

Efficient and accurate cell image segmentation through optimizer and loss function fine-tuning with quantization and pruning

Xhoena Polisi^{1,*}, Arban Uka², Daniel Avdiu², Dimitrios A Karras²

¹Department of Informatics, "Fan S. Noli" University, Korça, Albania; xpolisi@epoka.edu.al (X.P.)

²Department of Computer Engineering, EPOKA University, Tirana, Albania; auka@epoka.edu.al (A.U.)

davdiu20@epoka.edu.al (D.A.) dkarras@epoka.edu.al (D.A.K.).

Abstract: Convolutional deep learning is commonly and frequently used nowadays for high data throughput in image analysis and computer vision applications. The size of such models depends on the number of hyperparameters and the precision required for each hyperparameter. The training of these models requires enormous computational power for images of high dimensionality, resulting in trained models of large sizes. To solve this challenge, several approaches have been used, including quantization and pruning. These techniques have been proven to be effective in reducing the size of the models at the expense of lower accuracy, while in order to ensure higher performance, several hyperparameters need to be optimized. This research effort attempts to evaluate the effect of different combinations of classical and hybrid optimizer-loss functions with the U-Net convolutional deep learning model regarding their segmentation accuracy in cell imaging, while implementing both quantization and pruning. Different metrics are used to evaluate the training and testing process, such as precision, recall, F1 score, dice coefficients, Jaccard index, etc. It is found that the hybrid loss function combination of binary cross-entropy with Jaccard loss functions and RMSprop as the best optimizer results in the highest segmentation accuracy. These combinations of optimizers and loss functions are implemented with post-training quantization and quantization-aware training, and in both cases, the accuracy metrics were not compromised significantly, as they resulted in a decrease of less than 1%. Pruning, as a process to reduce the computational complexity and the size of the model, has proven to be effective by also providing the highest performance among other methods.

Keywords: Hybrid loss functions, Optimizer, Pruning, Quantization, Segmentation, U-net.

1. Introduction

Deep learning utilizes artificial neural networks with several layers to extract complex patterns from data [1] with the most common use including the processing and the analysis of images [2]. Image analysis includes three important steps as follows: i) object detection meaning clearly identifying the region of interest where the object is located [3] ii) object segmentation meaning accurately drawing a closed contour around an object of interest thus separating it from the background and iii) object classification in order to identify the category where the object belongs [4, 5]. A high accuracy in each of these steps leads to a better quantitative image analysis and performance [6]. In biomedical image analysis, an accurate cell image segmentation plays a vital role as proper cell image features extraction from the segmentation step leads to accurate quantitative data that can be used in various applications, where the most important one is apriori disease diagnosis [7]. Other applications such as drug delivery systems and analysis of cell behaviour with/without a biomaterial present, are also important as there are vast applications nowadays compared to decades earlier [8]. Cells have two major components, the nucleus and the cytoplasm, where the latter is deformable and more transparent

to illumination. The noise spectrum adds another challenge to the high dimensionality of the data and in order to overcome this suitable pre-processing is used. Pre-processing includes noise removal, edge enhancement, contrast enhancement and all these have proven to affect the overall accuracy of the trained models [8, 9]. Optimal bandwidth of filters was reported to lead to highest classification accuracies of Balb 3T3 cell types.

Numerous studies have been conducted in this field, and many deep learning architectures have been developed for this task, where U-Net architecture has had a large number of applications due to its simpler architecture, high performance and efficiency [7-11]. Its encoder-decoder structure, together with the implementation of skip connections, provides an efficient spatial and contextual information in an image, making it a preferred choice especially for image segmentation tasks and its implementation being added in various applications such as ImageJ etc. Even though there is a lot of research, optimization of the best parameters for the U-Net architecture for specific challenging tasks and datasets, it is still open research.

U-Net architecture does not require high computational power compared to more complex other architectures [12]. However, alterations such as quantization and pruning are being implemented and suggested in order to be more efficient in devices that offer limited memory and speed [13, 14]. Quantization is a process that reduces the model weights, precision values and activations functions, leading to less memory usage and faster computations processing. Pruning is a process that removes less critical parameters to reduce the model size. Tuning to define the optimal parameters for these variant architectures is a challenge as there is a need to balance efficiency and accuracy, since these tuned models can also degrade the performance compared to the original U-Net.

During the training process, the selection of the optimizer and loss function is very important as they directly influence the performance of the architecture, affecting also the segmentation results provided by the model. There is related research [15-18] done in finding the optimal optimizer and loss function in U-Net architecture. Their focus is more on MRI, CT scan, Ultrasound images. However, the number of research efforts is very limited for microscopy images, especially the brightfield microscope, which produces low quality images compared to others. Several studies focus on optimizing specific parameters in their tasks. Some find the best optimizer, whereas other studies find the best loss function. The effectiveness of different activation functions has been compared in the literature and often the highest performing one has been associated with higher computational complexity [19]. The choice of the optimizer proved to greatly affect the classification accuracy and the optimizers would compare differently across different architectures [20]. They measured the performance based on AUC values. Previous studies have used evolutionary optimisation of loss functions using genetic programming [21]. They found a new loss function that had a faster convergence and higher accuracy than cross entropy loss function in data classification.

Here, we advance the work presented in Polisi, et al. [22] by analysing different combinations of optimizers and loss functions with U-Net, quantized U-Net and U-Net with pruning for unstained cell image segmentation. The contribution of this work is the evaluation of the best combination of optimizers and loss functions when used with different architectures. This is done while monitoring for the size of the trained model using quantization and pruning. The base function for the optimizer will be Adam, and Binary Cross Entropy (BCE) will be the base for the loss function. The list of optimizers to be tested is the following: Adagrad, Nadam, RMSprop and for the loss functions the hybrid functions of Dice-BCE, dice-focal loss function, and Jaccard-BCE. The metrics used for evaluation are: the DICE coefficient, Intersection Over Union (IoU), Precision, Recall for the training and the validation phase. Whereas, in the testing phase the metrics used are Accuracy, F1 score, Jaccard, Precision and Recall. The findings in this work help to provide an optimal combination that provides high efficiency that can be later used in other cell image segmentation applications.

2. Materials

The dataset consists of a total number of 1088 cell images with their corresponding ground truth ones. The size per image is 256x256 (Figure 1). The dataset is divided into training, validation and testing: 654, 217, 217 images respectively.

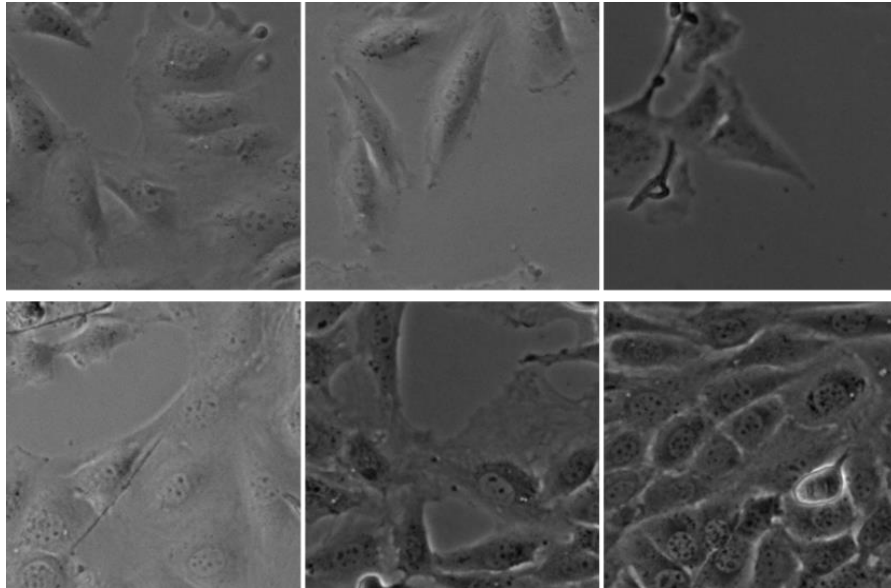


Figure 1.
Sample images of the dataset

3. Methods

The focus on this work is finding the best combination of architectures, optimizers and loss functions, by also using quantization and pruning to the U-Net architecture. Training and testing are done using GPU GeForce 2060 Super Graphics Card with 8 GB memory.

3.1. U-Net Architecture

U-Net architecture involves a convolutional neural network designed for image analysis, but gained its popularity from its high performance in image segmentation. The U shape, characteristic of this model, originates from its encoder-decoder structure with skip connections. The encoder down-samples the input image in every layer, aiming to find crucial information in the form of features, while the decoder up-samples these features to go back to spatial information. The skip connections link the corresponding encoder-decoder layers, by preserving the essential details and features. In this study, a basic U-Net architecture with 5 layers is implemented, using ReLU activations functions and batch normalization.

3.2. Quantized U-Net Architecture

Quantized U-Net is a modified architecture of base U-Net that uses the quantization process in order to reduce the numerical precision of the weights from 32-bit floating-point numbers to 8-bit integers. This process decreases the memory and computational power drastically, and reduces the model size significantly. This may result in less accurate but still acceptable cell segmentation, having applications in low cost, portable devices that produce results in real time. In this work two different quantization procedures are used: a) during training, referred to as DuringTQ and b) after the training, referred to as PostQ. Initially the batch size for quantization was 2, and then changed to six to lower the

amount of time needed for training (see Table 1). Due to the computational environment limitations, the batch size for the DuringTQ quantization is six compared to eight for all the other models.

Table 1.

Training and Testing time for the During TQ quantization when changing the batch size.

Batch Size	Time per Epoch (sec)	Total Time	Nr Images	Test Time (sec)	Nr Images Testing	Model Size (MB)
2	~120	~3h 35min	654 Training 217 Validation	46	217	364
6	~73	~2h 10min	654 Training 217 Validation	44	217	364

3.3. U-Net with Pruning

U-Net with pruning is a modified architecture of the base U-Net that uses the pruning process to remove the weights or connections that are not too important. This compression technique reduces the memory usage and the computations, resulting in a smaller and less complex architecture. It is again a useful technique in devices used in real time applications with limited small memory and limited computational power. In this work the pruning starts from 50% to 90% and is updated every 100 steps.

3.4. Set Of Optimizers and Loss Functions

This study aims to identify the best optimizer-loss function combination while taking as the base combination for U-Net architecture the BCE-Adam. Different combinations have been used herein that can be summarized as follows: i) set of optimizers such as Adagrad, Nadam, RMSprop and ii) set of loss functions: the hybrid functions of Dice- BCE, Dice-Focal and Jaccard-BCE.

Adam (Adaptive Moment Estimation) [23] uses adaptive learning rates and momentum-based gradient updates in order to assign the weights per each parameter. It keeps track of exponentially weighted moving averages of both gradients which allows for a fast convergence and more stable optimization. Due to its efficiency and robustness, it is a very popular choice as optimizer for various tasks and architectures. It was found to be the best optimizer when combined with VGG16 architecture for image classification [16].

Adagrad (Adaptive gradient descent optimizer) [24] uses the gradient information to dynamically update the learning rates of the parameters. If a parameter is not changed very often it is assigned a higher learning rate compared to the parameters who change frequently. The accumulation of squared gradients, however, can produce low learning rate over time, affecting the training process due to needing more time to converge.

Nadam (Nesterov-Adam) [25] is a modified version of Adam optimizer by adding Nesterov momentum to it. It combines the adaptive learning rates and momentum-based updates, predicting the future gradient direction which results in a higher convergence and accuracy. It is especially useful on the segmentation tasks where the information varies or it is very noisy.

RMSprop (Root Mean Square Propagation) [26] tries to reduce the learning rate limitations of Adagrad optimizer through gradient normalization. The normalization is done by removing the average value from the squared gradient values, thus making sure to have a more effective training. It is very effective in segmentation of non-stationary objects, making it a good choice for cell image analysis.

BCE (Binary Cross-Entropy) is a loss function that focuses on binary pixel classification problems where every pixel is either the background or it belongs to the object of interest. It calculates the cross-entropy of the predicted probabilities and the ground truth, ensuring an effective and accurate classification of the pixels. However, this may not be true on datasets that are not balanced, hence, it is often suggested to be combined with other loss functions.

Dice-BCE is a hybrid loss function combining Dice coefficient loss with binary cross-entropy. Dice coefficient loss focuses on the overlapping between the predicted segmentation and the ground truth

label, whereas BCE focuses more on the classification of the pixels. This hybrid combination provides a balance between the alignment of the segmentation via Dice with the local pixel accurate classification via BCE. It is usually effective on images where the object of interest is small compared to the background (such as cells for example).

Dice-Focal is a hybrid loss function combining Dice coefficient loss with Focal loss. Focal loss assigns higher weights to pixels that are hard to classify. This hybrid combination tries to have an accurate boundary segmentation via Dice with a focus on the right classification of the challenging regions, such as low contrast cytoplasm or overlapping cells.

Jaccard-BCE is a hybrid loss function that combines the Jaccard index loss with BCE loss. Jaccard index loss reduces the union of false positives and false negatives, hence producing a more accurate segmentation. This hybrid combination tries to balance the accuracy of the pixel level classifications with the overall segmentation results, making it an effective method for challenging tasks such as brightfield image segmentation.

This work focuses more on identifying the best combination among different optimizers and loss functions, by taking as the base scenario the combination between Adam optimizer and BCE loss function. Then, a combination between the other functions is provided by having each of them as a static parameter (1:M relationship instead of M:M). The different combinations are used for all three architectural models such as U-Net, quantized U-Net (during and post training) and U-Net with pruning.

3.5. Metrics Used for Evaluation

Different metrics are used to evaluate the combination of architectures with optimizers and loss functions. The training and validation process is evaluated through the Dice coefficient, Intersection Over Union (IoU), precision and recall in order to measure how accurate the predicted segmentation is compared to the ground truth. Dice coefficients find the intersection ratio between prediction and the mask whereas IoU considers both the intersection and union. Precision and recall are able to determine how good the model is in correctly identifying the region of interest (cells), by also minimizing the false negatives, giving a proper balance between sensitivity and specificity. The testing process is evaluated by accuracy, F1 score and Jaccard index. Accuracy provides the percentage of pixels that are correctly classified in the entire image. F1 score is the harmonic mean of precision and recall, generalizing them into a single metric. Jaccard index, which is similar to IoU, gives the evaluation of the predicted segmentation with the ground truth. Table 2 summarizes all these parameters.

Table 2.
Parameters used in this experiment.

Dataset	<ul style="list-style-type: none"> • Total nr. Images I /Labels L 1088 images I • Image size: 256x256 • Training dataset: 654 I+L • Validation dataset: 217 I+L • Testing dataset: 217 I+L
Architecture	<ul style="list-style-type: none"> • U-Net • Base • DuringTQ • PostQ • Pruned
Parameters of the architectures	<ul style="list-style-type: none"> • Batch size: 2, 6, 8 • Epoch nr.: 100
Optimizers	<ul style="list-style-type: none"> • Adam • Adagrad • Nadam • RMSprop
Loss functions	<ul style="list-style-type: none"> • Binary Cross entropy BCE • Dice-BCE • Dice-Focal • Jaccard-BCE
Combinations	<ul style="list-style-type: none"> • Adam-BCE (base model) • Adam- Dice BCE • Adam- Dice Focal • Adam – Jaccard BCE • Adagrad – BCE • Nadam – BCE • RMSprop- BCE
Metrics used for evaluation Training and validation: <ul style="list-style-type: none"> • Dice coefficient • Intersection over Union IoU • Precision PREC • Recall REC Testing: <ul style="list-style-type: none"> • Accuracy • F1 score • Jaccard index (IoU) • PREC • REC 	$Dice\ coefficient = \frac{2 * A \cap B }{ A + B }$ $IoU = \frac{ A \cap B }{ A \cup B }$ $PREC = \frac{TP}{TP + FP}$ $REC = \frac{TP}{TP + FN}$ $Accuracy = \frac{TP + TN}{TP + TN + FN + FN}$ $F1\ score = 2 * \frac{PREC * REC}{PREC + REC}$

4. Results

Previous work studied the combination of Dice and BCE loss function on medical imaging segmentation and concluded that the combination of the two loss functions provides a better performance for the model, as they tested them using hidden information and changing the values of the pixels [27].

The base U-Net results are reported on Figure 2 where no quantization or pruning is implemented. When using different accuracy metrics, the order from the highest performing metric to the lowest one is as follows: precision leads to highest segmentation performance, while dice coefficient, recall and IOU lead to the lowest segmentation results. The higher precision points also a statistical effect as this is

associated with a large number of true positives, and in image segmentation tasks like in this work this means that there is a large area covered by the cells.

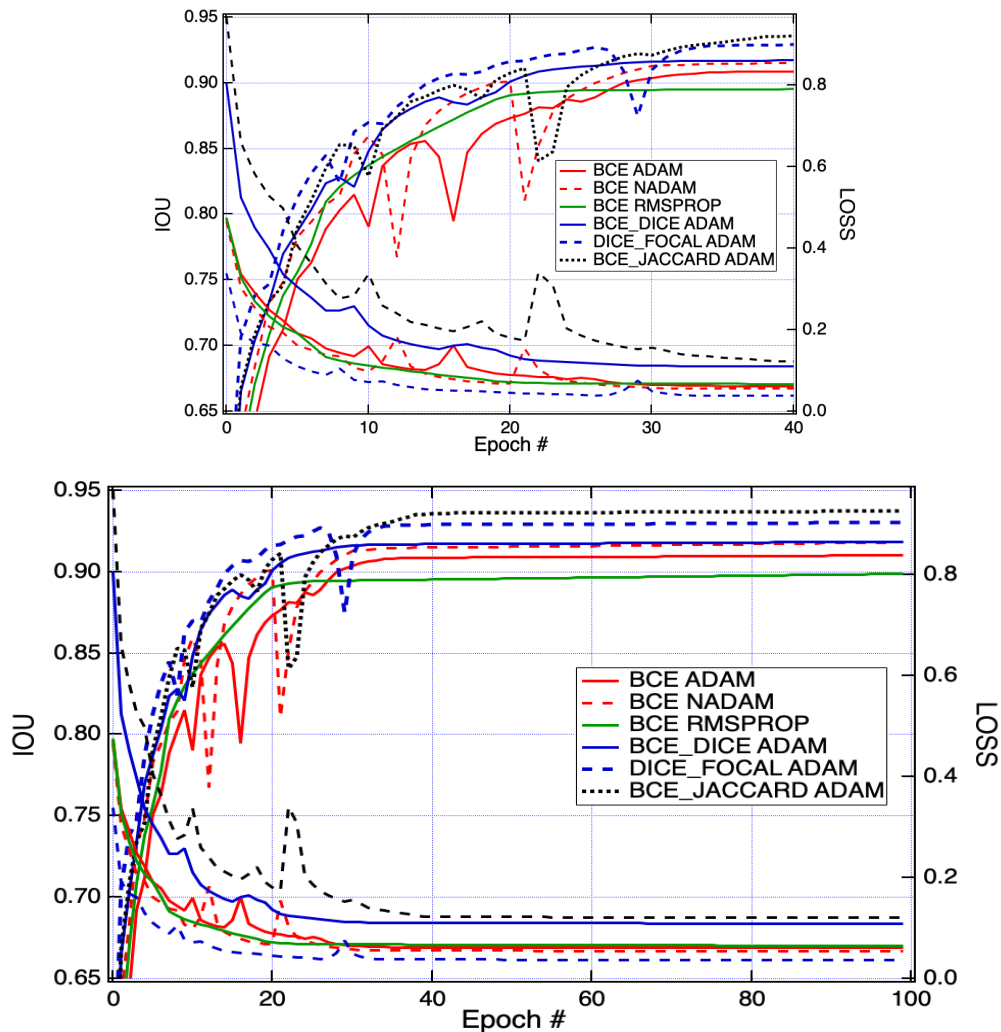


Figure 2. IoU values for different 'optimizer-loss function' combinations for Base U-Net model: a) the first 40 epochs, b) the whole training.

When comparing the loss functions, higher accuracy metrics are observed for two combinations of hybrid loss function used with Adam optimizer (see Figure 3). These two weighted loss functions are the combination of the dice loss with the focal loss (Dice-Focal) and that of the binary cross entropy loss with Jaccard loss function (BCE-Jaccard). Dice focal loss function combined with Adam optimizer has the fastest convergence and also reports the lowest loss function. Adagrad is found to be the worst performing optimizer. Previous work showed similar results where Adagrad was found to perform worse than RMSprop and Adam on weather datasets [28].

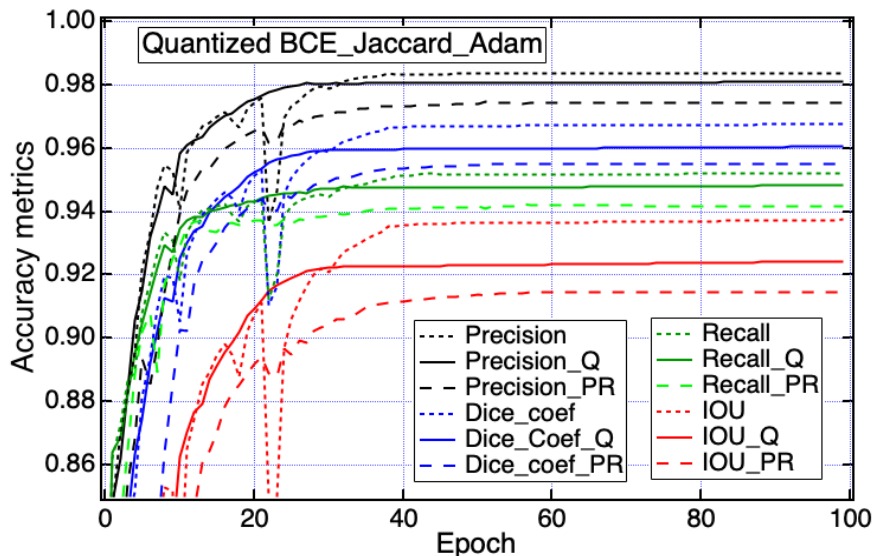


Figure 3.
Training metrics for BCE-JACCARD combination when using Base, PostQ and Pruned U-Nets.

The base combination of the optimizer-loss function fusion is Adam-Binary Cross Entropy. The aim is to find the best combination that provides the higher performance in the segmentation of the selected dataset. Figure 4 shows the visualization of the evaluation metrics of the training process for Base U-Net, PostQ and Pruned. The highest values are reported when RMSprop and ADAM are used as the optimizer. RMSprop outperforms the other optimizers in every model Base, PostQ and Pruned U-Net, having a very slight only difference with the Adam Optimizer. The highest accuracy metrics are observed when it is combined with Pruned U-Net. Previous studies have reported a superior performance of RMSprop in similar settings [29].

The loss functions that are taken in consideration are hybrid functions that combine the most common loss functions from the literature. Figure 5 shows their training performance on different metrics where Dice-Focal and Jaccard-BCE outperform the other loss functions, especially when combined with Pruned U-Net.

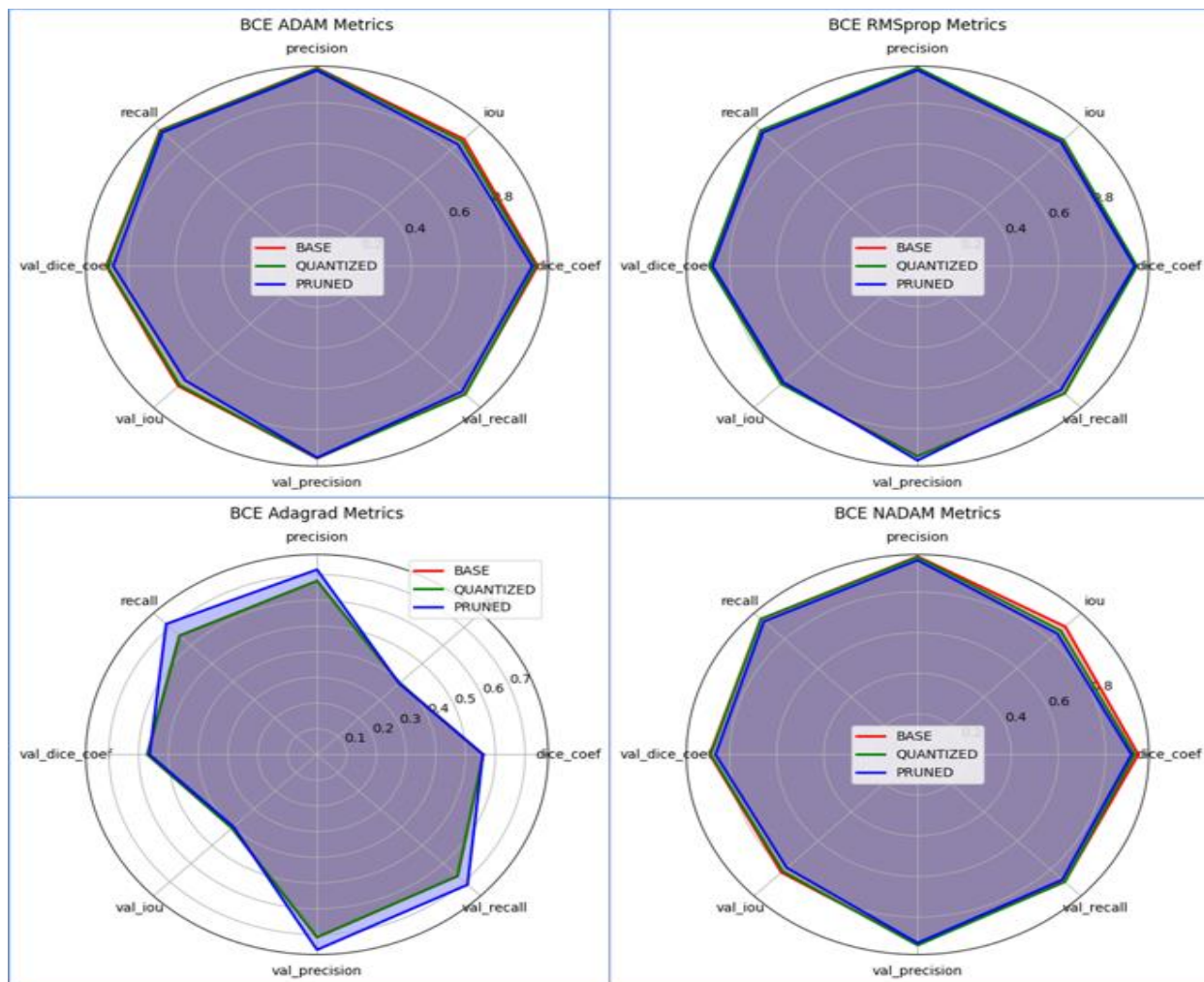


Figure 4. Different optimizers for Base, PostQ and Pruned U-Net representations of the training evaluation metrics.

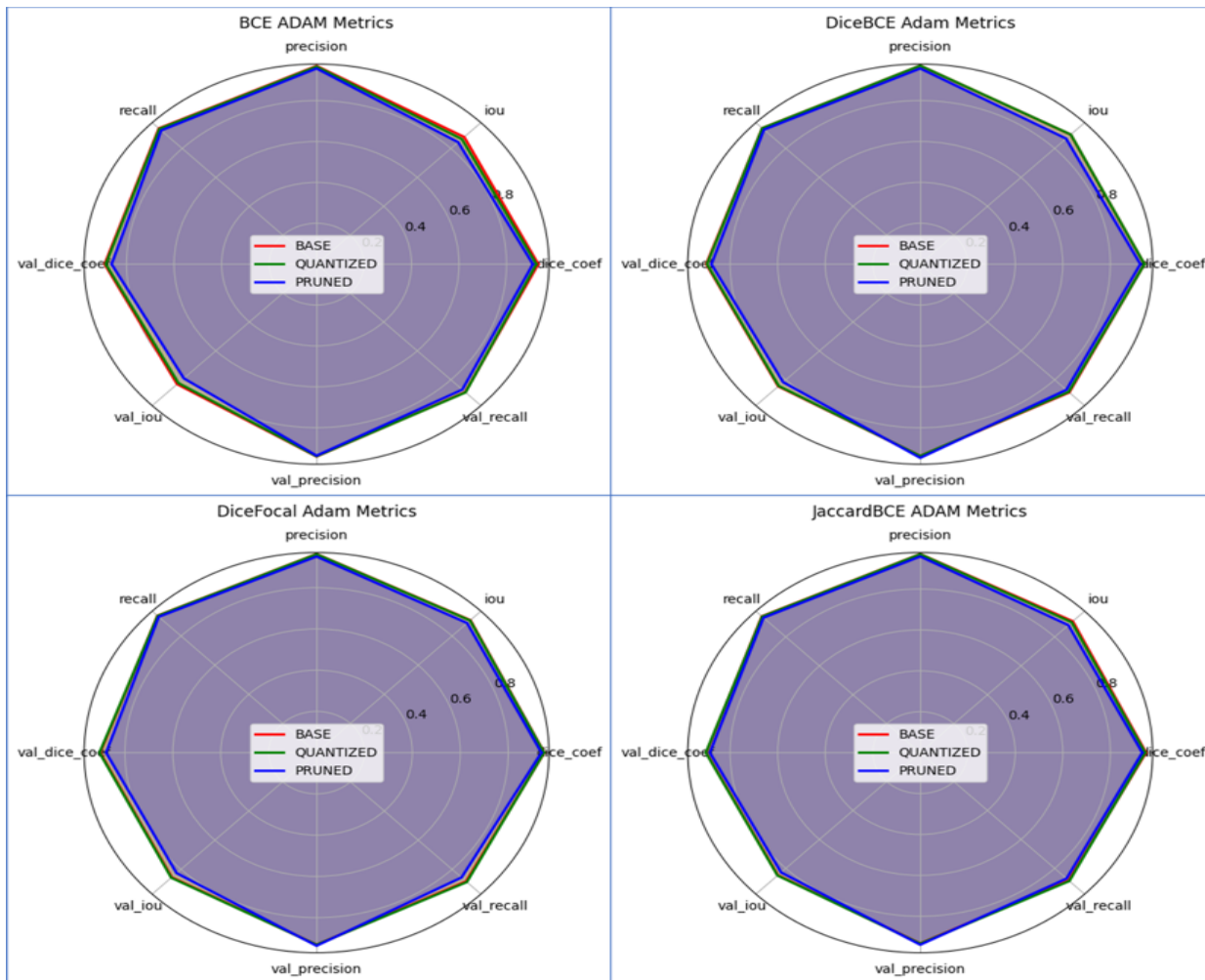


Figure 5. Different loss functions for Base, PostQ and Pruned U-Net representations of the training evaluation metrics.

The convergence to highest values is observed over the first 40 epochs of the training process and is associated with a learning rate of 10^{-4} and after the 40th epoch the learning rate drops down to 10^{-7} . Learning rate (LR) is an important parameter that leads to controlled convergence to local and global minima and needs control depending on the tasks. Different control mechanisms have been developed for the learning rate including exponential increase after each epoch [30]. LR can have two major regimes of low and high LR that are separated by a phase transition [31]. Depending on the architecture used, LR does have a narrow and stable regime. In order to ensure a gradual convergence its decay is often employed. This contribution can be substituted intuitively by increasing the batch size during training as reported in several studies [20, 32]. The same effect was observed in this work as a higher accuracy in the testing phase and a faster training time was observed when the batch size was increased from two to six (See Figure 6 below). Precision was 97.32% when trained with batch size of six and 96.43% for a batch size of two. When using the same loss function, RMSprop performs as the top optimizer with high convergence rate and high accuracy.

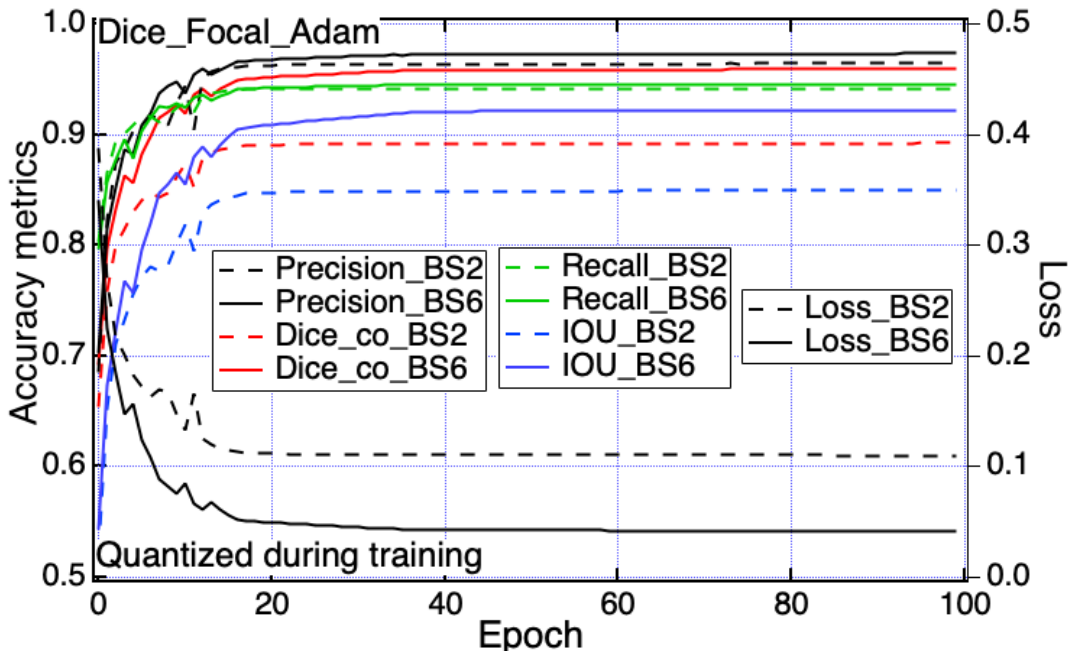


Figure 6. Quantized during the training model for batch size 2 and 6.

5. Discussion

The trained models are used to test another blind dataset, selected randomly for every combination, and left purposely out of the training and validation process in order to evaluate their performance. Table 3 summarizes the results of all combinations for Base, during training/Post quantization and Pruned U-Net models by showing the average of each metric of the testing phase. The metrics used are accuracy, F1 score, Jaccard, Recall and Precision. Figure 7 provides the grid heatmap of the metrics of the testing process and it is notable that Jaccard has the lowest values compared to other metrics while the combination BCE-Adagrad provides the lowest results compared to others models and combinations. Pruned U-Net outperforms the other architectures by having the highest F1 score, Jaccard and Accuracy (~0.95, 0.90 and 0.95 respectively). Base U-Net and DuringTQ U-Net show similar results, and PostQ U-Net sometimes has a slight drop on the values of recall, but also exhibiting an improved precision in several scenarios.

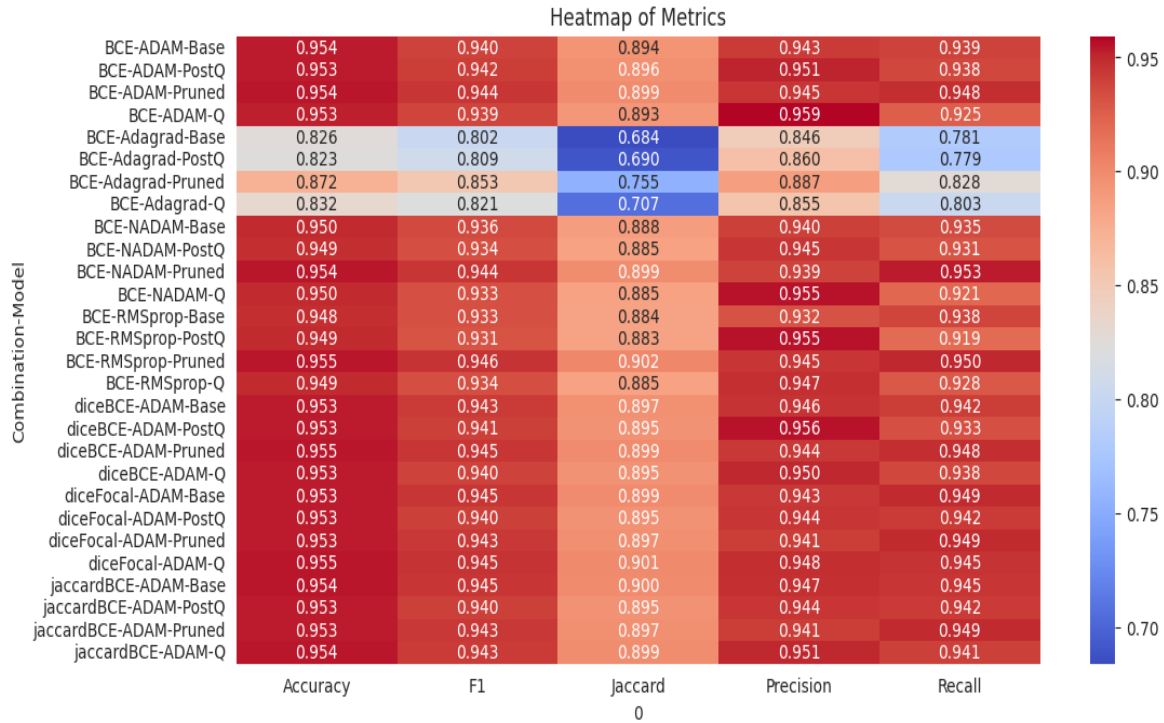


Figure 7.
Grid heatmap of Metrics of the testing phase for all combinations.

By analysing the testing score metrics, it is obvious that Pruned U-Net outperforms the other models consistently. During the training process, RMSprop optimizer was noted to perform best compared to the other optimizers, however, the higher performance was not reconfirmed during testing. During the testing phase Adam has slightly higher accuracy (~ 0.94 – 0.95 F1 score values and 0.95 accuracy) compared to RMSprop (~ 0.94 – 0.95 accuracy). This small difference may also come from the randomly selected images left out for testing. Adagrad has the lowest performance, with accuracy values dropped to ~ 0.82 – 0.87 and F1 score values among the range of $[0.80, 0.85]$.

Table 3.
Evaluation metrics for the testing phase of all different combinations.

Combination	Model	Accuracy	F1	Jaccard	Recall	Precision
BCE-ADAM	Base	0.9535	0.9395	0.8941	0.9394	0.9427
	PostQ	0.9531	0.9415	0.8962	0.9376	0.9508
	DuringTQ	0.9527	0.9387	0.8925	0.9253	0.9594
	Pruned	0.9542	0.9444	0.8992	0.9480	0.9446
BCE-NADAM	Base	0.9496	0.9359	0.8877	0.9348	0.9403
	PostQ	0.9485	0.9340	0.8850	0.9309	0.9450
	DuringTQ	0.9496	0.9331	0.8849	0.9215	0.9553
	Pruned	0.9541	0.9443	0.8991	0.9530	0.9393
BCE-RMSprop	Base	0.9481	0.9334	0.8835	0.9384	0.9318
	PostQ	0.9491	0.9313	0.8831	0.9192	0.9553
	DuringTQ	0.9493	0.9338	0.8849	0.9284	0.9475
	Pruned	0.954	0.9464	0.9019	0.9504	0.9445
BCE-Adagrad	Base	0.8259	0.8019	0.6842	0.7810	0.8459
	PostQ	0.8229	0.8085	0.6901	0.7786	0.8600
	DuringTQ	0.8315	0.8212	0.7067	0.8032	0.8551
	Pruned	0.8724	0.8528	0.7555	0.8277	0.8866
DiceBCE-ADAM	Base	0.9534	0.9427	0.8969	0.9416	0.9460
	PostQ	0.9531	0.9407	0.8952	0.9326	0.9557
	DuringTQ	0.9532	0.9399	0.8949	0.9376	0.9498
	Pruned	0.9546	0.9446	0.8994	0.9477	0.9439
DiceFocal-ADAM	Base	0.9533	0.9448	0.8992	0.9489	0.9434
	PostQ	0.9531	0.9404	0.8949	0.9422	0.9445
	DuringTQ	0.9548	0.9454	0.9013	0.9455	0.9483
	Pruned	0.9534	0.9434	0.8975	0.9494	0.9411
JaccardBCE-ADAM	Base	0.9541	0.9449	0.8997	0.9452	0.9471
	PostQ	0.9531	0.9404	0.8949	0.9422	0.9445
	DuringTQ	0.9544	0.9431	0.8987	0.9409	0.9506
	Pruned	0.9535	0.9434	0.8975	0.9494	0.9411
Jaccard BCE RMSprop	Base	0.947798	0.957535	0.897256	0.957799	0.956771
	PostQ	0.954473	0.944771	0.899828	0.940101	0.967221
	DuringTQ	0.951916	0.943059	0.896906	0.938187	0.951274
	Pruned	0.9561	0.9561	0.9314	0.9602	0.9432

Related to different loss functions tested in this work, BCE and Dice-BCE have very similar results with one another, where again Pruned U-Net provides the highest values even for the different loss function combinations. Dice-Focal hybrid loss function slightly improves Jaccard and F1 score values, and Jaccard-BCE have better Jaccard and recall values, indicating that these two loss functions provide better segmentation of the images in this type of datasets. The combination of Jaccard-BCE with RMSprop-Pruned U-Net has the highest metrics of all combinations, especially on AUC, Jaccard index

with a difference of 0.06 from the average values of the previous combinations, and REC. This again is shown on previous work, where this combination had the highest REC value [29].

The impact of the optimizer and loss function can be observed better when evaluating the cell segmentation of the images with their ground truth. Figure 8 and Figure 9 show the original image, the ground truth and cell segmentation of BCE-RMSprop and Jaccard-BCE with Adam combined with all different proposed models. It can be observed that compared to the first combination, the second one provides a better segmentation for these images, making the combination of the hybrid loss function Jaccard-BCE with ADAM a good choice for segmentation parameters. This combination can be later used for cell segmentation and cell counting, described in the previous work [33] which provide meaningful information related to cell shape.

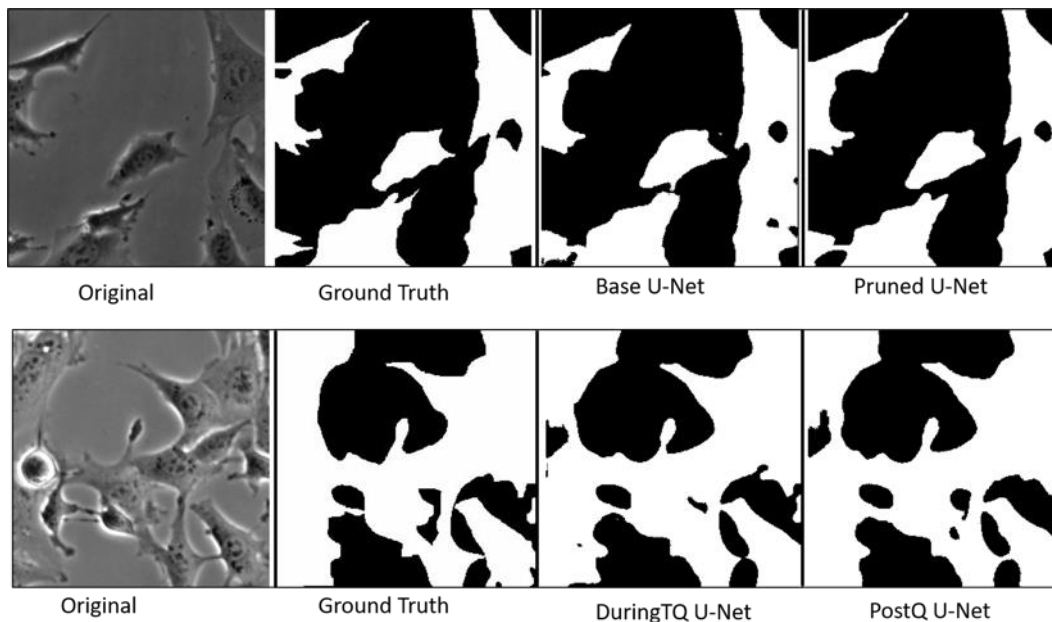


Figure 8. Segmentation of cell images when BCE-RMSprop is combined with different models.

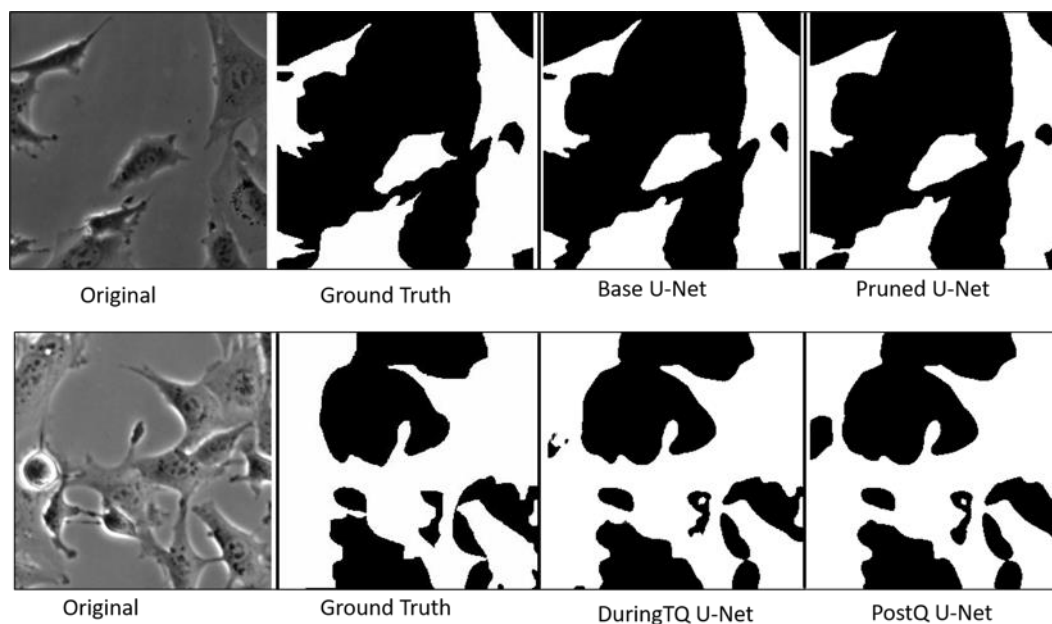


Figure 9. Segmentation of cell images when Jaccard-BCE-Adam is combined with different models.

6. Conclusion

Many deep learning trained models are required to be implemented on edge devices and microcontrollers that have limited memory and computational power. This need has led to the development of TinyML, an area that has gained momentum in all fields including medicine, agriculture, smart cities etc. Quantization and pruning are two of the most fundamental techniques allowing the successful deployment of trained models on portable computing units. In this work, the efficiency of these two techniques was tested for the segmentation of highly dimensional unstained brightfield cell images. The use of optimization techniques is associated with a decrease in the accuracy and to overcome this, several combinations of optimizers, loss functions and hybrid loss functions were tested. Binary cross entropy was reconfirmed to be a good loss function and similarly the hybrid loss function combining binary cross entropy and the jaccard loss performed very well too. A higher batch size of six when compared to a batch size of two reduced the training time and improved the accuracy by 1%. Adagrad was found to be the least performing optimizer for this dataset during the training and testing process, with RMSprop and ADAM performing the best. Pruned U-Net provides the highest performance during the testing phase, having accuracy value of 95%. The hybrid combination of Dice-Focal and Jaccard-Binary Cross Entropy loss functions provide higher performance compared to other combinations, making them more suitable for cell image segmentation of similar data. The findings of this work highlight the importance of choosing the right optimization and loss function techniques, paired with the appropriate model compression method to ensure accurate segmentation in cell imaging.

Funding:

This project received funding from the European Union's Horizon 2020 Research and Innovation Program under (Grant Number 760921) (PANBioRA).

Transparency:

The authors confirm that the manuscript is an honest, accurate, and transparent account of the study; that no vital features of the study have been omitted; and that any discrepancies from the study as planned have been explained. This study followed all ethical practices during writing.

Copyright:

© 2025 by the authors. This open-access article is distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

References:

- [1] S. A. Bini, "Artificial intelligence, machine learning, deep learning, and cognitive computing: What do these terms mean and how will they impact health care?," *The Journal of Arthroplasty*, vol. 33, no. 8, pp. 2358–2361, 2018. <https://doi.org/10.1016/j.arth.2018.03.065>
- [2] F. Xing, Y. Xie, H. Su, F. Liu, and L. Yang, "Deep learning in microscopy image analysis: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 10, pp. 4550–4568, 2017. <https://doi.org/10.1109/TNNLS.2017.2712514>
- [3] A. Uka, A. Tare, X. Polisi, and I. Panci, "FASTER R-CNN for cell counting in low contrast microscopic images," presented at the 2020 International Conference on Computing, Networking, Telecommunications & Engineering Sciences Applications (CoNTESA), IEEE, 2020.
- [4] X. Chen *et al.*, "Recent advances and clinical applications of deep learning in medical image analysis," *Medical Image Analysis*, vol. 79, p. 102444, 2022. <https://doi.org/10.1016/j.media.2022.102444>
- [5] X. Polisi, A. N. Halili, A. Uka, and C. Ciulla, "Two-stage unsupervised classification of cell health," presented at the 2023 International Conference on Computing, Electronics & Communications Engineering (iCCECE), IEEE, 2023.
- [6] S. Suganyadevi, V. Seethalakshmi, and K. Balasamy, "A review on deep learning in medical image analysis," *International Journal of Multimedia Information Retrieval*, vol. 11, no. 1, pp. 19–38, 2022.
- [7] C. Edlund *et al.*, "LIVECell—A large-scale dataset for label-free live cell segmentation," *Nature methods*, vol. 18, no. 9, pp. 1038–1045, 2021.
- [8] A. Uka, X. Polisi, J. Barthes, A. N. Halili, F. Skuka, and N. E. Vrana, "Effect of preprocessing on performance of neural networks for microscopy image classification," presented at the 2020 International Conference on Computing, Electronics & Communications Engineering (iCCECE), IEEE, 2020.
- [9] X. Polisi, A. Halili, C. E. Tanase, A. Uka, N. E. Vrana, and A. Ghaemmaghami, *Computer assisted analysis of the hepatic spheroid formation. In Computational Bioengineering and Bioinformatics: Computer Modelling in Bioengineering 8*. Switzerland: Springer International Publishing, 2020, pp. 117–126.
- [10] A. Uka, A. Ndreu Halili, X. Polisi, A. O. Topal, G. Imeraj, and N. E. Vrana, "Basis of image analysis for evaluating cell biomaterial interaction using brightfield microscopy," *Cells Tissues Organs*, vol. 210, no. 2, pp. 77–104, 2021. <https://doi.org/10.1159/000514153>
- [11] N. Siddique, S. Paheding, C. P. Elkin, and V. Devabhaktuni, "U-net and its variants for medical image segmentation: A review of theory and applications," *IEEE Access*, vol. 9, pp. 82031–82057, 2021. <https://doi.org/10.1109/ACCESS.2021.3089276>
- [12] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," presented at the Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, proceedings, part III 18, Springer International Publishing, 2015.
- [13] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3523–3542, 2021. <https://doi.org/10.1109/TPAMI.2021.3052174>
- [14] R. Azad *et al.*, "Medical image segmentation review: The success of u-net," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, pp. 10076 – 10095, 2024.
- [15] R. Younis, R. Ghnemat, and J. Al Sarairoh, "Fine-tuning U-net for medical image segmentation based on activation function, optimizer and pooling layer," *International Journal of Electrical & Computer Engineering (2088-8708)*, vol. 13, no. 5, pp. 5406–5417, 2023. <https://doi.org/10.11591/ijece.v13i5.pp5406-5417>
- [16] M. Montazerolghaem, Y. Sun, G. Sasso, and A. Haworth, "U-Net architecture for prostate segmentation: The impact of loss function on system performance," *Bioengineering*, vol. 10, no. 4, p. 412, 2023. <https://doi.org/10.3390/bioengineering10040412>
- [17] P. D. W. Ayu and G. A. Pradipta, "U-Net Tuning Hyperparameter for Segmentation in Amniotic Fluid Ultrasonography Image," presented at the 2022 4th International Conference on Cybernetics and Intelligent System (ICORIS), IEEE, 2022.

- [18] S. Saifullah and R. Dreżewski, "Automatic brain tumor segmentation using convolutional neural networks: U-net framework with pso-tuned hyperparameters," presented at the International Conference on Parallel Problem Solving from Nature, Cham: Springer Nature Switzerland, 2024.
- [19] B. Karlik and A. V. Olgac, "Performance analysis of various activation functions in generalized MLP architectures of neural networks," *International Journal of Artificial Intelligence and Expert Systems*, vol. 1, no. 4, pp. 111-122, 2011. <https://doi.org/10.5121/ijaies.2011.1409>
- [20] I. Kandel and M. Castelli, "The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset," *ICT Express*, vol. 6, no. 4, pp. 312-315, 2020. <https://doi.org/10.1016/j.ict.2020.07.004>
- [21] S. Gonzalez and R. Miiikkulainen, "Improved training speed, accuracy, and data utilization through loss function optimization," presented at the 2020 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2020.
- [22] X. Polisi, D. Avdiu, A. Uka, A. N. Halili, K. Kollcaku, and C. Ciulla, "Evaluation of cell segmentation using pruning and quantization," presented at the 2023 International Conference on Computing, Electronics & Communications Engineering (iCCECE), IEEE, 2023.
- [23] D. Kingma and J. B. Adam, "A method for stochastic optimization," presented at the International Conference on Learning Representations (ICLR), 2015.
- [24] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of machine learning research*, vol. 12, no. 7, pp. 2121-2159, 2011. <https://doi.org/10.5555/1953048.2078182>
- [25] T. Dozat, "Incorporating Nesterov momentum into Adam," in *Proceedings of the 4th International Conference on Learning Representations (ICLR 2016)*, 2016.
- [26] T. Tieleman and G. Hinton, "Divide the gradient by a running average of its recent magnitude. Coursera: Neural Networks for Machine Learning. Technical report," Retrieved: <https://www.coursera.org/learn/neural-networks-for-machine-learning>, 2017.
- [27] V. Rajput, "Robustness of different loss functions and their impact on network's learning," *Available at SSRN 4065778*, 2022.
- [28] C. Desai, "Comparative analysis of optimizers in deep neural networks," *International Journal of Innovative Science and Research Technology*, vol. 5, no. 10, pp. 959-962, 2020.
- [29] B. Neupane, J. Aryal, and A. Rajabifard, "Supervised domain adaptation for building extraction from off-nadir aerial images," *arXiv preprint arXiv:2311.03867*, 2023.
- [30] Z. Li and S. Arora, "An exponential learning rate schedule for deep learning," *arXiv preprint arXiv:1910.07454*, 2019.
- [31] A. Lewkowycz, Y. Bahri, E. Dyer, J. Sohl-Dickstein, and G. Gur-Ari, "The large learning rate phase of deep learning: The catapult mechanism," *arXiv preprint arXiv:2003.02218*, 2020.
- [32] S. L. Smith, P.-J. Kindermans, C. Ying, and Q. V. Le, "Don't decay the learning rate, increase the batch size," *arXiv preprint arXiv:1711.00489*, 2017.
- [33] X. Duro, V. Sharka, U. Arban, D. Karras, and A. Halili, "Deep learning-based detection, segmentation, and average area estimation of cells," *Journal of Natural and Technical Sciences*, vol. 29, no. 1, pp. 1-15, 2024.