MLP Coursework 4: Instance Segmentation of Nucleus Images

Zihan Zhang, Yi Wei, Zhengjun Yue

Abstract

Automatic segmentation of nucleus images is of great significance in medical treatment. In this project, deep models are used to do instance segmentation on nucleus images. The baseline experiments employ fully convolutional networks with the pretrained VGG16 backbone to semantically segment the nuclei and the background, and use open operation to separate individual nuclei. Then Mask R-CNN with the pretrained ResNet-101-FPN backbone is used to segment individual nuclei directly, where the backbone extracts the feature maps, the region proposal network proposes the regions of interest on the feature maps, and the head architecture predicts the mask of nucleus in every regions. The mean average precision of Mask R-CNN reaches 0.401 on the test set, which is much higher than the baseline (0.229). The results imply Mask R-CNN is an advanced method for instance segmentation with high precision and good generalization.

1. Introduction

Since various of illness, from rare disorders to common cold, could be a threat to human's life, the improvement of medical cure has become a great concern among researchers. Cell is the basic component of our body, containing nuclei which are distinctive in images so that can help researchers locate and detect cells. Basically, this leads to advance medical treatment. Hence, identifying nuclei is the first step of most medical analyses, which helps to locate cells in varied conditions to enable faster cure as well as to reduce time-to-market for new drugs. The topic is worthy of studying since it has great practical meaning of our daily life.

The current computational algorithms of nucleus segmentation work well in cases that the nuclei have common shapes and no overlap with each other. However, commonly, the nuclei have different shapes as well as the various of states under different microscope system, leading to the harder work of nucleus detection. Image processing algorithms do not perform well in such cases, and the large number of images and time-consuming work of detection are two main difficulties. The aim of our project is to use deep learning models to automate nucleus detection in different kinds of nucleus images and nuclei.

The main research question we focus on in this project is to increase the precision of instance segmentation for nucleus images with deep learning models, i.e fully convolutional networks (FCN) Shelhamer et al. (2017) as baseline and Mask R-CNN He et al. (2017) as advanced method. Good robustness and generalizability are also required in the task. Through learning the features of nuclei, our model should be able to identify positions of nuclei, given a mask to each nucleus. In other words, although an image could contain many nuclei, the deep models are expected to mask individual nucleus separately. In addition, the expected models should perform well, not only on the familiar images and nuclei, but also on the images from different microscope systems and different kinds of nuclei.

In this report, we firstly introduce the brief procedures of two methods for instance segmentation in section 2. Section 3 then describes the experiments of nucleus image segmentation. Specifically, our dataset, the preprocessing methods and the evaluation criterion are described in section 3.1 and section 3.2. In section 3.3, we briefly introduce the baseline experiments with FCN. Section 3.4 contains the main experiments with Mask R-CNN. In detail, we thoroughly introduce the architecture of Mask R-CNN in the subsection 3.4.1 as well as the experiment configurations in subsection 3.4.2. Then the experiment descriptions, results and analysis are included in the next two subsections. The effectiveness and generalization of the two networks are compared in section 3.5. Furthermore, section 4 introduces some related works on this kind of problems, including the progress and the applications of the corresponding methods in addition to the future work.

2. Methodology

In the section, we introduce several methods that are commonly used for instance segmentation, and they will be applied to our experiments in section 3. In our baseline experiments, we use FCN to semantically segment nuclei and background, and then apply open operation method to separate the masks of individual nuclei in the same images. In the main experiments, we use Mask R-CNN to obtain the mask of each nuclei.

2.1. Fully Convolutional Networks & Open Operation

Fully Convolutional Networks (FCN), proposed by Shelhamer et al. (2017), has a good performance in solving semantic segmentation problems. FCN uses CNN backbone to extract futures of input images, and then predict masks of all nuclei in images with the same resolutions given by deconvolutional layers, instead of fully connected

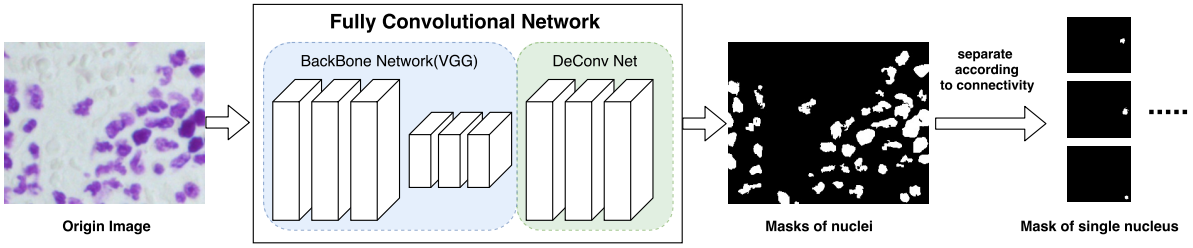


Figure 1. The process of nucleus image segmentation by FCN and open operation

layers in conventional CNN. Thus, FCN could deal with images with different sizes. The number of layers that output the images to deconvolutional layer decides the clarity of the restored outputs. The FCN-8s is chosen in our project since it could generate the finest boundary of the segmentation masks, compared with FCN-32s and FCN-16s. The inputs of FCN are the nucleus images, and the outputs are the corresponding masks that contains all nuclei.

FCN could only segment the foreground (nuclei) and background. However, we need to separate the masks of individual nuclei. Then we use open operation for separating. The main idea of open operation is eroding the edges of the nuclei to shrink their areas while keep their shapes, separating every mask of a single nucleus according to the connectivity, and then dilating the masks at the same degree as erosion. This operation could separate the masks more efficiently than simple separation by detecting connectivity, and keep the shapes and area of each mask. However, it is hard to balance the degree of erosion due to the different shapes of the nuclei.

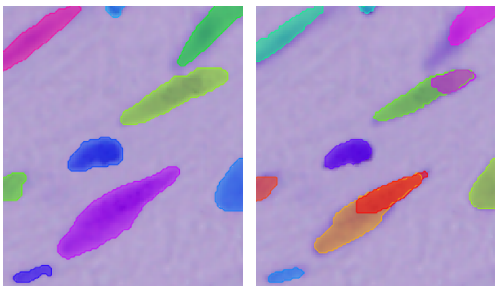
The whole process of FCN is shown in figure 1.

2.2. Mask R-CNN & Overlap Elimination

Mask R-CNN, introduced by He et al. (2017), is an outperformer in instance-level segmentation. It could efficiently detect instances in an image while simultaneously generating a high-quality segmentation mask for each instance.

Firstly, the convolutional backbone architecture of Mask R-CNN extracts the features of entire input images. See the details of the backbone in section 3.4.1. Then, region proposal network (RPN) (Ren et al., 2015) proposes the regions of interest (RoIs) that may contain instances. In this process, RPN generates many candidate anchors, and removes the ones beyond the bounding of images and the ones overlapped much with others. Then RPN adjusts these anchors to generate RoIs through classification and regression. See the details of RPN in section 3.4.4. Then region of interest feature alignment layer (RoIAlign) of Mask R-CNN can reduce different resolutions of RoIs to a fixed size. That is why the input images could have different sizes. Then, these RoIs are used in three tasks, including classification, bounding-box regression and mask prediction. For the first task, several fully connected layers are used to classify these RoIs. In our project, the RoIs are classified to foreground (nuclei) and background with confidence scores. The second task also employs fully connected layers to regress the predicted bounding-boxes of the instances in the RoIs, which are represented by the coordinate of a vertex and the length and width of each box. More importantly, the third branch will predict the masks through FCN. Finally, Mask R-CNN has a multitask loss function, which is the sum of the losses of the three tasks. The multitask loss function makes the model easy to train. The whole process is shown in figure 3.

Since Mask R-CNN predicts masks in the regions of interest (RoI), it is possible that these masks are overlapped, which is not allowed in the competition rules. To deal with the overlaps, we compare the overlapped masks, then keep the biggest one unchanged and subtract the overlaps from the other ones. We apply this strategy due to the evaluation of our task stated in section 3.2. As is shown in figure 2, Mask R-CNN may predicted more than one masks for the same nucleus, and some of them only mask part of the nucleus. This strategy intends to remain the entire masks and drop out the partial masks, and will be used at evaluation on the test set.



(a) ground truth (b) predicted masks

Figure 2. An example of overlapped masks

3. Experiments

In order to segment the nuclei individually in the images, we implemented a series of experiments, including the baseline experiments with FCN and the Mask R-CNN experiments.

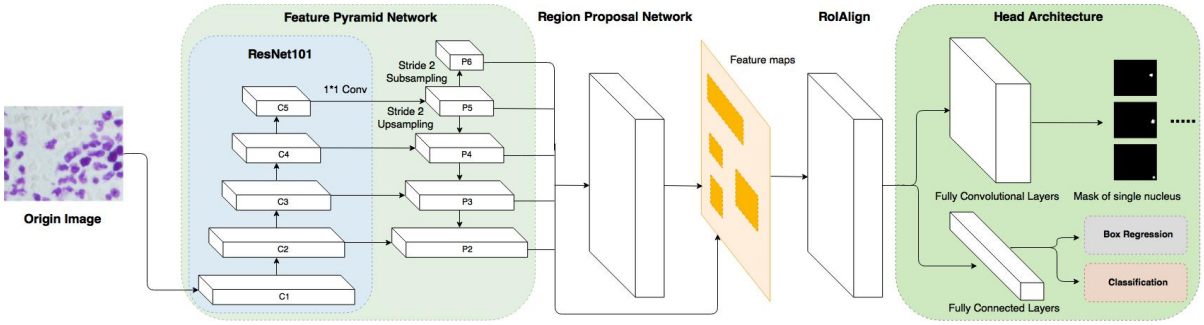


Figure 3. The architecture of Mask R-CNN.

3.1. Data Preparation

The dataset we explore is provided by Kaggle 2018 Data Science Bowl competition (https://www.kaggle.com/c/data-science-bowl-2018), which contains 670 nucleus images. Each image has multiple masks, each of which indicates a position of an individual nucleus. This labelled set is divided into a training set with 603 samples and a validation set with 67 samples. The competition also provided an image set which contains 65 nucleus images without masks, which is used as a test set and need to be evaluated on the official website of the competition. Since these images were acquired under various conditions, the colours of the images and the shapes of nuclei are quite different. These variations require our models have good generalizability. As is mentioned in section 2, FCN and Mask R-CNN allow inputs to have different sizes, so we do not need to reshape the images.

In the baseline experiments, we pre-processed the dataset by merging the masks of each nucleus image to single mask image that shows the positions of all the nuclei, in order to make the ground truth of the predictions in FCN.

Because the size of the dataset is small, we apply 10-fold cross validation strategy to evaluate predictions in our baseline experiments. However, Mask R-CNN training is much more time-consuming and costly. Due to the limitation of time and computational resources (see section 3.4.2), we do not use cross validation in Mask R-CNN experiments. Instead, we apply data augmentation to let the network learn more from the dataset and avoid overfitting. We augment the data by randomly changing the brightness of the input images and flipping the images from left to right and/or up to down, before inputting them into the networks.

Moreover, since we use pre-trained the Mask R-CNN model by COCO dataset (Lin et al., 2014), we subtract the mean of COCO dataset from all the nucleus images, in order to adapt our dataset for the pre-trained model.

3.2. Evaluation

We use the mean average precision (mAP) at different intersection over union (IoU) thresholds to evaluate our task. The threshold values range from 0.5 to 0.95 with a step size of 0.05:

$$t = \{0.5, 0.55, 0.6, \ldots, 0.95\}, step = 0.05 \tag{1}$$

An mask is considered true positive (TP) if the IoU between it and the ground truth is higher than the thresholds, and false positive (FP) otherwise. If the ground truth objects have no corresponding predicted masks, they are considered false negative (FN). The precision at each threshold is defined as the ratio of the number of TP to the sum number of TP, FP and FN. The average precision (AP) of a single image is then calculated as the average of the precisions at all IoU thresholds.

$$AP = \frac{1}{|thresholds|} \sum_{t} \frac{TP(t)}{TP(t) + FP(t) + FN(t)} \tag{2}$$

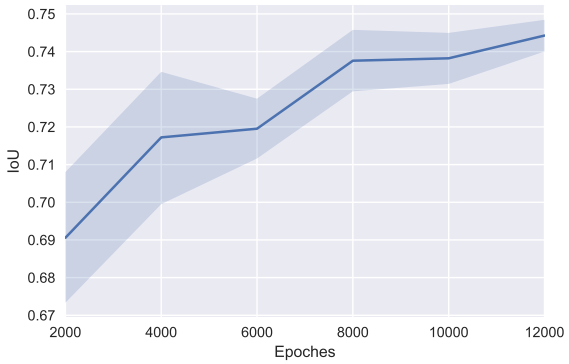
Finally, the mean average precision (mAP) of the task is the mean of all the APs of the individual images in the dataset.

3.3. Baseline Experiments

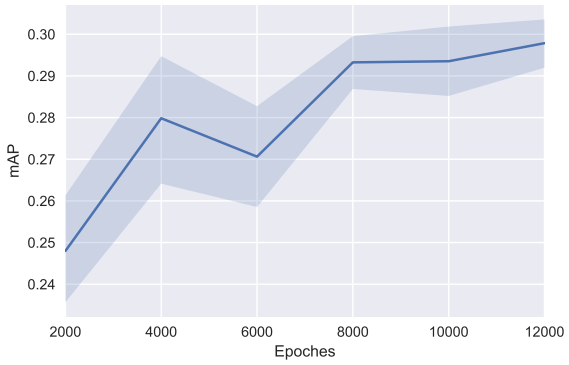
In our baseline, we constructed FCN-8s with the backbone of pre-trained VGG16 (Simonyan & Zisserman, 2014), as well as the separation method – open operation. ReLU (Nair & Hinton, 2010) is used here as activation function, and we dropout the weights at 0.5 rate in the last two fully convolutional layers to avoid overfitting. The optimization rule used here is Adam (Kingma & Ba, 2015), with a learning rate of 10^{-5}.

In addition, we applied deconvolution to the last fully convolutional layer and the last two max-pooling layers. The code is modified based on Teichmann et al. (2016). We trained the model for 12000 epochs, and evaluates at each 2000 epochs.

Figure 4(a) shows the average IoU of the whole pictures across 10-fold validation sets, which means the average coverage of masks on the whole pictures. By contrast, figure 4(b) shows the mAP, which is mentioned in section 3.2, on cross validation sets. It is clear that the mAP is approximately proportional to IoU. However, although the IoU is



(a) IoU



(b) mAP

Figure 4. IoU and mAP with standard deviation on 10-fold cross validation in baseline experiments.

not bad, the mAP shown in figure 4(b) is not ideal, which indicates that the separation of the masks in the same images restricts the score of mAP. As mentioned in section 2, the degree of erosion and dilation is hard to set. That is to say, with semantic segmentation methods, the task also high requires a good separation method. This shows the limitation of FCN, which cannot separate the masks automatically.

3.4. Experiments with Mask R-CNN

According to the results of the baseline experiments, semantic segmentation methods perform well in dividing the region in which cells locate, but they fail to tell how many cells are in the region and to segment the area of individual region. As is mentioned in section 3.3, the limitation of the semantic segmentation in this task is the high requirement of post-processing. Hence, we move forward to use Mask R-CNN, which is a state-of-the-art instance segmentation method and could directly predict the masks of individual nuclei.

3.4.1. Network Architecture

The architecture of Mask R-CNN could be divided into two part - backbone and head architecture.

In this series of experiments, we use pre-trained ResNet-101-FPN as the backbone of Mask R-CNN. Residual networks (ResNets) is a state-of-the-art CNN architecture (He et al., 2016). Instead of learning features directly, ResNets learn residuals of features through many residual modules. In each ResNet module, we add the original inputs to the outputs, making sure that what the network learn in this process is the residual, which is shown in figure 5. ResNet-101 used here contains 101 weighted layers, and we apply ReLU (Nair & Hinton, 2010) and batch normalization (Ioffe & Szegedy, 2015) between every two layers. The architecture of ResNet-101 is shown in table 1. Feature Pyramid Network (FPN), proposed by (Lin et al., 2017), uses an image pyramid to build a feature pyramid for each input. That is to say, for each image, FPN rescale its size several times and extract figure maps from every size. The low-level feature maps are extracted from the re-scale images and the high-level feature maps. The features of RoIs are extracted in different levels of feature maps according to their scale. It has been proved in He et al. (2017) that the ResNet-FPN backbone for feature extraction with Mask R-CNN could give excellent gains in both accuracy and speed.

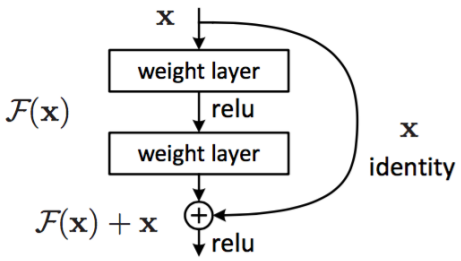


Figure 5. ResNet Module

stage	filter
C1	$(7 \times 7, 64, stride2)$
max_pool	$(3 \times 3, stride2)$
C2	$[(1 \times 1, 64) + (3 \times 3, 64) + (1 \times 1, 256)] \times 3$
C3	$[(1 \times 1, 128) + (3 \times 3, 128) + (1 \times 1, 512)] \times 4$
C4	$[(1 \times 1, 256) + (3 \times 3, 256) + (1 \times 1, 1024)] \times 23$
C5	$[(1 \times 1, 512) + (3 \times 3, 512) + (1 \times 1, 2048)] \times 3$

Table 1. The structure of ResNet-101. Every row stands for a stage, denoted by C1 to C5. C2 to C5 contains different numbers of ResNet Modules, and between any two of them there are pooling layers. Between every two weighted layers there are ReLU and batch normalization layers.

The features maps obtained by the backbone will be processed by RPN to get the regions of interest. Then these RoIs will be combined with the feature maps, being processed by RoIAlign layer to get two clusters of RoIs. One cluster has the shape of 7×7, and another has the shape of 14×14. The weights in RPN is also pre-trained and the training of RPN is separated from the training of Mask R-CNN, although they share the same backbone. After that, the two clusters of feature maps will enter into the head architecture. See the details of RPN in section 3.4.4.

The head architecture has two branches, and each receives a cluster of feature maps. The first one, receiving the RoI cluster of the shape 7×7, aims to do classification and bounding-box regression. The figures are filtered by a $3 \times 3 \times 256$ convolutional layer and dealt with by two fully connected layers with 1024 dimensions. Then a fully connected layer and softmax function help to transform these features to classes, and giving the confidence of the RoIs belonging to the foreground. At the meantime, another fully connected layer helps to transform these features to the positions of bounding-boxes. The second branch is used to predict the masks on the RoIs of the 14×14 shape, which contains several FCN to regress the masks. The outputs of this branch is 100 masks with the shape of 28×28. To avoid multi-masks being generated for the same nuclei, we apply non-maximum suppression strategy (NMS). That is to say, we check whether the IoUs between the masks exceed the fixed threshold. If so, only the masks with the highest confidence scores are remained, and the others are eliminated. NMS is also applied in RPN in section 3.4.4. In addition, the masks with the confidence lower than a fixed score will be eliminated, too. The remaining masks are the final outputs of Mask R-CNN. The details of the head architecture is shown in table 2.

branch 1	**branch 2**
input shape: $(7 \times 7 \times 256)$	input shape: $(14 \times 14 \times 256)$
conv-$(3 \times 3, 256)$	conv-$(3 \times 3, 256) \times 4$
fc-(1024)	deconv-$(2 \times 2, 256, stride2)$
fc-(1024)	conv-$(3 \times 3, 80)$
fc-(2) $\rightarrow$ softmax $\rightarrow$ class: *cs* or fc-(4) $\rightarrow$ bbox: (x,y,w,h)	conv-$(1 \times 1, 2) \rightarrow$ softmax $\rightarrow$ mask shape: (28×28)

Table 2. The head architecture of Mask R-CNN, where conv and fc stand for convolutional layers and fully connected layers. ReLU is applied in hidden layers. *cs* is the confidence score of the class, and (x,y,w,h) stands for the coordinates of a vertex of the bounding-box (bbox) and the width and height.

The entire network architecture is shown in figure 3.

3.4.2. Experiments Configuration

Due to the limitation of RAM of GPUs, we resize the images to 512×512 before inputting them into the networks. The parameters of Mask R-CNN is pre-trained on COCO dataset (Lin et al., 2014), except its head structure. The strides of each ResNet layer (from C1 to C5) of the FPN is 4, 8, 16, 32 and 64, respectively. The sizes of the square anchors generated in RPN are 32, 64, 128, 256 and 512. The ratios of width over height of the anchors are 0.5, 1 or 2. The NMS threshold in RPN is 0.7. The ratio of positive labels is 0.33. The details of the hyperparameters related to RPN are explained in section 3.4.4. The NMS threshold and confidence threshold in the mask prediction task is 0.3 and 0.7, respectively. The optimization rule is SGD with a learning rate of 10^{-3}, instead of Adam, because according to Wilson et al. (2017) SGD tends to find better optima than Adam.

Since we apply transfer learning from COCO dataset to the nuclei images by using the pre-trained weights in Mask R-CNN, we do not have to train all the layers. Actually, training all the layers is prone to result to significant overfitting, because the capacity of Mask R-CNN is very large and the number of the nucleus images is far smaller than COCO. Therefore, the strategy applied is to freeze most layers and retrain the remaining layers. In the experiments, the training process is divided into two stage. In stage 1, we only train the head architecture of the network for 100 epochs. Then, in stage 2, we train the last two stages (C4 and C5) of ResNet along with the head architecture for further 50 epochs. In addition, RPN is trained in both stages. Each training epoch contains 100 iterations. The evaluation on the validation set happens every 10 epochs from 5^{th} epoch to 145^{th} epoch. Evaluation is expensive so we cannot evaluate the model on every epoch. The code is modified based on Waleed et al. (2017). All experiments are carried out on a NVIDIA Quadro P5000 with RAM of 16GB.

3.4.3. Data Augmentation

As is mentioned before, because of the small size of dataset, we apply data augmentation to avoid overfitting. See the details of the augmentation methods in section 3.1. The results are shown in figure 6. We evaluate the effective of data augmentation on the validation set. At the beginning, the mAP of the predicted masks without data augmentation is higher than the mAP of the experiment with data augmentation. However, the former increases slowly and overfits at the last twenty epochs, while the latter exceeds the former at about the 40^{th} epoch, and increases more steadily than the former. In addition, the mAP of experiments with data augmentation consistently keeps higher than the mAP without data augmentation after around 80^{th} epoch. It is clear that data augmentation suppresses the overfitting and increases the mAP of the predicted masks. That is to say, more data helps the networks learn more from the features, and enhance the generalization of the networks. These experiments proves the effective of data augmentation, especially for small datasets. Therefore, data augmentation is applied in the following experiments.

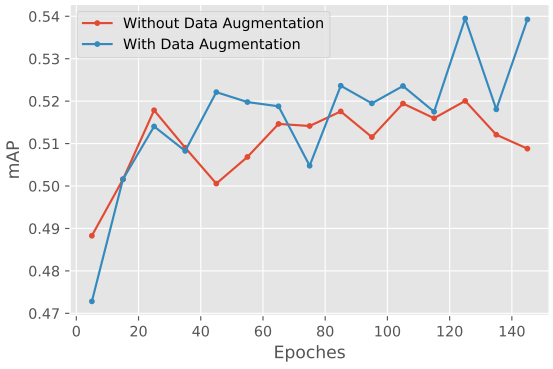
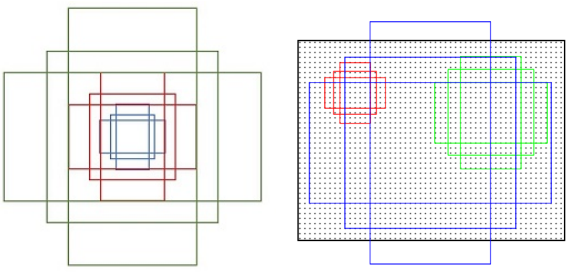


Figure 6. The mAP on validation set with and without data augmentation

3.4.4. RPN Optimization

RPN (Ren et al., 2015) receives the feature maps extracted by the backbone and proposes RoIs of the feature maps. RPN slides over the feature maps and generates anchors for all pixels. For each pixel, rectangle anchor boxes of different shapes, centred at the pixel, are generated. The sizes and shapes of the authors are controlled by the hyperparameters mentioned in section 3.4.2, and the generated authors look like figure 7. These anchors are the candidate RoIs. The anchors beyond the boundaries are removed. Among the remaining anchors, we compute the IoU of each anchor with the ground truth. The anchors with IoUs greater than 0.7 are labelled positive, and the ones with IoUs less than 0.3 are labelled negative. Only positive and negative anchors could enter into the sub-network, and the remaining anchors are eliminated. RPN firstly do bounding-box regression on these anchors, and then classify the anchors to foreground and background by giving confidence scores. As mentioned in section 3.4.1, these two task are trained separately with the training of Mask R-CNN. Then we sorts the remaining boxes by their scores, and clip their boundaries to the boundaries of the feature maps. Since these authors are easy to overlap with each other, we also apply NMS to eliminate the repeated authors for the same nuclei. Similar to NMS applied in section 3.4.1 at head architecture, among the overlapped anchors, only the ones with the highest scores are remained, and the others are eliminated. Until now, the number of negative anchors is much larger than the number of positive anchors, so the negative anchors will dominate the following training and significantly impact the loss function. Therefore, we select the positive anchors and the negative anchors with high confidence scores, according to the ratio of positive labels and the maximum number of outputs set in section 3.4.2, and these anchors are regarded as RoIs. Finally, these RoIs enter into the next block of Mask R-CNN – RoIAlign.



(a) anchors for one pixel (b) anchors in a feature map

Figure 7. The examples of anchors

In this section, we carefully tune there hyperparameters of RPN that control the number and quality of proposed RoIs, which is important because the final mask prediction are based on these RoIs. Good RoIs are the prerequisite and guarantee of the accurate mask prediction. These hyperparameters controls the number of anchors generated at the beginning, the threshold of non-maximum suppression and the number of anchors selected at the final stage, respectively. Higher generation number means more candidates, while lower thresholds and selection number means more serious filter rules. In the previous experiments, these hyperparameters are set as (256, 0.7, 200), and in this section we compare several other configurations.

At first, we carry out the experiments on different generation numbers - 128, 256 and 512, fixing the other two hyperparameters at (0.7, 200). The results of the experiments is shown in figure 8. The situation of 128 is worse than the other situations. The mAP of 128 increases significantly before the 40^{th} epoch. However, it fluctuates after that and does not see a clear boost, and overfits seriously after around the 115^{th} epoch. In terms of originally generating 256 anchors, although the mAP also fluctuates, the clear overall increase could be seen. The situation of 512 is no worse than 256. The mAP of 512 increases more steadily than 256, expect an abrupt decline at the 95^{th} epoch. Moreover, the rate of convergence of mAP of 512 is higher than mAP of 256, although there seems to be a little overfitting at the end of training. In general, randomly generating more anchors helps to improve the mAP, because more anchors means higher probability of framing more nuclei in the images, which is prone to increase the number of TP. Although the anchors generated randomly are filtered in the next stage, more candidates guarantees higher quality of RoIs.

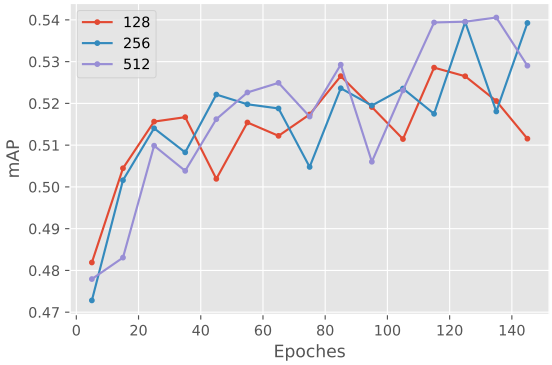


Figure 8. The mAP on validation set with different numbers of anchors generated at the beginning of RPN.

Although we increase the number of candidate anchors, the NMS thresholds and the number of output RoIs are also prone to impact the mAP of final masks. Thus, we decide to tune these two hyperparameters. As shown in figure 9, only increasing the the number of RoIs does not improve the mAP. Actually, it makes mAP a little bit lower, because more RoIs increase the risk of the appearance of FP. NMS threshold could adjust the degree of the suppression of multi-masks for the same nuclei, impacting the number of FP. Hence, the following experiments focus on NMS threshold with 350 RoIs. We do not choose 200 as the number of RoIs, because small number of RoIs may eliminate the anchors that uniquely frame a nucleus while do not have very high confidence scores. From figure 9, we could see that lower NMS threshold leads to better mAP, because of the reduction of FP. On the other hand, this threshold

cannot be too low, because it would cause the RoIs of the close nuclei being eliminated by NMS.

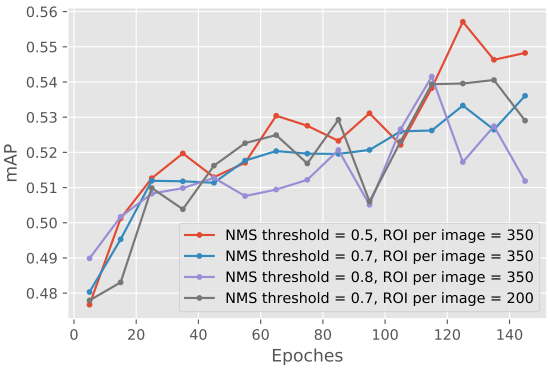


Figure 9. The mAP on validation set with different NMS thresholds and different numbers of proposed RoIs.

3.5. Effectiveness and Generalization

The figure 10 compares the effectiveness of FCN and Mask R-CNN by a segmentation example on the validation set, along with the ground truth. It can be seen that FCN fails to separate the masks of some nuclei, while this seems not to be a problem with Mask R-CNN. Both these two network miss some very small nuclei and mix the masks of some close nuclei. However, Mask R-CNN segments nuclei more carefully.

(a) FCN (b) Mask R-CNN (c) ground truth

Figure 10. Comparison of a segmentation result on the validation set among FCN, Mask R-CNN and the ground truth.

As mentioned in section 3.2, the ground truth of the masks of the test set is not provided, so we have to test the generalization of our models through official system. In terms of the baseline model, we get the mAP of 0.229 on the test set. However, the model of Mask R-CNN could achieve the mAP of 0.401, which is much higher than the baseline and rates 545/3118 at the leaderboard on 28^{th} March 2018. The highest score among participants is 0.569 by then.

Therefore, Mask R-CNN has a better effectiveness and generalization in segmenting nucleus images. Actually, Mask R-CNN adopts an instance-first strategy, which means that the instances will be segmented firstly and then given the masks, while FCN & open operation is a segmentation-first method, which predicts the whole masks firstly and then separates them. Generally speaking, since instance-first strategy does not highly require good separation methods, it performs better than segmentation-first strategy.

4. Related Works

4.1. R-CNN Series

The Region-based CNN (R-CNN) approach to bounding-box object detection (Girshick et al., 2014) is to apply Region Proposal Network (RPN) (Ren et al., 2015) to get some sets of proposal regions where objects are likely located, rather than applying classification network to every possible location and scale in the image. Then it applies the CNN for classification to each of these proposal regions which will end up being much more computationally tractable than trying to do all possible locations and scales. Faster R-CNN, as the name implies, is a more efficient model which extends R-CNN by learning the attention mechanism with a RPN using RoIPool to attend to RoIs on feature maps. As a current leading framework, Faster R-CNN consists of two stages. The first stage proposing candidate object bounding boxes is called RPN. The second stage extracts features using RoIPool from each candidate box and performs classification and bounding-box regression.

Mask R-CNN is kind of like a hybrid between semantic segmentation and object detection, including improvements made to Fast/Faster R-CNN (Ren et al., 2015) and FCNs. The extensions of Faster R-CNN are adding a branch network for predicting segmentation masks on each RoI and replace RoIPool with RoIAlign. FCN is applied to each RoI of the mask branch.

4.2. Instance Segmentation

Instance segmentation requires us not only the detection of all nuclei in each image correctly, but also precisely segmenting each instance, regardless of its shape and location. It can be seen as the combination of object detection classifying individual objects as well as localizing each with a bounding box, and semantic segmentation which aims to classify each pixel into a fixed set of categories without differentiating object instances.

There are different approaches to instance segmentation driven by the effectiveness of R-CNN. Earlier methods such as DeepMask (Pinheiro et al., 2015) let segmentation precede recognition resulting in low efficiency and less accurate. For instance, a complex multiple-stage cascade proposed by Dai et al. (2016) aims to predict segment proposals from bounding-box proposals, followed by classification. Recently, an approach called "fully convolutional instance segmentation" (FCIS) has been implemented by Li et al. (2017) where the key point is to predict a set of position-sensitive output channels fully convolutionally. Although this model simultaneously addresses object classes, boxes, and masks, leading to high speed of the system, it has trouble on dealing with overlapping instances and creating spurious edges.

There are also a group of ways to instance segmentation which are driven by semantic segmentation. The main

idea of them is to cut the pixels of the same category into different instances after getting the per-pixel classification results.

By comparison, Mask R-CNN adopts instance-first strategy rather than segmentation-first strategy. In addition, it is based on parallel prediction of masks and class labels, which makes our model simpler and more flexible.

4.3. Extensions of Mask-RCNN

Mask R-CNN framework has good generality and flexibility, thus has a wide range of applications. For example, one can adopt Mask R-CNN with ResNet-50-FPN backbone to predict human pose, i.e. keypiont detection (He et al., 2017), where the keypoints are the joints of a human body. The location of each keypoint is modelled as a one-hot mask, and evaluated by AP. The challenge of this task is not only the detection for each person in the given images, but also the prediction of the keypoints. In this process, we estimate each mask for each keypoint type (e.g. right knee, left sprain). For each of the K keypoints of an instance, the training target is a binary mask where only a single pixel is labeled as foreground, and the keypoints are processed in dependently. In this task, the keypoint-level localization accuracy requires a relatively high resolution output. There are three branches, including segment, mask and keypoint, which enable the unified system prediction to be efficient. The results can be seen in 11. As is shown, regardless of the various gestures and location, Mask R-CNN performs well in both person segmentation and keypoint detection.

(a) keypoint 1 (b) keypoint 2

Figure 11. Keypoint detection results on COCO test using Mask R-CNN (ResNet-50-FPN)

Since Mask R-CNN model is the state-of-art technique for instance segmentation,there will be much more other instance-level tasks exploited by Mask R-CNN.

4.4. Future Work

Due to the limitation of time and GPU computational resource, the more efficient and reliable experiments cannot be carried out in this project, but they are still worth to explore in the future. Firstly, k-fold validation strategy could be applied on Mask R-CNN to make the evaluation on the validation sets more convincing, and increase the effective of hyperparameter optimization. Secondly, ResNeXt (Xie et al., 2017) may perform better than ResNet as the backbone of Mask RCNN, since the former is an upgraded version of the latter and has the potential of better feature extraction. Also, other instance segmentation models, e.g. MaskLab (Chen et al., 2017), could be used on the nucleus segmentation task.

5. Conclusions

The project aims to segment the nuclei in the images acquired in different microscopy system by deep learning models. The baseline experiments are based on FCN and open operation separation method, while the main experiments are based on Mask R-CNN. FCN can only semantically segment the nuclei and the background but fails to separate the individual nuclei automatically, so the results of the experiments is highly constrained by the separation methods. In contrast, the strategy of Mask R-CNN is proposing RoIs on the feature maps extracted from original maps and then segmenting nuclei on individual RoIs, so it is exempted from the limitation of separation methods. After applying data augmentation and optimizing the hyperparameters, the Mask R-CNN achieves the mAP score of 0.401 on the test set, outperforming the baseline experiments where mAP score is only 0.229. That is to say, the instance-first method Mask R-CNN works much better than the segmentation-first method FCN in instance segmentation of nucleus images.

References

Chen, L.-C., Hermans, A., Papandreou, G., Schroff, F., Wang, P., and Adam, H. MaskLab: Instance Segmentation by Refining Object Detection with Semantic and Direction Features. *ArXiv e-prints*, December 2017.

Dai, Jifeng, He, Kaiming, and Sun, Jian. Instance-aware semantic segmentation via multi-task network cascades. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3150–3158, 2016.

Girshick, Ross, Donahue, Jeff, Darrell, Trevor, and Malik, Jitendra. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

He, Kaiming, Gkioxari, Georgia, Dollár, Piotr, and Girshick, Ross B. Mask R-CNN. *CoRR*, abs/1703.06870, 2017. URL http://arxiv.org/abs/1703.06870.

Ioffe, Sergey and Szegedy, Christian. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Bach, Francis and Blei, David (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 448–456, Lille, France, 07–09 Jul 2015. PMLR. URL http://proceedings.mlr.press/v37/ioffe15.html.

Kingma, Diederik P. and Ba, Jimmy. Adam: A method for stochastic optimization. *eprint arXiv:1412.6980*, 2015.

Li, Yi, Qi, Haozhi, Dai, Jifeng, Ji, Xiangyang, and Wei, Yichen. Fully convolutional instance-aware semantic segmentation. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pp. 2359–2367, 2017.

Lin, Tsung-Yi, Maire, Michael, Belongie, Serge J., Bourdev, Lubomir D., Girshick, Ross B., Hays, James, Perona, Pietro, Ramanan, Deva, Dollár, Piotr, and Zitnick, C. Lawrence. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. URL http://arxiv.org/abs/1405.0312.

Lin, Tsung-Yi, Dollár, Piotr, Girshick, Ross, He, Kaiming, Hariharan, Bharath, and Belongie, Serge. Feature pyramid networks for object detection. In *CVPR*, volume 1, pp. 4, 2017.

Nair, Vinod and Hinton, Geoffrey E. Rectified linear units improve restricted Boltzmann machines. In *Proc ICML*, pp. 807–814, 2010.

Pinheiro, Pedro O, Collobert, Ronan, and Dollár, Piotr. Learning to segment object candidates. In *Advances in Neural Information Processing Systems*, pp. 1990–1998, 2015.

Ren, Shaoqing, He, Kaiming, Girshick, Ross, and Sun, Jian. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pp. 91–99, 2015.

Shelhamer, Evan, Long, Jonathan, and Darrell, Trevor. Fully convolutional networks for semantic segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 39(4):640–651, April 2017. ISSN 0162-8828.

Simonyan, Karen and Zisserman, Andrew. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. URL http://arxiv.org/abs/1409.1556.

Teichmann, Marvin, Weber, Michael, Zoellner, Marius, Cipolla, Roberto, and Urtasun, Raquel. Multinet: Real-time joint semantic reasoning for autonomous driving. *arXiv preprint arXiv:1612.07695*, 2016.

Waleed, Ferriere, Phil, and Pruce, Cory. Mask r-cnn for object detection and instance segmentation on keras and tensorflow, 2017. URL https://github.com/matterport/Mask_RCNN.

Wilson, A. C., Roelofs, R., Stern, M., Srebro, N., and Recht, B. The Marginal Value of Adaptive Gradient Methods in Machine Learning. *ArXiv e-prints*, May 2017.

Xie, Saining, Girshick, Ross, Dollár, Piotr, Tu, Zhuowen, and He, Kaiming. Aggregated residual transformations for deep neural networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pp. 5987–5995. IEEE, 2017.