Segmentation of NKX2.5 Signal in Human Pluripotent Stem Cell-Derived Cardiomyocytes

Siem Jongsma[1], Verena Schwach[2], Simone A. Ten Den[2], Robert Passier[2,3], Fons J. Verbeek[1], and Lu Cao[1](✉)

[1] Leiden Institute of Advanced Computer Science, Leiden University, Niels Bohrweg 1, Leiden, The Netherlands
l.cao@liacs.leidenuniv.nl

[2] Applied Stem Cell Technologies, TechMed Centre, University of Twente, Enschede, The Netherlands

[3] Department of Anatomy and Embryology, Leiden University Medical Centre, Leiden, The Netherlands

Abstract. Human pluripotent stem cell-derived Cardiomyocytes (hPSC-CMs) become increasingly popular in recent years for disease modeling and drug screening. NKX2.5 gene is a key transcription factor that regulates cardiomyocyte differentiation. A human embryonic stem cell (hESC) reporter line with NKX2.5 in GFP signal allows us to monitor the specificity and efficiency of human cardiac differentiation. We intend to develop an automatic analysis pipeline for the NKX2.5 signal. However, the NKX2.5 signal captured from fluorescence microscopy is highly heterogeneous. It is not possible to be properly segmented using traditional thresholding methods. Therefore, in this paper, one machine learning method: enhanced Fuzzy C-Means clustering (EnFCM) and two deep learning models: U-Net and DeepLabV3+, are evaluated on the segmentation performance. Parameters have been tuned for each method so as to reach to the optimal segmentation performance. The results show that EnFCM reaches the performance of 0.85. U-Net and DeepLabV3+ have a superior performance. Their performances are 0.86 and 0.89 respectively.

Keywords: pluripotent stem cell derived cardiomyocyte · segmentation · machine learning · deep learning

1 Introduction

Stem cell technology is a rapidly developing field that potentially offers effective treatment for various diseases [11]. It provides a more faithful representation of the actual human diseases so that underlying mechanisms can be better understood. Stem cells can be used for the effective validation of safe medicines which could facilitate more predictive drug discovery and toxicity studies [28]. In addition, stem cells have the potential to replace animal experimentation in predictive toxicology [16].

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2023

C. Anutariya and M. M. Bonsangue (Eds.): DSAI 2023, CCIS 1942, pp. 170–184, 2023.
https://doi.org/10.1007/978-981-99-7969-1_13

Human pluripotent stem cell-derived Cardiomyocytes (hPSC-CMs) are valuable tools for disease modeling, assessing cardiotoxicity of drugs as well as identifying novel therapeutic compounds [19]. In recent years, tremendous efforts have been taken to integrate hPSC-CMs into high-throughput screening systems. Phenotypic analysis has been carried out and varied phenotypic readouts have been quantified including morphological changes [6,7], contractile properties [18,25],calcium transients [22,23] and electrophysiological parameters [20,32].

NKX2.5 is an essential transcription factor for activation and maintenance of the cardiac regulatory network. This transcription factor can be detected in cardiac progenitor cells, their progeny, and mature cardiomyocytes [17]. Therefore, genetically engineered NKX2.5 reporter cell line was developed to monitor cardiac cell populations during differentiation [10]. Recently, the hPSC NKX2.5 reporter line has been used for developmental toxicity testing [17].

Measuring NKX2.5 fluorescent signal from hPSC-CMs in a high-throughput manner is, however, challenging. We observed big variation of the green fluorescent protein (GFP) signal of NKX2.5 between batches and treatments as shown in Fig. 1 and 6. It can be caused by batch variation, differentiation efficiency as well as varied effect of treatments. The performance of traditional thresholding methods are not satisfactory, since a large amount of the methods can only segment part of the signal and cannot capture weak GFP signal in the fluorescent images. In this study, we are going to explore machine learning and deep learning methods for segmentation of NKX2.5 signal in hPSC-CMs.

2 Related Work

Image segmentation is a task in computer science that involves the delineation of regions of interest in an image. Segmentation is often used in medical or biomolecular imaging to automate or facilitate the division or recognition of specific structures in the images that are generated in the research [21].

There are many machine learning-based and deep learning-based methods which give superior performance for segmentation. For example, Fuzzy C-Means clustering is an unsupervised machine learning method which has been used for segmentation. It firstly assigns a degree of "belonging to foreground" for each pixel and sets the cut-off between foreground and background based on the minimization of intra-cluster variance [6].

In addition, a large number of deep learning models are successfully used in the research field of semantic segmentation. Semantic segmentation labels pixels in the image to the corresponding regions. One of the most widely used models in the field is U-Net. U-Net is a fully convolutional neural network that was first proposed in 2015 [27]. It is featured by its light weighted structure which makes it possible to train a deep learning model with a small training dataset such as thousands or even hundreds of images. It is extremely useful in the segmentation of a biological image dataset in which the number of training data is always limited.

There are several deep learning models which are commonly used as alternatives for U-Net [13] such as DeepLabV3+ [8] and Tiramisu [2]. From two studies in which DeepLabV3+ is used for similar segmentation tasks and directly compared to U-Net, it is concluded that DeepLabV3+ achieves slightly better performance than U-Net [9,14]. It is also found that U-Net does have a slightly better performance than DeepLabV3, which is the previous version of DeepLabV3+ [1]. In addition, it has been shown that DeepLabV3+ has a better performance than DeepLabV3 [31]. From two studies in which the performance of U-Net is compared to Tiramisu, it is concluded that Tiramisu has similar performance [12,15].

3 Methods

3.1 Preparation of the Cells

Double Reporter mRubyII-ACTN2 and GFP-NKX2.5 (DRRAGN) hPSCs were differentiated to hPSC-CMs as described in [26]. Around day 14 of differentiation, cells were dissociated and were FACS sorted for α-ActininmRybyII/w-Nkx2.5eGFP/w. Double positive CMs were seeded into 96 well special optics plates (PerkinElmer) at a density of 50,000 cells per well. The hPSC-CMs were maintained in a humidified incubator and were refreshed with CM-TDI medium twice a week [3]. 10–12 days after seeding, the hPSC-CMs were treated with dimethylsulfoxide (DMSO 4.23 mM) as control or with 1 μM of the anticancer drug Doxorubicin for 5 days.

3.2 Imaging

Images of NKX2.5 signal of hPSC-CMs were acquired using the high-throughput automated EVOS FL Auto 2 (Thermo Fisher) microscope equipped with a 40x Super-apochromat Olympus objective (NA 0.95) (Thermo Fisher, AMEP4754). The whole monolayer cell culture was scanned by automatically acquiring 55 images per well every 24 h for 5 days. During the 5 days, cells were maintained on the EVOS Onstage incubator.

3.3 Preparation of Ground Truth Data

The dataset, that is used in this study, consists of 1450 images from the hPSC-CMs research. The 1450 images are from 18 different batches. The size of the images is 1328 × 1048 and they are provided in a 8-bit gray scale format. The signal that is present in the images originates from the GFP signal representing the expression of NKX2.5 protein.

In order to train the deep learning models using these images, ground truths have to be created first. The ground truths, which is approved by the domain specialist, have to be manually created from the original images by converting it to a binary mask using the correct gray value threshold. This is done in the Fiji application [29] using a macro. By running the macro, the images in the selected directory are loaded one by one. A Gaussian blur is applied to suppress the noise in the image. Then, a threshold can be selected manually from which the binary mask is generated and saved. In Fig. 1, two examples are given for the original image and corresponding ground truth. In order to have a consistent set of ground truths, the annotator discussed with the biologists beforehand and did several trials together with the domain expert for quality control. Subsequently, all images were processed and were ready to feed into deep learning models for training.

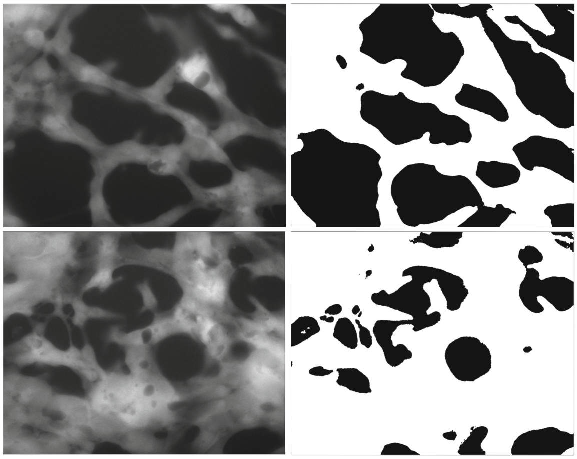


Fig. 1. Two examples of images from the dataset including the manually created ground truth binary mask.

3.4 Evaluation

An evaluation measure is used to assess the performance of the deep learning models. The metric that is used is Intersection over Union (IoU). For the calculation of the IoU metric, the intersection of the foregrounds of a ground truth image and a segmentation result is divided by the union of the same foregrounds, which can be seen in Fig. 2.

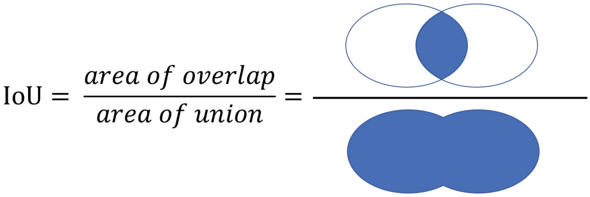


Fig. 2. Visual representation of Intersection over Union performance metric.

The implementation of the IoU metric is done using the MeanIoU function in the Python Keras library.

3.5 Fuzzy C-Means Clustering

The Fuzzy C-Means Clustering (FCM) is an unsupervised clustering method setting the threshold based on the minimization of intra-cluster variance. This method can successfully capture both strong and weak signals which is ideal to segment NKX2.5 signal. However, due to the high resolution of our images (1328 × 1048 pixels per image), it takes minutes (in a system with 2.80GHz processing speed and 8 GB RAM) to extract the signal from a single image. It is not optimal for a high throughput setup. In order to improve the processing speed, several improved versions based on FCM have been explored [5]. An accelerated version of the FCM Algorithm called EnFCM [30] was chosen to solve the speed problem. EnFCM treats each gray value from the histogram as a clustering candidate instead of each pixel from the image. Its energy function for minimization is expressed in Eq. 1:

$$J = \sum_{i=1}^{c} \sum_{l=1}^{q} n_l u_{il}^m \left\| l - v_i \right\|^2 \quad m > 1. \tag{1}$$

where c stands for the number of clusters, q represents the number of gray levels in the histogram, n_l is the number of pixels whose gray value equals to l. m is the fuzzyfication parameter. u_{il}^m is the degree of membership of gray level l. v_i is the center of the cluster. The iterative minimization of the objective function is realized by updating the membership u_{il} and the cluster centers v_i:

$$u_{il} = \frac{(v_i - l)^{-2/(m-1)}}{\sum_{j=1}^{c} (v_j - l)^{-2/(m-1)}} \quad \forall i = 1...c, \quad \forall l = 1....q,$$

$$v_i = \frac{\sum_{l=1}^{q} n_l u_{il}^m l}{\sum_{l=1}^{q} n_l u_{il}^m} \quad \forall i = 1...c.$$

In this way, the processing time is drastically reduced to 1–2 s per image using the same PC.

3.6 U-Net

The U-Net is a model that is built upon the architecture of the fully convolutional neural network [27]. The network consists of two parts: the contracting path and the expansive path. In the contracting path, the usual structure of a convolutional network is followed. This means that unpadded convolutions are applied, followed by a generic ReLU activation operator and a max pooling operation. Max pooling operations are applied for the downsampling of the image. In the expansive path, upsampling is performed first for every layer. Afterwards, up-convolution is done, followed by concatenation with the corresponding feature map that was acquired in the contracting path. As the last step, for every layer in the expansive path, ReLU activation is performed. The output is created by applying a sigmoid activation to acquire the correct range of values in the prediction.

3.7 DeepLabV3+

DeepLabV3+ (DLV3+) is a model that was proposed as an extension of the DeepLabV3 model. The structure of DLV3+ is similar to that of the U-Net. However, there are some important differences between the structure of U-Net and DLV3+. In DLV3+ a combination of atrous convolution and depthwise convolution is used, called atrous separable convolution. In this type of convolution the computational complexity is reduced, while still capturing multi-scale information [8]. ResNet is used as the backbone of the model for feature extraction in the encoder part [24]. Between the encoder and the decoder part, atrous spatial pyramid pooling is performed. This attempts to handle different object scales of a class in the image for better accuracy. The decoder part is comparable to the expansive path of the U-Net where the image is restored to its original size. The number of parameters of DLV3+ is considerably larger than for U-Net, so it is computationally more expensive to run.

3.8 Implementation and Experiments

The EnFCM is implemented as a Java plugin in ImageJ software [6]. We set the fuzzyfication parameter to 2. The parameter that we tuned is the prior probability of assigning pixels to foreground. We observed that the set [0.6, 0.7, 0.8] fits best to the actual signal coverage the best. A prior probability of 0.6 means that the chance of assigning the pixels to foreground is 60%.

The U-Net model is implemented in Python 3 using the Tensorflow Keras library. The implementation is based on the U-Net coding tutorial on GitHub (https://github.com/decouples/Unet/blob/master/unet.py). The platform Google Colaboratory (Colab) [4] is used to run the model and experiments. Colab has computational resources which can be used for running the code in a Python notebook on an online server. There are several hyperparameters which can be tuned in the U-Net model. An overview of the hyperparameters are given in Table 1. In addition, Xavier uniform initializer is used to initialize the weights in the layers. Adam optimization is used with a learning rate of 0.001.

Table 1. The parameters that are studied for the U-Net.

Parameter	Range that is studied
Number of images	[100, 200, 400, 500, 1000]
Number of epochs	[5, 10, 20, 25, 40]
Number of filter layers	[5, 8, 12, 16]
Batch size	[10, 20]

The set of images used is split into a training set and validation set by a ratio of 80/20 respectively. Several combinations of parameter values are tested to see which parameters have an effect on the performance of the model. Structured experiments are done for 100, 200 and 400 input images and an increasing number of training epochs. These small subsets were created from the total 1450 images by combining sets of images from all the different batches. Subsequently, the model is tested for 1000 input images.

The DLV3+ model is implemented in Python 3 using the Tensorflow Keras library, based on the implementation by Soumik Rakshit [24]. The model is adjusted to work for the binary segmentation task of this project. A pre-trained ResNet50 model (pre-trained on ImageNet) is used as backbone for the DLV3+ model for low-level features. Because the ResNet50 model is trained for use with images of size 512 × 512, the input images are resized to be this size. The output image has a size of 512 × 512 as well. This has to be taken into account when comparing the performance results of U-Net and DLV3+. He normal initializer is used to initialize the weights of layers. Again, Colab is used to run the code and carry out the experiments.

The testing procedure is carried out similarly to U-Net. Several combinations of values of parameters, as shown in Table 2, are tested. Evaluation is done using the same set of evaluation images as is used for U-Net for fair comparison between the two models.

Table 2. The parameters that are studied for the DLV3+.

Parameter	Range that is studied
Number of images	[100, 200, 300, 400, 1000]
Number of epochs	[10, 20, 30, 45]
Learning rate	[0.001, 0.01]

4 Results

4.1 EnFCM

In order to make EnFCM comparable to U-Net and DeepLabV3+, we evaluate the performance using the same testing dataset containing 50 images representing the total dataset distribution with ground truth. The three best results is shown in Table 3. As we can observe, when the prior probability is set to 0.70, the mean IoU performance reaches the highest score of 0.85. Reducing or increasing the prior probability does not help improve the segmentation performance. Therefore, 0.70 is the optimal prior probability for our image dataset with NKX2.5 signal.

Table 3. The top 3 prior probabilities that are studied for EnFCM.

Prior Probability	mIoU
0.60	0.822
0.70	0.850
0.80	0.773

4.2 U-Net

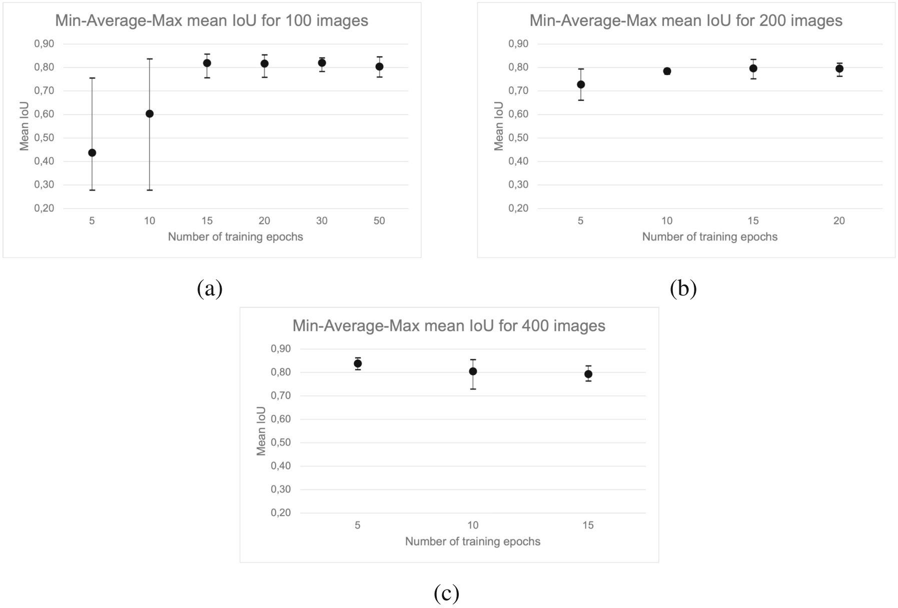
At first, various configurations of the parameter settings for the U-Net were tested. We observed that a larger training set does not necessarily result in a higher performance. An increased value for the number of filter layers resulted in a higher performance, but values for 8, 12 or 16 layers are comparable. A larger number of training epochs results in a higher performance. However, there is a plateau in performance improvement.

Based on the primary results stated above, a more structured experiment was carried out for the U-Net model using image sets of 100, 200 and 400 images to study the effect of the number of images and the number of training epochs. The parameters were kept constant at the following values: filter layers = 8, batch size = 10. The configurations are run three times to obtain an average of the performance.

The mean IoU results for 100 images are shown in Fig. 3(a). The average, lowest and highest values of the three runs are shown in the plot. It can be seen that for 5 and 10 training epochs the performance is very variable. For 15 training epochs and more, the performance is more stable and average mean IoUs above 0.80 are achieved. An increased number of training epochs above 15 does not considerably increase the performance further.

The results for 200 images are shown in Fig. 3(b). For 5 training epochs there is a larger variability in mean IoU values compare to the other numbers of training epochs. The average is also lower for 5 training epochs than for the other numbers of training epochs. The averages for 10, 15 and 20 training epochs differ slightly and the variability between the runs is comparable as well. Overall, an average mean IoU score of around 0.80 is achieved.

The mean IoU results for 400 images are shown in Fig. 3(c). The results of 5 training epochs give the highest performance and smallest variability between the runs. The average performance decreases for increasing numbers of training epochs. For 10 training epochs there is the largest variability between the runs.



(a) (b)

(c)

Fig. 3. Average mean IoU results for U-Net trained with a data set of 100, 200 and 400 images for increasing numbers of training epochs. The average is taken over three runs. The range of mean IoU score for the three runs is given using the bars and connected lines.

When the results for the three different numbers of training images are compared, the performance of the model does not increase when more training images are used. The average performance remains slightly above 0.80. A training set of 1000 images was tested as well to see if using almost all of the available images would improve the performance. However, this gave an average result of 0.794, which is lower than the results gotten from training using a lower number of images.

Final Model. A data set of 400 training images is created from the total 1450 images by combining sets of images from all the different batches. As a result, the created data set is the most representative of all the images that were gathered for this research. This general data set is used to train a final model using the parameter settings that is the most optimal from the previous experiments. The used parameters are: number of training epochs = 10, filter layers = 8, batch size = 10. ReLu is used as activation function. The mean IoU score of this model is 0.860. This model will be used to do a more detailed comparison between the U-Net and DLV3+ performance. The learning curve of the training of this model is shown in Fig. 4. In the curve it can be seen that the model has converged for both the training accuracy and the validation accuracy at around the third training epoch.

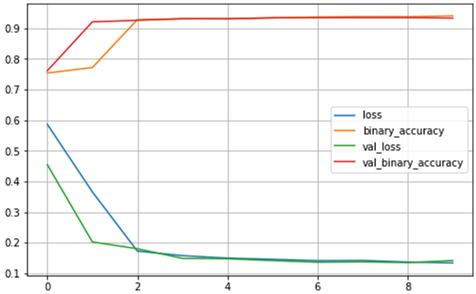


Fig. 4. Learning curve for the U-Net model. The model is trained for 10 epochs using a training set of 400 images.

4.3 DeepLabV3+

The DeepLabV3+ model is tested on performance for several configurations. First small sets of 100 and 200 images were used for the training with a small number of training epochs. This resulted in low performance scores. Then, the model was trained using 1000 images to determine if this improved the performance. It achieved a performance of above 0.75 when trained using 1000 images for 30 total training epochs. This value increased to 0.873 when the model was trained for 45 epochs. When the model was trained again for 45 epochs, but with a smaller number of training images (300 and 400), similar performance is achieved. The results indicate that if the model is trained for a larger number of epochs, a large set of training images is not essential for a high performance. Increasing the learning rate from 0.001 to 0.01 caused the model to converge faster, but the performance is less stable, and therefore lower, than for a learning rate of 0.001.

Final Model. The same general data set of 400 images is used to train a final model for DLV3+ as for U-Net. The model was created using the following parameter settings: number of training epochs = 45, batch size = 10, filter layers = 8 and learning rate = 0.001. The mean IoU score of this model is 0.890. This model will be used to do a more detailed comparison between the U-Net and DLV3+ performance. The learning curve of the training of this model is shown in Fig. 5. In the curve it can be seen that the validation loss is relatively high at the start of the training. It takes more than 30 epochs for the validation loss to decrease to a low value. The validation accuracy remains relatively low compared to the training accuracy for the first 33 epochs. This could indicate that the model is over-trained on the training data. However, after 33 epochs the validation accuracy increases to the same level as the training accuracy, which indicates that the model is generalized and is able to do proper predictions.

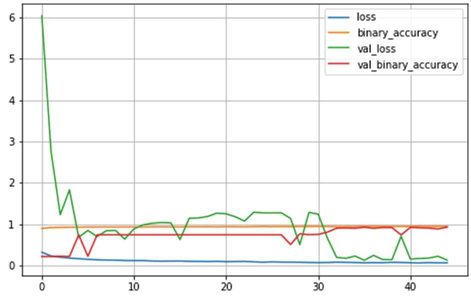


Fig. 5. Learning curve for the DLV3+ model. The model is trained for 45 epochs using a training set of 400 images.

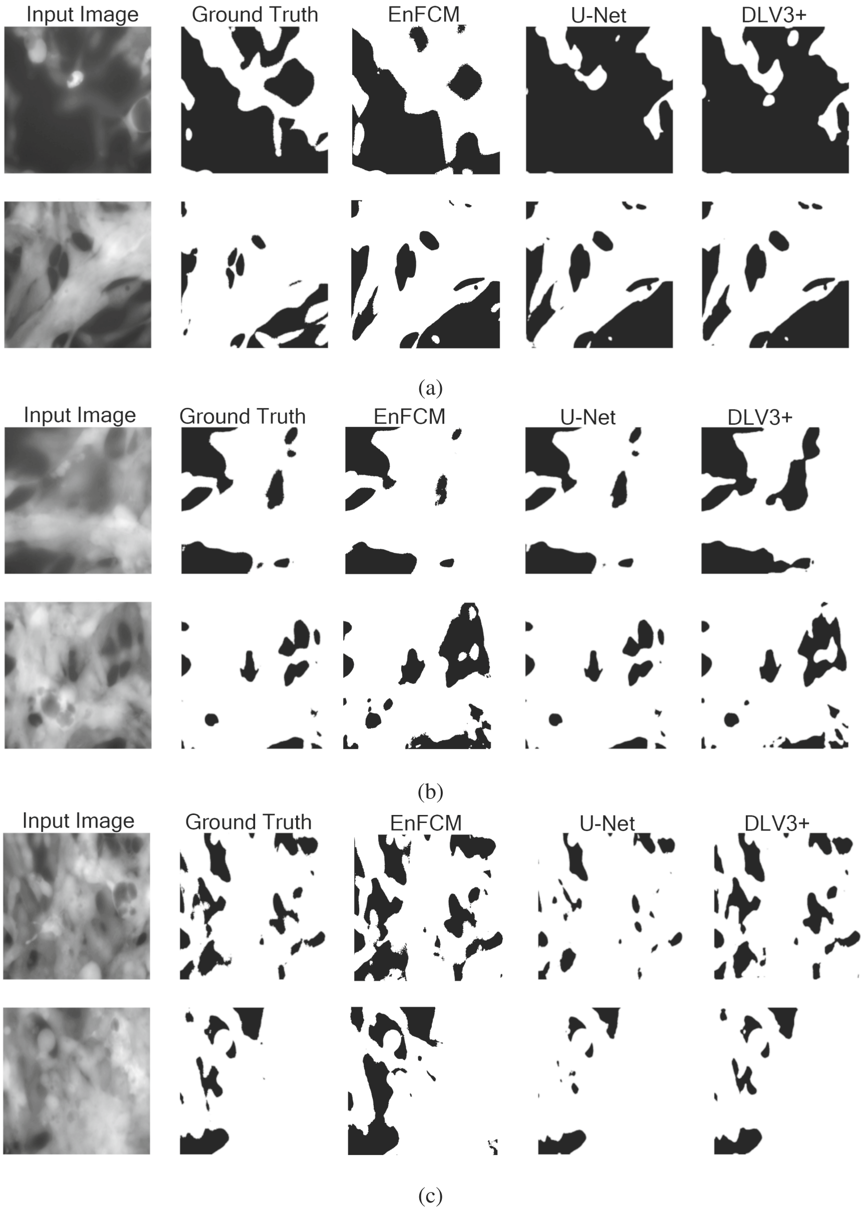
4.4 Comparison

The prediction results on the evaluation set for three methods are visualized (mIoU EnFCM = 0.850, U-Net = 0.860, mIoU DLV3+ = 0.890). Two examples are shown in Fig. 6(a). In these examples it can be seen that all three models did not give an accurate prediction when compared to the ground truth. EnFCM predicted too much foreground signal and both deep learning models predicted too much background instead of foreground signal. In Fig. 6(b), two examples are shown in which U-Net predicted more accurately when compared to the ground truth. In both examples the prediction by U-Net is very similar to the ground truth, but the prediction of EnFCM and DLV3+ contain more background region. In Fig. 6(c), two examples are shown in which DLV3+ predicted more accurately. In both examples DLV3+ showed a more correct amount of background signal compared to the EnFCM and U-Net predictions.

5 Discussion and Conclusion

In order to find a best method for segmentation of NKX2.5 signal for hPSC-CMs in a high throughput setup, three methods, including one machine learning method and two deep learning models, have been implemented, trained and evaluated. The first method that was evaluated, is EnFCM. It segments the NKX2.5 signal with a reasonable mean IoU of 0.85 when the prior probability parameter is set to 0.70. The speed of processing, which is within 1–2 s, is preferred for a high throughput setup. However, the segmentation performance is relatively lower than the other two deep learning models.

The second model is U-net. Several parameters were tuned for the U-Net model. The results showed that the U-Net model does not need a high number of training images and does not need to be trained for a high number of epochs to achieve a good performance, which was also stated in previous research [27]. A higher number of filter layers seems to improve the performance and the speed of convergence of the model slightly based on the number of training images. However, the model rapidly increases in size when the filter layers are increased, which causes the model to be very computationally expensive to run.



(a)

(b)

(c)

Fig. 6. Example prediction results.

Based on the results for the DeeplabV3+ model (Sect. 4.3), it can be concluded that the model is less efficient than the U-Net model. From the learning curves (Fig. 4 & 5) it can be seen that DLV3+ needs more training epochs to converge. The size of the DLV3+ model is also considerably larger than the size of the U-Net model (approx. 10 million parameters versus approx. 0.5 million parameters), which causes the model to be much less computationally efficient to train. Because of the ResNet50 backbone that is used for the implementation of the model in the encoder part, the sizes of the input and output images are set to 512 × 512. This limits the resolution of the output binary masks. This limitation could be solved by using a backbone that is trained using images of a larger size, but these were not available in the Keras library used at the time of the experiment. Increasing the learning rate in the DLV3+ model results in faster convergence of the model. The resulting model, however, is less stable and has, on average, a lower performance.

In Sect. 4.4, the prediction results of the three methods are visualized and compared. Based on visual inspection of the results, we observed that the predictions by U-Net are more accurate compare to the predictions by DLV3+ with respect to the ground truths. This is contradictory to the mean IoU performance scores of the models. This could be caused by that the images for the DLV3+ model are a smaller size and therefore the mean IoU score could give a slightly inaccurate indication when used to compare the models. Overall, we concluded that EnFCM predicts sometimes too much foreground signal and sometimes too much background region. It is due to the fact that there is a constant prior probability setting. In this study, we used 0.70. If the NKX2.5 signal in the image is less than 0.70, more foreground signal would be captured. If the signal in the image is more than 0.70, less foreground signal would be detected. In addition, we observed that DLV3+ predicts too much background signal. For the examples in which DLV3+ seemed to be more accurate, U-Net and EnFCM predicted too less background signal. The two examples, for which all three methods did not have an accurate prediction (Fig. 6(a)), have low NKX2.5 signal in the original image. This signal was not picked up, which could be explained by the large variance in the intensity of the signal in the image. However, for most images in the evaluation set the methods are able to distinguish most of the signal, so these are exceptions to the overall performance.

In conclusion, EnFCM provides reasonable predictions using the original size with a fast processing speed. the U-Net model is able to do the segmentation of images with a size of 1024 × 1024. The mean IoU performance of the U-Net is around 0.860. The model can converge to this score by training for at least 10 epochs using a train set of 100–400 images (batch size = 10 and filter layers = 8). The DeepLabV3+ model is able to do the segmentation of images with a size of 512×512. The mean IoU performance of this model is around 0.890. Convergence of this score can be achieved using a train set of 400 images for at least 35 epochs. The DeepLabV3+ model is computationally more expensive, needs more training epochs to converge and operates on images with a lower resolution. Adaptive learning rate could be tested in the future for faster convergence. Based on visual inspection, the prediction seems to be less accurate than U-Net.

This study has resulted in finding and validating a plausible segmentation method that can be integrated in the high throughput image analysis pipeline; i.e. U-Net. It enables automated monitoring of differentiation efficiency of hPSC-

CMs and facilitates screening of drugs for the toxicity and safety study. In the future, this work will be included in a high throughput analysis of phenotypical readouts for hPSC-CMs. The image-based phenotypical readouts can be combined with other high-throughput assays using functional and biochemical parameters to form a unique fingerprint for each drug under testing using hPSC-CMs as a cell model. It will further facilitate toxicity/safety screening and disease modeling, as well as drug discovery.

Acknowledgements. This work was supported by ERA-CVD 2016T092 and the Dutch Heart Foundation.

References

1. Abdollahi, A., Pradhan, B.: Integrating semantic edges and segmentation information for building extraction from aerial images using UNet. Mach. Learn. Appl. **6**, 100194 (2021)
2. Baghdadi, R., et al.: Tiramisu: a polyhedral compiler for dense and sparse deep learning (2020)
3. Birket, M.J., et al.: Expansion and patterning of cardiovascular progenitors derived from human pluripotent stem cells. Nat. Biotechnol. **33**(9), 970–979 (2015)
4. Bisong, E.: Google colaboratory. In: Building Machine Learning and Deep Learning Models on Google Cloud Platform, pp. 59–64. Apress (2019)
5. Cai, W., Chen, S., Zhang, D.: Fast and robust Fuzzy C-Means clustering algorithms incorporating local information for image segmentation. Pattern Recogn. **40**(3), 825–838 (2007)
6. Cao, L., van der Meer, A.D., Verbeek, F.J., Passier, R.: Automated image analysis system for studying cardiotoxicity in human pluripotent stem cell-derived cardiomyocytes. BMC Bioinform. **21**(1) (2020)
7. Cao, L., Schoenmaker, L., Ten Den, S.A., Passier, R., Schwach, V., Verbeek, F.J.: Automated sarcomere structure analysis for studying cardiotoxicity in human pluripotent stem cell-derived cardiomyocytes. Microscopy Microanal. **29**(1), 254–264 (2022)
8. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation (2018)
9. Dhingra, N., Chogovadze, G., Kunz, A.: Border-segGCN: improving semantic segmentation by refining the border outline using graph convolutional network. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 865–875 (2021)
10. Elliott, D.A., et al.: NKX2-5egfp/w hESCs for isolation of human cardiac progenitors and cardiomyocytes. Nat. Methods **8**(12), 1037–1040 (2011)
11. Fontes, P.A., Thomson, A.W.: Stem cell technology. BMJ **319**(7220), 1308 (1999)
12. Guzder-Williams, B.P.: Fully convolutional networks for landcover classification and landcover change. In: AGU Fall Meeting Abstracts, vol. 2018, pp. H34B–01 (2018)
13. Innolitics, Reinhold, J., Shrestha, Y.: How to Choose a Neural Net Architecture for Medical Image Segmentation (2020)
14. Khan, Z., Yahya, N., Alsaih, K., Ali, S.S.A., Meriaudeau, F.: Evaluation of deep neural networks for semantic segmentation of prostate in T2W MRI. Sensors **20**(11), 3183 (2020)

15. Kholiavchenko, M., et al.: Contour-aware multi-label chest X-ray organ segmentation. Int. J. Comput. Assist. Radiol. Surg. **15**(3), 425–436 (2020)
16. Kim, T.W., Che, J.H., Yun, J.W.: Use of stem cells as alternative methods to animal experimentation in predictive toxicology. Regul. Toxicol. Pharmacol. **105**, 15–29 (2019)
17. Lauschke, K., et al.: Creating a human-induced pluripotent stem cell-based NKX2.5 reporter gene assay for developmental toxicity testing. Arch. Toxicol. **95**(5), 1659–1670 (2021)
18. Miklas, J.W., Salick, M.R., Kim, D.H.: High-throughput contractility assay for human stem cell-derived cardiomyocytes. Circ. Res. **124**(8), 1146–1148 (2019)
19. Oikonomopoulos, A., Kitani, T., Wu, J.C.: Pluripotent stem cell-derived cardiomyocytes as a platform for cell therapy applications: progress and hurdles for clinical translation. Mol. Ther. **26**(7), 1624–1634 (2018)
20. Paci, M., et al.: All-optical electrophysiology refines populations of in silico human iPSC-CMs for drug evaluation. Biophys. J. **118**(10), 2596–2611 (2020)
21. Pham, D.L., Xu, C., Prince, J.L.: Current methods in medical image segmentation. Annu. Rev. Biomed. Eng. **2**(1), 315–337 (2000)
22. Prajapati, C., Pölönen, R.P., Aalto-Setälä, K.: Simultaneous recordings of action potentials and calcium transients from human induced pluripotent stem cell derived cardiomyocytes. Biology Open (2018)
23. Psaras, Y., et al.: CalTrack: high-throughput automated calcium transient analysis in cardiomyocytes. Circ. Res. **129**(2), 326–341 (2021)
24. Rakshit, K.: Keras documentation: multiclass semantic segmentation using DeepLabV3+ (2021)
25. Ribeiro, A.J.S., et al.: Contractility of single cardiomyocytes differentiated from pluripotent stem cells depends on physiological shape and substrate stiffness. Proc. Natl. Acad. Sci. **112**(41), 12705–12710 (2015)
26. Ribeiro, M.C., et al.: A cardiomyocyte show of force: a fluorescent alpha-actinin reporter line sheds light on human cardiomyocyte contractility versus substrate stiffness. J. Mol. Cell. Cardiol. **141**, 54–64 (2020)
27. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
28. Rubin, L.: Stem cells and drug discovery: the beginning of a new era? Cell **132**(4) (2008)
29. Schindelin, J., et al.: Fiji: an open-source platform for biological-image analysis. Nat. Methods **9**(7), 676–682 (2012)
30. Szilagyi, L., Benyo, Z., Szilagyi, S.M., Adam, H.S.: MR brain image segmentation using an enhanced Fuzzy C-Means algorithm. In: Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE Cat. No. 03CH37439), vol. 1, pp. 724–726 (2003)
31. Tsang, S.H.: Review: DeepLabv3+ - Atrous Separable Convolution (Semantic Segmentation) (2021)
32. Yamamoto, W., et al.: Electrophysiological characteristics of human iPSC-derived cardiomyocytes for the assessment of drug-induced proarrhythmic potential. PLoS ONE **11**(12), e0167348 (2016)