DenseNetFCN for Neuron Semantic Segmentation

Loris Pilotto, Moulik Choraria, Victor Stimpfling, Dr. Sahand Jamal Rahi
École Polytechnique Fédérale de Lausanne

***Abstract*—This paper describes the results of the machine learning master's course project offered by the Laboratory of the Physics of Biological Systems, which revolves around segmentation of neurons in 3D Videos.**

I. Introduction

With the development of 3D optical microscopy techniques, sufficiently high resolution videos of microscopic objects can be generated in a short amount of time. However, the manual labelling and annotation of the obtained 3D data remains a tedious task. Meanwhile, newer deep learning algorithms are showing an incredible performance on an impressive variety of tasks. Deep learning has shown, for instance great accuracy on biomedical data segmentation [2]. The goal of the project was then, to evaluate the effectiveness of a deep learning model in order to classify neurons in C.elegans. The specific task is to identify and distinguish two different nodule types (neuronal structures): the 'AIA terminal nodule' and the 'AIA central nodule' among background and other structures based on two annotated 3D movies of C. elegans brains.

II. Neural Network Architecture

Most state-of-the-art approaches for semantic image segmentation are built on Convolutional Neural Networks (CNNs). Several architectures in literature may be used for this task, and the authors of this paper settled on the Fully Convolutional DenseNets for Semantic Segmentation[1]. The motivation was two fold: First, the network achieves state of the art segmentation accuracy on certain tasks without much pre-training or post-processing, and that it does so with fewer trainable parameters compared to other approaches, significantly reducing training time.

Fully Convolutional Networks, an extension of basic CNN's, are known to achieve good performance on semantic image segmentation tasks with skip connections, which are used to combine fine grain information with coarse grain during upsampling. A ResNet [4] based FCN takes this idea further by adding shortcut connections that sum information from a previous layers. This paper utilizes a further improvement on this, that has been recently introduced: DenseNet [1].

DenseNet implements 'dense blocks' instead of 'residual blocks', which is composed of multiple convolutional layers. The output of each layer is concatenated with the input of the dense block and previous layers outputs, increasing the number of features in each layer (see figure 2 and Tab.I).

In the downsampling path, a series of dense blocks is applied. The input and its output are concatenated, thus augmenting the feature map. Then a 'transition down' block is applied as shown in Tab.II, which reduces the input dimensionality. In the upsampling path, a series of dense blocks is applied (without an input/output concatenation replaced by skip connections) followed by a 'transition up' block as shown in Tab. III to finally recover the input dimension.(Architecture summary 1)

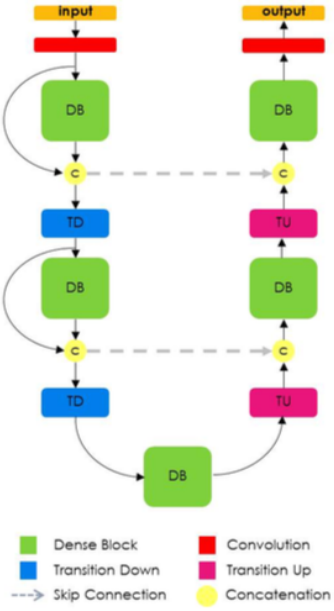


Fig. 1: DenseNet schematic

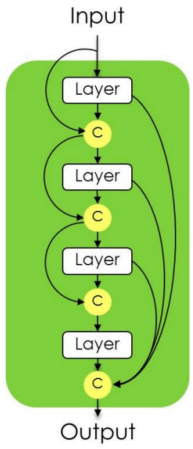


Fig. 2: Dense block with 4 layers

Layer
Batch Normalization
ReLu
f x 3 x 3 x 3 Convolution same padding

TABLE I: Structure of a layer within the dense block, f is the number of filters increasing as per growth rate

In this work, the authors focused on one architecture that was compatible with the available resources. Unless stated otherwise, the FCN had 4 dense blocks composed each of 3 layers. A growth rate of 16 was used thus each convolution layer within a dense block had an increasing number of filters starting at 16 and ending with 48. The global architecture is described in Tab. IV.

III. Dataset

The Laboratory of the Physics of Biological Systems provided two 3D video obtained by fluorescent microscopy of C.elegans brains. For this work, all frames were considered as different volumes not taking into account time continuity. Treating frames as separate images naturally increases the amount of data available for training, while also allowing to keep the architecture relatively simple. All frames in the 20190805_SJR3.2.2_w1_s2 video were 352x512x35, with a total of 286 frames. The preliminary training were all performed on down-sampled frames of size 80x128x35 due to resource constraints. In the z dimension for all frames, the 3 last layers are the boundary of the nematode brain and thus do not contain any information. They were thus truncated to obtain a final frame size of 352x512x32, which was convenient since the

Transition Down (TD)
Batch Normalization
ReLu
f x 1 x 1 x 1 Convolution same padding
2 x 2 x 2 Max Pooling

TABLE II: Structure of the transition down block: f stands for number of filters and is equal to 4^{th} dimension of input

Transition Up (TU)
f x 3 x 3 x 3 Transposed convolution stride = 2

TABLE III: Structure of the transition up block: f stands for number of filters and is equal to 4^{th} dimension of input

Architecture	Output shape
Input	(80, 128, 32, 1)
3 x 3 x 3 Convolution	(80, 128, 32, 48)
DB (3 layers) + TD	(40, 64, 16, 96)
DB (3 layers) + TD	(20, 32, 8, 144)
DB (3 layers) + TD	(10, 16, 4, 192)
DB (3 layers) + TD	(5, 8, 2, 240)
DB (3 layers)	(5, 8, 2, 272)
TU + DB (3 layers)	(10, 16, 4, 320)
TU + DB (3 layers)	(20, 32, 8, 272)
TU + DB (3 layers)	(40, 64, 16, 224)
TU + DB (3 layers)	(80, 128, 32, 176)
1 x 1 x 1 Convolution	(80, 128, 32, c)

TABLE IV: Detail standard network architecture c is the number of class

model requires each dimension to be multiple of 16 for working skip connections, thus simplifying down-sampling. All networks unless stated otherwise were trained on random frames extracted from the 20190805_SJR3.2.2_w1_s2 video..

IV. Experiments

The main objective of this project was to train a network which when fed 3D video frames of C.Elegans brains, could correctly differentiate between the two different nodules and hence make the annotation task easier for the manual annotator. In that regard, two properties are desired. The primary objective is that given a few frames from the video, the network is able to learn and generalize on the rest of the frames, which will be referred to as learning/generalisation over time. The secondary objective is the ability to differentiate between the same two nodules but across different C.Elegans brain samples to achieve generalisation, which should intuitively be a much harder task for the network due to availability of only 2 different samples.

Two popular metrics were used to evaluate the performance, the categorical accuracy (1) and the categorical F1 scores (2). For generalisation over time, these metrics were computed on a validation set from the same video which were unseen by the network while training. Dealing with the generalisation through videos, the same metrics were computed on a data set extracted from video 20190805_322_w1_s2.

$$accuracy := \frac{nbr\,pixels\,accurately\,classified}{total\,nbr\,of\,pixel\,in\,that\,class} \quad (1)$$

$$F1\,score = \frac{2*True\,positive}{2*True\,positive + False\,negative + False\,positive} \quad (2)$$

A. Pre-processing

One of the most powerful aspects of deploying a deep learning model for image data analysis is that it theoretically doesn't need much feature pre-processing. However, real life constraints such as memory and computational power limit this learning ability and thus one may explore possible pre-processing methods before training.

In the case of FCN-DenseNet, there is little need for implementing standard image processing techniques for image pre-processing simply because it itself consists of learnable filters of various sizes and therefore, is able to learn any morphological operator without human supervision. It has however been observed that noisy input data can adversely affect the performance of the network. Recently, Alexander et al. came up with a novel method to train networks to denoise from single noisy images, Noise2Void[3]. It is shown to be effective for biological data-sets, and thus was an interesting avenue to pursue.

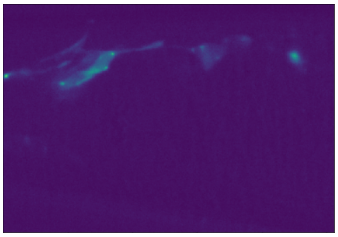


Fig. 3: Original-Z slice

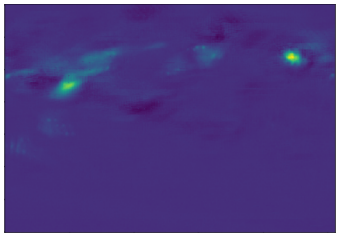


Fig. 4: N2V-Z slice

However this technique was not effective for the dataset, since as visible in the illustration, the blurring effect erodes the different feature boundaries causing them to somewhat blend into each other, rendering the available segmentation masks not fit for purpose. Secondly, it also seemed to over-amplify the noise with high pixel values(the yellow dots). Therefore, after discussions with the teacher, the authors decided not to explore N2V further for denoising, sticking to simple max-min normalization for preprocessing.

B. Improving Performance

As is often the case in segmentation tasks, there is a strong pixel imbalance in terms of the ratio of the background to the class labels in the data (approximately 1000:1). This imbalance can potentially cause the vanilla FCN to misclassify all relevant class labels as 0, and still achieve low training loss. Therefore, there is a need to reweigh the loss function, to cause it to penalize the network more in case the network misclassifies one of the relevant classes.

1) Weight adjustments: One of the most common methods of modifying the loss function in such a case is adding weights to each class, the weights being inversely proportional to the frequency of occurrence of each class.
$weight\,class\,c = \frac{total\,number\,of\,pixel}{number\,of\,pixel\,in\,class\,c}$.
After training a vanilla FCN on the initial task with the modified loss function, the following results were obtained on the validation set [V]. All predictions were reshaped to the initial size [352 x 512 x 35] and compared to the original masks.

Furthermore the same network was trained on the 20190805_322_w1_s2 video.

The results show that the two networks behave in a similar manner, justifying the choice to conduct all experiments on the

	Background	AIA terminal nodule (class 1)	AIA central nodule (class 2)
Categorical accuracy	0.994	0.988	0.97
F1 score	0.997	0.43	0.438

TABLE V: Accuracy and F1 scores for 3 classes, Video 1

	Background	AIA terminal nodule (class 1)	AIA central nodule (class 2)
Categorical accuracy	0.996	0.873	0.991
F1 score	0.998	0.392	0.154

TABLE VI: Accuracy and F1 scores for 3 classes, Video 2

20190805_SJR3.2.2_w1_s2 video, and generalize results on that basis. The results also showed however, that while the model has excellent categorical accuracies, the F1 scores were low. The model was seemingly too cautious, in that it took a lot of pixels around the structures of relevant classes which were background and labelled them as the relevant class. Moreover, the model also predicted some structures as neurons of interest, when in fact they were labelled as background in the mask. The conclusion was that the high weights were biasing the model towards the two classes, causing a low F1 score.

In order to rectify this behaviour, an experiment was conducted with different reduction factors introduced for the weights related to our neurons of interest, to understand this inherent trade-off. The factor tested are 0.01, 0.05, 0.1, 0.5, 1. All models were trained for only one epoch for time reason. Results are shown in Fig.5. As expected, the F1 score increased as the factor was decreasing while categorical accuracy was decreasing.

2) Mask processing: The initial task was to identify and distinguish the "AIA terminal nodule" from the "AIA central nodule". The network performances for this task were low due to the fact that it was not trained to ignore unwanted parts of the images (i.e. the noise and other cell types). To train the network to exclude unwanted cell types/noise as well, another class was added: pixels that don't look like background (for this example pixels with value over 300) and are not classified as "AIA terminal nodule" or "AIA central nodule". Training for 20 epochs on those masks gave us the results summarized in Tab.VII. It clearly increased the F1 scores while decreasing the accuracy. Nevertheless, this is an interesting trade off to mitigate the missing properties without sacrificing too much accuracy.

	Background	AIA terminal nodule (class 1)	AIA central nodule (class 2)
Categorical accuracy	0.999	0.94	0.92
F1 score	0.999	0.78	0.68

TABLE VII: Accuracy and F1 scores (20 epochs, 4 classes)

3) 4 classes Trial: The next approach was to benefit from the multiples labels provided in our training sets. Thus in the next experiment, four classes were created instead of 3, the fourth being composed of all other neurons (other type then the neurons of interest). This network was trained for 20 epochs. the results are summarized in Tab.VIII. Here again, F1 score generally increased while the categorical accuracy decreased. The mask were different than the ones of the previous experiments because they had sharper boundaries. That could potentially explain the differences in accuracy and F1 scores observed.

4) Depth: Theoretically, increasing the number of layers in a neural network can help improve its performance on the same task, since it allows the possibility to learn more abstract representations for better generalizations[5]. In reality however, depth is often limited by resource constraints, due to increased parameters.

	Background	F1_scores (class 1 & 2)	Accuracy (class 1 & 2)
Categorical Accuracy	0.997	0.895	0.94
F1 score	0.992	0.64	0.37

TABLE VIII: Accuracy and F1 scores (20 epochs, 4 classes)

The experiments in exploring the effect of depth were thus limited by the available memory constraints. However, a significant improvement in segmentation performance was observed with a marginal increase in depth, or more specifically on increasing the number of layers per dense block. The results are summarized in Table IX.

	Dense Block Layers	F1_scores (class 1 & 2)	Accuracy (class 1 & 2)
$Model_1$	3	(0.43, 0.438)	(0.988, 0.97)
$Model_2$	4	(0.556, 0.537)	(0.987, 0.978)

TABLE IX: Accuracy and F1 scores (10 epochs, 3 classes)

C. Generalisation over Samples

According to the results presented in Section IV-B, the network is able to learn well the distinguishing features of the two neurons within one video. Further experiments were conducted in order to check for the generalisation ability across different C.Elegans samples.

1) 3 classes classical model: Categorical accuracy and F1 scores were computed for the network trained for 10 epochs on the initial task. Calculation were performed on 20 random frames out of the 20190805_322_w1_s2 video. The results are presented in Table X.

	Background	AIA terminal nodule (class 1)	AIA central nodule (class 2)
Categorical accuracy	0.993	0.066	0.064
F1 score	0.996	0.013	0.027

TABLE X: Accuracy and F1 scores (10 epochs, 3 classes)

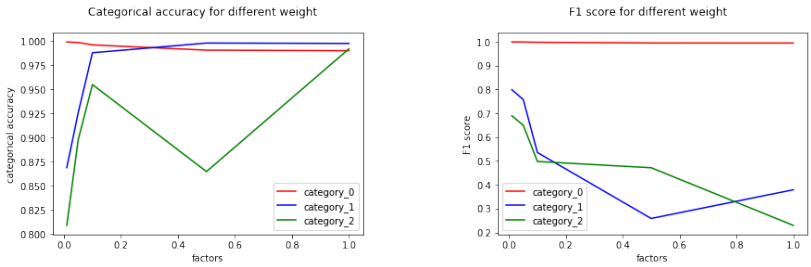
2) Data Augmentation: Data Augmentation is an effective technique to reduce over-fitting on the training set, thus improving the ability of a network to generalize[6]. To explore this idea, a custom data augmentation function was implemented to augment data on the fly during training. The implemented operations were shifts, rotations and zooming. The results were obtained on based on predictions of 20 random frames of the of the 20190805_322_w1_s2 video, and are summarized in the table XI. Much better accuracies are observed as compared to the classical model, while the F_1 scores remain approximately the same, indicating a possibility to utilize this for large scale training.

	Background	AIA terminal nodule (class 1)	AIA central nodule (class 2)
Categorical accuracy	0.975	0.332	0.067
F1 score	0.987	0.017	0.016

TABLE XI: Accuracy and F1 scores (20 epochs, 3 classes)

3) Mask processing: In order to check for generalisation through videos, the performance metrics were computed on the 20190805_322_w1_s2 video frames, with the model the same as the one presented in Section IV-B2. The calculations were performed on 20 random frames.Table XII summarises the results.

4) 4 classes model: The experiment was repeated with the 4 class model. The results are shown in Tab.XIII.



(a) Categorical accuracy (b) Categorical F1 scores

Fig. 5: Change in metrics as a function of reduction factors applied to the relevant class weights

	Background	AIA terminal nodule (class 1)	AIA central nodule (class 2)
Categorical accuracy	0.998	0.008	0.011
F1 score	0.999	0.005	0.007

TABLE XII: Accuracy and F1 scores (20 epochs, 3 classes)

	Background	AIA terminal nodule (class 1)	AIA central nodule (class 2)
Categorical accuracy	0.997	0	0.129
F1 score	0.998	0	0.0185

TABLE XIII: Accuracy and F1 scores (20 epochs, 4 classes)

D. Generalization Over Time

Since the networks performed well on the validation sets for the previous experiments, generalisation over time seems achievable. Additionally the authors of this paper were interested in determining the minimal set of data needed in order to have satisfactory generalisation performance over time. Two experiments were conducted to explore the same. The network was trained on the first 100 pictures and on 100 random frames. Both these tasks would require the same amount of effort for creating the manual annotations, but would offer insight into which may be the better strategy. The results are summarized in Tab.XIV and Tab.XV respectively.

	Background	AIA terminal nodule (class 1)	AIA central nodule (class 2)
Categorical accuracy	0.997	0.6523	0.618
F1 score	0.998	0.443	0.358

TABLE XIV: Accuracy and F1 scores when trained on 100 first frames (25 epochs, 4 classes)

	Background	AIA terminal nodule (class 1)	AIA central nodule (class 2)
Categorical accuracy	0.997	0.8546	0.8847
F1 score	0.998	0.655	0.482

TABLE XV: Accuracy and F1 scores when trained on 100 random frames (25 epochs, 4 classes)

V. Discussion

While the experiments offer a lot of insights, they also open up more avenues to explore. First, the fact that all experiments were conducted on down-sampled versions of the data means that the full resolution of the high quality imaging was not exploited. Another avenue is to explore increasing the size of the convolutional kernels, which although increases training time, could help to capture larger features. Similarly for resources reasons, the network was restricted in depth but for the task presented in this paper, depth might also be helpful as suggested by the results in Section.IV-B4. Of course, there is a need to be wary of the curse of dimensionality when working with 3D Convolutions, as the number of parameters can explode quickly. Additionally, another observation is that despite augmentation, models trained on one video, are not capable of generalizing over the other videos. It is important to exercise caution when drawing conclusions from this, since the sample set is too small. For instance, it may simply be that the variance of C.Elegans brains between different individuals may be too much to capture just from one instance. Results on a larger dataset, coupled with data augmentation may better answer this question.

Lastly, while generalization on samples still needs work, the generalization over sample yields positive results. Moreover, with the same amount of annotator effort, it is possible to achieve good accuracy and F_1 scores on the segmentation task by randomly sampling and annotating a small subset of frames for supervised training of the FCN-Densenet and using this network to generate labels for the remaining frames. All the code is publicly available on Github: https://github.com/LorisPilotto/githubMLRendu.

Acknowledgements

We are extremely grateful to Prof. Sahand Jamal Rahi and all the LPBS team for this project and their helpful suggestions, as well as to Prof. Martin Jaggi and Prof. Rudiger Urbanke for providing this opportunity to work in an interdisciplinary group. We are also thankful to GalDude33 [7] for a clean implementation of the desired network, as well as Google Colab for making free hardware available for training.

References

[1] Simon Jegou, Michal Drozdzal, David Vazquez, Adriana Romero, Yoshua Bengio *The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation.* https://arxiv.org/pdf/1611.09326.pdf, 2017

[2] Özgün Cicek, Ahmed Abdulkadir, Soeren S. Lienkamp, Thomas Brox and Olaf Ronneberger. *3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation*, https://arxiv.org/pdf/1606.06650.pdf, 2016

[3] Krull, Alexander, Tim-Oliver Buchholz, and Florian Jug. *Noise2Void - Learning Denoising from Single Noisy Images.* http://arxiv.org/abs/1811.10980, 2019

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun. *Deep Residual Learning for Image Recognition.* http://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html, 2016

[5] Ronen Eldan, Ohad Shamir *The Power of Depth for Feedforward Neural Networks* https://arxiv.org/abs/1512.03965, 2016

[6] Shorten, Connor, and Taghi M. Khoshgoftaar. *A Survey on Image Data Augmentation for Deep Learning*: Journal of Big Data 6, no. 1 (2019). https://doi.org/10.1186/s40537-019-0197-0

[7] GitHub: *GalDude33/DenseNetFCN-3D* https://github.com/GalDude33/DenseNetFCN-3D