

## Research Article

# A New Multiple-Distribution GAN Model to Solve Complexity in End-to-End Chromosome Karyotyping

Yirui Wu<sup>1</sup>,<sup>ORCID</sup> Xiao Tan,<sup>2</sup> and Tong Lu<sup>3</sup>

<sup>1</sup>College of Computer and Information, Hohai University, Nanjing, China

<sup>2</sup>Nanjing Tuofan Information Technology Company, Nanjing, China

<sup>3</sup>National Key Lab for Novel Software Technology, Nanjing University, Nanjing, China

Correspondence should be addressed to Yirui Wu; wuyirui@hhu.edu.cn

Received 7 February 2020; Revised 25 March 2020; Accepted 10 April 2020; Published 11 May 2020

Guest Editor: Xuyun Zhang

Copyright © 2020 Yirui Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With significant development of Internet of medical things (IoMT) and cloud-fog-edge computing, medical industry is now involving medical big data to improve quality of service in patient care. Karyotyping refers classifying human chromosomes. However, performing karyotyping task generally requires domain expertise in cytogenetics, long-period experience for high accuracy, and considerable manual efforts. An end-to-end chromosome karyotype analysis system is proposed over medical big data to automatically and accurately perform chromosome related tasks of detection, segmentation, and classification. Facing image data generated and collected by means of edge computing, we firstly utilize visual feature to generate chromosome candidates with Extremal Regions (ER) technology. Due to severe occlusion and cross overlapping, we utilize ring radius transform to cluster pixels with same property to approximate chromosome shapes. To solve the problem of unbalanced and small dataset by covering diverse data patterns, we proposed multidistributed generated advertising network (MD-GAN) to perform data enhancement by generating additional training samples. Afterwards, we fine-tune CNN for chromosome classification task by involving generated and sufficient training images. Through experiments in self-collected datasets, the proposed method achieves high accuracy in tasks of chromosome detection, segmentation, and classification. Moreover, experimental results prove that MD-GAN-based data enhancement contributes to classification results of CNN to a certain extent.

## 1. Introduction

Traditionally, medical images and sensor data are most common medical data to know health condition of patients. With big progress achieved by Internet of Medical Things (IoMT) [1], medical industry has greatly been improved with a new dimension towards intelligent and complex system based on multiple and multimodal medical data offered by environment of IoMT and edge computing [2, 3]. However, growing number and complexity of medical data require highly distinguished models to automatically perform identification or diagnosis. Deep learning models are thus adopted to handle large volume of medical data [4], due to their scalability to process either big data or small size data and significant power to analyze complex IoMT data with highly nonlinear functional system. Based on all these

advances in IoMT and deep learning, we aim to offer a case study on how to improve a specific medical application, i.e., chromosome karyotyping.

Essentially, karyotyping requires cytologists to pay attention on the problem of numerical abnormalities of chromosomes that may result in some genetic diseases, such as Down syndrome, cancer, genetic diseases, and birth defects [5]. Due to the fact that human usually own 24 categories of chromosomes (including 22 kinds of autosomes and 2 sex chromosomes), karyotyping could be comprehended as the process to identify and classify 24 classes of chromosomes from input cell pictures.

Karyotype analysis is a highly professional work even after years of expertise [6], requiring considerable manual efforts and quantity of time to produce accurate karyotyping results. Essentially, difficulty of chromosome karyotyping

lies in several aspects: (1) Karyotyping requires recognizing 24 classes of chromosomes, which is an essential multiple-class classification problem. Besides, shape details of same class chromosome vary little from person to person, which increases difficulty and needs researchers to extract high distinguishing features for accurate results. (2) Occlusions and touchingness among chromosomes often appear in input cell images, which requires an appropriate segmentation algorithm to generate relatively complete chromosome samples for further classification. (3) Chromosomes would appear with unpredictable shape, which is caused by non-rigid nature of chromosomes. Such phenomena would make it difficult to accurately extract media axes, thus resulting in accuracy drop with traditional chromosome methods of utilizing media axes based features. (4) Uneven Giemsa staining would produce local intensity varieties and unclear shape boundaries, which do harm to accurately recognize chromosomes with intensity or shape features. We show examples of these four challenges in Figure 1, where problems of multiple classes, occluding chromosomes, unpredictable shape, and local intensity varieties have been showed in (a), (b), (c), and (d), respectively. Above all, chromosome karyotyping is a tedious procedure for manual operations, and automatical chromosome karyotyping technology requires to face and solve several domain problems.

Based on the analysis of common karyotyping methods, we conclude that former methods [5, 7] consist of the following 4 steps (Figure 2):

- (1) *Step 1. Detection.* The method has to find chromosomes in cell images at first step. Note that separation of chromosomes from the image background is affected by the noise of nucleus, small staining points and so on.
- (2) *Step 2. Rough segmentation.* Once chromosome candidates are detected in cell images, the method has to decide candidates to be single chromosomes or chromosome clusters. Single chromosomes could skip Step 3 for classification, while chromosome clusters has to be performed with overlap segmentation.
- (3) *Step 3. Overlap segmentation.* In this step, most methods utilize semiautomatic algorithms to help separate touching and overlapping chromosomes, due to the complicated nature of chromosome clusters.
- (4) *Step 4. Classification.* After all the previous steps, methods should classify types of each single chromosome by either utilizing manual features [8, 9], such as size, centromere position, and banding pattern, or using deep learning methods [10, 11] to automatically extract unique features for classification. At last, method would arrange chromosomes by pairs to output a standard karyotyped image as shown in the last step of Figure 2.

With all these analysis, we could find that research on karyotyping is attractive and difficult, due to its special request to perform visual recognition task under sufficient medical expertise. Facing all these difficulties in whole pipeline and steps, we have cooperated with several doctors and biologists on developing new tools to assist automatical chromosome karyotyping, especially aiming at simplifying the current processes to facilitate user-friendly and end-to-end chromosome karyotyping. We thus proposed a karyotype analysis model that can expedite the task of karyotyping to maximal automation.

Figure 3 gives the pipeline of the proposed karyotype analysis model, which comprises three major steps. In first step represented as (a), chromosome candidates are extracted from input cell pictures. Applying multiple and various filters with features of low computation burden could help generate more regions of interests in an efficient way. Besides, the adopted filters own high capability to work under nature of noise and uneven staining. During the next step represented as (b), we divide overlapping and touching chromosome clusters into single chromosomes by utilizing geometry and intensity information. With the intention of maximizing automation, the proposed method is stable and efficient with instant visual feedback to make user-friendly segmentation possible. During classification step represented as (c), we first construct a multiple distribution GAN (MD-GAN) network to produce lots of labeled chromosome images based on the supposition of multiple data distributions. MD-GAN is designed with multiple distribution generators under a reasonable distribution consumption on original data, other than single generator adopted by the original GAN structure. By adopting more generators, MD-GAN is able to effectively cover diverse data modes and generate more labeled samples. After sample generation, we further fine-tune CNN classifier with enough and diverse samples to achieve results on chromosome labels. Such results are finally arranged in a karyotype picture to show classification output.

The contributions of the proposed method are threefold:

- (i) Facing challenges brought by the environment of IoMT and edge computing, we propose an end-to-end chromosome karyotyping model with accurate classification results. Stepwise improvement and high distinguishing power of deep neural networks help not only solve particular and domain problems of chromosome karyotyping, but also obtain high accuracy facing self-collected and low-quality chromosome dataset.
- (ii) Inspired by object proposal methods [12] of utilizing low computation and effective classifiers, ER method with various filters are applied to locate chromosome candidates, which not only avoids wrong detection results brought by complexity of IoMT environment, but also reduces computation by extracting heavy-computed feature representation for detection.

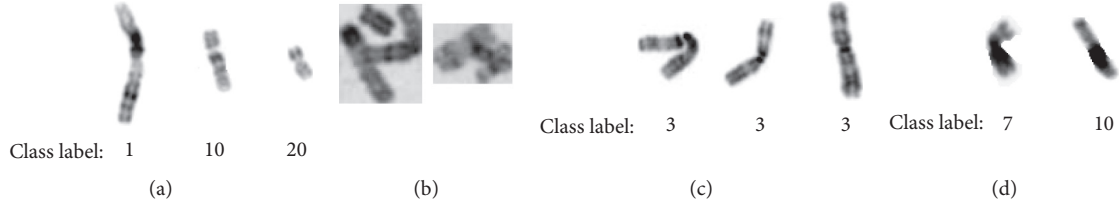


FIGURE 1: Challenges for chromosome karyotyping: (a) total 24 classes for classification (including 22 pairs of autosomes and 2 sex chromosomes XY), (b) occluding and touching chromosomes, (c) unpredictable shape with curving, and (d) local intensity varieties and unclear shape boundary.

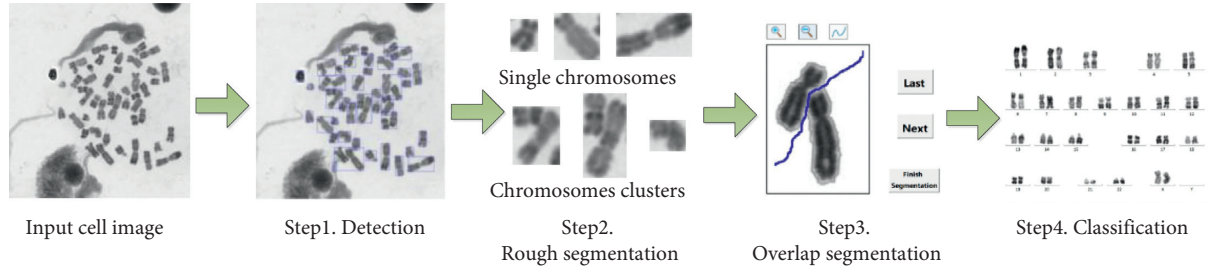


FIGURE 2: General steps of former automatic chromosome karyotyping method: (1) detecting chromosome regions from input cell image, (2) rough segmentation to generate single chromosomes and chromosomes clusters, (3) overlap segmentation on chromosome clusters to achieve individual chromosomes, and (4) apply classifiers on individual chromosomes to perform classification.

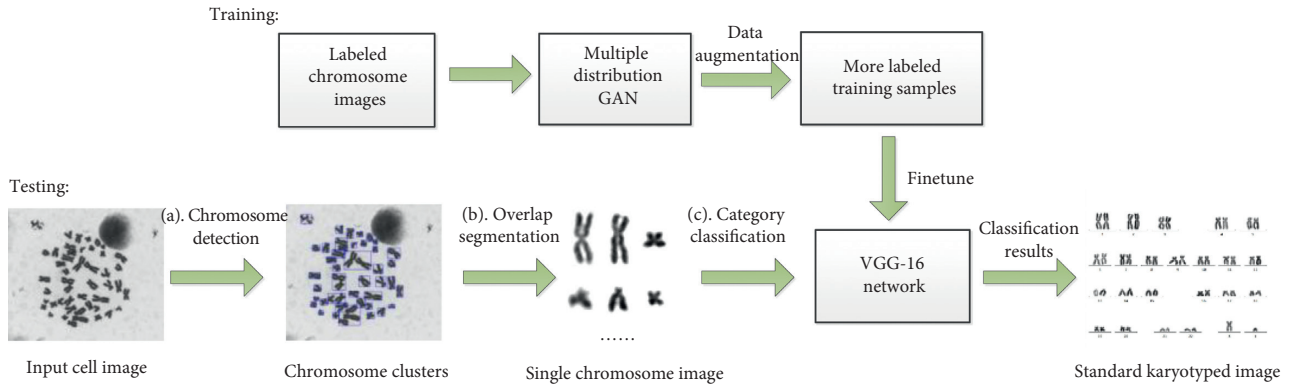


FIGURE 3: Pipeline for our chromosome karyotyping method: (a) detecting chromosome candidates with ER technology and various filters, (b) segmenting overlap or touching chromosome clusters into individual ones with the proposed segmentation tool, (c) first utilizing MD-GAN to perform data augmentation and then finetuning pretrained CNN with more training samples to classify chromosome category of individual chromosomes. Classification results are finally sorted as a standard karyotyped image.

- (iii) The proposed MD-GAN employs a mixture of data distributions to generate diverse training samples instead of using multiple generators, which not only overcomes the mode collapsing problem, but also saves computation and reduces complexity. We believe MD-GAN is beneficial to reduce complexity and leads to a reasonable deep learning model.

## 2. Related Work

We roughly categorize related methods and techniques into two classes, namely, chromosome karyotyping and data augmentation with GAN.

**2.1. Chromosome Karyotyping.** Karyotyping usually consists of chromosome detection, overlap segmentation, and category classification. The chromosome detection is designed to distinguish chromosomes from the background. Based on binarization, researchers either use global threshold with the Otsu method [13] or rethresholding scheme [14] to perform detection. However, detection processes probably fail due to uneven Giemsa staining, which is highly affected by setting threshold values. Other methods [15] extract features for detection on the basis of spatial and frequency domain information. However, those approaches are time-consuming due to the learning structure. Wu et al. [16] explores Extremal Regions (ER) to perform task of text candidate detection. Inspired by their work, ER detection method is

used to detect chromosomes candidates in contrast to complex cell background.

Although many methods are proposed for automated segmentation [17], it is still difficult to complete chromosome segmentation task, due to unpredictable shape and appearance caused by the nonrigid nature of chromosomes. Earlier, Lerner et al. [18] successfully realize classification driven segmentation, which combines correct selection of cluster untangling in the classification stage. Then, Charters and Graham [19] propose a scheme to collect subchromosomal banded spectral templates, which performs well in training models and successfully complete tasks, i.e., first identify super-nodes of chromosomal segments and then assemble the microsegments in a bottom up manner. Minaee et al. [15] propose an iterative version of segmentation method by utilizing geometric features of chromosome boundary, until all individual chromosomes are separated.

Since accurate segmentation approximations can be tedious for wider usage, and researchers begin to address the problem of obtaining tight boundary boxes. For example, a nonexpert crowdsourcing method is used to segment chromosomes from testing cell samples in [11]. Most systems currently adopted to perform automatic chromosome segmentation require interactive operations, leading it to be not suitable for massive work. How to automatically segment chromosome clusters in maximal degree and transform it as an appropriate version on cloud computing [20, 21] or edge computing [22–24] remains a hot topic.

During category classification, former methods usually pre-process chromosomes by straightening [8], since the curved and bent orientation is regarded as the critical factor to impede the performance. After straightening, researchers would extract features by manual designing [9], such as relative length of chromosome [25], centromere index [26], band profile [27], and so on. After then, such methods adopt classifiers [28] to achieve classification results. Nevertheless, methods built on manual design features may lose useful information and lead to low accuracy in classification.

Deep neural network has achieved significant performance in performing a large number of tasks, which encourages researchers to apply deep models on tasks of chromosome karyotyping [7]. Sharma et al. [11] firstly proposed a CNN-based method for classification of straightened and normalized chromosomes, which has surpassed the performance of conventional methods by a significant margin. Afterwards, Varifocal-Net [29] is proposed to highlight the capacity of zooming to local areas in an automatic way. Their Global-scale network (GNet) is designed to complete two tasks simultaneously, i.e., obtaining global features and identifying particular local areas. Meanwhile, local-scale network (L-Net) is responsible to locate local areas, which are further applied to extract local features. However, these methods have strict requirement for quantity of labeled training samples. Therefore, their performance cannot remain consistent when labeled data is scarce, even though sometime cloud-based privacy protection systems can help to some extent [30–32]. Chromosome classification is still a difficult task under conditions of small data and multiple categories.

**2.2. Data Augmentation with GAN.** One of the biggest matters for usage of deep learning models in domain of medical image analysis is lack of datasets with quantity of labeled samples. Essentially, researchers have tried other kinds of technologies from big data to solve this problem [33, 34]. Moreover, annotation task on medical images not only requires quantity of money and time-consuming to complete, but also has a high request on availability of professional doctors. Performance of deep learning models is highly related with size of training dataset.

GAN provides a new way to offer deep feature representation for effective learning. Owing to the GAN-based enhancement method, a large number of unlabeled images can be involved into learning process, since GAN requires little prior knowledge and is easy to implement. Thanks to its strong distributed modeling ability, GAN is quite fit in increasing number of training samples to make deep learning algorithms more effective [35].

Basic idea of GAN is built on the basis of a two-player game between generators and discriminators. In Figure 4, we demonstrate the basic structure of a typical GAN to generate a handwritten digital image. We can notice the generator generates images which should make the discriminator to judge as a natural image rather than a pseudoinage, while discriminator determines whether the input image generated by the generator is natural or not. After the training process, the GAN model will reach Nash equilibrium and could comprehend inherent representation of real images to continuously create a large number of images.

Thanks to the continuous optimization and development of GAN structure, more GAN-based applications have been developed. For example, Zheng et al. [35] propose the label smoothing regularization to assigns labels for unlabeled images generated by GAN, which could regularize the supervised model and improve the baseline. Zhu et al. [36] propose a data augmentation method to improve the classification of emotion images using GAN, which successfully refines data distribution and discovers proper margins among different categories. Similar to the proposed work, Bowles et al. [37] introduce Progressive Growing of GANs (PGGAN) network for two brain segmentation tasks, which proves data generated by generator can play an important role in training dataset. Their network reports that dice similarity coefficient (DSC) could be improved by 1–5 percentage. Frid-Adar et al. [38] adopt GAN to generate synthetic medical images on the basis of limited CT image datasets of 182 liver lesions. Afterwards, they apply GAN-generated sample on training CNN for classification task, which has achieved a significant improvement. These successful applications on variant domains, especially on medical images, encourage us to develop domain-adaptive GAN as a novel method of data augmentation for chromosome classification.

Essentially, modeling a specific GAN for medical domain is difficult, since GAN models often come across model collapse, i.e., the generator focuses to generate samples in several patterns rather than entire data space [39]. To tackle this matter, Salimans et al. [39] make use of minibatch discrimination trick to allow the discriminator to detect



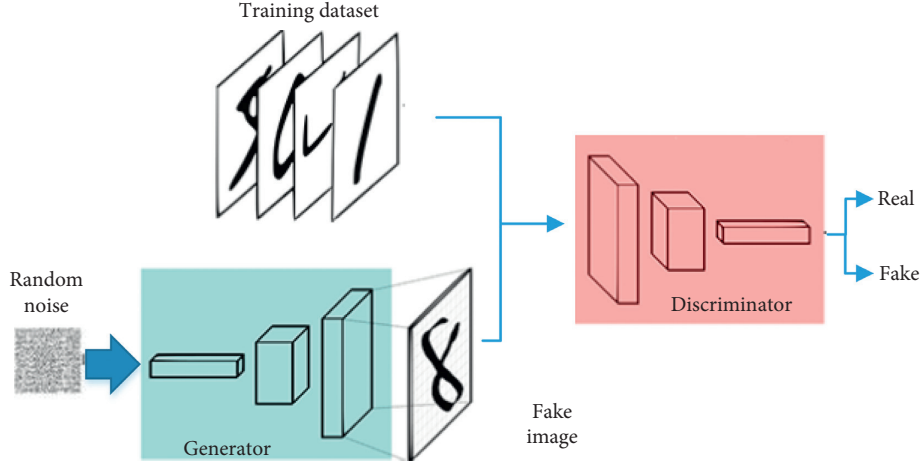


FIGURE 4: Architecture of a GAN model to generate samples of handwritten digital number images, where we can notice that random noise is introduced as input for generator.

samples that are unusually similar to other generated samples. Considering data augmentation as a method to transform task related data and simultaneously keep category labels, Ratner et al. [40] design a generative sequence model to perform domain-specific data transformations. Their model could be settled by users to design arbitrary, nondeterministic transformation function, thus fitting to apply in a variety of fields.

Another solution for such problem is the modification on GAN structure to decrease gradient loss for effective usage of all data. CycleGAN model [41] uses CNN model as a classifier and a novel gaming mechanism, i.e., a consistent loop structure between generator and discriminator, which shows large improvement of performance in data augmentation and classification accuracy during experiments. Hoang et al. [42] design an objective function to approximate data manifold by their induced distributions during training, whilst encouraging them to specialize in different data modes. However, their proposed method is high in computation with quantity of generators, compared with multiple distribution for generation.

Compared with original GAN, MD-GAN improves performance by utilizing multiple distributions for data augmentation, which is quite similar with core idea of Hoang et al. [42]. The reason to apply multiple distributions to construct generator lies in the fact that simple input of ordinal GAN might lead to similar outputs. In other words, original GAN can do harm to final classification by generating chromosomes with close appearances. Therefore, it is intuitive to construct multiple generators for high mode diversity, which brings disadvantages of complexity and high computation. To keep balance between performance and complexity, we propose to use multiple distributions other than multiple generators for chromosome generation, which is the key difference between MD-GAN and GAN.

### 3. Chromosome Candidate Detection

Based on the environment of IoMT with different of cameras, sensors, and sampling methods [43], it is easy for

researchers to collect quantity of cell images. By analyzing collected cell images, we find that chromosome samples are affected by two factors, i.e., quality of Giemsa staining and magnification times, since multiple categories of sensors and cameras adopted by IoMT bring complexity and multidimensional property of medical data [44, 45]. Specifically, uneven staining leads to different levels of contrast and unclear shape boundaries; meanwhile, magnification times make chromosome size inconsistent. Moreover, disruptors are similar with chromosomes in appearance, which might be misclassified as chromosomes. All these difficulties provide challenges for accurate localization and classification of chromosomes.

Following the consideration of constructing simple and effective classifiers, we firstly explore Extremal Regions (ER) [46] algorithm to generate candidate chromosome regions, which performs grouping for pixels of input cell images based on intensity contrast characteristics. The reason to adopt ER algorithm lies in several reasons. Firstly, ER is able to produce a small number of chromosome candidates with strictly similar intensity property. Secondly, ER is flexible to combine expert knowledge by constructing candidates filters, due to its generation of quantity of candidates. Last but not least, ER algorithm could offer more chromosome candidates to guarantee high recall rate of detection with low computation burden.

Specifically, we define an extremal region as a contiguous region  $Q$  for each pixel  $p \in Q$  satisfies

$$\begin{aligned} \exists q \in \partial Q: I(p) > I(q), \\ \text{or } I(p) < I(q), \end{aligned} \quad (1)$$

where  $I$  refers to the input cell image, the former and latter inequality represent maximum intensity region and minimum intensity region, respectively, and  $\partial Q$  is defined as outer region boundary:

$$\partial Q = q \in D, \quad Q: \exists p \in Q: q \Delta p, \quad (2)$$

where  $\Delta$  is defined as an adjacency (neighbourhood) relation. Essentially, outer region boundary  $\partial Q$  can be

comprehended as the pixel set, which is adjacent to at least one pixel of  $Q$  but not a part of  $Q$ .

We prefer ER algorithm for chromosome candidates generation other than MSER algorithm [46], since MSER algorithm is strict to generate maximally stable regions with local minimum of  $i$ , which is defined as

$$q(i) = \frac{|Q_{i+\Delta} \setminus Q_{i-\Delta}|}{|Q_i|}, \quad (3)$$

where  $Q_1, \dots, Q_{i-1}, Q_i$  is defined as nested ER sequences, operation  $|\cdot|$  represents cardinality and  $\Delta$  is a preset parameter in MSER algorithm. We show results of ER algorithm in Figure 5(a), where we can notice quantity of misclassifies required to be further processed.

Due to the existence of disruptors, chromosome candidates  $\{e_i\}$  generated by ER algorithm are low in accuracy with misclassifies, which could bring huge computation burden to latter modules, i.e., segmentation and classification module. To greatly improve accuracy of chromosome candidate detection method, we propose filter on basis of utilizing two inherent characteristics of chromosomes, i.e., shape descriptor, low-intensity variance inside each chromosome.

- (1) Geometric properties based filter: ER algorithm is easy to recognize nucleus or noisy points as chromosome candidates. Based on this observation, we propose to apply Hough algorithm to decide whether there are such noisy objects by trying to locate eclipses inside each candidate. Moreover, the proposed filter removes wrong candidates by using Euler number and candidate region area.
- (2) Intensity distribution-based filter: Inspired by observation that each chromosome has property of low-intensity variance, the proposed filter discards candidates with large-intensity variance values. On the basis of fact that accurate chromosome candidates should only have regions of background and chromosome, we construct histograms about values of intensity for each chromosome candidate. Afterwards, mean value of the maximum and sub-maximum numbers is adopted to calculate intensity variance  $V_i$  for each chromosome candidate

$$V_i = \frac{n_c \cdot \sum_{j \in e_{i,c}} (I(j) - A_{i,c})^2 + n_b \cdot \sum_{j \in e_{i,b}} (I(j) - A_{i,b})^2}{n_c + n_b}, \quad (4)$$

where subscripts  $c$  and  $b$  represent regions of chromosome and background inside chromosome candidate  $e_i$ , respectively, namely  $e_i = \{e_{i,c}, e_{i,b}\}$ ,  $n$  represents number of pixels inside different regions, and  $A$  represents average intensity values for different regions. Afterwards, the proposed method adopts chromosome candidates with low  $V_i$  as output for detection. Sample results after filtering are illustrated in Figure 5(b), where quantity of misclassifies are accurately filtered.

Figure 5(c) represents chromosome candidate results as input for latter segmentation module, where we could observe touching and overlapping candidates appear as chromosome clusters.

#### 4. Chromosome Clusters Segmentation

Inspired by the idea of [47] and high-consistency property of pixels of different segments for each overlapping chromosome candidate, we perform segmentation on candidates containing touching and overlap chromosomes by utilizing ring radius transform to cluster pixels with same property, thus approximating chromosome shape with eclipses. We have three steps in this subsection. Firstly, we achieve edge images with Canny Operator. Regrading edge images as input for the second step, we utilize RRT transform to locating medial axis, which are seed points for the third step. In the third step, we perform contour estimation with eclipse to segment overlap regions. Essentially, RRT is used to generate initial seed points with medial axis, which could greatly improve stability and convergence speed for overlap segmentation.

Firstly, we perform erosion operation on input chromosome candidate images and then obtain their correspondence edge images with Canny edge detector. Afterwards, we aim to extract seed points  $s_i$  for each candidate  $e_i$ , which could be considered as certain a priori information to perform contour estimation. Different from [47] utilizing Fast Radial Symmetry (FRS) transform on raw input images to achieve seed points, we apply ring radius transform (RRT) [48] on the extracted edge images to locate seed points. The reason to apply RRT algorithm lies in the fact that noise in raw chromosome images, like local intensity varieties and unclear shape boundaries, greatly affects performance of segmentation. With pre-processing of robust and effective Canny edge detector, we could relieve most effects brought by noise. On the basis of convinced edge images, RRT could achieve robust location results of seed points even facing difficulties of arbitrary orientation chromosome. Moreover, recall that medial axis extraction is a key procedure for traditional chromosome segmentation and classification methods, and RRT helps exactly locate media axis pixels for chromosome segmentation, which coincides with the idea of traditional methods.

By transforming the input edge image into a new form, RRT highlights local radial symmetry of input images to accurately locate the intermediate axis pixel. Specifically, values to represent radius are allocated to all pixels in the corresponding edge images, which is defined as the distance to its closest edge pixel:

$$\text{rad}(p) = \min_{\text{edge}(q)=\text{true}} \text{dist}(p, q), \quad (5)$$

where function  $\text{edge}(q)$  is utilized to examine whether pixel  $q$  is edge pixel and  $\text{dist}(p, q)$  refers to Euclidean distance from  $p$  to  $q$ . Afterwards, pixels with local minimum radius values are regarded as medial axis pixels. Finally, we utilize mean  $x$  and  $y$  values of local medial axis pixels as results of seed points  $s_i$ .

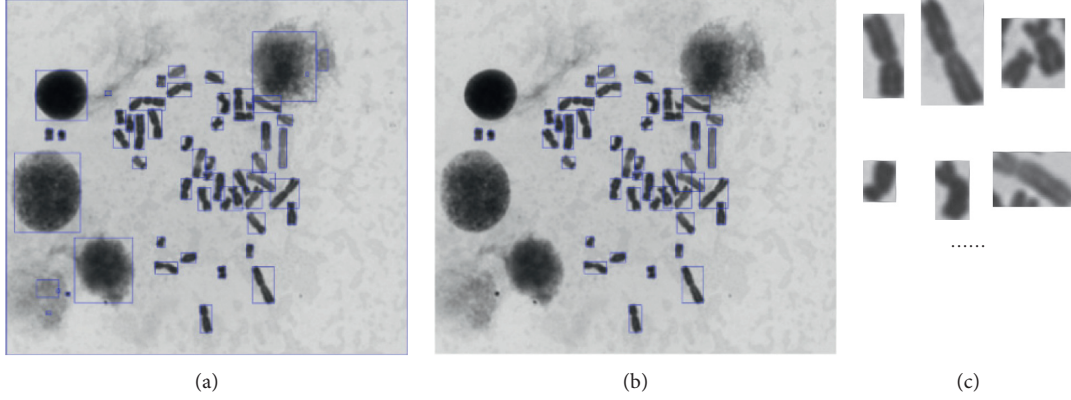


FIGURE 5: Results for candidate chromosome detection: (a) ER detection results with quantity of misclassifies, (b) chromosome candidates after geometric and intensity invariance filters, and (c) to the final results for chromosome candidates.

With localization of seed points, we decide belongs of each edge pixel  $p$  in overlapping areas with the following measurement:

$$\text{relevance}(p, s_i) = \frac{1 - \lambda}{1 + \text{dist}(p, s_i)} + \lambda \frac{\text{div}(p, s_i) + 1}{2}, \quad (6)$$

where function  $\text{dist}()$  and  $\text{div}()$  represent Euclidean distance and divergence function, respectively, and  $\lambda$  is the preset weight value. Due to the assigns of overlapping areas, some of the contour areas must be smaller than other areas. We thus fill up missing areas to complete task of contour estimation by fitting shape with ellipses, where shapes of ellipse are adopted to describe these partially observed objects. The reason to utilize ellipses other than rectangles relies on the fact that chromosomes are similar in shape with ellipses. In other words, ellipses can provide tighter boundary estimations than rectangles.

After contour estimation, we can provide useful bounding contours with ellipses as shown in Figure 6. From the resulting ellipses, we could find the individual chromosome candidates are represented in (a). Meanwhile, successfully segmented chromosome clusters are represented in (b). If the ellipse area is not an entire ellipse, we would discard these ellipses marked with blue rectangle in Figure 6. Figures 6(c) and 6(d) represent failure cases, which require manual modification to solve the problem of oversegmentation. After proper segmentation, we obtain several single chromosome images represented as  $\{x_i; i = 1, \dots, n_s\}$  required to be classified, where  $n_s$  refers to the number of single chromosome images.

## 5. Chromosome Classification with Data Augmentation

GAN-based data enhancement achieves significant performance in increasing data size and owns high discriminative ability to locate margins between similar categories. In fact, chromosome classification is a multiple-label classification task with not sufficient training samples. Therefore, applying data augmentation to expand training dataset is extremely effective in such classification tasks.

**5.1. Overall Workflow.** Following idea of data augmentation to improve classification accuracy, we try to solve the problem of unbalanced and small dataset for the training process with Multiple Distribution Generative Advertising Network (MD-GAN). Essentially, unbalanced problem in chromosome karyotyping is caused by heavily unbalanced distributions of realistic chromosome data. When applying original GAN, it could be easily trapped, i.e., generating similar samples without enough differential modes. This phenomenon makes unbalanced problem of chromosome dataset much worse with similar outputs. By adopting MD-GAN, the proposed method can guarantee to produce samples with a variety of modes, thus improving diversity of dataset to a certain extent. Afterwards, sufficient samples generated by MD-GAN are applied to fine-tune pretrained convolutional neural network (CNN) for accurate classification of chromosomes. These steps are presented in Algorithm 1, where we apply multiple MD-GAN to accomplish multiclass augmentation.

Given the discriminator  $S$  to determine true or fake of samples and the generator  $G$  to learn distribution of original data, training procedure for  $G$  could be regarded as a process to maximize the ratio of wrong classifications predicted by  $S$ . Meanwhile, training procedure for  $S$  can be regarded to minimize its own wrong classification rate. Based on these two procedures, training for GAN can thus be comprehended to minimax objective function:

$$\min_G \max_S E_{X \sim P_d(X)} [\log S(X)] + E_{Z \sim P_Z} [\log (1 - S(G(Z)))], \quad (7)$$

where  $x$  means real samples sampled from  $P_d$ ,  $Z$  is extracted from a normal distribution  $P_Z$ , and  $G(Z)$  induces generator distribution for data augmentation.

**5.2. Multiple Distribution Generator.** When applying original GAN on particular or domain-specific usage, GAN is easy to trap into the mode collapsing situation, namely, GAN generates similar samples even with different modes of inputs. In fact, GAN adopts stochastic gradient-based learning to optimize  $S$  and  $G$  in turns. Once achieving



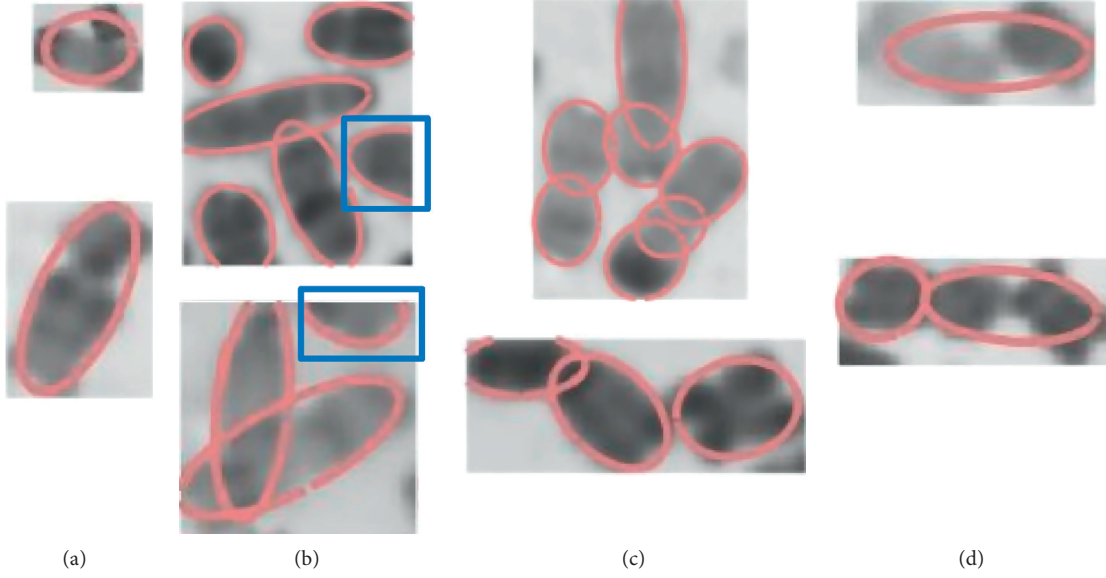


FIGURE 6: Results for chromosome overlapping segmentation: individual chromosome (a), successful segmentation of chromosome clusters (b), and failure cases requiring manual modification (c) and (d).

discrimination on generated data, GAN requires to reverse the optimization order, thus changing the minimax formula in equation (7) to a maximin one. During the reverse optimization process,  $G$  in GAN is forced to achieve mapping from each  $Z$  to  $X$  which is mostly possible to be regarded as real data, resulting in mode collapsing phenomenon. Such problem is more severe in particular or domain-specific applications, due to heavily unbalanced distributions of realistic data acquired from real life.

As using a single generator causes mode collapsing in the original GAN; Hoang et al. [42] propose to improve original GAN by designing multiple generators. However, usage of multiple generators brings problem of complexity in optimizing and huge increase in computation cost. To solve this problem, we propose to utilize multiple distributions instead of generators. Since Gaussian mixture model  $M$  is theoretically suitable to fit any complex distribution, we utilize it to build the proposed distribution generator  $D$ :

$$D_j(Z_j) = P_Z(Z_j) + M_j(Z_j), \quad j = 1, \dots, K, \quad (8)$$

where  $K$  and  $j$  represent the number and index corresponding to distribution generators, respectively,  $P_Z$  is normal distribution, and  $Z_j$  means vector whose values are random sample from the range between 0 and 1. Size of  $Z_j$  is determined on the basis of number of chromosome pictures. We thus define Gaussian mixture distribution  $M_j$  as

$$M_j(Z_j) = \sum_{k=1}^{n_j} \frac{1}{n_j} \cdot \phi(Z_j; \mu_{j,k}, \sigma_{j,k}), \quad (9)$$

where  $\phi$  is Gaussian distribution,  $n_j$  represents the number of distributions,  $\mu_{j,k}$  and  $\sigma_{j,k}$  are mean and variance corresponding to the  $k$ th Gaussian distribution, respectively. More distributions result in significant ability to generate samples. However, such setting brings a big increase in

computing consuming. In this case, it is particularly important to maintain a balance between generation diversity and computing. Through quantity of experiments,  $K$  and  $n_j$  are set as 8 to deal with different categories of chromosomes.

**5.3. GAN Structure Description.** The generator  $G(Z)$  takes multiple distribution  $D_k(Z)$  as input and computes a chromosome image as represented in Figure 7. The advanced network is built with four convolutional layers and a fully connected layer. A normalization layer and an ReLU activation function is designed after each convolutional layer. The first fully connected layer is in charge of reshaping input feature vector, and convolutional layers are designed to expand information based on trained parameters of filter kernels. After functioning of convolutional layers, normalization layers react to the expanded information over minibatch to stabilize the whole learning process and prevent the generator from collapsing.

The discriminator network is designed with a typical CNN architecture for classification task, which decides whether the input single chromosome image is an original or generated image. The proposed discriminator consists of four convolution layers, four pooling layers, and one fully connected layer. Batch normalization layer is used for stabilization as designs of generators. We adopt Leaky ReLU for activation function, which could prevent from vanishing gradient and speed up the training process. During training, the stochastic gradient descent method is used with the Adam optimizer, which achieves an adaptive moment estimation by involving first and second moments.

**5.4. fine-tune Process.** Data generated by MD-GAN is combined with the real data to fine-tune VGG-16 network for classification purpose, which is shown as blue in Figure 7.



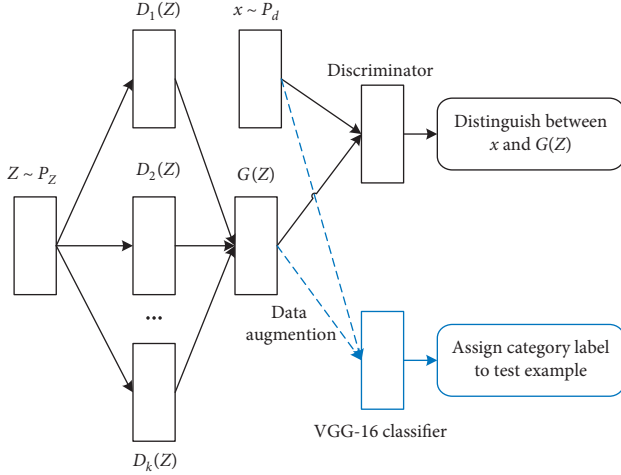


FIGURE 7: Structure of MD-GAN for data augmentation and VGG-16 network for classification, where  $Z \sim P_Z$  means the normal distribution,  $D_k(Z)$  refers to multiple distribution generators, and  $G(Z)$  and  $x \sim P_d$  represent generated training samples and original training samples, respectively.

The reason to fine-tune a pretrained network other than training from scratch lies in the fact that deep neural network generally requires large quantity of training samples to achieve distinguishability and generalization. However, we cannot get the minimum number of training samples, even with data augmentation technology. Therefore, we involve few-shot learning to fine-tune parameters and obtain better chromosome classification results.

Specifically, we retain parameters of early layers and modify parameters of higher level, which is proved by the fact that feature representation in early layer are general features to help prevent overfitting; meanwhile, features extracted in high level obtain more specific and semantic representation for chromosome classification through learning process.

## 6. Experimental Results

We demonstrate effectiveness of the proposed method for chromosome classification. Firstly, we introduce dataset and measurements. Secondly, we design a comparative experiment between MD-GAN and typical GAN to illustrate effectiveness of MD-GAN in data augmentation. Finally, two groups of comparative studies accompanying with sample images are conducted to show better performance of the proposed method than existing methods.

**6.1. Datasets and Measurement.** To prove the effectiveness of the proposed method, our co-operative hospital provides us with 120 cell and chromosome images including a total of 5,474 labeled samples. Chromosome images are randomly divided into two groups with 4600 and 874 images, which are utilized to perform training and testing. In fact, our obtained labeled data is not enough for a classification task of 24 classes with deep learning methods. However, achieving labeled data from doctors is high in time and money cost,

since labeling is an annoying and time-consuming task for doctors. This is the main reason for the usage of MD-GAN to generate more training samples as data augmentation. To compare chromosome classification results, we choose accuracy  $P^{24}$  for total classes of chromosome images as measurement. To clearly show classification results for a specific class, we define five  $P^2$ ,  $P^{10}$ ,  $P^{16}$ , and  $P^{22}$  to represent classification accuracy for the 2nd, 10th, 16th, and 22th pair of chromosomes.

**6.2. Data Augmentation Analysis.** We conduct two experiments in this subsection, where the former one is to perform comparisons between MD-GAN and typical GAN on capability of generating new data without collapsing, and the latter one is to indicate the performance of MD-GAN in generating new training samples for chromosome classification.

In order to verify that MD-GAN owns capability to learn complex distribution of real data space, we design a comparative experiment to generate samples of a Gaussian mixture distribution based on Denver grouping rule. After experiments, comparisons of distribution variances produced by typical GAN and MD-GAN are represented in Figure 8. Specifically, we can notice that GAN fails to learn data distribution after 35000 iterations to achieve convergence, due to occurrence of mode collapsing problem. Meanwhile, MD-GAN is able to learn complex Gaussian mixture distribution after performing iterative optimization of 70000 iterations. In Figure 8(f), we can observe that final results achieved by MD-GAN not only maintain key features, but also ensure diversity of generated data samples. However, MD-GAN generally requires more iterations to achieve convergence than typical GAN, since multiple input distribution largely increases complexity of MD-GAN, thus raising computation burden to a certain extent. Based on the above discussion, we can conclude that MD-GAN has higher capability than typical GAN in constructing complex distribution and preventing collapse problems.

In the second group of experiments, we firstly show comparisons between real and MD-GAN created chromosomes in Figure 9. We could observe the created samples not only are visually similar with real ones, but also own diverse patterns in appearance. Both features of generated chromosome images lead to improvement of classification accuracy with generated samples. In order to explore the relation between quality and number of generated chromosome images, we then conduct comparison experiments with different number of generated chromosome images. It is noted that we define number of generated chromosome images as  $n_i = 46 * n_p$ , where  $n_p$  refers to the number of persons and each person should be assigned with 46 chromosome images. Every two chromosome images for one person should be labeled by the same category from 23 classes, except for one pair of sex chromosomes. The reason to define  $n_i$  on the basis of  $n_p$  lies in the fact that we should keep class balance in generated chromosomes for better classification results.

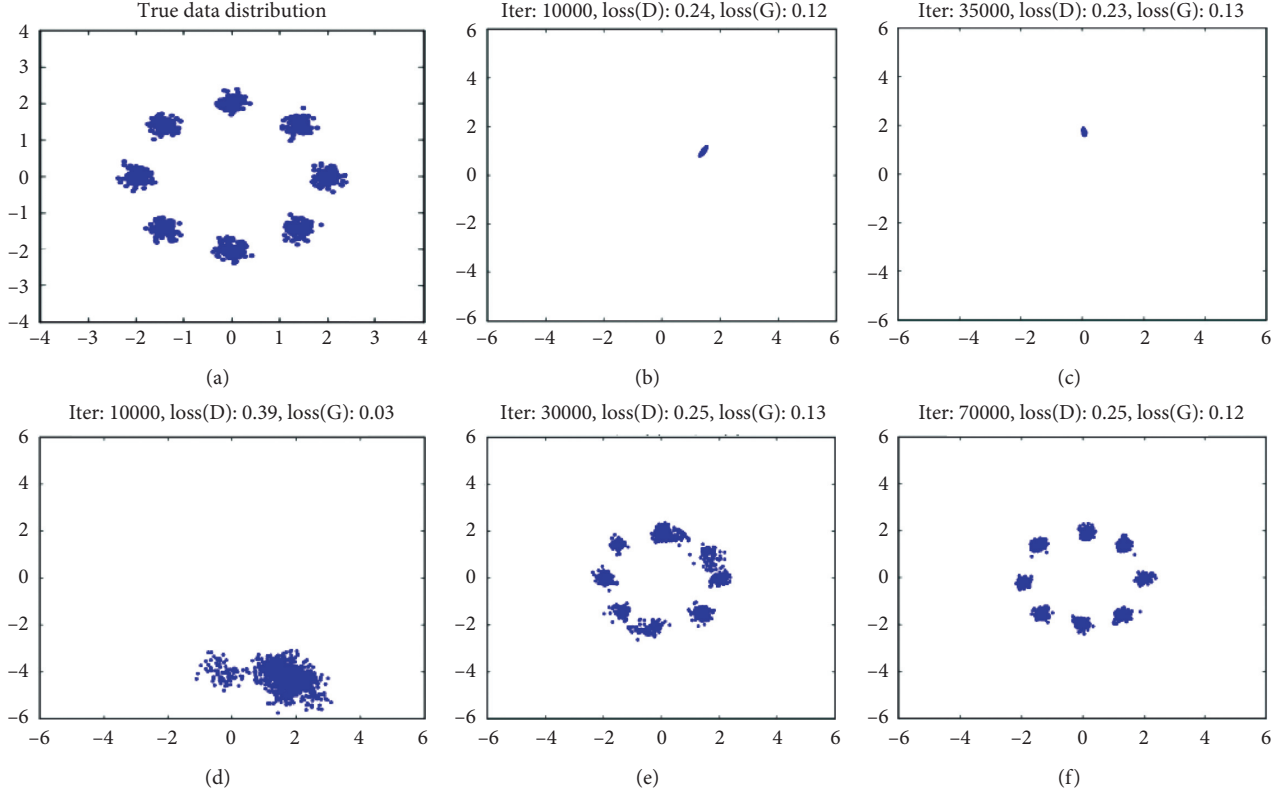


FIGURE 8: Comparisons of data distribution between a typical GAN and our proposed MD-GAN with different iterations: (a) the supposition of real data distribution, (b) and (c) results generated by typical GAN after 10000 and 35000 iterations, (d), (e), and (f) represent results achieved by MD-GAN after 10000, 35000, and 70000 iterations.

We show comparisons among generated chromosomes produced by MD-GAN with different  $n_p$  in Figure 10. We could observe that Figures 10(b) and 10(c) contain all diverse modes of chromosomes appeared in Figure 10(a), which proves that more modes could be generated by defining  $n_p$  with a larger value. However, Figure 10(c) contains several failure cases with fragments and noise points, which implies larger  $n_p$  would bring noise and artifacts for generated chromosome images, thus decreasing the ability of classification. Therefore, we need to keep balance on generated number  $n_p$  to produce chromosomes with more diverse modes and less artifacts. Main Reason to get failure cases lies in the fact that researchers often are short of measurement functions to justify how good a generated case is in vision appearance. To reduce the number of failure cases, our future work is to propose a novel perceptual loss function with doctors, which could define how similarity is between a generated and a true image in vision appearance.

**6.3. Chromosome Classification Analysis.** In this subsection, we firstly show performance on detection with ER algorithm. Then, we conduct two groups of comparative experiments to show the effectiveness of the proposed method, which compare classification ability of the proposed method with either different  $n_p$  or other comparative methods, respectively.

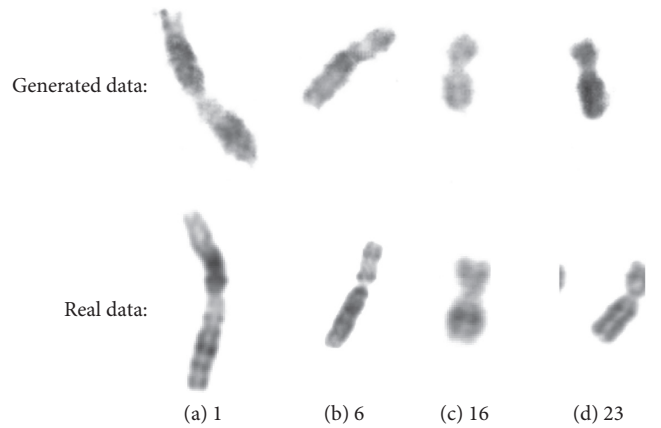


FIGURE 9: Comparison between real chromosomes and generated chromosomes achieved by MD-GAN, where we can notice generated chromosome images successfully imitate shape invariants of chromosomes. Note that numbers below samples are adopted to represent the corresponding class labels.

During experiments on detection, we compare accuracy of generating chromosome candidates by measurements of precision and recall. Specifically, we compare the proposed method with Otsu binarization method to show the effectiveness of the proposed method. Due to uneven staining nature of cell images and usage of a global threshold for binarization, Otsu method fails in some cases of uneven

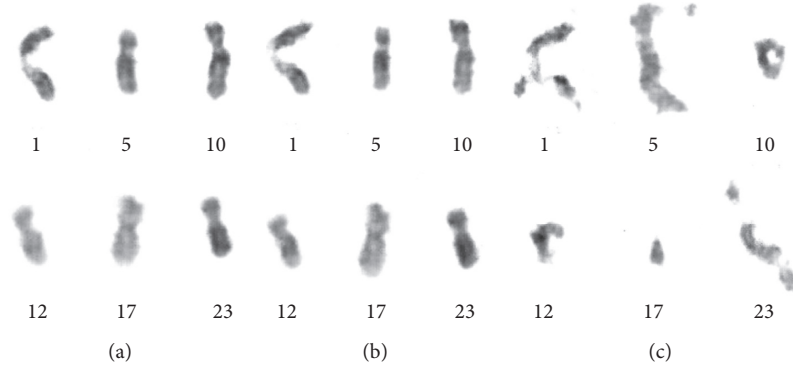


FIGURE 10: Comparisons among generated chromosome images achieved by MD-GAN with different number of generated chromosome images: (a) generated samples by setting  $n_p$  as 50 and (b) and (c) samples of good and poor quality by defining  $n_p$  as 250.

Giemsa staining, which appears with local intensity varieties and unclear shape boundaries during chromosome detection. In such case, Otsu method achieves 86.3% and 87.2% for detection accuracy and recall, respectively; meanwhile, detection accuracy and recall achieved by ER algorithm is 95.9% and 94.8%, which is high enough to guarantee for further classification process. For comparisons, we conduct experiments by utilizing ER with the first or second filter only, where we achieve accuracy and recall values as 89.6% and 95.2%, 88.2% and 95.6%, respectively. We can see filters greatly improve accuracy and decreases recall performance a little.

Following the idea of second experiment in data augmentation analysis, Table 1 offers the detailed statistics of classification results with different  $n_p$ , where CNNs are fine-tuned with samples of set formed by real chromosome images and  $46 * n_p$  ones, represented as CNN +  $n_p$ MG. The plot in Figure 11 compares accuracy performance achieved by the proposed method in terms of different  $n_p$  for data augmentation. From either Table 1 or Figure 11, we can notice a great accuracy decrease with more classes required to be classified, which can be proved by comparing among  $P^4$ ,  $P^{10}$ ,  $P^{16}$ ,  $P^{22}$ , and  $P^T$  for one method. This is due to multiple classes bring complexity for problem solving space, thus generally requiring more diverse and larger number of data to be adopted for training. By utilizing chromosome images generated by MD-GAN, a significant improvement in  $P^T$ , i.e., 4.6%, is achieved by models of CNN + 50MG. This is also true for other measurements for comparisons, which we can find improvement 1.2% in  $P^4$ , 12% in  $P^{18}$ , and 2.5% in  $P^{22}$ .

The increase in number of samples is not always beneficial to improve classification accuracy, which can be certificated by decrease of  $P^T$  by comparing between CNN + 50MG and CNN + 150MG from either Table 1 or from Figure 11. Such phenomenon of decrease can also be noticed on several specific classes of chromosomes. All these facts prove that larger setting of  $n_p$  will bring noise to classification, due to artifacts of chromosome images produced with larger  $n_p$ . This conclusion can also be proved by the second experiment of data augmentation analysis, which produce less visual desirable training samples with large  $n_p$ .

It is noted that we achieve inconsistent accuracy performance of  $P^4$  represented in Figure 11, which is caused by few testing samples with only one class of chromosome images.

From Figure 11, we could further conclude that setting  $n_p$  as 50 could maximally improve accuracy performance in our experiment. Therefore, we need to keep a balance on  $n_p$  for the purposes of increasing more sample modes and introducing less noise. It is noticed that  $n_p = 50$  is nearly half number of the original chromosome dataset, i.e., 119, which offers hints for researchers to perform data augmentation to improve classification accuracy.

In the second group of comparative study, we show the detailed statistics and the performed comparison between our CNN + 50MG and several comparative methods in Table 2. Specifically, we adopt CNN + 50MG as our proposed method to compare based on results of the former experiment. We implement CNN and Multi-Layer Perceptron (MLP) [49] with 2, 5 layers to be comparative studies. We implement MLP with different layers for comparing, since most traditional chromosome classification methods adopt MLP for classification like Lerner et al. [50], Ming and Tian [51], and so on. It is noted that we include two latest deep learning-based methods for comparisons, i.e., Sharma et al. [11] and Swati et al. [10], where the former method explores deep features for chromosome classification, and the latter one learns chromosome similarity via deep Siamese Networks to speed up classification with feedforward network classifier based on Multilayer Perceptron. We implement both deep learning-based methods by following their article. For the fairness of experiments, Sharma et al. [11] is implemented without pre-processing, i.e., straightening and bending. We follow Swati et al. [10] to modify an original version of deep Siamese Networks as a combination of Siamese Network and MLP.

From Table 2, we can notice deep neural networks including CNN, CNN + 50MG, Sharma et al. [11], and Swati et al. [10] achieve much higher accuracy than several traditional methods, including variant MLPs. These results prove the significant distinguish ability of deep neural network, especially for multiple class classification problem. Since Sharma et al. [11] without pre-processing is similar with original CNN in structure of neural networks, we could

**Data:** A small training set about preclassified chromosome images  $I_1, \dots, I_m$  with corresponding labels.

**Goal:** training a multiclass chromosome classifier with few labeled images.

**Algorithm Steps:**

**Step 1:** Pre-processing: Adopt the standard image manipulation augmentation techniques like rotation, translation, flipping, and so on to create more input images for the following module.

**Step 2:** GAN-based data augmentation: For each of the 24 classes of chromosomes, we use the corresponding training examples output by previous module to train a MD-GAN structure, which would help generate synthetic chromosome training samples to improve classification of such class. Therefore, we separately construct 24 MD-GAN models to accomplish tasks of data augmentation.

**Step 3:** fine-tune VGG-16 network: Use all the collected data, including original samples and samples created by pre-processing and MD-GAN, to fine-tune a pretrained VGG-16 classifier for accurate chromosome classification.

ALGORITHM 1: The proposed chromosome classification algorithm with GAN-based data augmentation technology.

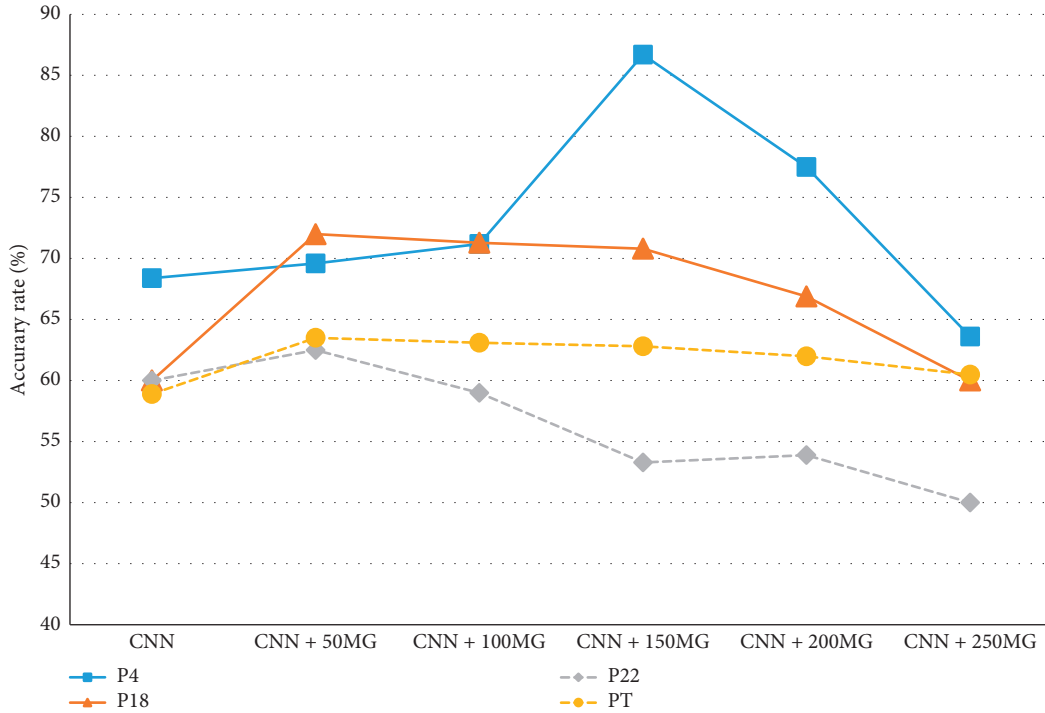


FIGURE 11: Comparison of accuracy rate achieved by the proposed method with different  $n_p$ .

TABLE 1: Accuracy of chromosome classification in dataset with different parameters, i.e.,  $n_p$ .

Method	$P^4$ (%)	$P^{18}$ (%)	$P^{22}$ (%)	$P^T$ (%)
CNN	68.4	60.0	60.0	58.9
CNN + 50MG	69.6	<b>72.0</b>	<b>62.5</b>	<b>63.5</b>
CNN + 100MG	75.2	71.3	59.0	63.1
CNN + 150MG	<b>86.7</b>	70.8	53.3	62.8
CNN + 200MG	77.5	66.9	53.7	62.0
CNN + 250MG	63.6	60.0	50.0	60.5

TABLE 2: Accuracy of chromosome classification on the collected dataset with several comparative methods.

Method	$P^4$ (%)	$P^{18}$ (%)	$P^{22}$ (%)	$P^T$ (%)
CNN	68.4	60.0	60.0	58.9
CNN + 50MG	69.6	<b>72.0</b>	62.5	<b>63.5</b>
Two layer MLP	58.3	54.2	52.9	51.3
Five layer MLP	62.1	55.3	53.9	53.1
Sharma et al. [11]	69.2	59.3	61.2	58.0
Swati et al. [10]	<b>70.8</b>	62.2	<b>63.2</b>	61.5

observe their similar performance on chromosome classification accuracy. Compared with Sharma et al. [11] and CNN, Swati et al. [10] improves classification accuracy by embedding more complex network architecture. It also achieves the highest classification accuracy value for the 4th and 22th chromosome. However, it still suffers from the

shortage of diverse modes of chromosomes, brought by the fact of small size of the collected dataset. The proposed method could improve chromosome classification accuracy with a proper generated number of chromosome images, which is proved by the best performance on recognizing the 18th chromosome and total chromosomes. The main



reason to achieve such improvement lies on the fact that we specially design MD-GAN structure to perform data enhancement, which brings features of pattern diversity and training stability to solve problem of small training data size.

## 7. Conclusion

We propose a chromosome karyotyping method to perform chromosome detection, segmentation, and classification automatically, which reduces complexity of medical data in both dimensions and volumes brought by IoMT environment. The proposed method consists of three stages, namely, *Chromosome Detection*, *Overlap Segmentation*, and *Category Classification*. In chromosome detection, we explore ER with geometry filters to obtain chromosome candidates. During overlap segmentation, we segment touching and overlapping chromosomes by utilizing geometry information of chromosomes. Finally during category classification, MD-GAN is proposed to generate more convinced training samples, which are further utilized to fine-tune VGG-16 network for chromosome classification. Experimental results not only show the efficiency of the proposed method, but also prove the improvement in accuracy by utilizing MG-GAN for training data augmentation. Based on cloud computing and other technologies [31, 52, 53], we will further develop MD-GAN for other medical applications under similar IoMT environment in the future, such as disease diagnosis and abnormal identification.

## Data Availability

The chromosome image data used to support the findings of this study were supplied by Yirui Wu under license and so cannot be made freely available. Requests for access to these data should be made to [Yirui Wu, wuyirui@hhu.edu.cn].

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work was supported by the National Key R&D Program of China under Grant 2018YFC0407901, Fundamental Research Funds for the Central Universities under Grant B200202177, Natural Science Foundation of China (Grant nos. 61702160, 61672273, and 6183200), and Science Foundation of Jiangsu under Grant BK20170892, Scientific Foundation of State Grid Corporation of China (Research on Ice-wind Disaster Feature Recognition and Prediction by Few-shot Machine Learning in Transmission Lines).

## References

- [1] Y. Wang, Z. Cai, Z.-H. Zhan, Y.-J. Gong, and X. Tong, "An optimization and auction-based incentive mechanism to maximize social welfare for mobile crowdsourcing," *IEEE Transactions on Computational Social Systems*, vol. 6, no. 3, pp. 414–429, 2019.
- [2] Y. Wang, Q. He, D. Ye, and Y. Yang, "Formulating criticality-based cost-effective fault tolerance strategies for multi-tenant service-based systems," *IEEE Transactions on Software Engineering*, vol. 44, no. 3, pp. 291–307, 2017.
- [3] P. Lai, Q. He, G. Cui et al., "Edge user allocation with dynamic quality of service," in *Proceedings of the International Conference on Service-Oriented Computing*, pp. 86–101, Springer, Toulouse, France, October 2019.
- [4] S. Zhang, X. Li, Z. Tan, T. Peng, and G. Wang, "A caching and spatial  $k$ -anonymity driven privacy enhancement scheme in continuous location-based services," *Future Generation Computer Systems*, vol. 94, pp. 40–50, 2019.
- [5] A. P. Britto and G. Ravindran, "A review of cytogenetics and its automation," *Journal of Medical Sciences (Faisalabad)*, vol. 7, no. 1, pp. 1–18, 2007.
- [6] X. Wang, B. Zheng, S. Li, J. J. Mulvihill, M. C. Wood, and H. Liu, "Automated classification of metaphase chromosomes: optimization of an adaptive computerized scheme," *Journal of Biomedical Informatics*, vol. 42, no. 1, pp. 22–31, 2009.
- [7] F. Abid and L. Hamami, "A survey of neural network based automated systems for human chromosome classification," *Artificial Intelligence Review*, vol. 49, no. 1, pp. 41–56, 2018.
- [8] M. J. Roshtkhari and S. K. Setarehdan, "A novel algorithm for straightening highly curved images of human chromosome," *Pattern Recognition Letters*, vol. 29, no. 9, pp. 1208–1217, 2008.
- [9] R. M. Nair, R. Remya, and K. Sabeena, "Karyotyping techniques of chromosomes: a survey," *International Journal of Computer Trends and Technology*, vol. 22, no. 1, 2015.
- [10] Swati, G. Gupta, M. Yadav, M. Sharma, and L. Vig, "Siamese networks for chromosome classification," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 72–81, IEEE, Venice, Italy, October 2017.
- [11] M. Sharma, O. Saha, A. Sriraman, R. Hebbalaguppe, L. Vig, and S. Karande, "Crowdsourcing for chromosome segmentation and deep classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 786–793, IEEE, Honolulu, HI, USA, July 2017.
- [12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 91–99, Montréal, Canada, 2015.
- [13] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
- [14] L. Ji, "Fully automatic chromosome segmentation," *Cytometry*, vol. 17, no. 3, pp. 196–208, 1994.
- [15] S. Minaee, M. Fotouhi, and B. Khalaj, "A geometric approach to fully automatic chromosome segmentation," in *Proceedings of the Signal Processing in Medicine and Biology Symposium*, pp. 1–6, IEEE, Philadelphia, PA, USA, December 2014.
- [16] Y. Wu, W. Wang, S. Palaiahnakote, and T. Lu, "A robust symmetry-based method for scene/video text detection through neural network," in *Proceedings of the IAPR International Conference on Document Analysis and Recognition*, pp. 1249–1254, IEEE, Kyoto, Japan, November 2017.
- [17] V. S. Balaji and S. Vidhya, "Separation of touching and overlapped human chromosome images," in *Advancements of Medical Electronics*, pp. 59–65, Springer, Berlin, Germany, 2015.
- [18] B. Lerner, H. Guterman, and I. Dinstein, "A classification-driven partially occluded object segmentation (CPOOS) method with application to chromosome analysis," *IEEE*

- Transactions on Signal Processing*, vol. 46, no. 10, pp. 2841–2847, 1998.
- [19] G. C. Charters and J. Graham, “Trainable grey-level models for disentangling overlapping chromosomes,” *Pattern Recognition*, vol. 32, no. 8, pp. 1335–1349, 1999.
  - [20] S. Zhang, G. Wang, M. Z. A. Bhuiyan, and Q. Liu, “A dual privacy preserving scheme in continuous location-based services,” *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 4191–4200, 2018.
  - [21] L. Qi, Q. He, F. Chen et al., “Finding all you need: web APIs recommendation in web of things through keywords search,” *IEEE Transactions on Computational Social Systems*, vol. 6, no. 5, pp. 1063–1072, 2019.
  - [22] S. Zhang, K.-K. R. Choo, Q. Liu, and G. Wang, “Enhancing privacy through uniform grid and caching in location-based services,” *Future Generation Computer Systems*, vol. 86, pp. 881–892, 2018.
  - [23] L. Qi, X. Zhang, S. Li, S. Wan, Y. Wen, and W. Gong, “Spatial-temporal data-driven service recommendation with privacy-preservation,” *Information Sciences*, vol. 515, pp. 91–102, 2020.
  - [24] X. Xu, Y. Chen, X. Zhang, Q. Liu, X. Liu, and L. Qi, “A blockchain-based computation offloading method for edge computing in 5G networks,” *Software: Practice and Experience*, 2019.
  - [25] J. Cho, S. Ryu, and S. Woo, “A study for the hierarchical artificial neural network model for giemsa-stained human chromosome classification,” in *Proceedings of IEEE Annual International Conference on Engineering in Medicine and Biology Society*, vol. 2, pp. 4588–4591, IEEE, San Francisco, CA, USA, September 2004.
  - [26] N. Madian and K. B. Jayanthi, “Analysis of human chromosome classification using centromere position,” *Measurement*, vol. 47, pp. 287–295, 2014.
  - [27] A. Khmelinskii, R. Ventura, and J. Sanches, “A novel metric for bone marrow cells chromosome pairing,” *IEEE Transactions on Biomedical Engineering*, vol. 57, no. 6, pp. 1420–1429, 2010.
  - [28] F. Abid and L. Hamami, “A survey of neural network based automated systems for human chromosome classification,” *Artificial Intelligence Review*, vol. 49, no. 1, pp. 41–56, 2018.
  - [29] Y. Qin, N. Song, H. Zheng et al., “Varifocal-net: a chromosome classification approach using deep convolutional networks,” 2018, <https://arxiv.org/abs/1810.05943>.
  - [30] X. Xu, X. Zhang, H. Gao, Y. Xue, L. Qi, and W. Dou, “Be-Come: blockchain-enabled computation offloading for IoT in mobile edge computing,” *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4187–4195, 2019.
  - [31] L. Qi, Y. Chen, Y. Yuan, S. Fu, X. Zhang, and X. Xu, “A QoS-aware virtual machine scheduling method for energy conservation in cloud-based cyber-physical systems,” *World Wide Web*, vol. 23, no. 2, pp. 1275–1297, 2020.
  - [32] X. Xu, Q. Liu, Y. Luo et al., “A computation offloading method over big data for IoT-enabled cloud-edge computing,” *Future Generation Computer Systems*, vol. 95, pp. 522–533, 2019.
  - [33] B. Zhao, Y. Wang, Y. Li, Y. Gao, and X. Tong, “Task allocation model based on worker friend relationship for mobile crowdsourcing,” *Sensors*, vol. 19, no. 4, p. 921, 2019.
  - [34] X. Xu, S. Fu, L. Qi et al., “An IoT-oriented data placement method with privacy preservation in cloud environment,” *Journal of Network and Computer Applications*, vol. 124, pp. 148–157, 2018.
  - [35] Z. Zheng, L. Zheng, and Y. Yang, “Unlabeled samples generated by GAN improve the person re-identification baseline in vitro,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3754–3762, IEEE, Venice, Italy, October 2017.
  - [36] X. Zhu, Y. Liu, and Z. Qin, “Data augmentation in classification using GAN,” 2017, <https://arxiv.org/abs/1711.00648>.
  - [37] C. Bowles, L. Chen, R. Guerrero et al., “GAN augmentation: augmenting training data using generative adversarial networks,” 2018, <https://arxiv.org/abs/1810.10863>.
  - [38] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, and H. Greenspan, “GAN-based synthetic medical image augmentation for increased CNN performance in liver lesion classification,” *Neurocomputing*, vol. 321, pp. 321–331, 2018.
  - [39] T. Salimans, I. J. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, “Improved techniques for training gans,” in *Proceedings of Neural Information Processing Systems*, pp. 2226–2234, Barcelona, Spain, 2016.
  - [40] A. J. Ratner, H. R. Ehrenberg, Z. Hussain, J. Dunnmon, and C. Ré, “Learning to compose domain-specific transformations for data augmentation,” in *Proceedings of Advances in Neural Information Processing Systems*, pp. 3236–3246, San Diego, CA, USA, 2017.
  - [41] J. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *Proceedings of IEEE International Conference on Computer Vision*, pp. 2242–2251, IEEE, Venice, Italy, October 2017.
  - [42] Q. Hoang, T. D. Nguyen, T. Le, and D. Phung, “Multi-generator generative adversarial nets,” 2017, <https://arxiv.org/abs/1708.02556>.
  - [43] Q. He, R. Zhou, X. Zhang et al., “Keyword search for building service-based systems,” *IEEE Transactions on Software Engineering*, vol. 43, no. 7, pp. 658–674, 2016.
  - [44] W. Gong, L. Qi, and Y. Xu, “Privacy-aware multidimensional mobile service quality prediction and recommendation in distributed fog environment,” *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 3075849, 8 pages, 2018.
  - [45] X. Xu, R. Mo, F. Dai, W. Lin, S. Wan, and W. Dou, “Dynamic resource provisioning with fault tolerance for data-intensive meteorological workflows in cloud,” *IEEE Transactions on Industrial Informatics*, p. 1, 2019.
  - [46] J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust wide-baseline stereo from maximally stable extremal regions,” *Image and Vision Computing*, vol. 22, no. 10, pp. 761–767, 2004.
  - [47] S. Zafari, T. Eerola, J. Sampo, H. Kalviainen, and H. Haario, “Segmentation of overlapping elliptical objects in silhouette images,” *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5942–5952, 2015.
  - [48] Y. Wu, P. Shivakumara, W. Wei, T. Lu, and U. Pal, “A new ring radius transform-based thinning method for multi-oriented video characters,” *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 18, no. 2, pp. 137–151, 2015.
  - [49] R. Collobert and S. Bengio, “Links between perceptrons, MLPs and SVMs,” in *Proceedings of International Conference on Machine Learning*, Shanghai, China, August 2004.
  - [50] B. Lerner, H. Guterman, I. Dinstein, and Y. Romem, “Medial axis transform-based features and a neural network for human chromosome classification,” *Pattern Recognition*, vol. 28, no. 11, pp. 1673–1683, 1995.
  - [51] D. Ming and J. Tian, “Automatic pattern extraction and classification for chromosome images,” *Journal of Infrared, Millimeter, and Terahertz Waves*, vol. 31, no. 7, pp. 866–877, 2010.

- [52] Y. Wang, Z. Cai, X. Tong, Y. Gao, and G. Yin, "Truthful incentive mechanism with location privacy-preserving for mobile crowdsourcing systems," *Computer Networks*, vol. 135, pp. 32–43, 2018.
- [53] X. Xu, Y. Li, T. Huang et al., "An energy-aware computation offloading method for smart edge computing in wireless metropolitan area networks," *Journal of Network and Computer Applications*, vol. 133, pp. 75–85, 2019.