IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 23, 2019, accepted October 5, 2019, date of publication November 5, 2019,
date of current version December 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2951723

Statistical Karyotype Analysis Using CNN and Geometric Optimization

NING XIE[1], XU LI[2], KANG LI[1], YANG YANG[1], AND HENG TAO SHEN[1]

[1] Center for Future Media, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
[2]Glasgow College, University of Electronic Science and Technology of China, Chengdu 611731, China

Corresponding author: Ning Xie (seanxiening@gmail.com)

This work was supported in part by the National Nature Science Foundation of China under Grant 61602088, Grant 61572108, and Grant 61632007, and in part by the Fundamental Research Funds for the Central Universities under Grant Y030190236010080011.

ABSTRACT Karyotype analysis is one of the main techniques of cytogenetics through medical image processing, which has an important role in modern medical treatment and diagnosis. The process of human karyotype analysis contains two key components: Firstly, chromosomes are segmented from metaphase chromosome digital images taken under a microscope. Chromosomes then are analyzed, compared, ordered and classified one by one carefully. Under this procedure, the operation on segmentation and classification is cumbersomely time-consuming, where traditional geometric or statistical methods only have limited effect due to low accuracy. Thus, in most conditions, human effort is still heavily required to monitor the workflow and correct the errors. In this paper, we present an integrated workflow to segment out and classify chromosomes automatically using a combination of multiple input convolutional neural networks (CNN) and geometric optimization, called mCNN_GO. We investigate Mask R-CNN to segment out chromosomes from metaphase chromosome images and train the mCNN_GO to classify the sub-images. To improve the performance of the segmentation network, we adapt a new feature-based approach to synthesize images on the labeled data. Furthermore, we develop a geometric algorithm to straighten the chromosomes before classification to ensure the consistency of the data. Experimental results demonstrate that our approach significantly outperforms the state-of-the-art methods on automatic karyotype analysis.

INDEX TERMS Biomedical image processing, image processing, machine learning.

I. INTRODUCTION

Medical image processing has become an important assistant tool of diagnosis and therapy. Nowadays, human karyotype analysis is of great significance for the clinical diagnosis of genetic diseases. Karyotype analysis is one of the main techniques of cytogenetics, including analyzing, comparing, sequencing and numbering of metaphase chromosomes through the captured chromosomes digital images. Before karyotyping, cells need to be cultured. As shown in Figure 1, when chromosomes are at the metaphase, they will be separated from the nucleus and stained onto glass slides for observation and photography under the optical microscope. The division and classification process of chromosomes will be done by professional doctors with the help of auxiliary equipment. This set of operation is time-consuming and laborious, which is mainly due to the following difficulties:

- Each piece of the captured image contains massive chromosomes. They need to be observed by laboratory doctors one by one. Especially, the cases with impurities, overlapping area and contacting will significantly increase the difficulty of the operation;
- After segmentation, the classification of chromosomes is also exhaustive and labor-intensive. In clinic, the chromosomes are often curved or bent heavily, which makes it harder to verify the classes of chromosomes.

To minimize the human effort consumption in karyotype analysis, Sharma *et al.* [1] provided a chromosome segmentation method based on crowdsourcing. This is the first who used CNN on chromosome classification. They also proposed several preprocessing methods such as the straightening and length normalization. However, the results could not meet the requirements of automatic karyotype analyzing. The accuracy of classification is relatively low and unsatisfactory. To improve the accuracy of chromosome segmentation and

The associate editor coordinating the review of this manuscript and approving it for publication was Yongtao Hao.

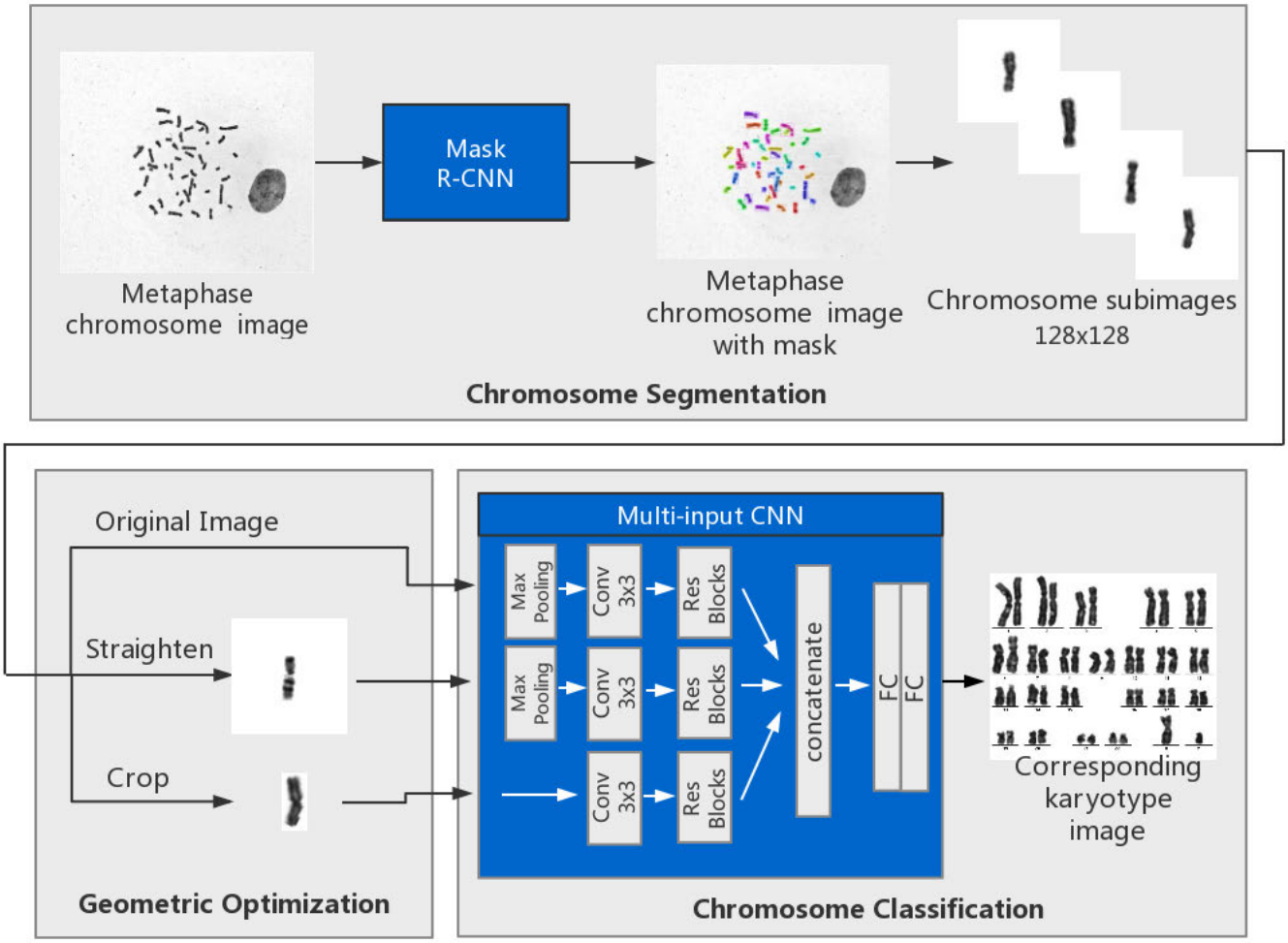


FIGURE 1. The pipeline of our system, includes: 1) Chromosome Segmentation 2) Geometric Optimization and 3) Chromosomes Classification.

classification, we make significant improvements to the existing approaches and increase the accuracy and availability of automatic karyotype analyzing.

In this paper, we provide an approach for performing karyotype analysis using deep learning. The metaphase chromosome images are fed into a segmentation network for instance segmentation, and then each single chromosome will be extracted from the whole picture. After that, we investigate a multi-input CNN to classify the extracted chromosomes into 24 classes for karyotype analysis. For certain cases where the extracted chromosomes are curved, we develop a geometric algorithm to straighten each chromosome before classification to ensure the consistency of the training data. The straighteng method is based on the medial axis of chromosome. This preprocessing is proved to improve the accuracy of classification. The main contributions of this paper are as follows:

1) This is the first segmentation network to accomplish the task of instance segmentation of metaphase chromosome images. Mask R-CNN [2] is investigated on the pixel-level segmentation.
2) The straightening algorithm based on the medial axis of chromosome is proposed, which considerably increase the classification accuracy. It improves the usability of our system on the clinic samples.
3) Using a multi-input CNN to classify the chromosomes, the accuracy has been significantly improved comparing to the classic baselines and predecessors.

II. RELATED WORKS

Karyotyping of chromosomes can be regarded as the task including two steps, respectively, segmentation and classification. To mitigate the human effort in karyotyping, numerous methods for segmentation and classification have been proposed.

A. CHROMOSOME SEGMENTATION

Segmenting out chromosomes from the meta-phase chromosome image is primarily crucial, because it will directly affect the accuracy of following classification process. Pham *et al.* [3] surveyed on the traditional segmentation methods widely used in medical image segmentation.

Charters and Graham [4] provided an algorithm to segment according to the comparison between the band profiles and the templates. Sharma *et al.* [1] used crowdsourcing to segment out the chromosomes and represented a method to evaluate the results. Many approaches based on the heuristic rules and geometry are also represented, such as [5]–[7]. But they had low robustness under the clinic conditions, where the shapes and distribution of chromosomes usually vary greatly. Furthermore, in clinic, the overlapping and contacting of chromosomes often appear, while the traditional methods like [8], [9] are highly depend on manual intervention and preset parameters, which means that they had certain limitation when the condition changes.

As the advent of deep learning, CNN have achieved excellent performance on many computer vision tasks. In medical image, scholars also tended to apply CNN for image segmentation. Long *et al.* [10] first used Fully Convolutional Networks (FCN) to do image segmentation. BenTaieb and Hamarneh [11] optimized the structure of FCN and successfully used it on medical image processing. Medical image segmentation network U-net offers high accuracy while training on a small amount of annotated data, and have superior performance on *Hela* cell segmentation and neuron segmentation tasks [12]. However, most of the existing models are designed for semantic segmentation, while karyotyping needs to segment out each instance of chromosomes. In this paper, we first use deep learning for chromosome segmentation, and we compared three widely used models: Mask R-CNN [2], FCIS [13] and YOLOv3 [14]. YOLOv3 can be trained and run fast while keeping high accuracy, especially on detecting tiny objects. However, YOLOv3 can only predict labels on bounding box level, while our task needs pixel-level prediction because chromosomes can be intensive in a small area. FCIS can generate masks at pixel-level, but the accuracy of it is lower than Mask R-CNN in most conditions. Thus, we choose Mask R-CNN as our segmentation network and the results show that it works well on the chromosome segmentation task.

A major impediment in deploying deep learning on chromosome detection and segmentation is the lack of large annotated data. Annotation in pixel-level is daunting for microscopic images like the metaphase chromosome image. The widely used method to collect annotated data is crowdsourcing [1]; however, the quality of these crowdsourcing data is hard to ensure, because the annotation of medical images is a tough task and requires operators to have a wealth of experience. As the advent of *generative adversarial networks (GAN)*, many synthesis methods based on GAN had been developed [15], [16]. However, these methods could only generate annotated images for semantic segmentation. They are not feasible for distinguishing instances. Dwibedi *et al.* [17] represented an approach to automatically extract object images and pasted them on backgrounds to synthesize training data with bounding box labels. This process could collect large amount of data rapidly for instance detection. Inspired from this work, we propose a synthesis method to generate pixel-level annotated data for chromosome segmentation. Extensive experiments demonstrated that our synthetic data has high realism so as to improve the segmentation performance considerably.

B. CHROMOSOME CLASSIFICATION

Chromosome classification is a well studied problem. Earlier approaches, such as [18], were highly relied on geometric features (like the length of chromosomes or the position of centromere) and the banding profiles of chromosomes. Robust performance started from introducing *medial axis transformation (MAT)* into chromosome classification [19]. As follows, many MAT-based studies have been conducted [20], [21]. Also, *multi-layer perceptron (MLP)* was applied to make the predictions [19].

Then, many CNN-based methods on chromosome classification have been developed. Sharma *et al.* [1] first used CNN to classify chromosome. In this method, the preprocessing procedure was also proposed to straighten those curved chromosomes. Their model was trained under 1600 individual images and tested on 200 images. The accuracy on preprocessed images is 86.7%. Because of the limitation of labeled images, Jindal *et al.* [22] represented a method based on *Siamense Network*, which has good performance under scarce training data. In detail, chromosomes were straightened by two different approaches in parallel before sending to Siamense Network. The accuracy on 209 images achieved 84.6% after being trained on 1296 images. Wu *et al.* [23] then proposed a *Multiple Distribution Generative Advertising Network (MD-GAN)* to generate labeled data and used a pre-trained CNN to classify chromosomes with samples generated by MD-GAN. However, their average precision only achieved 63.5%, which is further below the requirement for clinic application. One main challenge is the lack of labeled data. This may lead to overfitting, low accuracy and low robustness. Another challenge is that, the performance of learning model highly relies on the results of chromosome straightening. Our framework is built on the multi-input CNN [24]. It can extract features from not only preprocessed images, but also original images.

III. OVERVIEW

Figure 1 illustrates the whole workflow of our system. In the first stage, we input the original *Giemsa* stained images into the segmentation network. The mask of each individual chromosome is calculated as its output. Then, system extracts the chromosomes to sub-images separately. In the second stage, each sub-image will be geometrically optimized by cropping, rotation and straightening. In the last stage, the original and optimized sub-images are fed into our classification network. The entire workflow is fully automatic and has no need for human intervention. Due to the high complexity and uncertainty in clinic conditions, the accuracy of our system can vary widely in some most extreme case. Instead of replacing doctors during the karyotype analysis completely, we aim to optimizing the processing by minimizing doctors' load.

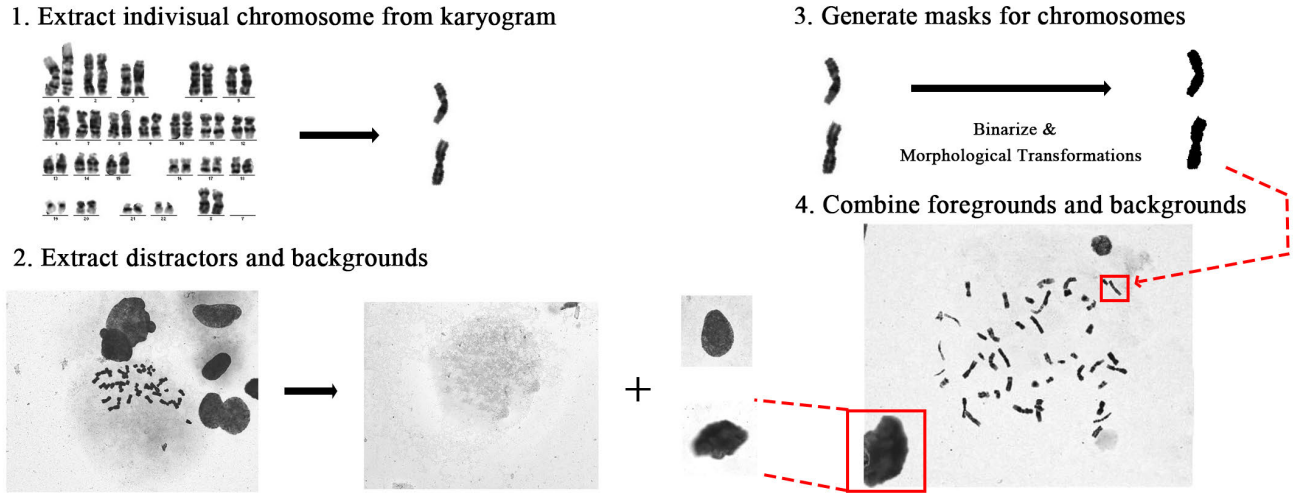


FIGURE 2. A simple but effective method to synthesize pixel-level annotated images for chromosome instance segmentation.

IV. METHODOLOGY

A. DATA COLLECTION AND SYNTHESIS

The dataset of segmentation consists of two groups: the original images and the synthetic images. The original set is provided by a hospital. Annotation is done manually. Due to huge quantity of the dataset, it is hard to annotate them all. In practical, we adapt and improve the 'cut and paste' image synthesis method developed by [17] to synthesize the annotated images. By doing so, it saves much human effort. Nevertheless, we demonstrate that all synthetic images are annotated in pixel level. The main steps of our method (as illustrated as Figure 2) are as follows:

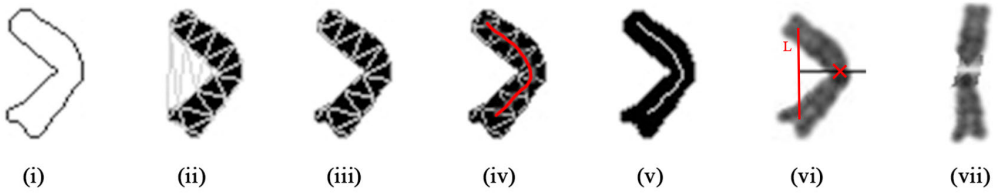
1) **Chromosome image collecting.** The sub-images of chromosome are extracted from the ordered *karyogram* provided by hospital in grayscale. We collect 5000 sub-images. Each one only contains a single chromosome in 128×128.
2) **Distractors and background image collecting.** We collect 50 kinds of distractors and 20 backgrounds without chromosomes from metaphase Giemsa stained images in grayscale.
3) **Foreground mask synthesizing.** Mask of each chromosome is synthetized by binarizing sub-images. Meanwhile, we calculate its bounding polygon.
4) **Foreground and background assembling**: 48 chromosomes and 2 to 6 distractors are chosen and pasted on a background randomly. To prevent unnecessary overlapping and obstruction, chromosomes are placed on the central area of backgrounds, while distractors are more intensive on the edge area. According to Dwibedi *et al.* [17], using various modes of blending to synthesize same images can increase the robustness of the training model. Thus, three different blending patterns are applied including Gaussian blurring, motion blurring, and box blurring.

B. SEGMENTATION USING MASK R-CNN

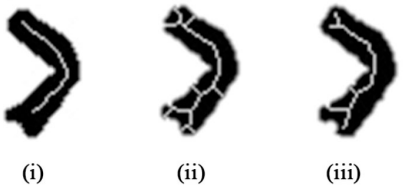
In the task of chromosome segmentation, we need to segment out every single chromosome from the complex meta-phase Giemsa staining image, where contains not only chromosomes, but also impurities. Moreover, chromosomes often overlap or contact on others. Our model needs to be powerful enough to complete following tasks: 1) Object detection, which means it should be able to detect the chromosomes from a messy background; 2) Pixel-level segmentation. Because the chromosomes can be intensive or intertwined, our model needs to distinguish and segment each chromosome precisely. In this part, we introduce Mask R-CNN to the task of chromosomes segmentation. Mask R-CNN is a general framework for object instance segmentation [2]. The backbone of Mask R-CNN is a standard convolutional neural network (*ResNet101* in our task) combined with a *Feature Pyramid Network (FPN)* as the component of feature extraction. *Region Proposal Network (RPN)* then scans the backbone feature map through sliding window so as to target areas called *anchors*, where contain objects. Using the RPN predictions, we can find the top anchors that are likely to contain objects and refine their configuration such as location and size. Finally, the *regions of interests (ROIs)* are calculated using RPN including ROI class, the optimal bounding box and the corresponding mask.

C. MEDIAL AXIS BASED GEOMETRIC OPTIMIZATION

In practical, chromosomes lay on the Giemsa staining image in disorder as illustrated in Figure 1, which makes classification process much harder and unstable. The straightening operation on chromosomes is often performed before classification. The results demonstrated that it can improve the accuracy of classification considerably [1], [22]. However, through the experiment on our dataset, we observe that these existing methods are only feasible on the chromosomes under



(a) (i)-(v): Locate the medial axis using Delaunay triangulation. (vi): Find the cutting point. (vii): Cut and rotate the upper and lower parts.



(b) (i): Our medial axis locating algorithm. (ii)(iii): Medial axis locating algorithms provided by scikit-image

FIGURE 3. **Illustration of our straightening algorithm.**

fairly flat gesture. The performance on the heavily curved ones is limited.

To solve it, we extend the existing the straightening approach via *Projection Vectors* [1] as follows:

1) **Medial axis extraction.** First, as shown in Figure 3(i)-(v), we binarize the chromosome sub-image and extract its contour. Then, sample the contour every α pixels and generate Delaunay triangles based on these sample points. After removing triangles outside the contour, we can get the medial axis **S** by connecting the center of triangles. The result shows that our approach is more clear and has fewer branches than the medial axis locating algorithm used in *scikit-image* [25].
2) **Cutting-point finding.** Javan-Roshtkhari and Setarehdan [26] found that, when rotating the original image of chromosome, the position of the bending point can be obtained from the change of histogram. However, when the width of the chromosome is relatively average or the total pixel amount is small, the histogram tends to be unchangeable during rotation and the bending point cannot be found. In our approach, we connect two ends of the medial axis S and get a line segment L (Figure 3(vi)). Because the two ends of the medial axis are actually the ends of the whole chromosome, the point farthest from L must be the bending point.
3) **Rotation.** After finding the cutting point P_{cut}, we can cut the image into two parts, and get the minimum bounding box $Bbox_l$ and $Bbox_h$:

$$Bbox_l = I\left[P_{low}, P_{cut}\right]$$
$$Bbox_h = I\left[P_{cut}, P_{high}\right]$$

where P_{low} and P_{high} means the end points of L. Then we can find the slope of $Bbox_l$ and $Bbox_h$ and rotate them to vertical to get $Bbox'_l$ and $Bbox'_h$.
4) **Pixel filling.** Because we want to maximize the horizontal banding features in the processed image. We apply a new interpolation method in our algorithm. For one pixel Pix^i_m:

$$Pix^i_m = \frac{1}{len(I_m)} \sum_{i=1}^{n} Pix^i_{en}$$

where I_m means the line vector of image I and Pix^i_{en} means effective ith pixel in the mth line. Experimental results on 90000 real images show that our method has high robustness and generality. As shown in Table 3, this method can improve the classification accuracy by two percentage points.

D. CHROMOSOME CLASSIFICATION USING MULTI-INPUT CNN

As mentioned in SectionII, we believe the low accuracy of existing classification models is because of the insufficient feature extraction and the changeable shape of chromosomes. Inspired by the multi-scale convolutional network developed by [24], we propose a multi-input CNN to classify the chromosomes, and the structure of our model is shown in Fig.1. Our mCNN_GO (multi-input CNN with Geometric Optimization) model takes three inputs at the same time: original image, straightened image and cropped image. We extract global-level features by the Global-Net, and the Local-Net can extract local-scale features from the cropped image, meanwhile, we feed the straightened image to the Straighten-Net in parallel, which can improve the robustness in case the original image is highly curved. This mCNN_GO model is composed by three stages: 1) Feature extraction using ResNet; 2) Combination of the three feature maps; 3) Classification via a MLP classifier based on the merged feature map.

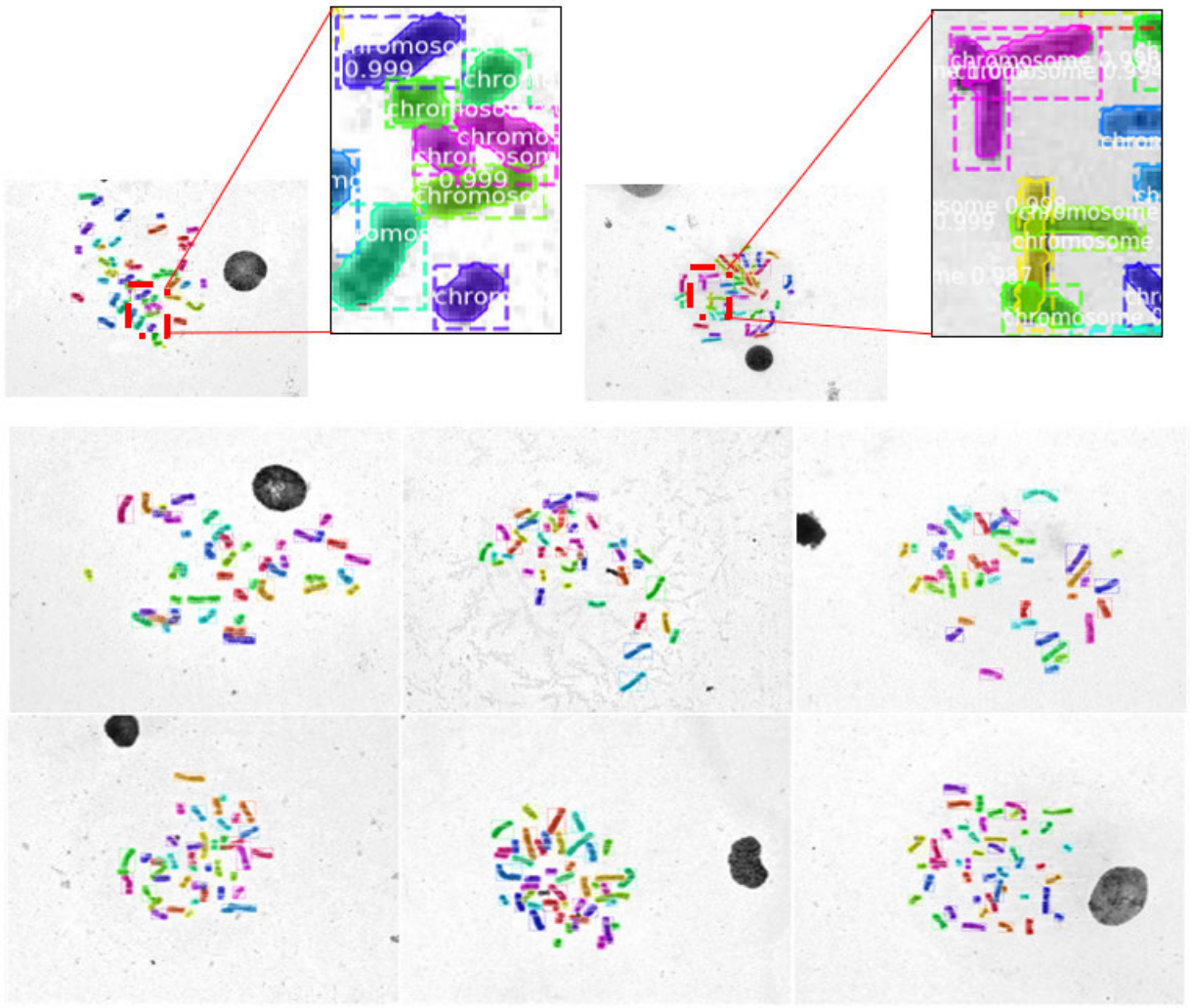


FIGURE 4. The mask generated by segmentation network. We can see that our model has high robustness and good performance on the segmentation of overlapping chromosomes.

Encouraged by the success of the residual learning [29], we apply *ResNet-50* as our backbone. The original image and straighten image will pass a maxpooling layer with a 2x2 kernel and a convolution layer with 3x3 kernel. The cropped image will be feed into ResNet directly. By doing this, we can maximum the extracted features.

Once the CNNs are optimized, the global-level, local-level and straightened features can be extracted and sent to the classifier. To make full use of the features, we build a concatenation layer with 4096 nodes to merge these features into a big feature map. Then a MLP classifier composed by two FC layers with 1024 nodes and one softmax layer will learn the correspondence between the features and the possibility of each classes.

To optimize this model, we first train the Global-Net, Local-Net and Straighten-Net individually until convergence and fine-tuning the models. Then, we combine them together. Fix the parameters of *ResNet* blocks and just train the upper layers. Then unlock all parameters and fine-tune the whole model under a very small learning rate (0.0005).

V. RESULTS AND DISCUSSION

A. RESULTS ON SEGMENTATION

In this section, we introduce the experimental results on the segmentation. As illustrated in Table 1, we set up six training datasets and two validation datasets by combining real data and synthetic data with multiple scheme of ratios. We train 6 models on each train set individually. Our models are built based on and [31]. The hyper-parameters is tuned during the process. The sizes of anchors of RPN is set to [8, 16, 32, 64, 128]. The aspect ratio is set to [0.5, 1, 2]. To ensure the accuracy of ROI extraction, the threshold of IoU is set to 0.7. We load the weights pre-trained on COCO dataset and just fine-tune the top layers. Adam optimizer is used for training with β_1=0.9, β_2=0.999. The learning rate is initialized to 0.001. Each model is trained for 60 epochs on

TABLE 1. Experimental results of our segmentation network trained on different datasets. *AP* means computing average precision over a IoU thresholds of 0.5-0.95; AP_{50} means average precision on IoU threshold 0.5. Results show that our segmentation network has a good performance, and the synthetic images can improve the accuracy of our network.

Train \ Test	100 Real (AP)	100 Real+100 Syn (AP)	100 Real (AP_{50})	100 Real+100 Syn (AP_{50})
343 Real Images	52.059	44.488	90.590	90.010
343 Syn Images	47.563	31.805	88.194	76.967
1000 Syn Images	53.476	35.450	91.657	83.855
50% Real + Syn	57.827	41.882	94.030	91.931
30% Real + Syn	57.841	43.248	94.563	**91.673**
20% Real + Syn	**59.998**	**44.794**	**95.644**	91.662

TABLE 2. Performance of classification model on each type.

Class	1	2	3	4	5	6	7	8	9	10	11	12	13
Precision	0.9941	0.9737	0.9862	0.9370	0.9611	0.9677	0.9615	0.9191	0.9251	0.9401	0.9715	0.9681	0.9681
Recall	0.9861	0.9927	0.9887	0.9554	0.9295	0.9841	0.9619	0.9337	0.9246	0.9446	0.9729	0.9678	0.9813
F1-score	0.9901	0.9831	0.9875	0.9461	0.9450	0.9758	0.9615	0.9264	0.9248	0.9423	0.9426	0.9644	0.9747
Class	14	15	16	17	18	19	20	21	22	X	Y	Avg	
Precision	0.9322	0.9490	0.9639	0.9693	0.9643	0.9426	0.9644	0.9425	0.9291	0.9493	0.9200	0.9567	
Recall	0.9397	0.9406	0.9639	0.9726	0.9469	0.9209	0.9571	0.9560	0.9301	0.9407	0.9327	0.9552	
F1-score	0.9360	0.9448	0.9639	0.9711	0.9556	0.9316	0.9608	0.9492	0.9296	0.9450	0.9263	0.9560	

one piece of NVIDIA GTX1080Ti GPU with a total batch size of 2.

The results are shown in Table 1. After training on 343 real images, our model has achieved 52.059 *AP* points and 90.590 AP_{50} points on 100 real test images. Alternatively, by combining 20% real data and 80% synthetic data in the training dataset, the *AP* points and AP_{50} points has been improved by 8 points and 5 points respectively. The experimental results on various test datasets show that the our synthetic data have high availability and can improve the segmentation performance. When IoU is set to 0.5, the accuracy of segmentation tested on 100 real images is 95.644%. In the case on 200 combined images, the accuracy is 91.673%. Figure 4 shows the results on real meta-phase chromosome images. We can observe that our model performs well, even when the chromosomes are overlapped and bowed.

B. RESULTS ON CLASSIFICATION

In our training dataset, 480000 pairs of the original images and the corresponding classified ordered output images of chromosomes are used. 90000 samples used for validation. 90000 samples are set as test set. All images are straightened and cropped into the size as 64x32. The preprocessed images and original images are sent to the classification network in parallel. As a multi-label classification task, we measure the performance of our classification by accuracy and F1 score.

Table 3 gives the result of each single model of us and the combined model. It also shows the accuracy of some popular CNN models and the results of predecessors. The mCNN_GO model has better performance than any single-input models. We analyze that the multi-input CNN can extract more valuable features and reduce the error of classification because the combined features have higher robustness. For comparison, we implemented the method [1]. It is then trained on our own dataset. The result shows that, after straightening, the accuracy is slightly higher than using AlexNet only. But, it highly relies on the results of preprocessing. Once the preprocessing result is not ideal, the accuracy of classification may decrease significantly. In our system, we feed both of the original images and preprocessed images into the classification network so as to minimize the uncertainty caused by bad preprocessing results. From the comparison, we can observe that our accuracy is 5% higher than others including AlexNet, VGG-16, ResNet-50 and DenseNet.

TABLE 3. Comparison of our methods with the state-of-the-art methods and predecessors.

Method	Accuracy
AlexNet [27]	0.8975
VGG-16 [28]	0.9023
ResNet-50 [29]	0.9445
DenseNet [30]	0.9414
Sharma *et al.* [1]	0.8670
Jindal *et al.* [22]	0.8460
Global-only	0.9445
Local-only	0.9220
Straighten-only	0.8200
Global & Local	0.9455
mCNN_GO (Ours)	**0.9570**

Table 2 gives the classification performance on each class. We can see that our mCNN_GO model has high robustness and good performance on most types. Our model performs best on Class 1, with the F1 score of 0.9901, and performs worst on Y chromosomes, but the F1 score is still greater than 0.92.

VI. CONCLUSION

We propose a automatic approach for karyotype analysis using deep learning and geometry optimization, and first use deep learning method to perform chromosome segmentation. We propose an simple but efficient approach to generate annotated data automatically for chromosome segmentation, which can improve the accuracy of deep models and build the foundation for deep learning applications in this field. We try to use Mask R-CNN to achieve the automation of chromosome segmentation and get a relatively good performance, especially on the overlapping and contacting chromosomes. For the highly curved chromosomes, we represent an optimization algorithm which has better generality than existing methods. Based on this optimization algorithm, we build a multi-input CNN to classify the chromosomes and the performance is better than other existing approaches. We are one step closer to the fully automated karyotype analysis.

For the future work, we may extend it to a hashing-based methods [32], [33] for this classification task, due to their high efficiency in storage and retrieval.

REFERENCES

[1] M. Sharma, O. Saha, A. Sriraman, R. Hebbalaguppe, L. Vig, and S. Karande, "Crowdsourcing for chromosome segmentation and deep classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jul. 2017, pp. 34–41.

[2] K. He, G. Gkioxari, P. Dollár, R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2961–2969.

[3] D. L. Pham, C. Xu, and J. L. Prince, "Current methods in medical image segmentation," *Annu. Rev. Biomed. Eng., Annu. Rev.*, vol. 2, no. 1, pp. 315–337, 2000.

[4] G. C. Charters and J. Graham, "Trainable grey-level models for disentangling overlapping chromosomes," *Pattern Recognit.*, vol. 32, no. 8, pp. 1335–1349, 1999.

[5] L. Ji, "Fully automatic chromosome segmentation," *Cytometry J. Int. Soc. Anal. Cytol.*, vol. 17, no. 3, pp. 196–208, 1994.

[6] P. Karvelis, A. Likas, and D. I. Fotiadis, "Identifying touching and overlapping chromosomes using the watershed transform and gradient paths," *Pattern Recognit. Lett.*, vol. 31, no. 16, pp. 2474–2488, 2010.

[7] S. Gagula-Palalic and M. Can, "Automatic segmentation of human chromosomes," *Southeast Europe J. Soft Comput.*, vol. 1, 2012.

[8] E. Grisan, E. Poletti, and A. Ruggeri, "Automatic segmentation and disentangling of chromosomes in Q-band prometaphase images," *IEEE Trans. Inf. Technol. Biomed.*, vol. 13, no. 4, pp. 575–581, Jul. 2009.

[9] S. Minaee, M. Fotouhi, and B. H. Khalaj, "A geometric approach to fully automatic chromosome segmentation," in *Proc. IEEE Signal Process. Med. Biol. Symp. (SPMB)*, Dec. 2014, pp. 1–6.

[10] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3431–3440.

[11] A. BenTaieb and G. Hamarneh, "Topology aware fully convolutional networks for histology gland segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent*. Springer, 2016, pp. 460–468.

[12] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent*. Springer, 2015, pp. 234–241.

[13] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2359–2367.

[14] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 2018, *arXiv:1804.02767*, [Online]. Available: https://arxiv.org/abs/1804.02767

[15] A. Lahiri, K. Ayush, P. Kumar Biswas, and P. Mitra, "Generative adversarial learning for reducing manual annotation in semantic segmentation on large scale miscroscopy images: Automated vessel segmentation in retinal fundus image as test case," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jul. 2017, pp. 42–48.

[16] B. Wang, Y. Yang, X. Xu, A. Hanjalic, and H. T. Shen, "Adversarial cross-modal retrieval," in *Proc. 25th ACM Int. Conf. Multimedia*, 2017, pp. 154–162.

[17] D. Dwibedi, I. Misra, and M. Hebert, "Cut, paste and learn: Surprisingly easy synthesis for instance detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 1301–1310.

[18] G. Ritter and M. T. Gallegos, "Outliers in statistical pattern recognition and an application to automatic chromosome classification," *Pattern Recognit. Lett.*, vol. 18, no. 6, pp. 525–539, 1997.

[19] B. Lerner, H. Guterman, I. Dinstein, and Y. Romem, "Medial axis transform-based features and a neural network for human chromosome classification," *Pattern Recognit.*, vol. 28, no. 11, pp. 1673–1683, 1995.

[20] C. Markou, C. Maramis, A. Delopoulos, C. Daiou, and A. Lambropoulos, "Automatic chromosome classification using support vector machines," Aristotle Univ. Thessaloniki, Greece, Tech. Rep, 2012.

[21] D. Ming and J. Tian, "Automatic pattern extraction and classification for chromosome images," *J. Infr., Millim., THz. Waves*, vol. 31, no. 7, pp. 866–877, 2010.

[22] S. Jindal, G. Gupta, M. Yadav, M. Sharma, and L. Vig, "Siamese networks for chromosome classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 72–81.

[23] Y. Wu, Y. Yue, X. Tan, W. Wang, and T. Lu, "End-to-end chromosome Karyotyping with data augmentation using GAN," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 2456–2460.

[24] W. Shen, M. Zhou, F. Yang, C. Yang, and J. Tian, "Multi-scale convolutional neural networks for lung nodule classification," in *Proc. Int. Conf. Inf. Process. Med. Imag.* Springer, 2015, pp. 588–599.

[25] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu, "Scikit-image: Image processing in Python," *PeerJ*, vol. 2, p. e453, Jun. 2014, doi: 10.7717/peerj.453.

[26] M. Javan-Roshtkhari and S. K. Setarehdan, "A new approach to automatic classification of the curved chromosomes," in *Proc. 5th Int. Symp. Image Signal Process. Anal.*, Sep. 2007, pp. 19–24.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556

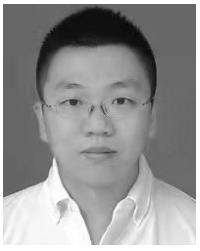
[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[30] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4700–4708.

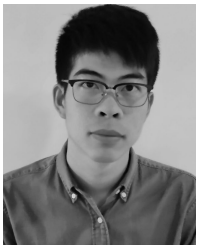
[31] W. Abdulla, *Mask R-CNN For Object Detection and Instance Segmentation on Keras and Tensorflow*. [Online]. Available: https://github.com/matterport/Mask_RCNN, 2017.

[32] Y. Luo, Y. Yang, F. Shen, Z. Huang, P. Zhou, and H. T. Shen, "Robust discrete code modeling for supervised hashing," *Pattern Recognit.*, vol. 75, pp. 128–135, Mar. 2018.

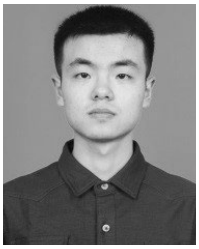
[33] M. Hu, Y. Yang, F. Shen, N. Xie, and H. T. Shen, "Hashing with angular reconstructive embeddings," *IEEE Trans. Image Process.*, vol. 27, no. 2, pp. 545–555, Feb. 2018.



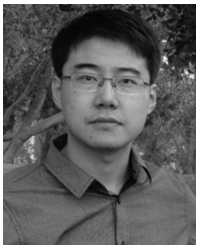
NING XIE received the M.E. and Ph.D. degrees from the Department of Computer Science, Tokyo Institute of Technology, Tokyo, Japan, in 2009 and 2012, respectively. In 2012, he was appointed as a Research Associate with the Tokyo Institute of Technology. From 2014 to 2016, he was an Assistant Professor with the School of Software Engineering, Tongji University. Since 2017, he has been an Associate Professor with the School of Computer Science and Engineering, UESTC. His research interests include computer graphics, game engine, and the theory and application of artificial intelligence and machine learning.



XU LI is currently pursuing the bachelor's degree in electrical and electronic engineering with the University of Electronic Science and Technology of China. His research interests include the image processing and deep learning.



KANG LI received the B.E. degree in computer science and technology engineering from the China University of Petroleum, Qingdao, China, in 2017. He is currently pursuing the M.Eng. degree with the University of Electronic Science and Technology. His current research interests include computer vision and deep learning.



YANG YANG received the bachelor's degree from Jilin University, in 2006, the master's degree from Peking University, in 2009, and the Ph.D. degree (supervisors: Prof. Heng Tao Shen and Prof. Xiaofang Zhou) from the University of Queensland (UQ), in 2012. He is currently with the University of Electronic Science and Technology of China. He joined the National University of Singapore (NUS) as a Research Fellow, working with Prof. T.-S. Chua, from 2012 to 2014. His research interests mainly focus on multimedia search, social media analysis, and machine learning.

HENG TAO SHEN received the B.Sc. (Hons.) and Ph.D. degrees in computer science from the Department of Computer Science, National University of Singapore, in 2000 and 2004, respectively. He is currently a Professor, the Dean of the School of Computer Science and Engineering, the Executive Dean of the AI Research Institute, and the Director of the Center for Future Media, University of Electronic Science and Technology of China. His current research interests include multimedia search, computer vision, artificial intelligence, and big data management. He has published over 250 peer-reviewed articles and received seven Best Paper Awards from international conferences, including the Best Paper Award from the ACM Multimedia 2017 and the Best Paper Award-Honourable Mention from ACM SIGIR 2017. He has served as the General Co-Chair for ACM Multimedia 2021 and the TPC Co-Chair for ACM Multimedia 2015, and is an Associate Editor of the *ACM Transactions of Data Science* (TDS), the IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP), the IEEE TRANSACTIONS ON MULTIMEDIA (TMM), and the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING (TKDE).