applied sciences

MDPI

Article

Trichomonas vaginalis Detection Using Two Convolutional Neural Networks with Encoder-Decoder Architecture

Xiangzhou Wang, Xiaohui Du *, Lin Liu, Guangming Ni, Jing Zhang, Juanxiu Liu and Yong Liu

MOEMIL Laboratory, School of Optoelectronic Information, University of Electronic Science and Technology of China, No. 4, Section 2, North Jianshe Road, Chengdu 610054, China; 201511050116@std.uestc.edu.cn (X.W.); liulin1979@uestc.edu.cn (L.L.); guangmingni@uestc.edu.cn (G.N.); zhangjing@uestc.edu.cn (J.Z.); juanxiul@uestc.edu.cn (J.L.); yongliu@uestc.edu.cn (Y.L.)

* Correspondence: xiaohuie@uestc.edu.cn

Abstract: Diagnosis of *Trichomonas vaginalis* infection is one of the most important factors in the routine examination of leucorrhea. According to the motion characteristics of *Trichomonas vaginalis*, a viable detection method is the use of a microscopic camera to record videos of leucorrhea samples and video object detection algorithms for detection. Most *Trichomonas vaginalis* is defocused and displays as shadow regions on microscopic images, and it is hard to recognize the movement of shadow regions using traditional video object detection algorithms. In order to solve this problem, we propose two convolutional neural networks based on an encoder-decoder architecture. The first network has the ability to learn the difference between frames and utilizes the image and optical flow information of three consecutive frames as the input to perform rough detection. The second network corrects the coarse contours and uses the image information and the rough detection result of the current frame as the input to perform fine detection. With these two networks applied, the metric value of the mean intersection over union of *Trichomonas vaginalis* achieves 72.09% on test videos. The proposed networks can effectively detect defocused *Trichomonas vaginalis* and suppress false alarms caused by the motion of formed elements or impurities.

Keywords: *Trichomonas vaginalis* detection; rough and fine detection; video object detection; convolutional neural network; encoder-decoder architecture

Citation: Wang, X.; Du, X.; Liu, L.; Ni, G.; Zhang, J.; Liu, J.; Liu, Y. *Trichomonas vaginalis* Detection Using Two Convolutional Neural Networks with Encoder-Decoder Architecture. *Appl. Sci.* **2021**, *11*, 2738. https://doi.org/10.3390/app11062738

Academic Editors: Nektarios Koukourakis and Robert Kuschmierz

Received: 24 February 2021
Accepted: 14 March 2021
Published: 18 March 2021

1. Introduction

Diagnosis of *Trichomonas vaginalis* (TV) infection is one of the most important factors in the routine examination of leucorrhea. The traditional manual microscopic examination method has the advantage of high detection rates, but its low efficiency mean it cannot meet the need for daily examination of a large number of leucorrhea samples. Therefore, fully automated leucorrhea examination equipment with an intelligent algorithm for TV detection is in urgently needed. Staining TV is a commonly used method of leucorrhea sample pretreatment. The advantages of this method are that the contours of TV after staining are clear and it is easy to distinguish from other formed elements or impurities. The staining process is complicated and time-consuming, so it is not suitable for integration into fully automated leucorrhea examination equipment. According to the motion characteristics of TV, a feasible detection method is using a microscopic camera to record videos of leucorrhea samples and adopt video object detection algorithms to identify it [1,2].

Traditional video object detection algorithms include frame difference methods, background difference methods, and optical flow methods. Frame difference methods determine the moving foreground object by comparing the difference between adjacent frames or three frames [3,4]. Background difference methods utilize the image information of previous frames of the video to establish a background model and then judge the foreground or background by comparing the difference between the current frame and the background model [5,6]. The background model is updated according to foreground and background

results of the current frame. Optical flow methods match the pixels in adjacent frames to obtain the motion direction and step of each pixel in the current frame [7].

Video object detection algorithms based on deep learning mainly include flow-based methods [8–10]. The principle of these methods is that in the feature extraction stage (encoder), only the feature maps of the key frames in the video are extracted. For non-key frames, the feature maps are generated by the feature maps of the key frames via the optical flow field [8,10]. In the classifier stage (decoder), the feature map of a single frame [8] or feature aggregation information of multiple frames [9] is used as the input to detect the moving object in the current frame. These flow-based methods improve the detection speed while ensuring the detection rate.

In the vertical direction of the microscope stage, the TV is located in different planes from the formed elements or impurities, due to its motion characteristic. Figure 1 shows one frame of the video with clear formed elements (epithelial cells and white blood cells). The number of TV is 9 in this frame. The TV and some backgrounds have been marked with red and green rectangles, respectively. Some enlarged image regions are shown in Figure 1a–d. In the vertical direction, the TV in (b) is close to the current plane and its outline can still be observed; the TV in (a) and (c) is far away from the current plane and is displayed as shadow regions in the image. By observing successive frames of the video, it can be seen that (d) is not TV but is a defocused formed element or impurity. It is not possible to accurately determine whether the defocused shadow region is TV or background by using only one single frame. For frame difference methods or background difference methods, it is necessary to lower the foreground judgment threshold to recognize the shadow regions where the defocused TV is located. However, some background regions will be mistakenly detected as the foreground regions, leading to false alarms. The artificial identification of defocused TV is mainly based on the characteristics of the shadow area and the continuous movement of TV. Therefore, image and optical flow information jointly determine the features of defocused TV. In flow-based methods [8–10], optical flow is mainly used for propagating feature maps between key frames and non-key frames, rather than as input information. The extracted features only contain image information and the trained network may mistakenly identify some stationary shadow regions in the background as TV.

In order to solve the above problems, we propose two convolutional neural networks based on an encoder-decoder architecture. The first network has the ability to learn the difference between frames and utilizes the image and optical flow information of three consecutive frames as the input to perform rough detection of the TV. The second network corrects the coarse contours and uses the image information and the rough detection result of the current frame as the input to perform fine detection of the TV. By combining these two networks, the defocused TV can be detected effectively and the false alarms caused by the motion of formed elements or impurities can be suppressed.

This article is organized as follows: Section 2 introduces our previous works on TV detection. The details of two convolutional neural networks with encoder-decoder architecture are described in Section 3. Section 4 introduces the dataset we used and the experimental results. Section 5 presents the discussion. Conclusions are provided in Section 6.

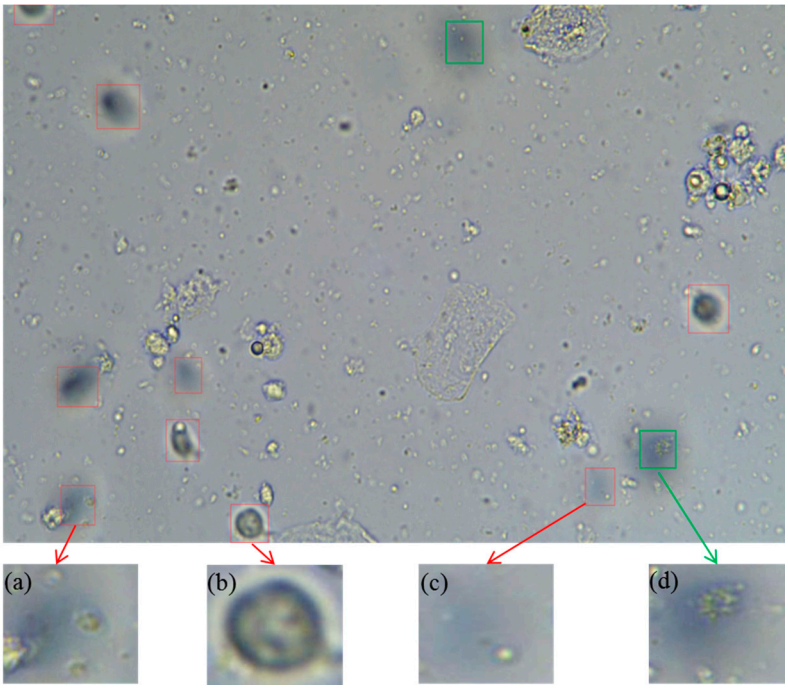


Figure 1. One frame in the video has clear formed elements. The trichomonas vaginalis (TV) and some backgrounds are marked with red and green rectangles, respectively. (**a**–**c**) show the enlarged TV and (**d**) shows the enlarged background.

2. Related Works

At present, the detection of TV is mostly focused on biochemical and staining detection methods, whereas detection based on images or videos is used less. In our previous works, we proposed two TV detection methods based on a traditional background difference model.

In the work [1], we proposed an improved Kalman background reconstruction algorithm to detect TV automatically. The first frame of a video is used to build the background and the additional top-hat transformation can eliminate the phenomena of tailing and ghosting. By introducing time information and neighborhood judgment into the background updating mechanism, this method candeal with the problems of falsely detected static areas and missing motion areas. The algorithm results show that this method can effectively suppress the noise caused by illumination mutation, lens shift, and focal length variation, providing strong adaptability and good robustness.

When the movement speed of the moving target is slow or the movement frequency is low, the performance of this Kalman background reconstruction algorithm can decline, resulting in a high omission ratio. In order to address the above limitations, we proposed an improved VIBE background reconstruction algorithm [2]. The background model adopts three main update strategies: the memoryless update, the time subsampling of the model and the update of the spatial domain. In order to simplify the judgment, the foreground image is extracted by the frame difference method. Similarly, time information is introduced to eliminate false alarms from impurities or formed elements. This improved method can effectively suppress false alarms caused by formed elements and missed detections caused by the background model updating during the movement.

TV is defocused due to its motion characteristics and in most cases it appears as flat shadow regions with little difference between adjacent frames. To detect moving shadow regions, it is necessary to reduce the judgment threshold of the foreground, but this can result in false alarms where some backgrounds are detected as TV. The above two methods mainly focus on the recognition of clear TV but fail for a defocused conditions. Therefore,

a detection method based on deep learning is proposed in this paper to solve the above problems.

3. Method

3.1. Convolutional Neural Network Based on Encoder-Decoder Architecture for Rough Detection

Using only one single frame cannot effectively distinguish TV from backgrounds, so the first convolutional neural network for rough detection needs to have the ability to learn the differences between adjacent frames. Dosovitskiy et al. [11] proposed two encoder-decoder network architectures (FlowNetSimple and FlowNetCorr) to calculate the optical flow between adjacent frames by deep learning methods. The calculation of optical flow only depends on the difference between frames rather than the image content of one single frame. This detection method is appropriate for the defocused TV detection problem, so the first convolutional neural network we propose uses the encoder-decoder architecture similar to FlowNetSimple [11]. Figure 2 shows the architecture of the rough detection network. The encoder and eecoder are shown in the red dashed box on the left and the green dashed box on the right, respectively.

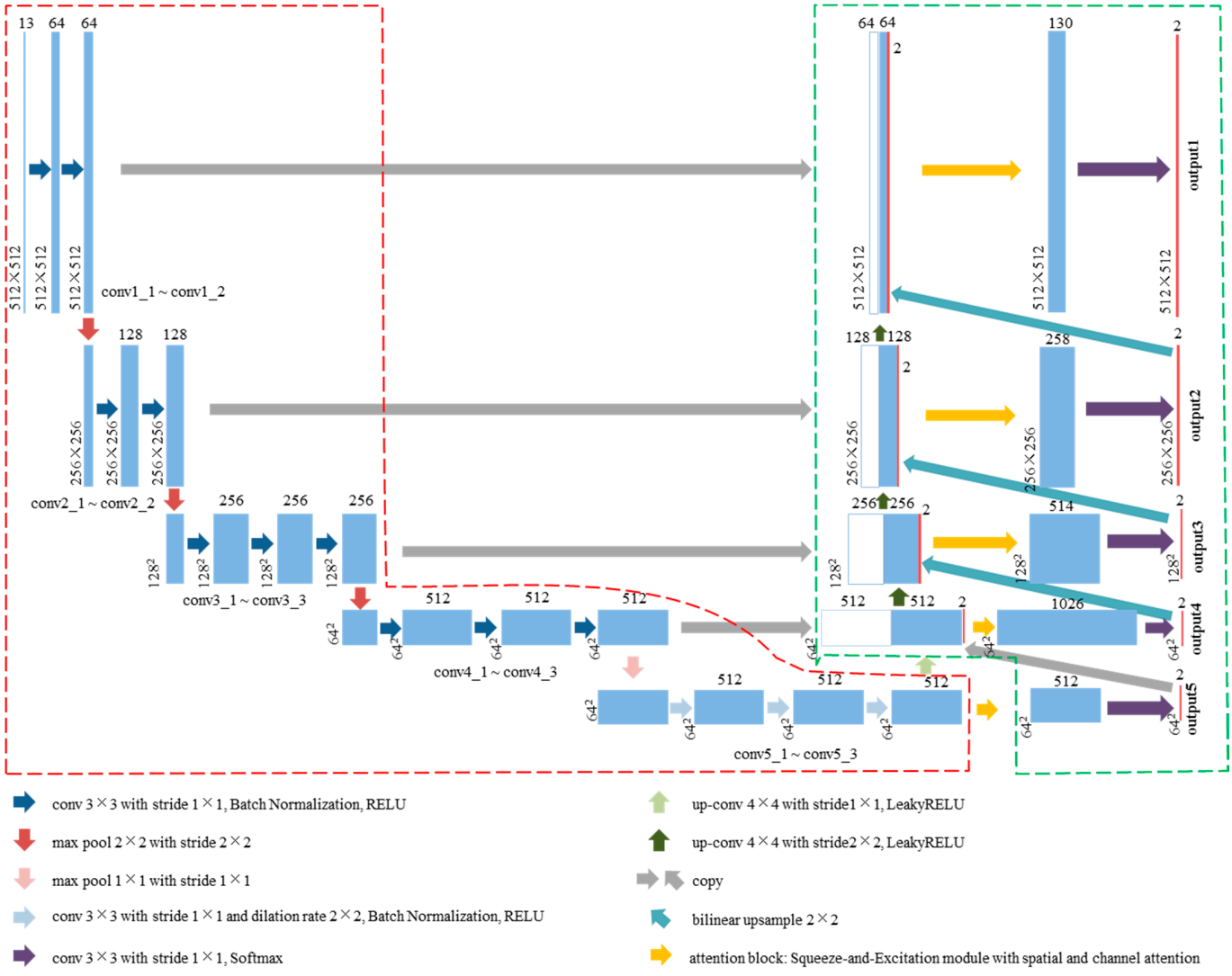


Figure 2. The architecture of the rough detection network. Each box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. The arrows denote the different operations. The encoder and decoder are shown in the red dashed box on the left and the green dashed box on the right, respectively.

3.1.1. Encoder

We concatenated the current frame and its preceding and following frames (RGB image, 3 channels), the optical flow from the previous frame to the current frame and from the current frame to the next frame (pixel movement of x and y direction, 2 channels) together along the channel axis, then fed it into the encoder. The input shape was 512 × 512 × 13. Through image and optical flow information, the rough detection network can learn the motion and morphological features of the TV, preventing the moving formed elements or impurities from being mistakenly detected. In order to reduce the missed detection of TV, we use information from three frames rather than two. Unlike FlowNetSimple [11], we use the layers 'conv1_1′ to 'conv5_3′ from VGG-16 [12] as the basic architecture of the encoder. The weights of 'conv1_2′ to 'conv5_3′ are initialized from the ImageNet pre-trained model. The stride of the 'pool4′ layer is set to (1, 1) and the 'conv5_1′ to 'conv5_3′ layers use dilated convolution with a dilation rate of (2, 2). The optical flow information between adjacent frames is calculated by the deep learning method proposed in [13].

3.1.2. Decoder

Based on the decoder architecture of FlowNetSimple [11], we added an attention block with a squeeze-and-excitation (SE) module [14] before each network output. As shown in Figure 3, the SE module contains spatial and channel attention modules, which can make it possible for the network to learn 'what' and 'where' to attend in the channel and spatial axes respectively. The spatial attention module generates a spatial attention map by utilizing the inter-spatial relationship of certain features. The input x (H × W × C_1) uses the convolution operation (kernel: [1, 1], stride: [1, 1], channels: 1) to obtain x_s(H × W × 1). Then we employ a simple gating mechanism with a sigmoid activation on x_s to obtain x'_s(H × W × 1). The spatial attention map $x_{spatial}$ (H × W × C_1) is generated by spatial-wise multiplication between the x'_s and the input x. The channel attention module can selectively enhance useful features and suppress invalid ones and produces a channel attention map. The x_c(1 × 1 × C_1) is generated by using a global average pooling operation on input x. After using full convolution (channels: C_3, $C_3 = C_1/4$) and Relu to x_c,x'_c(1 × 1 × C_3)was obtained. Then x'_c continuously executed fully convolution operation (channels: C_1) and sigmoid activation, obtaining x''_c(1 × 1 × C_1). The channel attention map $x_{channel}$(H × W × C_1) is generated by channel-wise multiplication between the x''_c and the input x. To get the output of the attention block, convolution (kernel: [3, 3], stride: [1, 1]), batch normalization and Relu are connected successively after adding two attention maps to.

In order to obtain dense per-pixel predictions, deconvolution is used to restore the rough feature map to the input size. There are 5 outputs with different scales: 'output5′ (64 × 64 × 2), 'output4′ (64 × 64 × 2), 'output3′ (128 × 128 × 2), 'output2′ (256 × 256 × 2), and 'output1′ (512 × 512 × 2). 'output5′ is generated by connecting the attention block (as shown in Figure 3) and convolution processing (kernel: [3, 3], stride: [1, 1], channels: 2, activation: softmax) after the feature map 'conv5_3′. As for 'output4′, this is created by convoluting 'concat4′, which is produced by concatenating 'deconv4′ 'conv4_3′, and 'output5′. 'deconv4′ is the feature map generated by the up-convolution (kernel: [4, 4], stride: [1, 1], channels: 512, activation: Leaky Relu with alpha 0.1) of 'conv5_3′. 'conv4_3′ is the low-level feature map used by skip-connection. The production of 'output3′, 'output2′, and 'output1′ are the same as 'output4′.

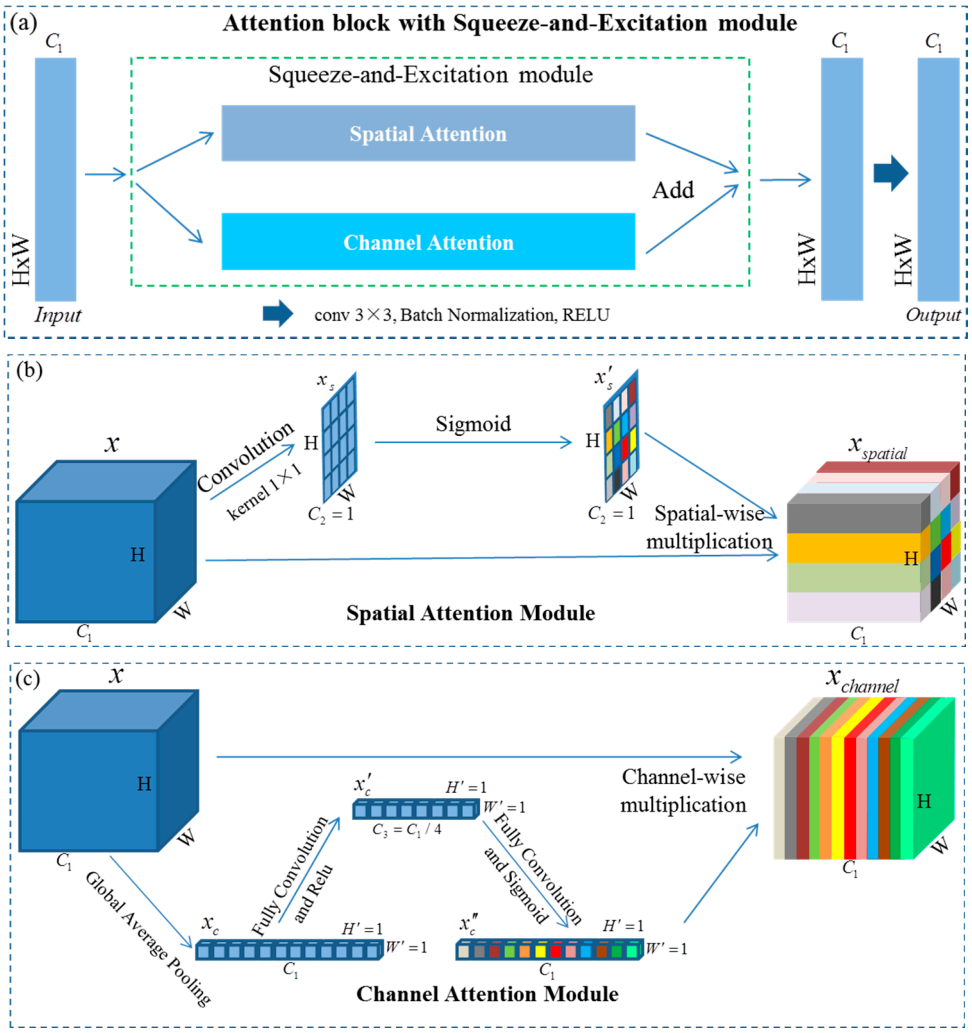


Figure 3. Architecture of attention block with squeeze-and-excitation (SE) module. (**a**) Overall structure of the attention block; (**b**) structure of spatial attention module; (**c**) structure of channel attention module. H, W, and C represent the height, width, and channel of the feature map, respectively.

3.1.3. Training

For the training phase, Adam [15] was selected as the optimizer. The learning rate was set to 10^{-5} and focal loss [16] (gamma: 2.0, alpha: 0.7) was the loss function. The loss weights for 'output1′ to 'output5′ were 1.0, 0.8, 0.8, 0.6, and 0.6. The size of the images and optical flow were rescaled to 1536x1024 and regions of fixed-size 512 × 512 were randomly cropped. The data augmentation methods include horizontal and vertical flip, rotation from $[-5°, 5°]$, translation from $[-10, 10]$ for x and y, and scaling from [0.8, 1.2]. The attention blocks introduce a large number of trainable parameters, which makes the network difficult to train. To solve this problem, we first trained the network without the attention mechanism, and then used the optimal model on the validation set as the pretrained model for transfer learning. Finally, we added the attention blocks before the five outputs and fine-tuned the network.

3.1.4. Inference

In the inference phase, inputs were rescaled to 1536 × 1024 and cropped as image patches with a fixed-size of 512 × 512 in the x and y directions with a step size of 256. We created an array I_{out} with a size of 1536 × 1024 × 2 to save results. 'output1′ was the unique output, which was packed into the corresponding region of I_{out}. The overlapping

regions took the maximum value. Finally, the category corresponding to the maximum of two channels was the predicted result for each pixel in I_{out}.

3.2. Convolutional Neural Network Based on Encoder-Decoder Architecture for Fine Detection

The outlines of the TV extracted by the rough detection network are coarse, so the second fine detection network we proposed needs to have a function to correct the contours. To solve the problem of video object segmentation, Perazzi et al. [17] proposed the MaskTrack method, which add the predicted mask image of the previous frame to the input of the network. The extra mask channel is meant to provide an estimate of the visible area of the object in the current frame, its approximate location and its shape, which is the inspiration for the second fine detection network to modify the rough detection result.

3.2.1. Encoder and Decoder

The fine detection network adopts the same architecture of the rough detection network but removes the extra attention blocks before five outputs. In addition, we stacked the current frame (RGB image, 3 channels) and the rough detection result for the current frame (binary image, 1 channel) together as the input of the encoder. The input shape was $512 \times 512 \times 4$.

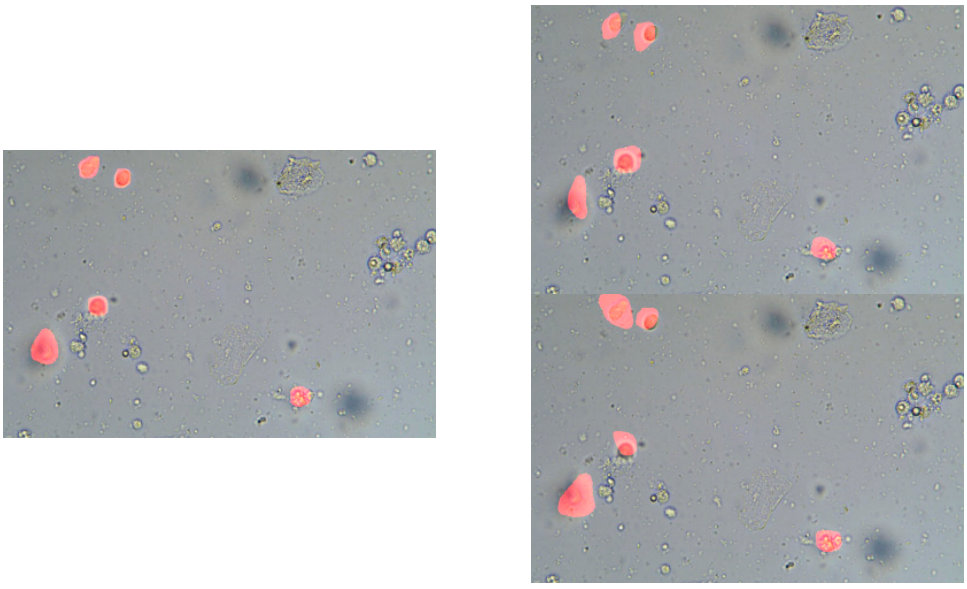
3.2.2. Training

In the training phase, instead of using the results for the rough detection network as the training set, we constructed random rough detection results artificially, due to false alarms and missed detections for the rough detection network. To obtain random rough detection results, we used affine transformations and non-rigid deformations via thin-plate splines [18] to deform the ground truth images. Because the motion of TV is independent, we deformed each TV randomly.

In order to prevent large distortion, we only kept the deformed result with an intersection over union value (between its original region and the deformed result) larger than 10% for non-rigid deformations. Affine transformations includes rotation from $[-15°, 15°]$, translation from $[-20, 20]$ for the x and y directions, and scaling from [0.5, 2.0]. A morphological dilation operation with a disc structuring element (15 pixels in diameter) was applied to remove the details of TV contours after the transformations. Examples of the generated rough detection results are shown in Figure 4. The optimization algorithm, loss function and other parameters used in the training phase were the same as those of the rough detection network. Since the TV regions have been randomly deformed, we only used the data augmentation methods of horizontal and vertical flip.

3.2.3. Inference

In order to reduce false alarms, the extra input mask of the fine detection network is obtained as follows. First, we apply a morphological dilation operation with a disc structuring element (15 pixels in diameter) to the rough detection result of the previous and current frames. Then the two dilated binary images perform an AND operation. Figure 5 shows the inference phase of the fine detection network. If there is no rough detection result for the previous frame, the rough detection result for the current frame is dilated as the input mask. The outputs are saved in the same way as the rough detection network.



(**a**) (**b**)

Figure 4. Examples of the generated random rough detection results. For convenience of display, we have used red to mark the TV regions on the original image. (**a**) Ground truth of TV; (**b**) two examples of the generated rough detection result.

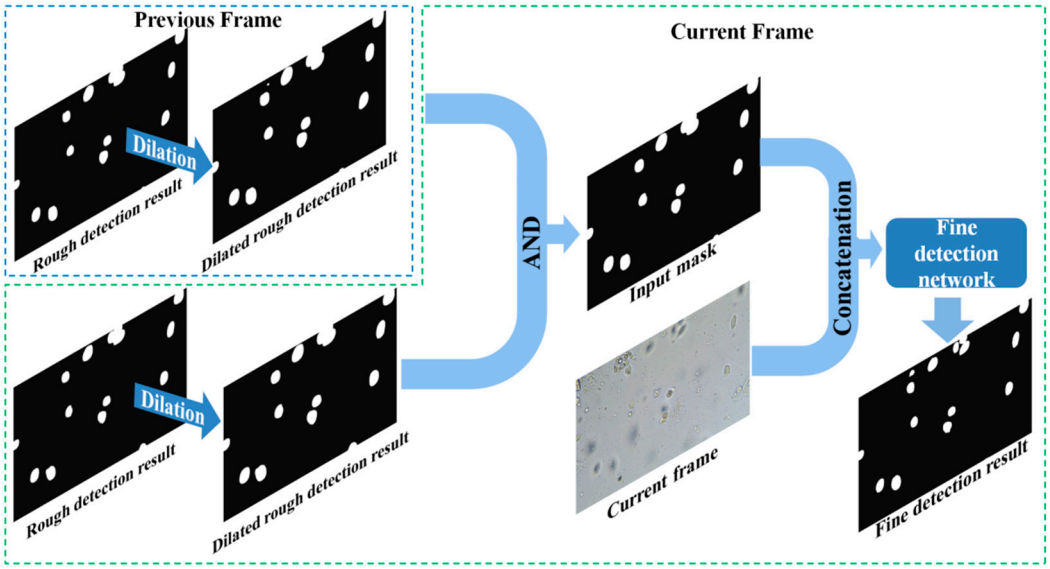


Figure 5. The inference phase of the fine detection network.

4. Experimental Results

4.1. Dataset and Optical System

There were six videos containing TV in our dataset and the frame number of each video is shown in Table 1. The image size of each frame was 1920 × 1200. All videos were shot under this condition: adjusting the z position of the microscope stage to make the formed elements clearest. The positions and shapes of the TV were constantly changing due to its motion characteristics and most of the time it appeared as shadow regions in the videos. For the convenience of comparison, we manually labeled the TV in all video frames, obtaining a total of 2520 annotated images for analysis (ground truth of TV, pixel value 0 for background regions, pixel value 1 for TV regions). For the two convolutional neural networks, we used video1 to video2 as the training set, video3 as the validation set, and video4 to video6 as the test set.

Table 1. The frame number of each video in our dataset.

Video Name	Video1	Video2	Video3	Video4	Video5	Video6
Frame number	221	250	406	498	433	712

As shown in Figure 6, the optical system for capturing TV videos contains a biological microscope and a charge-coupled device (CCD) camera. We used a CX31 biological microscope (Olympus, Tokyo, Japan) equipped with a 40× objective lens (CFI BE2 Plan Achromat, Nikon, Tokyo, Japan) which has a numerical aperture (NA) of 0.65. An EXCCD01400KMA CCD camera (Motic, Xiamen, China) with a pixel size of 6.45 μm × 6.45 μm is used for exposure and the exposure time was 40 ms. The field of view (FOV) was 0.41 mm × 0.26 mm.

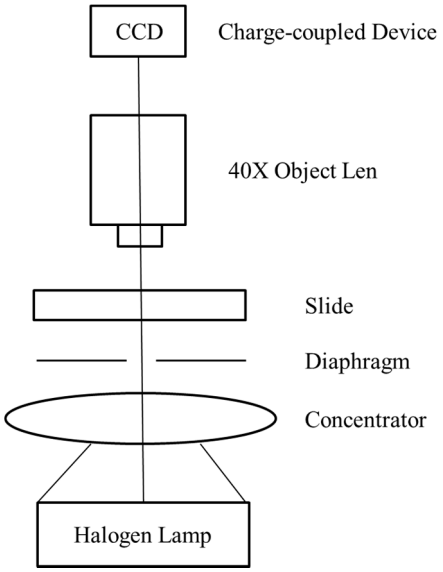


Figure 6. The optical system for capturing TV videos.

4.2. Metric

In this study, we verified the effectiveness of the proposed detection networks by calculating the intersection over union (IoU) metric of TV. The calculation formula was as follows:

$$\text{IoU} = \frac{TP}{TP + FP + FN} \tag{1}$$

where True Positive (*TP)* is the number of correctly detected TV pixels; False Positive (*FP)* is the number of background pixels incorrectly classified as TV; and False Negative (*FN)* is the number of TV pixels incorrectly classified as background. In addition, we calculated the precision and recall metrics to evaluate the degree of false alarm and missed detection of TV. The calculation formula was as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

4.3. Results

The results of the rough detection network for the training, validation and test set are shown in Table 2. It can be seen that for the test set, the rough detection network has a higher value of mean recall, but the increasing false detection regions lead to a decrease in mean precision value and mean IoU value.

Table 2. Results of the rough detection network.

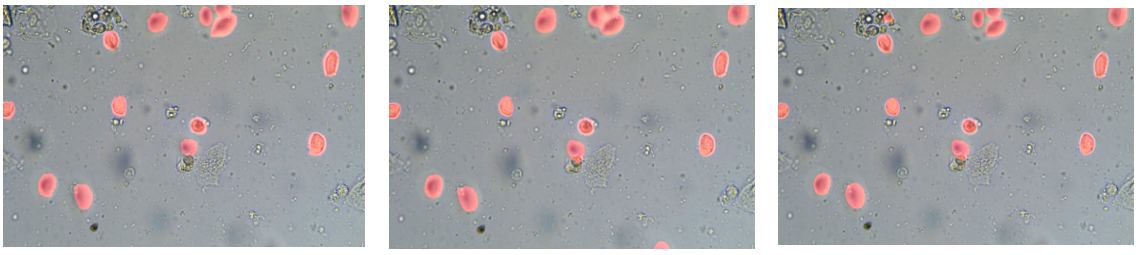
	Video Name	Mean Recall	Mean Precision	Mean IoU
Training set	video1	92.76%	85.81%	80.04%
	video2	90.71%	84.96%	77.91%
Validation set	video3	84.62%	83.66%	72.08%
Test set	video4	89.84%	74.28%	67.94%
	video5	92.73%	75.79%	71.32%
	video6	89.44%	78.26%	71.51%

The results of the fine detection network are shown in Table 3. With the fine detection network applied for the validation and test sets, the mean recall value decreases slightly, the mean precision value and mean IoU value are improved and the mean average IoU value of three test videos achieves 72.09%. The experimental results indicate that the proposed fine detection network can correct the boundary of TV. The mean IoU value in the training set is high and the boundaries of the TV are very close to the ground truth. The correction effect for TV is limited, however, the shadow region of false alarms after correction is enlarged and reduces the mean IoU value of video2. The specific analysis is discussed in Section 5.

Table 3. Results of the fine detection network.

	Video Name	Mean Recall	Mean Precision	Mean IoU	Mean Average IoU
Training set	video1	89.35%	89.35%	80.19%	78.14%
	video2	90.38%	84.14%	76.09%	
Validation set	video3	84.20%	85.33%	73.12%	73.12%
Test set	video4	89.65%	78.86%	70.13%	72.09%
	video5	92.06%	78.59%	73.34%	
	video6	86.45%	82.30%	72.80%	

Figure 7 shows the results of the two detection networks for one frame in video4. In this image, there are three TV adjacent to each other. In the rough detection result, the prediction regions of the three TV are stuck together. After using the fine detection network, some of the adhesion areas are eliminated. We uploaded the results of the fine detection network for the six videos online and the details can be found in the Supplementary Information.



(**a**) (**b**) (**c**)

Figure 7. Comparison between the rough detection result and the fine detection result. For convenience of display, we have used red to mark the TV regions on the original image. (**a**) Ground truth of TV; (**b**) rough detection result; (**c**) fine detection result.

4.4. The Operating Environment and Running Times

We used tensorflow2 framework to build our algorithm. The operating system is Ubuntu and we run this algorithm on a GTX TITAN Xp GPU. The code for calculating optical flow is from LiteFlowNet2 [13]. The overall detection starts from the second frame

of the video. One calculation process includes image reading and scaling (1920 × 1200 × 3 -> 1536 × 1024 × 3), optical flow calculation, slicing (15 × 512 × 512 × 13 or 15 × 512 × 512 × 4), rough detection and fine detection. In the process of inference, the batch size of the rough and fine detection network is 8 each time. The average running times for optical flow calculation, slicing, rough and fine detection are shown in Table 4.

Table 4. The average running times for one calculation process of our algorithm.

	Optical Flow Calculation	Slicing	Rough Detection	Fine Detection	Total
Running times	0.545 s	0.050 s	1.556 s	0.875 s	3.026 s

5. Discussion

5.1. Selection of the Optical Flow Calculation Method

In this section, we discuss the choice of optical flow calculation method. The traditional optical flow calculation method has a high calculation accuracy, but the high computation cost make itunsuitable for real-time detection. The optical flow calculation method based on deep learning has the advantage of fast calculation speed and acceptable precision, which has been an important research subject in respect of deep learning in recent years.

As shown in Figure 8, we compared 4 optical flow calculation methods which are based on deep learning. Figure 8a,b show the two adjacent frames, (c) shows the TV ground truth of frame (a), and frames (d) to (g) are the visualized images of optical flow calculated using the 4 methods. It can be seen from Figure 8d,e that flownet2.0 [19] and LiteFlowNet2 [13] can effectively capture the motion of the shadow regions where the defocused TV is located. Finally, we chose the LiteFlowNet2 [13] method, which has the faster calculation speed.

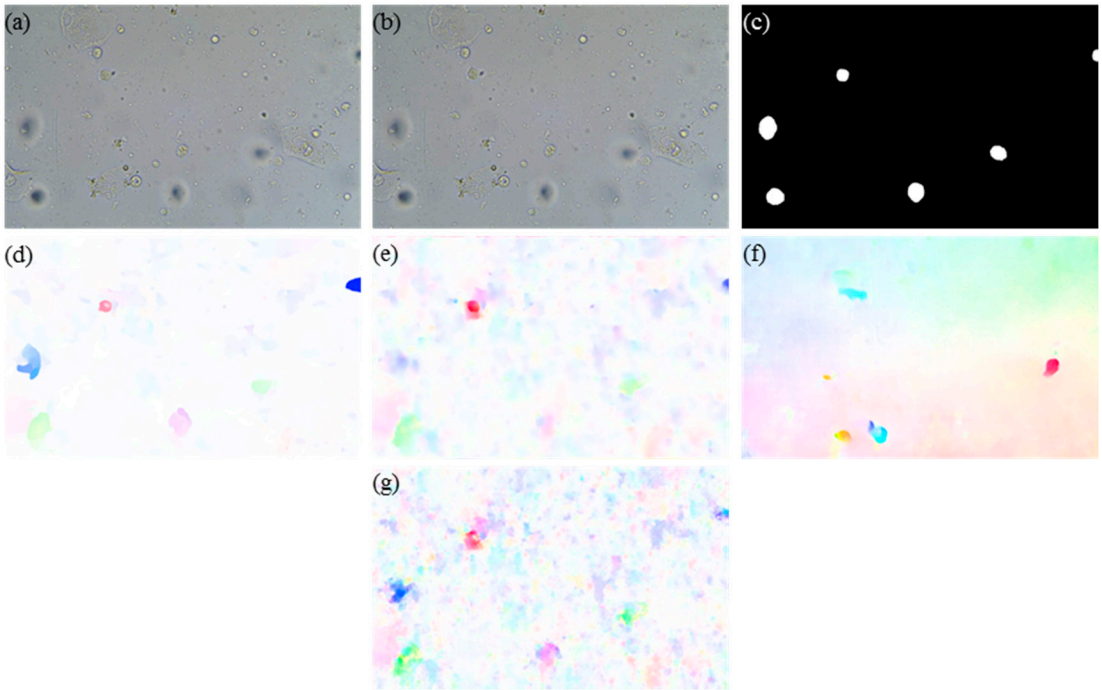


Figure 8. Comparison between 4 optical flow calculation methods based on deep learning. (**a**,**b**) show the two adjacent frames; (**c**) shows the TV ground truth of frame (**a**); (**d**–**g**) are the visualized images of optical flow calculated by Flownet2.0 [19], LiteFlowNet2 [13], Pwcnet [20], and Spynet [21].

5.2. Ablation Study

In this section, we investigate the effects of each component in the model framework of the rough detection network.

5.2.1. Network Architecture

Table 5 summarizes the differing performance of rough detection using different encoder or adding attention blocks. We chose the original FlowNetSimple network [11] as the baseline model. After modifying the encoder to VGG16 [12], the optimal value of the mean IoU for the validation set achieved a significant improvement to 71.50%. Compared with the original decoder structure, the attention block with the SE module [14] is able to recover fine details for TV and improve the mean IoU value to 72.08%. Finally, VGG16 [12] was used as the encoder and attention blocks with a SE module were adopted in the decoder.

Table 5. Differing performance of rough detection using different encoder or adding attention blocks.

	Architecture Variant	**The Optimal Value of Mean IoU on Validation Set (Video3)**
FlowNetSimple [11]	None	68.19%
Encoder changed	VGG16 [12]	71.50%
	Resnet50 [22]	68.31%
	modified Xception [23]	64.99%
Attention blocks added	VGG16 [12] + SE [14]	72.08%

5.2.2. Attention Module

The influence of different attention modules on the performance of rough detection network is shown in Table 6. We replaced the SE [14] module in the attention block with other classical attention modules such as non-local [24], CBAM [25], and dual attention (DA) [26]. Due to limitations in the memory size of the graphics boards, we deleted the attention blocks before 'output1′, 'output2′ and 'output3′. Similar to the training phase of the rough detection network, we used the trained network without an attention mechanism as the pretrained model for transfer learning. The data in Table 6 indicate that the SE module has the ability to identify the information pertinent to the TV with a better performance.

Table 6. The influence of different attention modules on the performance of the rough detection network.

Attention Module Added	**Video3**	**Video4**	**Video5**	**Video6**
None	71.50%	67.75%	70.60%	69.76%
+SE [14]	72.08%	67.94%	71.32%	71.51%
+Non Local [24]	71.73%	66.64%	69.65%	67.81%
+CBAM [25]	70.75%	66.82%	70.62%	71.25%
+DA [26]	71.16%	66.90%	70.80%	70.37%

5.2.3. Network Inputs

Table 7 summarizes how the input information affects the performance of the rough detection network. In order to simplify the comparison process, we used VGG16 [12] as the encoder and removed the attention blocks in the decoder. The results demonstrate that optical flow information is necessary and using three consecutive frames could obtain a better result than two. We also tested the inputs of five consecutive frames with optical flow, but the network performance for the test set decreased. Finally, three consecutive frames with optical flow were adopted as the input of the rough detection network.

Table 7. The influence of different input information on the performance of the rough detection network.

Input Information	Video3	Video4	Video5	Video6
Two adjacent frames (6 channels)	71.52%	66.51%	67.49%	68.18%
Two adjacent frames with optical flow (8 channels)	70.94%	67.87%	69.51%	67.92%
Three consecutive frames with optical flow (13 channels)	71.50%	67.75%	70.60%	69.76%
Five consecutive frames with optical flow (23 channels)	71.86%	66.86%	67.86%	67.12%

5.3. Comparison with Traditional Video Object Detection Methods

In this section, we compare our method with some traditional video object detection methods. For convenience of comparison, we have only compared the mean IoU metrics. In order to reduce the interference of background noise, we first used a median filtering algorithm on the images (11×11 kernel for Three frame difference [4] and 29×29 kernel for Gaussian Mixed Model (GMM) model [5]). As shown in Table 8, the mean IoU value of our method is much higher than the other traditional algorithms. Since most of the TV in our dataset is defocused, it appears as flat shadow regions with little difference between adjacent frames. Therefore, whether using the frame difference method [4] or the background difference method [5], it is necessary to reduce the judgment threshold of the foreground, resulting in false alarms where some backgrounds are detected as TV. The above problem also exists with the improved Kalman [1] and improved VIBE [2] methods that we proposed. These two methods can recognize clear instances of TV, but fail on defocused images.

Table 8. The comparison between our method and traditional video object detection methods.

Method	Video1	Video2	Video3	Video4	Video5	Video6
Three frame difference [4]	34.60%	28.07%	27.39%	32.33%	33.34%	39.23%
GMM model [5]	45.51%	35.26%	34.87%	39.97%	44.99%	36.76%
Improved Kalman [1]	44.52%	39.87%	39.48%	42.95%	47.93%	51.15%
Improved VIBE [2]	55.51%	51.74%	53.29%	56.82%	58.71%	57.73%
This paper	80.19%	76.09%	73.12%	70.13%	73.34%	72.80%

5.4. The Performance of the Rough Detection Network Using Different Outputs

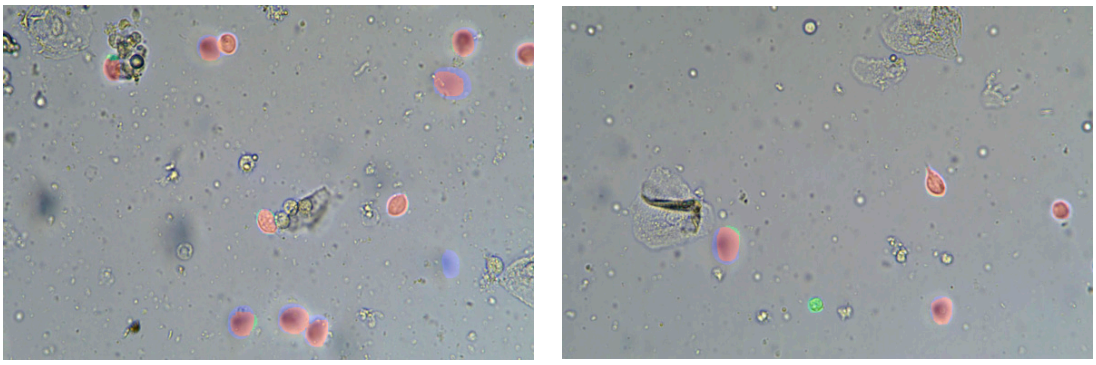
We only used 'output1′ as the final result, although there are five 'output' for the rough detection network. Therefore, we compared the impact of different 'output' on the performance of the rough detection network. For 'output2′ to 'output5′, we enlarged the image size to 512×512 by bilinear interpolation. The mean IoU values of 'output1′ to 'output5′ are shown in Table 9. The results of 'output1′, 'output2′, and 'output3′ are similar. In order to improve the detection speed, the results of 'output3′ to 'output5′ were calculated alone while discarding the subsequent network structure. Furthermore, due to the reduction in the network size, we stack the cropped patch images together and a rescaled image with the shape of 1536×1024 could be detected by the rough detection network immediately.

Table 9. The mean IoU values of 'output1′ to 'output5′ for rough detection network.

Output	Video1	Video2	Video3	Video4	Video5	Video6
output1	80.04%	77.91%	72.08%	67.94%	71.32%	71.51%
output2	80.02%	77.95%	72.00%	67.85%	71.28%	71.25%
output3	79.41%	77.40%	71.64%	67.53%	71.05%	70.83%
output4	76.19%	74.77%	70.76%	65.78%	68.84%	68.15%
output5	75.96%	73.40%	70.42%	65.30%	68.40%	68.09%

5.5. Limitations of Our Trichomonas Vaginalis Detection Method

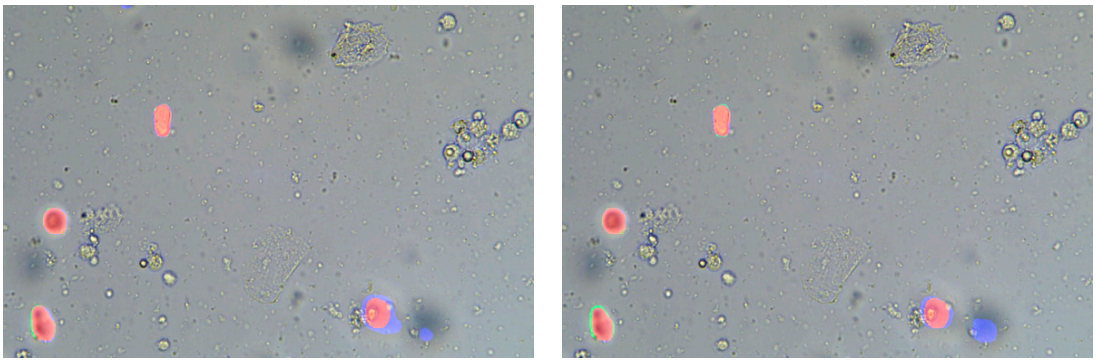
Most of the TV in the training set videos is defocused and therefore the rough detection network is sensitive to the shadow regions between frames, which often lead to false alarms. As shown in Figure 9a, the shadow region of the background with the blue marks moves slightly with the sample liquid and is mistakenly detected by the rough detection network at the bottom right of the image. In addition, the principle of the rough detection network mainly depends on the difference between frames. As shown in Figure 9b, the TV with the green mark is similar to white blood cells and its position is basically unchanged in this video, leading to little difference between frames and missed detection.



(**a**) (**b**)

Figure 9. The limitation of the rough detection network. For convenience of display, we have used red to mark the *TP* regions, green to mark *FN* regions and blue to mark *FP* regions of TV on the original image. (**a**) The false alarms of TV; (**b**) the missed detection of TV.

The main function of the fine detection network we proposed is to correct the contours of TV and eliminate the short-term false alarms. Therefore, if the rough detection network mistakenly detects or misses the TV, the result will not improve or get worse with the fine detection network applied. For example, the fine detection network could not detect the missed TV in Figure 9b. Figure 10 shows the rough and fine detection results for one frame in video2. The blue-marked shadow region at the bottom right of Figure 10a is falsely detected by the rough detection network. After adopting the fine detection network, as shown in Figure 10b, it cannot be eliminated but is enlarged by the correction function of the fine detection network.



(**a**) (**b**)

Figure 10. Limitations of the fine detection network. For convenience of display, we have used red to mark the *TP* regions, green to mark *FN* regions and blue to mark *FP* regions of TV on the original image. (**a**) The rough detection result; (**b**) the fine detection result.

In future work, we need to address the above limitations and further study the problem of TV recognition in a flowing liquid samples.

6. Conclusions

In this paper, we proposed two convolutional neural networks based on an encode-decoder architecture to solve the problem of defocused TV recognition in videos shot by microscopic cameras. The first rough detection network we proposed realizes the coarse detection of the TV by learning the difference between adjacent frames. The second fine detection network we proposed achieves correction of the contours of TV for rough detection results. By combining these two networks, the mean average IoU value of the TV achieved 72.09% for our test videos. The experimental results show that our proposed networks can effectively detect defocused TV and suppress the false alarms caused by the motion of formed elements or impurities.

Supplementary Materials: The fine detection results for six videos are available online at www.github.com/wxz92/Trichomonas-Vaginalis-Detection.

Author Contributions: Investigation, L.L.; resources, G.N.; data curation, G.N.; methodology, X.W.; writing—original draft preparation, X.W.; writing—review and editing, X.D.; project administration, Y.L.; funding acquisition, J.Z. and J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (No. 61905036), the Fundamental Research Funds for the Central Universities (University of Electronic Science and Technology of China) (No. ZYGX2019J053) and China Postdoctoral Science Foundation (2019M663465).

Institutional Review Board Statement: The study was conducted according to the guidelines of the Declaration of Helsinki, and approved by the Institutional Review Board of University of Electronic Science and Technology of China (protocol code: 106142021030903).

Informed Consent Statement: Written informed consent has been obtained from the patients to publish this paper. All samples were anonymization.

Data Availability Statement: The algorithm codes and our dataset will be released online at www.github.com/wxz92/Trichomonas-Vaginalis-Detection (accessed on 24 February 2021).

Acknowledgments: We would like to express our thanks to Yu-Tang Ye and the staff at the MOEMIL laboratory, who collected and counted the samples used in this study.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hao, R.; Wang, X.; Zhang, J.; Liu, J.; Ni, G.; Du, X.; Liu, L.; Liu, Y. Automatic detection of trichomonads based on an improved Kal-man background reconstruction algorithm. *JOSA A* **2017**, *34*, 752–759. [CrossRef] [PubMed]
2. Du, X.; Liu, L.; Wang, X.; Zhang, J.; Ni, G.; Hao, R.; Liu, J.; Liu, Y. Trichomonas Detection in Leucorrhea Based on VIBE Method. *Comput. Math. Methods Med.* **2019**, *2019*, 1–10. [CrossRef] [PubMed]
3. Jain, R.; Nagel, H.-H. On the Analysis of Accumulative Difference Pictures from Image Sequences of Real World Scenes. *IEEE Trans. Pattern Anal. Mach. Intell.* **1979**, *1*, 206–214. [CrossRef] [PubMed]
4. Yang, M.-M.; Guo, Q.-P. A motion detection algorithm based on three frame difference and background difference. In Proceedings of the 8th International Symposium on Distributed Computing and Applications to Business Engineering and Science, Wuhan, China, 16–19 October 2009; pp. 487–490.
5. Stauffer, C.; Grimson, W. Adaptive background mixture models for real-time tracking. In Proceedings of the 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149), Fort Collins, CO, USA, 23–25 June 2003; Institute of Electrical and Electronics Engineers (IEEE): New York, NY, USA; Volume 2, p. 257.
6. Barnich, O.; Vanogenbroeck, M. ViBe: A powerful random technique to estimate the background in video sequences. In Proceedings of the IEEE International Conference on Acoustics, Speech & Signal Processing, Taipei, Taiwan, 19–24 April 2009; pp. 945–948.
7. Xu, L.; Jia, J.; Matsushita, Y. Motion Detail Preserving Optical Flow Estimation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 1744–1757. [CrossRef] [PubMed]

8. Zhu, X.; Xiong, Y.; Dai, J.; Yuan, L.; Wei, Y. Deep Feature Flow for Video Recognition. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4141–4150.
9. Zhu, X.; Wang, Y.; Dai, J.; Yuan, L.; Wei, Y. Flow-Guided Feature Aggregation for Video Object Detection. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 408–417.
10. Zhu, X.; Dai, J.; Yuan, L.; Wei, Y. Towards High Performance Video Object Detection. In Proceedings of the 2018 IEEE/Cvf Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 7210–7218.
11. Dosovitskiy, A.; Fischer, P.; Ilg, E.; Hausser, P.; Hazirbas, C.; Golkov, V.; Van Der Smagt, P.; Cremers, D.; Brox, T. FlowNet: Learning Optical Flow with Convolutional Networks. In Proceedings of the 2015 IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 2758–2766.
12. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
13. Hui, T.W.; Tang, X.; Loy, C.C. A lightweight optical flow CNN-revisiting data fidelity and regularization. *arXiv* **2019**, arXiv:1903.07414. [CrossRef] [PubMed]
14. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2018; pp. 7132–7141.
15. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. In Proceedings of the International Conference Learn, San Diego, CA, USA, 5–8 May 2015.
16. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollar, P. Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 2980–2988.
17. Perazzi, F.; Khoreva, A.; Benenson, R.; Schiele, B.; Sorkine-Hornung, A. Learning Video Object Segmentation from Static Images. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) IEEE, Honolulu, HI, USA, 21–26 July 2017; pp. 3491–3500.
18. Bookstein, F.; Green, W.D.K. A Thin-plate splines and the decomposition of deformations. *Math. Methods Med. Imaging* **1993**, *2*, 14–28.
19. Ilg, E.; Mayer, N.; Saikia, T.; Keuper, M.; Dosovitskiy, A.; Brox, T. FlowNet 2.0: Evolution of Optical Flow Estimation with Deep Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; Institute of Electrical and Electronics Engineers (IEEE): New York, NY, USA; pp. 1647–1655.
20. Sun, D.; Yang, X.; Liu, M.-Y.; Kautz, J. PWC-Net: CNNs for Optical Flow Using Pyramid, Warping, and Cost Volume. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8934–8943.
21. Ranjan, A.; Black, M.J. Optical flow estimation using a spatial pyramid network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4161–4170.
22. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. *arXiv* **2015**, arXiv:1512.03385.
23. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic im-age segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 801–818.
24. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local Neural Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; Institute of Electrical and Electronics Engineers (IEEE): New York, NY, USA; pp. 7794–7803.
25. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the European conference on computer vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
26. Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; Lu, H. Dual Attention Network for Scene Segmentation. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; Institute of Electrical and Electronics Engineers (IEEE): New York, NY, USA; pp. 3141–3149.