Dennis Schmidt*, Andreas Rausch, Thomas Schanze

Deep learning-based recognition of cell structures in fluorescence microscopy sequences with respect to their morphology on cells infected with Marburg virus

Abstract: The Institute of Virology at the Philipps-Universität Marburg is currently researching possible drugs to combat the Marburg virus. This involves classifying cell structures based on fluoroscopic microscopic image sequences. Conventionally, membranes of cells must be marked for better analysis, which is time consuming. In this work, an approach is presented to identify cell structures in images that are marked for subviral particles. It could be shown that there is a correlation between the distribution of subviral particles in an infected cell and the position of the cell's structures. The segmentation is performed with a "Mask-R-CNN" algorithm, presented in this work. The model (a region-based convolutional neural network) is applied to enable a robust and fast recognition of cell structures. Furthermore, the network architecture is described. The proposed method is tested on data evaluated by experts. The results show a high potential and demonstrate that the method is suitable.

https://doi.org/10.1515/cdbme-2020-3129

1 Introduction

In recent years, deep learning approaches have become very relevant for computer vision. They have made reliable real-time object recognition or face recognition possible, which are used in our daily life's, for example in smartphones or self-driving cars. The flexibility and universality of deep learning (DL) make it a powerful tool. Therefore, the use of its remarkable skills in the analysis of medical or microscopic images seems to be an effective method to optimize, support and accelerate pharmaceutical research [1], [2].

The implementation and application of a deep learning neural network for recognition, classification and labelling of cell structures in fluorescence-based microscopic images is presented in this work. The results of a neural network highly depend on the quality of the training data. The used data frequently suffer from low contrast and a low signal-to-noise ratio (SNR). Thus, in some cases it is even difficult for the human eye to recognize structures of interest. The advantage of machine learning is that it can be trained for nearly any function under almost all circumstances, regardless of whether the images are noisy or whether objects in a category have a large variation in shape or not. Furthermore, the use of a state-of-the-art model in object detection, e.g. "Mask R-CNN", promises good results.

Note: "Mask R-CNN outperforms all existing, single-model entries on every task, including the COCO 2016 challenge winners." [3]

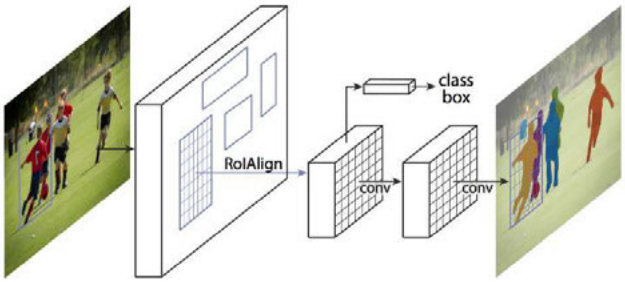

Fig. 1: A simple schematic to visualize the architecture of the Mask R-CNN detector is shown. After being passed through a set of different sub-networks, the output shows the original image, with bounding boxes drawn around the detected objects, as well as masks outlining the shapes of the objects.

* **Corresponding author: Dennis Schmidt,** Institut für Biomedizinische Technik (IBMT), FB Life Science Engineering (LSE), Technische Hochschule Mittelhessen (THM), Gießen, Germany, E-Mail: Dennis.schmidt@lse.thm.de
Andreas Rausch, Thomas Schanze, Institut für Biomedizinische Technik (IBMT), FB Life Science Engineering (LSE), Technische Hochschule Mittelhessen (THM), Gießen, Germany

2 Material and Methods

2.1 Mask R-CNN

In this paper an implementation of a region-based convolutional neural network (Mask R-CNN) is investigated [3]. As a direct successor to Faster R-CNN [4], this neural network is not only capable of classifying and locating objects in images, but also of predicting their shape. While the base architecture of the Mask R-CNN is still the same as that of Faster R-CNN, a new branch has been added that now performs the task of creating masks. In addition to the bounding box and the label, a mask is drawn over the original input image.

The architecture of Mask R-CNN consists of multiple consecutive networks. First a convolutional neural network (CNN) creates a feature map of the input image.

This feature map is scanned by the region proposal network (RPN), which then proposes areas most likely to contain an object. This minimizes the area to be closely examined to relevant areas, which saves valuable computing power and time. These regions are also called region of interest (RoI). They are usually rectangles of various shapes. For the process of classification equally sized squares are needed. In Faster R-CNN this is solved by RoI-pooling. In Mask R-CNN an improved method has been introduced: RoI-Align. While a lot of information about the RoI is lost during RoI-pooling, RoI-Align avoids this through bilinear interpolation.

Note: “These quantizations introduce misalignments between the RoI and the extracted features. While this may not impact classification, which is robust to small translations, it has a large negative effect on predicting pixel-accurate masks.” [3]

The output of RoI-Align can then be used by the fully connected network (FCN), to classify the object and calculate the regression of the bounding box. The regression is performed to correct slightly misplaced boundary boxes.

In the new secondary branch, introduced in [3], the same output is used to create corresponding masks using convolutional layers. An important aspect when choosing a second branch is the decoupling of classification and mask prediction, which in turn increases the accuracy of instance segmentation. For classification, a fully connected layer is used in which the input has to be transformed into a vector. But for a mask the spatial “pixel-to-pixel correspondence” is very important [3]. Therefore, convolutional layers are used.

Finally, a loss function is calculated which indicates how often and to what extent an incorrect prediction was made. By gradient descent, hyperparameters, like weights and filters, are adjusted to minimize the loss. This completes an iteration of the neural network. Then a new iteration with slightly optimized weights and training data is started. This process is repeated until satisfactory results are obtained. Afterwards, the objects can be classified by analyzing the proposed areas of cell structures.

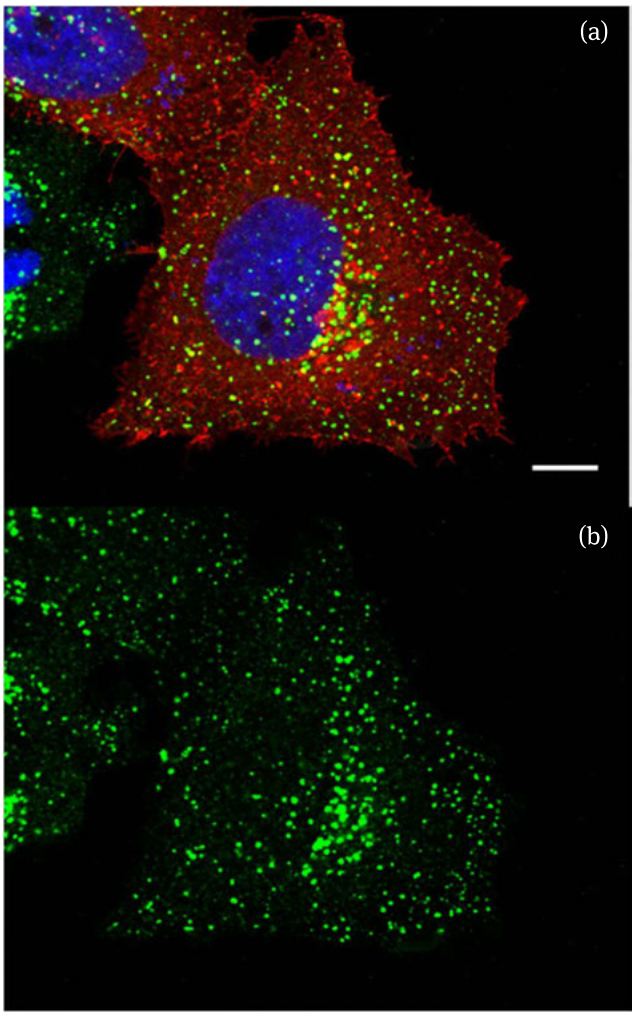

Fig. 2: (a) A fluorescence-based microscopy image of cells infected by the Marburg virus. Chemicals were used to obtain colored, highly detailed images on which the cell structures are clearly visible. Each color channel is related to different chemical that cling to different cell structures. In this picture all three color channels are shown. While the cell membrane is shown in red, the cell nuclei are stained blue and subviral particles in green. (b) Here only the green channel is shown. These images were used for the training, as the target images are also only recorded in one channel.

When the network is used on unclassified images, it recognizes, labels, and draws a mask around objects in the image under the condition that it has been trained on these types of objects. Together with the label, a percentage is given which represents the calculated probability of how correct the detection is. A threshold defines which predictions are displayed. In the case seen in Fig. 3 the minimum confidence was set to 90%. This is the final output of the Mask R-CNN.

2.2 Training

The chosen framework for this implementation was TensorFlow, a machine learning extension for Python [5]. The training data were provided by the Institute for Virology, Philipps-University, Marburg. Using certain chemicals that colorize and thus help to visualize cell structures a lot of information about shapes and arrangements of these cell structures can be obtained. Therefore, the data can be considered validated, as it has been confirmed by experts of the Institute of Virology, Philipps-Universität, Marburg. Nevertheless, the annotation of these images had to be done manually to obtain a ground truth for the algorithm. For that, the VGG annotator software [6] has been used. This tool creates a JSON-file containing all annotations. If this file is placed in the same folder as the training batch, the annotations become readable for the network.

Although colored images were provided as seen in Fig. 2 (a), the training batch was reduced to the images in the green channel (Fig. 2 (b)) for input data and the other channels as target data. This was done because the virologists generally measure only the green channel to evaluate subviral particle behavior. The green dye makes the subviral particles inside the cell visible. Analysis showed that training with grayscale images gave better results than training with colored images. However, the annotations in the green channel were taken over from the other channels. The assumption is, that the distribution of the subviral particles holds information about the cell morphology. To confirm, whether cell regions affect subviral particle behavior, the network only learns to localize cell structures by analyzing the distribution of subviral particles.

In total the training batch consists of 22 images, 2 images were used for validation. The images have gone through data augmentation/surrogation to simulate diversity. This was done to provide the network with a greater variety of images to improve its robustness and flexibility towards low-quality or noisy images. The network has been trained to differentiate between 3 different classes including cell boundary, i.e. cell membrane, nucleus, and background. The training was run for 120 epochs each with 22 steps. The weights of all layers were trained. End of training was achieved after about 30 hours.

3 Results

3.1 Mask R-CNN application to images of Marburgvirus infected cells

In Fig. 3 the results of the “Mask R-CNN” object detection when used on unclassified, microscopic images of infected cells are shown. As seen in (a), all cell membranes have been

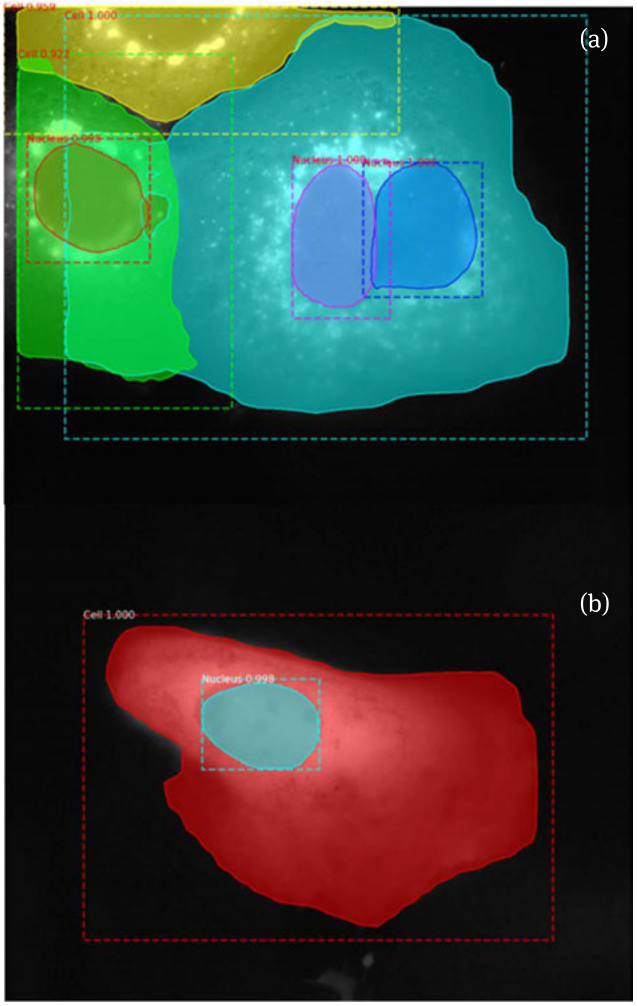

Fig. 3: (a) Visible in this microscopic image are three infected cells and four nuclei. The white spots that accumulate around the nuclei are subviral particles. The dotted lines form the bounding boxes, which show the predicted sizes and locations of the structures. Class names are printed in the upper left corner of the corresponding bounding box. Percentages next to the labels indicate the probability that the prediction is correct. While three nuclei (red, violet, blue) were successfully detected, the fourth nucleus at the upper edge of the image was not detected at all. The cell membranes were detected and covered with green, yellow, and light blue masks. (b) Another analysis result of a microscopic fluorescent image of a single cell. Bounding boxes are shown as blue and red dotted lines. Class labels were assigned correctly. The predicted shapes of the detected objects are displayed as red/blue masks.

detected, although the mask predictions are inaccurate. In (a), the green mask does not cover the cell completely because the bounding box has not been placed in the correct position and a small space towards the left edge of the screen remains empty. The light blue mask is overlapping with parts of the adjacent cell, falsely including it. It can be assumed that a second cell lies behind the first in the middle, so that it appears as if one cell has two nuclei. Nevertheless, all classification labels were assigned correctly. The calculated percentages of probability that the detection was correct (confidence) range from 99.8% to 100% for the nuclei and from 99.2% to 100% for the cell membranes.

In Fig. 3 (b) there was only one cell depicted. The cell as well as its nucleus have been framed with bounding boxes. Also, the correct labels have been given by the neural network. The confidence percentages are 100% for the cell membrane and 99.8% for the nucleus.

4 Discussion

The results of our application of Mask R-CNN, show that it is well suited for the task of automatic object recognition and classification.

In Fig. 3 all cells and nuclei except one were identified and marked. If the network has detected a cell or a nucleus the chosen sizes and locations of the drawn bounding boxes are correct in most cases. Although the training data contained images in which cells and nuclei are only partially visible at the edges of these images, the nucleus at the upper edge of Fig. 3 (a) was not recognized. In contrast, the cell containing this nucleus has been very well marked (yellow). As can be seen in Fig. 3 (a), the network has problems in distinguishing between cells that are directly adjacent or overlapping. They are often recognized as only one large cell. Although there were two cells in front of each other, the network recognized the two nuclei as individual instances. Despite the fact, that some cell structures were not successfully recognized and classified, in no case was the background erroneously identified as a cell structure. The results in Fig. 3 (b) match the manually evaluated data. Furthermore, a significant correlation between the distribution of subviral particles and the placement of cell structures can be observed. Large numbers of subviral particles can be detected in the periphery of the cell nuclei, shown as bright spots in Fig. 2 and Fig. 3 (a). As the network is able to mark the cell structures in unclassified images, although it only has the distribution of subviral particles as information, it seems to have learned the following pattern: If there are large numbers of subviral particles in a small area, there must be a nucleus nearby. Thus, the assumption was supported that the distribution of subviral particles holds information about the morphology of the infected cell. Future work will focus more deeply on the hypothesis that images of subviral particles can be used to infer cell structures such as membranes or cell nuclei, which are not stained in the images used for analysis.

Due to the use of a standard computer CPU for running Mask-R-CNN, the training/learning process of the network of about 30 hours was quite high in comparison with the relatively low number of iterations. As shown in [3], sufficient training usually requires several powerful GPUs and about 160k iterations. Furthermore, a large data set with great diversity (e.g. COCO data set with 80,000 train images and 40,000 validation images [7]) has an immense positive influence on the accuracy of the network. In our case, it was not possible to obtain a data set of this quantity in the medical field. The reason for this is, that qualified annotations in medical images are often time consuming and must be done by experts. Consequently, a more optimal way of using small data sets must be implemented, e.g. cross-validation and bootstrapping. Another approach would be transfer learning, which uses an already trained model with a related task for a second task.

However, the introduced application of a neural network for the automatic recognition of cell structures in fluorescence-based microscopic images can be a reliable tool under the right conditions.

Acknowledgement: The authors would like to thank Stephan Becker, Olga Dolnik and Sandro Halwe at the Institute of Virology, Philipps-Universität Marburg for providing fluorescence images.

Authors statement

Research funding: The author state no funding involved. Conflict of interest: Authors state no conflict of interest. Material and Methods: Informed consent: Informed consent is not applicable. Ethical approval: The conducted research is not related to either human or animals use.

References

[1] Schmidhuber, Jürgen. “Deep learning in neural networks: An overview.” *Neural networks: the official journal of the International Neural Network Society* 61 (2015): 85-117

[2] Voulodimos et al. Deep learning for computervision: A brief review. Comp. Int. and neurosc. (2018).

[3] Kaiming He, Georgia Gkioxari, Piotr Dollár, Ross Girshick. Mask R-CNN. 2017 IEEE International conference on computer vision (ICCV), Venice, 2017, pp. 2980-2988

[4] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. MIT Press, 2015, Cambridge, MA, USA, 91-99

[5] TensorFlow: Large-scale machine learning on heterogeneous distributed systems. Preliminary white paper, 2015.

[6] Abhishek Dutta and Andrew Zisserman. 2019. The VIA Annotation software for images, audio and video. In proceedings of the 27th ACM international conference on multimedia (MM ’19), October 21–25, 2019, Nice, France. ACM, New York, NY, USA, 4 pages.

[7] Lin, T.-Y.; Maire, M.; Belongie, S.; Bourdev, L.; Girshick, R.; Hays, J.; Perona, P.; Ramanan, D.; Zitnick, C. L. & Dollár, P. (2014), 'Microsoft COCO: Common Objects in Context'