Fully Automatic Karyotyping via Deep Convolutional Neural Networks

Chengyu Wang[1], Limin Yu[2], *Member, IEEE,* Jionglong Su[3], Juming Shen[4], Valerio Selis[5], Chunxiao Yang[6], and Fei Ma[7]

***Abstract*—Chromosome karyotyping is an important yet labor-intensive procedure for diagnosing genetic diseases. Automating such a procedure drastically reduces the manual work of cytologists and increases congenital disease diagnosing precision. Researchers have contributed to chromosome segmentation and classification for decades. However, very few studies integrate the two tasks as a unified, fully automatic procedure or achieved a promising performance. This paper addresses the gap by presenting: 1) A novel chromosome segmentation module named ChrRender, with the idea of rendering the chromosome instances by combining rich global features from the backbone and coarse mask prediction from Mask R-CNN; 2) A devised chromosome classification module named ChrNet4 that pays more attention to channel-wise dependencies from aggregated informative features and calibrating the channel interdependence; 3) An integrated Render-Attention-Architecture to accomplish fully automatic karyotyping with segmentation and classification modules; 4) A strategy for eliminating differences between training data and segmentation output data to be classified. These proposed methods are implemented in three ways on the public Q-band BioImLab dataset and a G-band private dataset. The results indicate promising performance: 1) on the joint karyotyping task, which predicts a karyotype image by first segmenting an original microscopical image, then classifying each segmentation output with a precision of 89.75% and 94.22% on the BioImLab and private dataset, respectively; 2) On the separate task with two datasets, ChrRender obtained AP50 of 96.652% and 96.809% for segmentation, ChrNet4 achieved 95.24% and 94.07% for classification, respectively. The COCO format annotation files of BioImLab used in this paper are available at *https://github.com/Alex17swim/BioImLab* The study introduces an integrated workflow to predict a karyotyping image from a Microscopical Chromosome Image. With state-of-the-art performance on a public dataset, the proposed Render-Attention-Architecture has accomplished fully automatic chromosome karyotyping.**

***Index Terms*—Chromosome karyotyping, fully automatic, classification, instance segmentation.**

I. Introduction

There are 46 chromosomes from a normal human cell, which can be divided into 22 pairs of autosomes and the 23rd pair of sex chromosomes [1]. Chromosomes can only be visible and distinguished in the metaphase of the cell cycle. There are three standard techniques for staining the metaphase chromosome to make them visible under a microscope: (1) mFISH: multicolor fluorescence in situ hybridization with five color dyes [2], [3], (2) Q-Band: staining with the fluorescent dye nitrogen mustard quinacrine [4], and (3) G-band: staining with the dye of Giemsa [1]. The microscopical chromosome image (MCI, as Fig. 4 (a)) is the metaphase cell image taken through the microscope. It separates the chromosomes from it and rearranges the chromosomes by their type. A Karyotyping Image (KI, as Fig. 4 (f)) is formed by lining up the chromosomes according to their type. For decades, cytologists have used karyotyping images to diagnose genetic diseases via checking numerical or structural abnormality [5], [6], such as prenatal screening for Down's syndrome (21st chromosome), Sex chromosome abnormality (23rd or 24th), Edward'syndrome (18th), Patau'syndrome (13th), *cri du chat* syndrome (5th), or cancer caused by neoplasia-associated chromosomal rearrangements [7]. The karyotyping procedure plays an important role in processing the images and diagnosing the diseases, which is not only tedious and labor-intensive but also error-prone due to the fatigue of operators [8]. Many studies have contributed to the development of computer-based automatic processes. The computer-aided karyotyping process often involves segmentation and classification [1] [8]. Chromosome segmentation on MCI is the first step in karyotyping. By classifying each pixel from a potential individual chromosome, a normal MCI can be segmented into 46 Individual Chromosome Images (ICIs, as Fig. 4 (c)). The major challenge in this task is to distinguish the touching or overlapping chromosomes from clusters [9]. A typical procedure for conventional methods consists of the following steps [10]: (1) pre-process, (2) binary and morphological, (3) contour to identify the boundary, (4) curvature graph with the cut point, (5) possible separation line over the overlapped region, (6) cut-points identification and drawing, (7) final separation on the overlapped region.

[1]HeXie Management Research Centre, College of Industry-Entrepreneurs, Xi'an Jiaotong-Liverpool University, Suzhou, Jiangsu 215000, China

[2]Department of Electrical and Electronic Engineering, Xi'an Jiaotong-Liverpool University, Suzhou, Jiangsu 215000, China

[3]School of AI and Advanced Computing, XJTLU Entrepreneur College (Taicang), Xi'an Jiaotong-Liverpool University, Suzhou, Jiangsu 215000, China

[4]4Institute of Leadership and Education Advanced Development, Academy of Future Education, Xi'an Jiaotong-Liverpool University, Suzhou, Jiangsu 215000, China

[5]Department of Electrical Engineering and Electronics, University of Liverpool, L69 3GJ Liverpool, U.K.

[6]Suzhou Precision Medical Technology Company Ltd., Suzhou, Jiangsu 215100, China

[7]School of Mathematics and Physics, Xi'an Jiaotong-Liverpool University, Suzhou, Jiangsu 215000, China

This work was supported in part by the National Natural Science Foundation of China (Grant No. 61501380), the Qinglan Project of Jiangsu Province of China, Laboratory of Computational Physics (Grant No.6142A05180501), Xi'an Jiaotong-Liverpool University (XJTLU) Research Development Fund (RDF-17-02-51), XJTLU Research Enhancement Fund (REF-19-01-04 and REF-18-01-04), Key Programme Special Fund (KSF) in XJTLU (KSF-E-32 and KSF-E-21), and XJTLU Construction of a Bioinformatics Platform for Precision Medicine (RDS10120180041).

Corresponding author: Fei Ma (e-mail: Fei.Ma@xjtlu.edu.cn).

Traditional procedures for computer-aided chromosome classification include pre-processing, feature extraction, and classification [1]. Poletti *et al.* researched a public Q-band chromosome dataset (BioImLab [11]) and obtained an average classification accuracy of 94% on 5474 chromosomes [4] in 2012. Their method first estimated the medial chromosome axis, extracted image features, and appraised the orientation by rescaling and normalizing. Finally, they devised a neural network to predict the label. Wang *et al.* proposed a patch-based algorithm for classifying chromosome images from the public dataset Multiplex-fluorescence in situ hybridizations (M-FISH) [12] in 2017. Their method was based on high-order singular value decomposition. In the experiments, they achieved a higher correct classification ratio than other methods such as sparse representation classification and improved adaptive fuzzy c-means clustering (IAFCM). Wang *et al.* proposed an artificial neural networks architecture for chromosome classification [13] in 2009. The blocks in the first stage classified chromosomes into seven classes. The second stage blocks assign each chromosome a label from the seven classes. They improved the accuracy from 67.5% to 97.5% with the second layer of seven classifiers. Qin *et al.* proposed a method named Varifocal-Net to classify the type and polarity of individual chromosomes [14] in 2018. Their approach consisted of two networks for global and local feature extracting. Experiments on a private dataset of 1909 karyotype images reported 99.2% accuracy with metric of per patient case. Lin *et al.* studied both classification and segmentation of chromosome images with a deep learning strategy [15], [16]. They achieved an accuracy of 95.98% in 24 types classification task [15], and 94.09% accuracy in cluster type identification task [16] which classified a chromosome cluster into four classes, i.e., the chromosome of 'single', 'touching', 'overlapping', 'touching and overlapping'. The tasks of 24 chromosome types classification experimented on a dataset consisting of 2990 individual chromosomes from 65 normal karyotype images. Another task of identifying five chromosome cluster types was carried out on a dataset with 6592 samples from 500 staid microphotograph images. They received 8.5% and 1.92% accuracy improvements compared to the baseline, respectively.

Deep learning (DL) technology has achieved great success in computer vision since 2012. One of the most important DL techniques is the deep convolutional network [17]. DL improves the performance of chromosome karyotyping significantly [8], [18], [19]. Pardo *et al.* used spectral together with spatial information. They proposed an 'end to end semantic segmentation' approach in 2018 [18], which used the foundations of the classical CNN architecture of VGG [20] and SPP [21]. They measured their results in terms of correct classification ratio, i.e., the ratio of all the correctly classified chromosomes pixels divided by the toto chromosome pixels. Results on the whole 84 Vysis images were 83.91%, and 87.41% on the set with 18 low-quality images removed.

Redmon *et al.* proposed a one-stage architecture, i.e., YOLO, to obtain the category probability and location coordinates, eliminating the need for region proposals [22]. YOLO has been used in breast cancer detection [23], nucleus detection for tumor [24], and pedestrian detection [25], [26].

Bai *et al.* devised a three-step architecture in 2020 by employing UNET and YOLOv3 network [27] . Their dataset consisted of 130 G-band chromosome images augmented to 1300 images, including 27600 individual chromosomes. The percentage of correctly extracted chromosomes achieved 99.3%, i.e., 2283 chromosomes were extracted from 2300.

Several researchers have contributed to the fully automatic karyotyping with a combination of chromosome segmentation and classification in a joint task. Ning *et al.* developed an integrated workflow in 2020 [19], using the Mask R-CNN [28] to segment the input MCI and a devised CNN module to classify the sub-images. Due to the lack of annotation data, they used a combination dataset consisting of real images and synthesized images for the segmentation task. A best AP50 score of 95.644% was obtained in the study. The proposed multi-input CNN with Geometric Optimization network accepts three types of images, including original, straightened, and cropped. After being trained on 480000 samples, they reported 95.67% of precision, 95.52% of recall, and 95.60% F1-score. Mano *et al.* studied a method to detect numerical abnormalities in 2020 [8]. Their main idea is (1) to use YOLOv2 [29] followed by post-processing to separate individual chromosomes, and (2) to update the VGG19 [20] for chromosome classification. They separated all 1350 chromosomes from 29 MCIs in the test set with perfect results (e.g., 100% accuracy) on a private dataset from the Center of Excellence in Genomic Medicine Research, containing 147 *non-overlapped* metaphase images. Their method obtained a classification accuracy of 95.04%, recall of 94.84%, the precision of 94.90% on the private dataset, and 94.11% of classification accuracy, 93.86% of recall, 94.51% of precision on the public dataset BioImLab. Xiao *et al.* enumerated chromosomes from a dataset of 1375 MCIs in 2020 [30]. By adding a Hard Negative Anchors Sampling to the regional proposal network, their model 'DeepACEv2' may extract partial chromosomes. The segmentation task achieved a Whole Correct Ratio (the percentage of images in the testing set with NO error instances) of 71.39% and an mAP of 99.60%. At the same time, the reference baseline of Faster RCNN-101 also obtained an mAP of 99.39%. Nevertheless, both [19], [8], and [30] did not report the accuracy of fully karyotyping, i.e., the classification results on the segmentation output from the same public dataset (e.g. BioImLab).

Somasundaram *et al.* introduced a method that outperforms previous algorithms' classification accuracy, achieving 98.7% accuracy, an area under the curve of 0.97, and an abnormality detection accuracy of 98.4% [31]. Madian *et al.* have a particular focus on chromosomes that are straight, bent, or severely bent [32]. Their proposed method is particularly effective in pinpointing the centromeres of chromosomes regardless of their degree of bending. Devaraj *et al.* proposed algorithm including curvature analysis, overlap and touch region detection, axis tracking, and hypothesis split for chromosome segmentation [33]. The overall accuracy for separating touching chromosomes is 98.06%, while for overlapped chromosomes is 96.79%.

In the field of computer-assisted chromosome karyotyping, most researchers focus on a single task of either segmentation

[16] or classification [1]. However, the practical utilization of classification is on the segmentation results, which differ significantly from the classification dataset obtained by segmenting individual chromosomes from a karyotyping image. Few researchers report the performance of implementing a classification model of the segmentation outputs from a publicly available dataset [14] [30], making their work difficult to evaluate.

This study bridges the gap by combining the segmentation and classification tasks into an integrated architecture named Render-Attention-Architecture, a fully karyotyping Architecture that employs the Render strategy for segmentation tasks and Attention structure for the classification task. Experiments are implemented on the public Q-band BioImLab and G-band private datasets. The contributions of the present work are threefold. First, it obtain the crisp edges of segmented instances by using the idea of constructing 2D images from 3D objects. Second, it proposes an efficient architecture to produce KI from MCI with an integrated chromosome segmentation and classification workflow. It demonstrates the effectiveness of the rotating & cropping process in the fully karyotyping task. Finally, it devises a network for the chromosome classification task, built by stacking the proposed ChrBlock, which contains an attention structure.

The rest of this paper is organized as follows: Section II explains the proposed methods and the experiment design. The results listed in Section III demonstrate the efficiancy of the architecture. Our work is concluded in Section IV.

II. Materials and methods

A. Chromosome datasets

Two datasets are used in this study, containing both segmentation and classification data, and are empirically split into three sets: training (85%), validating (5%), and testing (10%). The first is the public Q-band BioImLab dataset. It has been used in many studies since 2008 [8], [9], [11]. Segmentation data of BioImLab contains 117 cells, and each cell has one KI and at least one MCI. Hence there are 117 KIs and 162 MCIs. Of all KIs, 83 correspond to a single MCI, 24 correspond to two, nine correspond to three, and one KI corresponds to four. After scrutinizing all MCIs of BioImLab, six images were found containing less than 15 ICIs each. They were retained in the training set since they are from cells with more than one MCI.

A normal cell corresponds to an MCI and 46 ICIs (it can be more or less than 46 ICIs for an abnormal cell.). The MCI or ICIs from the same cell is named 'one case,' identified with a unique case name. In this view, 5474 ICIs were separated from 119 cases of the BioImLab classification data. Two cases have neither corresponding KI nor MCI. The testing and evaluating set for segmentation contain 16 (10%, case NO. 5, 10, 18, 28, 34, 41, 44, 57, 62, 65, 66, 68, 78, 84, 89, and 96) and eight (5%) MCIs which randomly picked from 83 single MCIs. The rest 138 (85%) cases were used as the training set. The second G-band dataset is provided by a company in Suzhou, China. It consists of 841 MCIs and 38686 ICIs. The criterion of the sub-set setting is the same as BioImLab. The images in the classification training set are flipped to reduce the influence of the 'top and bottom' problem but not flipped in evaluating and testing set. Note that *the cases from the testing set for both classification and segmentation tasks are the same, which guarantees the testing data are never fed into the network before testing for both segmentation and classification.*

The background of the raw images was adjusted to contrast by the scanning device (Leica CytoVision) to make the background whiter and the foreground darker. In the segmentation training procedure, the images were first enhanced by resizing to 256 pixels in height and width, then random flipping by 0.5. In the classification training procedure, the images were cropped with redundant edges and rotated to vertical, as detailed explained in II-B2.

B. Methods

The overall schematic representation of the proposed Render-Attention-Architecture is given in Fig. 1, which is fed with an MCI and outputs a processed KI. It is based on harnessing a segmentation module with two additional modules, i.e., a classification module and a rotating & cropping module.

1) Segmentation Model: Segmentation architecture with DL can be divided into a 'one-stage strategy', which predicts bounding boxes and calculates class probabilities simultaneously, and a 'two-stage strategy', which includes additional 'box proposal' stages to predict bounding boxes. 'Two-stage' strategy generally performs better on the localization than 'one-stage', although often slower [28], [34].

With DL methods, the predicted object boundaries are not always fine enough, with each pixel being assigned with only a unique label [35]. The issue is amplified in the chromosome segmentation task due to a few facts, e.g., chromosomes may be touching or overlapping with each other; pixels in an overlapping region belong to multiple instances. In chromosome segmentation, boundary details contribute essential information to distinguish different instances.

Consider rendering a 3D model into a 2D image. The continuous input of physical occupancy can be represented by physical and geometric reasoning from a physical entity. In contrast, the rendered 2D output can be a regular grid, which is non-uniform and discrete. Recently, researchers have tackled the issue of non-uniform representations with the idea of octrees [36], mesh [37], and neural network [38].

Analogously, the input of underlying continuous MCI entity can be encoded by feature maps from a two-stage segmentation network (e.g., mask R-CNN [28]), and the output of ICIs instance grid can be rendered by visiting the real-value of feature maps via interpolation (e.g., bilinear interpolation). By borrowing the structure of PointRend [35], the segmentation module 'ChrRender' was designed in this paper, using hyper-parameters and pre-trained parameters from Detectron 2 [39]. The architecture of ChrRender is given in the left part of Fig. 1, with procedures as follows:

1, predict the chromosome instance label, bounding box and *coarse masks* by mask R-CNN [28].

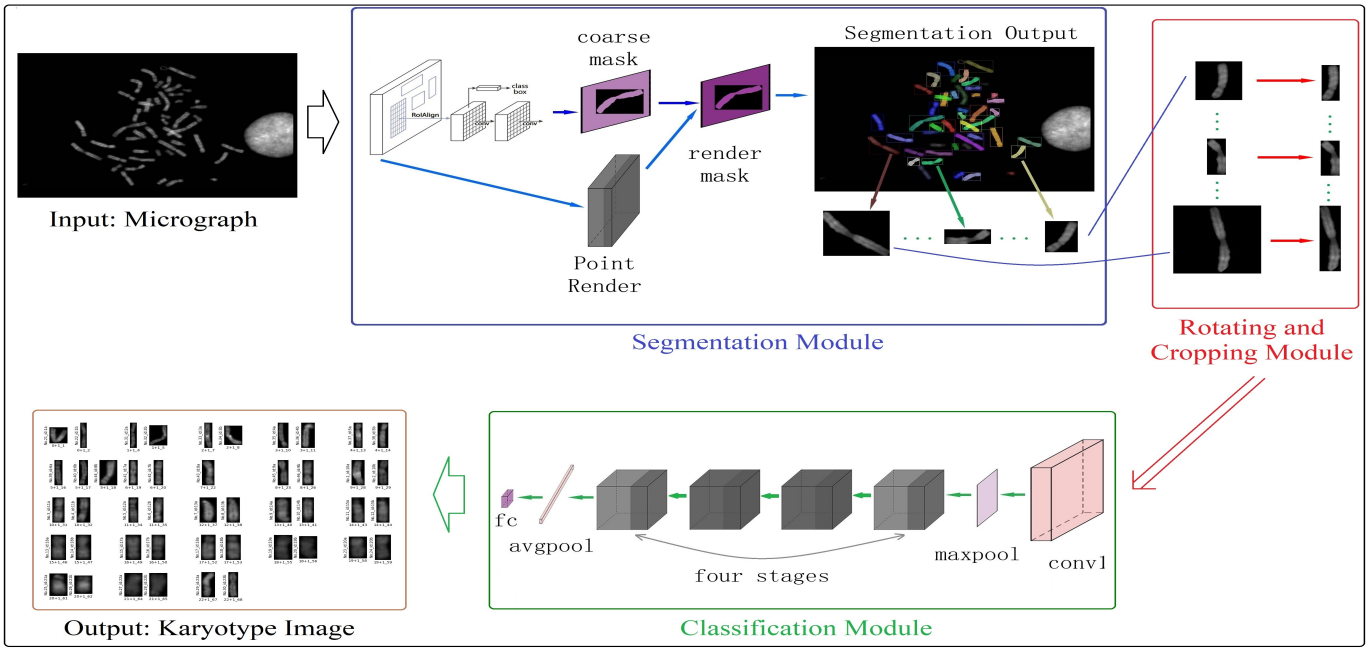


Fig. 1: The overall structure of Render-Attention-Architecture: (1) the left box with a light blue border shows the segmentation module. It employs Point Render to output *render mask* instead of *coarse mask* in Mask R-CNN [19], [28]. The segment predictions are separated for classification. (2) the right box with a green border demonstrates the classification network consisting of three phases. The basic ChrBlock Unit stacks four stages in the middle phases. (3) a 'rotating & cropping' module in the middlebox with a red border is implemented to connect the segmentation module and the classification module to make the segmentation task's output data coincident with the classification task's input.

2, with the strategy of adaptive subdivision [28], iteratively render each *coarse mask* as given in Fig. 2. The *point-wise feature* combines (concatenates) the *coarse mask* with the extracted *fine feature vector*, as defined in:

$$P_k = \sum_{i,j}^{W_k,H_k} C_{(i,j)} + F_{(i+u,j+v)} \tag{1}$$

where P_k is the *point-wise feature* of the kth instance; (i,j) is the point location of the kth instance whose size is $W_k \times H_k$; $C_{(i,j)}$ is the *coarse mask* value of point (i,j); (u,v) is the relative location from the point (i,j), $u \in [0,1)$ and $v \in [0,1)$, note 1 is the maximum distance between two points location; $(i+u, j+v)$ is the absolute location of interpolated point; $F_{(i+u,j+v)}$ is the *fine feature* value which calculated by bilinear interpolated from four neighbor points $(x_i, y_j), (x_i, y_{j+1}), (x_{i+1}, y_j), (x_{i+1}, y_{j+1}))$, which is defined as:

$$\begin{aligned} F_{(i+u,j+v)} = \\ (1-u)(1-v)f(i,j) + (1-u)vf(i,j+1) \\ + u(1-v)f(i+1,j) + uvf(i+1,j+1) \end{aligned} \tag{2}$$

where $f(i,j)$ is the pixel value of point (i,j).

3, replace the N points in a coarse mask with *re-classified N points*, and output the *render mask* as the final segmentation output, boundaries of which are crisper.

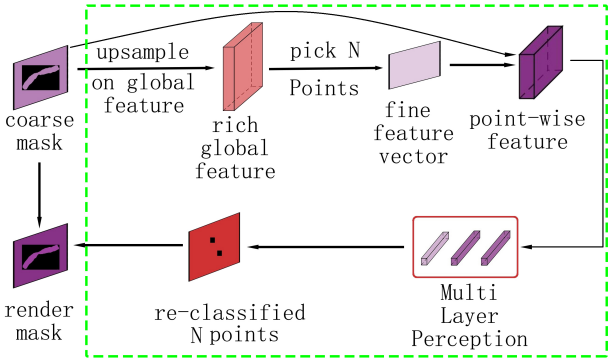


Fig. 2: The procedure of rendering coarse masks to.

2) Rotating & Cropping module: In the task of chromosome classification, the input ICI data are different to KIs in which the chromosomes have been rotated vertically. Hence the segmented ICIs need to be rotated for proper rotating before being input into the classification procedure. In addition, during the rotating of ICIs, there will be redundant edge filling to the image, which provides little valuable information for the classification task.

The differences between ICIs from classification datasets and segmentation output are likely to lead to poor performance of integrated fully automatic karyotyping. Results in left of Table I indicate that to get the best performance, rotating & cropping should be *implemented in both training and inference procedures.*

In study [19], Ning *et al.* feed the classification module

TABLE I: The left table is the results of three combinations with rotating & cropping strategy. With model ChrNet4, strategy C outperforms A on the Q-band BioImLab dataset with 27.44%, i.e., from 67.8% to 95.24%. The values in bold give the best results, while red is from the basic strategy A, i.e., neither rotating nor cropping in both training and testing procedures.
The right table is the predicted bounding box results of three models on two datasets (ds): Q-band BioImLab (Q) and G-band private (G). FasterR, MaskR, and ChrR is short for Faster-RCNN101, Mask-RCNN50, and ChrRender, respectively. The values in bold illustrate that ChrRender outperforms Faster-RCNN101.

strategy	dataset	train strategy	testing strategy	class accuracy (%)
A	Q-band BioImLab	none	none	67.80
B		none	rotate+crop	65.36
C		rotate+crop	rotate+crop	**95.24**
A	G-band private	none	none	93.43
B		none	rotate+crop	88.62
C		rotate+crop	rotate+crop	**94.07**

module	ds	mAP	AP50	AP75	APs	APm
FasterR	Q	73.752	96.627	87.754	75.780	73.525
MaskR		75.279	95.518	91.375	77.281	73.123
ChrR		**75.872**	**96.652**	**90.507**	77.086	75.200
FasterR	G	81.039	95.769	91.209	77.488	82.007
MaskR		81.810	96.789	92.523	77.915	82.744
ChrR		**82.232**	**96.809**	**92.530**	78.306	83.058

with triple ICIs (original, straightened, cropped), reducing the non-coincident influence between the training and testing data. We take a step further by making the classification training data and the segmentation test data consistent in chromosome orientation (vertical) and margin distance.

The 'rotating & cropping' component consists of the following two steps:

1, position an image with Canny and Hough Lines methods. The algorithm is described in algorithm 1:

Algorithm 1 Image rotating

Input: img0: an original single chromosome image array [h, w, c]
Output: img2: a positioned image array [h, w, c]

extract the chromosome $edge$ with Canny method
extract $lines$ with Hough line transformation method
for ρ, θ *in* $lines[0]$ **do**
    $a = cosine(\theta), b = sine(\theta)$
    $x0 = a * \rho, y0 = b * \rho$
    calculate two vertexes $(x1, y1), (x2, y2)$ via $x0, y0, a, b$
calculate tanh value $tanh$ of $(x1, y1), (x2, y2)$
calculate degree $angle$ from $tanh$
rotate image with degree $angle$ and save it as img2
return img2

2, crop the positioned ICI by removing redundant edge areas.

3) Classification Model: He *et al.* studied the Residual Block in their proposed ResNet in 2016 [40]. They solved the training degradation problem and made ResNet one of the baseline modules in many CNN classification tasks. Xie *et al.* improved ResNet with exposed 'cardinality' dimension and proposed ResNeXt module in 2017 [41]. By setting a cardinality (group) value of 32 and base width value of 4, ResNeXt has a similar number of parameters as ResNet50, but with higher accuracy. Jie *et al.* proposed SENet by stacking the Squeeze-and-Excitation unit in 2018 [42], which won first place in the ILSVRC in that year.

In this study, inspired by ResNet, ResNeXt, and SeNet, a classification network named ChrNet4, which is a CNN Network for Chromosome classification with base width four in ChrBlocks, was proposed. The overall architecture of ChrNet4 is given in the right part of Fig. 1 which is bordered dark green. ChrNet4 is built on the ChrBlock, which consists of:

- three successive convolutional layers with kernel sizes one by one, three by three, and one by one, respectively.
- a batch normalization layer following each convolutional layer and activated by a Relu layer.
- an attention layer is given in Fig. 3 (a), consisting of an average pooling layer, two linear layers followed by a leak Relu and a sigmoid activation, respectively.

Fig. 3 (b) describes the concise architecture of ChrNet4, which can be divided into three phases:

- in the 1st phase, the input image size was shrunk to 112×112 by a convolutional layer, then 56×56 by a max-pooling layer.
- in the 2nd phase, the ChrBlock has four stages. Each stage was constructed by accumulating the ChrBlock twice, three times, and twice, respectively. The input channels of the four stages are 64, 128, 256, and 512 in sequence. The cardinality dimension G and base width W are set to 32 and 4, respectively [41]. The equivalent expression of the *ChrBlock input channel* C is computed by:

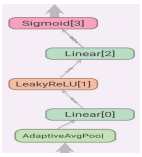
$$C = c \times \frac{W}{C_0} \times G$$

  where channel $c \in [64, 128, 256, 512]$; first channel C_0 is 64, which is the output channel of stage 1.
- in the 3rd phase, an average pooling layer is followed by a fully connected layer, which outputs the final 24 dimension prediction of 24 chromosome classes.

4) Loss: The joint loss in (3) was implemented for the classification module. The left term is cross-entropy loss, and the right term is center loss, which was studied in previous papers [43], [44]. Center loss can improve the classification performance by condensing the inner-class feature, as well as enlarging the inter-class distance.

$$L = -\sum_{i=1}^{C} y_i \log S_i + \alpha \sum_{j=1}^{B} \left\| x_j - c_{y_j} \right\|_2^2 \qquad (3)$$

where C and B stand for the number of classes and batch size, respectively. $y_i \in \{0, 1\}$, where 1 stands for a positive sample from class i, and 0 for negative. S_i is the ith softmax layer output. x_j is the jth sample feature vector, and c_{y_j} is the y_jth class feature center. $\alpha \in (0, 1)$ is used to control the



(a)

Output size	112x112	56x56	56x56	28x28	14x14	7x7	1x1
ChrNet4	conv k=7x7, s=2	maxpool k=3x3, s=2	{conv, 1x1, 128; conv, 3x3, 128; conv, 1x1, 256} x2	{conv, 1x1, 256; conv, 3x3, 256; conv, 1x1, 512} x3	{conv, 1x1, 512; conv, 3x3, 512; conv, 1x1, 1024} x3	{conv, 1x1, 1024; conv, 3x3, 1024; conv, 1x1, 2048} x2	average pool, 24-d fc

(b)

Fig. 3: Structure of 'chrblock': (a) demonstrates the 'attention layer'. (b) shows the chrblock structure: the left two columns compose the first phase; the middle four columns are stage 1, stage 2, stage 3 and stage 4 of the second phase; the right most column is the 3rd phase.

weight of seocond term (center loss) in (3). The value of α is set as the same value used in [43], which is 0.5.

C. Metric

The following metrics are commonly used in object detection studies [45] which are also used in this study.

- True positive (TP) is the number of correct predictions, false positive (FP) is the number of wrong predictions, true negative (TN) is the number of true negative predictions, and false negative (FN) is the number of false negative predictions. Precision $P = TP/(TP + FP)$, recall $R = TP/(TP + FN)$.
- IoU (Intersection over Union) defined in (4) is a measurement based on the Jaccard Index, which is a coefficient of similarity for two sets of data. IoU counts the overlapping area between the predicted bounding box B_p and the ground-truth bounding box B_{gt} and is divided by the area of union between them. Given a threshold thr (0.30, etc.), a predicted object area and ground truth area with $IoU \geqslant thr$ are considered as TP, otherwise as FP.

$$J(B_p, B_{gt}) = IoU = \frac{area(B_p \cap B_{gt})}{area(B_p \cup B_{gt})} = \frac{overlap}{union} \quad (4)$$

- Given N predictions, in which each TP prediction corresponding to a unique Recall value R and Precision $P(R)$, Average Precision (AP) is defined as:

$$AP = AP_{all} = \sum_{n} (R_{n+1} - R_n) P_{interp}(R_{n+1})$$

where $n \in N$, and $P_{interp}(R_{n+1}) = \max_{\tilde{R}:\tilde{R} \geqslant R_{n+1}} P(\tilde{R})$.

For total C classes and $i \in C$,

$$mAP = \frac{1}{C} \sum_{i=1}^{C} AP_i \quad (5)$$

Besides mAP, AP50, AP75, APs and APm [45] are also used to evaluate the segmentation performance. The results of joint karyotyping in this study are assessed by a Fully Karyotyping Accuracy (FKA), which is the ratio of correctly predicted ICI number divided by the total number of ground truth (always be 46; the Fully Karyotyping Precision and Recall are also divided by 46), and an Only Classifying Accuracy (OCA) is the ratio divided by segmentation output number (may not be 46).

III. Results

Three segmentation methods and four classification modules are implemented to compare the performance of the proposed methods on two datasets. Following this, the classification results of segmentation output are reported, with metrics of both Fully Karyotyping Accuracy and Only Classifying Arrucacy defined in section II-C.

A. Segmentation results: From MCI to Segmented Bounding Box

To train and evaluate the segmentation module, first, pretrained parameters were loaded from ImageNet [38] of Detectron 2 [31]. Then, the modules were fine-trained with 50000 iterations by feeding with chromosome segmentation data. Finally, the parameters with the best performance on the evaluating set were used for the joint fully automatic karyotyping.

The right of Table I gives the segmentation performance of three segmentation modules: Faster RCNN-101, Mask RCNN, and ChrRender. Comparing the previous studies with the same metric, ChrRender performs better:
(1) Xiao *et al.* reported *mAP* with an increase of *0.21%* percent (99.39% to 99.60%) as compared to the baseline Faster RCNN-101 over a dataset which contains 1375 G-band images [30]. With the same baseline Faster RCNN-101 and metric *mAP*, *ChrRender achieve 2.12% increasement (73.752 to 75.872) on BioImLab and 0.842% (81.039 to 82.232) on private dataset, which both surpass 0.21% reported in [30].* Besides, ChrRender also achieves 0.025% (96.627 to 96.652) and 2.753% (87.754 to 90.507) increase on BioImLab, as well as 1.04% (95.769 to 96.809) and *1.321%* (91.209 to 92.530) on private dataset, with metric of AP50 and AP75 respectively. Compared to the *71.39%* of 'Whole Correct Ratio' (WRC) obtained in [30], WRC is 54.1% on the private dataset in this research, 'Missing Single Rate' is 21.2% and 'Missing One at the most Rate' is *75.3%* (54.1 % (WRC, missing zero) + 21.2% (missing one)). We found the baseline Faster RCNN-101 obtained mAP of *99.39%* in [30]. In our study, however, its mAP is *81.039%* on the private dataset in this research, which indicates our private dataset is more challenging. In the BioImLab test set, the same baseline Faster RCNN-101 got mAP 73.752%, and the best result of correctly segmented ICIs is 43, i.e., WRC is 0.

(2) Ning *et al.* reported AP50 95.644% in the segmentation task over 443 images, in which 100 images were used for

testing [19]. This work report better performance, which is *96.652%* on Q-band BioImLab and *96.809%* on G-band private dataset.

(3) Although Mano *et al.* reported the perfect result (AP=0.9923) on the segmentation task in their report [8], there are *rarely overlapped chromosomes in MCI.* They report only classification results on the more challenging dataset BioImLab but not with the segmentation results.

(4) Wang *et al.* [46] reported segmentation results of AP50 95.9% on the same BioImLab dataset, which sligthly beghined 96.652% in this work.

B. Classification Results: From ICI to KI

The different classification modules and their performance are listed in Table II. On *BioImLab*, the proposed ChrNet4 achieved 95.24% classification accuracy, which surpassed previous studies (94% in [4], 94.11% in [8]) on the same dataset. As one of the key values of this work is to combine segmentation with classification, the performance of the classification task is expected to be evaluated on segmentation results from which metrics have to be manually calculated due to lack of ground truth label. To release the heavy work of doing that, the models are still evaluated on the classification dataset but processed by rotating & cropping strategy. The strategy is also used on segmentation outputs and subsequent aligning of the classification input with segmentation outputs. The metrics can be automatically calculated.

Considering 'with pre-trained parameters', 'use Center Loss', and 'employ rotating & cropping component', five strategies of training model refer to: (a) without any; (b) only with 'pre-trained'; (c) with 'pre-trained' and 'rotating & cropping'; (d) with all; (e) with 'Center Loss' and 'rotating & cropping'. It is found in Table II that:

* from strategy (a) to (d), results stepped improved for all baseline models, except (c) to (d) for ResNet50. (a) reports the worst results below 40%, and (d) demonstrates more than 91% on Accuracy, Recall, and Precision.
* with the pre-trained parameters, i.e., from (a) to (b), the performance dramatically enhanced for all baseline models by doubling.
* the 'rotating & cropping' process, i.e., from (b) to (c), contributes more than 20% classification accuracy for ResNet50 and SeNet50, and 13% for ResNeXt50.
* when Center Loss is added, i.e., from (c) to (d), the performance of ResNeXt50 and SENet50 will increase 1%, but ResNet50 declines 2%.
* without pre-trained parameters, but with Center Loss and 'rotating & cropping', i.e., from (d) to (e), all baseline models' performance increase slightly on the BioImLab except RexNext50 which declines 1%, and increase about 3% on the private dataset. This indicates that if adequate data is available, a model could perform better without transfer learning, i.e., without pre-trained parameters from a universal dataset such as ImageNet.
* with strategy (e), the proposed ChrNet4 performs better than all baseline modules *on both datasets*, as demonstrated in the last two lines at the bottom.

TABLE II: The results of different classification methods on two datasets: Acc is accuracy, Rec is recall, and P is precision. The *italic float numbers* indicate they are from the G-band private dataset. The **float number in bold** are the best results in the group. Parameters size (MB) are *62.39* for ChrNet4, 89.96 for ResNet50, 87.85 for ResNeXt50, and 99.46 for SENet50.

model	dataset+ strategy	Acc (%)	Rec (%)	P(%)
ResNet50	BioImLab + (a)	32.34	32.63	26.55
	BioImLab + (b)	72.42	72.63	73.93
	BioImLab + **(c)**	**93.75**	**93.83**	**95.30**
	BioImLab + (d)	91.30	91.42	92.92
	BioImLab + (e)	91.71	91.55	93.22
	Private + (d)	*91.43*	*91.32*	*92.29*
	Private + (e)	*93.25*	*93.02*	*94.06*
ResNeXt50	BioImLab + (a)	35.05	35.63	28.98
	BioImLab + (b)	79.21	79.04	78.73
	BioImLab + (c)	92.26	92.38	93.86
	BioImLab + (d)	**93.34**	**93.44**	**94.35**
	BioImLab + (e)	92.26	92.22	94.08
	Private + (d)	*90.84*	*90.84*	*91.95*
	Private + (e)	*93.55*	*93.24*	*93.97*
SENet50	BioImLab + (a)	13.86	13.67	07.56
	BioImLab + (b)	71.06	71.32	70.91
	BioImLab + (c)	91.58	91.71	93.98
	BioImLab + (d)	**93.21**	93.08	95.27
	BioImLab + **(e)**	93.07	**93.17**	**95.46**
	Private + (d)	*90.28*	*90.04*	*91.12*
	Private + (e)	*93.68*	*93.33*	*93.82*
ChrNet4	BioImLab+ (e)	**95.24**	**95.32**	**97.01**
	Private + (e)	***94.07***	***93.76***	***94.18***

C. Fully Karyotyping Results: From MCI to KI

The procedure of Render-Attention-Architecture is demonstrated in Fig. 4: an original microscopical image (a) is fed into the segmentation module, (b) is the segmentation results with the mask; next, the individual chromosomes (c) are extracted and positioned (d); the images were then cropped by removing redundant edge areas (e) (cropped ICIs). Finally, the cropped ICIs are fed to the classification module to identify their types and form the predicted karyotyping image (f) (predicted KI). FKA and OCA are employed to evaluate the performance. Specialists examined the results of the fully karyotyping procedure, given in table III.

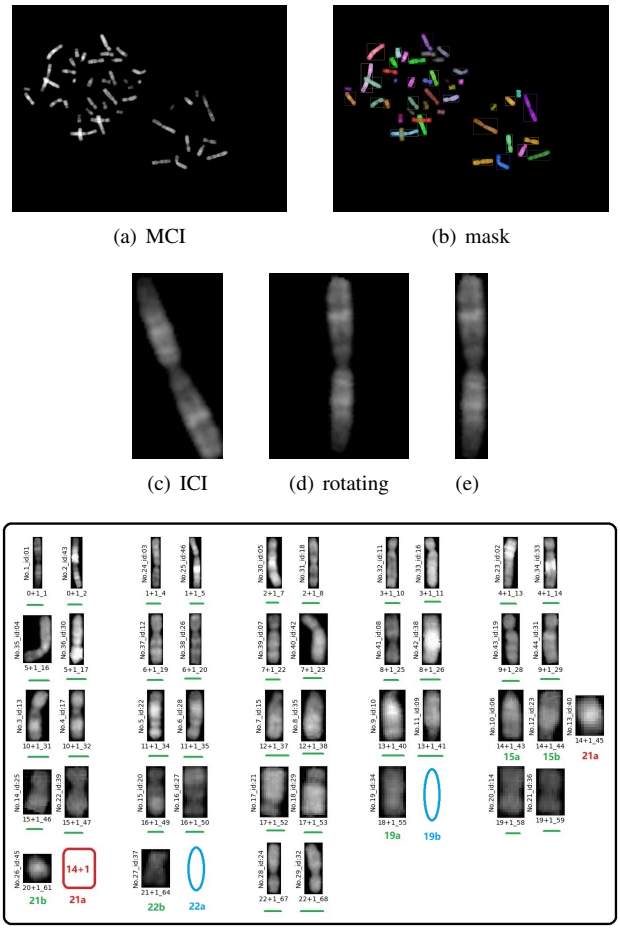
Table III indicates that with the proposed Render-Attention-Architecture, FKA and OCA achieved 83.70% and 85.65% on BioImLab, 89.05% and 89.26% on the private dataset, respectively. Fig. 5 (a) and (b) demonstrates the confusion matrix on the testing sets of two datasets. Suppose a predicted ICI is not ensured for correct or does not belong to any class, which is likely caused by poor segmentation. In that case, it will be classified as background to calculate recall and precision. It is also found in Fig. 5:

(1) In BioImLab dataset (Fig. 5 (a)), 616 (83.70%) ICIs are successfully recognized among all 736 images, 120 (16.30%) ICIs failed, in which 36 (5%) are labelled as background. Chromosomes from No. 21 only obtain a 37.5% correct rate (12 out of 32), indicating it is the most challenging class. It is also used in diagnosing Down's syndrome. *Chromosomes from No. 14 are wrongly predicted as No. 15 for ten times out of 32, and it is further discussed in Fig. 5 (c) to (j).*

(2) Among 3910 ICIs from the private dataset (Fig. 5 (b)), 3482 (89.05%) are correctly classified, and 428 (10.95%) are the wrong classification in which 230 (5.88%) ICIs assigned as background. No. 24 is the most confusing chromosome,

TABLE III: The results of prediction of MCI with Render-Attention-Architecture: *The results were identified and provided by specialists.* The head of '$\geqslant 42$' indicated that in one case, at least 42 ICIs (91.3% of 46) were corrected classified in the predicted KI, and '< 33' denotes no more than 33 (71.7% of 46) were identified. The column of the head is the number of cases that match the condition.

dataset	total case	$\geqslant 42$ (91.3%)	$\geqslant 37$ (80.4%) < 42 (91.3%)	$\geqslant 33$ (71.7%) < 37 (80.4%)	< 33 (71.7%)	OCA (%)	Fully Karyotyping		
							Accuracy (%) (FKA)	Recall (%)	Precision (%)
BioImLab	16	5	7	2	2	85.65	83.70	83.85	**89.75**
private	85	48	25	8	4	89.26	89.05	88.73	**94.22**



(a) MCI (b) mask

(c) ICI (d) rotating (e)

(f) KI

Fig. 4: Example of joint karyotyping procedure on case No.78 of BioImLab: (a): microscopical chromosome image, (b): segmentation results with masks, (c): individual chromosome image separated from (b), (d): image of (c) after 'rotating', (e): image cropping by removing redundant edge background from (d), (f): final classification results. All classified ICIs were put together. Only 44 ICIs were segmented from No.78, and 43 were correctly classified. Chromosome 21a, marked with red, was wrongly classified as 15; the chromosomes of 19b and 22a, marked with blue, were not segmented out.

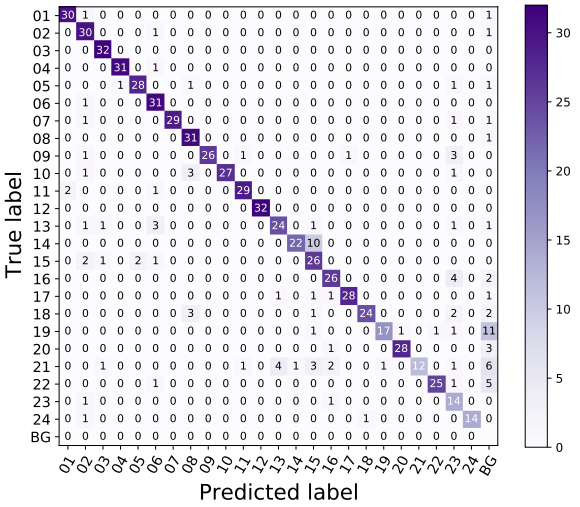
wrongly predicted to the background 23 times out of 170, partly caused by a cut-off in the segmentation procedure.

IV. Discussion and Conclusions

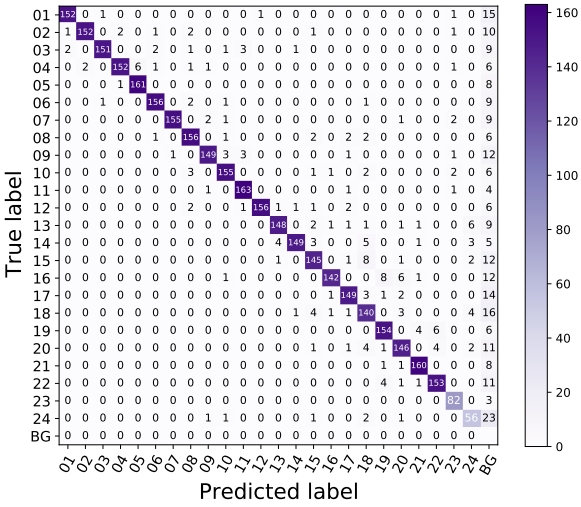
Class Activation Mapping (CAM) was proposed by Zhou *et al.* in 2016 to expose the implicit attention of an image for the CNN network by localizing deep representation [47]. By using Grad-CAM [48] to add a heat-map, the most focused area can be recognized when a model is predicting the label of an input image. Fig. 5 (c) to (j) demonstrates the CAM examples from case 44 in BioImLab, in which all ICIs of class 14 are classified as 15. The heat-map images appear more colorful in the right bottom part (red and green) than the top (blue), indicating the network focuses more on these areas. We assume that one reason for ICIs from class 14 being wrongly predicted as class 15 in Fig. 5 (c) to (j) is that the right bottom parts are too similar.

To the best of our knowledge, this work is the first to achieve fully automatic karyotyping (from MCI to KI) with the metric of fully karyotyping precision beyond 89% on two challenging datasets and chromosomes were stained with different techniques, i.e., 89.75% on the public Q-band BioImLab dataset and 94.22% on a G-band private dataset. A medical doctor verifies the results. Mano et al. report a classification accuracy of 95.04% on the CEGMR dataset [8]. Still, the following limitations are worth noting: (1) The method describes obtaining KI by classifying segmentation output from MCI and shows a simulated graphical user interface. However, it needs to be clarified in the experimental section whether the results of the classification task are obtained on the segmented output or only on the classified dataset. (2) There are few cases of chromosome overlap in MCI in CEGMR, therefore, sidesteps the challenge of chromosome segmentation task, and the more challenging BioImLab dataset was only used for classification in the work of [8], as presented on pages 11 and 15 and in Figure 20. Our research remedies these shortcomings: (1) The performance of the classification task is reported separately for the classification dataset and the segmentation output. (2) The experimental dataset is more challenging, containing touching and overlapping chromosomes in MCI. One is a public dataset (BioImLab), which allows the results to be compared with other studies.

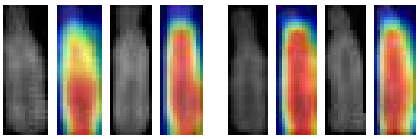
This research proposes a novel approach to improve the edge fineness of chromosome segmentation instances and integrate segmentation and classification tasks to achieve a fully automatic karyotyping architecture. This paper has solved the complex problems encountered in the clinical application of automatic karyotyping to a large extent. With the automatic predicted karyotyping images, the computer-assisted task of



(a) confusion matrix of BioImLab dataset



(b) confusion matrix of private dataset



(c) 14A (d) 14A (e) 14B (f) 14B (g) 15a (h) 15a (i) 15b (j) 15b

Fig. 5: (a) and (b) are Confusion matrix from two datasets: The horizontal axis is the predicted label of ICIs, and the vertical axis is the true label, 'BG' is short for 'background,' which does not belong to any class. The diagonal represents the correct predictions, and the others are wrong predictions. (c) to (j) are Example of heat-map images from BioImLab: The coloured image is the CAM results of the grey ICI left to it. The label below each image is the ground truth label. (c) and (e) are wrongly predicted as class 15, marked with capital 'A' or 'B.' In this case, 42 ICIs are successfully predicted from 46 segment predictions, i.e., both FAR and OCA are 91.30%.

specific chromosome abnormal detection can be further investigated.

Although the fully automatic karyotyping method presented in this study achieved good results and hence may have practical value, the methods can be further improved in a few aspects: (1) The segmentation architecture is likely to be improved by studying the chromosome hard anchor [30] to optimize bounding box proposals and, consequently, to increase the capability of recognizing long curved chromosomes. (2) Taking an MCI (cell) as the fundamental object, the predicted KI classifies 23 classes, with each class containing two ICIs (for a normal cell). This characteristic can be used and explored by studying the 'cell loss' or taking additional predictions (e.g., top three classification prediction) into account.

Declaration of Competing Interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships.

Acknowledgment

This study uses a public dataset and a dataset from a local precision medical company to verify the proposed method. The local company has signed agreements from all data providers in their data collection. For this project the hosting university has assessed the ethical requirement and an ethical clearance certificate has been issued.

References

[1] F. Abid and L. Hamami. A survey of neural network based automated systems for human chromosome classification. *Artificial Intelligence Review*, 49(1):41–56, Jan 2018.

[2] E. Schrock et al. Multicolor spectral karyotyping of human chromosomes. *science*, 273(5274):494–497, JUL 1996.

[3] W. C. Schwartzkopf, A. C. Bovik, and B. L. Evans. Maximum-likelihood techniques for joint segmentation-classification of multispectral chromosome images. *IEEE Transactions on Medical Imaging*, 24(12):1593–1610, 2005.

[4] E. Poletti, E. Grisan, and A. Ruggeri. A modular framework for the automatic classification of chromosomes in q-band images. *Computer Methods and Programs in Biomedicine*, 105(2):120–130, 2012.

[5] B. Feng et al. Down syndrome prediction/screening model based on deep learning and illumina genotyping array. In *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 347–352, Nov 2017.

[6] D. A. Griffiths. Shifting syndromes: Sex chromosome variations and intersex classifications. *Social Studies of Science*, 48(1):125–148, 2018.

[7] F. Mitelman. Recurrent chromosome aberrations in cancer. *Mutation research-Reviews in Mutation Research*, 462(2-3):247–253, APR 2000.

[8] M. S. Kharraz, L. A. Elrefaei, and M. A. Fadel. Automated system for chromosome karyotyping to recognize the most common numerical abnormalities using deep learning. *IEEE Access*, 8:157727–157747, 2020.

[9] E. Poletti, F. Zappelli, A. Ruggeri, and E. Grisan. A review of thresholding strategies applied to human chromosome segmentation. *Computer Methods and Programs in Biomedicine*, 108(2):679–688, 2012.

[10] D. Somasundaram and V. R. Vijay Kumar. Separation of overlapped chromosomes and pairing of similar chromosomes for karyotyping analysis. *Measurement*, 48:274–281, FEB 2014.

[11] E. Poletti, E. Grisan, and A. Ruggeri. Automatic classification of chromosomes in q-band images. In *2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1911–1914, 2008.

[12] M. Wang, T. Z. Huang, J. Y. Li, and Y. P. Wang. A patch-based tensor decomposition algorithm for m-fish image classification. *Cytometry Part A*, 91A(6):622–632, 2017.

[13] XW. Wang, B. Zheng, SB. Li, JJ. Mulvihill, MC. Wood, and H. Liu. Automated classification of metaphase chromosomes: Optimization of an adaptive computerized scheme. *Journal of Biomedical Informatics*, 42(1):22 – 31, 2009.

[14] Y.L. Qin et al. Varifocal-net: A chromosome classification approach using deep convolutional networks. *IEEE Transactions on Medical Imaging*, 38(11):2569–2581, 2019.

[15] C. Lin et al. Cir-net: Automatic classification of human chromosome based on inception-resnet architecture. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pages 1–1, 2020.

[16] C. Lin et al. A novel chromosome cluster types identification method using resnext wsl model. *Medical Image Analysis*, 69(101943), 2021.

[17] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[18] E. Pardo, J. M. T. Morgado, and N. Malpica. Semantic segmentation of mfish images using convolutional networks. *Cytometry Part A*, 93A(6):620–627, 2018.

[19] N. Xie, X. Li, K. Li, Y. Yang, and H. T. Shen. Statistical karyotype analysis using cnn and geometric optimization. *IEEE Access*, 7:179445–179453, 2019.

[20] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *3rd International Conference on Learning Representations*, May 2015.

[21] K. M. He, X. Y. Zhang, S. Q. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *Computer Vision - Eccv 2014, Pt Iii*, 8691:346–361, 2014.

[22] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[23] F. Prinzi, M. Insalaco, A. Orlando, S. Gaglio, and S. Vitabile. A yolo-based model for breast cancer detection in mammograms. *Cognitive Computation*, 16(1):107–120, 2024.

[24] R. Rong, H. Sheng, K. W. Jin, F. Wu, D. Luo, Z. Wen, C. Tang, D.M. Yang, L. Jia, M. Amgad, et al. A deep learning approach for histology-based nucleus segmentation and tumor microenvironment characterization. *Modern Pathology*, 36(8):100196, 2023.

[25] S.S.G. Bagi, B. Moshiri, H.G. Garakani, M. Crowley, and P. Mehrannia. Real-time pedestrian detection using enhanced representations from light-weight yolo network. In *2022 8th International Conference on Control, Decision and Information Technologies (CoDIT)*, volume 1, pages 1524–1529. IEEE, 2022.

[26] S. Mishra and S. Jabin. Real-time pedestrian detection using yolo. In *2023 International Conference on Recent Advances in Electrical, Electronics & Digital Healthcare Technologies (REEDCON)*, pages 84–88. IEEE, 2023.

[27] H. Bai, T. Zhang, C. Lu, W. Chen, F. Xu, and Z. B. Han. Chromosome extraction based on u-net and yolov3. *IEEE Access*, 8:178563–178569, 2020.

[28] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, Venice, Italy, Oct 2017.

[29] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525, 2017.

[30] L. Xiao et al. Deepacev2: Automated chromosome enumeration in metaphase cell images using deep convolutional neural networks. *IEEE Transactions on Medical Imaging*, pages 1–1, 2020.

[31] D. Somasundaram. Machine learning approach for homolog chromosome classification. *International Journal of Imaging Systems and Technology*, 29(2):161–167, 2019.

[32] N. Madian, KB. Jayanthi, D. Somasundaram, and S. Suresh. Identifying centromere position of human chromosome images using contour and shape based analysis. *Measurement*, 144:243–259, 2019.

[33] D. Somasundaram, N. Madian, and S. Suresh. Mathematical approach for segmenting chromosome clusters in metaspread images. *Experimental Cell Research*, 418(1):113251, 2022.

[34] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, June 2017.

[35] A. Kirillov, Y. Wu, K. He, and R. Girshick. Pointrend: Image segmentation as rendering. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9796–9805, 2020.

[36] M. Tatarchenko, A. Dosovitskiy, and T. Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2107–2115, 2017.

[37] G. Gkioxari, J. Johnson, and J. Malik. Mesh r-cnn. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9784–9794, 2019.

[38] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4455–4465, 2019.

[39] YX. Wu, A. Kirillov, F. Massa, W. Y. Lo, and R. Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

[40] KM. He, XY. Zhang, SQ. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, Las Vegas, NV, USA, June 2016.

[41] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5987–5995, 2017.

[42] L. Shen J. Hu and G. Sun. Squeeze-and-excitation networks. In *31st IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE Conference on Computer Vision and Pattern Recognition, pages 7132–7141, 2018.

[43] YD. Wen, KP. Zhang, ZF. Li, and Y. Qiao. A discriminative feature learning approach for deep face recognition. In *Computer Vision – ECCV 2016*, pages 499–515, Amsterdam, Netherlands, 2016.

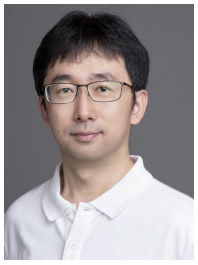
[44] CY. Wang, LM. Yu, X. Zhu, JL. Su, and F. Ma. Extended resnet and label feature vector based chromosome classification. *IEEE Access*, 8:201098–201108, 2020.

[45] R. Padilla, S. L. Netto, and E. A. B. da Silva. A survey on performance metrics for object-detection algorithms. In *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 237–242, 2020.

[46] C. Wang, L. Yu, J. Su, T. Mahy, V. Selis, C. Yang, and F. Ma. Down syndrome detection with swin transformer architecture. *Biomedical Signal Processing and Control*, 86:105199, 2023.

[47] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2921–2929, June 2016.

[48] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 618–626, 2017.



Chengyu Wang received the bachelor's degree from Jiangnan University, the master's degree from Southeast University, and the Ph.D. degree from the University of Liverpool.

He is currently an Assistant Professor with the HeXie Management Research Centre, College of Industry-Entrepreneurs, Xi'an Jiaotong-Liverpool University. In the past five years, he has published five academic papers as the first author in SCI journals and international conferences. His main research interests include medical image processing, deep learning, large language models, multimodal learning, and integrating HeXie management theory with computable organization models and bionics.

Limin Yu (Member, IEEE) received the B.Eng. degree in telecommunications engineering and the M.Sc. degree in radio physics/underwater acoustic communications from Xiamen University, China, and the Ph.D. degree in telecommunications engineering from The University of Adelaide, Australia.

She is currently an Associate Professor with the Electrical and Electronic Engineering Department, Xi'an Jiaotong-Liverpool University, Suzhou, China. Her research interests include sonar detection, wavelet analysis, filter bank design, broadband system design, high-mobility wireless local area networks, sensor networks, and medical image analysis.

Jionglong Su received the Ph.D. degree in statistics in Warwick and the Ph.D. degree in automatic control and systems engineering in Sheffield.

Before joining XJTLU, he was with the University of Warwick, University College London, and Nazarbayev University, where he was the MathsHead. He is currently the Deputy Dean of the School of Artificial Intelligence and Advanced Computing, XJTLU Entrepreneur College (Taicang). His research interests include bioinformatics, artificial intelligence, and medical image processing.

Juming Shen received the Ph.D. degree in higher educational studies from Queensland University of Technology, in 2013. He is currently the Head of the Learning Institute for Future Excellence, Xi'an Jiaotong-Liverpool University. He has published dozens of international peer-reviewed articles. His research interests include digital education, educational semiotics, and educational policy studies.

Valerio Selis received the Ph.D. degree in electrical engineering and electronics from the University of Liverpool, Liverpool, U.K. Currently, he is a Lecturer with the Advanced Networks Research Group, University of Liverpool. His recent research was focused on the Adaptive Communications Transmission Interface Project in collaboration with Queen's University Belfast, Plextek Ltd., and the Defence Science and Technology Laboratory (Dstl). Moreover, he has been the Product Development Director of Traffic Observation via Management Ltd. His research interests include trust management, the Internet of Things, machine learning, molecular communications, nano-networks, and nanomedicine.

Chunxiao Yang received the master's degree in clinical medicine and the Ph.D. degree in medicine from Fourth Military Medical University, China.

From 2014 to 2015, she studied as a Visiting Scholar with the Department of Pathology, Brigham and Women's Hospital, Harvard Medical School. Her primary research interests include the clinical cytogenetics of malignant tumors, including different subtypes of leukemia and lymphoma.

Fei Ma received the B.Sc. and M.Sc. degrees in computational mathematics from Xiamen University, China, and the Ph.D. degree in applied mathematics from Flinders University, Australia.

Upon joining Xi'an Jiaotong-Liverpool University, Suzhou, China, he has been an Analyst with Symbion, Australia, a Research Associate and a Lecturer with Flinders University, and a Software Engineer with Kingdee Company Ltd., Shenzhen. He is currently a Professor with the Applied Mathematics Department, Xi'an Jiaotong-Liverpool University. His research interests include medical and biomedical image analysis, including computer-aided breast cancer diagnosis with mammogram, cell image and chromosome image analysis, big data analytics, AGV scheduling and routing, and nonnegative matrices.