DeepCeNS: An end-to-end Pipeline for Cell and Nucleus Segmentation in Microscopic Images

Nabeel Khalid*, Mohsin Munir*, Christoffer Edlund†, Timothy R Jackson‡,
Johan Trygg†§, Rickard Sjögren†§, Andreas Dengel*¶, Sheraz Ahmed*
*German Research Center for Artificial Intelligence (DFKI) GmbH, Kaiserslautern 67663, Germany
firstname.lastname@dfki.de
†Sartorius Corporate Research, Sweden
‡Sartorius, BioAnalytics, Royston, United Kingdom
firstname.lastname@Sartorius.com
§Computational Life Science Cluster (CLiC), Umeå University, Sweden
¶Technische Universität Kaiserslautern, Kaiserslautern 67663, Germany

***Abstract*—With the evolution of deep learning in the past decade, more biomedical related problems that seemed strenuous, are now feasible. The introduction of U-net and Mask R-CNN architectures has paved a way for many object detection and segmentation tasks in numerous applications ranging from security to biomedical applications. In the cell biology domain, light microscopy imaging provides a cheap and accessible source of raw data to study biological phenomena. By leveraging such data and deep learning techniques, human diseases can be easily diagnosed and the process of treatment development can be greatly expedited. In microscopic imaging, accurate segmentation of individual cells is a crucial step to allow better insight into cellular heterogeneity. To address the aforementioned challenges, DeepCeNS is proposed in this paper to detect and segment cells and nucleus in microscopic images. We have used EVICAN2 dataset which contains microscopic images from a variety of microscopes having numerous cell cultures, to evaluate the proposed pipeline. DeepCeNS outperforms EVICAN-MRCNN by a significant margin on the EVICAN2 dataset.**

***Index Terms*—biomedical, healthcare, deep learning, cell segmentation, nucleus segmentation**

I. Introduction

High-throughput microscopy is becoming widely used to study biological phenomena due to its high spatial and temporal resolution, which has led to its wider use in cell biology research and the pharmaceutical industry. Thanks to advances in instrumentation, vast amounts of raw image data can easily be collected which allows computer vision techniques to extract biologically meaningful information at scale to help find new insights. In particular, instance segmentation of individual cells in a microscopic image with many cells in view allows quantification of single cellular features, such as shape or movement patterns, providing rich insight into cellular heterogeneity. Compared to natural images, microscopic images suffer from certain challenges including low contrast as well as irregularly shaped and overlapping cells, making segmentation challenging. Convolutional neural network-based approaches are showing promising results in microscopic image analysis [1] and datasets to train instance segmentation models, such as EVICAN2 [2], are now available to further explore deep learning-based approaches for cell segmentation.

Unlike the MS-COCO dataset [3] which contains images of everyday objects, the objects in the microscopic images are comparatively difficult to segment, especially when cells are densely packed. In our pipeline, we have proposed parameters for features extraction and anchor sizes to detect and segment cell and nucleus with improved precision on the EVICAN2 [2] dataset.

We propose a novel pipeline DeepCeNS – an end-to-end pipeline for Cell and Nucleus Segmentation. The main contribution of this study are as follows:

- An end-to-end cell and nucleus detection pipeline based on Cascade Mask R-CNN [4] with ResNeSt [5] backbone.
- Extensive evaluation of existing and proposed method on EVICAN [2] dataset from different perspectives. Evaluation results show the superiority of the DeepCeNS with a significant margin of 21.8% (average precision at Jaccard index above 50%) in comparison to state-of-the-art (SotA). EVICAN-MRCNN [2].

II. Literature Review

There exists a lot of studies focusing on detection and segmentation of cell and nucleus using traditional computer vision algorithms [6]–[9] and deep learning approaches [2], [4], [10]–[14].

In traditional computer vision approaches, procedures like background subtraction, thresholding, and watershed algorithm, etc. are extensively used for segmentation. Most commonly known online tools which utilize these methods are CellProfiler [15] and FogBank [16]. Hu et al. (2004) [6] proposed an algorithm in which they used improved snake [17], also known as active contour models to segment cell nucleus in microscopic images from esophageal cells. Saha et al. (2016) [9] proposed an algorithm based on Fuzzy C-Means (FCM) clustering to segment nucleus in pap smear images

which is crucial for early diagnosis of cervical cancer.
Deep Learning (DL) approaches for biomedical segmentation have open doors to many applications mainly due to the fact that it requires less domain knowledge as it is completely driven by data [18]. In contrast to traditional computer vision approaches, feature representations are automatically learned by the DL models. U-net was proposed by Ronneberger et al. [12] which outperformed all the other contestants in the ISBI 2015 cell tracking and segmentation challenge despite being trained on less than 50 images. The introduction of U-net elucidated what we can achieve with the fusion of biomedical imaging and Artificial Intelligence (AI). It opened the doors to the biomedical image-based cellular research and after that, other deep learning-based algorithms like DeepCell [13] and Usiigaci [14] were introduced. Another DL approach for automatic nucleus segmentation was presented by Johnson (2018) [10]. Schwendy et al. (2020) [2] proposed an extension of Mask R-CNN [19] named, EVICAN-MRCNN [2]. It was trained on EVICAN2 dataset [2] including more than 3500 images for training. Stringer et al. (2020) [20] proposed a generalist algorithm for cellular segmentation called Cellpose. In that method, training set mask images were transformed to vector flow representation that can be predicted by the neural network. Cellpose was trained on the dataset containing over 70,000 segmented objects.

III. Methodology

Fig. 1 provides a system overview of the proposed pipeline. For cell and nucleus detection and instance segmentation we have a proposed a pipeline, DeepCeNS which is based on Cascade Mask R-CNN [4], Feature Pyramid Network [3], ResNeSt-200 [5] and Deformable Convolution [21]. The proposed pipeline is divided into three blocks.

A. Backbone Network

Feature Pyramid Network (FPN) [3] along with ResNeSt [5] defines the backbone network of our pipeline which extract feature maps from the input image at different scales. FPN uses bottom-up, top-down pathways and lateral connections to combine low resolution highly semantic strong features with high resolution semantically weak features. The bottom-up pathway utilizes ResNeSt-200 to extract features from the input image at different scales. As we go up the convolution layers in the ResNeSt, the semantic value for each layer increases and spatial resolution decreases. The output of each convolution layer of ResNeSt is used in the top-down pathway which constructs higher resolution layers from the semantic rich layer. ResNeSt [5] is composed of a modular split attention block that enables attention across feature-map groups which help different network branches to capture cross-feature interactions and learn diverse representations. These blocks are then stacked according to ResNet-style [22]. Deformable convolution is also being used in the ResNeSt. The deformable convolution suits best for our application in terms of cell deformity [23]. Unlike conventional convolution which operates on a pre-defined grid for an input image, deformable convolution has a deformable grid, so that each point on the grid is moved by a learnable offset. The convolution then operates on these moved grid points, hence the name deformable convolution.

B. Region Proposal Network

After the extraction of multi-scale features from the backbone network, these features are then passed onto a Regional Proposal Network (RPN). The purpose of RPN is to detect regions that contain objects and match them to the ground truth. It does so by generating anchor boxes on the input image. The anchor generator parameter used to detect and segment objects in MS-COCO [3] dataset overlooks most of the small cells and nucleus instances when transferred to this task. Unlike MS-COCO and other commonly used image datasets, the sizes of some cells and nucleus in the EVICAN2 [2] dataset are very small. After extensive experimentation, the anchor sizes and anchor aspect ratios were selected that fit adequately for the task at hand. The details of the anchor sizes and aspect ratios are given in Section VI. The anchors generated are then matched to the ground truth by taking Intersection over Union (IoU) between anchors and ground truth. If IoU is larger than the defined threshold, i.e. 0.7, the anchor is linked to one of the ground truth boxes and assigned to the foreground. If the IoU is greater than 0.3, it is marked as background and ignored otherwise. In this way, a lot of proposals are generated, but only 2000 proposals are selected after applying non-maximum suppression [4].

C. Prediction Head

After the successful generation of proposals, the next block in our pipeline is the prediction head. At the prediction head we have the following 3 inputs:

a. Ground truth boxes
b. Proposal boxes from RPN
c. Feature maps from FPN

We have used Cascade Mask R-CNN [4] as our prediction head which is an extension of Cascade R-CNN [4]. Cascade R-CNN is an object detection architecture that is a multi-stage extension of R-CNN [24]. It addresses the problem of degrading performance with increased IoU thresholds. The output of one stage of the cascade is used as input for training the next stage. In this way, each stage aims to find a good set of close false positives to train the next stage. By adding mask head to the cascade, we get Cascade Mask R-CNN. The Cascade R-CNN contains multiple detection branches unlike Mask R-CNN [25], where we have a segmentation branch in parallel to the detection branch. We have three strategies from the authors [4] on where to add the segmentation branch and how many branches to add. First two strategies suggest adding a single mask prediction head at either the first or the last stage of Cascade R-CNN. The third strategy suggests adding a segmentation head to each stage of the Cascade R-CNN. For our implementation, we are using three-stage Cascade Mask R-CNN with a third strategy where the segmentation branch is added at the final stage of Cascade R-CNN.
Proposal boxes from the RPN are first sampled by using a

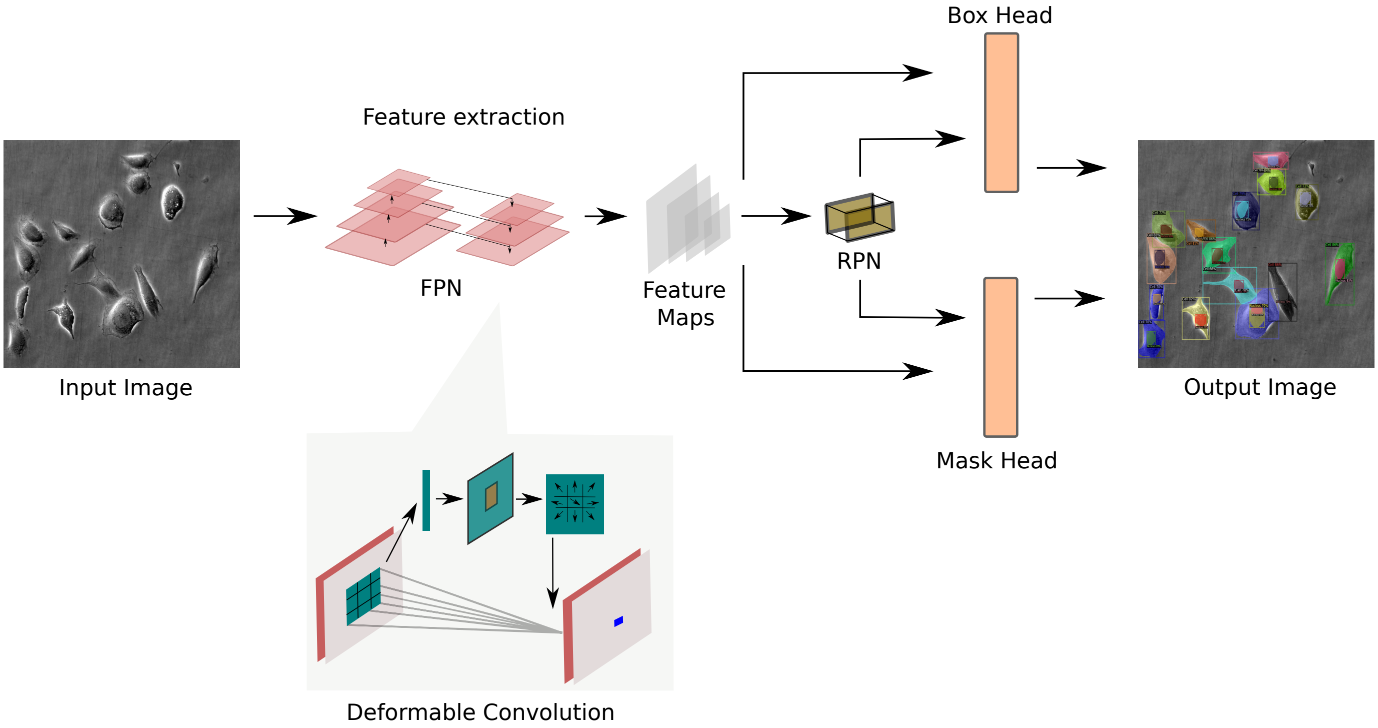


Fig. 1: System overview of DeepCeNS. Input image is passed to the proposed pipeline and the output image with detection and segmentation is produced.

threshold of 0.5 at first stage, 0.6 at the second stage and 0.7 at the third stage of Cascade Mask R-CNN. The proposals which have higher IoU scores than the threshold are regarded as foreground and the rest as background. This process is done during training only. After the sampling of the proposal boxes in each stage, the next step in each stage is ROI pooling in which the rectangle regions of the features maps are cropped as specified by the proposal boxes. After the ROI pooling, the cropped features are fed to the box and mask head of the final stage of Cascade Mask-RCNN. The box head classifies the object within the ROI and fine-tunes the shape and position of the box.

The mask head is composed of a small Fully Convolutional Network (FCN) applied to each ROI which predicts a segmentation mask in a pixel-to-pixel manner [25] to achieve the task of instance segmentation.

IV. Dataset

In this study, we have used EVICAN2 [2] dataset. It consists of images from multiple types of microscopes across 30 different cell cultures. Two variations of this dataset are defined in [2] i.e. EVICAN2 and EVICAN60. In the EVICAN2 dataset, cell and nucleus instances are combined for all the 30 different cell cultures to form two generic classes i.e. cell and nucleus. The EVICAN60 dataset consists of 60 classes (30 classes for cell and nucleus each). In our study, we are using the EVICAN2 dataset. A total of 52,959 instances are present in the 4640 partially annotated images in the training and validation dataset. The test set contains 1057 instances in 98 fully annotated images. Based on the image quality characteristics, the test set is divided into three difficulty levels as done in [2]:

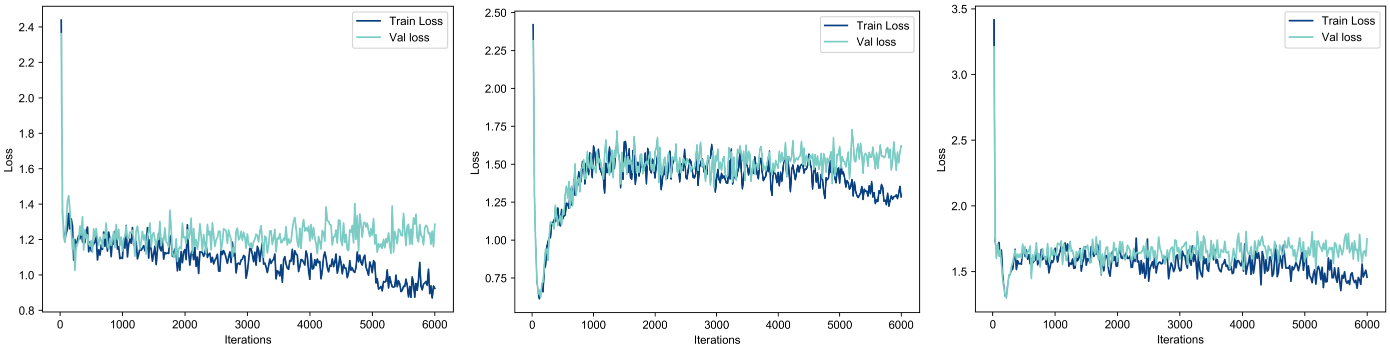
- Difficulty level 1 (Easy): Contains 33 images with 374 instances. 1084 instances are mistakenly reported in [2].
- Difficulty level 2 (Medium): Contains 33 images with 356 instances. 1036 instances are mistakenly reported in [2].
- Difficulty level 3 (Difficult): Contains 32 images with 327 instances. 1102 instances are mistakenly reported in [2].

With regard to heterogeneity, no other dataset like Allen Cell Explorer [26], ISBI [12], Gurari [27], BBBC [28] achieves heterogeneity greater than the EVICAN2 [2] dataset. The annotations were exported as COCO annotation style [3], but only included the segmentation annotation for the instances. For our algorithm development, we calculated the bounding box and pixel area for each instance so we can have a better insight into the detection scores and different area ranges scores alongside the segmentation score. For training and validation, we only took into account the annotated images and discarded the extra background images (750 images in the train set and 250 images in the validation set) with no instances.

V. Evaluation Metrics

To evaluate the performance of the proposed pipeline we are following the standard COCO evaluation protocol [3]. In COCO evaluation metrics, mean average precision is reported at different IoU thresholds and on different area ranges.

Average Precision (AP) is the precision averaged across all



(a) Cell segmentation

(b) Nucleus segmentation

(c) Cell and Nucleus segmentation

Fig. 2: Loss plots for experimental settings. First column (a) represents the training and validation loss of experimental setting 1 (cell segmentation). Second column (b) represents the training and validation loss of experimental setting 2 (nucleus segmentation). Third column (c) represents the training and validation loss of experimental setting 3 (cell and nucleus segmentation).

unique recall levels. Mean Average Precision (mAP) is the mean of average precision across all N classes.

$$mAP = \frac{\sum_{i=1}^{N} AP_i}{N} \tag{1}$$

For the evaluation, we have reported mean average precision for both object detection and segmentation tasks at different IoU thresholds of 0.5 (mAP50), 0.75 (mAP75) and 0.5:0.95 in the steps of 0.05 (mAP). In order to identify performance of model on objects of different sizes, we have also included mAP for different area ranges. Objects with area less than 32^2 (1024 pixel^2) belong to mAPs (small). mAPm (medium) is for the objects in area ranges of 32^2 to 96^2 (9216 pixel^2) and mAPl (large) is for objects with area larger than 96^2.

VI. Experimental Setup

We have compared the DeepCeNS with EVICAN-MRCNN [2] to analyze the performance of the proposed pipeline. DeepCeNS and EVICAN-MRCNN are evaluated in three different experimental settings. Different experimentation manifests the impact and limitations of cell and nucleus detection and segmentation when treated individually and collectively. In the first two settings namely Cell segmentation and Nucleus segmentation, cell and nucleus detection and segmentation are done separately, whereas in the third experimental setting, namely Cell and Nucleus segmentation, cell and nucleus detection and segmentation are carried out collectively. We have retrained EVICAN-MRCNN using their publicly available code[1]. The training for EVICAN-MRCNN and DeepCeNS are done one NVIDIA GeForce GPU and four NVIDIA V-100 GPUs respectively.

We leveraged transfer learning to train DeepCeNS by using MS-COCO pre-trained model [3] for all the settings. The pre-trained model was fine-tuned on EVICAN2 dataset by using Stochastic Gradient Descent (SGD) [29]. Training for all the experiments is performed with a base learning rate of 0.02 and momentum of 0.9. 3x learning rate schedule is used with learning rate reduced by the factor of 10 at 5,000 and then at 5,500 iterations. Linear learning rate warm-up of 0.001 is used to train the first thousand iterations. The anchor sizes for each setting (explained in Hyper-parameter and modeling for each setting) were set after careful consideration of the objects (cell, nucleus) pixel area in the images. Anchors aspect ratios were set to 0.25, 0.5, 1, 2, 4 for all the settings. The checkpoints for evaluation were chosen based on the lowest validation loss and higher validation average precision. Fig. 2 shows the training and validation loss of the DeepCeNS for each setting.

Synchronized batch normalization [30] is used to normalize the images to center all the images in the dataset around zero. For normalizing the input images, the pixel values are subtracted by the global average pixel mean and then divided by the global standard deviation. The pixel mean and pixel standard deviation for the dataset were calculated as 114.83, 114.83, 114.83, and 15.455, 15.455, 15.455 respectively. For data augmentation, images are flipped horizontally on a random basis to reduce the risk of over-fitting. All the images are re-scaled from their original size to (1024, 1024).

A. *Experimental Setting 1: Cell segmentation*

In this experimental setting, the objective is to detect and segment only cells. There are 21,106 cell instances in 3,714 images of the training dataset, 5,317 cell instances in the 926 images of the validation dataset, and 525 cell instances in the 98 images of test set distributed among 3 difficulty levels.

1) Hyper-parameter and Modeling: The anchor sizes for this setting were set to 2, 6, 17, 31, 64, 127, 256, 512, 1024 pixels after analyzing the histogram of cell pixel areas in the train set. The checkpoint at 1,000 iterations was chosen for evaluation.

2) Results: Table I shows the cell detection and segmentation results of EVICAN-MRCNN and DeepCeNS on the

[1] EVICAN-MRCNN code:
https://github.com/MischaSchwendy/EVICAN-MRCNN

TABLE I: Evaluation results of setting 1 (cell segmentation). The best results are shown in bold. DeepCeNS outperforms EVICAN-MRCNN comprehensively in all the metrics at different difficulty levels.

Model	**Difficulty level**	**mAP**		**mAP50**		**mAP75**		**mAPs**		**mAPm**		**mAPl**	
		Det.	**Seg.**	**Det.**	**Seg.**	**Det.**	**Seg.**	**Det.**	**Seg.**	**Det.**	**Seg.**	**Det.**	**Seg.**
EVICAN-MRCNN [2]	**1**	26.48	24.61	55.25	50.03	18.03	21.95	11.91	13.64	28.76	26.66	24.23	22.14
	2	4.28	3.19	11.14	7.63	4.02	2.38	8.56	7.35	1.83	0.82	6.63	5.34
	3	5.92	4.70	14.82	12.89	3.31	1.74	0.00	0.00	6.14	5.03	6.75	5.30
Deep-CeNS	**1**	**64.82**	**63.38**	**88.31**	**89.25**	**73.32**	**74.25**	**41.01**	**43.23**	**67.25**	**63.96**	**64.81**	**66.33**
	2	**32.92**	**29.73**	**53.56**	**54.17**	**37.99**	**29.18**	**18.95**	**18.89**	**32.46**	**26.51**	**38.62**	**37.98**
	3	**23.61**	**21.64**	**40.48**	**40.99**	**24.08**	**24.06**	**0.69**	**0.47**	**26.08**	**21.85**	**24.91**	**26.19**

TABLE II: Evaluation results of setting 2 (nucleus segmentation). The best results are shown in bold. DeepCeNS outperforms EVICAN-MRCNN on all counts.

Model	**Difficulty level**	**mAP**		**mAP50**		**mAP75**		**mAPs**		**mAPm**		**mAPl**	
		Det.	**Seg.**	**Det.**	**Seg.**	**Det.**	**Seg.**	**Det.**	**Seg.**	**Det.**	**Seg.**	**Det.**	**Seg.**
EVICAN-MRCNN [2]	**1**	14.59	13.95	34.53	33.54	7.81	10.61	9.01	8.36	21.76	21.38	0.00	0.00
	2	4.89	4.51	13.36	11.15	1.25	1.52	5.07	4.62	5.33	3.81	5.94	8.17
	3	1.01	1.15	2.68	2.32	0.11	0.99	0.69	0.89	1.72	1.58	0.00	0.00
Deep-CeNS	**1**	**39.12**	**36.46**	**77.05**	**74.43**	**32.56**	**30.56**	**31.79**	**27.81**	**48.56**	**46.44**	**23.20**	**42.53**
	2	**19.48**	**18.46**	**41.08**	**39.60**	**14.44**	**14.44**	**20.37**	**20.11**	**17.76**	**16.52**	**37.67**	**33.74**
	3	**8.87**	**8.86**	**23.57**	**23.28**	**6.62**	**5.36**	**10.13**	**9.46**	**7.88**	**8.95**	**10.64**	**11.56**

EVICAN2 dataset. In terms of segmentation mAP, DeepCeNS outperforms EVICAN-MRCNN by 38.7% for the difficulty level 1. It is observed that both models find it easy to detect and segment cells for the difficulty level 1. There is a significant drop in the scores for difficulty levels 2 and 3.

B. Experimental Setting 2: Nucleus segmentation

In this experimental setting, the objective is to detect and segment the nucleus only. All the nucleus instances from 30 different cell cultures are combined to form one class i.e. Nucleus. There are 21,211 nucleus instances in 3,714 images of the training dataset, 5,325 nucleus instances in the 926 images of the validation dataset, and 525 nucleus instances in the 98 images of test set.

1) Hyper-parameter and Modeling: It was noted that compared to the cell instances, the nucleus instances are relatively smaller in areas. The anchor sizes of 2, 6, 8, 12, 17, 24, 31, 64, 127, 256 pixels were chosen for this experiment after analyzing the pixel areas of nucleus instances in the training dataset. The checkpoint at 2,500 iterations was chosen for evaluation.

2) Results: The results in Table II show that detecting and segmenting the nucleus is difficult as compared to the cell. The detection and segmentation score for DeepCeNS is 39.12% and 36.46% respectively for the difficulty level 1. DeepCeNS outperforms EVICAN-MRCNN in terms of both detection and segmentation tasks for all the three difficulty levels by a vast margin.

C. Experimental Setting 3: Cell and Nucleus segmentation

In this experimental setting, the objective is to detect and segment both cell and nucleus collectively. All the cell and nucleus instances over 30 cell cultures are taken as one class each. 42,317 instances exist in 3,717 images for the train set, 10,642 instances in 926 images for the validation set, and 1,053 instances in 98 images for the test set.

1) Hyper-parameter and Modeling: The anchor sizes were selected by observing the areas of both cell and nucleus instances. It was noted that the anchor sizes used in experimental setting 1 for cell detection and segmentation suit well for this setting because it covers wide ranges of areas and the nucleus instances also exist within that range. The checkpoint at 1,000 iteration was chosen for the evaluation.

2) Results: Table III shows the results of DeepCeNS and EVICAN-MRCNN. mAP50 of EVICAN-MRCNN for the difficulty level 1 is 61.58% and our model outperforms that by more than 20%. In addition to the metrics reported in Table III, we have also included AP for each class as well i.e APC in the Table III represents AP for class cell and APN for class nucleus. DeepCeNS outperforms the EVICAN-MRCNN on all counts.

VII. Analysis and Discussion

In this section, we discuss and compare the results of EVICAN-MRCNN [2] (which is the current SotA at EVICAN2 dataset) and DeepCeNS. In the experimental setting 1 (Cell segmentation), DeepCeNS achieves a superior mAP as compared to SotA for both cell detection and segmentation for the difficulty level 1. The mAP scores also drop with the decline in image quality characteristics, such as increased cell-cell contact (colony formations) and invisible cell outlines, as mentioned in [2] for the difficulty levels 2 and 3. Analyzing the area ranges scores for the experimental setting 1, both implementations find it harder to detect and segment cells of small area as compared to medium and large area cells, but the proposed pipeline performs fairly better in detecting and

TABLE III: Evaluation results of setting 3 (cell and nucleus segmentation). The best results are shown in bold. DeepCeNS performs better than EVICAN-MRCNN for all the difficulty levels.

Model	Difficulty level	mAP		mAP50		mAP75		mAPs		mAPm		mAPl		mAPC		mAPN	
		Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.	Det.	Seg.
EVICAN-MRCNN [2]	1	32.23	32.20	61.29	61.58	30.21	31.70	32.60	31.07	36.85	36.38	21.07	22.86	43.24	43.37	21.21	21.04
	2	14.90	13.57	33.45	30.95	11.64	10.48	18.17	16.75	14.66	13.20	12.94	12.48	17.63	15.61	12.71	11.52
	3	9.91	8.53	20.44	20.79	8.46	4.36	6.22	5.29	11.64	9.45	8.38	7.81	16.35	14.03	3.47	3.02
Deep-CeNS	1	**53.80**	**52.56**	**83.41**	**83.40**	**58.31**	**57.31**	**46.64**	**48.83**	**58.98**	**57.32**	**55.54**	**55.52**	**68.27**	**65.39**	**39.34**	**39.74**
	2	**27.85**	**26.12**	**48.10**	**47.87**	**31.39**	**28.91**	**22.89**	**20.72**	**28.69**	**24.29**	**32.58**	**36.72**	**35.57**	**32.25**	**20.14**	**19.98**
	3	**17.02**	**16.92**	**34.56**	**33.81**	**16.84**	**15.79**	**8.83**	**7.89**	**16.93**	**15.59**	**19.40**	**23.61**	**24.24**	**23.54**	**9.79**	**10.30**

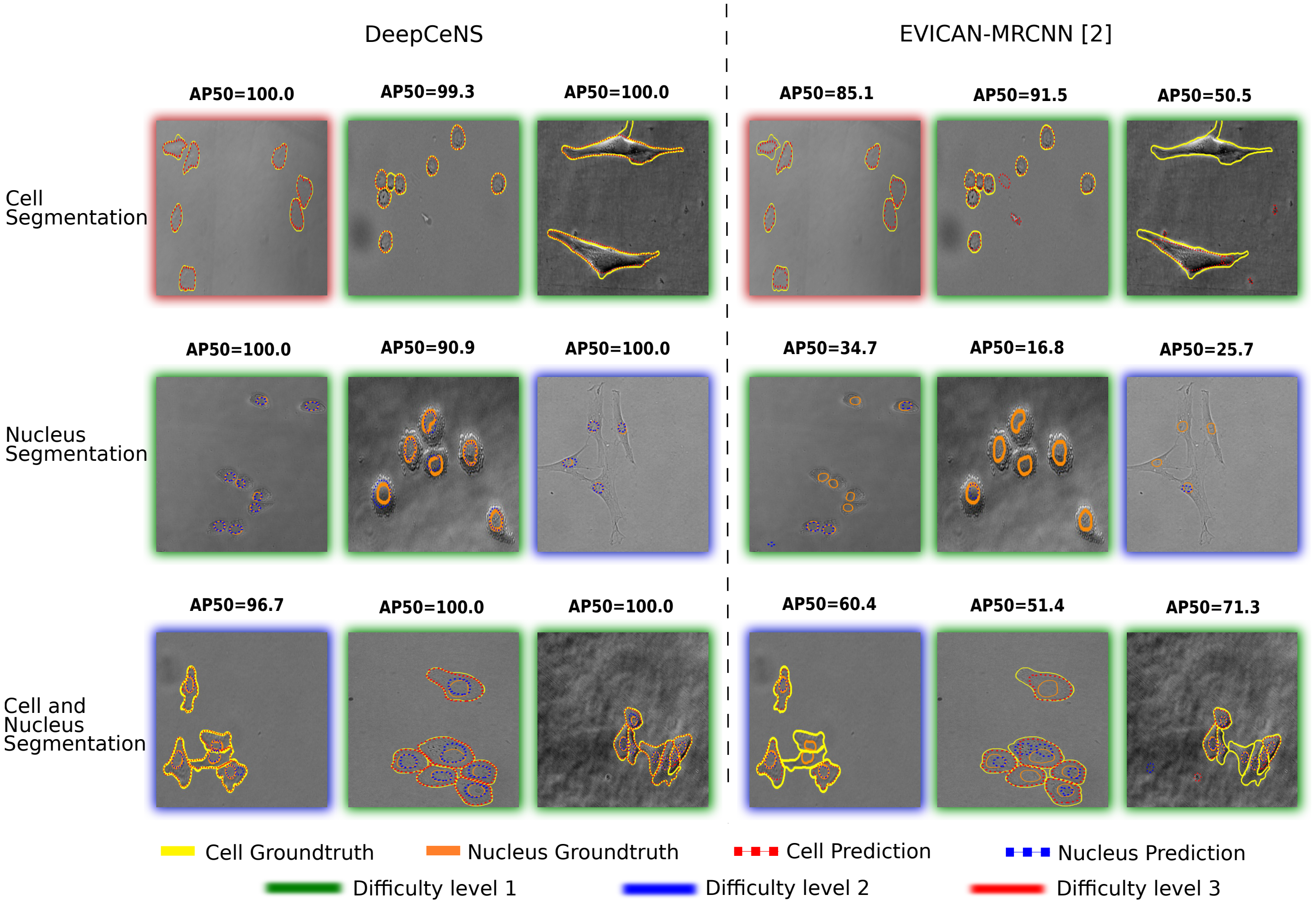


Fig. 3: Inference results of DeepCeNS and EVIVAN-MRCNN (SotA) are shown on sample images. In this figure, we have selected samples on which DeepCeNS performed adequately. The groundtruth masks are shown in solid yellow and orange lines, while the predicted masks are shown in dotted red and blue lines for cell and nucleus respectively. The first three columns show the results of DeepCeNS and the last three columns represent the results of EVICAN-MRCNN. Each row represents different experimental settings. The glow around the image represents to which difficulty level the test image belongs (green, blue and red for difficulty level 1, 2 and 3 respectively).

segmenting cells of small area.

The nucleus detection and segmentation score in experimental setting 2 (Nucleus segmentation) is nearly half of the detection and segmentation score of cell as in experimental setting 1 for both implementations. This could be because unlike cells, there are no sharp boundaries for the nucleus because it is an inner structure of a cell. Similar to experimental setup 1, as we go up the difficulty level, the mAP scores drop to more than half for difficulty level 2 and more than one third for the difficulty level 3 as compared to the difficulty level 1.

For experimental setting 3 (Cell and Nucleus segmentation) Table III, it is evident that the mAP for each category, i.e. cell and nucleus improves when they are trained and tested collectively rather than individually as in experimental setting 1 and 2. The increase in the number of training instances causes the model to learn more diversity which is the reason

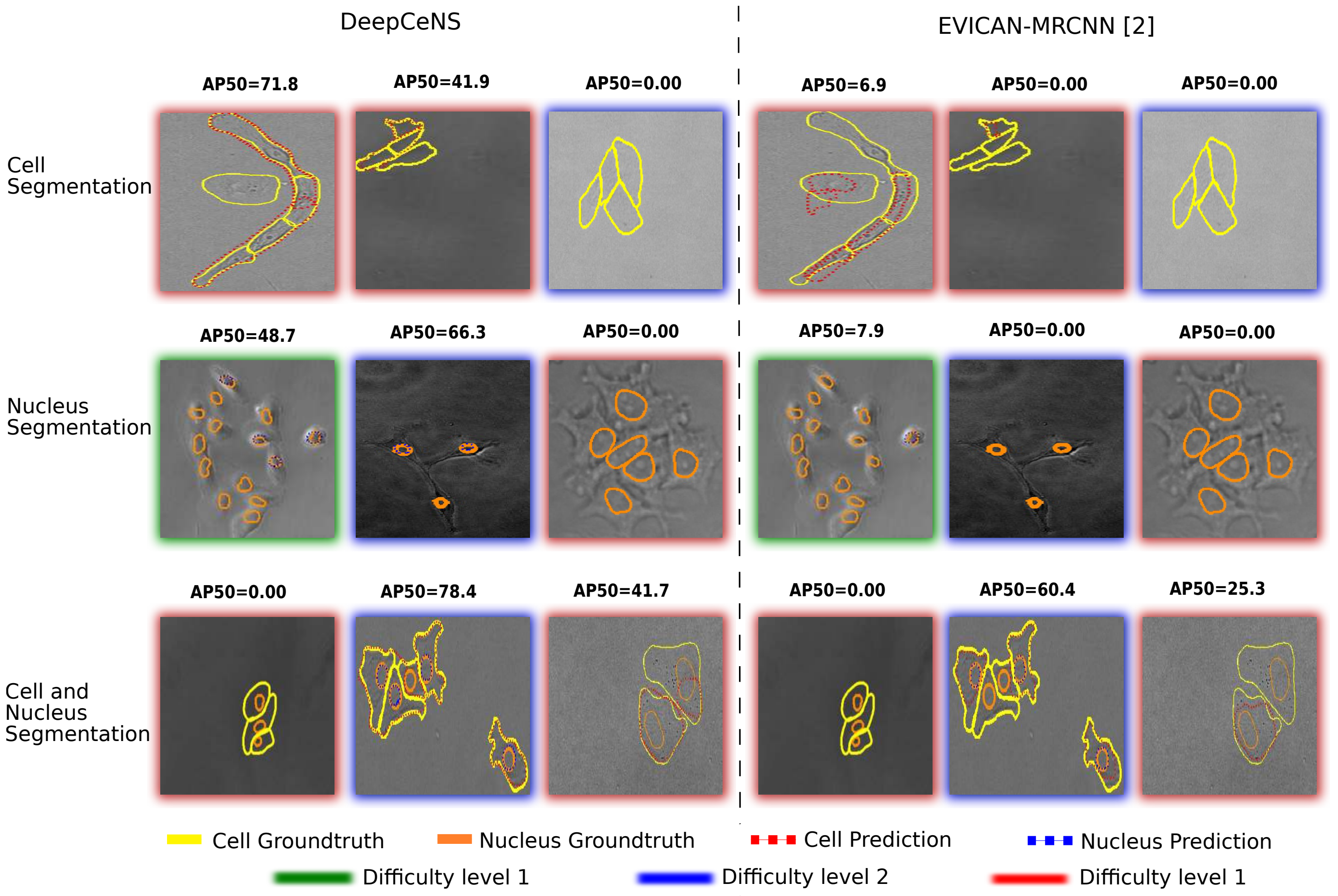


Fig. 4: Inference results of some samples where DeepCeNS and EVICAN-MRCNN (SotA) performed inadequately.

for the improved performance for both classes.

Fig. 3 shows the inference results on some samples where our proposed pipeline performed adequately. For comparison, we have also added EVICAN-MRCNN inference results of the same samples. The first three columns show the results of DeepCeNS and the last three columns represent the results of EVICAN-MRCNN. The groundtruth masks are shown in solid yellow and orange lines, while the predicted masks are shown in dotted red and blue lines for cell and nucleus respectively. The AP50 on top of every sub-image represents the average precision score at IoU threshold of 0.5. The predictions with detection score above 0.5 are shown in the inference results. The first row (cell segmentation) represents the results of experimental setting 1, the second row (nucleus segmentation) represents the results of the experimental setting 2 and the third row (cell and nucleus segmentation) represents the results of the experimental setting 3. The difficulty level of each sample is represented by the glow of green, blue and red for difficulty levels 1,2 and 3 respectively. In these samples, the cell segmentation results of DeepCeNS have nearly 100% mAP50, whereas EVICAN-MRCNN performs relatively poorly with miss detections and false-positive detections. For the nucleus segmentation, the EVICAN-MRCNN fails to predict many nucleus instances, while DeepCeNS accurately predicts most of the nucleus instances. The same trend can be seen for both cell and nucleus segmentation in the third row where DeepCeNS achieves higher scores and segments both cell and nucleus instances, whereas EVICAN-MRCNN fails to detect most of the cell and nucleus instances and there are also some false predictions.

Fig. 4 shows the inference results on some of the samples where both methods perform inadequately. The representation is the same as in Fig. 4. For the cell segmentation, DeepCeNS fails to predict one cell instance in the first image, and no predictions are made for the third image. The performance of EVICAN-MRCNN for cell segmentation is worse as compared to DeepCeNS. Similarly, for the nucleus segmentation, there are many missed predictions, but DeepCeNS performance is relatively better. For the cell and nucleus segmentation, there are some missed cell predictions and most of the nucleus instances are left undetected. The reason for the bad performance is the image quality characteristics. Most of the images where we don't have any predictions either belong to difficulty level 2 or 3. For these difficulty level images, we have many cell-cell contacts where the cell outlines are not visible and the nuclei of the cell are often invisible [2].

With the proposed approach in this study, we have outperformed the EVICAN-MRCNN [2] (SotA) method by a wide margin of 21.80% in terms of mAP at Jaccard index above 50% for segmentation.

VIII. Conclusion

In this study, we have proposed a novel pipeline for cell and nucleus detection and segmentation. DeepCeNS provides an improved method to segment individual cells and nuclei from microscopic images, which helps better quantification of individual cell appearance and behavior. We have achieved state-of-the-art results by outperforming EVICAN-MRCNN [2] with an improvement of 21.80% in terms of mean average precision at Jaccard index above 50% for segmentation. The average precision score for the small instances is still low as compared to large instances. EVICAN2 dataset used for training is partially annotated, where all the cells and nucleus instances are not labeled. Still, our pipeline performs relatively well for detecting and segmenting cell and nucleus instances. The results can further be improved with the availability of a fully annotated dataset. In the future, we would like to investigate a hybrid approach that incorporates traditional computer vision approaches in conjunction with deep learning to make the cell and nucleus segmentation process more robust. Similarly, other augmentation techniques like zooming, rotation, changing brightness, and contrast can be applied for further gain.

Acknowledgment

This work was supported by SAIL (Sartorius Artificial Intelligence Lab) project. We thank all members of the Deep Learning Competence Center at the DFKI for their comments and support.

References

[1] E. Moen, D. Bannon, T. Kudo, W. Graf, M. Covert, and D. Van Valen, "Deep learning for cellular image analysis," *Nature methods*, 2019.

[2] M. Schwendy, R. E. Unger, and S. H. Parekh, "Evican–a balanced dataset for algorithm development in cell and nucleus segmentation," *Bioinformatics*, 2020.

[3] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014.

[4] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

[5] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, Z. Zhang, H. Lin, Y. Sun, T. He, J. Mueller, R. Manmatha *et al.*, "Resnest: Split-attention networks," *arXiv preprint arXiv:2004.08955*, 2020.

[6] M. Hu, X. Ping, and Y. Ding, "Automated cell nucleus segmentation using improved snake," in *2004 International Conference on Image Processing, 2004. ICIP'04*. IEEE, 2004.

[7] N. Lassouaoui and L. Hamami, "Genetic algorithms and multifractal segmentation of cervical cell images," in *Seventh International Symposium on Signal Processing and Its Applications, 2003. Proceedings.* IEEE, 2003.

[8] T. Jiang and F. Yang, "An evolutionary tabu search for cell image segmentation," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 2002.

[9] R. Saha, M. Bajger, and G. Lee, "Spatial shape constrained fuzzy c-means (fcm) clustering for nucleus segmentation in pap smear images," in *2016 international conference on digital image computing: techniques and applications (DICTA)*. IEEE, 2016.

[10] J. W. Johnson, "Adapting mask-rcnn for automatic nucleus segmentation," *arXiv preprint arXiv:1805.00500*, 2018.

[11] Y. Song, L. Zhang, S. Chen, D. Ni, B. Li, Y. Zhou, B. Lei, and T. Wang, "A deep learning based framework for accurate segmentation of cervical cytoplasm and nuclei," in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. IEEE, 2014.

[12] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015.

[13] D. A. Van Valen, T. Kudo, K. M. Lane, D. N. Macklin, N. T. Quach, M. M. DeFelice, I. Maayan, Y. Tanouchi, E. A. Ashley, and M. W. Covert, "Deep learning automates the quantitative analysis of individual cells in live-cell imaging experiments," *PLoS computational biology*, 2016.

[14] H.-F. Tsai, J. Gajda, T. F. Sloan, A. Rares, and A. Q. Shen, "Usiigaci: Instance-aware cell tracking in stain-free phase contrast microscopy enabled by machine learning," *SoftwareX*, 2019.

[15] A. E. Carpenter, T. R. Jones, M. R. Lamprecht, C. Clarke, I. H. Kang, O. Friman, D. A. Guertin, J. H. Chang, R. A. Lindquist, J. Moffat *et al.*, "Cellprofiler: image analysis software for identifying and quantifying cell phenotypes," *Genome biology*, 2006.

[16] J. Chalfoun, M. Majurski, A. Dima, C. Stuelten, A. Peskin, and M. Brady, "Fogbank: a single cell segmentation across multiple cell lines and image modalities," *Bmc Bioinformatics*, 2014.

[17] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active contour models," *International journal of computer vision*, 1988.

[18] N. O'Mahony, S. Campbell, A. Carvalho, S. Harapanahalli, G. V. Hernandez, L. Krpalkova, D. Riordan, and J. Walsh, "Deep learning vs. traditional computer vision," in *Science and Information Conference*. Springer, 2019.

[19] W. Abdulla, "Mask r-cnn for object detection and instance segmentation on keras and tensorflow," https://github.com/matterport/Mask_RCNN, 2017.

[20] C. Stringer, T. Wang, M. Michaelos, and M. Pachitariu, "Cellpose: a generalist algorithm for cellular segmentation," *Nature Methods*, pp. 1–7, 2020.

[21] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2017.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[23] M. Zhang, X. Li, M. Xu, and Q. Li, "Automated semantic segmentation of red blood cells for sickle cell disease," *IEEE Journal of Biomedical and Health Informatics*, 2020.

[24] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014.

[25] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017.

[26] B. Roberts, A. Haupt, A. Tucker, T. Grancharova, J. Arakaki, M. A. Fuqua, A. Nelson, C. Hookway, S. A. Ludmann, I. A. Mueller *et al.*, "Systematic gene tagging using crispr/cas9 in human stem cells to illuminate cell organization," *Molecular biology of the cell*, 2017.

[27] D. Gurari, D. Theriault, M. Sameki, B. Isenberg, T. A. Pham, A. Purwada, P. Solski, M. Walker, C. Zhang, J. Y. Wong *et al.*, "How to collect segmentations for biomedical images? a benchmark evaluating the performance of experts, crowdsourced non-experts, and algorithms," in *2015 IEEE winter conference on applications of computer vision*. IEEE, 2015.

[28] V. Ljosa, K. L. Sokolnicki, and A. E. Carpenter, "Annotated high-throughput microscopy image sets for validation." *Nature methods*, 2012.

[29] R. G. Wijnhoven and P. de With, "Fast training of object detection using stochastic gradient descent," in *2010 20th International Conference on Pattern Recognition*. IEEE, 2010.

[30] H. Zhang, K. Dana, J. Shi, Z. Zhang, X. Wang, A. Tyagi, and A. Agrawal, "Context encoding for semantic segmentation," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018.