over-parameterized networks, *e.g.*, there are more parameters than needed to solve the problem. This over-parameterization could lead to memorization of the training samples, resulting in poor generalization. Then, to avoid overfitting, the training of this type of networks is usually addressed by configuring, carefully testing and adjusting the hyperparameters (SERRE, 2019). Despite the proven results, this is a time-consuming and resource-intensive approach. This work goes in the opposite direction by fixing an architecture for all tasks and assuming that there is a subset of models within $\{f_{\theta_i}\}$ that leads to acceptable solutions. Then, the efforts are concentrated in finding a model such that the performance of f for the task is improved. A feasible approach to do this is by modeling the loss function, which intuitively defines the morphology of the error surface, $\mathcal{C}(\theta) = \mathbb{E}[\mathcal{L}(\cdot)]$. Finding better-performed models is very related with the morphology of the error surface because the performance of a model is indirectly improved by minimizing the cost. However, some of the most well-adopted loss functions for pixel-level classification and regression does not constitute the best surrogate for the measurement of interest (GOODFELLOW; BENGIO; COURVILLE, 2016). In particular, some of the loss functions used in segmentation problems are very unstable and unable to obtain good generalization in the presence of class imbalance and under-represented regions. Additionally, besides the Mean Square Error (L2) loss function be the most used loss for pixel-level regression problems (ZHAO et al., 2016), it has a very small penalization for small errors leading to the appearance of many artifacts or long training times. Then, the modeling of new loss functions aiming to overcome previously related problems is of great importance for obtaining better-performed Deep Learning systems. Especially, the study of the effects in the performance when using different loss functions is interesting for helping in the understanding of the training process. Furthermore, its applicability is

independent of the pixel-level architecture used to solve the task, *e.g.*, the performance of a very well-performed architecture could be increased even more by modeling a better loss function.

At the same time, instance segmentation task is of high interest for biomedical image community. For example, in developmental cell biology studies, signals of interest needs to be quantified on a per cell basis. This requires segmenting every cell in many images, accounting to hundreds or thousands of cells per experiment. Despite the success of current deep learning solutions, the challenge begins when clustered cells, weak annotation and imbalanced classes are accounted in the task. Complex architectures (XU et al., 2017; HE et al., 2017) try to overcome some of these situations but mostly relying on the assumption of correct annotated data. In this work each challenge is addressed by proposing a different loss functions, but always maintaining robustness to the previous one, *e.g.*, first clustered cell separation is performed, later clustered cell separation and weak supervision, and lastly the three challenges are aimed in the same loss.

On the other hand, the study of multi-focus image fusion is also of great relevance for image processing field. The applications include medical and biological imaging, video surveillance and digital photography. The lack of a supervised dataset for the task, as well as the challenge for identifying the focus map and proper fusion rule makes the problem very interesting to investigate. Overall, a fast and accurate fusion for images with high resolution is important for the practical use of such solution in current mobile devices.

1.2 OBJECTIVES

This work aims to propose new loss functions on two chosen pixel-level classification and regression tasks, using a fixed Convolutional Neural Network architecture.

The specific objectives of this research are:

- To model and implement new segmentation loss functions for clustered cell separation.
- To model and implement new segmentation loss functions considering weak annotation.
- To model and implement new segmentation loss functions addressing highly imbalanced classes.
- To model and implement new pixel-level regression loss functions for multi-focus image fusion.

To achieve these objectives the study of the literature regarding loss functions designing for pixel-level regression and classification tasks, as well as the most used architectures for instance segmentation and multi-focus image fusion is performed. After modeling and

implementing the proposals their validity is evaluated and compared with different loss functions from the literature.

1.3 CONTRIBUTIONS

The contributions of this work can be summarized as follows:

- The proposal of five new loss functions for pixel-level tasks. The first two proposals comprises fully supervised instance segmentation for clustered cells. Then, a weakly supervised generalization is proposed using a robust shape aware weight map. A Youden based regularization term was then introduced for accounting high imbalance in pixel-level classification. Finally, a new loss function with bound regularization was proposed for pixel-level regression.
- The proposal of a multi-class semantic segmentation framework for instance segmentation. Instance segmentation is cast as a semantic segmentation by creating a new touching class to enforce separation of clustered cells. Later, a fourth class is introduced in regions between near non-touching cells for obtaining better contour adequacy. This approaches are combined with a new touching region contrast modulation for learning separation with scarce data.
- The proposal of a multi-class Thresholded maps and Watershed-based post-processing for instance segmentation problems. The methods are introduced for improving separation between adjoining cells with uncertain touching classifications.
- The proposal of a bi-variable U-Net architecture for multi-source pixel-level tasks. Supervised training is attained by using a new synthetic data creation for multi-focus image fusion.

The publications which comprise this thesis are listed below:

- **Multi-class Weighted Loss for Instance Segmentation of Cluttered Cells.** Fidel A. Guerrero Peña; Pedro Marrero Fernandez; Tsang Ing Ren; Mary Yui; Ellen Rothenberg; Alexandre Cunha. *In: Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2018. (PEÑA et al., 2018)
- **Instance Segmentation of Biological Cells under Weakly Supervised Conditions.** Fidel A. Guerrero Peña; Pedro Marrero Fernandez; Tsang Ing Ren; Alexandre Cunha. *In: South California Machine Learning Symposium (SOCALML)*, 2019.
- **Weakly Supervised Instance Segmentation of Biological Cells.** Fidel A. Guerrero Peña; Pedro Marrero Fernandez; Tsang Ing Ren; Alexandre Cunha. *In: Workshop of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI-MIL)*, 2019. (PEÑA et al., 2019b)

- ***J* Regularization Improves Imbalanced Multiclass Segmentation.** Fidel A. Guerrero Peña; Pedro Marrero Fernandez; Paul Tarr; Tsang Ing Ren; Elliot Meyerowitz; Alexandre Cunha. Available at <<https://arxiv.org/abs/1910.09783>>, 2019. (PEÑA et al., 2019)
- **Burst Ranking for Blind Multi-Image Deblurring.** Fidel A. Guerrero Peña; Pedro Marrero Fernández; Tsang Ing Ren; Jorge J.G. Leandro; Ricardo Nishihara. In: *IEEE Transactions on Image Processing (TIP)*, 2020. (PEÑA et al., 2020)
- **A Multiple Source Hourglass Deep Network for Multi-focus Image Fusion.** Fidel A. Guerrero Peña; Pedro Marrero Fernandez; Tsang Ing Ren; Germano Crispim Vasconcelos; Alexandre Cunha. Available at <<https://arxiv.org/abs/1908.10945>>, 2019. (PEÑA et al., 2019a)

1.4 THESIS STRUCTURE

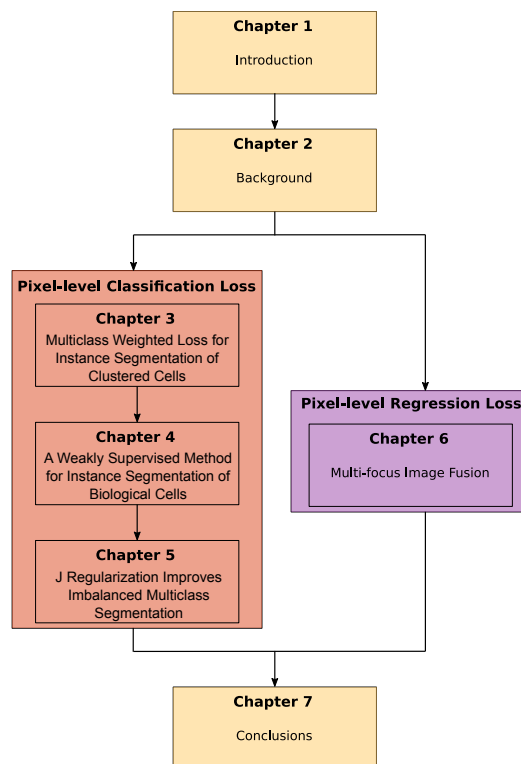
Besides this Introduction chapter, this thesis is divided into six additional chapters. Chapters from 3 to 6 are associated with either a submitted or published paper. Chapters 3, 4, and 5 aims to solve instance segmentation task but incrementally addressing a new challenge at each chapter. Chapter 6 aims to solve multi-focus image fusion task. Figure 2 shows the diagram of the organization of the thesis. The remaining chapters are organized as follows:

- Chapter 2: In this chapter the main concepts of deep neural networks training, the selected architecture, and a review of the literature on loss functions designing for pixel-level regression and classification tasks are presented. The most relevant works for biomedical instance segmentation, and multi-focus image fusion, are also presented.
- Chapter 3: This chapter introduces a new touching-based data modification and the modeling of two new loss functions for fully supervised biomedical image segmentation. The results and comparison with classical segmentation loss functions are also shown. The method address the challenge of separating clustered cells. The content of this chapter has been published in the IEEE International Conference on Image Processing (ICIP), 2018. (PEÑA et al., 2018)
- Chapter 4: This chapter introduces a new touching contrast modulation and a new loss function for weakly supervised biomedical images instance segmentation. The results and comparison with other loss functions from the literature for instance and zero-shot segmentation are also presented. The method address separation of clustered cells and weak supervision. The content of this chapter has been pub-

lished in Medical Image Computing and Computer-Assisted Intervention Workshop (MICCAI-MIL), 2019. (PEÑA et al., 2019b)

- Chapter 5: This chapter introduces a new proximity class augmentation and a new Youden based regularization for Cross Entropy loss function. Evaluation over 2D and 3D datasets, as well as a comparison with different loss functions are presented. The method address separation of clustered cells, weak supervision, and highly imbalanced classes. The content of this chapter is available at <<https://arxiv.org/abs/1910.09783>>. (PEÑA et al., 2019)
- Chapter 6: In this chapter a new regression loss function and a multi-source architecture for multi-focus image fusion tasks are proposed. The results under different conditions are shown and compared with literature methods. The content of this chapter is available at <<https://arxiv.org/abs/1908.10945>>. (PEÑA et al., 2019a)
- Chapter 7: In this chapter the conclusions of the thesis, limitations, and future works are presented.

Figure 2 – Thesis structure diagram. Contributions for pixel-level classification are divided into three chapters that increasingly address clustered cells, weakly supervision, and imbalanced data. Contributions for pixel-level regression are shown in Chapter 6.



Source: The author (2019)

2 BACKGROUND

This chapter presents a quick review of deep neural networks training and the U-Net architecture. Loss functions for pixel-level classification and regression, as well as a few key concepts from mathematical morphology used in the rest of this thesis are described. Most well-known methods for instance segmentation and multi-focus image fusion are also presented.

2.1 DEEP NEURAL NETWORKS TRAINING

The traditional way to introduce newcomers to the neural networks field is through a connectionist approach (HAYKIN, 2009). In this kind of definition, the focus is on network topology and operation within the computation units. However, as discussed before, effectively training a network is a very complicated process that depends on more than the architecture. In fact, the learning process is the same as solving a complex optimization problem where the value of the cost \mathcal{C} is minimized.

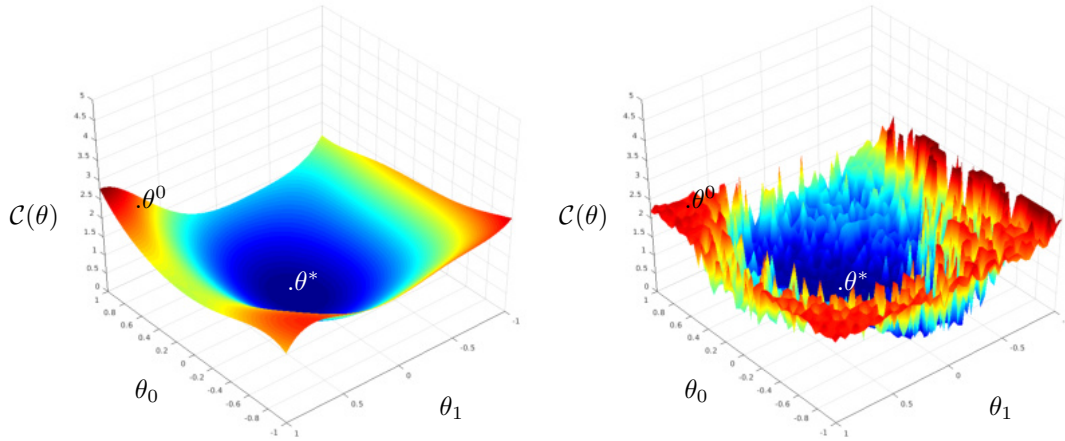
Minimizing the cost \mathcal{C} involves defining a loss function \mathcal{L} such that small values are obtained whenever a good fit between y and \hat{y} is attained. Therefore, the value of the cost serves as a surrogate measurement of model performance on a given training dataset. Let f be a network that has only two parameters, *e.g.*, $\theta \in \mathbb{R}^2$. An ideal convex-like error surface looks like in Figure 3A where the ordinate and abscissa axes represent the values for θ_0 and θ_1 respectively, being $\theta = (\theta_0, \theta_1)$. Here, the height of the landscape corresponds to the value of the cost \mathcal{C} for a particular θ . A minimization problem over an error surface is the same as finding an optimal combination of weights where the minimum value \mathcal{C} is attained. The point θ^* in the figure has the smallest value of \mathcal{C} in the error surface, and it is known as a minimizer for \mathcal{C} (GOODFELLOW; BENGIO; COURVILLE, 2016). Then, the challenges of effectively training a neural network are very related to the characteristics of the explored error surface.

In practice, loss landscapes of deep neural networks are more similar to the one shown in Figure 3B. The main features of these error surfaces are the presence of many local minima, saddle points, and a high dimensionality (BRAY; DEAN, 2007; DAUPHIN et al., 2014; CHOROMANSKA et al., 2015). Thus, the harder it is to navigate the error surface, the more difficult it gets to train a neural network. In the end, the height for every point within the error surface is the same as the expected value of the loss function $\mathbb{E}[\mathcal{L}(.)]$ over a given dataset. This means that the loss function has a direct influence in the landscape morphology and characteristics, and, therefore, a core role when training a neural network.

Because the error surface is entirely unknown, a random initialization of the weights θ^0 is a common approach for beginning the search with. However, special care must be taken

when choosing initialization distribution. The problems are very related to exploding and vanishing gradients (GLOROT; BENGIO, 2010) when setting either too large or too small values for the weights. Despite that several methods (GLOROT; BENGIO, 2010; HE et al., 2015) have been proposed for initializing the parameters, usually Glorot initialization (GLOROT; BENGIO, 2010) works well in most cases.

Figure 3 – Example of (A) convex-like error surfaces and (B) loss landscape of a neural network with an initial θ^0 and optimal θ^* solution.



A - convex-like error surface

B - loss landscape of neural network

Source: The author (2019)

The directed search algorithm that allows finding a minimizer θ^* given an initial solution θ^0 is called Gradient Descent (CAUCHY, 1847). This is a first-order iterative optimization algorithm, and the basic idea is to move over the surface by looking at each instant for the direction where the expected value of the loss is minimized. Formally, the direction is obtained as the exact opposite to the gradient vector,

$$\nabla \mathcal{C}(\theta) = \begin{bmatrix} \frac{\partial \mathcal{C}(\theta)}{\partial \theta_0} \\ \vdots \\ \frac{\partial \mathcal{C}(\theta)}{\partial \theta_m} \end{bmatrix}, \quad (2.1)$$

being the weights update rule expressed as

$$\theta^{n+1} = \theta^n - \gamma \cdot \nabla \mathcal{C}(\theta^n) = \theta^n - \gamma \cdot \sum_{(x_i, y_i) \in S} \nabla \mathcal{L}(y_i, f(x_i; \theta^n)), \quad (2.2)$$

where θ^n is the weights vector in the n -th iteration of the algorithm, and γ represents the velocity of the movement over the error surface towards the negative gradient direction, also known as the learning rate. Setting either too high or too small values for the learning rate leads to bouncing or too long optimizations respectively. The values are usually set

between $\gamma \in (10^{-6}, 1)$ (BENGIO, 2012) depending on the task to be solved. Adaptive strategies (HINTON, 2012; TIELEMAN; HINTON, 2012; KINGMA; BA, 2015) are also common to dynamically change the value of the learning rate according to the morphology of the surface. The Adaptive Moment estimation strategy (KINGMA; BA, 2015), also known as Adam, was used here because it accelerates the search in the direction of the minima while preventing the search in the direction of oscillations. Its update rule is computed as:

$$\theta^{n+1} = \theta^n - \gamma \cdot \frac{\nu^n}{\sqrt{s^n + \epsilon}} \cdot \nabla \mathcal{C}(\theta^n) \quad (2.3)$$

in which γ is the initial learning rate, $\nu^n = \beta_1 \cdot \nu^{n-1} - (1 - \beta_1) \cdot \nabla \mathcal{C}(\theta^n)$ is the exponential average of gradients, and $s^n = \beta_2 \cdot s^{n-1} - (1 - \beta_2) \cdot \nabla^2 \mathcal{C}(\theta^n)$ is the exponential average of squares of gradients. The hyperparameter β_1 is kept at 0.90, β_2 is kept at 0.99, and ϵ is chosen to be around 10^{-10} , being the defaults values suggested by the authors (KINGMA; BA, 2015).

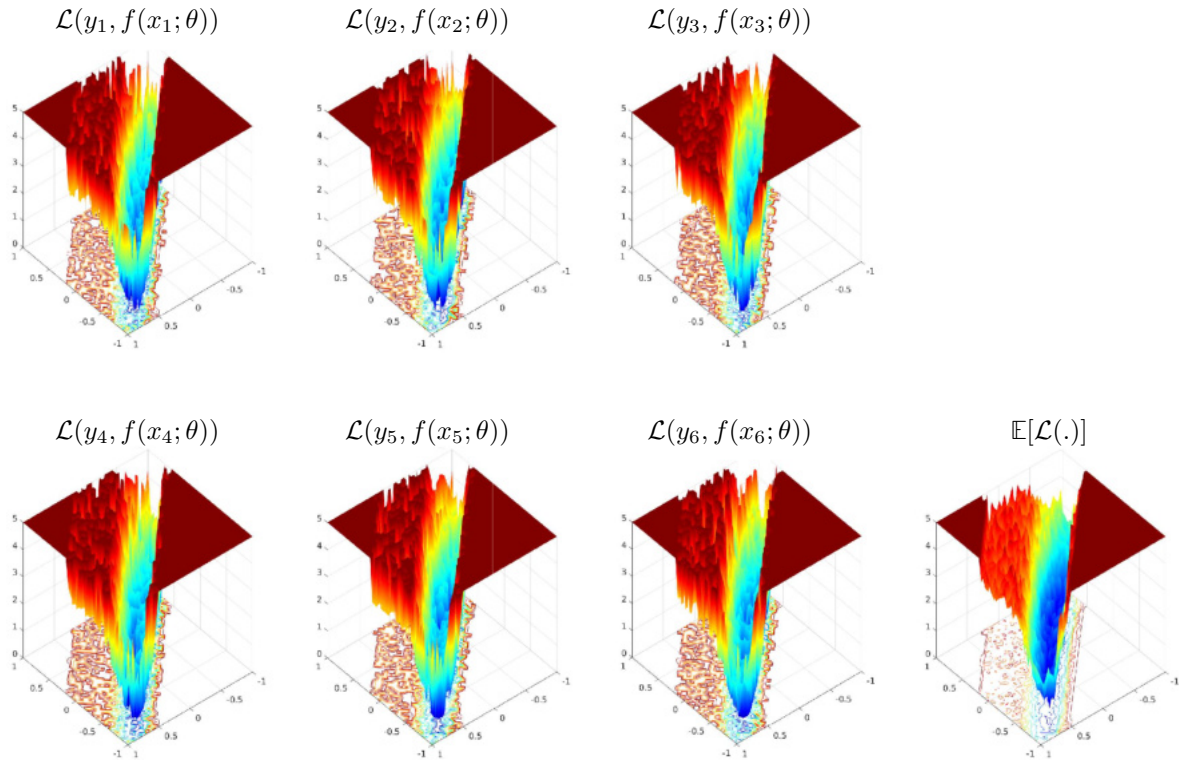
However, navigating complex error surfaces, like the one shown in Figure 3B, are challenging for Gradient Descent. In general, the algorithm is more prone to get stuck in a local minima or saddle point because the cost over S tends to create an error surface with same characteristics. An example of this can be seen in Figure 4, where six loss landscapes and their average are shown. As observed in the figure the average surface maintains the same noisy behavior. Randomness helps in such situations by optimizing at each iteration over a different error surface. This way, the probability of getting stuck at a local zero-gradient region is reduced. The method is called Stochastic Gradient Descent (SGD) (KIEFER; WOLFOWITZ, 1952), and it is the standard for training deep neural networks. The updates are computed by using the gradient of the loss of a randomly sampled subset (without replacement) of the training set, $\mathfrak{M} \subset S$.

$$\theta^{n+1} = \theta^n - \alpha \cdot \sum_{(x_i, y_i) \in \mathfrak{M}} \nabla \mathcal{L}(y_i, f(x_i; \theta^n)), \quad (2.4)$$

where α can be either a static or adaptive learning rate and (x_i, y_i) pairs are usually randomly sampled following a uniform distribution.

Regardless of the network architecture, Gradient Descent or its stochastic version can always be applied. Because the connections are usually forward, f is a composed function, and derivative respect to the weights can be computed using the chain rule.

Figure 4 – Error surfaces for six different elements along with expected value of the loss.



Source: The author (2019)

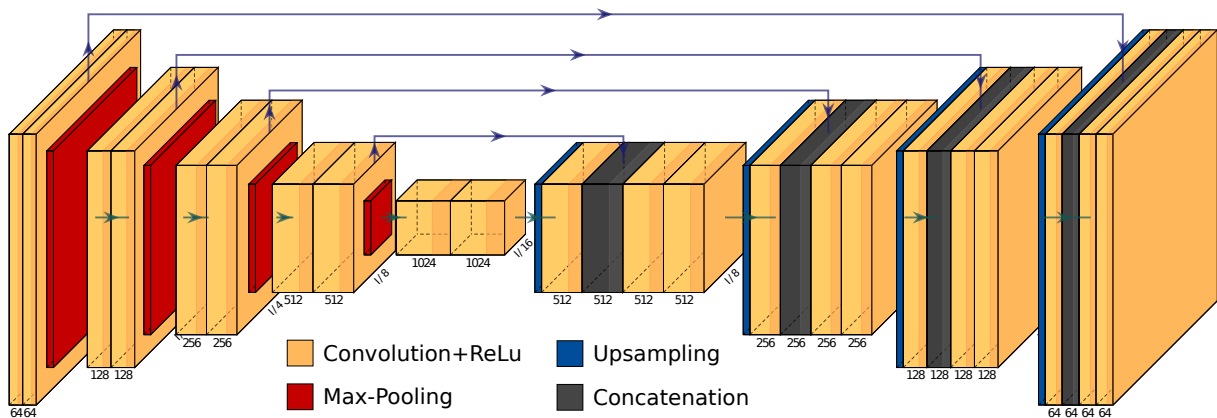
2.2 PIXEL-LEVEL ARCHITECTURE

Several architectures for pixel-level tasks have been proposed in the past few years (LONG; SHELHAMER; DARRELL, 2015; RONNEBERGER; FISCHER; BROX, 2015; JÉGOU et al., 2017; CHEN et al., 2017; CHEN et al., 2018; WU et al., 2019). However, one of the most well-known, straightforward, and well-behaved architecture is U-Net (RONNEBERGER; FISCHER; BROX, 2015). Here it has been opted to use U-Net due to its proven success for different kinds of tasks (ISOLA et al., 2017; AITTALA; DURAND, 2018). This is an encoder-decoder type of network where the first half of the layers contracts the width and heights of feature maps increasing the analyzed receptive field, while the second half transforms the representation to a domain with the same dimension of the input. The full scheme of the architecture is shown in Figure 5.

In the encoder path every block is a sequence of two 3×3 convolution with ReLu layers, followed by a 2×2 max-pooling operation. The reduction of feature maps resolution is contrasted with the increase of the number of kernels. A significantly smaller representation is learned in the bottleneck block (two 3×3 convolutions with ReLu layers followed by $2 \times$ upsampling layer), forcing the identification of sufficiently relevant features to describe the inputs. Then, the second half of the layers acts as a reconstruction path leading to a feature space with the same width and height of those of the source

inputs. Each block applies two 3×3 convolutions with ReLu layers followed by a $2 \times$ upsampling layer. In the decoder path, the number of kernels is doubled for each block thus maintaining symmetry respect to the first half of the network. Skipping connections linking the same depths in the encoder and decoder branches are used to localize and propagate high-resolution features. The linking is performed by concatenating the feature maps of corresponding contraction and expansion layers. This ensures that detailed features learned in the encoder are combined with more global features from the decoder. The network does not have any fully connected layers, and in this Thesis, a 1 pixel padding was used for convolutions operations to maintaining the same size of the input map.

Figure 5 – Overall scheme of the encoder-decoder type of network U-Net.



Source: The author (2019)

The network has more than 31 millions of learnable parameters, *e.g.*, $\theta \in \mathbb{R}^{31,000,000}$, and as it can be seen in Table 1 the weights are concentrated in convolutional layers. The column of trainable parameters refers to the number of weights in the kernels plus the biases for each kernel.

A softmax layer can be added to the end of U-Net in case of pixel-level classification problems. Then, the goal is to assign a class to every pixel by applying a decision rule to a probabilistic output map.

Table 1 – Output tensor size, convolution kernels shapes and trainable parameters for each layer in U-Net network assuming an RGB input image of 1024×1024 .

Layer (type)	Feature maps shape	Kernels shapes	Number of trainable parameters
Input-0	[batch, 3, 1024, 1024]		0
Conv2d-1	[batch, 64, 1024, 1024]	[3, 3, 3, 64]	1,792
ReLU-2	[batch, 64, 1024, 1024]		0
Conv2d-3	[batch, 64, 1024, 1024]	[64, 3, 3, 64]	36,928
ReLU-4	[batch, 64, 1024, 1024]		0
MaxPool2d-5	[batch, 64, 512, 512]		0
Conv2d-6	[batch, 128, 512, 512]	[64, 3, 3, 128]	73,856
ReLU-7	[batch, 128, 512, 512]		0
Conv2d-8	[batch, 128, 512, 512]	[128, 3, 3, 128]	147,584
ReLU-9	[batch, 128, 512, 512]		0
MaxPool2d-10	[batch, 128, 256, 256]		0
Conv2d-11	[batch, 256, 256, 256]	[128, 3, 3, 256]	295,168
ReLU-12	[batch, 256, 256, 256]		0
Conv2d-13	[batch, 256, 256, 256]	[256, 3, 3, 256]	590,080
ReLU-14	[batch, 256, 256, 256]		0
MaxPool2d-15	[batch, 256, 128, 128]		0
Conv2d-16	[batch, 512, 128, 128]	[256, 3, 3, 512]	1,180,160
ReLU-17	[batch, 512, 128, 128]		0
Conv2d-18	[batch, 512, 128, 128]	[512, 3, 3, 512]	2,359,808
ReLU-19	[batch, 512, 128, 128]		0
MaxPool2d-20	[batch, 512, 64, 64]		0
Conv2d-21	[batch, 1024, 64, 64]	[512, 3, 3, 1024]	4,719,616
ReLU-22	[batch, 1024, 64, 64]		0
Conv2d-23	[batch, 1024, 64, 64]	[1024, 3, 3, 1024]	9,438,208
ReLU-24	[batch, 1024, 64, 64]		0
Upsample-25	[batch, 1024, 128, 128]		0
Conv2d-26	[batch, 512, 128, 128]	[1536, 3, 3, 512]	7,078,400
ReLU-27	[batch, 512, 128, 128]		0
Conv2d-28	[batch, 512, 128, 128]	[512, 3, 3, 512]	2,359,808
ReLU-29	[batch, 512, 128, 128]		0
Upsample-30	[batch, 512, 256, 256]		0
Conv2d-31	[batch, 256, 256, 256]	[768, 3, 3, 256]	1,769,728
ReLU-32	[batch, 256, 256, 256]		0
Conv2d-33	[batch, 256, 256, 256]	[256, 3, 3, 256]	590,080
ReLU-34	[batch, 256, 256, 256]		0
Upsample-35	[batch, 256, 512, 512]		0
Conv2d-36	[batch, 128, 512, 512]	[384, 3, 3, 128]	442,496
ReLU-37	[batch, 128, 512, 512]		0
Conv2d-38	[batch, 128, 512, 512]	[128, 3, 3, 128]	147,584
ReLU-39	[batch, 128, 512, 512]		0
Upsample-40	[batch, 128, 1024, 1024]		0
Conv2d-41	[batch, 64, 1024, 1024]	[192, 3, 3, 64]	110,656
ReLU-42	[batch, 64, 1024, 1024]		0
Conv2d-43	[batch, 64, 1024, 1024]	[64, 3, 3, 64]	36,928
ReLU-44	[batch, 64, 1024, 1024]		0

Total of trainable parameters: 31,379,140

2.3 LOSS FUNCTIONS

As stated before, the loss function is the one in charge of measuring the agreement between the expected and obtained outputs. A proper definition of such is crucial for getting suitable error surfaces and for the entire training process in general. However, most of the time, the metric of interest for solving a task is not differentiable. This means that Gradient Descent-based optimization is not possible. In such situations, a tractable surrogate for the metric of interest is optimized instead. Despite there is a high number of loss functions in the supervised deep learning category, depicting all would be unfeasible due to the extensive research in the area. Instead, this work focus only on pixel-level loss functions. Specifically, the most relevant losses for both semantic segmentation and pixel-level regression are explored. Loss functions for Generative Adversarial Networks, Metric Learning and unsupervised approaches are not included here because they fall out of the scope of this research.

In pixel-level classification, also called semantic segmentation, each pixel of a given input must be assigned to one class. The problem refers to the prediction of an integer value, where each class is assigned a unique integer value from 0 to $(C - 1)$, being C the number of classes. Because minimizing expected 0-1 loss is intractable in practice (GOODFELLOW; BENGIO; COURVILLE, 2016), the problem is cast as predicting the probability of the example belonging to each known class. In this category, two significant groups of loss functions can be identified: information theory-based and region-based loss functions.

The most well-known and used function into the information theory-based approach is the Cross Entropy (CE),

$$\mathcal{L}(y, z) = -\frac{1}{|\Omega|} \sum_{l=0}^{C-1} \sum_{p \in \Omega} y_l(p) \cdot \log z_l(p), \quad (2.5)$$

where $y: \Omega \rightarrow \{0, 1\}^C$ is called the one-hot representation of the segmentation ground truth and $z: \Omega \rightarrow \mathbb{R}^C$ is the output probability map. Although Cross Entropy has been vastly used in Machine Learning before, its first application for Deep Learning-based semantic segmentation can be traced back to the work of Long et al. (LONG; SHELHAMER; DARRELL, 2015).

Although acceptable solutions are usually obtained with CE, its definition assumes that all pixels/voxels have the same importance for the training process. One of the main disadvantages of this definition is its ineffectiveness in the case of imbalanced classes (ZHOU et al., 2017a). A generalization of CE loss, known as Weighted Cross Entropy (WCE), is more suitable for these situations, that requires the creation of custom weights maps,

$$\mathcal{L}(y, z) = -\frac{1}{|\Omega|} \sum_{l=0}^{C-1} \sum_{p \in \Omega} \omega_l(p) \cdot y_l(p) \cdot \log z_l(p), \quad (2.6)$$

where ω_ϱ is a weight map. The standard for this weight map for imbalance class problems is known as Balance Weight Map (BWM) (ZHOU et al., 2017a) and it is expressed as $\omega^B(p) = \sum_{p \in \Omega} 1/n_l(p)$, being $n_l(p)$ the number of pixels in the class l , such that $y_l(p) = 1$.

One of the first uses of WCE for DL-based semantic segmentation was in the work of Ronneberger et al. (RONNEBERGER; FISCHER; BROX, 2015) along with a weight map for increasing the focus of the loss function on background regions between two near objects.

$$\omega_\sigma^{UNET}(p) = \omega^B(p) + \nu \cdot \exp\left(-\frac{(\phi_1(p) + \phi_2(p))^2}{2\sigma^2}\right), \quad (2.7)$$

where ϕ_1 denotes the distance to the border of the nearest object, and ϕ_2 is the distance to the edge of the second-closest object.

Two years later, a dynamic weight map variation of WCE was introduced by (LIN et al., 2017). This loss function, known as Focal loss, was initially applied with success for object detection tasks (LIN et al., 2017), but later adapted for image segmentation in (ZHOU et al., 2017b) allowing to focus more the attention on the regions that were wrongly segmented.

$$\mathcal{L}(y, z) = -\frac{1}{|\Omega|} \sum_{l=0}^{C-1} \sum_{p \in \Omega} (1 - z_l(p))^2 \cdot \omega_\varrho(p) \cdot y_l(p) \cdot \log z_l(p). \quad (2.8)$$

Inside the region-based loss function category, the most used loss is the Soft Dice function (MILLETARI; NAVAB; AHMADI, 2016) that was initially proposed for binary segmentation and later generalized for multi-class problems. Several variants for Dice (FIDON et al., 2017; HASHEMI et al., 2018; YANG; KWEON; KIM, 2019) and its combination with Cross Entropy (WONG et al., 2018) and Focal loss (ZHU et al., 2019) has been proposed recently, aiming to drive optimizations towards a solution that maximizes the F1 score,

$$\mathcal{L}(y, z) = \sum_{l=0}^{C-1} \left(1 - \frac{2 \cdot \sum_{p \in \Omega} y_l(p) \cdot z_l(p)}{\sum_{p \in \Omega} y_l^2(p) + \sum_{p \in \Omega} z_l^2(p)} \right). \quad (2.9)$$

However, because the denominator in the expression can be zero, numerical instabilities are very common with this approach. A year later a follow up of the Dice loss function led to the creation of the Tversky loss function (SALEHI; ERDOGMUS; GHOLIPOUR, 2017) and its combination with Focal loss (ABRAHAM; KHAN, 2019),

$$\mathcal{L}(y, z) = \sum_{l=0}^{C-1} \left(1 - \frac{\sum_{p \in \Omega} y_l(p) \cdot z_l(p)}{\sum_{p \in \Omega} y_l(p) \cdot z_l(p) + \alpha \sum_{p \in \Omega} (1 - y_l(p)) \cdot z_l(p) + \beta \sum_{p \in \Omega} y_l(p) \cdot (1 - z_l(p))} \right) \quad (2.10)$$

A more recent approach was introduced by Berman et al. (BERMAN; TRIKI; BLASCHKO, 2018) for direct optimization of the mean intersection-over-union loss (also known as

Jaccard index) in neural networks.

$$\mathcal{L}(y, z) = \frac{1}{C} \sum_{l=0}^{C-1} \sum_{p \in \Omega} m_l(p) \cdot g_m(p), \quad (2.11)$$

where the error m_l is defined as:

$$m_l(p) = \begin{cases} 1 - z_l(p) & \text{if } y_l(p) = 1 \\ z_l(p) & \text{otherwise} \end{cases}, \quad (2.12)$$

and g_m is the difference between the consecutive values of the cumulative distribution of m .

Despite other approaches (KERVADEC et al., 2019; LEE et al., 2019; CLOUGH et al., 2019) have been recently proposed for applying shapes regularization into the loss function, their goal is to segment only one object. Then, applying these to images with hundred of cells is unpractical. Because this work focus on instance segmentation, *e.g.*, more than one object needs to be segmented, this kind of loss functions are not considered.

On the other hand, in pixel-level regression each pixel must be assigned a real-valued quantity. Some of the tasks into this category are Depth Estimation (ALHASHIM; WONKA, 2018), Style Transfer (KOTOVENKO et al., 2019), Super Resolution (ZHAO et al., 2019), Denoising (YUE et al., 2019), Deblurring (AITTALA; DURAND, 2018), Fusion (LIU et al., 2017), among others. The loss functions can be further grouped in per-pixel and perceptual losses.

The Mean Squared Error loss, also known as L2, is the standard per-pixel loss function used in regression problems. Mean Squared Error loss is computed as the mean of the squared differences between expected and predicted values for each pixel. The result is always positive independently of the sign of the predicted and actual output, and accurate estimations lead to values of 0. The squaring means that more significant mistakes result in more error than smaller ones, meaning that the model is punished for higher disagreements. In general L2 loss for a C channels output can be computed as:

$$\mathcal{L}(y, z) = \frac{1}{C \cdot |\Omega|} \sum_{l=0}^{C-1} \sum_{p \in \Omega} (y_l(p) - z_l(p))^2. \quad (2.13)$$

However, small disagreements are also significant, especially in the final steps of the optimization. Then, the L2 measurement usually leads to the appearance of noisy artifacts. In cases where robustness to outliers is required, the Mean Absolute Error, also known as L1 loss, is the more appropriate approach. The loss is calculated as the average of the absolute difference between the actual and predicted values for every pixel.

$$\mathcal{L}(y, z) = \frac{1}{C \cdot |\Omega|} \sum_{l=0}^{C-1} \sum_{p \in \Omega} |y_l(p) - z_l(p)|. \quad (2.14)$$

A general expression that accounts the behavior of both L2 and L1 is called Huber loss (CAVAZZA; MURINO, 2016) and it is defined as:

$$\mathcal{L}_\delta(y, z) = \begin{cases} \frac{1}{2}a^2 & \text{for } |a| \leq \delta \\ \delta \left(|a| - \frac{1}{2}\delta \right) & \text{otherwise} \end{cases}, \quad (2.15)$$

where $|a|$ and a^2 are the expression for L1 and L2 loss respectively.

When the expected output is an image, enhancement of edges can be further ensured by using image gradient as a regularization term of the loss function (CHAITANYA et al., 2017),

$$\mathcal{L}(y, z) = \frac{1}{C \cdot |\Omega|} \left(\sum_{l=0}^{C-1} \sum_{p \in \Omega} |y_l(p) - z_l(p)| + |\nabla y_l(p) - \nabla z_l(p)| \right), \quad (2.16)$$

where y and z are images and $\nabla y, \nabla z$ their respective gradients.

Nevertheless, per-pixel loss functions do not capture visually coherent similarities. Perceptual losses try to tackle this problem by measuring the agreement in terms of structural similarities. This is the case of the loss function proposed by Johnson et al. (JOHNSON; ALAHI; FEI-FEI, 2016), which combines the L2 loss with a pre-trained VGG (SIMONYAN; ZISSERMAN, 2014) network φ also used as part of the loss function. The idea is to compute the difference between y and z by taking the average of the L2 but in the feature space defined by F features maps that are obtained evaluating y and z into VGG.

$$\mathcal{L}(y, z) = \frac{1}{F \cdot \sum_{k=0}^F |\Omega_k| + C_k} \sum_{k=0}^F \sum_{l=0}^{C_k} \sum_{p \in \Omega_k} (\varphi_k(y_l)(p) - \varphi_k(z_l)(p))^2, \quad (2.17)$$

where φ_k represents the k -th feature map given by the pre-trained network φ . Note that the weights of the network φ are never updated. One of the major drawbacks of this approach is that it requires using VGG for extracting the features of the obtained response z at each iteration, which is time-consuming and computationally expensive.

A Structural Similarity-based (SSIM) loss function was proposed in the same year (ZHAO et al., 2016), aiming to obtain a perceptual loss with visually pleasing results. Based on the SSIM definition, the loss is defined as:

$$\mathcal{L}(y, z) = \sum_{l=0}^{C-1} 1 - \frac{2 \cdot \mu_{y_l}(p) \cdot \mu_{z_l}(p) + C_1}{\mu_{y_l}^2(p) + \mu_{z_l}^2(p) + C_1} \cdot \frac{2 \cdot \sigma_{y_l, z_l}(p) + C_2}{\sigma_{y_l}^2(p) + \sigma_{z_l}^2(p) + C_2}, \quad (2.18)$$

where $\mu_y(p)$ and $\mu_z(p)$ are the intensities average over a neighborhood of p for y and z respectively. Similarly, σ^2 is the variance for each patch. Although structural similarities are enforced, colors shift are common with this kind of measurement (ZHAO et al., 2016).

Task specific loss functions like the one proposed for High Dynamic Range reconstruction (EILERTSEN et al., 2017) and Inpainting (LIU et al., 2018) were not considered in this work.

2.4 MATHEMATICAL MORPHOLOGY

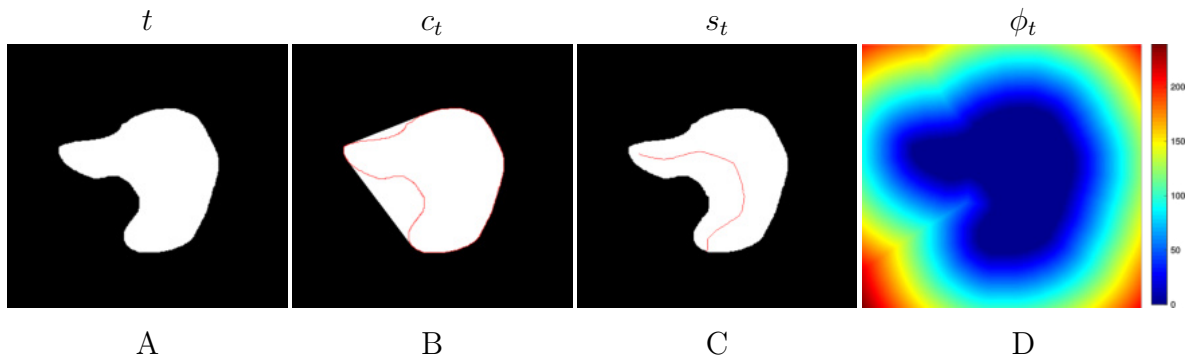
Some of the concepts used in the following chapters are part of the Mathematical Morphology sub-field. The methods in this area use shape information of each object in the image. The primary operations are dilation, $h \oplus s = \{z | (\hat{s}_z \cap h \neq \emptyset)\}$, and erosion, $h \ominus s = \{z | (s_z \subseteq h)\}$, where s is a structural element, \hat{s} is the reflection of the structural element, and $z \in \Omega$ is a pixel/voxel location. Then, the morphological opening is defined as $h \circ s = (h \ominus s) \oplus s$, and the closing as $h \bullet s = (h \oplus s) \ominus s$. Additionally, top-hat transform is defined as $\bar{\varrho}_s = h - (h \circ s)$, and bottom-hat transform as $\varrho_s = (h \bullet s) - h$ (GONZALEZ; WOODS, 2007).

Convex Hull. Let t be the mask of an object in a binary image. The convex hull c_t of the blob is defined as the smallest convex mask that contains the elements of t . In this definition, a convex mask refers to a blob where, for any two chosen points of the contour, the line between them does not contain any other contour point (GONZALEZ; WOODS, 2007). The algorithm begins selecting an initial contour point and includes the relative leftmost contour point successively counterclockwise. An example is shown in Figure 6B.

Binary Image Skeleton. The skeleton of a binary mask t refers to a one-pixel wide representation of t that maintains the general morphology of the mask. Several approaches exist in the literature for computing such kinds of representations. In this work, the thinning algorithm (LAM; LEE; SUEN, 1992) is adopted because of its computational efficiency and less branched skeleton. The skeleton for s_t is shown in red in Figure 6C on top of the mask t .

Distance Transform. Given a mask t , its distance transform ϕ_t is defined as $\phi_t(p) = \min ||p - q||_2^2$ where $p, q \in \Omega$ and $t(q) > 0$. In other words, for every pixel/voxel in the image, its Euclidean distance to the closest non-zero point is computed. Figure 6D shows the distance transform for a binary blob. In this work, the linear time algorithm proposed in (MAURER; QI; RAGHAVAN, 2003) was used.

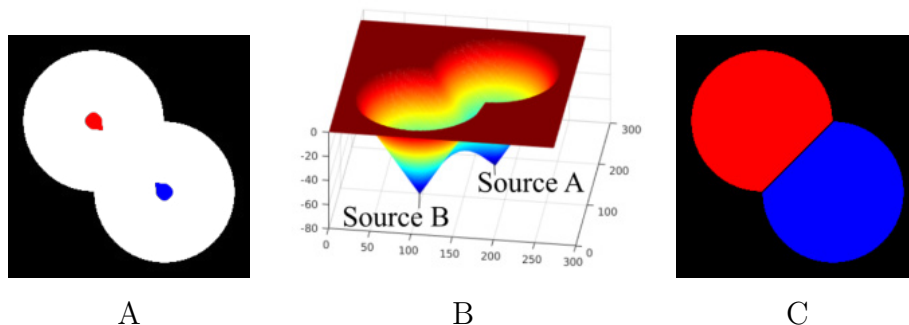
Figure 6 – Example of (A) a mask t and its (B) convex hull c_t , (C) skeleton s_t , and (D) distance transform ϕ_t . The mask contour and skeleton are shown in red in (B) and (C) for better visualization.



Source: The author (2019)

Watershed Transform. The Watershed Transform (WT) is defined as the successively flooding of a topological surface, and a barrier is created wherever two sources of water meet. In this work, the Watershed with markers (MEYER, 1994) approach was adopted being every source of water associated to a different instance number. In Figure 7A is shown a blob h along with the initial markers represented in color red and blue, respectively. The regions marked by the seed increasingly grow following the topological surface (Figure 7B). The points where the regions meet delimitate the division between both instances like the one shown in Figure 7C.

Figure 7 – Example of (A) a binary image h along with two initial markers represented in color red and blue respectively, (B) the topological surface to be flooded, and (C) the final segmentation segmentation.



Source: The author (2019)

2.5 INSTANCE SEGMENTATION

One of the most important and challenging task in pixel-level classification is instance segmentation. Long et al. (LONG; SHELHAMER; DARRELL, 2015) proposed the first DL-based solution for the task using a Fully Convolutional Network (FCN) which improved the image-level classification in a Convolutional Neural Network (CNN) to a pixel-level classification. This allowed segmentation maps to be generated for images of any size, and it was much faster compared to the then common patch classification approach. In the same year, Ronneberger et al. (RONNEBERGER; FISCHER; BROX, 2015) introduced U-Net, a FCN encoder-decoder type of network architecture, together with a Weighted Cross Entropy loss function to segment biomedical images. This network was a breakthrough, achieving remarkable segmentation results, from cells to organs.

Browet et al. (BROWET et al., 2016), working with mouse embryo cells, estimated pixel probabilities for cell interior, borders, and background and then minimized an energy cost function to match the class probabilities via graph-cuts. In this work it is favored to avoid the pitfalls of graph-cuts and the thresholding adopted in their formulation to define seeds within cells. Chen et al. (CHEN et al., 2016) proposed DCAN, a contour aware FCN to segment glands from histology images towards improving the automatic diagnosis

of adenocarcinomas. They also modeled a loss with contours which led them to win the *2015 MICCAI Gland Segmentation Challenge* (SIRINUKUNWATTANA; PLUIM; CHEN, 2017) confirming the advantages of explicitly learning contours. Zhang et al. (ZHANG; YARKONY; HAMPRECHT, 2014) developed a learning-based method to do correlation clustering of superpixels and obtain a contour segmentation of each cell in bright field and phase-contrast images, with particular attention to almost transparent cells. Recently, Xu et al. (XU et al., 2017) proposed a three-branch network to segment individual glands in colon histology images.

However, a significant limitation of all previous methods is the need for near-perfect annotations. Although several approaches were proposed recently for working in weakly supervised conditions with color images (LI; ARNAB; TORR, 2018; REDONDO-CABRERA; BAPTISTA-RÍOS; LÓPEZ-SASTRE, 2019), usually a prior knowledge of daily life images is used to solve the problem. Probably the most popular instance segmentation solution is Mask R-CNN (HE et al., 2017) that uses two stacked networks for detection followed by segmentation of natural images. Others have used three stacked networks for semantic segmentation and regression of a Watershed energy map allowing separating nearby objects (BAI; URTASUN, 2017). In (BRABANDERE; NEVEN; GOOL, 2017; FATHI et al., 2017) the authors use loss functions for regressing pixel-level embeddings that are later grouped. In (KERVADEC et al., 2019), the authors propose a weakly semantic segmentation method for biomedical images. They include prior knowledge in the form of constraints into the loss function for regularizing the size of the segmented object. The work in (YANG et al., 2017) proposes a way to keep annotations at a minimum while still capturing the essence of the signal present in the images. The goal is to avoid excessively annotating redundant parts, present due to many repetitions of almost identical cells in the same image. In (LIANG et al., 2018), the authors also craft a tuned loss function applied to improve segmentation on weakly annotated gastric cancer images.

2.6 MULTI-FOCUS IMAGE FUSION

Multi-focus Image Fusion is a pixel-level regression task within the image restoration subfield. The problem consist in, given two sources frames with different focus planes, obtaining an all-in-focus image. Depending on the adopted fusion, the methods can be classified either as a transform domain or a spatial domain based approach (NEJATI; SAMAVI; SHIRANI, 2015). While most methods fall into the first category, recent advances in neural networks have attracted the attention to spatial domain approaches, mostly due to efficiency improvements.

Transform domain methods. This class of methods, such as in every transformation approach in Computer Vision, attempts to solve the problem in another domain. In multi-focus methods, one usually transforms the source images to a multi-scale domain, a subset of coefficients is selected or filtered from each source, and then a fusion

of the decomposed coefficients is applied generating a reconstructed image in the corresponding domain. Finally, an inverse transform creates an all-in-focus spatial image. Main contributions in this area are in transformation selection, filtering of coefficients, and formulation of fusion rules. Some of the methods employ Gradient Pyramid (PETROVIC; XYDEAS, 2004), Wavelet Transforms (LEWIS et al., 2007), Contourlet Transform (ZHANG; GUO, 2009) and Discrete Cosine Transform (HAGHIGHAT; AGHAGOLZADEH; SEYEDARABI, 2010), (HAGHIGHAT; SEYEDARABI, 2011). These methods usually have higher computational costs due to the transform and inverse transform operations. Some methods do not even specify the domain, but they try to learn the best feature space to solve the problem. Examples include the approaches based on Independent Component Analysis and Sparse Representation (YANG; LI, 2010).

Spatial domain methods. Differently, to the previous approach, methods in this category try to reconstruct the all-in-focus image using intensity information. The formulation usually relies on the proposal of a focus metric that allows selecting the sharpest pixel within the sources. A sequence of filtering or morphological operations is also typical in this kind of methods. Some of the most representative approaches include the Image Matting for fusion (LI et al., 2013) and the Guided Filtering Fusion (LI; KANG; HU, 2013), both proposed by Li, Kang, and Hu with results comparable to transform domain strategies but without the associated computational cost incurred by transformations. However, their manually designed morphological filtering assumes specific priors that may not apply to all images.

New spatial methods use deep learning as an alternative to hand-crafted solutions. Their main contributions are on the creation of network architecture and training datasets. Since the proposed architectures are generally Siamese based, these methods use a local neighborhood feature approach where every pixel is classified either as blurred or sharp. Despite the good results, morphological post-processing is still needed to resolve global features, *e.g.*, filling holes. This increases the execution time as well as might add an unnecessary constraint to the solution space, no small holes, for example. The work most related to this research was proposed by Xiang Yan et al. (YAN et al., 2018), which employs a structural similarity (SSIM) based loss function to achieve end-to-end unsupervised learning. However, differently to the proposal in this work, Xiang Yan et al. use a Siamese-based architecture with several intermediate average fusions. This is a common approach in image fusion (LIU et al., 2017; TANG et al., 2018) but it lacks flexibility when compared to multiple sources models where all frames are processed at the same time (ZAGORUYKO; KOMODAKIS, 2015). Another method related to the approach presented here is the segmentation-based model proposed by Liu et al. (LIU et al., 2017). In their Siamese CNN method, the multi-focus image fusion is treated as a pixel classification problem. However, the post-processing required to combine the classification of each patch from the image increases the total execution time.

3 MULTICLASS WEIGHTED LOSS FOR INSTANCE SEGMENTATION OF CLUSTERED CELLS^[1]

In this chapter two new loss functions are introduced for fully supervised image segmentation. The context is medical imaging, and the motivation is the need of the biologists to quantify and model the behavior of blood T-cells which might help in the understanding of their regulatory mechanisms and ultimately help researchers in their quest for developing an effective immunotherapy cancer treatment. The challenge in terms of the optimization is that, different to natural images datasets that have a vast amount of data, medical images datasets are usually smaller, and therefore, finding a well-performed model is difficult.

3.1 INTRODUCTION

It is not fully understood how blood stem cells differentiate over time to generate all blood cell types in the body nor what are the mechanisms that drive their specialization. T-cells are descendants of blood stem cells with an important role in emerging immunotherapy cancer treatments (ROSENBERG; RESTIFO, 2015). The main interest is to determine how decisions are made by individual progenitor T-cells under controlled environmental conditions (ROTHENBERG; MOORE; YUI, 2008). To carry out experiments, individual T-cells are isolated in microwells where they grow and proliferate for approximately six days. Multiple cell divisions occur in each microwell leading to a dense cell population originated from a single cell. Multichannel images are acquired at time intervals to follow cell development, which can then be quantified by analyzing fluorescent signals expressing specific markers of differentiation. Segmenting individual cells is necessary to measure signal activation per cell and to count how many cells are active over time (see Figure 8).

The difficulties are in segmenting adjoining cells. These can take any shape, when clustered or isolated, and their touching borders have nonuniform patterns defeating classical

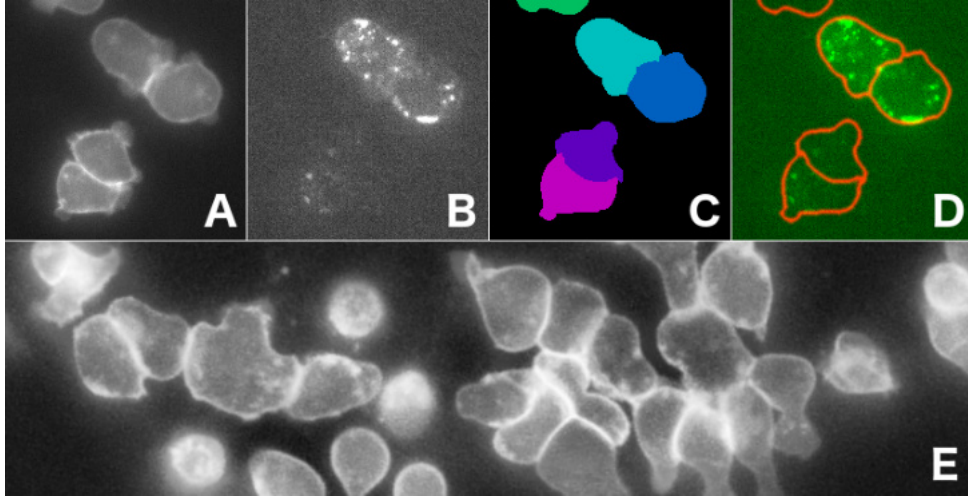
^[1] Fidel A. Guerrero Peña; Pedro Marrero Fernandez; Tsang Ing Ren; Mary Yui; Ellen Rothenberg; Alexandre Cunha. Centro de Informática, Universidade Federal de Pernambuco, Brazil; Division of Biology and Biological Engineering, California Institute of Technology, USA; Center for Advanced Methods in Biological Image Analysis, California Institute of Technology, USA. Published in: IEEE International Conference on Image Processing (ICIP), 2018.

©2018 IEEE. Reprinted, with permission, from Fidel A. Guerrero Peña, Pedro Marrero Fernandez, Tsang Ing Ren, Mary Yui, Ellen Rothenberg, Alexandre Cunha. Multiclass Weighted Loss for Instance Segmentation of Clustered Cells, IEEE International Conference on Image Processing, October 2018.

In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of UFPE's products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

segmentation approaches. Weak boundaries are also troubling (Figure 8E). Furthermore, the total pixel count on adjoining borders is considerably smaller than the pixel count for the other image parts which contributes to numerical optimization difficulties when training a neural network with imbalanced data (HE; GARCIA, 2009) and without a properly calibrated loss function. The situation is exacerbated in large clusters where cells might overlap making it difficult, even for the trained eye, to locate cell contours.

Figure 8 – Example of cells marked by the *mTomato* fluorophore are shown in (A). Their corresponding signal of interest, *CD25*, which changes over time, is expressed in some cells (B). The goal is to segment individual cells, as shown in (C), and colocalize *CD25* to measure its concentration within each cell (D) and consequently count how many cells are active at any given time. In this illustration, the top two cells are fully active as reported by their high *CD25* content. Colored masks in (C) are for illustration purpose only. A typical cluttering of T-cells is presented on panel E.



Source: PEÑA *et al.* (2018)

3.2 MULTICLASS SHAPE-BASED WEIGHTED CROSS ENTROPY LOSS FUNCTIONS

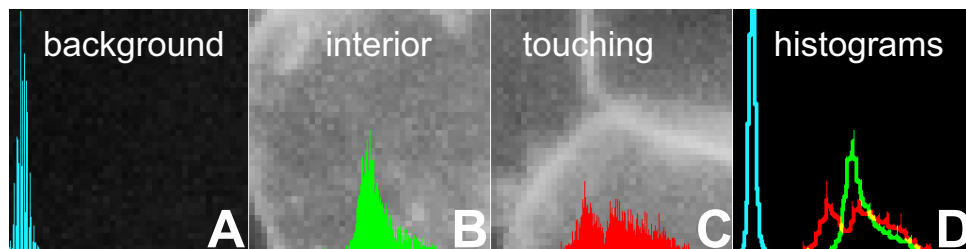
Let $S = \{(x_1, h_1), \dots, (x_N, h_N)\}$ be the training set, with cardinality $|S| = N$, where $x_k: \Omega \rightarrow \mathbb{R}^+, \Omega \subset \mathbb{R}^2$, is a gray-level image and $h_k: \Omega \rightarrow \{0, \dots, C\}$ its segmentation ground truth with $C + 1$ classes. Let (x, h) be a generic tuple from S . Here it is called h^0 and h^1 , respectively, the background and foreground subsets of a binary ground truth h , and more generally $h^l = \{p \mid h(p) = l, p \in \Omega\}$ with cardinality $n_l = |h^l|$ for non-binary cases. The pixel indicator function $\mathbb{1}_{h^l}(p)$ is written simply as $y_l(p)$, *e.g.* $y_l(p) = 1$ if $p \in h^l$, otherwise $y_l(p) = 0$. The connected components of h , $h^T = \{t_j \mid t_j \cap t_i = \emptyset, j \neq i\}, \cup_j t_j = h^1$, are the non-empty masks for all trainable cells in x . For a mask t , Γ_t represents its contour and c_t gives its convex hull. Let $\Gamma = \cup_t \Gamma_t$ be the set of all contour pixels in h . A mask admits a skeleton s_t here computed using the thinning algorithm

(LAM; LEE; SUEN, 1992). The notation $\phi_h: \Omega \rightarrow \mathbb{R}$ refers to the distance transform of an image that assigns to every pixel of h its Euclidean distance to the closest non-background pixel (MAURER; QI; RAGHAVAN, 2003), as described in Section 2.4. The goal of training a segmentation network is to obtain a segmentation map \hat{h} as close as possible to h , $\hat{h} \approx h$, given the image x . When x is evaluated in the segmentation network f , a probability map $z: \Omega \rightarrow \mathbb{R}^{C+1}$ is obtained such that $z_l(p)$ reports the probabilities of pixel p belonging to the class l . Then, the binary segmentation \hat{h} can be obtained from z by applying a decision rule.

3.2.1 Class Augmentation

Touching cells in an image x share a common boundary, which, by construction, is a one-pixel wide background gap separating their respective connected components in h^T . Some authors, *e.g.*, (RONNEBERGER; FISCHER; BROX, 2015; XU et al., 2017), consider the one pixel wide gaps in h separating connected components to be part of the background but with larger weights. By doing so, it might be diminished the discriminative power of the network as the foreground and background intensity distributions overlap to some extent causing separation of pixels more difficult, as suggested by the histograms shown in Figure 9. In the figure it can be seen the difference between the signatures of touching borders, cell interiors, and background. If touching pixels are considered background pixels for the purpose of training the network with only two classes, the distance between the classes, foreground and background, would not be as pronounced as if three separate classes are considered. This way, background is far off the other two classes leaving interior and touching regions to be resolved. In this work a multiclass learning approach for binary segmentation of clustered objects is proposed, which it is expected to enhance the discriminative resolution of the network and hence obtain a more accurate segmentation of individual cells.

Figure 9 – Example of distinct intensity and structural signatures of the three predominant regions: background (A), cell interior (B), in-between cells (C). The combined histogram curves for comparison is show in (D). This distinction led us to adopt a multiclass learning approach which helped resolve the narrow bright boundaries separating touching cells, as seen in (C).



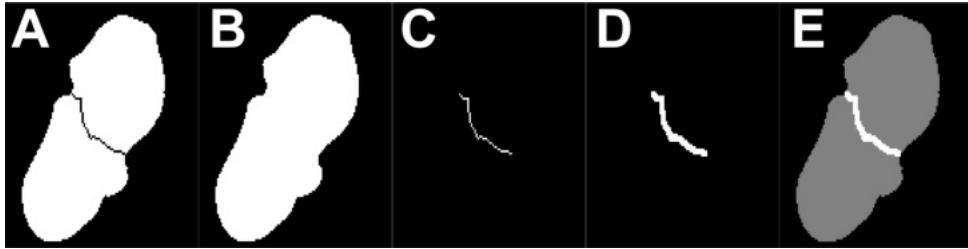
Source: PEÑA *et al.* (2018)

For performing label augmentation on the binary h it is created a third class corre-

sponding to touching borders. This is done using morphological operations (Algorithm 1). By design, this new class occupies a slightly thicker region than the original gap between cells. Then, the training is done using an augmented h and the resulting map z will have an extra class representing the distribution for touching pixels.

The goal in this step is, given a C classes label h , to return a $C+1$ classes label h' . In this work the input annotation had $C = 2$ classes (Figure 10A). First, a morphological closing, *e.g.*, a dilation followed by an erosion, is performed. With this operation it is obtained a cluster of cells as a single connected component, Figure 10B. Then, the difference between closed region and original cluster returns the cell division pixels (Figure 10C), followed by a morphological dilation (Figure 10D). Addition of obtained region with original 2 classes label as in step 4 of Algorithm 1 results in a 3 classes label as shown in Figure 10E. All h' values above $C + 1$ are assigned to last class guarantying that dilated pixels of division regions do not fall outside labels range $l \in [0, C + 1]$ (step 5 Algorithm 1). The notation $h'|_{[0, C+1]}$ is used to refer that h' is constrained to the domain $[0, C + 1]$. In this work a 3×3 squared structuring element s_e was used.

Figure 10 – Example of (A) two classes ground truth, (B) cluster of cells after morphological closing, (C) touching region and (D) its morphological dilation, and (E) final three classes label augmentation.



Source: PEÑA *et al.* (2018)

Algorithm 1: Augment ground truth

Input: h, s_e

Output: h'

- 1 $h' \leftarrow (h \oplus s_e) \ominus s_e;$
 - 2 $h' \leftarrow h' - h;$
 - 3 $h' \leftarrow h' \oplus s_e;$
 - 4 $h' \leftarrow h + (\max(h) + 1) * h';$
 - 5 $h' \leftarrow h'|_{[0, \max(h)+1]};$
-

3.2.2 Focus Weights Map

Yet, another challenge of typical biomedical datasets is the high imbalance of the classes. Despite the most common approach for treating the problem is the Balanced Cross Entropy (ZHOU *et al.*, 2017a) loss function, an improvement in the results for clustered objects was shown by Ronneberger *et al.* (RONNEBERGER; FISCHER; BROX, 2015) when using

custom weight maps. In this work it is proposed to use higher weights to alleviate the imbalance of classes in the training data and to emphasize cell contours, especially at touching borders, while maintaining lower weights for the abundant, more homogeneous, easily separable background pixels. However, it is also critical that background pixels around cell contours should carry proportionally higher weights as they help to capture cell borders more accurately especially in acute concave regions. The Weighted Cross Entropy loss function (RONNEBERGER; FISCHER; BROX, 2015) is here used to focus learning on important but underrepresented parts of an image:

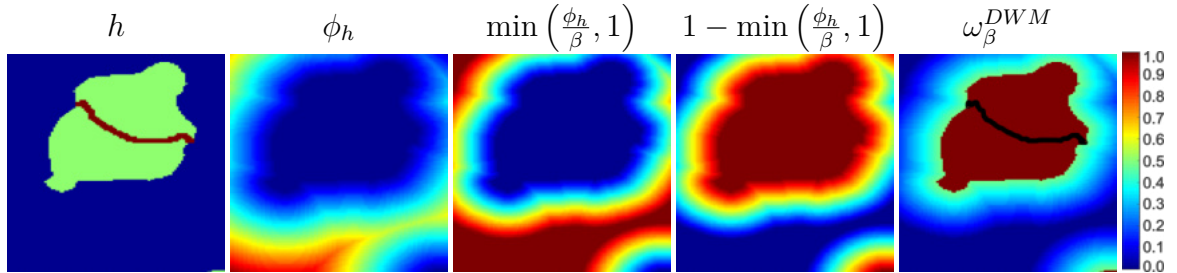
$$\mathcal{L}(y, z) = - \sum_{l=0}^C \sum_{p \in \Omega} \omega_{\varrho}(p) \cdot y_l(p) \cdot \log z_l(p) \quad (3.1)$$

where ω_{ϱ} is a known weight map parameterized by ϱ , $y_l(p)$ is the class indicator function, and $z_l(p)$ is the probability of pixel p belonging to class l , where $C \in \{1, 2\}$ depending on the annotation used, *e.g.*, binary h or multiclass h' respectively. The first proposed loss function uses Equation 3.1 as base and a distance transform based weight map (DWM) computed as:

$$\omega_{\beta}^{DWM}(p) = \omega^B(p) \cdot \left(1 - \min \left(\frac{\phi_h(p)}{\beta}, 1 \right) \right) + 1 \quad (3.2)$$

where $\beta \geq 1$ is a control parameter that decays the weight from the contour, and $\omega^B(p) = 1/n_l$, for $p \in h^l$, is the class imbalance weight (ZHOU et al., 2017a), inversely proportional to the number of pixels in the class. Typically $|h^0| > |h^1| > |h^2|$, but the weights hold regardless. Note that ω_{β}^{DWM} becomes one for $\phi_h > \beta$, and for $\phi_h \in [1, \beta]$, a linear decay $\omega_{\beta}^{DWM}(p) = \omega^B(p) \cdot (1 - \phi_h(p)/\beta) + 1$ is obtained for background pixels $p \in h^0$. Non-background pixels ($\phi_h = 0$) have class constant weights $\omega^B + 1$. An step by step example for computing this weight map is shown in Figure 11. Figure 13C shows an example of ω_{β}^{DWM} with $\beta = 30$.

Figure 11 – Example of computation of each term in Equation 3.2 for going from the semantic segmentation ground truth to the final DWM weight map. Color code is normalized to maximum weight value with red representing higher weights and blue small weights.



Source: The author (2019)

It turns out that segmenting valid minutiae, *e.g.*, cell tip in Figure 12A, usually in the form of narrow regions, requires relatively stronger weights. This leads to the formulation

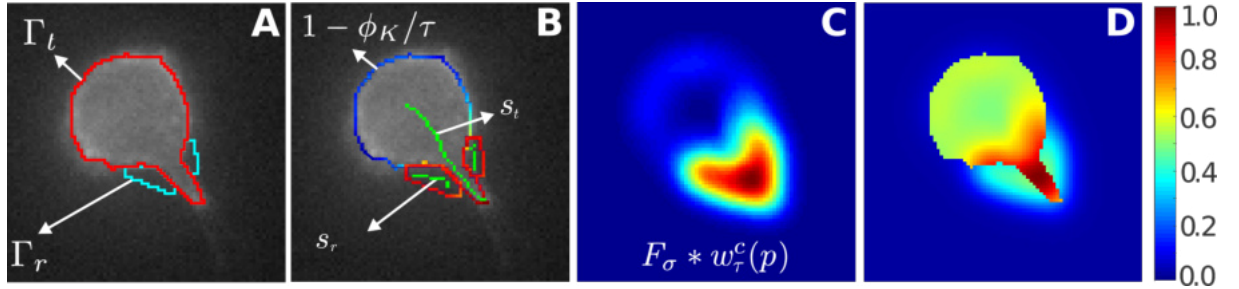
of a shape aware weight map to take into account small but important nuances around contours. Let $r_t = c_t \setminus t$ be the concave complement of $t \in h^T$. Let K be a binary image with skeletons $s_t \cup s_r$ as foreground pixels, and ϕ_K the distance transform over K . Let $\Gamma_H = \Gamma_t \cup \Gamma_r$. Then, the second proposal based on a shape aware weight map (SAW) is

$$\omega_{\tau,\sigma}^{SAW}(p) = \omega^B(p) + F_\sigma * \omega_\tau^c(p) + 1 \quad (3.3)$$

where convolution with filter F_σ , which combines copy padding and Gaussian smoothing (Figure 12C), propagates ω_τ^c values, shown in Figure 12B, from Γ_H to neighboring pixels,

$$\omega_\tau^c(p) = \begin{cases} 1 - \phi_K(p)/\tau & \text{for } p \in \Gamma_H \\ 0 & \text{otherwise} \end{cases} \quad (3.4)$$

Figure 12 – Example of (A) cell contour Γ_t and concave complement contour Γ_r with (B) its respective skeletons s_t and s_r . The contour points importance $\omega_\tau^c(p)$, for $p \in \Gamma$, is also shown in (B). Finally, (C) copy padding and Gaussian smoothing is applied and (D) sum to the class balance weight $\omega^B + 1$. Color code is normalized to maximum weight value with red representing higher weights and blue small weights.

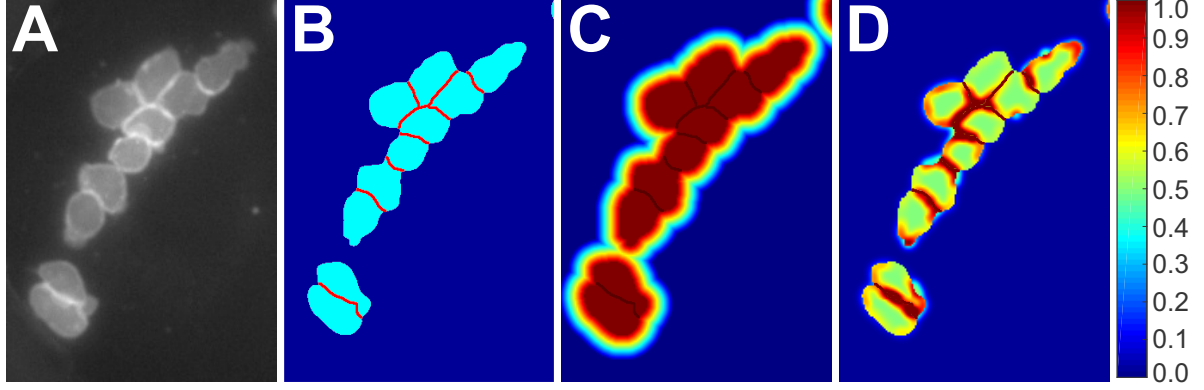


Source: PEÑA *et al.* (2018)

In Equation 3.4, $\tau = \sup_{p \in \Gamma_H} \phi_K(p)$ represent a distance normalization factor. In this work it was iterated F_σ twenty times to broadly propagate ω^c . This last term measures the shape complexity for each cell t by computing distances to the skeletons of the mask and of its concave complement to assess how narrow are the regions around the contours. Small distances give rise to large weights. The value of τ governs the distance tolerance and it is application dependent. Note that SAW assigns large weights to small objects without any further processing or loss function change contrary to what has been proposed by Zhou *et al.* (ZHOU *et al.*, 2017b). Examples of SAW for single and touching cells are shown in Figure 12D and Figure 13D respectively.

Both proposed weights maps acts as an specialization of the general Weighted Cross Entropy. Each proposal creates a family of loss functions $\{\mathcal{L}_\varrho\}_\varrho$ representing different error surfaces.

Figure 13 – An example of clustered cells is shown in (A). The weight maps, from left to right, are the (B) class balancing weight map ω^B , the (C) proposed distance transform based weight map ω^{DWM} , and the (D) proposed shape aware weight map ω^{SAW} . Color code is normalized to maximum weight value with red representing higher weights and blue small weights.

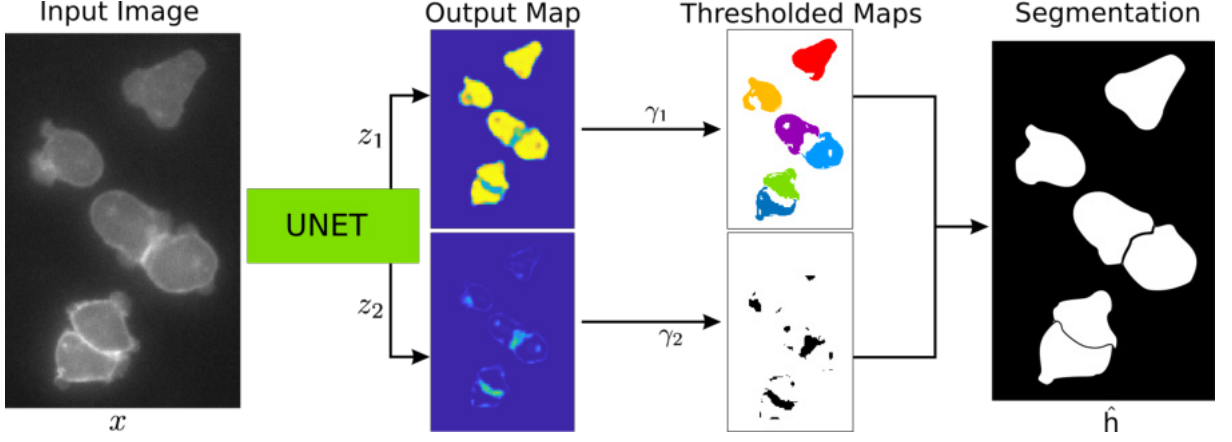


Source: PEÑA *et al.* (2018)

3.2.3 Assignment of Touching Pixels

Because the task is binary segmentation, the touching pixels identified by the network, according to the generated probability map z , need to be distributed to adjacent cells. The usual approach is to classify with Maximum a Posteriori (MAP) where the segmentation is obtained with $\hat{h}(p) = \arg \max_l z_l(p)$. However, since pixels in the touching and cell regions of an image have similar intensity distributions, it is expected some classification confusion in these areas. Therefore, a hard classification strategy different than MAP is needed. This is done in Algorithm 2 where each pixel p , for which it has been determined that $p \in \hat{h}^2$, is assigned to its closest adjacent cell. The method uses two given thresholds γ_1 and γ_2 as decision rules to build the final binary segmentation \hat{h} where \hat{h}^1 contains the segmented cell masks. The threshold γ_2 determines touching pixels and γ_1 determines cell masks: **if** $z_2(p) > \gamma_2$ **then** $p \in \hat{h}^2$, and **if** $z_1(p) > \gamma_1$ **and** $z_2(p) \leq \gamma_2$ **then** $p \in \hat{h}^1$. The rest of the pixels are assigned to the background class. For obtaining a binary segmentation, all pixels in touching class are assigned to the closest mask always they are not equidistant to two different cells (steps 4-9 Algorithm 2). As a final step, morphological hole filling and small object deletion are applied to eliminate spurious regions. A diagram of the proposed thresholded maps post-processing is shown in Figure 14.

Figure 14 – Overall segmentation scheme with touching pixels assignments and thresholded maps approach.



Source: The author (2019)

Algorithm 2: Pixel class assignment

Input: z, γ_1, γ_2

Output: h'

```

1 if  $z_2(p) > \gamma_2$  then  $p \in \hat{h}^2$ ;
2 if  $z_1(p) > \gamma_1$  and  $p \notin \hat{h}^2$  then  $p \in \hat{h}^1$ ;
3 for all  $p$  such that  $p \in \hat{h}^2$  do
4    $q_0 \leftarrow \arg \min_{q_0 \in \hat{h}^1} \|p - q_0\|_2^2$ ;
5    $q_1 \leftarrow \arg \min_{q_1 \in \hat{h}^1 \text{ and } q_1 \neq q_0} \|p - q_1\|_2^2$ ;
6   if  $\|p - q_0\|_2^2 = \|p - q_1\|_2^2$  then
7      $\hat{h}(p) \leftarrow 0$  // equidistant touching pixels belongs to background
8   else
9      $\hat{h}(p) \leftarrow 1$  // the rest is assigned to foreground

```

3.3 EXPERIMENTS AND RESULTS

For demonstrating the advantages of the proposals in the training process, a manually curated T-cell segmentation dataset containing thirteen images of size 1024×1024 pixels was used. The data was augmented with warping and geometrical transformations (rotations, random crops, mirroring, and padding) in every training iteration. Ten images were used for training (RONNEBERGER; FISCHER; BROX, 2015). Here it is called UNET2 to the use of U-Net with a binary ground truth and the near objects weights maps from (RONNEBERGER; FISCHER; BROX, 2015). The same model with 3 classes label augmentation is referred to as UNET3. DWM and SAW refer to training U-Net architecture using, respectively, the proposed ω^{DWM} and ω^{SAW} weights maps. In this work it is used FL for referring to the Focal loss function (ZHOU et al., 2017b) which was applied in the segmenta-

tion of small objects using an adaptive weight map. All networks were equally initialized with the same normally distributed weights using Xavier’s method (GLOROT; BENGIO, 2010), *e.g.*, fixed seed for random numbers generators led to the same initialization for the weights of all networks. After training, binary segmentations are created using the pixel assignment algorithm described in Section 3.2.3. Please note that, for guaranteeing same conditions during training, the order of minibatches, data augmentation, network architecture, and initialization is ensured to be same for all networks during the entire training, and the differences in the solutions are only obtained because the loss functions define different error surfaces.

To compare computed contours to ground truth the F1 score was adopted. To allow small differences in the location of contours, an uncertainty radius $\iota \in [2, 7]$, measured in pixels, is used for the F1 calculation, following (ESTRADA; JEPSON, 2009). Table 2 compares the results from different methods for several *radii*. For all *radii* the proposed methods outperform the other approaches in terms of segmentation results. Better contour adequacy is obtained mainly with SAW for $\iota < 6$ in the training set. In the testing phase, however, higher generalization can be observed with SAW for all *radii*. The proposed DWM was ranked second best.

Table 2 – F1 scores for different contour uncertainty *radii*. The method SAW performed better than others, with DWM the second best on training data.

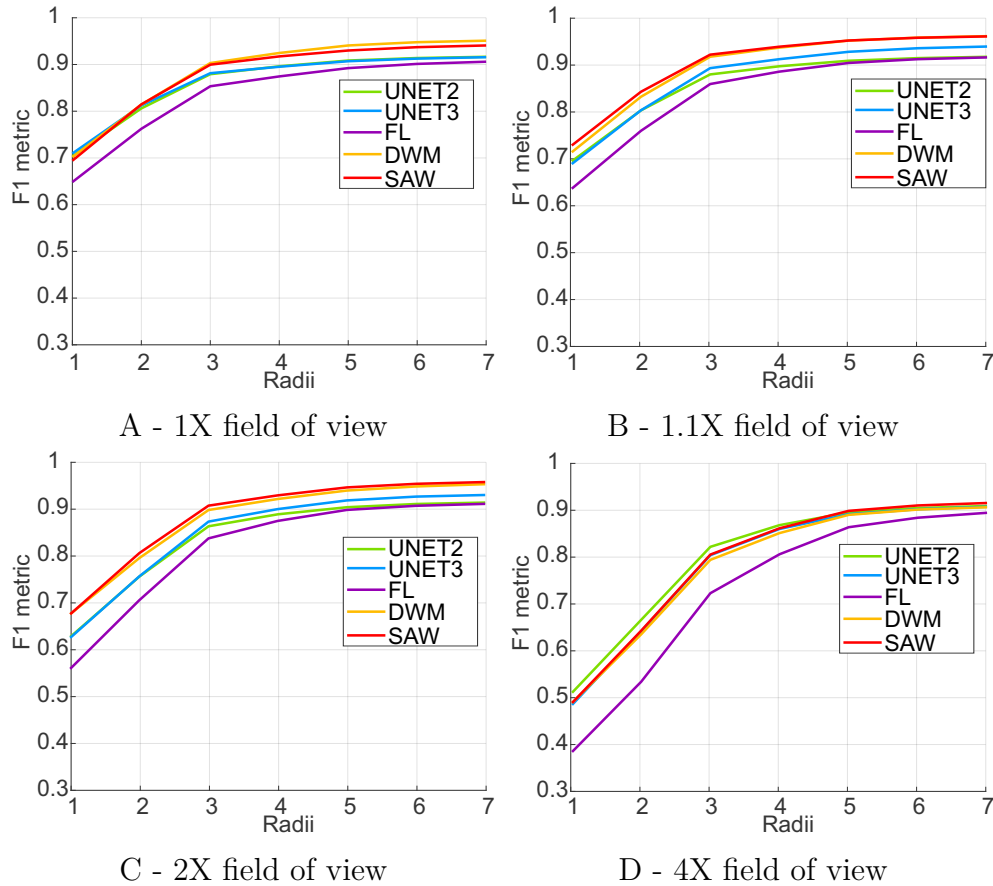
Radius	2	3	4	5	6	7
Training set						
UNET2	0.7995	0.8762	0.8936	0.9053	0.9109	0.9137
UNET3	0.7997	0.8896	0.9087	0.9244	0.9320	0.9356
FL	0.7559	0.8557	0.8821	0.9007	0.9087	0.9125
DWM (Proposed)	0.8285	0.9139	0.9333	0.9484	0.9546	0.9578
SAW (Proposed)	0.8392	0.9183	0.9353	0.9485	0.9544	0.9573
Testing set						
UNET2	0.6158	0.7116	0.7368	0.7627	0.7721	0.7828
UNET3	0.6529	0.7505	0.7770	0.8021	0.8158	0.8238
FL	0.5434	0.6566	0.6958	0.7263	0.7414	0.7505
DWM (Proposed)	0.6749	0.7847	0.8156	0.8398	0.8531	0.8604
SAW (Proposed)	0.7332	0.8298	0.8499	0.8699	0.8800	0.8860

Source: PEÑA *et al.* (2018)

Plots of F1 score for different *radii* and fields of view are shown in Figure 15 for all methods. The experiment included image sizes 1024×1024 , 900×900 , 500×500 , and 250×250 pixels corresponding to 1X, 1.1X, 2X and 4X fields of view. Objects look smaller to the network when the resolution is reduced compromising the segmentation.

FL performed poorly when the field of view was increased. In most of the cases best performance was obtained using SAW and DWM.

Figure 15 – F1 scores for radii $\iota \in [1, 7]$ in (A) 1X, (B) 1.1X, (C) 2X, and (D) 4X field of view size for each model. F1 values were consistently better for SAW and DWM in most of the cases.



Source: PEÑA *et al.* (2018)

To measure the cell detection success, every recognized cell with Jaccard Index (CSURKA *et al.*, 2004) greater than 0.5 is counted as True Positive. Contrary to the Intersection over Union (IoU) metric for detection (HOSANG *et al.*, 2016) which uses bounding boxes, the Jaccard Index calculates the instance adequacy from object segmentation. Precision, Recall and F1 are calculated as described by Ozdemir *et al.* (ÖZDEMİR *et al.*, 2010). Table 3 shows the recognition metrics for all the approaches. In this regard it can be seen that the proposed methods outperform with high margin all the other methods. The SAW method showed an improvement of 6% over DWM for the training set and an improvement of 14% for the testing set. UNET2 behaved poorly in clustered cells, unable to separate them. The combination of background and touching regions by UNET2 into a single class prevented the proper detection of individual cells. These encouraging results suggest that combining multiclass learning with shape aware weights maps might be advantageous to achieve improved segmentation results. On the other hand, differences between training

and testing results restates that the small size of the training data is harmful for obtaining a good generalization.

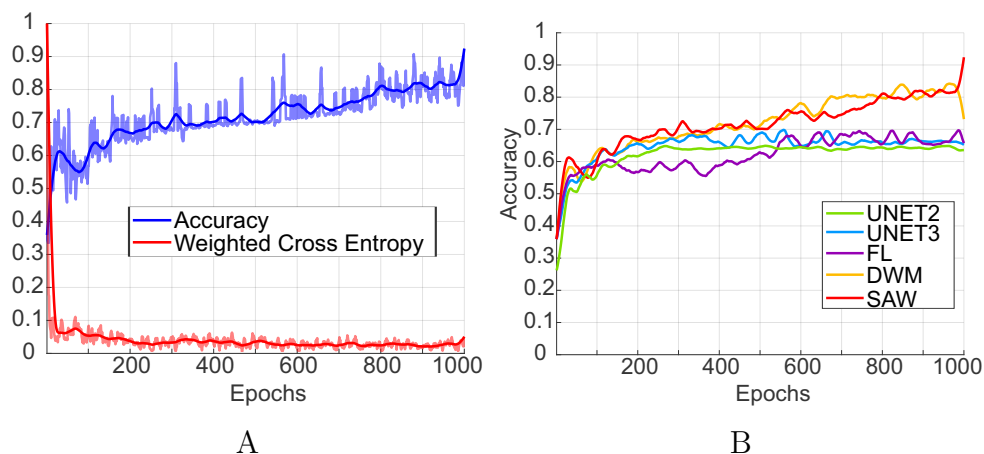
Table 3 – Detection metrics for Jaccard Index above 0.5 is much pronounced for SAW meaning it can detect more cells than the other methods.

	UNET2	UNET3	FL	DWM (Proposed)	SAW (Proposed)
Training set					
Precision	0.6506	0.7553	0.7276	0.8514	0.8218
Recall	0.4187	0.6457	0.4076	0.7191	0.8567
F1 metric	0.5096	0.6962	0.5225	0.7797	0.8389
Testing test					
Precision	0.5546	0.7013	0.6076	0.7046	0.8113
Recall	0.2311	0.3717	0.2071	0.5195	0.6713
F1 metric	0.3262	0.4858	0.3089	0.5980	0.7347

Source: PEÑA *et al.* (2018)

SAW training convergence can be viewed in Figure 16A with training weighted accuracy shown in blue and loss in red. Even when loss oscillates during optimization product to a high learning rate value ($lr = 0.1$) and SGD behavior, a tendency for it to decrease can be observed along the training. Loss diminution reflects directly in accuracy where a high trend to increase is observed. Models accuracies are compared in Figure 16B suggesting that the proposed loss functions improve the optimization during learning process with respect to previous approaches.

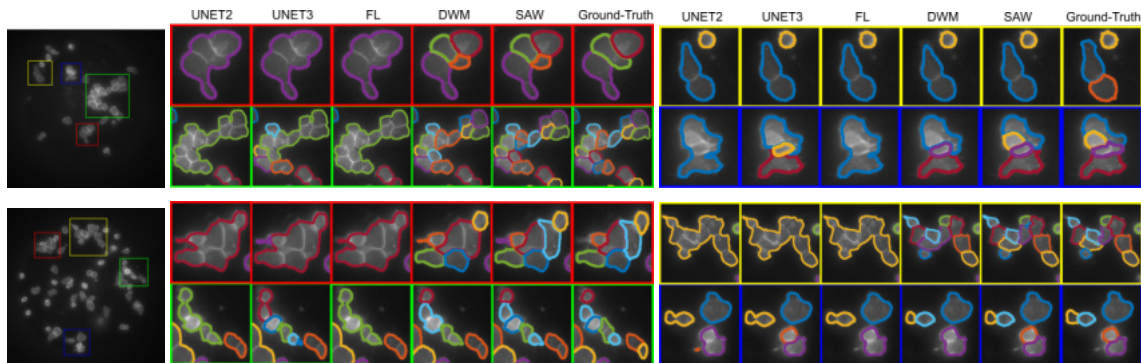
Figure 16 – Class weighted accuracy (blue) and Weighted Cross Entropy (red) of SAW network for every epoch are shown in (A). In (B) it is observed the models accuracy during training with outperforming rates of proposed DWM and SAW.



Source: PEÑA *et al.* (2018)

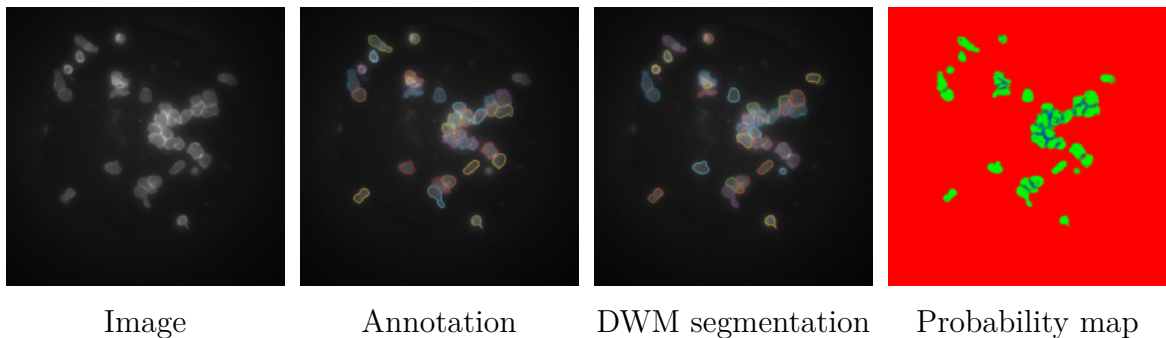
In Figure 17 it can be observed some segmentations results highlighting the improvement of proposed methods over the others. An example of wrong cluster division product to weak boundaries cases can be observed in the yellow panel of the first row of the Figure 17. The resulting segmentation over an image is shown in Figure 18. Although the segmentation is very similar to the ground truth, some uncertainty can be seen in the probability map. This occurs because these loss functions assume a fully supervised annotation, *e.g.*, no mistakes in the annotation. However, as can be seen in the example the top-right cell is missing in the annotation and the network struggles to both learn to segment and discard this object, causing confusion in the final probability map. Additional examples of segmentations obtained with each loss function can be seen in Figure 19. T-cells images used in the experiments were acquired by specialist from the Rothenberg Lab at the California Institute of Technology.

Figure 17 – Examples of segmentation contour obtained with UNET2 ($\gamma_1 = 0.50, \gamma_2 = 0.06$), UNET3 ($\gamma_1 = 0.40, \gamma_2 = 0.06$), FL ($\gamma_1 = 0.50, \gamma_2 = 0.16$), DWM ($\gamma_1 = 0.45, \gamma_2 = 0.06$), and SAW ($\gamma_1 = 0.50, \gamma_2 = 0.11$) and ground truth delineations for eight regions of two images. Results are for the best combination of γ_1 and γ_2 thresholds. Contour colors are merely used to illustrate the separation of individually segmented regions.



Source: PEÑA *et al.* (2018)

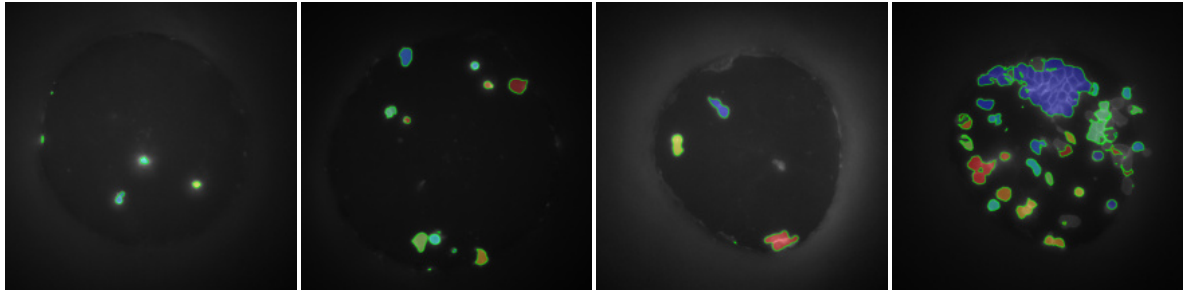
Figure 18 – Example of image with its corresponding annotation and obtained probability map and segmentation using DWM. Probability map is show as an RGB image whith background (red), cell (green) and touching (blue) classes.



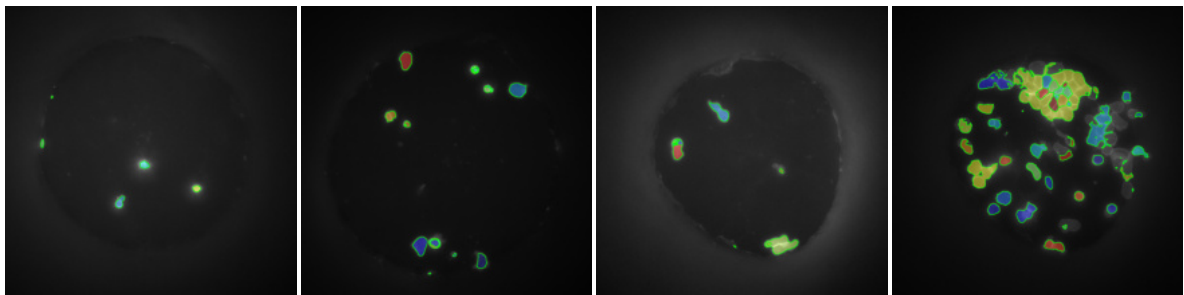
Source: The author (2019)

Figure 19 – Instance segmentation of clustered cells of four images. Instances colors and green contour are merely used to illustrate the separation of individually segmented regions.

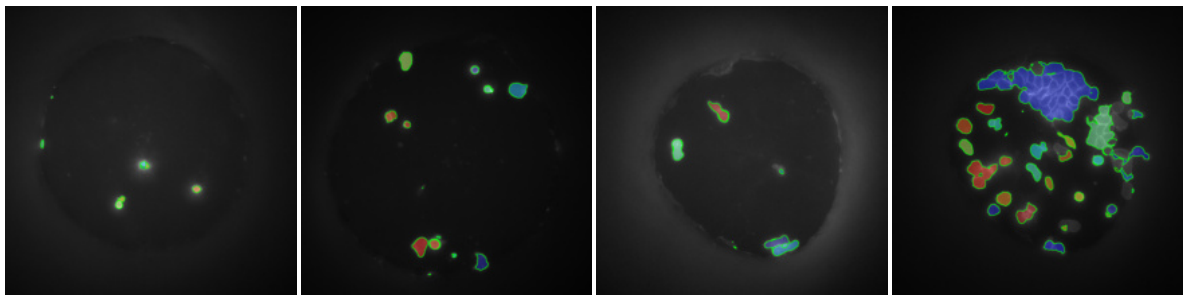
Binary Weighted Cross Entropy with UNET weight map



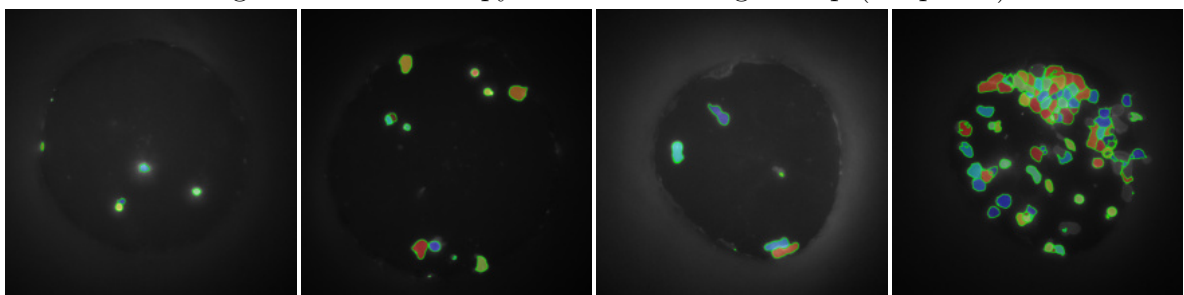
Focal loss



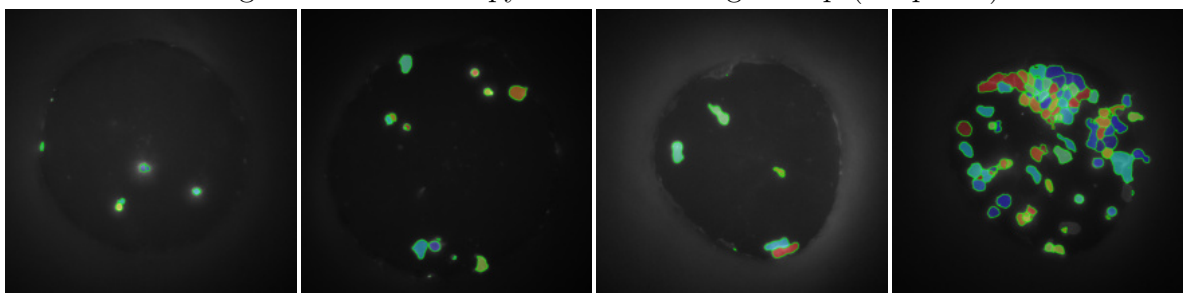
Weighted Cross Entropy with UNET weight map



Weighted Cross Entropy with DWM weight map (Proposed)



Weighted Cross Entropy with SAW weight map (Proposed)



3.4 CONCLUSIONS

In this chapter a new touching-based class augmentation and two new information theory-based loss functions derived from Weighted Cross Entropy were introduced. Also, a new thresholded maps post-processing was proposed for transforming the obtained probability map into a binary segmentation in cases of dime probability maps. Experiments performed over a challenging T-cells segmentation dataset showed the feasibility of the proposal. In particular, an increase of the performance for both segmentation and instances detection was observed with DWM and SAW for fixed training conditions and by only varying the loss function. These are very encouraging results because they show that finding better performed models is not conditioned on the ability to craft new architectures, but also by modeling an appropriated loss function. Despite the improvements in the results, differences between train and test sets performances suggest that a small training dataset is not sufficient to obtain a high generalization. Weak supervisions also showed to harm the learning process, leading to confusion between classes in the resulting probability map.

4 A WEAKLY SUPERVISED METHOD FOR INSTANCE SEGMENTATION OF BIOLOGICAL CELLS^[2]

In this chapter a new weakly supervised loss function is presented to perform instance segmentation of cells present in microscope images. The motivation is that, because annotation of biomedical images can be scarce, incomplete, and inaccurate, the optimized error surface usually leads to solutions with poor performance and bad generalization. To overcome the curse of reduced learning data, a loss function operating in three classes that drives the optimizer to classify underrepresented regions and promote separating adjacent cells properly is proposed. Different to binary segmentation where only a binary mask is available for each image, the instance segmentation problem deal with separated masks for each object.

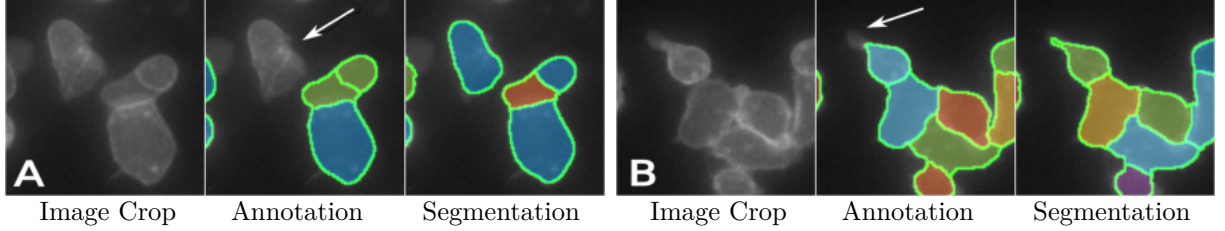
4.1 INTRODUCTION

Instance segmentation is predominantly done in biomedical image analysis as it allows characterizing individual objects of interest in an image. For example, in cell biology studies one generally needs to quantify signals, *e.g.*, protein concentration, on a per cell basis. This suggests segmenting many individual cells in the images when fully supervised training is considered. However, full annotation is expensive, time-consuming, and it is often inaccurate and incomplete when done at the lab (see Figure 20). These problems can be exacerbated when only a few specialists can perform annotations or when an annotation protocol is not in place.

Here, as in the previous chapter, the cells present complex shapes, *e.g.*, with small necks, slim invaginations, and protrusions, requiring a more attentive to details segmentation model when compared to round, mostly convex shapes. Also, small edges and slim parts, equally important for the segmentation result can be easily dismissed by the optimizer if their contribution is not explicitly accounted for and on par with other more dominant regions.

^[2] Fidel A. Guerrero Peña; Pedro Marrero Fernandez; Tsang Ing Ren; Alexandre Cunha. Centro de Informática, Universidade Federal de Pernambuco, Brazil; Center for Advanced Methods in Biological Image Analysis, California Institute of Technology, USA. Published in: Medical Image Computing and Computer-Assisted Intervention Workshop (MICCAI MIL), 2019.
Reprinted/adapted by permission from Springer Nature Customer Service Centre GmbH: Cham, Springer. Fidel A. Guerrero Peña, Pedro Marrero Fernandez, Tsang Ing Ren, Alexandre Cunha. A Weakly Supervised Method for Instance Segmentation of Biological Cells. In: Domain Adaptation and Representation Transfer and Medical Image Learning with Less Labels and Imperfect Data by Wang Q. et al., ©2019.

Figure 20 – Examples of (A) incomplete and (B) inaccurate annotations of training images, pointed by arrows above. The goal of weakly supervised methods is to be able to segment well under uncertainty and limited data as shown in the examples of a missing cell and slim part, respectively, in the right panels of (A) and (B).



Source: PEÑA *et al.* (2019b)

4.2 MULTICLASS SHAPE-BASED WEAKLY SUPERVISED LOSS FUNCTION

Despite the proven results of related approaches for weakly supervised instance segmentation task, the methods rely on big datasets. For dealing with this problem here it is used U-Net as in the previous chapter due to its simplicity and proven results in small datasets. Then, the instance segmentation problem is formulated as a semantic segmentation problem where objects segmentation and separation of cells are obtained at once.

In this context the training instance segmentation set is defined as $S = \{(x_j, g_j)\}_{j=1}^N$ where $x_j: \Omega \rightarrow \mathbb{R}^+$ is a single channel gray image defined on the regular grid $\Omega \in \mathbb{R}^2$, and $g_j: \Omega \rightarrow \{0, \dots, m_j\}$ its instance segmentation ground truth map which assigns to a pixel $p \in \Omega$ a unique label $g_j(p)$ among all $m_j + 1$ distinct instance labels, one for each object, including background, labeled 0. For a generic (x, g) , $t_i = \{p \mid g(p) = i\}$ contains all pixels belonging to instance object i , hence forming the connected component (segment) of object i . Due to label uniqueness, $t_i \cap t_j = \emptyset, i \neq j$, a pixel cannot belong to more than one instance thus satisfying the panoptic segmentation criterion (KIRILLOV *et al.*, 2019). Let $h: \Omega \rightarrow \{0, \dots, C\}$ be a semantic segmentation map, obtained using g , which reports the semantic class of a pixel among the $C + 1$ possible semantic classes, and $y: \Omega \rightarrow \mathbb{R}^{C+1}$ its one hot encoding mapping. That is, for vector $y(p) \in \mathbb{R}^{C+1}$ and its l -th component $y_l(p)$, it is obtained that $y_l(p) = 1$ iff $h(p) = l$, otherwise $y_l(p) = 0$. Let $n_l = \sum_{p \in \Omega} y_l(p)$ be the number of pixels of class l , and $\eta_k, k \geq 1$, the $(2k + 1) \times (2k + 1)$ neighborhood of a pixel in Ω . In this work it was adopted $k = 2$.

4.2.1 From Instance to Semantic Ground Truth

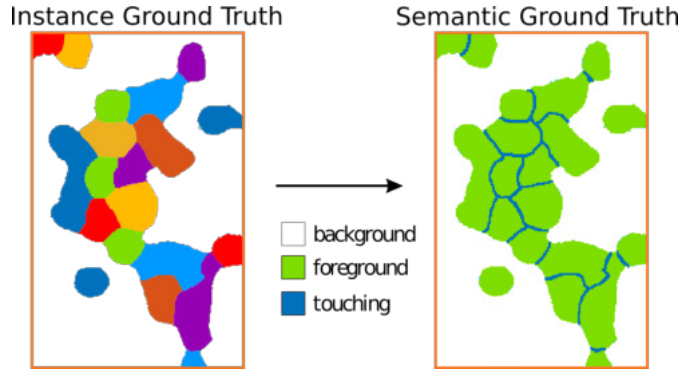
To transform an instance ground truth to semantic ground truth, the three semantic classes scheme of the previous chapter was adopted, *e.g.*, image background, cell interior, and touching region between cells. This seems to be feasible because the intensity distribution of input images in those regions remains multi-modal. The semantic ground truth

h is defined as

$$h(p) = \begin{cases} 0 & \text{if } g(p) = 0 & \text{-- background} \\ 2 & \text{if } \sum_{p' \in \eta_k(p)} [g(p') \neq g(p)] \cdot [g(p') \neq 0] > 1 & \text{-- touching} \\ 1 & \text{otherwise} & \text{-- cell} \end{cases} \quad (4.1)$$

where $[\cdot]$ refers to Iverson bracket notation (BERMAN; TRIKI; BLASCHKO, 2018): $[b] = 1$ if the boolean condition b is true, otherwise $[b] = 0$. Equation 5.2.1 assigns class 0 to all background pixels, assigns class 2 to all pixels whose neighborhood η_k contains pixels of another connected component, and assigns class 1 to cell pixels not belonging to touching regions. For an example of transforming an instance ground truth to semantic see Figure 21.

Figure 21 – Example of transformation from instance g to semantic h segmentation ground truth.



Source: PEÑA *et al.* (2019b)

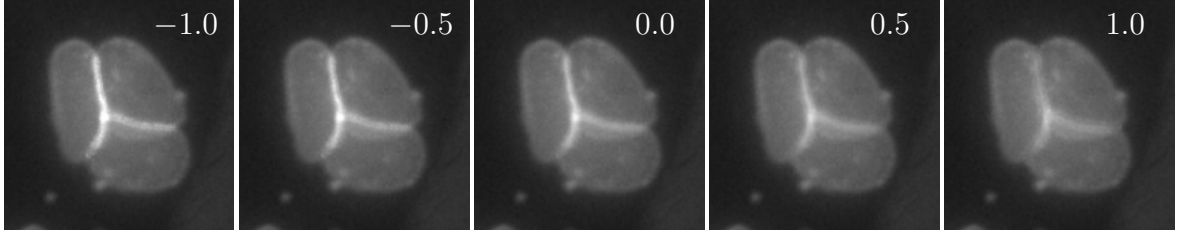
4.2.2 Touching Region Augmentation

Touching regions have the lowest pixel count among all semantic classes, having few examples to train the network. They are in general brighter than their surroundings, but not necessarily, with varying values along its length. To train with a larger gamut of touching patterns, including weak edges, it was augmented existing ones by modulating their pixel values according to the following expression: $x_a(p) = (1 - a) \cdot x(p) + a \cdot \tilde{x}(p)$, applied only when $h(p) = 2$, where \tilde{x} is the median filtered image of x (a 7×7 window was used). For values $a < 0$ ($a > 0$) is increased (decreased) the contrast. During training, the parameter a was generated following a uniform random distribution, $a \sim U(-1, 1)$. An example of this modulation is shown in Figure 22.

4.2.3 Robust Weight Maps for Weak Annotations

Cross Entropy loss function is the most common loss function used when training U-Net as in this work. However, its usefulness is limited for weakly supervised problems because

Figure 22 – Example of contrast modulation around touching regions. In this figure, higher, $a = -1.0, -0.5$, and lower, $a = 0.5, 1.0$, contrast examples are shown. $a = 0$ gives the original image.



Source: PEÑA *et al.* (2019b)

the network also learns errors in the annotation. The Weighted Cross Entropy (WCE) (RONNEBERGER; FISCHER; BROX, 2015) is a generalization of this function where a pre-computed weight map assigns to each pixel its importance for the learning process. This allows creating a customizable loss function family $\{\mathcal{L}_\rho\}_\rho$ for specific tasks like weakly supervised instance segmentation. As expressed before the WCE is defined as:

$$\mathcal{L}(y, z) = - \sum_{l=0}^C \sum_{p \in \Omega} \omega_\rho(p) \cdot y_l(p) \cdot \log z_l(p) \quad (4.2)$$

Weak annotations in this work are in the form of incomplete and inaccurate segments. Given that recent results have shown that training data for pixel-level tasks (*e.g.*, denoising, segmentation) are statistically correlated within an image, and that selecting a small set of pixels for training might be sufficient (LI; ARNAB; TORR, 2018), it is proposed a contour based weight map to assist in the instance segmentation with weak supervision.

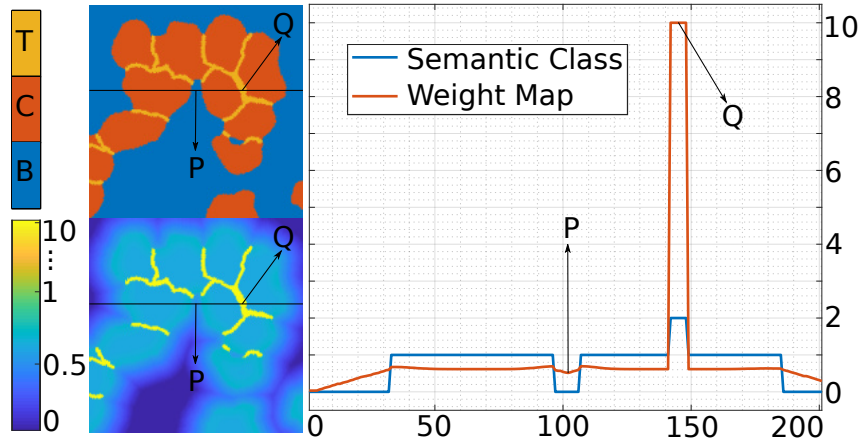
Let $R(u) = u^+$ be the rectified linear function, ReLu, and $\varphi_\beta(u) = R(1 - u/\beta)$, $u \in \mathbb{R}$, a rectified inverse function saturated in $\beta \in \mathbb{R}^+$. The proposed Triplex Weight Map, W^3 is computed as:

$$\omega_{\beta, \nu, \sigma}(p) = \begin{cases} \nu/n_0 + \nu \cdot \varphi_\beta(\phi_h(p)) / n_1 & \text{if } h(p) = 0 \\ \nu/n_1 + \nu \cdot \varphi_\beta(\phi_K(p)) & \text{if } h(p) = 1, p \in \Gamma \\ \nu/n_1 + \omega_{\beta, \nu, \sigma}(\zeta_\Gamma(p)) \cdot \exp(-\phi_\Gamma^2(p)/\sigma^2) & \text{if } h(p) = 1, p \notin \Gamma \\ \nu/n_2 & \text{if } h(p) = 2 \end{cases} \quad (4.3)$$

where Γ is the cell contour, ϕ_h is the distance transform over h that assigns to every pixel its Euclidean distance to the closest non-background pixel, ϕ_K and ϕ_Γ are, respectively, the distance transforms with respect to the skeleton of cells and cell contours, and $\zeta_\Gamma: \Omega \rightarrow \Omega$ gives the closest contour pixel to a given pixel p . The W^3 model almost disregard all background pixels distant at least β to a cell contour by setting $\omega_\rho(p) = \nu/n_0$. This way, cells that are eventually not annotated and located beyond β from annotated cells have a very low importance during training, since, by design, the weights on those unannotated regions are close to zero. The goal behind the recursive expression for foreground pixels

is to create a set of Gaussians centered on each pixel of the contour. The Gaussians have amplitudes which are inversely proportional to their distances to the cell skeleton. The weight at a foreground pixel is simply the value of the Gaussian centered on the point closest to this pixel on the contour. The touching region is assigned a constant weight for class balance, larger than all other weights. Figure 23 shows an example of weights values for a fixed row of a semantic ground truth and its corresponding W^3 weight map.

Figure 23 – Example of semantic classes and weights values for a row (black line) in h and $\omega_{\beta,\nu,\sigma}$ with $\beta = 30, \nu = 10$, and $\sigma = 3$. A semantic ground truth and its corresponding weight map are shown in top left and bottom left images. Points P and Q denotes background tip and touching region respectively.



Source: The author (2019)

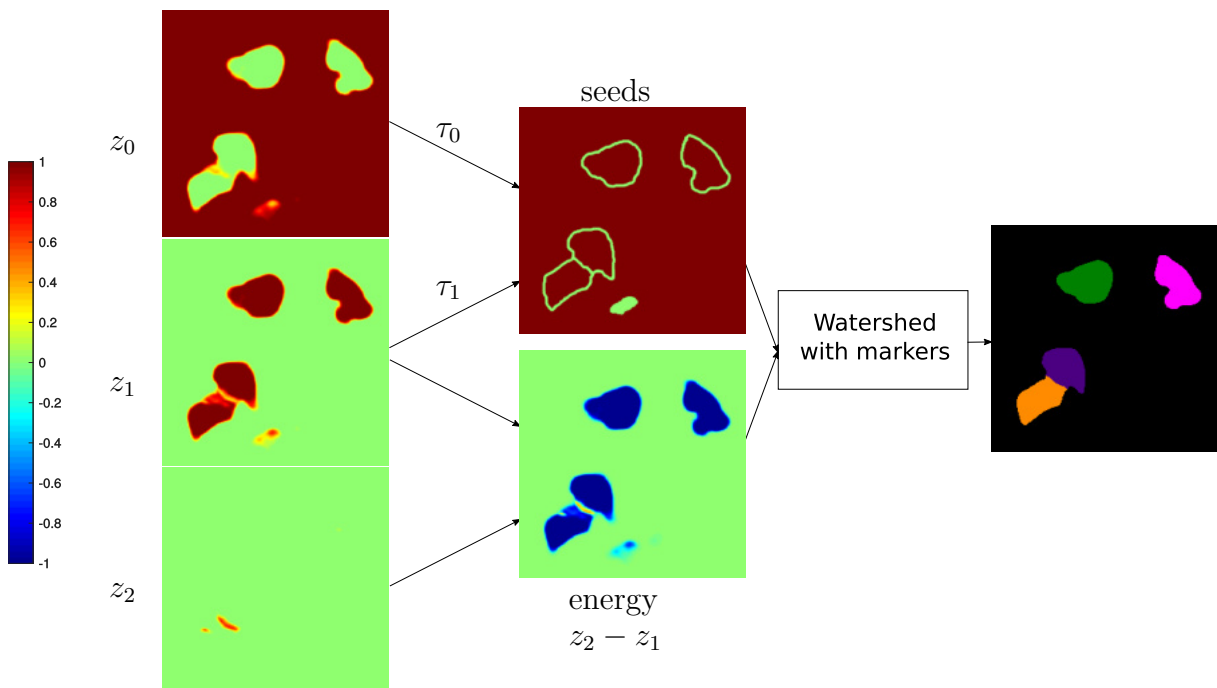
4.2.4 From Semantic to Instance Segmentation

After training the network for semantic segmentation, the transformation from semantic to panoptic instance segmentation is performed. For this, a decision rule over the output probability map z needs to be applied to hard classify each pixel, \hat{h} . However, since pixels in the touching and cell regions of an image have similar intensity distributions it is expected some classification confusion in those regions if MAP is used. A simple approach is to use an instance segmentation version of thresholding (TH) post-processing (Algorithm 2) as a decision rule where the parameters γ_1 and γ_2 control, respectively, the assignment of pixels to cell and touching classes: $\hat{h}(p) = 2$ if $z_2(p) \geq \gamma_2$, and $\hat{h}(p) = 1$ if $z_1(p) \geq \gamma_1$ **and** $z_2(p) < \gamma_2$, and 0 otherwise. Finally, the estimated instance segmentation \hat{g} labels each cell segment \hat{t}_i and distributes touching pixels to the closest component.

$$\hat{g}(p) = \begin{cases} 0 & \text{if } \hat{h}(p) = 0 \\ i & \text{if } \hat{h}(p) = 1 \text{ and } p \in \hat{t}_i \\ \hat{g}(\zeta_{\Gamma}(p)) & \text{otherwise} \end{cases} \quad (4.4)$$

Another alternative for post-processing is to segment using the Watershed Transform (WT) with markers (MEYER, 1994). Here, WT is applied on a topographic map built from the probability map with seeds formed by pixels p satisfying $z_0(p) \geq \tau_0$ or $z_1(p) \geq \tau_1$. These seeds are basically the pixels in the background and cell regions whose probabilities are larger than given thresholds τ_0 and τ_1 . Then, the topographic map is the surface defined by the difference between touching and cell probability maps, $z_2 - z_1$. The overall procedure can be seen in Figure 24.

Figure 24 – Overall segmentation scheme of the proposed Watershed-based approach.



Source: The author (2019)

4.3 EXPERIMENTS AND RESULTS

To evaluate and validate the weakly supervised approach a U-Net was trained, as stated before, initialized with normally distributed weights using Xavier method (Glorot; Bengio, 2010). For a better comparison, all the seeds of random numbers generators were fixed, and therefore all methods began with the same initialization θ_0 and used the same mini-batch for training, including augmentations for the images. Then, the different solutions obtained with each loss function are uniquely affected by the morphology of the error surface. In the following the Lovász-Softmax loss function (Berman; Triki; Blaschko, 2018) ignoring the background class is referred as LSMAX. Here the Weighted Cross Entropy using class Balance Weight Map is called BWM, near objects weight map is called UNET (Ronneberger; Fischer; Brox, 2015), and the proposed Triplex Weight map is named W^3 . The per-class average of the probability maps obtained with BWM, UNET

and W^3 , followed by a softmax, is called here COMB. Despite changing the architecture is out of the scope of this thesis, the classical proposal-based method Mask R-CNN (MR-CNN) (HE et al., 2017) was included in the comparison for establishing a reference point with the state-of-the-art in instance segmentation.

All networks were trained over a cell segmentation dataset containing twenty-eight images of size 1024×1024 pixels with weak supervision in the form of incomplete and inaccurate annotations. The optimizer Adam (KINGMA; BA, 2015) with $lr = 10^{-4}$ was used for training. The number of epochs and minibatch size was 1000 and 1 respectively. For training purpose, random mirroring, rotations, warping, gamma, and touch contrast modulation data augmentations were applied.

For evaluation of the detection, the Precision (P05) and the Recognition Quality (RQ) (KIRILLOV et al., 2019) of instances with Jaccard index above 0.5 were used. To evaluate the segmentation, the Segmentation Quality (SQ) computed as the average Jaccard of matched segments (KIRILLOV et al., 2019) was computed. For overall evaluation of both detection and segmentation, it was used the Panoptic Quality (PQ) metric (KIRILLOV et al., 2019). Higher values of all these measurements implies better performance.

Because Thresholded Maps and Watershed post-processings depends on two parameters, it was performed the exploration over the parameters space. Table 4 shows a comparison of the different post-processing for the best combination of the parameters for Thresholds (TH) and Watershed (WT). Although Lovász-Softmax is one of the most promising loss functions, the small training dataset and minibatch size harmed the optimization process in earlier iterations resulting in poor performance. For most values of thresholds with TH post-processing, the average combination (COMB) improved the overall result because of the reduction of False Positives (see P05 column). Also, in most cases, the proposed W^3 approach obtained better SQ value than the other methods suggesting better contour adequacy. The best combination of parameters for each post-processing method is shown in Table 5.

Table 4 – Results for different post-processing methods where TH and WT represent the best thresholds combination for Threshold and Watershed post-processing respectively.

Methods	MAP				TH				WT			
	P05	RQ	SQ	PQ	P05	RQ	SQ	PQ	P05	RQ	SQ	PQ
LSMAX	0.3871	0.3236	0.7455	0.2408	0.4348	0.3119	0.7171	0.2286	0.4000	0.3149	0.7073	0.2237
BWM	0.6756	0.5580	0.8674	0.4858	0.8583	0.8504	0.8769	0.7476	0.8193	0.8405	0.8831	0.7437
UNET	0.6801	0.5381	0.8418	0.4556	0.8413	0.8508	0.8791	0.7492	0.8708	0.8600	0.8850	0.7621
W^3 (Proposed)	0.7384	0.6305	0.8721	0.5513	0.8477	0.8439	0.8994	0.7604	0.9028	0.8775	0.8995	0.7896
COMB(Proposed)	0.7587	0.6129	0.8698	0.5351	0.8952	0.8851	0.8908	0.7889	0.8925	0.8759	0.8944	0.7837

Source: PEÑA *et al.* (2019b)

Because of the overlapping between touching and cell intensities distributions, a softer classification was obtained in these regions (Figure 25). Then, wrong cell separation is

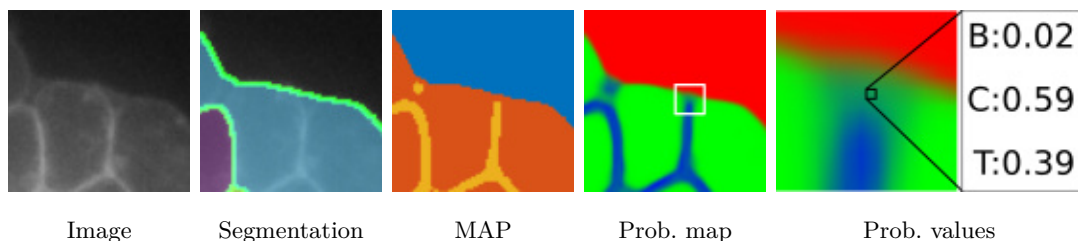
Table 5 – Best thresholds combinations for TH and WT in each method.

Methods	TH		WT	
	γ_1	γ_2	τ_0	τ_1
LSMAX	0.50	0.10	0.50	0.50
BWM	0.50	0.10	0.30	0.90
UNET	0.50	0.10	0.60	0.70
W^3	0.60	0.20	0.40	0.90
COMB	0.60	0.10	0.60	0.80

Source: The author (2019)

obtained as a result of the MAP post-processing leading to worse performance when compared with other post-processing approaches.

Figure 25 – Example of wrong cell separation using Maximum A Posteriori post-processing because of the confusion in the probability map. Probability maps are showed as RGB images where the channels correspond with background (B), cell (C) and touching (T) classes respectively.

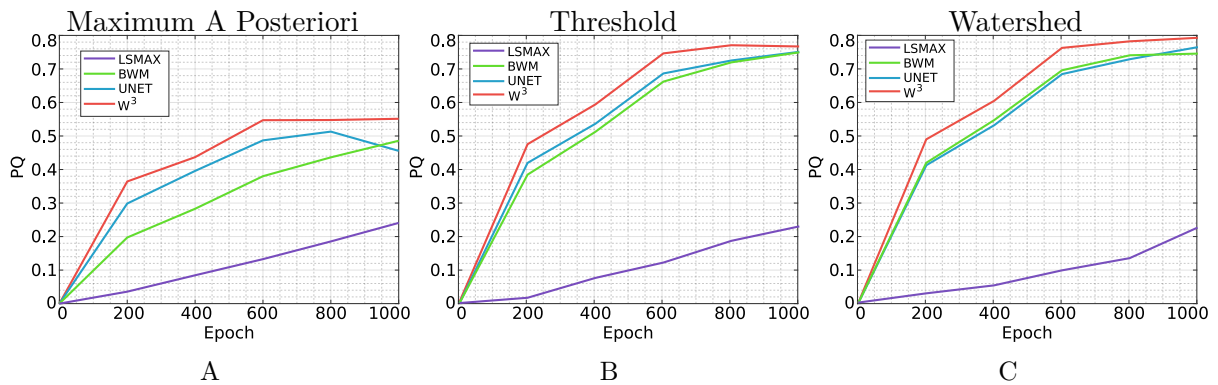
Source: PEÑA *et al.* (2019b)

The behavior in Table 4 remains during training as shown in Figure 26. As can be seen in the figure, the proposed W^3 method had a faster convergence for local minimal than related approaches. This is due to near to contour pixels miss-classifications have more importance for the optimization, creating peaks in the error surface. This mountains in the landscape serve as a constraint to possible wrong optimization paths while creating leaned surfaces that helps Adam optimizer to move faster.

An example of cell segmentation obtained over a test image with each approach, including MRCNN, is shown in Figure 27. In the experiments, MRCNN was able to detect correctly isolated and nearly adjacent cells (second row), but in a high-density cluster, both bounding box proposals and segmentation were deteriorated. BWM and UNET tend to miss-classify background pixels in neighboring cells (second row) because estimated contours are generally beyond cells membrane. W^3 had the better detection and segmentation performance with a slight but essential improvement of contour adequacy over COMB. An example of segmentation over two weakly annotated training images after the optimization ends is shown in the right panels of Figure 20AB.

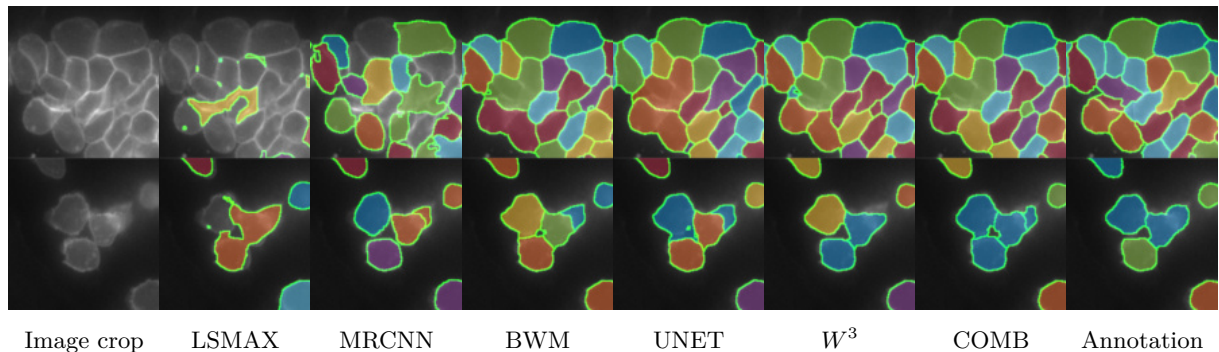
Because weakly supervised learning is very related to the generalization capability, it

Figure 26 – Panoptic Quality (PQ) metric during training for Lovász-Softmax (LSMAX), Weighted Cross Entropy with class Balance Weight Map (BWM), UNET weight map, and Triplex Weight Map (W^3) methods using (A) Maximum A Posteriori (MAP), (B) Thresholded Maps (TH), and (C) Watershed Transform (WT) post-processing.



Source: PEÑA *et al.* (2019b)

Figure 27 – Example of segmentation results with Lovász-Softmax (LSMAX), Mask R-CNN (MRCNN), Weighted Cross Entropy with class Balance Weight Map (BWM), UNET weight map, Triplex Weight Map (W^3) and average combination (COMB). Instances colors and green contour are merely used to illustrate the separation of individually segmented regions.

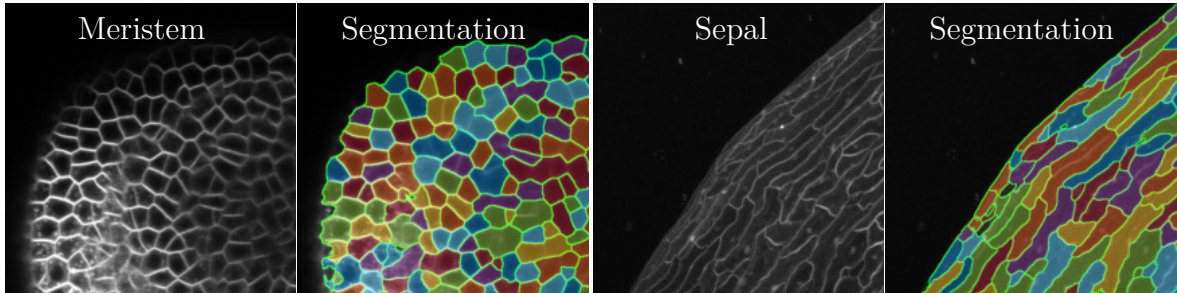


Source: PEÑA *et al.* (2019b)

was tested the robustness of W^3 for segmenting instances of near-domains images. An example of panoptic segmentation over a Meristem and Sepal image crops with the network trained over the T-cells dataset is showed in Figure 28. As can be seen, an acceptable solution is obtained, and presumably can be improved with a fine-tuning over a few annotated examples. The presented proposal probed to generalize well even when trained with a small dataset. The related used data augmentation made the neural networks suitable for segment similarly intensity distributed images of other domains.

In Figure 29 it can be seen a comparison between the obtained probability maps for both SAW (Section 3.2.2) and W^3 along with their corresponding instance segmentation. As show in the figure, the confusion between classes in SAW was significantly reduced with

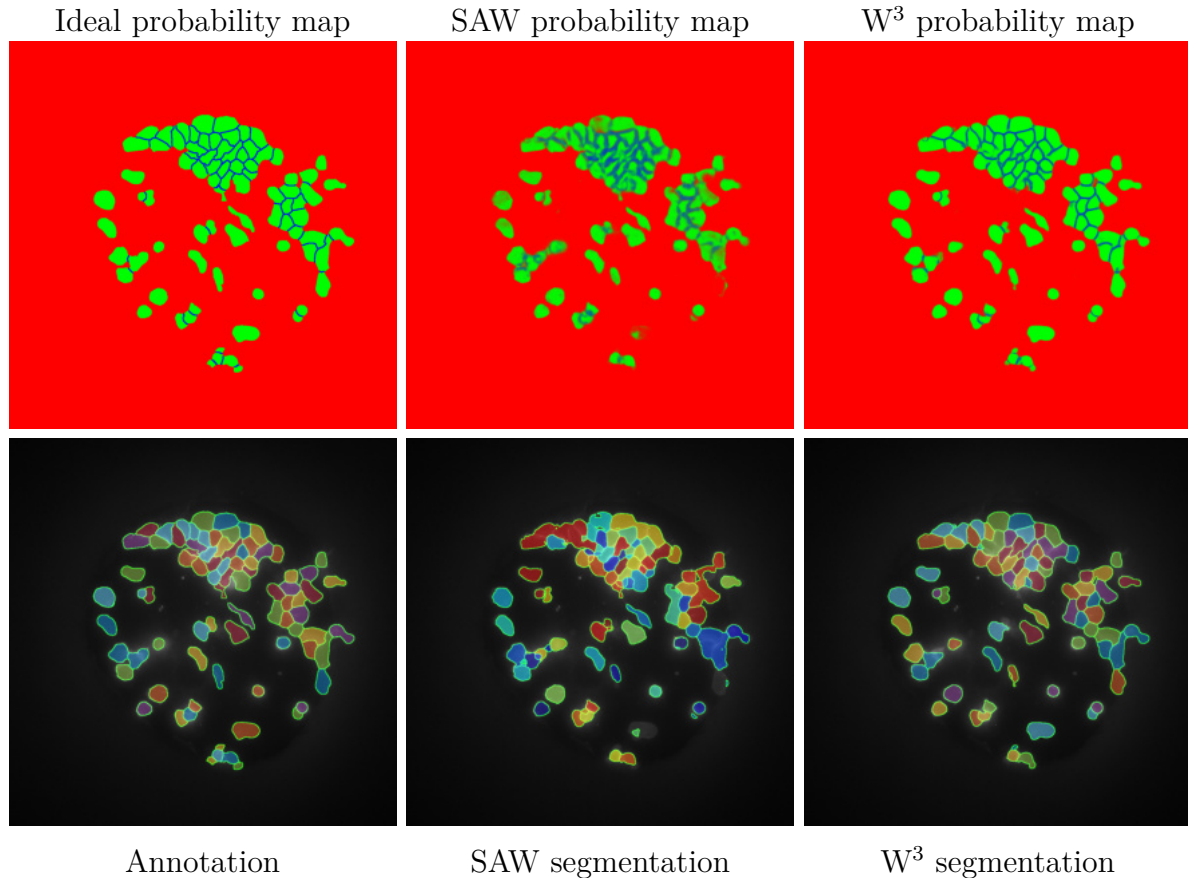
Figure 28 – Zero-shot panoptic segmentation for meristem and sepal images with W^3 approach trained for T-cells images. Instances colors and green contour are merely used to illustrate the separation of individually segmented regions.



Source: PEÑA *et al.* (2019b)

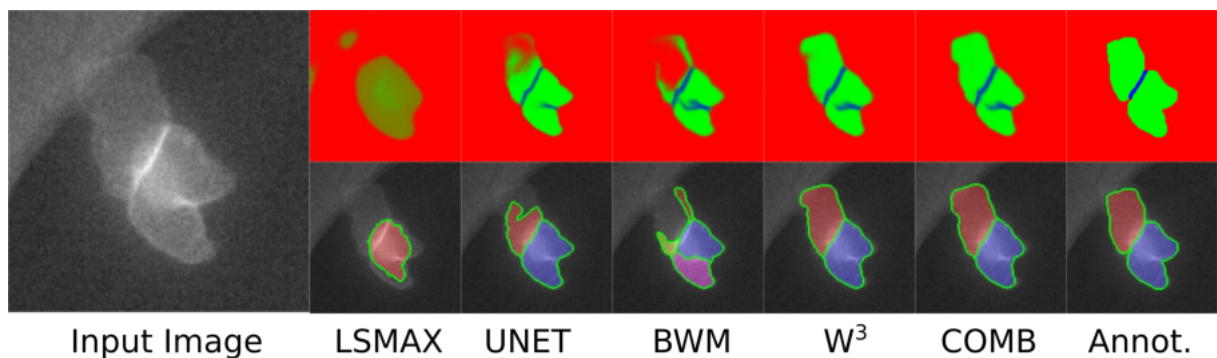
the new weakly supervised loss function. As result a higher number of cells are detected in this complex image. This behavior was also observed when compared with competitive approaches (Figure 30). Other examples of segmentations obtained with each method can be seen in Figure 31. T-cells images used in the experiments were acquired by specialist from Rothenberg Lab at the California Institute of Technology.

Figure 29 – Example of ideal probability map with its corresponding annotation and obtained probability maps and segmentation by SAW and proposed W^3 loss function. Instances colors and green contour are merely used to illustrate the separation of individually segmented regions. Probability map is show as an RGB image with background (red), cell (green) and touching (blue) classes.



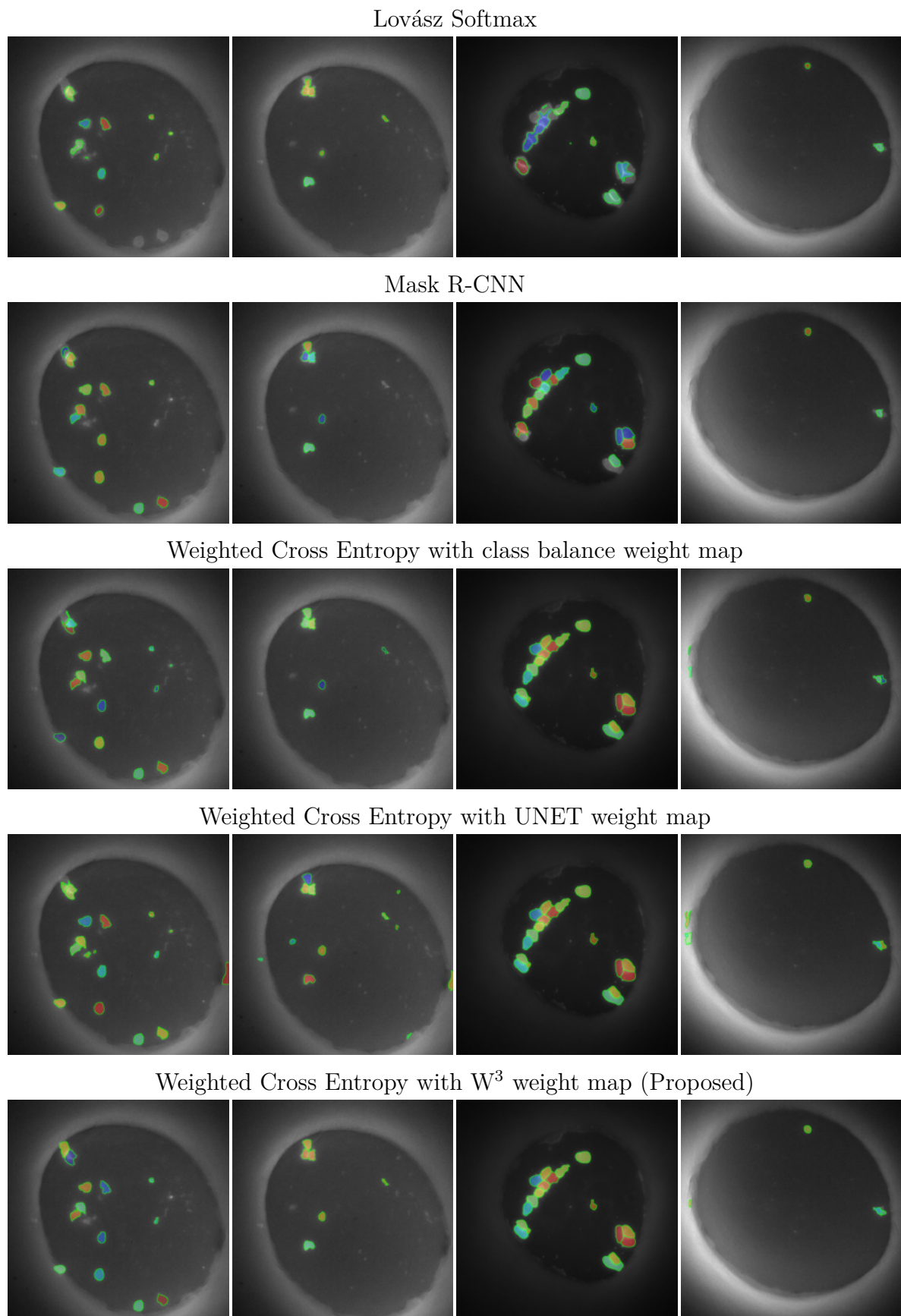
Source: The author (2019)

Figure 30 – Example of confusion reduction with W^3 when compared with LSMAX, UNET, and BWM. Probability map is show as an RGB image with back-ground (red), cell (green) and touching (blue) classes.



Source: The author (2019)

Figure 31 – Weakly supervised biomedical image instance segmentation of four images.



Source: The author (2019)

4.4 CONCLUSIONS

In this chapter a new weakly supervised extension of the Weighted Cross Entropy was introduced. The proposal, along with a new contrast modulation for data augmentation, enabled to train U-Net to effectively segment crowded cells with incomplete and inaccurate annotations. A new Watershed-based post-processing allowed the improvement of models performance when compared with other approaches like Thresholded maps and Maximum a Posteriori. The experiments showed a better detection rates, contour adequacy and faster convergence when the proposed W^3 loss function is used for training in comparison to others loss functions. Obtained results suggest that the methodology proposed here, *e.g.*, solving instance segmentation with a semantic approach, is more adequate in cases with highly clustered objects and small size datasets than the common Mask R-CNN approach. The proposed contrast modulation augmentation allowed the panoptic segmentation of cells in different domain but with near intensity distribution. Despite the performance improvement, the proposed Watershed and Thresholded maps post-processing depends on two parameters that needs to be manually adjusted. Although the general confusion in the output probability map was reduced, obtaining high probabilities for pixels in the minority touching class showed to remain challenging.

5 J REGULARIZATION IMPROVES IMBALANCED MULTICLASS SEGMENTATION^[3]

In this chapter a new loss formulation to improve the multiclass segmentation of clustered cells under weakly supervised conditions is proposed. A Youden regularization term is added to the Cross Entropy loss to enhance the separation of touching and immediate cells further while promoting sharp segmentation boundaries with high adequacy. This regularization intrinsically supports class imbalance. Another training class representing gaps between immediate cells is added to help the network to identify narrow gaps as background and no longer as touching regions. The proposal works for both 2D and 3D images, from bright field to confocal stacks, containing different types of cells.

5.1 INTRODUCTION

The long-term goal of this work has been the automatic segmentation of cells found in different modalities of microscopy images so that it can ultimately help in the quantification of biological studies. The task remains a challenge particularly when cells are densely packed in clusters exhibiting a range of signals and when training with a small number of weak annotations (see Figure 32A). Separation of clustered cells is specially difficult when shared edges have low contrast and are similar to cell interiors. Weak annotations, when incomplete and inaccurate, can harm the learning process as the optimizer might be confused when deciding if annotated and non annotated regions with same patterns must be segmented or not. Additionally, typical biomedical images have a highly class imbalance that increases with the number of dimensions. Then, balancing weights (RONNEBERGER; FISCHER; BROX, 2015; LONG; SHELHAMER; DARRELL, 2015; SUDRE et al., 2017) or *equivatches* (BERMAN; TRIKI; BLASCHKO, 2018) are required for obtaining acceptable solutions. Despite the usefulness of such balancing methods, the optimization may be difficult, especially for higher dimensions problems. This occurs because weak supervision in the minority class leads to a very noisy gradient causing instabilities during training. The solution proposed in this chapter aims at solving these problems with advances in loss formulation, class imbalance handling, multiclass classification, and weak annotation.

^[3] Fidel A. Guerrero Peña; Pedro Marrero Fernandez; Paul T. Tarr; Tsang Ing Ren; Elliot M. Meyerowitz; Alexandre Cunha. Centro de Informática, Universidade Federal de Pernambuco, Brazil; Howard Hughes Medical Institute, California Institute of Technology, USA; Division of Biology and Biological Engineering, California Institute of Technology, USA; Center for Data-Driven Discovery, California Institute of Technology, USA; Center for Advanced Methods in Biological Image Analysis, California Institute of Technology, USA. Available at <<https://arxiv.org/abs/1910.09783>>, 2019.

5.2 IMBALANCED MULTICLASS WEAKLY SUPERVISED LOSS FUNCTION

Let a d -dimensional single-channel image be defined as $x: \Omega \rightarrow \mathbb{R}^+$, where $\Omega \subset \mathbb{R}^d$ is a regular grid with $d \in \mathbb{Z}^+$. The elements $p \in \Omega$ are called pixels in the case $d = 2$ and voxels when $d = 3$. Then, the goal of panoptic segmentation is to assign to each element $p \in \Omega$ a semantic label, and instance identification if p belongs to a countable category (KIRILLOV et al., 2019). For learning to solve such task, a training set $S = \{(x_1, g_1), \dots, (x_N, g_N)\}$ is given, where for every image x its instance segmentation ground truth g is known. In general, an annotation g can be expressed as $g: \Omega \rightarrow \{0, \dots, m_j\}$ representing a d -dimensional mapping where $g(p) = 0$ for elements in the background, and assigning a unique label $g(p) > 0$ for each object within the image. Here, the task is cast as a semantic segmentation problem by modifying the approach proposed in Section 4.2.1 for transforming the instance annotation g into a semantic ground truth h , and generalizing for higher dimensions by using a $(2k + 1)^d$ neighborhood $\eta_k(p)$, $k \geq 1$, of an element $p \in \Omega$. Let $y: \Omega \rightarrow \mathbb{R}^{C+1}$ be the one-hot representation of the C -classes semantic mapping $h: \Omega \rightarrow \{0, \dots, C\}$, and $n_l = \sum_{p \in \Omega} y_l(p)$ the number of elements of class l . Let $\varrho_e: \Omega \rightarrow \mathbb{R}^+$ be the bottom hat transform over g using the structuring element e . The bottom hat transform is defined as the difference between the morphological closing of g and the original map g . An hyper-sphere was used as structuring element whose size was empirically determined for every dataset. The expected output for the method is a probability map z for every pixel or voxel such that $z \approx y$. Finally, a post-processing similar to the one proposed in Section 4.2.4 is applied to build a panoptic segmentation \hat{g} from z .

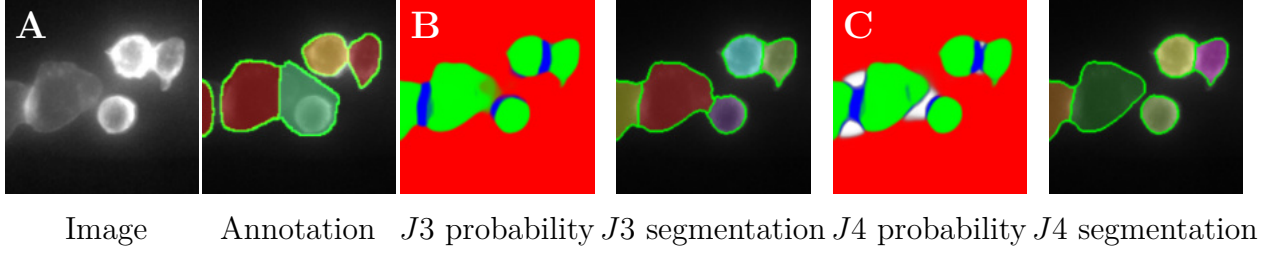
5.2.1 Gap Class

Using three semantic classes for image background, cell interior, and a touching region between cells has shown in the previous chapter that leads to an increase of network discriminative power when segmenting adjoining cells. However, such a definition also leads to the misclassification of background regions of near touching cells (see Figure 32B). By introducing a new training class representing the gap between nearby cells, the network can now classify the regions separating nearby cells as background. The new class is here named gap – white pixels shown in Figure 32C. These regions are obtained using the bottom hat transform. Given an instance annotation g , a semantic ground truth h with four classes is defined here as

$$h(p) = \begin{cases} 0 & \text{if } g(p) = 0 \text{ and } \varrho_e(p) = 0 - \text{background} \\ 3 & \text{if } g(p) = 0 \text{ and } \varrho_e(p) > 0 - \text{gap} \\ 2 & \text{if } g(p') \neq g(p) \text{ and } g(p') \neq 0, \forall p' \in \eta_k(p) - \text{touching} \\ 1 & \text{otherwise} - \text{cell} \end{cases}$$

If p is in the background and lies in the bottom hat transform, then p is a gap pixel/voxel, $h(p) = 3$. Here $k = 2$ is used for all experiments.

Figure 32 – Example of (A) near touching cells image with its respective weak annotation, and the obtained (B) three and (C) four classes output probability map and instance segmentation. The exclusion of dimmed, unseen cells during annotation is not intentional. Image is enhanced only for better visualization.



Source: PEÑA *et al.* (2019)

5.2.2 J Regularization

The J statistic was formulated by statistician William J. Youden to improve the rating performance of diagnostic tests of diseases (YOU DEN, 1950). A high J index implies that the test could predict with high probability if an individual was diseased or not. An ideal test would be able to eliminate false negatives (sick, at risk individuals falsely reported as healthy) and false positives (healthy individuals falsely reported as sick) thus always reporting with certainty diseased (true positive) and healthy (true negative) individuals. The effectiveness of this index in binary classification is due to the equal importance it gives to correctly classifying the subjects belonging and *not* belonging to a class, thus giving equal weight to true positive (sensitivity) and true negative (specificity) rates. The main difference of this statistic with other more prevalent measurements like the F1 score is that it considers both true positives rates (TPR) and true negatives rates (TNR) predictions for measuring the agreement between the expected and obtained classification.

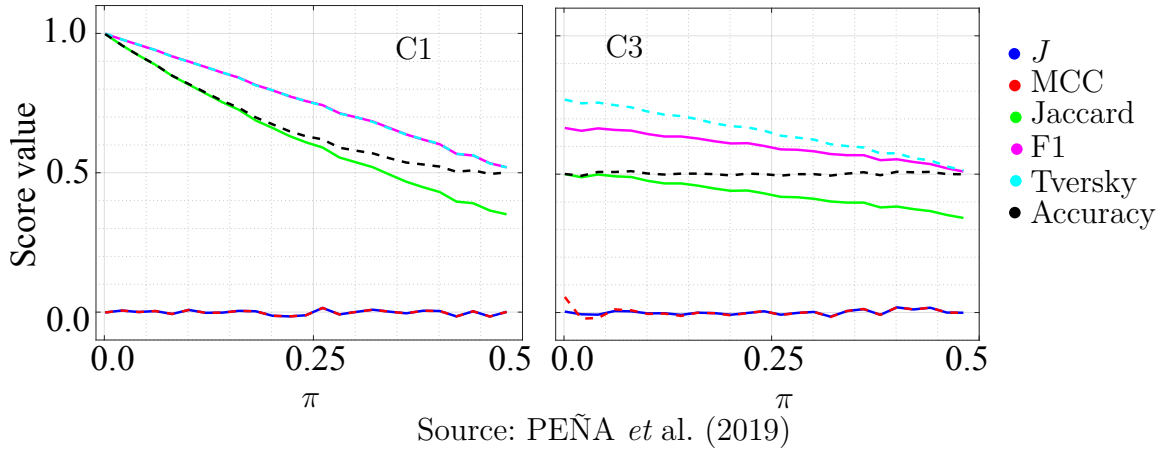
$$J = \text{sensitivity} + \text{specificity} - 1 = TPR \cdot TNR - FPR \cdot FNR$$

where FPR and FNR are false positive rates and false negative rates respectively. J is thus a suitable measure for predicting segmentation with imbalanced classes: biomedical datasets typically holds that $n_0 \gg n_1 \gg n_2 \approx n_3$, *e.g.*, touching and gap classes are comprised of a few pixels/voxels when compared to background and cell classes. Despite the practical range of above expression being $(0, 1)$, here its theoretical interval $(-1, 1)$ (SHAN, 2015) is used to penalize negatives correlations.

Including specificity in the score computation allows to have a more robust measurement in case of highly imbalanced data. For the analysis of these measurements some ideas from (BOUGHORBEL; JARRAY; EL-ANBARI, 2017) were used. As can be seen in Figure 33 the performance of J under different imbalance ratios $\pi = n_1/n_0, \pi \in (0.01, 0.50)$,

is similar to the Matthews Correlation Coefficient (MATTHEWS, 1975) (MCC), which is a well-known performance measurement for highly imbalanced data (BOUGHORBEL; JARRAY; EL-ANBARI, 2017). This is not the case for common metrics as Jaccard index, F1 score, Tversky index, and Accuracy that are usually used for training of neural networks by using a loss surrogate (MILLETARI; NAVAB; AHMADI, 2016; SALEHI; ERDOGMUS; GHOLIPOUR, 2017; BERMAN; TRIKI; BLASCHKO, 2018). The most common surrogate for Accuracy is the Cross Entropy loss (GOODFELLOW; BENGIO; COURVILLE, 2016).

Figure 33 – Performance of classifiers C1 and C3 (BOUGHORBEL; JARRAY; EL-ANBARI, 2017) measured by Youden (J), Matthews Correlation Coefficient (MCC), Jaccard, F1, Tversky, and Accuracy scores for different imbalance ratios π . Youden and MCC are the only ones invariant to all imbalance ratios.



For comparing the correlation between Youden's index J and Matthews Correlation Coefficient, the settings for classifier C3 from (BOUGHORBEL; JARRAY; EL-ANBARI, 2017) were used. C1 is a random prediction where each class has the same imbalance ratio π as in the ground truth. C3 is a random prediction with uniform distribution for all classes, $\pi = 0.5$. Then, the linear correlation between MCC and J values for imbalance ratios π of 0.01, 0.25, and 0.50 was measured by using Pearson's Correlation Coefficient. Figure 34 shows that even for high imbalance between classes ($\pi = 0.01$) a high linear correlation of 0.92 is obtained between both measurements. This supports the idea that Youden's index is robust for binary imbalanced class problems.

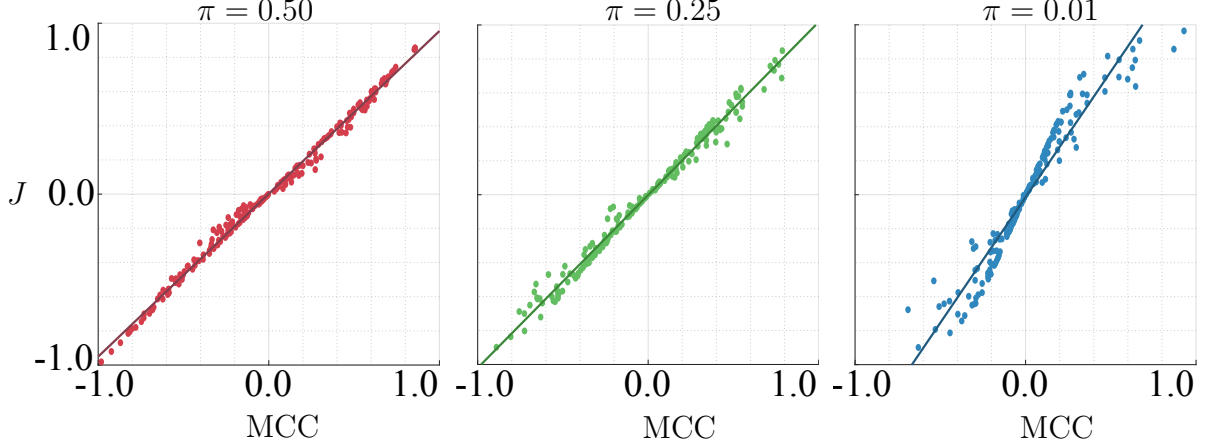
Assuming a binary segmentation problem, a surrogate for J can be defined as

$$\mathcal{L}_J(y, z) = -\lambda \log \left(\frac{1 + J}{2} \right) = -\lambda \log \left(\frac{1 + \alpha \cdot \beta - \gamma \cdot \delta}{2} \right) \quad (5.1)$$

where α , β , γ and δ are soft definitions for TPR, TNR, FPR, and FNR respectively. Adding one and dividing by two has the effect of shifting the original $(-1, 1)$ interval to $(0, 1)$. Based on Equation 5.1, a multiclass surrogate for J can be defined as the pairwise loss expression:

$$\mathcal{L}_J(y, z) = -\sum_{i=0}^C \sum_{k=0}^C \lambda_{i,k} \log \left(\frac{1 + \alpha_i \cdot \beta_{i,k} - \gamma_{i,k} \cdot \delta_i}{2} \right) \quad (5.2)$$

Figure 34 – Correlation between values of MCC and J for different imbalance ratios π . The linear correlation was measured using Pearson Correlation Coefficient with values of 0.92 for $\pi = 0.01$, 0.99 for $\pi = 0.25$, and approximately 1.0 for $\pi = 0.5$.



Source: PEÑA *et al.* (2019)

where λ is a user-defined weighting matrix and $\lambda_{i,k}$ is an element of the matrix. α_i , $\beta_{i,k}$, $\gamma_{i,k}$ and δ_i represent the soft definitions for TPR, TNR, FPR, and FNR respectively where i is considered to be the positive class and k the negative one. Here definitions close to the one used for Soft Dice (MILLETARI; NAVAB; AHMADI, 2016) and Tversky (SALEHI; ERDOGMUS; GHOLIPOUR, 2017) loss functions are used,

$$\alpha_i = \frac{\sum_{p \in \Omega} z_i(p) \cdot y_i(p)}{\sum_{p \in \Omega} y_i(p)}, \beta_{i,k} = \frac{\sum_{p \in \Omega} (1 - z_i(p)) \cdot y_k(p)}{\sum_{p \in \Omega} y_k(p)},$$

$$\gamma_{i,k} = \frac{\sum_{p \in \Omega} z_i(p) \cdot y_k(p)}{\sum_{p \in \Omega} y_k(p)}, \delta_i = \frac{\sum_{p \in \Omega} (1 - z_i(p)) \cdot y_i(p)}{\sum_{p \in \Omega} y_i(p)}$$

In practice, because $\gamma_{i,k} = 1 - \alpha_i$ and $\delta_i = 1 - \beta_{i,k}$, the term $\gamma_{i,k} \cdot \delta_i = (1 - \alpha_i) \cdot (1 - \beta_{i,k}) = 1 + \alpha_i \cdot \beta_{i,k} - \alpha_i - \beta_{i,k}$. Then, Equation 5.2 can be rewritten as:

$$\begin{aligned} \mathcal{L}_J(y, z) &= - \sum_{i=0}^C \sum_{k=0}^C \lambda_{i,k} \log \left(\frac{1 + \alpha_i \cdot \beta_{i,k} - 1 - \alpha_i \cdot \beta_{i,k} + \alpha_i + \beta_{i,k}}{2} \right) \\ &= - \sum_{i=0}^C \sum_{k=0}^C \lambda_{i,k} \log \left(\frac{\alpha_i + \beta_{i,k}}{2} \right) \end{aligned}$$

By evaluating α_i and $\beta_{i,k}$ expressions into above equation and using the definition $n_l = \sum_{p \in \Omega} y_l(p)$, the loss can be expressed as:

$$\begin{aligned}
\mathcal{L}_J(y, z) &= - \sum_{i=0}^C \sum_{k=0}^C \lambda_{i,k} \log \left(\frac{\sum_{p \in \Omega} z_i(p) \cdot y_i(p)}{2n_i} + \frac{\sum_{p \in \Omega} (1 - z_i(p)) \cdot y_k(p)}{2n_k} \right) \\
&= - \sum_{i=0}^C \sum_{k=0}^C \lambda_{i,k} \log \left(\frac{\sum_{p \in \Omega} z_i(p) \cdot y_i(p)}{2n_i} - \frac{\sum_{p \in \Omega} z_i(p) \cdot y_k(p)}{2n_k} + \frac{\sum_{p \in \Omega} y_k(p)}{2n_k} \right)
\end{aligned}$$

Taking z_i as a common factor and using the fact that $\sum_{p \in \Omega} \frac{y_k(p)}{n_k} = 1$, an expression for Equation 5.2 can be obtained and rewritten as:

$$\mathcal{L}_J(y, z) = - \sum_{i=0}^C \sum_{k=0}^C \lambda_{i,k} \log \left[\frac{1}{2} + \sum_{p \in \Omega} z_i(p) \cdot \left(\frac{y_i(p)}{2n_i} - \frac{y_k(p)}{2n_k} \right) \right] \quad (5.3)$$

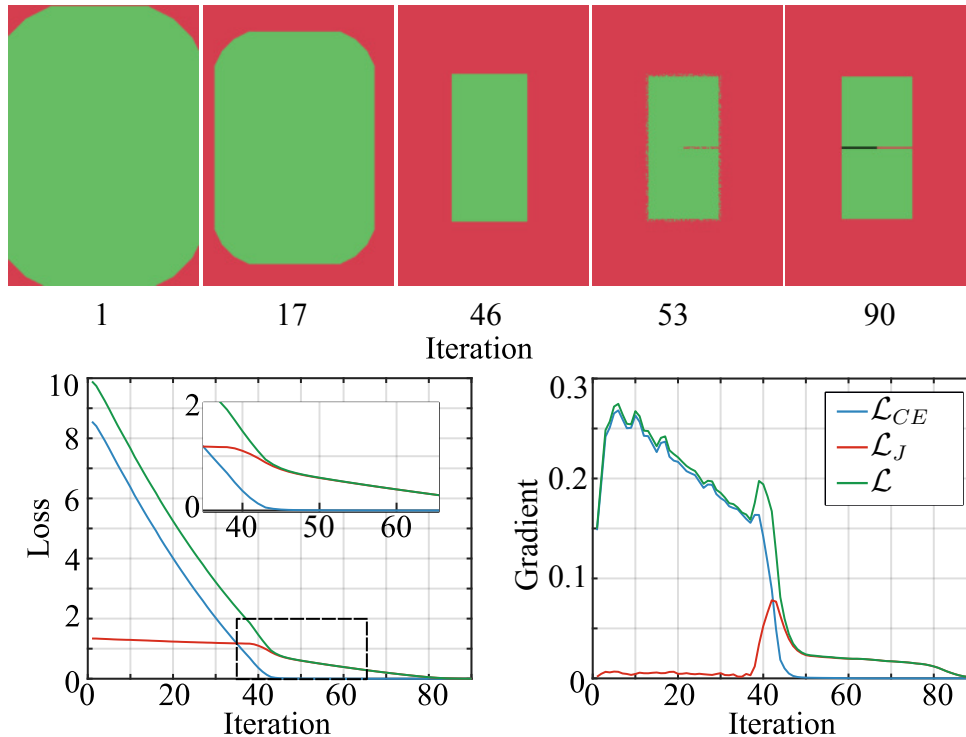
This formulation favors obtaining high difference between probabilities of elements in different classes. Here, Equation 5.3 is used as a regularization term for assisting the Cross Entropy (CE) loss, $\mathcal{L}_{CE}(y, z) = -\frac{1}{|\Omega|} \sum_{l=0}^C \sum_{p \in \Omega} y_l(p) \cdot \log z_l(p)$. This is, from all solutions with equal values of CE, the one with higher difference between class is favored. Then, the loss function used for training in this work is $\mathcal{L}(y, z) = \mathcal{L}_{CE}(y, z) + \mathcal{L}_J(y, z)$.

At the beginning of training, when cell class begins to be learned, it appears as a large region that progressively shrinks until it fits the cell. At the same time, the probabilities for every class slowly increase, reaching near one values first for majority classes. This same behavior was simulated here, and the value of the loss for each term in \mathcal{L} was measured independently. Figure 35 shows some examples of the segmentation obtained at different iterations as well as the values of the loss and gradient. As can be seen in the first 40 iterations, Cross Entropy leads the optimization while Youden term has values of gradient near to zero. At iteration 46 can be observed that a small concavity belonging to background class and the touching class is missing, but Cross Entropy has gradient values near to zero. On the other hand, Youden gradient increases as the value for the loss begins to decrease very fast, enforcing the appearance of the touching class and the right classification of pixels in the small gap region.

5.2.3 Gap Output Assignment

A semantic segmentation can be obtained from an output probability map by using the Maximum A Posteriori (MAP) decision rule, $\hat{h}(p) = \arg \max_l z_l(p)$. However, since the fourth class represents uncertainty in the classification, the elements in this category are assigned to the second most likely class. In the end, the post-processing is the same of applying MAP over the first three classes of the probability map, $\hat{h}(p) = \arg \max_{l \in \{0,1,2\}} z_l(p)$. From this point an instance segmentation is achieved by a sequence of labeling operations of each region in the semantic segmentation map following Section 4.2.4.

Figure 35 – Simulations performed for analyzing the behavior of Cross Entropy and Youden index-based loss functions. During the first iterations, the segmentation is shrunk until it fits the ground truth, with a slow increase of the probabilities for each class. After this point, the probabilities are gradually increased for all classes at each time step resulting in CE values near to zero but higher values of J .



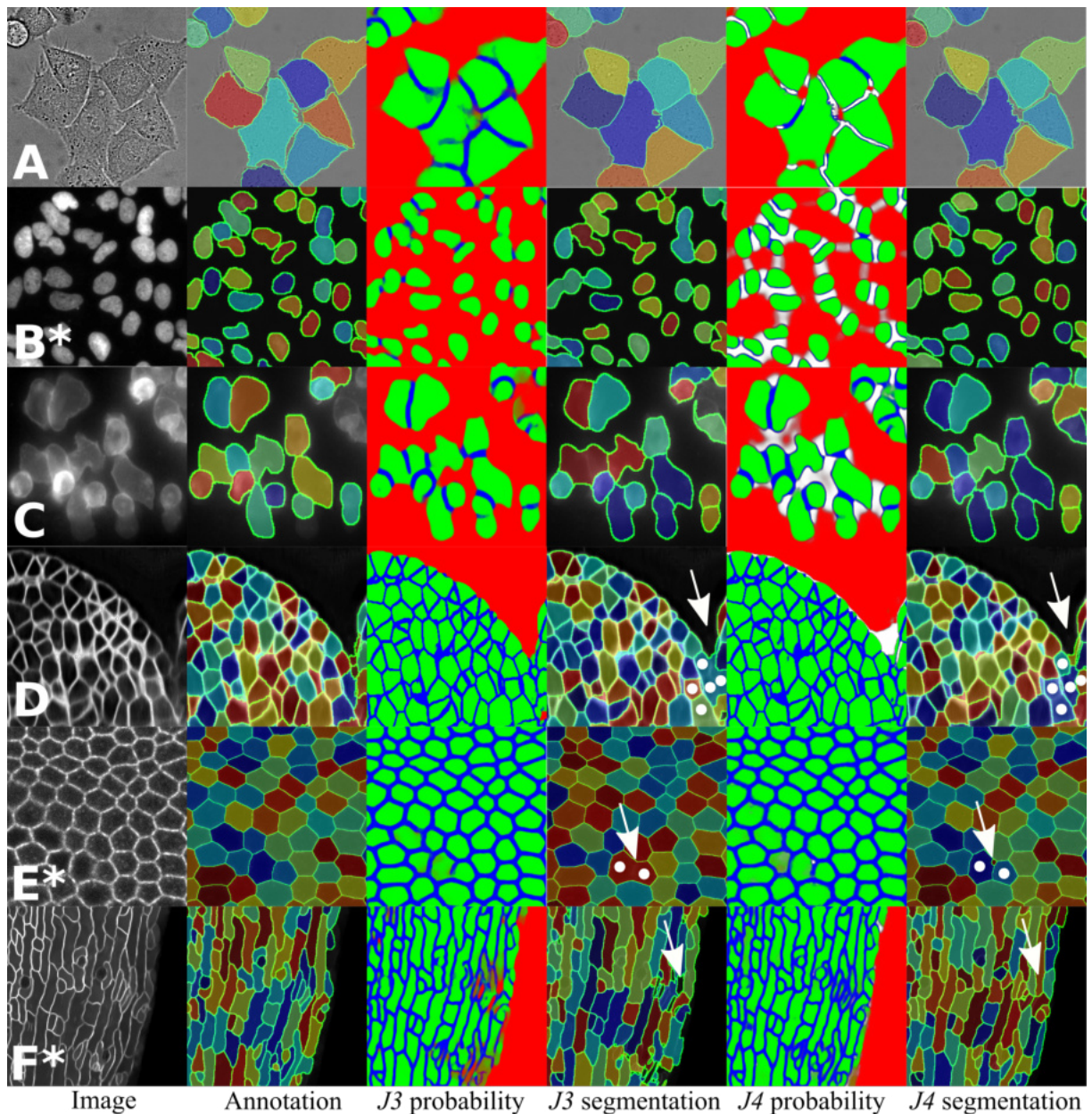
Source: PEÑA *et al.* (2019)

5.3 EXPERIMENTS AND RESULTS

For comparing different loss functions the U-Net architecture (RONNEBERGER; FISCHER; BROX, 2015) was used. For initialization of the weights the Xavier method with normal distribution was used (GLOROT; BENGIO, 2010). For 3D volumes, the same architecture was adopted but having 3D convolutions instead of 2D (MILLETARI; NAVAB; AHMADI, 2016). All networks are guaranteed to begin with the same random set of weights. Mini-batches and random data augmentation are also guaranteed to be the same during training by fixing all random seeds. The loss function Weighted Cross Entropy with class Balance (BWM), Weighted Cross Entropy with Triplex weight map (W^3) from Section 4.2.3, and Cross Entropy with Dice score regularization (DSC) (ISENSEE *et al.*, 2019) over three classes were used for comparison with the proposed Cross Entropy with Youden-based regularization on three ($J3$) and four ($J4$) classes. A Watershed post-processing (WT) is also used in the comparison for assisting networks with uncertainty in touching separation.

The influence of the proposed gap class during training was also analyzed by comparing $J3$ and $J4$ over DIC Hela dataset (ISBI, 2019), a 3D confocal image stacks of Meristem

Figure 36 – Segmentation results for Hela cells (A), Hela *nuclei* (B*), T-cells (C), meristem cells (a YZ-slice of the 3D segmented stack is shown) (D), Drosophila cells (E*), and sepal cells (z projection) (F*) images using networks trained with $J3$ and $J4$ loss functions. Probability maps are shown as RGB images with background (red), cell (green), and touching (blue) classes. For $J4$, the proximity prediction is shown in white. Asterisks (*) indicate zero-shot instance segmentations with networks trained exclusively over T-cells (C). Colors are to show cell separation. Original images were enhanced to help visualization. Whites arrows and circles are used to indicate some differences between $J3$ and $J4$.



Source: PEÑA *et al.* (2019)

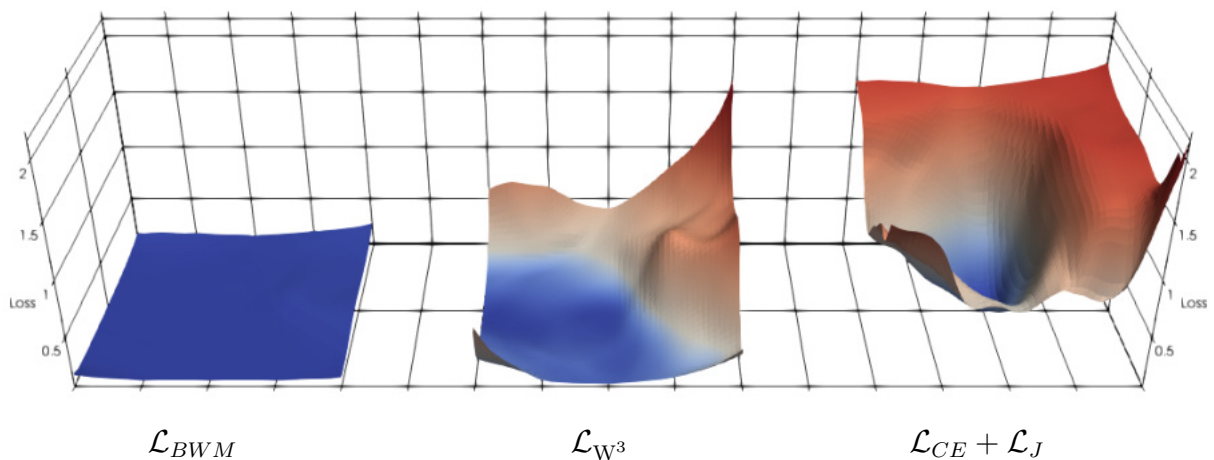
cells (WILLIS *et al.*, 2016), and the T-cells cells dataset. Zero-shot segmentation of Hela cells (LJOSA; SOKOLNICKI; CARPENTER, 2012), Drosophila and Sepal dataset was obtained by

using a model trained over the T-cells data. For training purposes, the optimizer Adam (KINGMA; BA, 2015) with an initial learning rate of 10^{-4} was used. Data augmentation included random rotation, mirroring, gamma correction, touching contrast modulation, and warping. For performance evaluation Precision (P05) and F1 score (RQ) were used for cell detection rates. Segmentation Quality (SQ) and Panoptic Quality were used for measuring contour adequacy and instance segmentation qualities, respectively (KIRILLOV et al., 2019).

5.3.1 Loss Landscape Visualization

The approach proposed by Li et al. (LI et al., 2018) for loss landscape visualization was used to have an insight into the kind of modifications of the error surface obtained with different loss functions. Of course, such type of reduction of the dimensionality of the actual error surface is not representative of the entire landscape morphology. However, the surfaces can be used for comparing the behavior of loss functions around a fixed optimizer. For this experiment, the same plane was selected, and the center of each surface corresponds with a fixed optimizer θ^* . As can be seen in Figure 37 the landscape for the Weighted Cross Entropy with class balance \mathcal{L}_{BWM} is almost flat for the entire plane slice. On the other hand, the Triplex weight map loss has a better penalization than BWM for networks that makes mistakes in higher weighted regions, but near-constant loss for areas around the optimizer. Presumably, these flat regions are composed of networks that produced small probabilities values at the end of the touching area. However, the proposed Cross Entropy with Youden regularization $\mathcal{L}_{CE} + \mathcal{L}_J$ obtain a better discrimination for solutions near the optimizer.

Figure 37 – Loss landscape visualization around a fixed optimizer of Weighted Cross Entropy with class balance \mathcal{L}_{WCE} , proposed Triplex weight map \mathcal{L}_{W^3} , and introduced Cross Entropy with Youden-based regularization $\mathcal{L}_{CE} + \mathcal{L}_J$.



Source: PEÑA et al. (2019)

5.3.2 Instances Segmentation Performance

Table 6 shows a performance comparison of U-Net trained over T-cell dataset with different loss functions. As can be seen, Watershed (WT) post-processing effectively increased the performance for BWM, DSC and W^3 when compared with Maximum a Posteriori (MAP) approach. However, WT method depends on two parameters that need to be found. On the other hand, networks trained with the proposed Youden-based loss function are able to improve the instances detection rates by only using the parameter-free MAP post-processing. This occurs because touching regions' probabilities are very high, obtaining a better separation between adjacent cells. Because this is a weakly annotated dataset (see annotation in Figure 32), a bias in the value of the parameter SQ toward loose segmentations can be observed.

Table 6 – Performance comparison of U-Net trained over T-cell dataset using Weighted Cross Entropy with class Balance (BWM), Cross Entropy with Dice regularization (DSC), Weighted Cross Entropy with Triplex weight map (W^3), and Youden based regularization over three ($J3$) and four ($J4$) classes.

Loss function	Post	P05	RQ	SQ	PQ
BWM	MAP	0.6756	0.5580	0.8674	0.4858
DSC	MAP	0.9028	0.7674	0.9011	0.6923
W^3	MAP	0.7384	0.6305	0.8721	0.5513
BWM	WT	0.8193	0.8405	0.8831	0.7437
DSC	WT	0.8726	0.8269	0.8925	0.7390
W^3	WT	0.9028	0.8775	0.8995	0.7896
$J3$ (Proposed)	MAP	0.9127	0.9069	0.8733	0.7921
$J4$ (Proposed)	MAP	0.9334	0.9353	0.8689	0.8132

Source: PEÑA *et al.* (2019)

The proposed loss function was used for measuring the gap class influence. Table 7 shows obtained results over each dataset. The best Panoptic Quality for all cases was obtained with four classes. An improvement on the Segmentation Quality is observed for the first two datasets as a direct consequence of the fourth class (see first three rows in Figure 36). However, as stated before, weak annotations, as in the case of T-cells and Meristem datasets, bias the SQ value, decreasing the performance even when the obtained segmentation has better contour adequacy. Figure 36C shows an example of the segmentation and probability map using $J3$ and $J4$ for T-cell dataset. In Figure 36D can be seen an XY-plane segmentation of the Meristem volume.

An example of 3D segmentation of the Meristem training with the proposed $J4$ loss function is shown in Figure 38. The segmentation for a portion of the volume are only shown due to the difficulties in visualizing the difference between obtained volumes. This crop has been previously segmented using the watershed with markers technique, which

Table 7 – Results obtained over different datasets show the benefits of using the additional gap class. In all cases a higher PQ value is obtained for $J4$. A (*) indicates zero-shot segmentation.

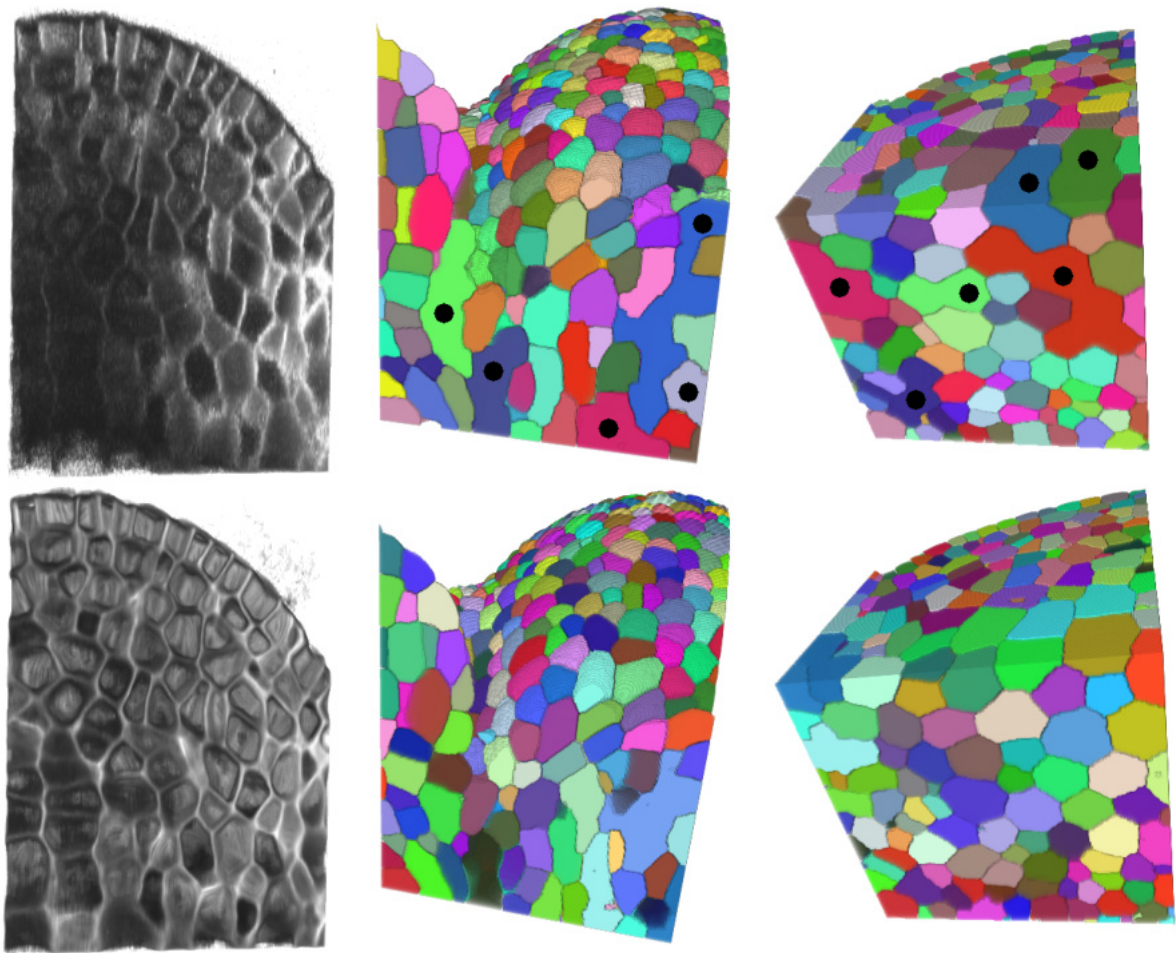
Loss function	Dataset	RQ	SQ	PQ
$J3$	DIC	0.8950	0.8547	0.7633
$J4$	DIC	0.8884	0.8833	0.7841
$J3$	HELA*	0.8527	0.8475	0.7237
$J4$	HELA*	0.9046	0.8574	0.7764
$J3$	TCELLS	0.9069	0.8733	0.7921
$J4$	TCELLS	0.9353	0.8689	0.8132
$J3$	MERISTEM 3D	0.8829	0.8820	0.7787
$J4$	MERISTEM 3D	0.8947	0.8804	0.7878

Source: PEÑA *et al.* (2019)

is considered as an approximate ground truth. Enhancing the signal quality improves segmentation, as shown for those undersegmented regions of the noisy stack manually marked with black circles. The trained network can process large, $1024 \times 1024 \times 508$, meristem stacks in under 9 minutes using 2 Nvidia K80 GPU cards (31 minutes in a single card).

Examples of the segmentation obtained with $J3$, $J4$, and Dice regularizations for 2D images are shown in Figure 39. T-cells images were acquired by specialist from Rothenberg Lab at the California Institute of Technology. 3D segmentations obtained with $J4$ can be seen in Figure 40. Meristem volumes showed in the figure were acquired by specialist from Meyerowitz Lab at the California Institute of Technology.

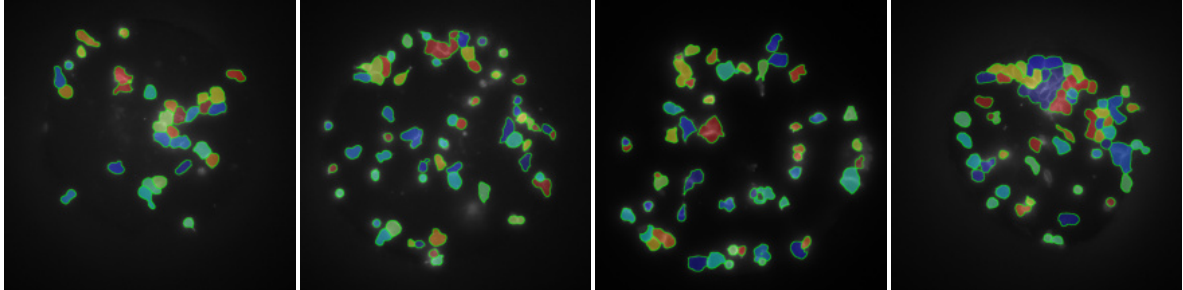
Figure 38 – Example of 3D segmentation using the proposed $J4$ loss function. Original and enhanced versions (left column) of a meristem portion image stack and their respective segmentations (two views on the middle and right columns). Instances colors and black contour are merely used to illustrate the separation of individually segmented regions. Colors are randomly assigned for every cell.



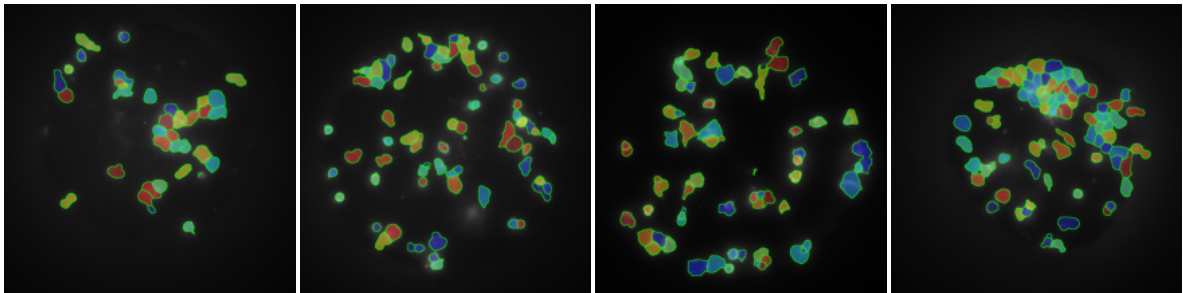
Source: PEÑA *et al.* (2019)

Figure 39 – Examples of 2D weakly supervised biomedical image instance segmentation with $J3$, $J4$, and Dice regularizations. Instances colors and green contour are merely used to illustrate the separation of individually segmented regions.

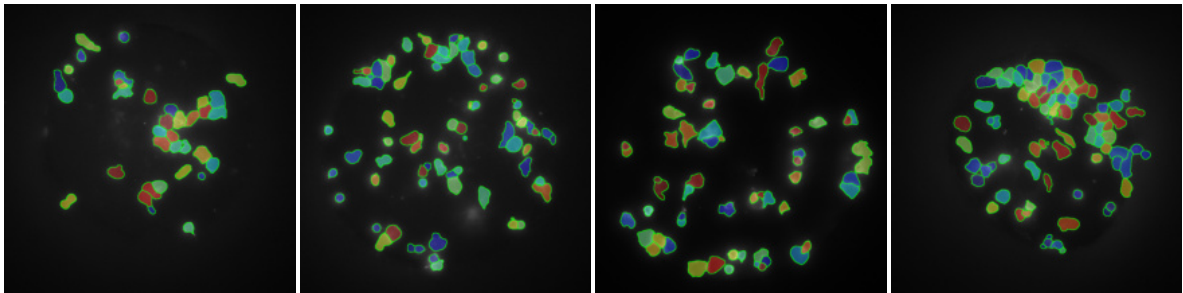
Cross Entropy with Dice regularization



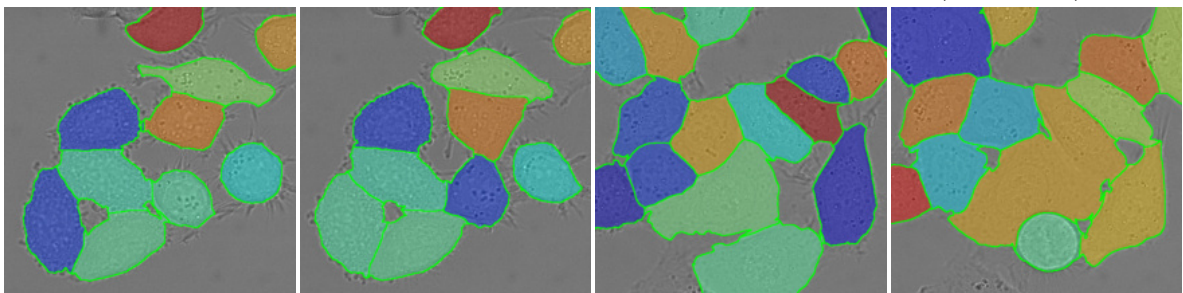
Cross Entropy with Youden regularization over three classes (Proposed)



Cross Entropy with Youden regularization over four classes (Proposed)



Cross Entropy with Youden regularization over three classes (Proposed)



Cross Entropy with Youden regularization over four classes (Proposed)

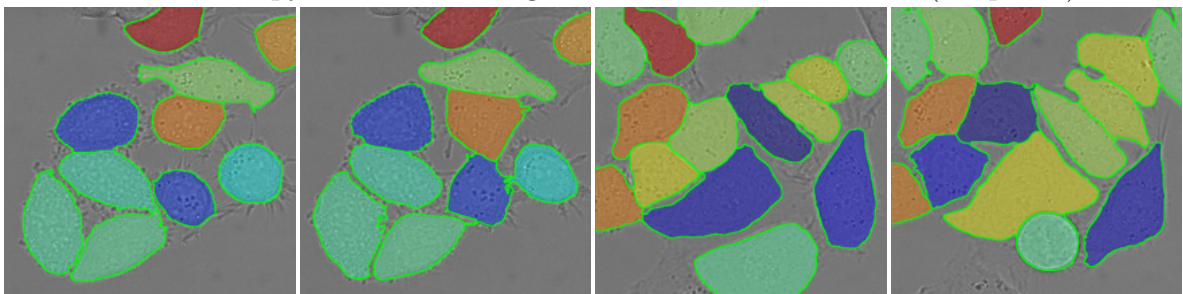
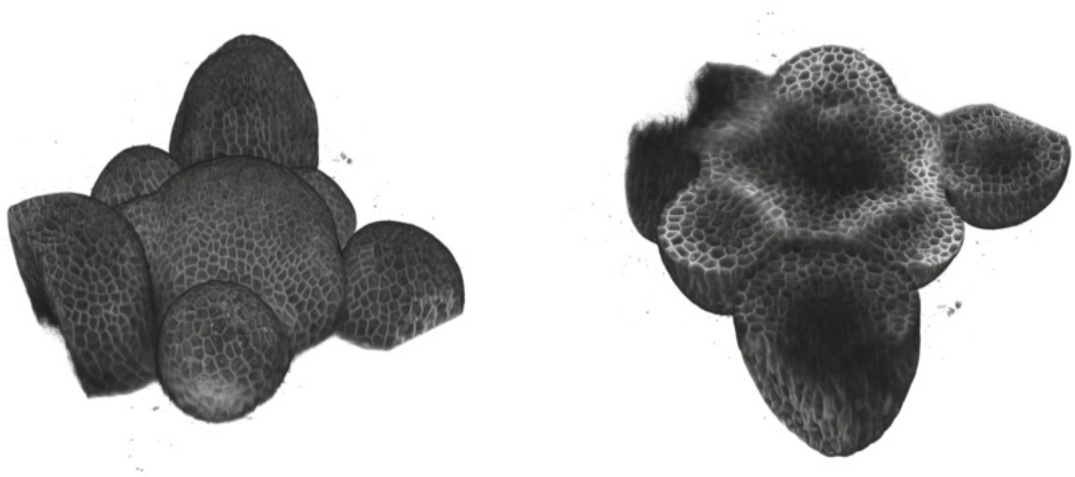
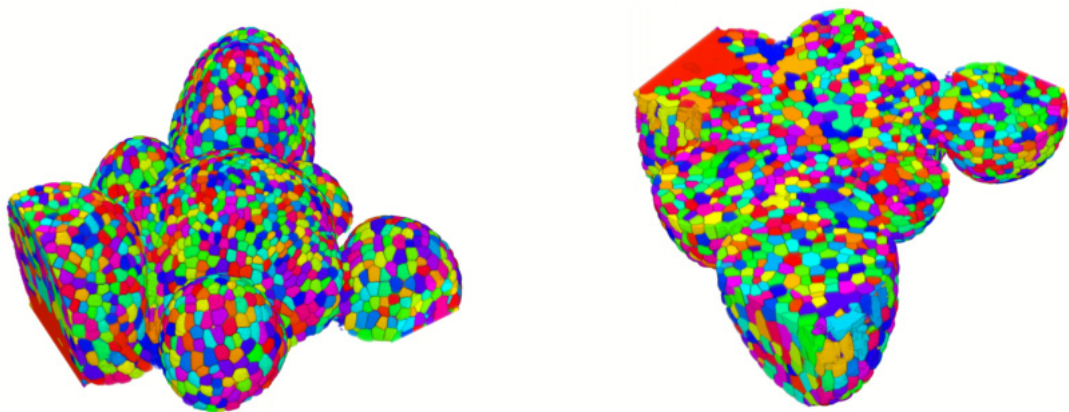


Figure 40 – 3D weakly supervised instance segmentation with $J4$ over two views of the meristem volumen. Instances colors and black contour are merely used to illustrate the separation of individually segmented regions. Colors are randomly assigned for every cell.

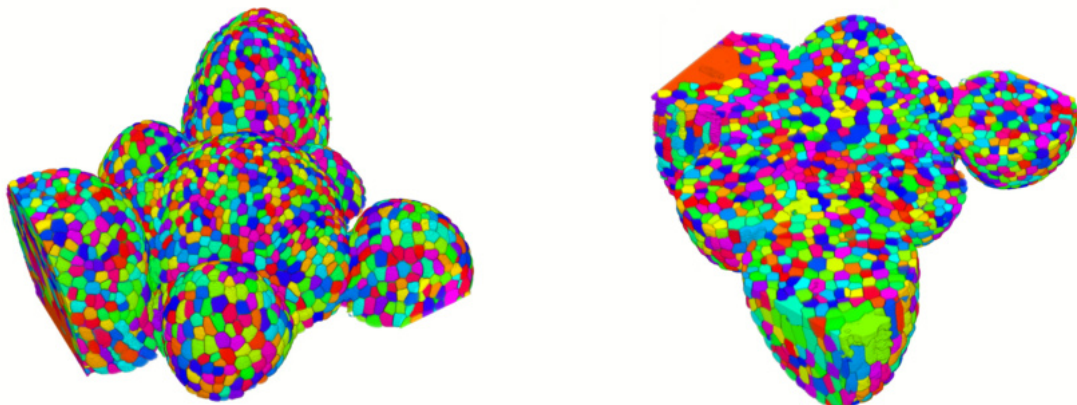
pWUS volumen



Cross Entropy with Youden regularization over three classes (Proposed)



Cross Entropy with Youden regularization over four classes (Proposed)



5.4 CONCLUSIONS

In this chapter a new Youden regularization term for accounting high class imbalance was introduced. Youden imbalance robustness was shown through several simulations. Better contour adequacy was enforced by adding a new class for background regions between near cells. The approach proved to be feasible for 2D and 3D instance segmentation of highly clustered cells even in the presence of weak annotations. The results showed an improvement in the performance by using the parameter-free Maximum A Posteriori post-processing. The proposed approach also revealed to be feasible for segmenting images of near domains never seen before during training. An increment of the detection rate in almost all datasets was observed when the proposed fourth class was used. Contour adequacy in weakly supervised cases was also improved, as seen from visual inspection. Landscape analysis and performance evaluation with different loss functions suggest obtaining better-performed models when the proposed loss function is used.

6 A MULTIPLE SOURCE HOURGLASS DEEP NETWORK FOR MULTI-FOCUS IMAGE FUSION^[4]

Multi-focus Image Fusion seeks to improve the quality of an acquired burst of images with different focus planes. In this chapter two fast and straightforward approaches are proposed for image fusion based on deep neural networks. The solution uses a U-Net architecture trained in an end-to-end fashion. The designed training loss function for the regression-based fusion includes learning of both the activity level measurement and the fusion rule. Despite there is a vast amount of data available for this task, the optimization challenge remains because the typical loss functions for regression are usually insufficient for solving the problem.

6.1 INTRODUCTION

Usually, the limited depth-of-field of digital cameras causes only one image of the plane to stay in focus while the others appear blurred. This focus plane is composed of all objects near to a fixed focus point. Taking several shots with different focus points allows the capture of a burst of images where all focus planes become available. The process of reconstructing the entirely focused image by estimating the sharpest pixel values using frame information is named Multi-focus Image Fusion (MFIF), see Figure 41. The resulting focused image is known in the literature as the all-in-focus image and is typically used for further computer processing. Thus, MFIF can be described as a pre-processing step that improves the quality of the acquired burst of images (TAN et al., 2017; TANG et al., 2018). Applications of MFIF include, but are not limited to, medical and biological imaging, video surveillance and digital photography (GANGAPURE; BANERJEE; CHOWDHURY, 2015; KONG; LEI; ZHAO, 2014). Many challenges, such as identifying the focus map in each frame, selecting the fusion function to combine the focus planes and performing a quick and reliable combination of images remain as open issues, making the multi-focus image fusion an interesting problem to investigate.

Most of the existing MFIF method contributions rely on proposals for new activity level measurements and/or fusion rules to solve the task. However, in recent years, this practice has been simplified through the employment of deep convolutional neural networks (CNN), and several deep learning based methods have been introduced to create faster and simpler MFIF approaches.

^[4] Fidel A. Guerrero Peña; Pedro Marrero Fernandez; Tsang Ing Ren; Germano Crispim Vasconcelos; Alexandre Cunha. Centro de Informática, Universidade Federal de Pernambuco, Brazil; Center for Advanced Methods in Biological Image Analysis, California Institute of Technology, USA. Available at <<https://arxiv.org/abs/1908.10945>>, 2019.

Figure 41 – Example of different focus source images and the all-in-focus resulting image. The sources A and B represent the same image in different focal planes.



Source: PEÑA *et al.* (2019a)

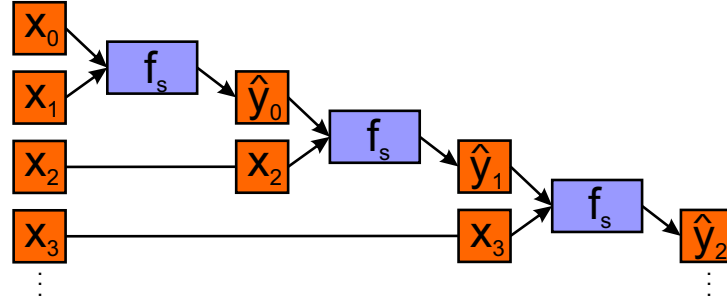
6.2 MULTI-FOCUS IMAGE FUSION LEARNING

Despite the increasing interest of deep learning community in MFIF problems, a direct regression technique has not been proposed because the associated complexity of the regression is not well addressed with current regression loss functions. Here, the multi-focus image fusion problem is formulated as a multiple source segmentation/regression process where two frames are given to a Convolutional Heteroencoder (U-Net), and an RGB all-in-focus image is obtained as a result.

The set of all multi-focus image pairs is defined as $\mathbb{X} = \{\mathbf{x}_k \mid \mathbf{x}_k = (x_{kA}, x_{kB})\}$, where $x_k: \Omega \rightarrow \mathbb{R}^3, \Omega \subset \mathbb{R}^2$, is an RGB source image. For MFIF task the training set is defined as $S = \{(\mathbf{x}_i, y_i)\}_{i=0}^l$, with cardinality $|S| = l$, where $\mathbf{x}_k \in \mathbb{X}$ is a source image pair and $y_k: \Omega \rightarrow \mathbb{R}^3$ is an all-in-focus ground truth image. Here, all images are assumed to be in the normalized range $[0, 1]$. Let $\mathbf{x} = (x_A, x_B)$ be a generic source tuple of \mathbb{X} and y its focused ground truth. The goal is to find a fusion function $f(\mathbf{x})$ which takes two sources frames with different focus as input and obtain a fused image \hat{y} as close as possible to the latent image y , $\hat{y} \approx y$. Note that a fusion function f must be indepen-

dent to pair order and therefore must meet the commutative law. This is regarded as $f(\mathbf{x}) = f(\bar{\mathbf{x}})$ where $\bar{\mathbf{x}}$ is the reverse order of the tuple \mathbf{x} , $\bar{\mathbf{x}} = (x_B, x_A)$. Here, the function f is approximated using U-Net (RONNEBERGER; FISCHER; BROX, 2015), which is a well-known hourglass architecture. The commutative property is ensured through an appropriate training protocol as described later. Although f is bi-variable, a generalization for bursts $\mathbf{x}^n = (x_0, \dots, x_n)$ with $n + 1$ frames can be defined as the n -th functional power f^n , $f^n(\mathbf{x}^n) = (f \circ f^{n-1})(\mathbf{x}^n)$, where \circ represent the partial composition operation, *e.g.*, $f^2(\mathbf{x}^2) = (f \circ f)(\mathbf{x}^2) = f(f(x_0, x_1), x_2)$. Figure 42 shows the overall process for multi-focus fusion of n frames.

Figure 42 – Overall method scheme for a multi-focus fusion of an input burst. The images within the burst are incrementally fused through the n -th functional power f^n .



Source: PEÑA *et al.* (2019a)

6.2.1 Multi-focus Image Fusion Dataset

Training the neural network to predict the latent focused image given two blurry inputs requires a vast amount of training data. However, there is not a public multi-focus image fusion dataset with the all-in-focus ground truth available. Therefore, a dataset is synthetically generated in this work to allow the training of such CNN. A potential idea would be to apply blur in some randomly selected patches of a sharp image y , and create the pair \mathbf{x} with the blurred and sharp patches, *e.g.*, if x_A is blurred then x_B is its corresponding sharp patch from y . This approach was used recently by Liu *et al.* (LIU *et al.*, 2017), where the ImageNet classification dataset was used to generate the training data. However, because the network here used is not a patch classification approach, the final input sources are required to contain a focus map where focused and blurred regions appear in the same frame. Following this idea, the data generation method proposed in (TANG *et al.*, 2018) simulate situations where an image patch include both focused and de-focused regions. This is done by defining 12 masks of blurred and unchanged areas used as a focus map. Nevertheless, this small size set of masks might be insufficient to model the latent focus maps space significantly. Also, creating an MFIF dataset by hand

is very expensive, given the enormous amount of ground truth data required to train the network.

Here, a new dataset generation is proposed by applying synthetic blur to randomly selected objects instances extracted from the MS COCO segmentation dataset (LIN et al., 2014). This dataset contains highly varied real-world images collected from the internet and its segmentation ground truth. Let $E = \{(y_0, g_0), \dots, (y_m, g_m)\}$ be a panoptic segmentation set where y_k is an image and g_k is its segmentation mask, $g_k: \Omega \rightarrow \{0, \dots, \gamma_k\}$ being γ_k the number of segmented objects. Let (y, g) be a generic tuple from E where there are γ segmented objects. Let $\Gamma \subset \{0, \dots, \gamma\}$ be a randomly selected subset of objects of g . Then, a focus map set $G = \{p \mid c(p) \in \Gamma\}$ can be defined where $c(p)$ returns the object number assigned to pixel p , $c: \Omega \rightarrow \{0, \dots, \gamma\}$. A binary focus map $g^b: \Omega \rightarrow \{0, 1\}$, is then defined as $g^b(p) = \mathbb{1}_G(p)$ where $\mathbb{1}_G$ is the indicator function over G , *e.g.*, $g^b(p) = 1$ if $c(p) \in \Gamma$, otherwise $g^b(p) = 0$.

A Gaussian blur kernel h_σ is created using a uniformly generated standard deviation $\sigma \sim U(1, 5)$. Then, a blurred image $\bar{y} = y * h_\sigma$ is obtained by convolving the focused image with the blur kernel. Finally, a multi-focus input tuple $\mathbb{x} = (x_A, x_B)$ is generated on-the-fly using the focus map g^b and the blurred and sharp versions of the frame y (Equation 6.1).

$$\mathbb{x} = (\bar{y} \cdot g^b + y \cdot (1 - g^b), \bar{y} \cdot (1 - g^b) + y \cdot g^b) \quad (6.1)$$

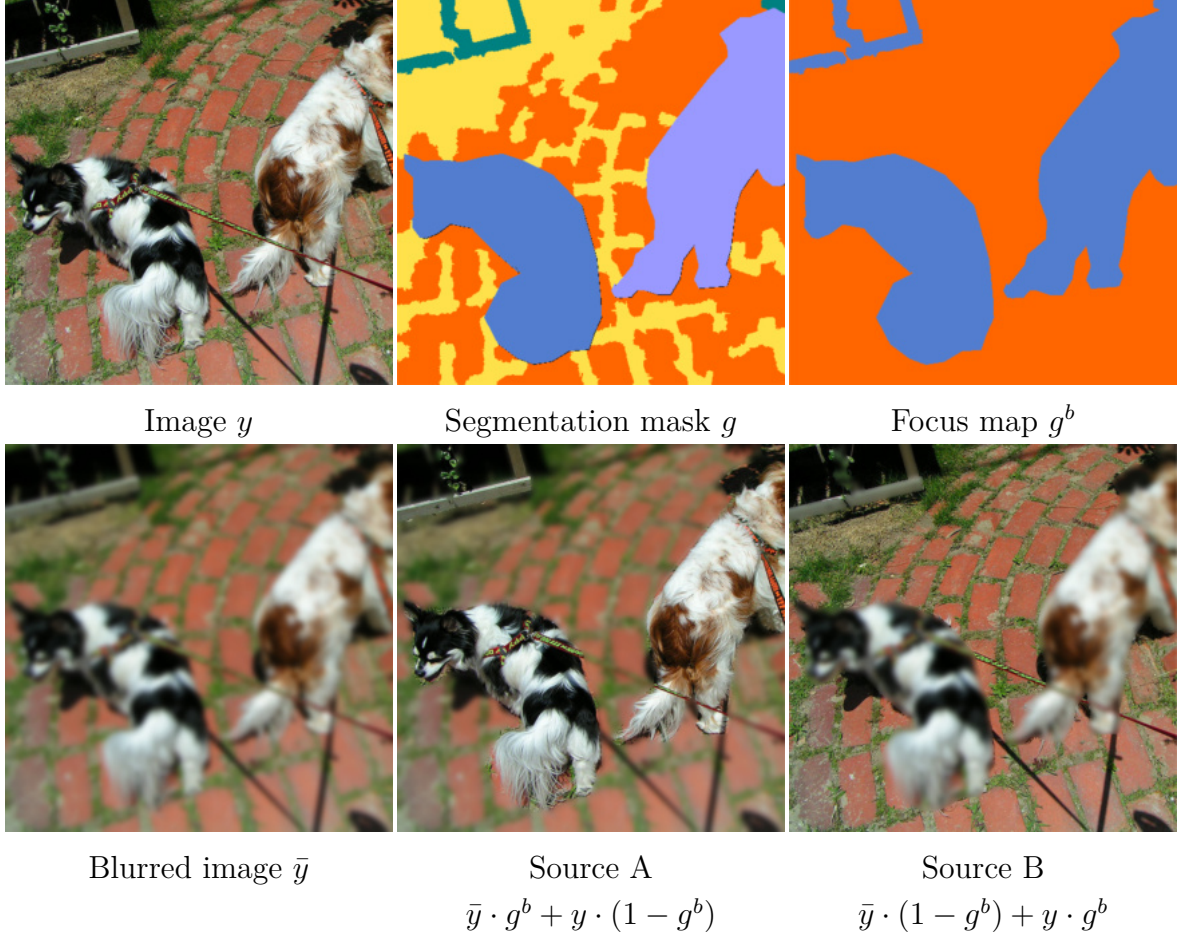
A generated sample of the proposed realistic synthetic dataset is shown in Figure 43, with the corresponding sharp image y and its segmentation mask g . Some objects randomly selected were taken as background, leaving the rest in the foreground, resulting in the focus map g^b . Finally, the generated source frames computed according to Equation 6.1, are shown in the last row. Hence, this approach can provide an endless amount of training data.

6.2.2 Multiple Sources Hourglass Network

In this research two methodologies are proposed for training the U-Net architecture to approximate the fusion function f . An extension for multiple input sources is proposed here based on the results of (ZAGORUYKO; KOMODAKIS, 2015) to learn a similarity function. In their work the superiority of multiple source approaches was validated when compared with Siamese methods, which takes a single image as input in the feature extraction path. Nevertheless, the scheme proposed by Zagoruyko et al. for single value regression is generalized here to full RGB images regression/segmentation tasks.

The two variants of the hourglass architectures presented here that seeks to solve the multi-focus image fusion problem are called Hourglass Fusion Segmentation network (HF-Seg) and Hourglass Fusion Regression network (HF-Reg). Figure 44 shows the overall multi-source U-Net architecture.

Figure 43 – Example of synthetic tuple \mathbf{x} created by applying the MFIF dataset using MS COCO image y and its segmentation mask g . The focus map g^b was created using two classes as background and the other three objects as foreground. The blurred image \bar{y} and resulting sources (x_A, x_B) are shown in the second row.

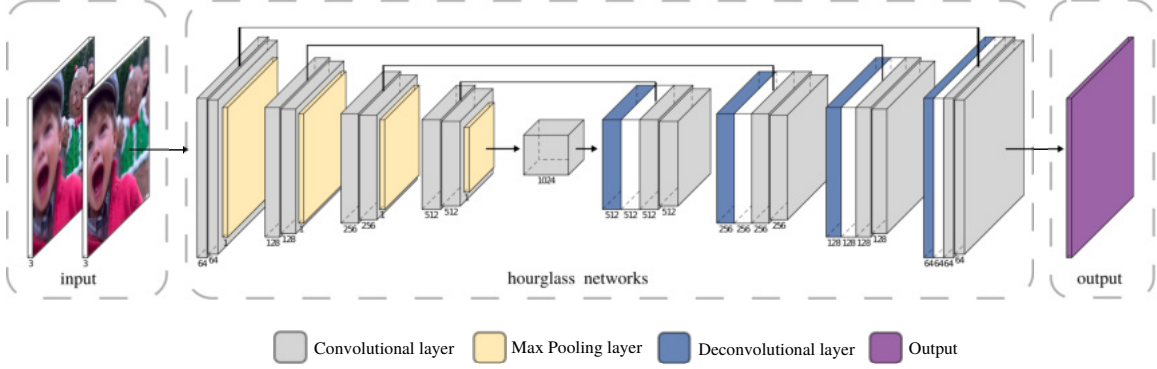


Source: PEÑA *et al.* (2019a)

Fusion map prediction (HF-Seg network). The first proposal uses the hourglass network for fusion map estimation. This is based on the ideas of (TANG *et al.*, 2018; LIU *et al.*, 2017) for obtaining a focus map. Differently to theirs, here, the problem is defined as a segmentation process where the HF-Seg architecture receive two RGB sources as a 6-channels map $\mathbf{x} = (x_A, x_B)$, and the output is fed into a Softmax layer, used to obtaining a two-channel segmentation map $\mathbf{z} = (z_0, z_1)$. In practice, this segmentation map represents the predicted fusion map and its complement, $z_0 = 1 - z_1$. After obtaining the focus map, the resulting fused image can be inferred by applying a fusion rule. The fusion function f_S is expressed as the pixel-wise weighted-average rule of the network output map (LIU *et al.*, 2017; TANG *et al.*, 2018):

$$f_S(\mathbf{x}) = z_0 \cdot x_A + z_1 \cdot x_B \quad (6.2)$$

Figure 44 – Multiple sources hourglass networks for multi-focus image fusion. Sources are showed separated in the figures but the input block is 6 channels depth map. For HF-Reg the output layer corresponds to the all-in-focus regressed image. In the case of HF-Seg, the output layer is a 2-channel feature map, and values $z_i(p)$ represents the probability of selecting pixel p from input source i .



Source: PEÑA *et al.* (2019a)

Training of such network requires the ground truth of the fusion map for every input pair \mathbf{x} to be known. However, during the synthetic sources generation, the focus map g^b is obtained. Then, the HF-Seg training is carried out by using the Binary Cross Entropy (BCE) loss function:

$$\mathcal{L}_S(\mathbf{z}, g^b) = -\frac{1}{|\Omega|} \sum_{p \in \Omega} g^b(p) \cdot \log(z_0(p)) + (1 - g^b(p)) \cdot \log(z_1(p)) \quad (6.3)$$

where $\mathbf{z} = (z_0, z_1)$ is the output of HF-Seg and g^b is created as described in Section 6.2.1. In terms of optimization, this approach follows the same idea of previous chapters, despite the amount of data makes it possible to use BCE directly without further weighting.

All-in-focus image regression (HF-Reg network). Although the HF-Seg approach is straightforward, the fusion rule has to be previously established (Equation 6.2). Then, this network works better in problems where a focus map and a fusion rule can be used, such as in multi-focus image fusion. However, a more general model can be derived from the HF-Seg method to learn the best fusion rule for source combination automatically. This second proposal uses an end-to-end approach where the hourglass network is used to regress the all-in-focus image directly. Here, the fusion function input is also a 6-channel map. The architecture remains as a sequence of convolutions and max-pooling in the encoder and convolutions-upsampling blocks in the decoder. Differently, to the segmentation approach, the output feature block is a 3-channels map \hat{y} corresponding to an RGB focused image. In this approach, the learning process requires an appropriate regression loss function rather than the BCE. In this context, let $y = (y_0, y_1, y_2)$ be a ground truth focused image where y_0 , y_1 and y_2 are its RGB channels respectively. Similarly, the estimated RGB all-in-focus image is given by $\hat{y} = (\hat{y}_0, \hat{y}_1, \hat{y}_2)$. The regression loss function is defined as in Equation 6.4, where φ_α is an intensity dissimilarity function. Note that the

proposed loss function is the sum of the mean distance for each channel, rather than the mean distance of all channels. This per-channel loss has shown to be better for color estimation because averaging the errors of the 3 channels usually leads to a grayscale output space. Also, when values are regressed, the output space during training is not bounded as opposed to the previous segmentation approach. This lack of boundaries can bring difficulties to obtain an output map in the expected range. To this end, a regularization term that forces the convergence of minimum and maximum values of each channel was added to the loss function. This regularization term penalizes more severely fused images with low contrast or intensity values outside the interval $[0, 1]$, assuring the output map to be in the right range. The idea behind this regularization is to create mountains in the loss landscape wherever a solution leads to an invalid image, *e.g.*, $\hat{y}(p) \notin [0, 1]$.

$$\mathcal{L}_R(\hat{y}, y) = \frac{1}{|\Omega|} \sum_{i=0}^2 \sum_{p \in \Omega} \varphi_\alpha(y_i(p), \hat{y}_i(p)) + \sum_{i=0}^2 |\min(y_i) - \min(\hat{y}_i)| + \sum_{i=0}^2 |\max(y_i) - \max(\hat{y}_i)| \quad (6.4)$$

Among the possible dissimilarity functions such as the Mean Square Error or L2 norm, and L1 norm, here is defined φ_α as the Normalized Positive Sigmoid (NPS) between two intensities parameterized by α :

$$\varphi_\alpha(y, \hat{y}) = \frac{2}{e^{-\alpha \cdot |y - \hat{y}|} + 1} - 1 = \frac{e^{\alpha \cdot |y - \hat{y}|} - 1}{e^{\alpha \cdot |y - \hat{y}|} + 1} \quad (6.5)$$

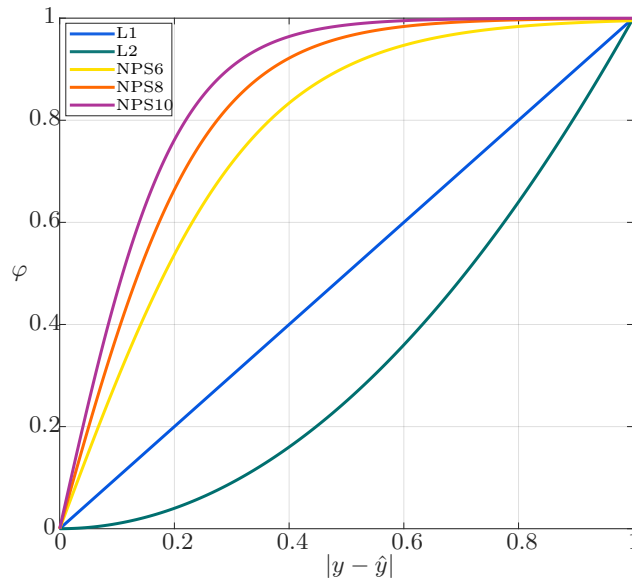
Given the ground truth intensity y and the estimated intensity \hat{y} , the minimum metric value is obtained when $\hat{y} = y$, $\varphi_\alpha(y, y) = 0$. Also, the maximum value is approximately 1 for $\alpha > 5$, $\lim_{|y - \hat{y}| \rightarrow \infty} \varphi_\alpha = 1$. Furthermore, with the proposed NPS, a lower decay is observed when compared to the usual L2 and L1 approaches. This behavior forces the creation of leaned surfaces favoring a faster convergence of the optimizer. Figure 45 shows the error mapping for the L1 norm, L2 and NPS for different values of α , *e.g.*, NPS6, NPS8 and NPS10 corresponding to $\alpha = 6, 8$ and 10 , respectively.

The simplicity and power of the proposed network allow performing image fusion without further post-processing. This regression approach concedes the learning of the best fusion function, and it is not limited to problems where the fusion map can be obtained, *e.g.*, multi-modal fusion, and multi-exposure fusion.

6.2.3 Implementation Details

To fulfill the commutative law, required for all fusion functions, an appropriate training protocol was employed. For every generated tuple $\mathbf{x} = (x_A, x_B)$, the inverted tuple $\bar{\mathbf{x}} = (x_B, x_A)$ was also forwarded in the same minibatch. In the HF-Reg network training, any further ground truth modification for $\bar{\mathbf{x}}$ is needed, because the all-in-focus image y remains the same. However, for the HF-Seg approach, the ground truth focus map needs

Figure 45 – Distances mapping for L1, L2, NPS6, NPS8 and NPS10 dissimilarity functions.



Source: PEÑA *et al.* (2019a)

to be inverted, *e.g.*, $1 - g^b$, so the obtained reconstruction remains as close as possible to y .

Because the best pixel value that can be obtained belongs to one of the sources, *e.g.*, the multi-focus image fusion problem can be seen as a selection problem where $y(p)$ is either equal to $x_A(p)$ or $x_B(p)$, a posterior post-processing for selecting the nearest value can be applied. Let \hat{y} be a fused image obtained by the regression network $f_R(\mathbf{x})$. The final all-in-focus image is obtained as follows:

$$\hat{y}^*(p) = \begin{cases} x_A(p) & \text{if } \|\hat{y}(p) - x_A(p)\|^2 < \|\hat{y}(p) - x_B(p)\|^2 \\ x_B(p) & \text{otherwise} \end{cases} \quad (6.6)$$

6.3 EXPERIMENTS AND RESULTS

To evaluate and validate the proposed approach several experiments were conducted. For establishing a comparison the Image Matting for fusion (IM) (LI *et al.*, 2013), the variance based image fusion in Discrete Cosine Transform domain (DCT) (HAGHIGHAT; AGHAGOLZADEH; SEYEDARABI, 2010) and with consistency verification (DCT+CV) (HAGHIGHAT; SEYEDARABI, 2011), the Guided Filtering Fusion (GFF) (LI; KANG; HU, 2013) and the deep Convolutional Neural Network (CNN) (LIU *et al.*, 2017) approaches were used. The nearest source color post-processing explained in the previous section is referred to as Near. The experiments were conducted over synthetic and real datasets with different number of images within the burst.

Segmentation and regression networks were trained over the proposed synthetic multi-focus dataset using the provided training split in the MS COCO dataset. The optimizer Adam (KINGMA; BA, 2015) with its defaults parameters was applied, and the initial learning rate was set to 10^{-5} . The number of epochs and mini-batch sizes was 1000 and 3 respectively. For training purpose random crops of 400×400 and random mirroring were applied. The size of the crops are mainly determined by the RAM of the video card used for training. The network’s initialization was made with normally distributed weights using Xavier’s method (GLOROT; BENGIO, 2010). For test phase it was used the test split of MS COCO dataset for synthetic data creation and the Lytro dataset for real-image experimentation. In this last one, the size of the original images was used since, after learning the kernels, the networks are size invariant.

6.3.1 Commutativity

All fusion functions must produce the same all-in-focus image no matter the order of the sources frames are presented. Since the hourglass network input is a six-channel map, \mathfrak{x} and $\bar{\mathfrak{x}}$ represents different objects, and therefore, the output might be different. However, due to the training protocol, the learned fusion function leads to approximately the same point in the output space for inputs \mathfrak{x} and $\bar{\mathfrak{x}}$, ensuring the required commutative property. Figure 46 shows two different pairs \mathfrak{x} from the real dataset, and the results obtained applying a forward of the tuple and its reverse into each proposed network. As can be seen, no significant differences are observed in the all-in-focus images. The obtained mean squared error between $f(\mathfrak{x})$ and $f(\bar{\mathfrak{x}})$ was in the order of 10^{-5} for all images and can not be visually perceived. This property remained for all tested images.

6.3.2 Multi-focus Image Fusion Metrics

Quantitative evaluation analysis for image fusion problems is a challenging task since the reference all-in-focus images are unknown. Among the several proposals introduced in the literature, there is no precise measurement that is considered the best. Here some of the most used metrics like Normalize Mutual Information Q_{MI} , Tsallis Entropy Q_{TE} , Nonlinear Correlation Information Entropy Q_{NCIE} , Gradient-based Q_G , Phase Congruency Q_P , Piella-Heijmans Q_S , and Chen-Blum Q_{CB} are explored. For the Q_{MI} was followed Hosny definition because it reduces the bias of the original Q_{MI} metric toward the sources. Every metric belongs to one of the four groups of objective assessment metrics, information theory, feature-based, structural similarity-based, and human perception inspired. Higher metrics values mean better fusion quality. A detailed explanation of each of the metrics can be found in (LIU et al., 2012). Despite the generalized use of these metrics, it was found that computing the agreement of the resulting image with every source, including blurred regions of the sources, may not represent a good measurement of the fusion quality. Liu et al. (LIU et al., 2012) also arrives at this conclusion in their work.

Figure 46 – Example of fusion results for tuples with normal and reversed order. In the first row are shown the frames within the tuples. In second and third row are shown the fusion results with the regression and segmentation networks respectively for both normal and reverse order evaluations.







Source: PEÑA *et al.* (2019a)

An example of bias toward the source is shown in Figure 47. The first image in the figure refers to the output of the HF-Reg network without Near post-processing, followed by the same image after the nearest post-processing. Dummy A and Dummy B images correspond with the outputs of the methods that return exactly the source A and B, respectively. As can be seeing in the figure, most metrics get higher values when the output is one of the sources. This means that a dummy method that outputs an input image has a better metric value than others that returns a visually acceptable all-focused image. The behavior is expected because most of the metrics find a quality value using the similarity between the resulting image within each source. Then, when an all-in-focus image is obtained with a subtle colors variation respect to the sources, the metrics values

highly decrease as in the case of HF-Reg network without Near. The values of the metrics for dummies methods in this example even super-passes most of the literature methods, so caution must be taken when using objective assessment metrics to give a conclusive result. The full reference Structural SIMilarity index (SSIM) between the resulting fused image and the all-in-focus ground truth was also computed in the synthetic dataset for a stronger comparison.

Figure 47 – Example of the values of the fusion metrics for HF-Reg without and with Near post-processing and two dummy methods that returns the first (Dummy A) and second (Dummy B) image of the tuple as a result for the fusion.

	HF-Reg (without Near)	HF-Reg (with Near)	Dummy A	Dummy B
				
Q_{MI}	0.8463	1.1097	1.2812	1.2812
Q_{TE}	0.3616	0.3766	0.4432	0.4435
Q_{NCIE}	0.8212	0.8336	0.8631	0.8628
Q_G	0.6255	0.6768	0.5330	0.6614
Q_P	0.7151	0.7610	0.7210	0.8007
Q_S	0.9510	0.9473	0.8536	0.8841
Q_{CB}	0.7336	0.7806	0.6955	0.7591

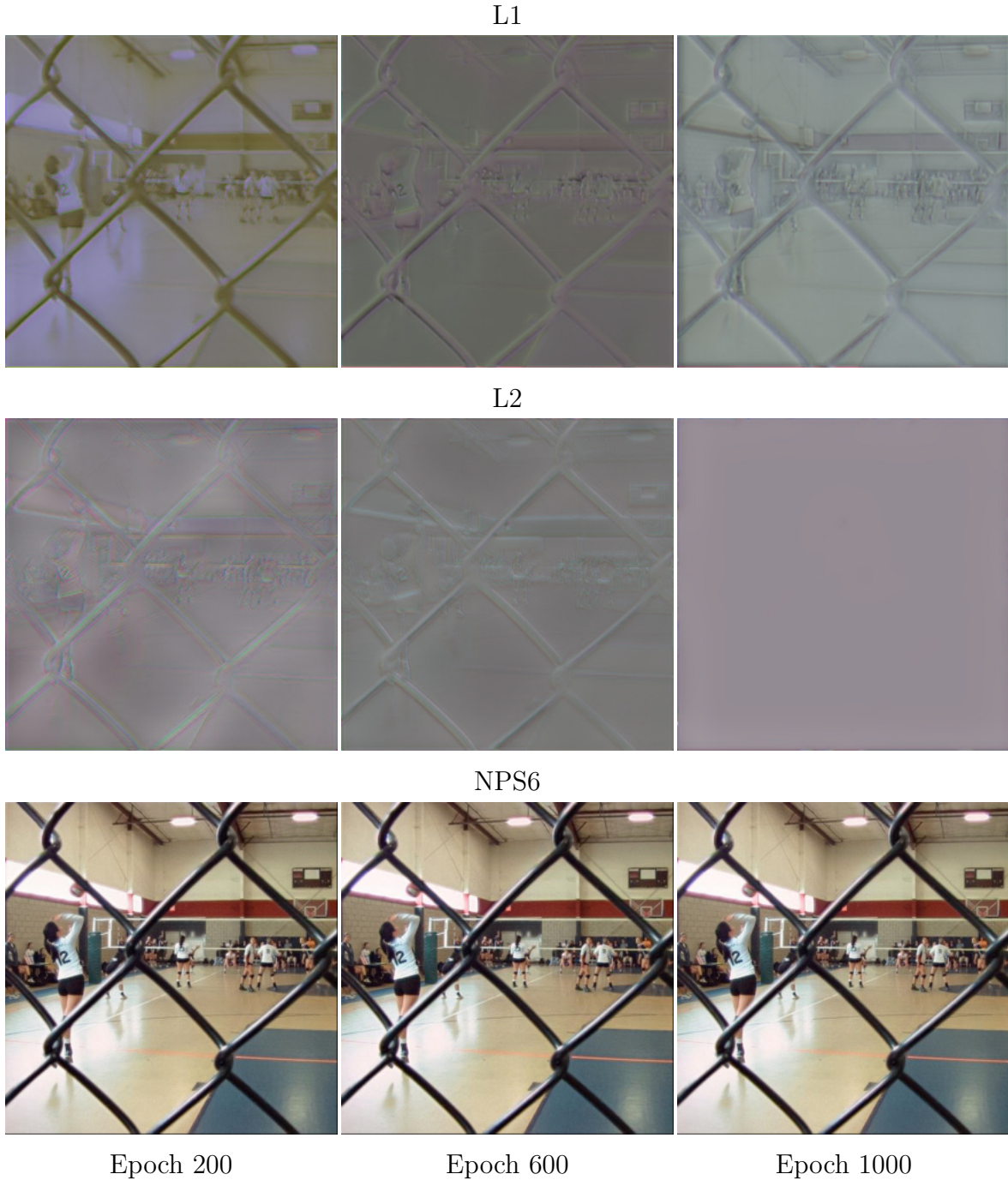
Source: PEÑA *et al.* (2019a)

6.3.3 Loss Function - L1 vs L2 vs NPS

This experiment has the objective of demonstrating the feasibility of the proposed NPS loss function. For this, the HF-Reg architecture was trained over the synthetic MS COCO multi-focus dataset but using L1, L2, and the proposed NPS6. The training hyper-parameters are the same as described at the beginning of the section. After training during 1000 epochs, a synthetic multi-focus test dataset was created for evaluation purposes. This dataset was composed of 100 randomly selected images from the test data of the MS COCO panoptic segmentation, and then the multi-focus data creation process was applied. Despite the usefulness of the L1 and L2 loss functions in other regression problems, here was found it difficult to regress the appropriated all-in-focus image. The obtained output during different epochs of the training are shown in Figure 48 for every training function over a real image from the Lytro dataset. It can be observed that L1 and L2 loss functions fail to obtain a valid image during the entire training. However,

with the proposed NPS6 loss function, the colors and contrast of the regressed image are well estimated even in earlier epochs.

Figure 48 – Example of the multi-focus image fusion obtained with intermediate L1, L2 and HF-Reg networks during the training.



Source: PEÑA *et al.* (2019a)

The behavior is corroborated by the mean error curve over the synthetic dataset (Figure 49). This figure shows the mean L1 difference between estimated all-in-focus image \hat{y} and the ground truth y over different epochs. The y-axis is shown in log scale for

better interpretation. It can be seen a better convergence when NPS is used, succeeding to obtain a visually good solution for the MFIF problem.

Figure 49 – Logarithm of the error over the synthetic multi-focus test dataset.

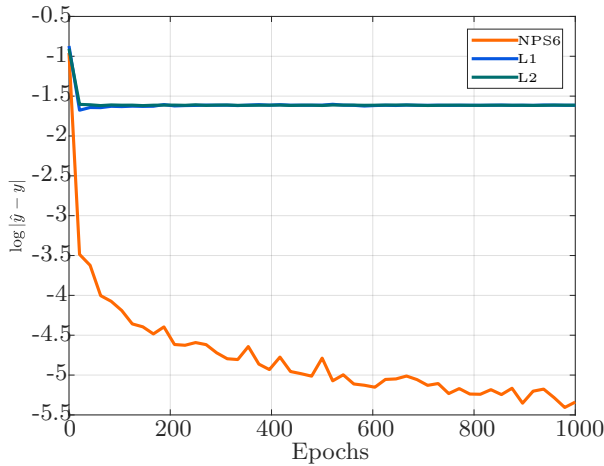
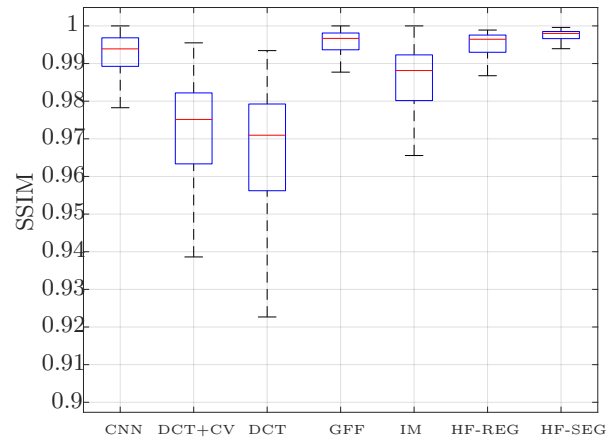


Figure 50 – Box plot for SSIM reference metric over the synthetic multi-focus test dataset.



Source: PEÑA *et al.* (2019a)

6.3.4 Two Source Synthetic Dataset

For evaluation purposes, the proposed methods were evaluated in the synthetic multi-focus test dataset. The dataset had 100 pairs with its corresponding all-in-focus ground truth. Because the reference image is known, the SSIM metric between the obtained reconstruction and the ground truth was used in the evaluation. Figure 50 shows the obtained box plots with the SSIM metric for every tested method. A high mean with a small variance is observed for HF-Seg method that has values of SSIM nearly to 1 for most of the pairs. The HF-Reg also behaves well, obtaining comparable results to GFF and lower variance with respect to CNN. In three of the seven objective assessment metrics, the proposals had higher mean and lower variance than the state-of-the-art (Table 8). However, despite the higher mean value in some references and multi-focus metrics, no statistically significant difference was measured for the results of the CNN, GFF, HF-Reg, and HF-Seg, according to the Friedman (FRIEDMAN, 1940) test and Nemenyi (NEMENYI, 1962) post-hoc.

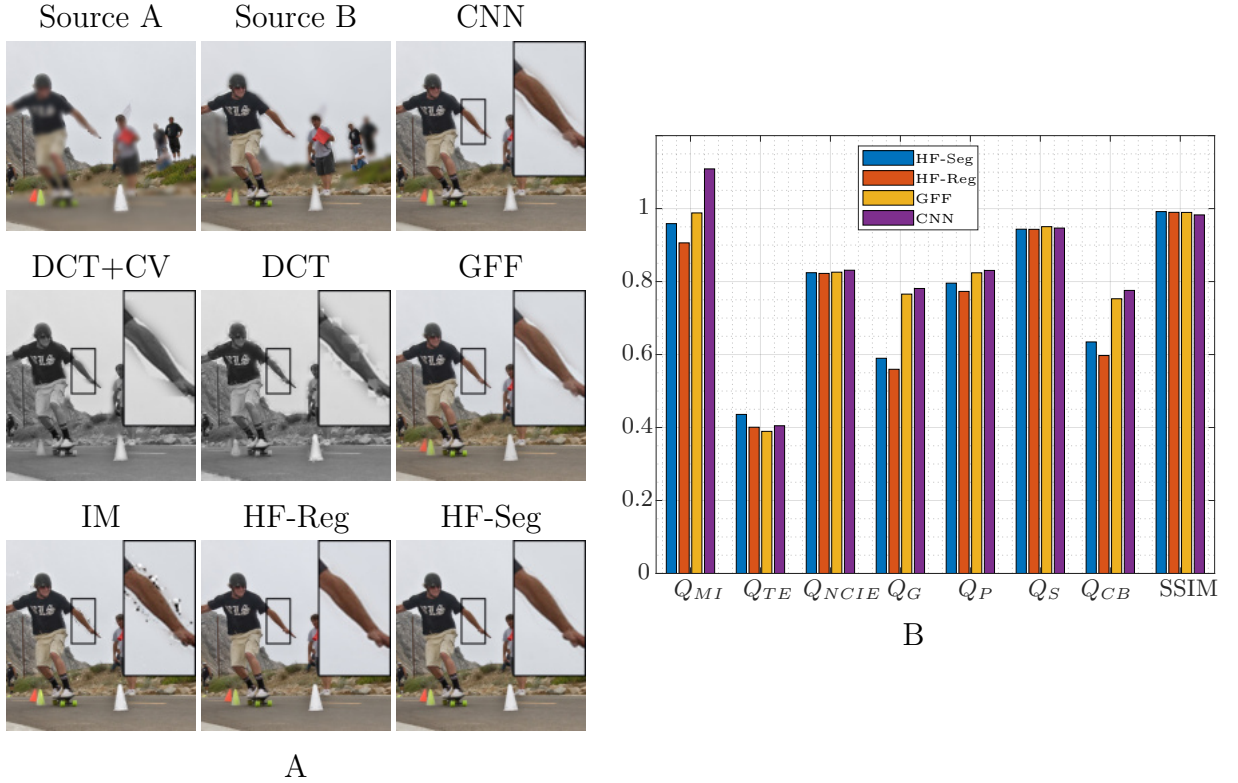
For almost every pair in the synthetic test dataset, the CNN, GFF, HF-Reg, and HF-Seg approaches return a similar focused image with very few differences in terms of pixels colors. However, as stated before, some times, the metrics can confuse the judgment of the fusion quality, as shown in the example of Figure 51A. For this pair, CNN and GFF outperform the proposed approaches in term of metrics except for Q_{TE} and SSIM (bar graph in Figure 51B), but on a visual inspection over Figure 51A, HF-Reg, and HF-Seg obtained a better quality fusion.

Table 8 – Mean and standard deviation of the objective assessment over the synthetic multi-focus test dataset.

Metrics	CNN	DCT+CV	DCT	GFF	IM	HF-Reg (Proposed)	HF-Seg (Proposed)
Q_{MI}	1.1467 ± 0.1474	0.9014 ± 0.1777	0.8827 ± 0.1726	1.0920 ± 0.1695	1.1350 ± 0.1501	1.1828 ± 0.1107	1.1924 ± 0.1156
Q_{TE}	0.4101 ± 0.0412	0.3869 ± 0.0458	0.3810 ± 0.0452	0.4049 ± 0.0417	0.4055 ± 0.0418	0.4129 ± 0.0328	0.4152 ± 0.0352
Q_{NCIE}	0.8425 ± 0.0111	0.8275 ± 0.0100	0.8263 ± 0.0092	0.8390 ± 0.0121	0.8418 ± 0.0113	0.8432 ± 0.0087	0.8445 ± 0.0095
Q_G	0.7499 ± 0.0405	0.6788 ± 0.0620	0.6759 ± 0.0615	0.7526 ± 0.0390	0.7365 ± 0.0447	0.6714 ± 0.0895	0.7182 ± 0.0548
Q_P	0.7985 ± 0.0816	0.7376 ± 0.0870	0.6959 ± 0.0963	0.7964 ± 0.0811	0.7426 ± 0.0831	0.7414 ± 0.1110	0.7722 ± 0.0938
Q_S	0.9566 ± 0.0159	0.9411 ± 0.0211	0.9408 ± 0.0210	0.9586 ± 0.0144	0.9440 ± 0.0210	0.9493 ± 0.0174	0.9548 ± 0.0153
Q_{CB}	0.8198 ± 0.0383	0.7112 ± 0.0621	0.6838 ± 0.0666	0.8125 ± 0.0376	0.7950 ± 0.0515	0.7449 ± 0.0844	0.7719 ± 0.0572

Source: PEÑA *et al.* (2019a)

Figure 51 – Example of synthetic test example, most methods have (B) higher values in objective assessment metrics. However, with a visual inspection (A) it can be observed that the proposed methods show a better quality fusion.



Source: PEÑA *et al.* (2019a)

6.3.5 Two Sources Real Dataset

The Lytro two sources dataset was used to evaluate the methods under real multi-focus environment. This dataset has 20 pairs of multi-focused images captured with the Lytro camera that uses the Light-field technology, allowing to expand the depth of field after the image was taken. Since the all-in-focus ground truth is not available, only the objective assessment metrics were used in this experiment. Table 9 shows the same behavior than in synthetic setting, *e.g.*, the proposals had higher mean and lower variance for the first three metrics.

Table 9 – Mean and standard deviation of the objective assessment over the Lytro multi-focus two sources dataset.

Metrics	CNN	DCT+CV	DCT	GFF	IM	HF-Reg (Proposed)	HF-Seg (Proposed)
Q_{MI}	1.1467 ± 0.1107	0.8476 ± 0.1419	0.8347 ± 0.1403	1.0932 ± 0.1209	1.1376 ± 0.1045	1.1538 ± 0.0865	1.1758 ± 0.0968
Q_{TE}	0.3994 ± 0.0299	0.3702 ± 0.0380	0.3656 ± 0.0381	0.3969 ± 0.0320	0.3961 ± 0.0287	0.3984 ± 0.0268	0.4020 ± 0.0286
Q_{NCTE}	0.8425 ± 0.0080	0.8259 ± 0.0081	0.8251 ± 0.0077	0.8390 ± 0.0081	0.8420 ± 0.0078	0.8423 ± 0.0066	0.8443 ± 0.0076
Q_G	0.7234 ± 0.0280	0.6939 ± 0.0328	0.6853 ± 0.0353	0.7182 ± 0.0307	0.7159 ± 0.0301	0.6636 ± 0.0420	0.7096 ± 0.0315
Q_P	0.8488 ± 0.0395	0.8140 ± 0.0490	0.7633 ± 0.0658	0.8465 ± 0.0395	0.8205 ± 0.0472	0.8004 ± 0.0432	0.8387 ± 0.0408
Q_S	0.9466 ± 0.0124	0.9377 ± 0.0143	0.9367 ± 0.0147	0.9467 ± 0.0123	0.9419 ± 0.0131	0.9418 ± 0.0131	0.9447 ± 0.0130
Q_{CB}	0.8058 ± 0.0381	0.7230 ± 0.0395	0.7030 ± 0.0466	0.7929 ± 0.0400	0.7922 ± 0.0408	0.7550 ± 0.0477	0.7898 ± 0.0439

Source: PEÑA *et al.* (2019a)

There was not a statistically significant difference in the values of the metrics for the proposals compared to the CNN and GFF approaches. Some examples of the obtained all-in-focus images with HF-Reg and HF-Seg are shown in Figure 52.

Figure 52 – Example of fusion results with HF-Reg and HF-Seg over the Lytro two sources real dataset.

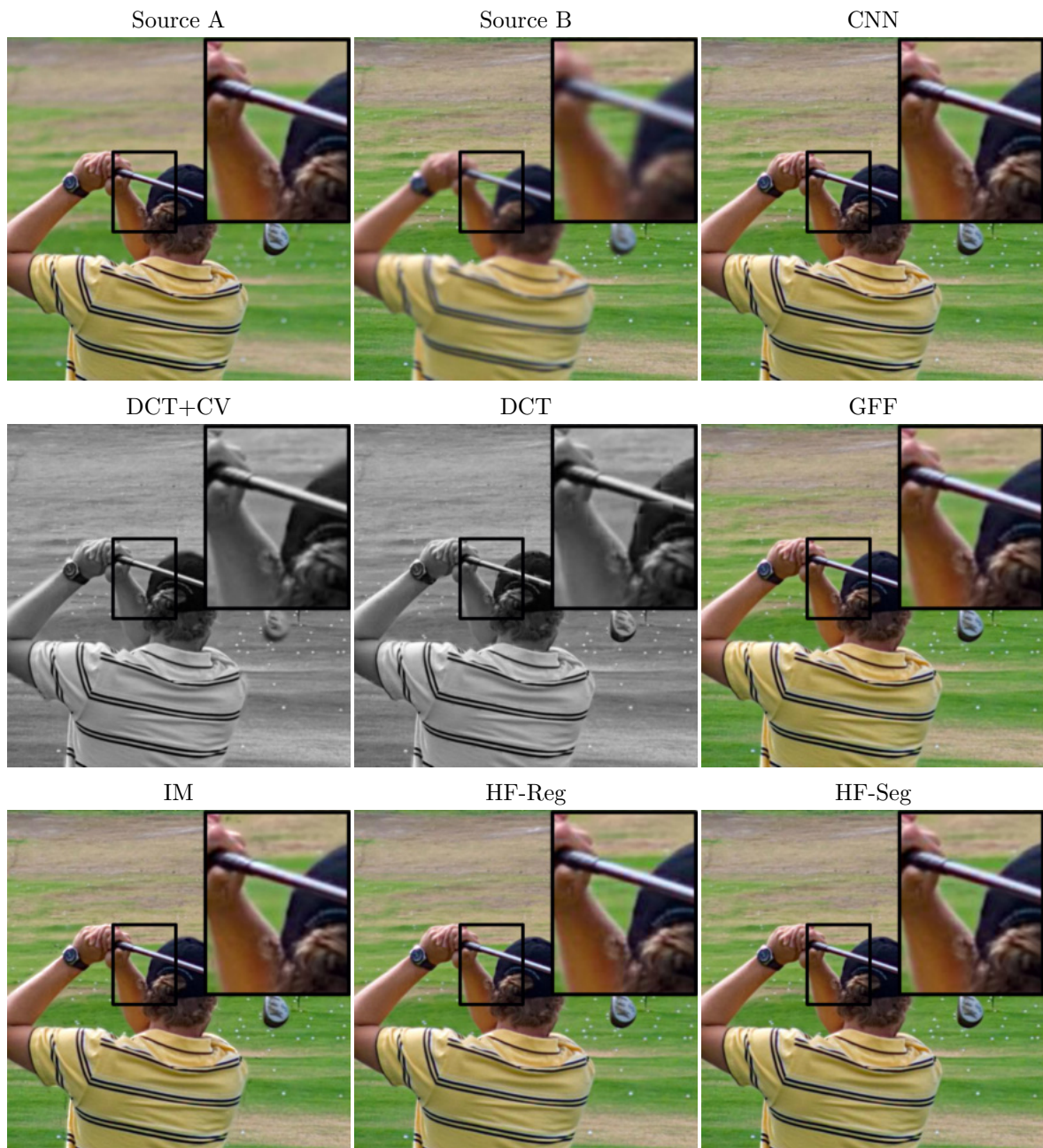


Source: PEÑA *et al.* (2019a)

However, an advantage of the proposals is that it was not applied any further morphological operation in the post-processing step. The problem with this kind of operation is that the size and shape of the structural elements restrict the solution space. An example

of this is shown in Figure 53 for the "golf" image of the Lytro dataset. A visually comparable result is obtained with CNN, GFF, HF-Reg, and HF-Seg. But, a careful inspection into the marked area reveals that, contrary to HF-Reg and HF-Seg, the consistency verification steps in CNN and GFF causes a wrong fusion in the gap region.

Figure 53 – Example of fusion results with different literature methods and the proposed HF-Reg and HF-Seg methods over the "golf image" of the Lytro 2 dataset.



Source: PEÑA *et al.* (2019a)

6.3.6 Three Sources Real Dataset

To show the performance of the methods with more than two sources was used the Lytro three-source real dataset. The dataset has four triplets of multi-focused images also captured with the Lytro camera. The 3-functional power of fusion functions was computed in each case. Since the objective assessment metrics are defined for two sources, the evaluation was done visual and as observed in Figure 54, the methods can correctly obtain an all-in-focus image. Here, a better reconstruction is obtained with the HF-Seg network for the keyboard triplet fusion. These results are obtained because the accumulation of errors during the fusion is worst when a regression is done.

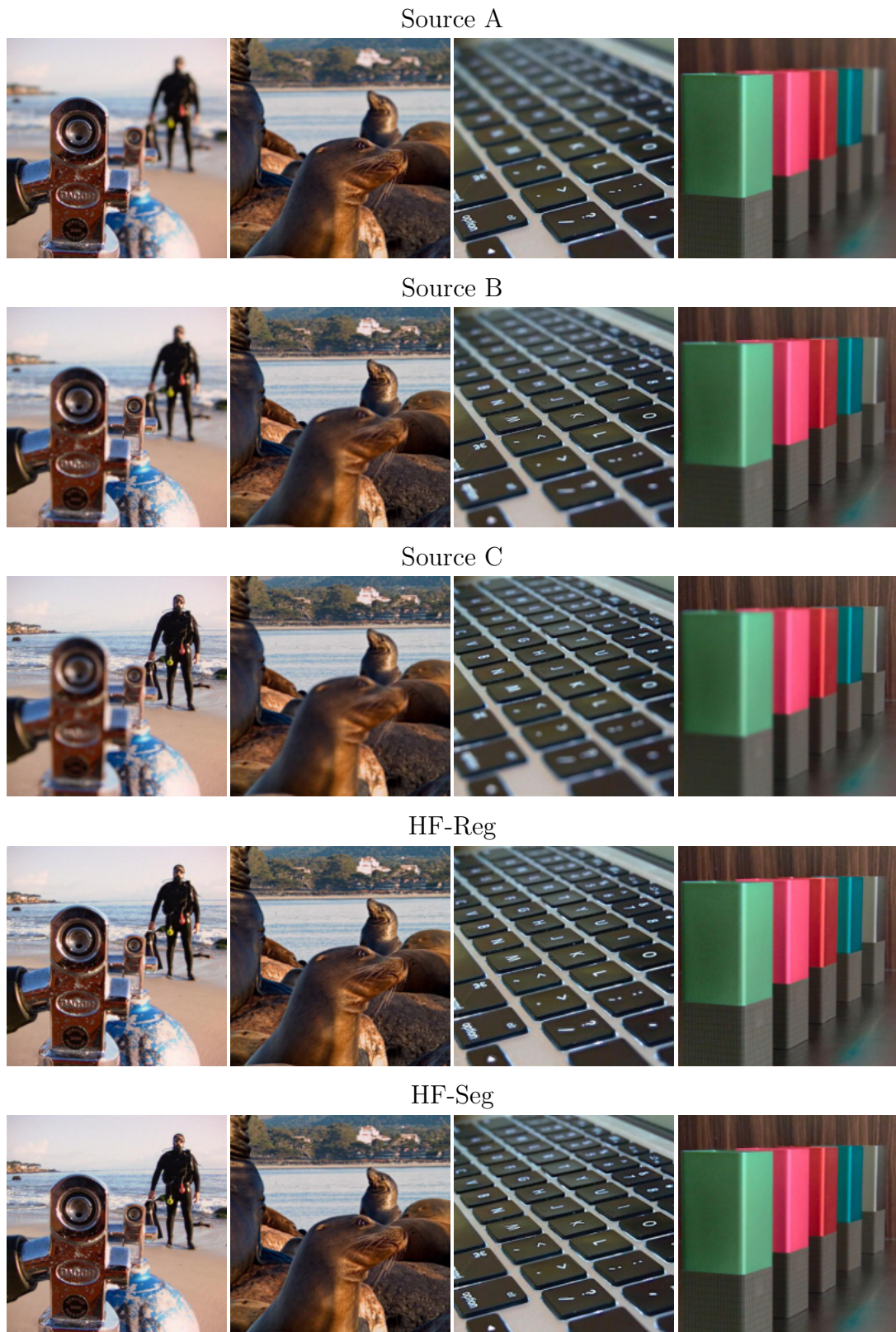
6.3.7 Execution Time

One of the core concerns of MFIF methods is to have low execution time. To better comparison, the original implementations proposed by the authors of the compared methods were used. For a fair evaluation was also included the time reported by the authors of CNN (LIU et al., 2017) since the available implementation in Matlab is much slower than the reported. All methods were tested on the same computer with an Intel(R) Core(TM) i7-6800K 3.40 GHz CPU and 64GB RAM. An Nvidia GeForce GTX 1070 GPU with PyTorch deep learning framework was used for HF-Seg and HF-Reg. The time for loading the data was not considered for all methods. The synthetic multi-focus image dataset with 100 pairs was used for the experiment. Three different image sizes 520×520 , 260×260 and 130×130 were tested. Table 10 shows the average execution time for the 100 images pairs. As can be seen, the proposals have high computational efficiency when compared with the other methods. This computation result makes the methods appropriated for near real-time applications where the multi-focus fusion is required. As shown in the experiments, this high efficiency does not decrease the performance that is comparable or superior in most situations to the state-of-the-art.

6.3.8 Applications of HF-Reg

The HF-Reg network can regress an image that does not need to be composed of pixels of the source if Near post-processing is not considered, obtaining an improved filtered version. An example of this can be observed in Figure 55 for an HF-Reg network trained during 500 epochs for a multi-focus fusion of noisy inputs. This kind of filtering and fusion was achieved by only applying Gaussian noise with a variable variance to the synthetic sources, and applying the previously described training protocol with NPS6.

Figure 54 – Example of fusion results over the Lytro three sources real dataset.



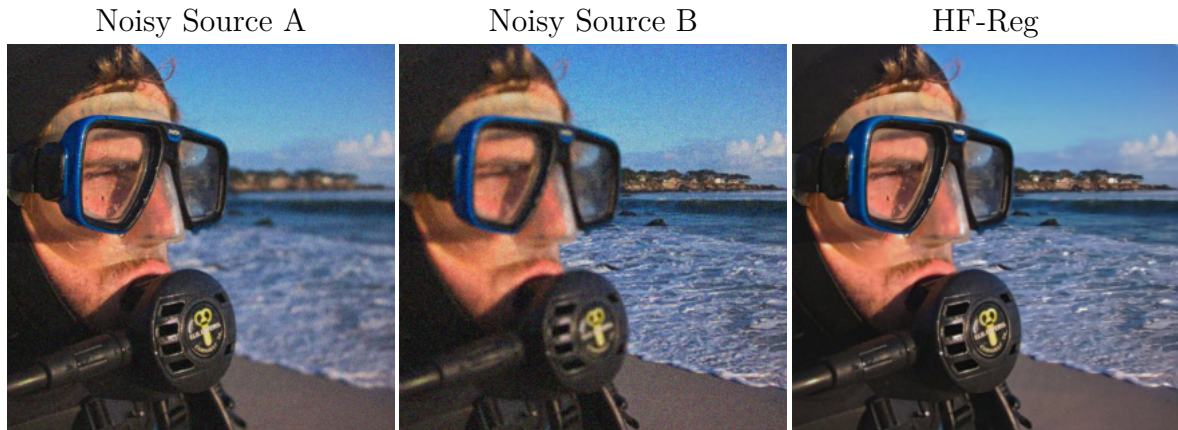
Source: PEÑA *et al.* (2019a)

Table 10 – Execution time for each mfif method with three different image size. Time unit is second.

Method	520×520	260×260	130×130
CNN GPU (reported in (LIU et al., 2017))	0.7800	-	-
CNN slight GPU (reported in (LIU et al., 2017))	0.3300	-	-
CNN (Matlab)	94.7100	33.8200	12.6821
IM	2.7095	0.8112	0.4870
DCT+CV	0.7648	0.2982	0.1949
GFF	0.1280	0.0373	0.0149
HF-Seg	0.0026	0.0023	0.0022
HF-Reg	0.0023	0.0022	0.0021

Source: PEÑA *et al.* (2019a)

Figure 55 – Example of multi-focus image fusion and filtering of noisy sources.



Source: PEÑA *et al.* (2019a)

6.4 CONCLUSIONS

In this chapter a multi-source architecture for multi-focus image fusion problem was presented. Supervised learning was attained by using a new multi-focus synthetic data creation. The first approach uses a semantic segmentation and the weighted average fusion rule for obtaining the all-in-focus image. A new pixel-level regression loss function was proposed obtaining comparable results to a network trained with HF-Seg method, but with the additional capability of filtering the sources. Regression based solution achieved comparable results to state-of-the-art methods while trained to learn both the activity level measurement and the fusion rule at once. Regression based solution proved to be more suitable than traditional L2 and L1 loss functions. Both approaches showed comparable results with literature methods, but with a higher computational efficiency.

7 CONCLUSIONS

In this work, new loss functions were proposed aiming to improve the performance of U-NET architecture for pixel-level classification and regression tasks. For studying the behavior of the proposals, biomedical image instance segmentation and multi-focus image fusion tasks were chosen. For all studied tasks, proper loss function modeling improved the performance of the fixed architecture. The subset of models obtained in this research led to acceptable solutions. Additionally, experimentation suggests the applicability of the proposed loss functions with any existing pixel-level architecture for improving its performance. This result reinforces the idea that there is no need to further overparametrize models for finding feasible solutions.

For fully supervised biomedical image segmentation, two new loss functions were developed on top of a proposed three classes semantic segmentation framework. The proposals included shape information in the form of weight maps for penalizing more severely the errors made on underrepresented parts of the image. Even with a small training dataset, only ten images, the F1 detection rate improvement was of 0.25 when compared with similar approaches from the literature. The best segmentation rate was also obtained with the proposal with F1 values of 0.8860. This performance was attained by using a new thresholded map post-processing.

A new W^3 loss function was proposed for accounting weak supervision using near to zero weights for the background. By further applying a new touching contrast modulation over the input image and increasing the amount of training data a model with Panoptic Quality values of 0.79 was founded. W^3 detection precision was above 0.90, improving the performance over a network trained with UNET weight map (0.87). Also, the Watershed-based post-processing proposed here was able to further enhance segmentation with PQ improvement of 0.03 over the proposed thresholded maps approach, and 0.24 over Maximum A Posteriori. In general, the method proved to be good in separating adjoining cells with low touching probability between them, and to remove spurious regions in the probability map. The proposed semantic based methodology also showed improvement over the detection based approach Mask R-CNN, with a PQ difference of 0.09. Zero-shot instance segmentation over Meristem and Sepal images was also achieved by using only twenty-eight T-cells images for training the network. This is a very encouraging result considering the inability of deep neural networks to generalize well in domains never seen before when a small training dataset is used.

A loss function that uses a surrogate for Youden index as a regularization term was introduced for generalization to 3D weakly supervised instance segmentation. The approach works on four classes with the proposed semantic framework, aiming to obtain better contour adequacy. The proposed regularization term showed to be sufficient for segmentation

in highly imbalance classes cases. The approach further improved the performance over the T-cell dataset obtaining PQ values of 0.81 by using the parameter-free post-processing Maximum A Posteriori. The detection rate was improved to a great extent missing only 9 cells of 138. The quality of obtained segmentation was also improved, and was verified by visual inspection that in some cases surpassed human annotations. The experiments in several 2D and 3D biological image datasets showed similar performance, PQ above 0.77. The analysis of the loss landscape suggest a better differentiation when the proposed $J4$ loss function is used. 3D instance segmentation of $1024 \times 1024 \times 508$ volume was achieved in nine minutes which is a very low execution time considering the complexity of the task, detecting 6890 cells of 7128. Zero-shot segmentation was also attained with Youden based loss function showing the feasibility of the proposal.

Finally, a new pixel-level loss function for regression problems was proposed. The function was used in multi-focus image fusion problem, proving to be sufficient for providing feasible solutions while similar loss functions of the literature were not able to find. The particularities of the task, *e.g.*, input is a pair of images, lead to the creation of a new bi-variable U-Net, resulting just in a small increase of 0.006% of the number of parameters. The visual inspection showed the benefits of the proposed regression approach, being able to learn the best fusion rule for solving the task. For better comparison of the results other methods from the literature were used. In particular, the fusion around the contours of the objects in the scene was more visually pleasant with the regression approach than with the approaches considered in this study. Supervised training was achieved thanks to the proposed synthetic data creation that proved to be good enough for generalizing to real acquired bursts. According to the explored metrics, the results were comparable to other approaches, but with significant improvements in processing time (60 times faster than GFF). Finally, a joint fusion and filtering approach proved the feasibility of the proposal for multi-focus image fusion of noisy sources, that is not possible with current pixel selection methods.

7.1 LIMITATIONS

In this work it is assumed that exist at least one combination of the weights that can provide an acceptable solution. However, there is not a guarantee that for any given architecture this always holds. Instance segmentation methods in this work are limited to panoptic cases, *e.g.* disjoint connected components. This means that images with overlap between instances are segmented assuming an empty intersection. The most common case of failure in the proposed semantic segmentation based approach is dangling touching separation that cause merging neighboring connected components. Despite the proven advantages of proposed multi-focus image fusion approaches, exist a limitation for doing a proper quantitative analysis. This is related with the difficulties to acquire a real dataset, that impede using reference metric. Furthermore, current non-reference metrics showed

to be insufficient to correctly quantify the performance.

7.2 FUTURE WORKS

Several topics may be useful for further investigation as future works to this research. Using other architectures in combination with proposed loss functions seems to be a promising path for improving performance over different tasks. Finding new representations for the output layer is another interesting topic that can lead to generalizing this work for instance segmentation but considering non-disjoint connected components. Unsupervised approaches for automatic determination of the best number of classes in instances segmentation methods could be also examined. Investigating dynamic loss functions that changes its behavior over time is also an interesting approach to future research. Incorporation of perceptual terms in the regression loss can be also investigated for learning more robustly objects gradient. Further noise sensibility analysis for the proposed regression loss function is also recommended. Applying the proposed pixel-level regression loss for other kind of image restoration task seems to be also promising.

REFERENCES

- ABRAHAM, N.; KHAN, N. M. A Novel Focal Tversky Loss Function with Improved Attention U-Net for Lesion Segmentation. In: *Proceedings of the IEEE International Symposium on Biomedical Imaging (ISBI)*. Italy: IEEE, 2019. p. 683–687.
- AITTALA, M.; DURAND, F. Burst Image Deblurring Using Permutation Invariant Convolutional Neural Networks. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Germany: Springer, 2018. p. 731–747.
- ALHASHIM, I.; WONKA, P. High Quality Monocular Depth Estimation Via Transfer Learning. *arXiv preprint arXiv:1812.11941*, 2018.
- BAI, M.; URTASUN, R. Deep Watershed Transform for Instance Segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Hawaii: IEEE, 2017. p. 5221–5229.
- BENGIO, Y. Practical Recommendations for Gradient-based Training of Deep Architectures. In: *Neural Networks: Tricks of the Trade*. Berlin: Springer, 2012. p. 437–478.
- BERMAN, M.; TRIKI, A. R.; BLASCHKO, M. B. The Lovász-softmax Loss: A Tractable Surrogate for the Optimization of the Intersection-over-Union Measure in Neural Networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Utah: IEEE, 2018. p. 4413–4421.
- BOUGHORBEL, S.; JARRAY, F.; EL-ANBARI, M. Optimal Classifier for Imbalanced Data Using Matthews Correlation Coefficient Metric. *PloS One*, Public Library of Science, v. 12, n. 6, p. e0177678, 2017.
- BRABANDERE, B. D.; NEVEN, D.; GOOL, L. V. Semantic Instance Segmentation with a Discriminative Loss Function. *arXiv preprint arXiv:1708.02551*, 2017.
- BRAY, A. J.; DEAN, D. S. Statistics of Critical Points of Gaussian Fields on Large-dimensional Spaces. *Physical Review Letters*, APS, v. 98, n. 15, p. 150201, 2007.
- BROWET, A.; VLEESCHOUWER, C. D.; JACQUES, L.; MATHIAH, N.; SAYKALI, B.; MIGEOTTE, I. Cell Segmentation with Random Ferns and Graph-cuts. In: *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. Arizona: IEEE, 2016. p. 4145–4149.
- CAUCHY, A. Méthode Générale pour la Résolution des Systemes d'Équations Simultanées. *Comp. Rend. Sci. Paris*, v. 25, n. 1847, p. 536–538, 1847.
- CAVAZZA, J.; MURINO, V. Active Regression with Adaptive Huber Loss. *arXiv preprint arXiv:1606.01568*, 2016.
- CHAITANYA, C. R. A.; KAPLANYAN, A. S.; SCHIED, C.; SALVI, M.; LEFOHN, A.; NOWROUZEZHAI, D.; AILA, T. Interactive Reconstruction of Monte Carlo Image Sequences Using a Recurrent Denoising Autoencoder. *ACM Transactions on Graphics (TOG)*, ACM, v. 36, n. 4, p. 98, 2017.

- CHEN, H.; QI, X.; YU, L.; HENG, P.-A. DCAN: Deep Contour-aware Networks for Accurate Gland Segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas: IEEE, 2016. p. 2487–2496.
- CHEN, L.-C.; PAPANDREOU, G.; KOKKINOS, I.; MURPHY, K.; YUILLE, A. L. Deeplab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFS. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, v. 40, n. 4, p. 834–848, 2017.
- CHEN, L.-C.; ZHU, Y.; PAPANDREOU, G.; SCHROFF, F.; ADAM, H. Encoder-decoder with Atrous Separable Convolution for Semantic Image Segmentation. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Germany: Springer, 2018. p. 801–818.
- CHOROMANSKA, A.; HENAFF, M.; MATHIEU, M.; AROUS, G. B.; LECUN, Y. The Loss Surfaces of Multilayer Networks. In: *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*. California: PMLR, 2015. p. 192–204.
- CLOUGH, J. R.; OKSUZ, I.; BYRNE, N.; ZIMMER, V. A.; SCHNABEL, J. A.; KING, A. P. A Topological Loss Function for Deep-learning Based Image Segmentation Using Persistent Homology. *arXiv preprint arXiv:1910.01877*, 2019.
- CSURKA, G.; LARLUS, D.; PERRONNIN, F.; MEYLAN, F. What is a Good Evaluation Measure for Semantic Segmentation? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, v. 26, p. 1, 2004.
- DAUPHIN, Y. N.; PASCANU, R.; GULCEHRE, C.; CHO, K.; GANGULI, S.; BENGIO, Y. Identifying and Attacking the Saddle Point Problem in High-dimensional Non-convex Optimization. In: *Proceedings of the International Conference of Advances in Neural Information Processing Systems (NeurIPS)*. Canada: Curran Associates Inc., 2014. p. 2933–2941.
- EILERTSEN, G.; KRONANDER, J.; DENES, G.; MANTIUK, R. K.; UNGER, J. HDR Image Reconstruction from a Single Exposure Using Deep CNNs. *ACM Transactions on Graphics (TOG)*, ACM, v. 36, n. 6, p. 178, 2017.
- ESTRADA, F. J.; JEPSON, A. D. Benchmarking Image Segmentation Algorithms. *International Journal of Computer Vision*, Springer, v. 85, n. 2, p. 167–181, 2009.
- FATHI, A.; WOJNA, Z.; RATHOD, V.; WANG, P.; SONG, H. O.; GUADARRAMA, S.; MURPHY, K. P. Semantic Instance Segmentation Via Deep Metric Learning. *arXiv preprint arXiv:1703.10277*, 2017.
- FIDON, L.; LI, W.; GARCIA-PERAZA-HERRERA, L. C.; EKANAYAKE, J.; KITCHEN, N.; OURSELIN, S.; VERCAUTEREN, T. Generalised Wasserstein Dice Score for Imbalanced Multi-class Segmentation using Holistic Convolutional Networks. In: *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI) Brainlesion Workshop*. Canada: Springer, 2017. p. 64–76.
- FRIEDMAN, M. A Comparison of Alternative Tests of Significance for the Problem of M Rankings. *The Annals of Mathematical Statistics*, JSTOR, v. 11, n. 1, p. 86–92, 1940.

GANGAPURE, V. N.; BANERJEE, S.; CHOWDHURY, A. S. Steerable Local Frequency Based Multispectral Multifocus Image Fusion. *Information Fusion*, Elsevier, v. 23, p. 99–115, 2015.

GLOROT, X.; BENGIO, Y. Understanding the Difficulty of Training Deep Feedforward Neural Networks. In: *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*. Italy: PMLR, 2010. p. 249–256.

GONZALEZ, R. C.; WOODS, R. E. Image Processing. *Digital Image Processing*, v. 2, p. 1, 2007.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. United States of America: MIT Press, 2016.

HAGHIGHAT, A.; AGHAGOLZADEH, A.; SEYEDARABI, H. Real-time Fusion of Multi-focus Images for Visual Sensor Networks. In: *Proceedings of the Iranian Conference on Machine Vision and Image Processing (MVIP)*. Iran: IEEE, 2010. p. 1–6.

HAGHIGHAT, A. A. A.; SEYEDARABI, H. Multi-focus Image Fusion for Visual Sensor Networks in DCT Domain. *Computers and Electrical Engineering*, Elsevier, v. 37, n. 5, p. 789–797, 2011.

HASHEMI, S. R.; SALEHI, S. S. M.; ERDOGMUS, D.; PRABHU, S. P.; WARFIELD, S. K.; GHOLIPOUR, A. Asymmetric Loss Functions and Deep Densely-connected Networks for Highly-imbalanced Medical Image Segmentation: Application to Multiple Sclerosis Lesion Detection. *IEEE Access*, IEEE, v. 7, p. 1721–1735, 2018.

HAYKIN, S. S. *Neural Networks and Learning Machines*. New York: Prentice Hall, 2009.

HE, H.; GARCIA, E. A. Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering*, IEEE, v. 21, n. 9, p. 1263–1284, 2009.

HE, K.; GKIOXARI, G.; DOLLÁR, P.; GIRSHICK, R. Mask R-CNN. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Italy: IEEE, 2017. p. 2980–2988.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Delving Deep into Rectifiers: Surpassing Human-level Performance on ImageNet Classification. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Chile: IEEE, 2015. p. 1026–1034.

HINTON, G. E. A Practical Guide to Training Restricted Boltzmann Machines. In: *Neural Networks: Tricks of the Trade*. Berlin: Springer, 2012. p. 599–619.

HOSANG, J.; BENENSON, R.; DOLLÁR, P.; SCHIELE, B. What Makes for Effective Detection Proposals? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, v. 38, n. 4, p. 814–830, 2016.

HUDSON, M. *AI Researchers Allege that Machine Learning is Alchemy*. 2018. Available at: <<https://www.sciencemag.org/news/2018/05/ai-researchers-allege-machine-learning-alchemy>>.

ISBI. *Cell Tracking Challenge*. 2019. Available at: <<http://celltrackingchallenge.net/2d-datasets/>>.

- ISENSEE, F.; PETERSEN, J.; KLEIN, A.; ZIMMERER, D.; JAEGER, P. F.; KOHL, S.; WASSERTHAL, J.; KOEHLER, G.; NORAJITRA, T.; WIRKERT, S. nnU-Net: Self-adapting Framework for U-Net-Based Medical Image Segmentation. In: *Bildverarbeitung für die Medizin*. Wiesbaden: Springer, 2019. p. 22–22.
- ISOLA, P.; ZHU, J.-Y.; ZHOU, T.; EFROS, A. A. Image-to-image Translation with Conditional Adversarial Networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Hawaii: IEEE, 2017. p. 1125–1134.
- JÉGOU, S.; DROZDZAL, M.; VAZQUEZ, D.; ROMERO, A.; BENGIO, Y. The One Hundred Layers Tiramisu: Fully Convolutional Densenets for Semantic Segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. Hawaii: IEEE, 2017. p. 11–19.
- JOHNSON, J.; ALAHI, A.; FEI-FEI, L. Perceptual Losses for Real-time Style Transfer and Super-resolution. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Amsterdam: Springer, 2016. p. 694–711.
- KERVADEC, H.; DOLZ, J.; TANG, M.; GRANGER, E.; BOYKOV, Y.; AYED, I. B. Constrained-CNN Losses for Weakly Supervised Segmentation. *Medical Image Analysis*, Elsevier, v. 54, p. 88–99, 2019.
- KIEFER, J.; WOLFOWITZ, J. Stochastic Estimation of the Maximum of a Regression Function. *The Annals of Mathematical Statistics*, Institute of Mathematical Statistics, v. 23, n. 3, p. 462–466, 1952.
- KINGMA, D. P.; BA, J. Adam: A Method for Stochastic Optimization. In: *Proceedings of the International Conference on Learning Representations (ICLR)*. California: JMLR, 2015.
- KIRILLOV, A.; HE, K.; GIRSHICK, R.; ROTHER, C.; DOLLÁR, P. Panoptic Segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. California: IEEE, 2019. p. 9404–9413.
- KONG, W.; LEI, Y.; ZHAO, H. Adaptive Fusion Method of Visible Light and Infrared Images Based on Non-subsampled Shearlet Transform and Fast Non-negative Matrix Factorization. *Infrared Physics & Technology*, Elsevier, v. 67, p. 161–172, 2014.
- KOTOVENKO, D.; SANAKOYEU, A.; MA, P.; LANG, S.; OMMER, B. A Content Transformation Block for Image Style Transfer. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. California: IEEE, 2019. p. 10032–10041.
- LAM, L.; LEE, S.-W.; SUEN, C. Y. Thinning Methodologies-a Comprehensive Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, v. 14, n. 9, p. 869–885, 1992.
- LEE, M. C. H.; PETERSEN, K.; PAWLOWSKI, N.; GLOCKER, B.; SCHAAP, M. TETRIS: Template Transformer Networks for Image Segmentation with Shape Priors. *IEEE Transactions on Medical Imaging*, IEEE, 2019.
- LEWIS, J. J.; O'CALLAGHAN, R. J.; NIKOLOV, S. G.; BULL, D. R.; CANAGARAJAH, N. Pixel-and Region-based Image Fusion with Complex Wavelets. *Information Fusion*, Elsevier, v. 8, n. 2, p. 119–130, 2007.

- LI, H.; XU, Z.; TAYLOR, G.; STUDER, C.; GOLDSTEIN, T. Visualizing the Loss Landscape of Neural Nets. In: *Proceedings of the International Conference of Advances in Neural Information Processing Systems (NeurIPS)*. Canada: Curran Associates Inc., 2018. p. 6389–6399.
- LI, Q.; ARNAB, A.; TORR, P. H. Weakly and Semi-Supervised Panoptic Segmentation. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Germany: Springer, 2018. p. 102–118.
- LI, S.; KANG, X.; HU, J. Image Fusion with Guided Filtering. *IEEE Transactions on Image Processing*, IEEE, v. 22, n. 7, p. 2864–2875, 2013.
- LI, S.; KANG, X.; HU, J.; YANG, B. Image Matting for Fusion of Multi-focus Images in Dynamic Scenes. *Information Fusion*, Elsevier, v. 14, n. 2, p. 147–162, 2013.
- LIANG, Q.; NAN, Y.; COPPOLA, G.; ZOU, K.; SUN, W.; ZHANG, D.; YU, G. Weakly-Supervised Biomedical Image Segmentation by Reiterative Learning. *IEEE Journal of Biomedical and Health Informatics*, IEEE, v. 23, n. 3, p. 1205–1214, 2018.
- LIN, T.-Y.; GOYAL, P.; GIRSHICK, R.; HE, K.; DOLLÁR, P. Focal Loss for Dense Object Detection. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. Italy: IEEE, 2017. p. 2980–2988.
- LIN, T.-Y.; MAIRE, M.; BELONGIE, S.; HAYS, J.; PERONA, P.; RAMANAN, D.; DOLLÁR, P.; ZITNICK, C. L. Microsoft COCO: Common Objects in Context. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Switzerland: Springer, 2014. p. 740–755.
- LIU, G.; REDA, F. A.; SHIH, K. J.; WANG, T.-C.; TAO, A.; CATANZARO, B. Image Inpainting for Irregular Holes Using Partial Convolutions. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. Germany: Springer, 2018. p. 85–100.
- LIU, Y.; CHEN, X.; PENG, H.; WANG, Z. Multi-focus Image Fusion with a Deep Convolutional Neural Network. *Information Fusion*, Elsevier, v. 36, p. 191–207, 2017.
- LIU, Z.; BLASCH, E.; XUE, Z.; ZHAO, J.; LAGANIERE, R.; WU, W. Objective Assessment of Multiresolution Image Fusion Algorithms for Context Enhancement in Night Vision: A Comparative Study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, v. 34, n. 1, p. 94–109, 2012.
- LJOSA, V.; SOKOLNICKI, K. L.; CARPENTER, A. E. Annotated High-throughput Microscopy Image Sets for Validation. *Nature Methods*, NIH Public Access, v. 9, n. 7, p. 637–637, 2012.
- LONG, J.; SHELHAMER, E.; DARRELL, T. Fully Convolutional Networks for Semantic Segmentation. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston: IEEE, 2015. p. 3431–3440.
- MATTHEWS, B. W. Comparison of the Predicted and Observed Secondary Structure of T4 Phage Lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, Elsevier, v. 405, n. 2, p. 442–451, 1975.

- MAURER, C. R.; QI, R.; RAGHAVAN, V. A Linear Time Algorithm for Computing Exact Euclidean Distance Transforms of Binary Images in Arbitrary Dimensions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, IEEE, v. 25, n. 2, p. 265–270, 2003.
- MCCARTHY, J.; MINSKY, M. L.; ROCHESTER, N.; SHANNON, C. E. A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, August 31, 1955. *AI Magazine*, v. 27, n. 4, p. 12–12, 2006.
- MEYER, F. Topographic Distance and Watershed Lines. *Signal Processing*, Elsevier, v. 38, n. 1, p. 113–125, 1994.
- MILLETARI, F.; NAVAB, N.; AHMADI, S.-A. V-net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. In: *Proceedings of the International Conference on 3D Vision (3DV)*. California: IEEE, 2016. p. 565–571.
- MINAR, M. R.; NAHER, J. Recent Advances in Deep Learning: An Overview. *arXiv preprint arXiv:1807.08169*, 2018.
- NEJATI, M.; SAMAVI, S.; SHIRANI, S. Multi-focus Image Fusion Using Dictionary-based Sparse Representation. *Information Fusion*, Elsevier, v. 25, p. 72–84, 2015.
- NEMENYI, P. Distribution-free Multiple Comparisons. In: *Biometrics*. Washington DC: International Biometric Society, 1962. v. 18, n. 2, p. 263.
- ÖZDEMİR, B.; AKSOY, S.; ECKERT, S.; PESARESI, M.; EHRLICH, D. Performance Measures for Object Detection Evaluation. *Pattern Recognition Letters*, Elsevier, v. 31, n. 10, p. 1128–1137, 2010.
- PEÑA, F. A. G.; FERNANDEZ, P. D. M.; REN, T. I.; YUI, M.; ROTHENBERG, E.; CUNHA, A. Multiclass Weighted Loss for Instance Segmentation of Cluttered Cells. In: *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. Greece: IEEE, 2018. p. 2451–2455.
- PEÑA, F. A. G.; FERNÁNDEZ, P. D. M.; REN, T. I.; VASCONCELOS, G. C.; CUNHA, A. A Multiple Source Hourglass Deep Network for Multi-Focus Image Fusion. *arXiv preprint arXiv:1908.10945*, 2019.
- PEÑA, F. A. G.; FERNANDEZ, P. D. M.; REN, T. I.; CUNHA, A. A Weakly Supervised Method for Instance Segmentation of Biological Cells. In: *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI) MIL Workshop*. China: Springer, 2019. p. 216–224.
- PEÑA, F. A. G.; FERNANDEZ, P. D. M.; REN, T. I.; LEANDRO, J. J. G.; NISHIHARA, R. Burst Ranking for Blind Multi-Image Deblurring. *IEEE Transactions on Image Processing*, IEEE, v. 29, p. 947–958, 2020.
- PEÑA, F. A. G.; FERNANDEZ, P. D. M.; TARR, P. T.; REN, T. I.; MEYEROWITZ, E. M.; CUNHA, A. J Regularization Improves Imbalanced Multiclass Segmentation. *arXiv preprint arXiv:1910.09783*, 2019.
- PETROVIC, V. S.; XYDEAS, C. S. Gradient-based Multiresolution Image Fusion. *IEEE Transactions on Image Processing*, IEEE, v. 13, n. 2, p. 228–237, 2004.

REDONDO-CABRERA, C.; BAPTISTA-RÍOS, M.; LÓPEZ-SASTRE, R. J. Learning to Exploit the Prior Network Knowledge for Weakly-Supervised Semantic Segmentation. *IEEE Transactions on Image Processing*, IEEE, v. 28, n. 7, p. 3649–3661, 2019.

RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional Networks for Biomedical Image Segmentation. In: *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*. Germany: Springer, 2015. p. 234–241.

ROSENBERG, S. A.; RESTIFO, N. P. Adoptive Cell Transfer as Personalized Immunotherapy for Human Cancer. *Science*, American Association for the Advancement of Science, v. 348, n. 6230, p. 62–68, 2015.

ROTHENBERG, E. V.; MOORE, J. E.; YUI, M. A. Launching the T-cell-lineage Developmental Programme. *Nature Reviews Immunology*, Nature Publishing Group, v. 8, n. 1, p. 9, 2008.

SALEHI, S. S. M.; ERDOGMUS, D.; GHOLIPOUR, A. Tversky Loss Function for Image Segmentation Using 3D Fully Convolutional Deep Networks. In: *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI) MLMI Workshop*. Canada: Springer, 2017. p. 379–387.

SERRE, T. Deep Learning: The Good, the Bad, and the Ugly. *Annual Review of Vision Science*, Annual Reviews, v. 5, p. 399–426, 2019.

SHAN, G. Improved Confidence Intervals for the Youden Index. *PloS One*, Public Library of Science, v. 10, n. 7, p. e0127272, 2015.

SIMONYAN, K.; ZISSERMAN, A. Very Deep Convolutional Networks for Large-scale Image Recognition. *arXiv preprint arXiv:1409.1556*, 2014.

SIRINUKUNWATTANA, K.; PLUIM, J. P.; CHEN, H. Gland Segmentation in Colon Histology Images: The Glas Challenge Contest. *Medical Image Analysis*, Elsevier, v. 35, p. 489–502, 2017.

SUDRE, C. H.; LI, W.; VERCAUTEREN, T.; OURSELIN, S.; CARDOSO, M. J. Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations. In: *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI) ML-CDC Workshop*. Canada: Springer, 2017. p. 240–248.

SUN, S.; PANG, J.; SHI, J.; YI, S.; OUYANG, W. Fishnet: A Versatile Backbone for Image, Region, and Pixel Level Prediction. In: *Proceedings of the International Conference of Advances in Neural Information Processing Systems (NeurIPS)*. Canada: Curran Associates Inc., 2018. p. 754–764.

TAN, J. H.; FUJITA, H.; SIVAPRASAD, S.; BHANDARY, S. V.; RAO, A. K.; CHUA, K. C.; ACHARYA, U. R. Automated Segmentation of Exudates, Haemorrhages, Microaneurysms Using Single Convolutional Neural Network. *Information Sciences*, Elsevier, v. 420, p. 66–76, 2017.

TANG, H.; XIAO, B.; LI, W.; WANG, G. Pixel Convolutional Neural Network for Multi-focus Image Fusion. *Information Sciences*, Elsevier, v. 433, p. 125–141, 2018.

TIELEMAN, T.; HINTON, G. Lecture 6.5 - RMSProp, COURSERA: Neural Networks for Machine Learning. *Technical Report*, 2012.

WILLIS, L.; REFAHI, Y.; WIGHTMAN, R.; LANDREIN, B.; TELES, J.; HUANG, K. C.; MEYEROWITZ, E. M.; JÖNSSON, H. Cell Size and Growth Regulation in the Arabidopsis Thaliana Apical Stem Cell Niche. *Proceedings of the National Academy of Sciences*, National Acad Sciences, v. 113, n. 51, p. E8238–E8246, 2016.

WONG, K. C.; MORADI, M.; TANG, H.; SYEDA-MAHMOOD, T. 3d Segmentation with Exponential Logarithmic Loss for Highly Unbalanced Object Sizes. In: *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Spain: Springer, 2018. p. 612–619.

WU, H.; ZHANG, J.; HUANG, K.; LIANG, K.; YU, Y. FastFCN: Rethinking Dilated Convolution in the Backbone for Semantic Segmentation. *arXiv preprint arXiv:1903.11816*, 2019.

XU, Y.; LI, Y.; WANG, Y.; LIU, M.; FAN, Y.; LAI, M.; ERIC, I.; CHANG, C. Gland Instance Segmentation Using Deep Multichannel Neural Networks. *IEEE Transactions on Biomedical Engineering*, IEEE, v. 64, n. 12, p. 2901–2912, 2017.

YAN, X.; GILANI, S. Z.; QIN, H.; MIAN, A. Unsupervised Deep Multi-focus Image Fusion. *arXiv preprint arXiv:1806.07272*, 2018.

YANG, B.; LI, S. Multifocus Image Fusion and Restoration with Sparse Representation. *IEEE Transactions on Instrumentation and Measurement*, IEEE, v. 59, n. 4, p. 884–892, 2010.

YANG, L.; ZHANG, Y.; CHEN, J.; ZHANG, S.; CHEN, D. Z. Suggestive Annotation: A Deep Active Learning Framework for Biomedical Image Segmentation. In: *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*. Canada: Springer, 2017. p. 399–407.

YANG, S.; KWEON, J.; KIM, Y.-H. Major Vessel Segmentation on X-ray Coronary Angiography using Deep Networks with a Novel Penalty Loss Function. In: *International Conference on Medical Imaging with Deep Learning (MIDL)*. England: PMLR, 2019. v. 102.

YOUDEM, W. J. Index for Rating Diagnostic Tests. *Cancer*, Wiley Online Library, v. 3, n. 1, p. 32–35, 1950.

YUE, Z.; YONG, H.; ZHAO, Q.; ZHANG, L.; MENG, D. Variational Denoising Network: Toward Blind Noise Modeling and Removal. *arXiv preprint arXiv:1908.11314*, 2019.

ZAGORUYKO, S.; KOMODAKIS, N. Learning to Compare Image Patches Via Convolutional Neural Networks. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Boston: IEEE, 2015. p. 4353–4361.

ZHANG, C.; YARKONY, J.; HAMPRECHT, F. A. Cell Detection and Segmentation Using Correlation Clustering. In: *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*. Boston: Springer, 2014. p. 9–16.

-
- ZHANG, Q.; GUO, B.-I. Multifocus Image Fusion Using the Nonsubsampled Contourlet Transform. *Signal Processing*, Elsevier, v. 89, n. 7, p. 1334–1346, 2009.
- ZHAO, H.; GALLO, O.; FROSIO, I.; KAUTZ, J. Loss Functions for Image Restoration with Neural Networks. *IEEE Transactions on Computational Imaging*, IEEE, v. 3, n. 1, p. 47–57, 2016.
- ZHAO, X.; LIAO, Y.; LI, Y.; ZHANG, T.; ZOU, X. FC2N: Fully Channel-Concatenated Network for Single Image Super-Resolution. *arXiv preprint arXiv:1907.03221*, 2019.
- ZHOU, X.; YAO, C.; WEN, H.; WANG, Y.; ZHOU, S.; HE, W.; LIANG, J. EAST: An Efficient and Accurate Scene Text Detector. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Hawaii: IEEE, 2017. p. 5551–5560.
- ZHOU, X.-Y.; SHEN, M.; RIGA, C.; YANG, G.-Z.; LEE, S.-L. Focal FCN: Towards Biomedical Small Object Segmentation with Limited Training Data. *arXiv preprint arXiv:1711.01506*, 2017.
- ZHU, W.; HUANG, Y.; ZENG, L.; CHEN, X.; LIU, Y.; QIAN, Z.; DU, N.; FAN, W.; XIE, X. AnatomyNet: Deep Learning for Fast and Fully Automated Whole-volume Segmentation of Head and Neck Anatomy. *Medical Physics*, Wiley Online Library, v. 46, n. 2, p. 576–589, 2019.