Fat-U-Net: Non-Contracting U-Net for Free-Space Optical Neural Networks

Riad Ibadulla[a], Constantino C. Reyes-Aldasoro[a], and Thomas M. Chen[a]

[a]City, University of London, Northampton Square, London, UK, EC1V 0HB

ABSTRACT

This paper describes the advantages and disadvantages of adapting the U-Net architecture from a traditional GPU to a 4f free-space optical environment. The implementation is based on an optical-based acceleration called FatNet and thus this adaption is called Fat-U-Net. Fat-U-Net neglects the pooling operations in U-Net, but maintains a similar number of weights and pixels per layer as U-Net. Our results demonstrate that the conversion to Fat-U-Net offers significant improvement in speed for segmentation tasks, with Fat-U-Net achieving a remarkable $\times 538$ acceleration in inference compared to U-Net when both are run on optical devices and x37 acceleration in inference compared to the results provided by U-Net on GPU. The performance loss after conversion remains minimal in two datasets, with reductions of 4.24% in IoU for the Oxford IIIt pet dataset and 1.76% in IoU of HeLa cells nucleus segmentation.

Keywords: FatNet, HeLa segmentation, Optical Neural Network, segmentation

1. INTRODUCTION

The introduction of deep learning in computer vision applications has completely changed how digital images are processed and analysed. The application of Deep learning approaches to image segmentation has demonstrated remarkable results.[1–3] However, as the complexity of these machine learning models grows, so does the computational demand and the difficulty of real-time applications. While hardware accelerators, such as graphics processing units (GPUs), field-programmable gate arrays (FPGAs), and application-specific integrated circuits (ASICs), have emerged as a potential solution to this challenge, their effectiveness may be limited in the long run as Moore's Law begins to lose its predictive power.[4]

Advances in optical computing have shown the potential of optical accelerators to complement electronics-based hardware accelerators. Since optical computing is unaffected by Moore's law, it can be used for deep learning through optical accelerators, offering advantages such as higher bandwidth, faster processing, no resistance, reduced power consumption and immunity to thermal disturbances.[5] Two primary methods exist for building optical neural networks: the free space approach employing spatial light modulators (SLMs), and the silicon photonics method which utilises Mach–Zehnder interferometers (MZIs). The free space approach relies on light travelling through mediums like air, outer space, or a vacuum, unlike silicon photonics which relies on guided light paths. While the silicon photonics technique offers higher speed, with potential clock speeds of several GHz, it lags behind the free-space method in terms of parallel processing capabilities.[6] Free-space optical accelerators provide massive parallelism capabilities, and 4f optical accelerators can perform convolution operations faster than the traditional electronic processor with theoretical infinite resolution.[7] In practice, they are limited by the resolution of the modulators and the speed of the cameras used. In this research, we focus on the 4f free-space approach as described in Li et al.[8] in order to accelerate the inference and training of convolutional neural networks (CNNs) for image segmentation.

One of the main tasks in computer vision, semantic segmentation aims to partition an image into meaningful segments by assigning a class to each pixel. According to Peng *et al.*,[9] semantic segmentation is considered a pixel-wise classification problem, and a well-designed segmentation model should simultaneously encompass two tasks, classification and localisation. It was observed that these tasks are naturally "contradictory", as improving

Further author information: (Send correspondence to R.I.)
R.I.: E-mail: riad.ibadulla@city.ac.uk

one tends to diminish the other. This is because the classification model is insensitive to translation or rotation, while the localisation task should contain information regarding the appropriate coordinates in the output mask. For that reason, the classification models consist of pooling operations to extract the features at every scale. When having small 3x3 kernels on the deepest layers of the classification network, the kernel size - feature map resolution ratio is larger than on the shallow layers. Hence the features of the deeper layers can be affected by more pixels of the original image. This is why classification networks are mostly pyramid- or cone-shaped. Ideally, a barrel-shaped segmentation model would locate pixels of each class more precisely.

Although Peng *et al.*[9] proposed their own architecture, Global CNN, the well-established U-Net[1] can also address both problems simultaneously, where the contracting path solves the classification task, expanding path and skip connections support the localisation. However, it can be seen that most of the segmentation networks,[2,10] including U-Net, use an encoder-decoder structure or contain some piece of architecture for pulling the feature maps down in order to extract the features.[11] These networks simply inherit the successful structure of predecessor networks like AlexNet[12] or LeNet,[13] which are meant to do the classification task. It is important to keep in mind that higher-resolution feature maps and kernels are more suitable for segmentation tasks since high-resolution kernels have a higher effective receptive field than de-facto standard 3x3 kernels.

One of the key advantages of employing small kernels and cone encoder-decoder architectures is the speed of inference on CPU/GPU-based hardware. On the other hand, this acceleration in optical environments can be provided intrinsically by the optics. In previous work, we proposed the FatNet[14]conversion for classifier CNNs which reduced the number of channels and increased the kernel size and resolution of the feature map accordingly by keeping same number of parameters and pixels in each feature maps of the network. This conversion makes the network more suitable for 4f free-space optical acceleration.

In order to understand the reason behind it, it is worth looking at the principle of work in the 4f optical neural network accelerator. The 4f setup consists of an input laser, two convex lenses, and modulators. The idea is based on the Fourier transform properties of the convex lenses and performs the convolution operation based on the convolution theorem. Any convex lens projects a Fourier transform of the input object located on the front focal plane onto the back focal plane.[15] At this point, it can be pointwise multiplied by the kernel in the Fourier domain. After passing through the second lens, the multiplied output is converted back into the spatial domain and read by the camera. The process described above can perform the convolution operation using the 4f system, but in order to be able to apply the method to the convolutional neural network, it is essential to read the output of the 4f system, apply the activation function electronically and repeat the process. This causes the main bottleneck of the optical acceleration. Moreover, it is important to note that the resolution of the input and the kernel will not affect the system's frame rate. Hence, in order to maximise the utilisation of the system, the number of conversions to electronics should be reduced, but the resolution should be used to our advantage.

One of the obvious ways to utilise the high-resolution capabilities of the 4f system is to tile the inputs and kernels and perform convolutions in parallel, in other words, perform the batch tiling. According to Li *et al.*,[16] the high-resolution capabilities of 4f system can also be used to tile the channels and kernels, to perform several 2D convolution operations of one convolutional layer simultaneously. However, the FatNet algorithm ensures faster training in the 4f optical accelerator by reducing a number of channels and increasing the resolutions of the feature maps and kernels of CNNs, while relying on batch tiling. As the resolution is not an obstacle for the 4f optical accelerator, while fewer convolution operations mean fewer optics-electronics conversions. However, it can be assumed the FatNet conversion is even more suitable for segmentation tasks, which we have proposed in our work, and developed a Fat-U-Net, which is described in section 2.4.

Our previous work[14] was based on the conversion of the ResNet-18 into the FatNet. In this work, we demonstrate the possibility of expanding the FatNet further for segmentation tasks with U-Net, turning it into a Fat-U-Net. Notably, Fat-U-Net achieves a theoretical $\times 538$ faster inference when run on optical devices and $\times 37$ acceleration in inference compared to the results provided by U-Net on GPU.

Moreover, this work demonstrates the validity of the FatNet conversion algorithm. We trained other networks, called Intuitive Fat-U-Nets, with fewer channels and larger kernels, which did not adhere to the FatNet conversion principles. These networks were converted from U-Net based on the number of weights, without considering the number of pixels in each feature map. Despite this, none could outperform the original Fat-U-Net in terms of performance.

The performances of U-Net and Fat-U-Net implementations are compared using the Oxford IIIt pet dataset and HeLa cells dataset.

2. MATERIALS AND METHODS

2.1 Oxford-IIIT Pet

The first dataset analysed was the Oxford-IIIT pet dataset[17] developed by the Visual Geometry Group, consisting of 7,359 images and 37 pet categories, each containing approximately 200 images. These images exhibit significant differences in scale, pose, and lighting conditions. Each image is accompanied by ground truth annotations, including breed identification, head region of interest (ROI), and pixel-level trimap segmentation. Since our network is focused on segmentation, we are not taking the classes into account but focusing on the segmentation of the pets and backgrounds. The dataset provides a train-test split, where 3,680 images are designated for training and 3,669 for testing. Because the images are of different sizes, the images are resized to 160x160 in this work. The intensity of the channels of the dataset is normalised between 0 and 1, and the centring is performed with the mean of (0.485, 0.456, 0.406) and standard deviation of (0.229, 0.224, 0.225) for each RGB channel, respectively.

2.2 HeLa Cells

The second case analysed a high-resolution dataset of HeLa cells observed with Serial Block Face Scanning Electron Microscopy. It consisted of $8192 \times 8192 \times 518$ voxels[18] from which a $2000 \times 2000 \times 300$ region of interest (ROI) with a single cell has been cropped.[19] The hand-segmented ground truth (GT) with four classes (background, cell, nuclear envelope, and nucleus) is publicly available only for the ROI[20] and GT can be generated with image-processing algorithms .[21] In this work, we focus only on the segmentation of the nucleus.

Since our version of Fat-U-Net was designed for 160x160 images, we have prepared patches of 160x160 from odd-numbered slices of the ROI with 50% overlap. Taking only half of the slices and accounting for the 529 patches within each slice, we generated 79,350 image pairs along with their corresponding ground truth masks. Before saving the dataset of patches, all patches were low-passed filtered with a Gaussian filter. For that reason, we perform the same Gaussian filtering every time when evaluating new data.

Performing the data split among the shuffled patches could potentially result in a training or test set biased towards a specific class due to the inclusion of an excessive number of background images. Therefore, we performed a train-test split on a per-slice basis. Slices $(1, 11, 21, \ldots, 281, 291)$ were set as test slices, and slices $(5, 25, 45, \ldots, 285)$ were set as validation slices. Originally the rest of the data was used for training. However, as the shallowest and deepest slices contain only background, this leads to data imbalance and the training slices were set only to the slices where the cell and the nucleus are fully visible in the middle of the ROI. With this strategy, the training slices were defined within the range of 97 to 183 with step 2, excluding slices ending with 1 and 5, which resulted in 26 slices and 13,754 patches, such as (97, 99, 103, 107, 109,..., 177, 179). Although the number of training patches may appear limited, it is sufficient for binary nucleus segmentation. In contrast to Karabag *et al.*,[21] we ensured that our model evaluation did not include any slices that were part of the training process.

No data augmentation was applied to the dataset in this study. However, normalisation was performed to scale the data values between 0 and 1. Furthermore, centring was conducted using the calculated mean and standard deviation values, which were determined to be 0.6379 and 0.0855, respectively.

2.3 U-Net

U-Net[1] is a CNN architecture initially developed for the segmentation of biomedical images. Its unique architecture, consisting of contracting and expanding paths, allows it to capture local and contextual information effectively, leading to impressive segmentation results. The contracting path of the network can be seen as the typical CNN used for classification. It consists of blocks of convolutional layers, activation functions and pooling operations for feature extraction at different scales.

The expanding path of the U-Net serves for the upsampling of the extracted features to reconstruct the segmentation mask of the input image. This is achieved using transposed convolution operations to upsample the

feature maps and concatenation with the corresponding feature maps of the same resolution from the contracting path. The main role of the skip connections is to conserve the spatial information that is lost during the pooling process in the contracting path. Our implementation of U-Net is shown in Figure 1(a). It contains five stages, and unlike the original implementation of U-Net by Ronnenberg *et al.*,[1] it does not require cropping of the feature maps when performing skip concatenation, as it only uses convolutions with the "same padding".

2.4 Fat-U-Net

The idea of *Fat* layers, i.e., layers where there is no reduction in size, was introduced in[14] for the conversion of the CNN for classification into a form which is more compatible with 4f free-space optical accelerators. The underlying principle of FatNet conversion is to maintain the constant number of trainable parameters and the pixels in each layer while increasing the resolution of feature maps and kernels and decreasing the number of channels in each layer. By making this conversion, the network takes full advantage of the high-resolution capabilities of the 4f system, thereby optimising its performance and efficiency in the context of free-space optical acceleration. Since the main bottleneck of the free-space 4f accelerator is the latency of the camera, the fewer convolution operations that the networks have, the fewer optic-electronics conversions are required. Eventually, the cone-shaped classifier convolutional networks turn into barrel-shaped networks with higher-resolution feature maps and high-resolution kernels, which sometimes reach the size of the feature maps making it a "Fat" Layer.

The original FatNet conversion, designed specifically for the classification task, maintains the same architecture as the original network until the feature maps are pooled down to the resolution with a number of pixels less than or equal to the number of classes. It is posited that when it comes to the FatNet conversion for the segmentation, pooling may be unnecessary, and the input resolution can be preserved throughout the entire network. Consequently, increasing the resolution of kernels while keeping the resolution of the feature maps constant would decrease the feature map-to-kernel resolution ratio, emulating the effect of pooling the feature maps without actual pooling implementation. This approach can significantly increase the inference time of the network run on the 4f free-space optical accelerator and hypothetically retains localisation accuracy even more effectively.

U-Net contracting		weights	pixels	New layers		FatU-Net adjusted	
Channels	kernel			Channels	kernel	Channels	kernel
3×64	3	1,728	1,638,400	3×64	3	3×32	5
64×64	3	36,864	409,600	32×32	6	32×32	6
64×128	3	73,728	819,200	32×32	9	32×16	12
128×128	3	147,456	204,800	16×16	24	16×16	24
128×256	3	294,912	409,600	16×16	34	16×8	48
256×256	3	589,824	102,400	8×8	96	8×8	96
256×512	3	1,179,648	204,800	8×8	136	8×10	122
512×512	3	2,359,296	51,200	10×10	160	10×10	160
512×1024	3	4,718,592	102,400	10×18	160	10×20	154
1024×1024	3	9,437,184	102,400	20×20	160	20×20	160

Table 1. Construction table for Fat-U-Net's first half out of the U-Net's contracting path.

Since the original FatNet was designed for classification, only the contracting path of the U-Net was converted into the FatNet. Table 1 presents the Fat-U-Net equivalent of the FatNet construction table, as described in.[14] The table is used to compute the number of weights per layer, excluding the bias and the number of pixels per layer. The algorithm ensures the convolutional layers with the same number of input and output channels within convolutional blocks have an equal number of input and output channels after the conversion too. Upon completing the conversion of the contracting path of the U-Net into FatNet, the path was mirrored to generate the "expanding path", and the kernel sizes were recalculated to match the number of weights from the original layers. Since the so-called expanding path of the Fat-U-Net does not actually require upsampling, we have replaced the deconvolution operations with the simple 3x3 convolutions as illustrated in Figure 1(b).

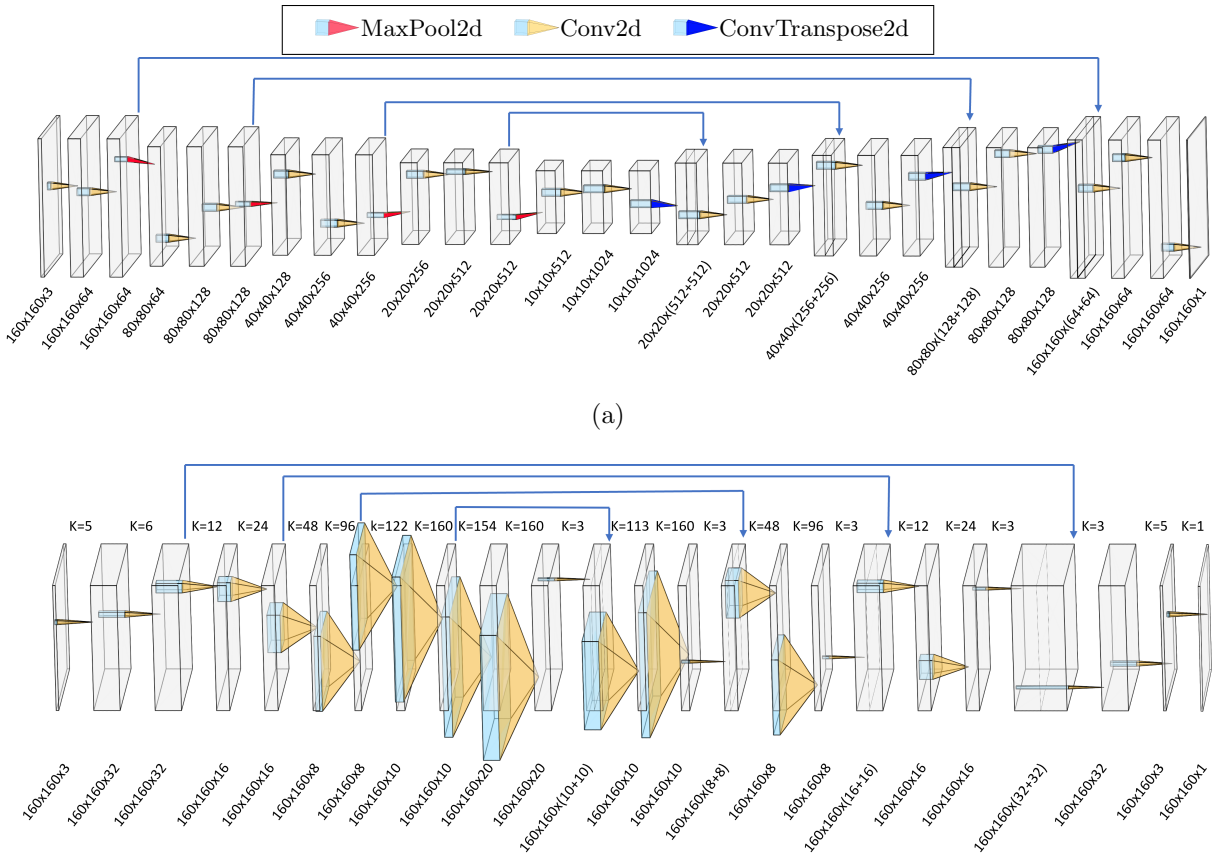


Figure 1. Graphical representation of our implementation of U-Net and Fat-U-Net architectures. (a) U-Net architecture, with all kernel sizes 3x3, MaxPool with kernels size of 2x2 and deconvolution operations with a kernel size of 3x3. (b) Fat-U-Net architecture derived from our implementation of U-Net, with the varying kernel sizes indicated as K at each layer.

3. EXPERIMENTS AND RESULTS

U-Net and the Fat-U-Net equivalent were implemented and tested in two segmentation tasks of the Oxford IIIt pet and HeLa cells. Performance was assessed by pixel-wise Accuracy, Intersection over Union (IoU), and Dice Score:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$IoU = \frac{TP}{TP + FP + FN} \tag{2}$$

$$DiceScore = \frac{2TP}{2TP + FP + FN} \tag{3}$$

Inference time was also measured on GPU and theoretically calculated for the optics to demonstrate the acceleration on a 4f free-space optical device. Overall if our U-Net implementation contains 3,833,984 convolution operations, its Fat-U-Net equivalent contains only 7,123. Since the resolution does not affect the speed of inference

in the 4f free-space optics, the inference of Fat-U-Net in optics will be **538 times faster** than U-Net if both run in optics. This acceleration is possible with only a small sacrifice in performance, as seen in Tables 3, 4.

We have measured the inference time of Nvidia A100 with both U-Net and Fat-U-Net, and compared the results to the calculated theoretical inference time on 4f optical accelerator based on Li *et al.*[16] The results are shown in Table 2 for batch sizes of 1, 32, and 144. The batch size of 144 was chosen because it is the maximum possible batch size for the 4f system with 4k resolution, if batch tiling is applied.

Based on the results in Table 2, at the batch size of 144, the acceleration of inference of Fat-U-Net with 4f optics, compared to U-Net run on high-end GPU, is **37 times**.

Model and device	**Batch 1**	**Batch 32**	**Batch 144**
U-Net (Optics)	1920.00	59.900	13.300
Fat-U-Net (Optics)	3.46	0.108	**0.024**
U-Net (GPU)	4.55	0.894	0.883

Table 2. Inference time in milliseconds of U-Net and its Fat-U-Net equivalent model per image with different batch sizes run on 4f accelerator and Nvidia A100. The frame rate for 4f system was approximated at 2 MHz, and Nvidia A100 GPU was measured experimentally.

3.1 Oxford IIIt pet

Training of the Oxford IIIt pet dataset used the Adam optimiser; the learning rate was set to 1e-4 with a batch size of 16 and a number of epochs of 250. The training data went through augmentation during training, by random shift, scale, rotation, RGB shift, random brightness and contrast. We have used the BCEWithLogitsLoss of PyTorch, which combines Binary Cross-Entropy loss with the sigmoid layer. We have ensured that our U-Net results adhered to state-of-the-art standards before converting them into the Fat-U-Net and conducting the comparison of evaluation metrics between Fat-U-Net, its backbone U-Net, and previous research employing the Oxford IIIt pet dataset as a benchmark (Table 3).

Model	**Accuracy (%)**	**IoU (%)**	**Dice Score (%)**
U-Net (our implementation)	**95.33**	89.32	**94.33**
Fat-U-Net (ours)	93.40	85.08	91.87
SEU-Net[22]	-	$\approx$ 77.00	-
ICNet[23,24]	90.79	75.12	-
ConRec (20% of dataset) [25,26]	-	-	90.00
U-Net (as per Sundarrajan *et al.*)[27]	-	33.30	46.40
U-Net+VGG16[27]	-	89.40	94.20
U-Net+InceptionV3[27]	-	**91.60**	91.50

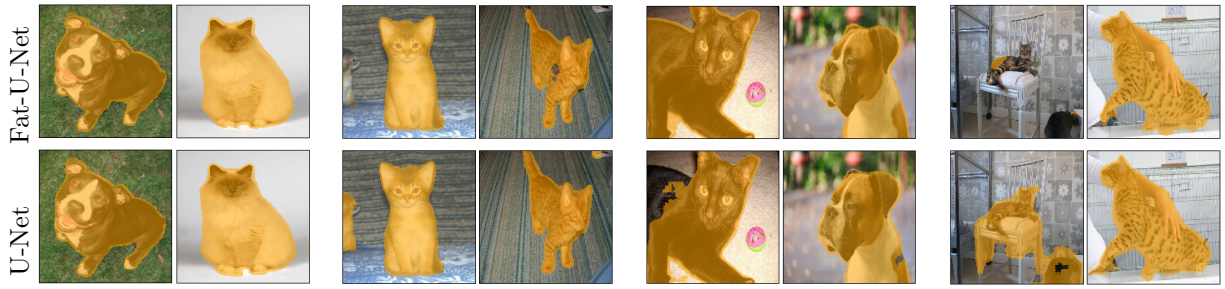
Table 3. Comparison of the evaluation results of the accuracy, mIoU, and Dice score of U-Net and its Fat-U-Net equivalent along with other works for Oxford IIIt pet.

We have visualised the predicted mask on the data for both U-Net and Fat-U-Net in Figure 2, to understand where the segmentation is excellent, where it is unacceptable, and where Fat-U-Net outperforms U-Net or vice versa.

3.2 HeLa cells

The Adam optimiser was used to train HeLa nucleus segmentation as well, with the inclusion of a weight decay set at 1e-4. We have applied two dropout layers with a probability of 50% to the beginning and end of the bridge section of U-Nets. The learning rate was set to 1e-3, with a batch size of 32 and a number of epochs of 20. The loss function remained the same, BCEWithLogitsLoss, which combines binary cross-entropy loss and the sigmoid layer.

Both U-Net and Fat-U-Net models were evaluated with four scenarios: (1) the complete set of odd slices ranging from 1 to 300, (2) the middle-range of odd slices (150-200) where the nucleus is visible, and (3)-(4) then



(a) (b) (c) (d)

Figure 2. Qualitative results of Oxford IIIt pet dataset. (a) Examples of perfect segmentation by both algorithms. (b) Examples of U-Net performing better than Fat-U-Net. (c) Examples of Fat-U-Net outperforming U-Net. (d) Bad segmentation examples by both algorithms

Model	**Acc.(all) (%)**	**IoU(all) (%)**	**Acc.(150-200) (%)**	**IoU(150-200) (%)**
U-Net	95.71	66.32	99.59	**97.15**
U-Net (Test data)	**95.75**	**66.59**	99.57	97.11
Fat-U-Net	95.31	64.27	99.42	96.05
Fat-U-Net (Test data)	95.42	64.83	99.43	96.25
4 stage U-Net[21]	93.46	51.38	**99.66**	97.12

Table 4. Performance comparison of our implementation of five staged U-Net, its Fat-U-Net equivalent, and a four staged U-Net implemented in.[21] Evaluating Accuracy and IoU Metrics Across the entire dataset and 150-200 range for all odd and test slices that have not participated in the training process.

repeating the same strategies for the test slices (See Table 4). The results of the first two scenarios, which include both training and validation slices, were comparable to the results of the work of Karabag *et al.* [21]

Since GT was only available for the ROI, which is one cell of the larger $8192 \times 8192 \times 518$ datasets, qualitative tests were performed training on one cell and testing in an adjacent cell as demonstrated in Figure 3. Moreover, the qualitative tests were also performed on the segmentation of the larger original image of 8192×8192 containing all the cells (Figure 4).

All qualitative evaluation was performed for both U-Net and Fat-U-Net, for comparison purposes.

3.3 Validity of FatNet

To demonstrate the efficacy of the FatNet conversion, we have trained alternate networks with fewer channels and larger kernels. These networks, which we call Intuitive Fat-U-Nets, deviate from the FatNet conversion formula by focusing only on the number of weights and not considering the pixel count in each feature map. Three versions of Intuitive Fat-U-Net were designed and shown in Table 5.

Among all networks, Intuitive Fat-U-Net 1 is the closest to the original U-Net as the channels in the bottleneck rise up to 128, with the largest kernel size being 24. However, even Intuitive Fat-U-Net 1 performed worse than the Fat-U-Net as it can be seen in Table 6. While the Intuitive Fat-U-Net 3, the closest network to the Fat-U-Net with the largest kernel size, 153, performed the worst of all networks.

4. DISCUSSION

In this study, we successfully demonstrated that the FatNet conversion of in silico networks to optical devices is more efficient for segmentation tasks than for classification. For comparison, Ibadulla *et al.*[14] reported an acceleration of 8.2 times for the ResNet-18, if ResNet-18 and FatNet run on the optical device. This work shows a remarkable 538 faster inference of Fat-U-Net compared to the U-Net under the same conditions and $\times 37$ acceleration in inference compared to the results provided by U-Net on GPU. Moreover, from Table 2 it can be seen that the GPU Nvidia A100, being one of the best hardware accelerators, outperforms optical accelerator of

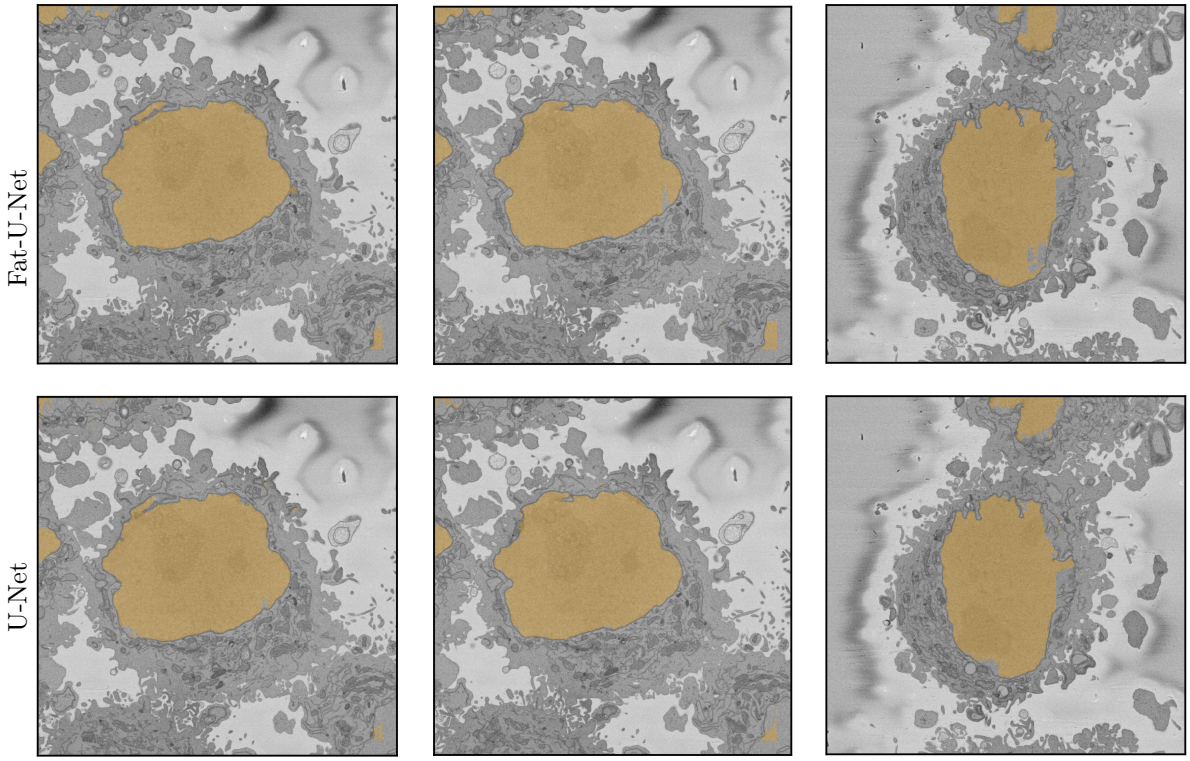


Figure 3. Qualitative results of HeLa dataset. (a) train slice 119 (b) test slice 121 (c) Unseen cell, taken from the larger field.

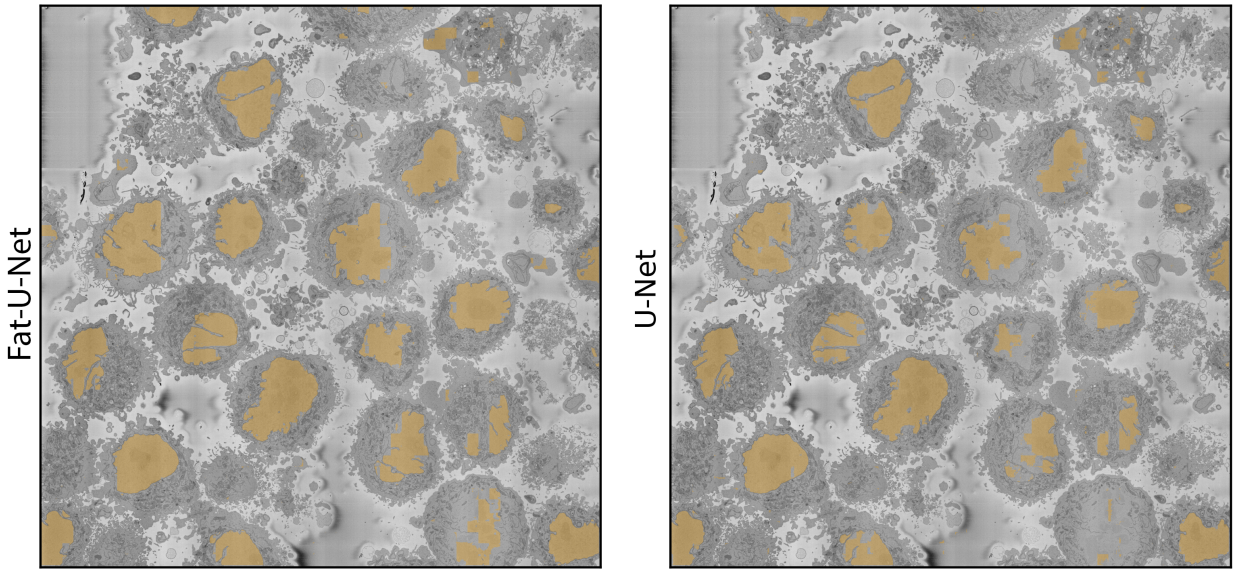


Figure 4. U-Net and Fat-U-Net segmentation results on 8192×8192 images.

2 MHz frame rate when running U-Net, but stays slow for all batch sizes when compared to Fat-U-Net run on optical accelerator. Given that the 4f optical device is meant to accelerate only the convolution operations, it is intuitive that fully convolutional networks like U-Net are ideally suited for the 4f accelerators, as they do not even need any amendments of the dense layers, as required in the classification.

With the speed advantages of Fat-U-Net established, our next objective was to validate its performance. We

Layer	Intuitive Fat-U-Net 1		Intuitive Fat-U-Net 2		Intuitive Fat-U-Net 3	
	Channels	Kernel	Channels	Kernel	Channels	Kernel
Conv block 1	$3 \to 8$	8	$3 \to 4$	12	$3 \to 4$	12
	$8 \to 8$	24	$4 \to 4$	48	$4 \to 4$	48
Conv block 2	$8 \to 16$	24	$4 \to 8$	48	$4 \to 8$	48
	$16 \to 16$	24	$8 \to 8$	48	$8 \to 8$	48
Conv block 3	$16 \to 32$	24	$8 \to 16$	48	$8 \to 10$	61
	$32 \to 32$	24	$16 \to 16$	48	$10 \to 10$	77
Conv block 4	$32 \to 64$	24	$16 \to 32$	48	$10 \to 16$	85
	$64 \to 64$	24	$32 \to 32$	48	$16 \to 16$	96
Conv block 5	$64 \to 128$	24	$32 \to 64$	48	$16 \to 20$	121
(bottleneck)	$128 \to 128$	24	$64 \to 64$	48	$20 \to 20$	153
DeConv 1	$128 \to 64$	3	$64 \to 32$	3	$20 \to 16$	3
Conv block 6	$128 \to 64$	24	$64 \to 32$	48	$32 \to 16$	96
	$64 \to 64$	24	$32 \to 32$	48	$16 \to 16$	96
DeConv 2	$64 \to 32$	3	$32 \to 16$	3	$16 \to 10$	3
Conv block 7	$64 \to 32$	24	$32 \to 16$	48	$20 \to 10$	77
	$32 \to 32$	24	$16 \to 16$	48	$10 \to 10$	77
DeConv 3	$32 \to 16$	3	$16 \to 8$	3	$10 \to 8$	3
Conv block 8	$32 \to 16$	24	$16 \to 8$	48	$16 \to 8$	48
	$16 \to 16$	24	$8 \to 8$	48	$8 \to 8$	48
deconv4	$16 \to 8$	3	$8 \to 4$	3	$8 \to 4$	3
Conv block 9	$16 \to 8$	24	$8 \to 4$	48	$8 \to 4$	48
	$8 \to 3$	24	$4 \to 3$	55	$4 \to 3$	55
segmenter	$3 \to 1$	1	$3 \to 1$	1	$3 \to 1$	1

Table 5. Comparison of the architectures of the Intuitive Fat-U-Nets. Unlike a Fat-U-Net, which is converted using a FatNet algorithm for the conversion, these intuitive networks were developed manually by choosing smaller channel sizes and computing the new kernel sizes without taking into account the number of pixels in the feature map.

Model	Oxford IIIt pet		HeLa cells	
	Acc	IoU	Acc	IoU
Intuitive Fat-U-Net 1	92.71	83.71	99.08	93.90
Intuitive Fat-U-Net 2	89.39	77.84	97.95	87.58
Intuitive Fat-U-Net 3	89.18	76.98	98.45	89.75
Fat-U-Net	**93.40**	**91.87**	**99.43**	**96.25**

Table 6. Other "Large kernel/Few Channel" architectures in comparison with Fat-U-Net.

initially trained the U-Net to state-of-the-art standards before converting it to Fat-U-Net. Our U-Net implementation is marginally outperformed only by networks with pre-trained VGG16 and Inception V3 contracting paths (Table 3). As our implementation was trained from scratch, we believe it met the required standards before conversion. Fat-U-Net sacrificed only 1.93% in pixel accuracy, 4.24% in IoU, and 2.46% in Dice score. These results compare favourably to classification problems, where the accuracy drop was 6%.

Qualitative results in Figure 2 reveal that U-Net and Fat-U-Net exhibit distinct behaviour in various scenarios. Figure 2(a) is the demonstration of the perfect segmentation by both algorithms in instances where pets are clearly visible against a monochromatic background. Interestingly, in Figure 2(b), U-Net outperforms Fat-U-Net by segmenting a background cat, which is not part of the ROI. However, we can see the advantage of Fat-U-Net in Figure 2(c), where it has perfectly segmented both animals, in contrast to U-Net, which incorrectly classified some pixels of the cat and dog as background.

Our evaluation of Fat-U-Net for HeLa cell nucleus segmentation proved successful. Compared to the 4-staged U-Net,[21] our 5-stage U-Net implementation demonstrated marginally better performance on middle-range slices

and achieved a 14.94% higher IoU for all slices. It is important to consider that the ground truth for ROI cells includes only the segmentation of the central cell, excluding adjacent cells. Nevertheless, both U-Net and Fat-U-Net managed to segment these nuclei even with noisy ground truth data (Figure 3). Consequently, the segmented mask outperforms the ground truth on side slices (non-150-200), resulting in a lower IoU for all slices compared to middle-range slices. After converting to Fat-U-Net, the performance loss was smaller than in the Oxford IIIt pet dataset evaluation, at approximately 1% for middle-range slices and 2% for all slices. For the large images, Fat-U-Net provided better results than U-Net as can be seen in the cells on the bottom right.

To assess Fat-U-Net's performance in the original optical setup, it was trained using the 4f simulator. While this simulator does not completely replicate the real optics' performance, it demonstrated comparable results in the training for Hela Cells segmentation. Notably, it achieved an IoU of 95.58% on test slices 150-200 and 65.34% on all test slices.

5. CONCLUSION AND FUTURE WORK

In our research, we have successfully extended the application of FatNet conversion to the task of segmentation, by adapting the U-Net architecture for use with free-space optical accelerators. We have achieved 538 times fewer convolution operations in Fat-U-Net compared to U-Net, meaning 538 times faster inference when both networks run with the optical accelerator and 37 times faster inference compared to U-Net run on GPU. Both networks were evaluated across the Oxoford IIIt pet dataset and HeLa cell nucleus segmentation, on which we have achieved state-of-the-art performance. When it comes to the performance loss, the maximum loss was 4.24% in the test IoU for the Oxford IIIt pet dataset and 1.76% in the test IoU of HeLa cells nucleus segmentation, making the FatNet transformation even more preferable than the classification.

As this research primarily focuses on Fat-U-Net conversion, future work could investigate segmentation using only the contracting path of Fat-U-Net, to explore the advantages of high-resolution kernels in detail. Hypothetically, a U-Net with an extensive receptive field like in Fat-U-Net would not require skip connections. However, our experiments with U-Net and Fat-U-Net without skip connections yielded unsatisfactory results, even after removing the 3x3 convolutions that replaced transposed convolutions. A possible explanation is that Fat-U-Net maintains the U-Net architecture, and instead of pooling down feature maps, it increases kernel resolution, resulting in a feature map/kernel ratio similar to U-Net. Therefore, future work will include investigating the possibility of the enhancement of the effective receptive field by dropping the skip connections.

REFERENCES

[1] Ronneberger, O., Fischer, P., and Brox, T., "U-Net: Convolutional Networks for Biomedical Image Segmentation," (May 2015). arXiv:1505.04597 [cs].

[2] Badrinarayanan, V., Kendall, A., and Cipolla, R., "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39**, 2481–2495 (Dec. 2017).

[3] Farabet, C., Couprie, C., Najman, L., and LeCun, Y., "Learning Hierarchical Features for Scene Labeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **35**, 1915–1929 (Aug. 2013).

[4] Waldrop, M. M., "The chips are down for Moore's law," *Nature News* **530**, 144 (Feb. 2016). Cg_type: Nature News Section: News Feature.

[5] Lin, X., Rivenson, Y., Yardimci, N. T., Veli, M., Luo, Y., Jarrahi, M., and Ozcan, A., "All-optical machine learning using diffractive deep neural networks," *Science* **361**, 1004–1008 (Sept. 2018). Publisher: American Association for the Advancement of Science.

[6] Sui, X., Wu, Q., Liu, J., Chen, Q., and Gu, G., "A Review of Optical Neural Networks," *IEEE Access* **8**, 70773–70783 (2020).

[7] Miscuglio, M., Hu, Z., Li, S., George, J. K., Capanna, R., Dalir, H., Bardet, P. M., Gupta, P., and Sorger, V. J., "Massively parallel amplitude-only Fourier neural network," *Optica* **7**, 1812–1819 (Dec. 2020). Publisher: Optica Publishing Group.

[8] Li, B., Ersoy, O. K., Ma, C., Pan, Z., Wen, W., and Song, Z., "A 4F optical diffuser system with spatial light modulators for image data augmentation," *Optics Communications* **488**, 126859 (2021).

[9] Peng, C., Zhang, X., Yu, G., Luo, G., and Sun, J., "Large Kernel Matters – Improve Semantic Segmentation by Global Convolutional Network," in [*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*], 4353–4361 (2017).

[10] Noh, H., Hong, S., and Han, B., "Learning Deconvolution Network for Semantic Segmentation," in [*Proceedings of the IEEE International Conference on Computer Vision*], 1520–1528, IEEE, Santiago, Chile (2015).

[11] Long, J., Shelhamer, E., and Darrell, T., "Fully Convolutional Networks for Semantic Segmentation," in [*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*], 3431–3440 (2015).

[12] Krizhevsky, A., Sutskever, I., and Hinton, G. E., "ImageNet Classification with Deep Convolutional Neural Networks," in [*Advances in Neural Information Processing Systems*], **25**, Curran Associates, Inc. (2012).

[13] LeCun, Y., Jackel, L. D., Bottou, L., Cortes, C., Denker, J. S., Drucker, H., Guyon, I., Muller, U. A., Sackinger, E., Simard, P., and others, "Learning algorithms for classification: A comparison on handwritten digit recognition," *Neural networks: the statistical mechanics perspective* **261**(276), 2 (1995).

[14] Ibadulla, R., Chen, T. M., and Reyes-Aldasoro, C. C., "FatNet: High-Resolution Kernels for Classification Using Fully Convolutional Optical Neural Networks," *AI* **4**, 361–374 (June 2023). Number: 2 Publisher: Multidisciplinary Digital Publishing Institute.

[15] Culshaw, B., "The Fourier Transform Properties of Lenses," in [*Introducing Photonics*], 132–135, Cambridge University Press, Cambridge (2020).

[16] Li, S., Miscuglio, M., Sorger, V., and Gupta, P., "Channel Tiling for Improved Performance and Accuracy of Optical Neural Network Accelerators," *ArXiv* (2020).

[17] Parkhi, O. M., Vedaldi, A., Zisserman, A., and Jawahar, C. V., "Cats and dogs," in [*2012 IEEE Conference on Computer Vision and Pattern Recognition*], 3498–3505 (June 2012).

[18] Peddie, C. J., Jones, M. L., and Collinson, L. M., "Serial Block Face SEM of HeLa cell pellet with 10 nm pixels and 50 nm slices (benchmark dataset)," (May 2019). 10.6019/EMPIAR-10094.

[19] Peddie, C. J., Jones, M. L., and Collinson, L. M., "Cropped regions from Serial Block Face SEM of HeLa cell pellet with 10 nm pixels and 50 nm slices (benchmark dataset)," (Aug. 2020). 10.6019/EMPIAR-10478.

[20] Karabağ, C., Jones, M., and Reyes-Aldasoro, C. C., "Multiple Nuclei HeLa cell ground truth images with four labels (nuclear envelope, nucleus, rest of the cell, and background) for deep learning architecture training.," (Mar. 2022). doi.org/10.5281/zenodo.6355622.

[21] Karabağ, C., Ortega-Ruíz, M. A., and Reyes-Aldasoro, C. C., "Impact of Training Data, Ground Truth and Shape Variability in the Deep Learning-Based Semantic Segmentation of HeLa Cells Observed with Electron Microscopy," *Journal of Imaging* **9**, 59 (Mar. 2023). Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.

[22] Sangalli, M., Blusseau, S., Velasco-Forero, S., and Angulo, J., "Scale-Equivariant U-Net," in [*33rd British Machine Vision Conference 2022, London, UK*], {BMVA} Press, London, UK (Nov. 2022).

[23] Edwards, J. and El-Sharkawy, M., "uICNet: Lightweight Image Segmentation," in [*2022 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*], 99–104 (Oct. 2022).

[24] Zhao, H., Qi, X., Shen, X., Shi, J., and Jia, J., "ICNet for Real-Time Semantic Segmentation on High-Resolution Images," in [*Computer Vision – ECCV 2018*], Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y., eds., *Lecture Notes in Computer Science*, 418–434, Springer International Publishing, Cham (2018).

[25] Dippel, J., Lenga, M., Goerttler, T., Obermayer, K., and Höhne, J., "Transfer Learning for Segmentation Problems: Choose the Right Encoder and Skip the Decoder," (July 2022). arXiv:2207.14508 [cs].

[26] Dippel, J., Vogler, S., and Höhne, J., "Towards Fine-grained Visual Representations by Combining Contrastive Learning with Image Reconstruction and Attention-weighted Pooling," (Feb. 2022). arXiv:2104.04323 [cs].

[27] Sundarrajan, K., Rajendran, B. K., and Balasubramanian, D., "Fusion of Ensembled UNET and Ensembled FPN for Semantic Segmentation," *Traitement du Signal* **40**, 297–307 (Feb. 2023).