Crowdsourcing for Chromosome Segmentation and Deep Classification

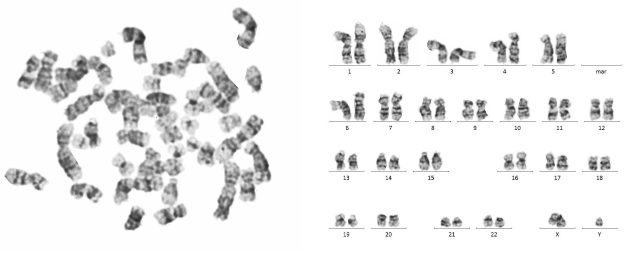
Monika Sharma[a⋆][*], Oindrila Saha[b⋆], Anand Sriraman[a],
Ramya Hebbalaguppe[a], Lovekesh Vig[a], Shirish Karande[a]
[a]TCS Research, New Delhi, India, Email : monika.sharma1@tcs.com
[b]Indian Institute of Technology, Kharagpur, India, Email : oindrila13saha@gmail.com

Abstract

Metaphase chromosome analysis is one of the primary techniques utilized in cytogenetics. Observations of chromosomal segments or translocations during metaphase can indicate structural changes in the cell genome, and is often used for diagnostic purposes. Karyotyping of the chromosomes micro-photographed under metaphase is done by characterizing the individual chromosomes in cell spread images. Currently, considerable effort and time is spent to manually segment out chromosomes from cell images, and classifying the segmented chromosomes into one of the 24 types, or for diseased cells to one of the known translocated types. Segmenting out the chromosomes in such images can be especially laborious and is often done manually, if there are overlapping chromosomes in the image which are not easily separable by image processing techniques. Many techniques have been proposed to automate the segmentation and classification of chromosomes from spread images with reasonable accuracy, but given the criticality of the domain, a human in the loop is often still required. In this paper, we present a method to segment out and classify chromosomes for healthy patients using a combination of crowdsourcing, preprocessing and deep learning, wherein the non-expert crowd from CrowdFlower is utilized to segment out the chromosomes from the cell image, which are then straightened and fed into a (hierarchical) deep neural network for classification. Experiments are performed on 400 real healthy patient images obtained from a hospital. Results are encouraging and promise to significantly reduce the cognitive burden of segmenting and karyotyping chromosomes.

1. Introduction

Chromosomes are elongated rope like structures in the cell nucleus that contain the human body's genetic code. The human body has 23 pairs of chromosomes. Chromosomal analysis or karyotyping is a useful technique to detect genetic abnormalities like Down syndrome, Edwards syndrome, Chronic myelogenous leukemia, and Turner syndrome. These abnormalities can manifest in the form of known chromosomal translocations and segments that correspond to different disorders. Karyotyping is performed by culturing cells and during metaphase separating the chromosomes from the nucleus of the cells and staining them on a slide to allow for micro-photography. Figure 1(a) shows a sample chromosome slide. Finally, the chromosome images are analyzed by experts to classify and segregate the different chromosome segments.



(a) **(b)**

Figure 1. (a) A metaphase chromosome image and (b) corresponding karyotype image.

Despite the diagnostic importance of karyotyping chromosomes, considerable manual time and effort is required for segmenting out and classifying the chromosomes in images from a cell culture. In this paper, we propose a pipeline for automatic segmentation and classification of chromosomes that combines the use of non-expert crowd for annotating the chromosome segments and a deep classification model for categorizing the individual chromosomes. The slides are fed to a crowdsourcing platform CrowdFlower [11] to annotate the chromosome boundaries, which are then extracted and fed into the classification engine for karyotyping. We perform straightening of bent chromosomes before feeding the chromosome images to a deep neural network for classification because it improves the classification accuracy as described in Section 5. The main

[*]Note: ⋆ indicates equal contribution

contributions of the paper are as follows :

- The use of crowdsourcing for generation of chromosome segments as opposed to clinicians manually segmenting and annotating the chromosome images during karyotyping. We also address the challenges of spurious or spam markings as well as maximize coverage of segmentation labels as discussed in Section 4.1.2.
- The use of deep learning for classification of chromosome images. To the best of our knowledge, deep learning hasn't been explored in the realm of chromosome classification. We perform some pre-processing of chromosome segments like straightening of bent chromosomes and chromosome-length normalization before feeding the images to the deep convolutional network (CNN) for classification. The proposed CNN architecture is shown in Figure 7.

The objective is not to replace the domain expert but to significantly reduce the cognitive load involved during the segmentation and classification task, and to allow the expert to correct for any errors made by the system.

The rest of the paper is organized as follows : Section 2 discusses prior attempts at automated karyotyping of chromosomes. Section 3 provides the details of the dataset used and Section 4.1 explains our proposed methodology of using a crowdsourcing platform for chromosome segmentation. We discuss the straightening of bent chromosomes, pre-processing of chromosome segments followed by classification of chromosomes in Section 4.4. Subsequently, we present details of experiments and results in Section 5. Conclusions and avenues for future work are outlined in Section 6.

2. Related Work

The most challenging problems in karyotyping are the segmentation and classification of overlapping chromosomes in metaphase spread images, and numerous attempts have been made in the literature to automate overlapping chromosome segmentation with limited success. This could be because of situations, such as, *unsplit clusters* which could be another main contributor of false positives and *chromosome fragmentation* that could increase the false negative rate as broken chromosomes cannot be used for further analysis. A rule based approach for segmentation and classification was proposed in [8], where the rule parameters would adapt to each cell. This method would fail when (a) chromosome arms are widely separated and (b) the chromosomes are very pale. To overcome this, Charters and Graham [4] proposed a collection of sub chromosomal banding profile templates in order to train models to recognize the chromosome segments to first oversegment, and subsequently combined mini-segments in a bottom up fashion. Agam and Dinstein [1] utilized minsets to separate out overlapping grayscale chromosomes using shape based hypothesis testing. Lerner used a neural network to identify the relevant cut points that indicate separation after automatic binarization using principal component analysis for feature selection [10]. Another work has looked at finding the concave points on the image contour and constructing heuristics for chromosome separation [12]. A geometric based approach to fully automatic chromosome segmentation [13] was proposed which separates individual chromosomes from clusters one at a time. This method utilizes skeletonization, ellipse fitting and utilizes the convex hull area to identify overlapping chromosomes. We observed that the aforementioned work did not generalize across the datasets and our custom dataset in particular. Hence, deep learning based approaches which have proven transfer learning capabilities are explored in our paper. Recently, deep learning based approaches have been suggested for segmentation in a Kaggle competition but have only been applied for the pairwise chromosome separation problem [7], not for clusters. We try to address the problem via proposing a pipeline that is a combination of crowdsourcing with deep learning to tackle chromosome segmentation and classification.

Manual effort for ground truth creation via segmentations of microscopic images is an essential step for biomedical analysis. The tediousness and time consuming nature of the task makes it difficult to scale the availability of substantial ground truth for the training phase. Similarly, it becomes difficult to deploy at scale, human-machine workflows, where segmentation by humans can be fed to machine classifiers. A. Sorokin and D. Forsyth were the first to explore the use of crowds for image segmentation [18]. Several subsequent contributions have emerged (see [9] for a comprehensive review). However, a large percentage of our work has focused on obtaining a tight fitted bounding box as an approximation to exact segmentation can prove tedious for a crowd. Consequently, the HCI community has shown interest in modeling the crowd behaviour in segmentation [14] and also in reducing the segmentation task to clicks on an object [3]. However, these techniques and observations do not readily extend to tasks where several objects have to be segmented in a single image which is a common requirement for microscopic images. Gold standard questions or qualification questions are often employed for controlling the quality of crowd work, and [5] utilizes such inbuilt functionalities from CrowdFlower for nucleus segmentation. However, these approaches are susceptible to changes in the behaviour of a worker. The work by Sameki et. al. [15] does consider worker behaviour as well as image features to identify spam, however, they also consider microscopic images only with an individual object. Finally,

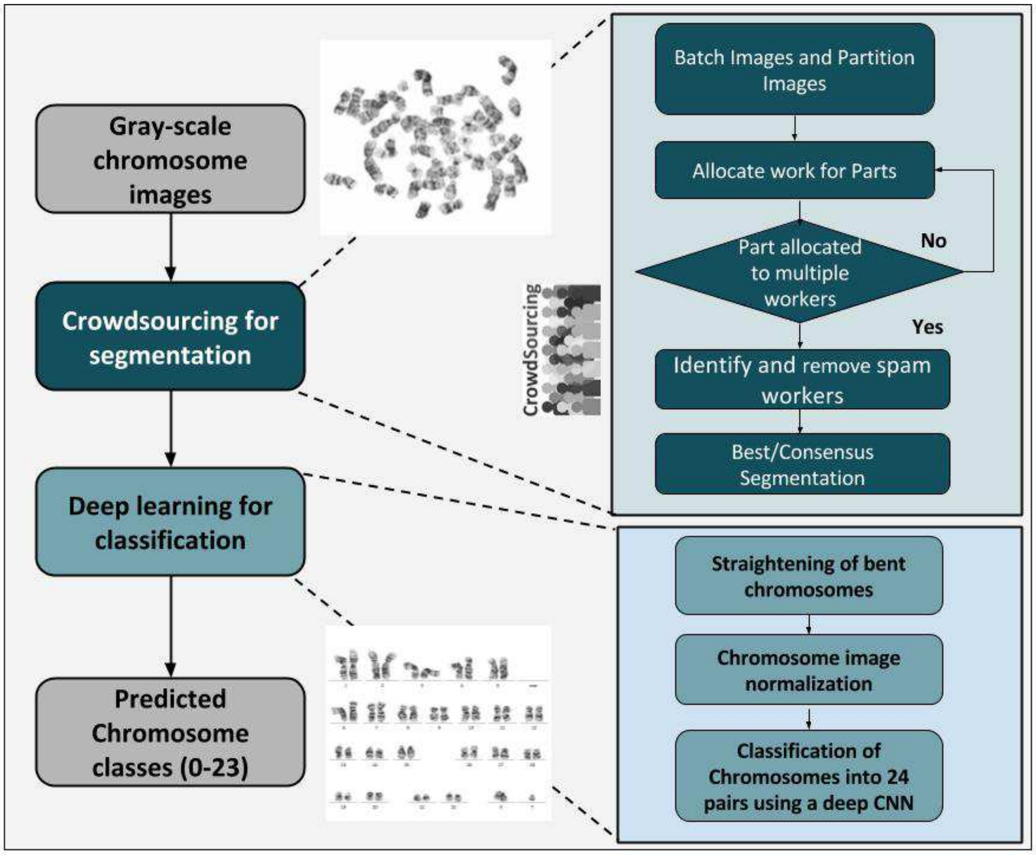

Gray-scale chromosome images
Crowdsourcing for segmentation
Deep learning for classification
Predicted Chromosome classes (0-23)
CrowdSourcing
Batch Images and Partition Images
Allocate work for Parts
Part allocated to multiple workers
No
Yes
Identify and remove spam workers
Best/Consensus Segmentation
Straightening of bent chromosomes
Chromosome image normalization
Classification of Chromosomes into 24 pairs using a deep CNN

Figure 2. Proposed pipeline for Automatic Karyotyping of Chromosome Images. Initially, the gray-scale chromosome image is passed to the non-expert crowds from CrowdFlower for segmentation of chromosomes. The challenges like, wrong or spam markings as well as maximize coverage of segmentation labels at boundary regions of chromosomes, are also addressed in the crowdsourcing module 4.1.2. The segmented chromosomes are then sent to the classification module where the bent chromosomes are straightened, length-normalized and passed through a trained deep CNN network as shown in Figure 7. We get chromosome class labels (0-23) as the output of this network.

Schlesinger et. al. [17] present an approach called *iaSTAPLE* for segmenting the epithelial cells of a fly wing. However, this work requires image features to evaluate the quality of crowd work. Such an approach may not be feasible at initial stage of learning the segmentation and classification. The key challenges when working with a crowd is to identify spurious or spam markings, as well as maximize coverage. In this work, we consider the use of non expert crowds from CrowdFlower for segmentation of chromosomes. We propose steps towards addressing both of the aforesaid challenges.

3. Dataset

The dataset comprised of 400 stained images with varying degrees of overlap between chromosomes, out of which 200 were kept for testing and the remaining for training and validation. The sample stained images are shown in Figure 1(a).

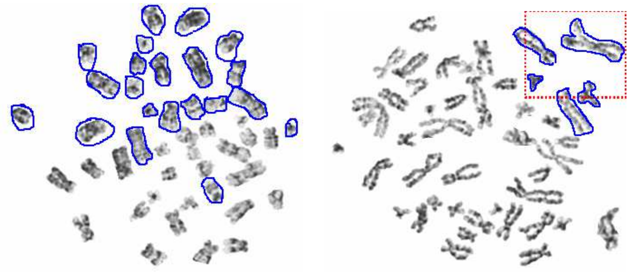
4. Proposed Methodology

The proposed pipeline for automatic karyotyping of chromosomes mainly consists of two major modules, namely, Chromosome Segmentation via crowdsourcing and Chromosome Classification using deep CNN. These modules are described in detail as follows :

4.1. Chromosome Segmentation via the crowd

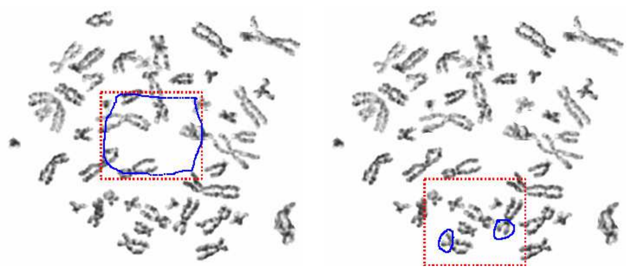
4.1.1 System and Workflow

We recruited workers from CrowdFlower to segment the chromosomes in a given image. We provided them an interface where they used their mouse to draw an outline around each chromosome present in the image. Workers were asked to draw these outlines around all the chromosomes that they identified. We initially considered asking a single worker to mark all chromosomes (as shown in Figure 3(a)) in an image, while creating redundancy by allocating the same image to multiple workers. However, this



(a) **(b)**

Figure 3. Examples of crowd marking in Phase 1 and Phase 2.



(a) **(b)**

Figure 4. Examples of erroneous crowd responses.

led to two drawbacks. (1) *Poor coverage* as several workers would feel fatigued and would drop off without completing the microtask. On average, a worker would only mark $\approx$ 20 chromosomes. (2) *Poor mixing* as each worker would be cross-evaluated only by another 4 workers allocated the same image. In order to circumvent this problem, we ask the workers to mark chromosomes that intersect or lie completely within an area highlighted by a dotted red rectangle as shown in Figure 3(b). We observed significantly improved coverage with this change. However, several workers would cut a marking off at the border of the red-dotted line. We were able to reduce this effect with very explicit instructions that provided the crowd workers with screen shots and examples describing chromosome marking across the boundaries.

We anticipated and observed two types of spammers: i) workers that were marking a large outline covering all the chromosomes in their grid and ii) workers not marking / partially marking chromosomes (See Figure 4). In addition, some of the workers would fuse the marking for overlaid chromosomes. In the subsequent section, we provide filtering procedure to guard against the above described spam responses and selection of segmentation labels.

4.1.2 Modeling and Algorithms

We have m crowd workers and n images. Let the I_i represent the i^{th} image. Each image can be further partitioned into t parts, with I_{ij} representing the j^{th} part of the i^{th} image. Let S_{ij} be the set of workers who could provide segmentation for I_{ij}. Let H_k be a set of tuples (i, j) representing the parts that worker k had been assigned. Further, let c_{ijk} be the number of segments marked by the k^{th} worker for I_{ij}.

- **Spammer Identification**

  We employ the following filtering procedure to remove spammers. Let C_{ij} be the mode of c_{ijk} calculated over set S_{ij}. If all workers disagree on the count, then we declare C_{ij} to be equal to median, and in case of a tie, we choose the higher value. Further, a worker's reliability is measured by

  $$a_k = \Sigma_{(i,j)\in H_k} \mathbf{1}(\mid C_{ijk} - C_{ij} \mid \leq \tau) \qquad (1)$$

  which represents the number of times a worker is in close agreement with the mode. We remove all workers with reliability below a threshold. This filtering mechanism removes the most obvious spammers who tend to mark segments with little correlation to the true chromosomes.

  In the second phase of filtering, we shall remove the spammers who could possibly have a slight adversarial attitude, or have misunderstood instructions and provide consistently poor segmentations. Let O_{ijkl} be the l^{th} segment marked by worker k on I_{ij}. We define a score $T(O_{ijkl})$ in terms of the best match provided by some other worker:

  $$T(O_{ijkl}) = \max_{\substack{c \neq k \\ \forall b,c,d}} \frac{Area(O_{ijkl} \cap O_{ibcd})}{Area(O_{ijkl} \cup O_{ibcd})} \qquad (2)$$

  Thus, the quality of a worker can be described by the expected quality of his marking, $q(k) = \mathbb{E}[T(O_{ijkl})]$. We remove all workers with quality below a chosen threshold.

- **Consensus Segmentation**

  Once the quality workers are identified, we proceed towards identifying the best segmentation labels. We select chromosomes in a greedy fashion on the basis of the score $T(.)$. Whenever a segmentation label is selected, we remove labels from all other workers with a significant overlap with the selected marking. This process is repeated till there are no more segmentation labels left to be selected.

4.2. Chromosome Classification using deep CNN

After the individual chromosomes have been segmented out from the image, they are fed to a classifier to determine the type of chromosome. This involves some preprocessing steps to improve classification as follows :

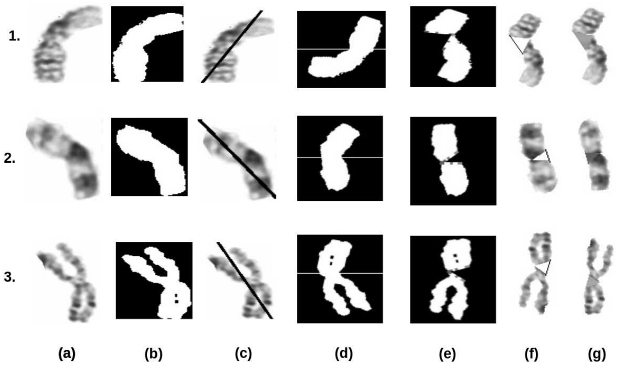


Figure 5. Chromosome straightening process showing (a) original image, (b) binary image, (c) orientation of bending, (d) bending-axis, (e) image after stitching operation, (f) line drawn to fill the empty area after straightening and (g) final straightened image.

4.2.1 Straightening of Chromosomes

One of the main challenges in the automatic classification of chromosome images obtained from a light microscope is that often chromosomes are bent in different orientations. As the point and extent of bending varies diversely for different chromosomes, the problem of classification becomes more complex. Therefore, we use an automatic straightening method [6] to straighten the bent chromosomes. The method presented in [6] is effective for straightening highly curved chromosomes but does not perform well for slightly less curved chromosomes. We added some modifications to make the former chromosome straightening algorithm more widely applicable, as described below:

4.2.2 Find Bending Orientation

After binarization of the chromosome image as mentioned in [6], we determine the bending orientation of the chromosome, i.e., whether a particular chromosome is straight or bent. This is done based on the fact that an upright tight fitting rectangle for a straight chromosome contains less blank area as compared to the area for bent chromosomes. Therefore, we define a *whiteness value* 'W' as ratio of the sum of pixel values of a binarized chromosome image (which represents the total number of white pixels as all black pixels are of value = 0) and total area of the tight fitting rectangle. The chromosomes with $W \geq W_T$ are labeled as straight chromosomes, where W_T is the whiteness threshold whose value is determined empirically to be 170 for our dataset. Further, we find the direction of bending of curved chromosomes. The direction of bending is needed for a later step to determine the final orientation of the straightened chromosomes. We fit a line to the binarized chromosome as shown in Figure 6 and use the sign of slope of this line to find out the direction of bending. The chromosomes bent towards the right would have a positive slope for the fitted line.

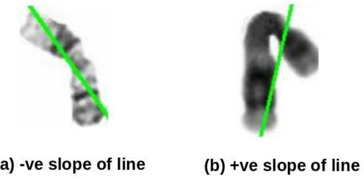


Figure 6. Figure showing chromosomes (a) bent towards the left having negative slope and (b) bent towards the right having positive slope of fitted line, respectively.

4.2.3 Find Bending Centre of Curved Chromosomes

We used the same method as proposed in [6] to locate the bending centre of curved chromosomes. Prior to locating the maxima and minima of the horizontal projection vector, we smoothened out the distribution curve of horizontal projection vectors by applying a *Savitzky Golay* filter [16] to ignore small deflections which may contribute to unwanted local maxima or minima. As a result of this step, we split the chromosome into two sub-images containing one arm each along the bending axis as shown in Figure 5(d), which is where the chromosome is thinnest.

4.2.4 Chromosome Arms Stitching and Reconstruction

Each sub-image contains one arm of the chromosome which is approximately a straight object. The two sub-images must now be rotated so that the two arms are in the same direction. For this purpose, each sub-image is rotated from $-90°$to $90°$ while its vertical projection vector is calculated at each rotation step. Due to the particular shape of each arm of the chromosome, the vertical projection vector will demonstrate minimum width if the arms are in the vertical position inside the sub-image. In a similar manner, the upper arm is rotated so as to be in the vertical position. These same transformations are applied to the real gray scale image rather than its binary version.

The stitching of the two arms is done by cropping out the lower black part of aligned upper arm and upper black part for aligned lower arm and shifting upper image horizontally allowing the upper part of the chromosome to lie correctly on the lower part. The shifting is done such that the lowest white pixel of the upper image is just on top of topmost white pixel of lower image as shown in Figure 5(e).

As we can see from Figure 5(e) after stitching of the chromosome arms, some pixels of chromosome image are lost. We perform reconstruction to address this issue. In this process, the two outer end points (unjoined) of the empty part of the stitched chromosome are found and joined using

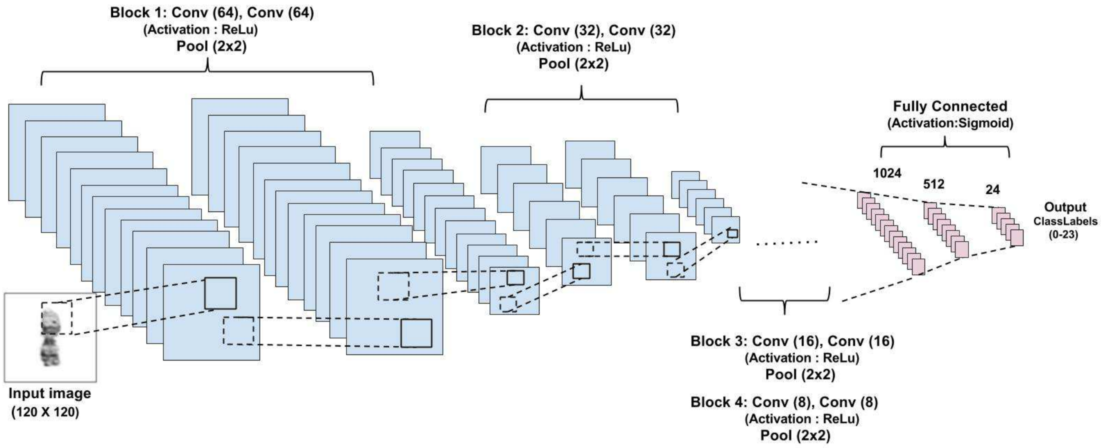


Figure 7. CNN architecture used for classification of chromosomes.

a single straight line as can be seen in Figure 5(f). The pixels in the area enclosed are then filled with the mean value of the pixels at the same horizontal level as the empty pixel as shown in Figure 5(g). This is done as chromosomes have horizontal bands. Thus, the shade of pixels at the same horizontal level of the straightened chromosome should be the same.

4.3. Chromosome Length Normalization

The chromosome segment-images are of varying sizes as a result of segmentation via crowdsourcing. As observed from the karyotype image shown in Figure 1(b), the most distinct features of different chromosomes are the length of chromosomes and the centromere position. To preserve this distinguishing feature, we perform length normalization of chromosome segment images using centromere position, which is described as follows :

4.3.1 Centromere Position of Chromosome

The chromosome centromere is the thinnest part of the chromosome. For straight chromosomes, the centromere is located by finding out the row number where the sum of row pixels is the lowest, i.e., it has the least number of white pixels or width. In case of curved chromosomes, the bending centre is the centromere position which is located as discussed in Section 4.2.3.

4.3.2 Length of Chromosome

When a chromosome bends, the surface towards which it is bent contracts in length and the outer surface expands. It is assumed that the length of the medial axis of the object stays the same length after bending. After straightening the chromosome, we calculate the true length of the chromosome by adding together the distance from the centre of the upper cut line to the upper edge and the distance from the centre of the lower cut line to the lower edge. We normalize the true length across each karyotype image of 23 pairs of chromosomes to a value between 0 and 100.

4.4. Chromosome Classification

In humans, each cell normally contains 23 pairs of chromosomes, for a total of 46. Twenty-two of these pairs, called autosomes, are identical in both males and females. The 23rd pair, the sex chromosomes X and Y, differ in males and females. Therefore, we classify the chromosomes into 24 classes using a deep convolutional network. The convolutional network, as shown in Figure 7, consists of four blocks where each block contains two convolutional layers with Relu activation, one dropout and one maxpooling layer. These blocks are followed by two fully connected layers with sigmoid activation and a softmax layer of 24 units at the end. The number of filters in the four blocks are different. This network was trained using categorical crossentropy loss and Stochastic Gradient Descent optimization.

5. Experimental Results

We evaluate the efficacy of our crowd filtering recommendation by considering a control set of 50 images each of which was known to contain 46 chromosomes. However, this fact was not revealed to the crowd workers. Our crowd task was created by dividing each image into (3x3) 9 parts, and asking each worker to provide segmentations for 10 parts chosen from 10 different images. We paid each worker 50 cents for completing the task, and the task re-

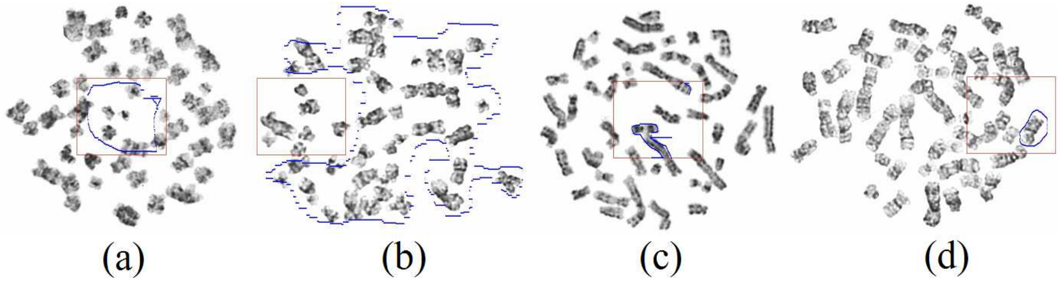


Figure 8. Examples of responses removed during filtering steps. The red rectangle outlines the grid assigned to the worker. The different types of spam identified are (a) one large marking within the grid, (b) large markings outside the grid, (c) incomplete markings and (d) marking only one chromosome per grid.

ceived an average satisfaction score of 4.3/5 from the workers. A handful of workers left the job without completion and thus, we had a total of 230 workers making a contribution. We employed a threshold of $\tau = 2$ and $k \geq 3$ as a first step for filtering. This removed the contributions of 32 workers. Further, we evaluated the mean T(.) score for the remaining workers and used a threshold of 0.4 which removed an additional 91 workers. Figure 8 shows responses by workers who were removed during this stage.

Subsequently, we employed our consensus step, however, we observed that a few spurious markings with very low score of T(.) were not getting eliminated. Hence, a threhold of 0.1 was employed on T(.) to allow for a segmentation label to be selected in the final recommendation. We observed after these steps that we were able to identify on average 35.9 chromosomes per image. Figure 9 shows some of the example segmentations obtained post the filtering and consensus steps.

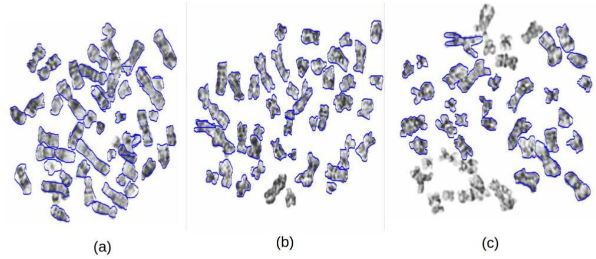


Figure 9. Sample annotations from the crowd.

As mentioned earlier in Section 3 that the complete experimental dataset consists of 400 complete greyscale images, out of which 200 were used for testing and the remaining for classifier training and validation. We manually annotated 1800 individual chromosome images with their chromosome types, while maintaining class balance. We used 1600 of these images (derived from the 200 full images in the training set) for training and validation sets for training a deep CNN classifier. We tested the trained classifier on remaining 200 chromosome images (from the 200 full images in the test set). Without straightening and pre-processing, the average classification accuracy obtained was 68.5%. However, with preprocessing, the classification accuracy improved to 86.7%. These results are very likely to improve with more annotated training data for classification. Moreover, we provide an interface to doctors for correcting any errors during crowdsourced segmentation and automated classification as shown in Figure 10. The doctor can select a particular chromosome marking from the left to focus on its corresponding classification on the right. If the expert finds any errors in either the segmentation or the classification, they can modify these and save the corrected response in the system.

6. Conclusion and Future Work

One of the objectives of this excercise was to examine the hypothesis that the crowd can generate sufficient training data for future segmentation to be performed via a deep segmentation model like Segnet [2]. Even acknowledging that 200 training images are unlikely to be sufficient for this purpose, we went ahead and trained a Segnet variant to segment out the chromosomes based on the annotations from the crowd. The results are encouraging, with Segnet obtaining a classification accuracy of over 80%, suggesting that given sufficient training data, Segnet could potentially learn to disambiguate the different chromosomes. To further validate this approach, we downloaded the kaggle dataset [7] of 13434 annotated images of overlapping chromosome pairs and attempted to train Segnet to distinguish between background pixels, overlapping chromosome regions and non-overlapping chromosome regions. We used 10000 images to train, 3000 to validate and 434 images to test our model. Segnet does well on this task with an accuracy exceeding 97% on the test set for all three classes. However, the same model when applied to model clusters with more than two overlapping chromosomes, fares poorly. This is not surprising given that the datasets are quite different. Despite this, there appears to be significant incentive to building deep

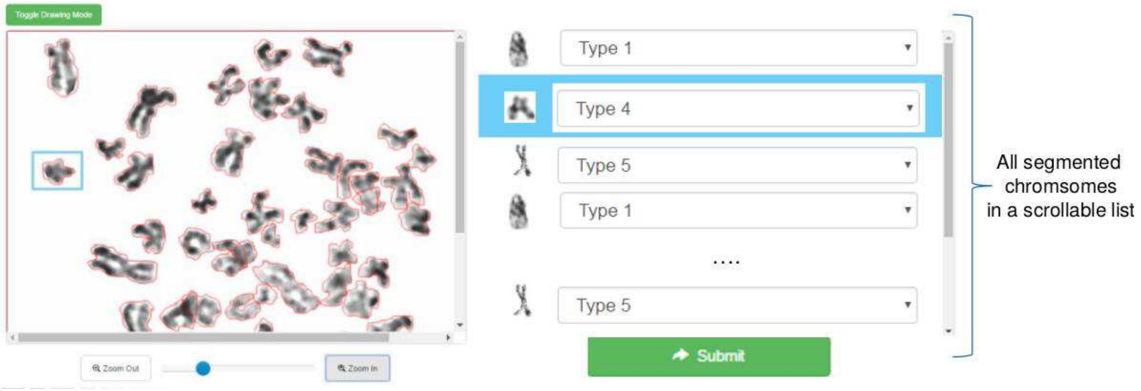


Figure 10. Interface provided to doctors for correcting any errors made during crowd-sourced segmentation and deep classification

models for segmentation as they manage to find more natural overlapping regions than traditional image processing methods. Future work involves building a deep segmentation engine to separate the chromosomes more efficiently even with chromosomal translocations and do active learning to minimize dependence on the crowd for segmentation.

References

[1] G. Agam and I. Dinstein. Geometric separation of partially overlapping nonrigid objects applied to automatic chromosome classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11):1212–1222, 1997. 2

[2] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *CoRR*, abs/1511.00561, 2015. 7

[3] A. Carlier, V. Charvillat, A. Salvador, X. Giro-i Nieto, and O. Marques. Click'n'cut: Crowdsourced interactive segmentation with object candidates. In *Proceedings of the 2014 International ACM Workshop on Crowdsourcing for Multimedia*, pages 53–56. ACM, 2014. 2

[4] G. C. Charters and J. Graham. Trainable grey-level models for disentangling overlapping chromosomes. *Pattern Recognition*, 32(8):1335–1349, 1999. 2

[5] H. Irshad, L. Montaser-Kouhsari, G. Waltz, O. Bucur, J. Nowak, F. Dong, N. W. Knoblauch, and A. H. Beck. Crowdsourcing image annotation for nucleus detection and segmentation in computational pathology: evaluating experts, automated methods, and the crowd. In *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, pages 294–305. NIH Public Access, 2014. 2

[6] M. Javan-Roshtkhari and S. K. Setarehdan. A new approach to automatic classification of the curved chromosomes. In *Image and Signal Processing and Analysis, 2007. ISPA 2007. 5th International Symposium on*, pages 19–24. IEEE, 2007. 5

[7] Jeanpat. Overlapping chromosomes, learn to resolve them, 2010 (accessed December 2014). 2, 7

[8] L. Ji. Fully automatic chromosome segmentation. *Cytometry Part A*, 17(3):196–208, 1994. 2

[9] A. Kovashka, O. Russakovsky, L. Fei-Fei, K. Grauman, et al. Crowdsourcing in computer vision. *Foundations and Trends® in Computer Graphics and Vision*, 10(3):177–243, 2016. 2

[10] B. Lerner. Toward a completely automatic neural-network-based human chromosome analysis. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 28(4):544–552, 1998. 2

[11] C. V. P. Lukas Biewald. Crowdflower: Ai for your business, December 2007. 1

[12] N. Madian and K. Jayanthi. Overlapped chromosome segmentation and separation of touching chromosome for automated chromosome classification. In *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*, pages 5392–5395. IEEE, 2012. 2

[13] S. Minaee, M. Fotouhi, and B. H. Khalaj. A geometric approach to fully automatic chromosome segmentation. In *Signal Processing in Medicine and Biology Symposium (SPMB), 2014 IEEE*, pages 1–6. IEEE, 2014. 2

[14] M. Sameki, D. Gurari, and M. Betke. Characterizing image segmentation behavior of the crowd. 2

[15] M. Sameki, D. Gurari, and M. Betke. Icord: Intelligent collection of redundant dataa dynamic system for crowdsourcing cell segmentations accurately and efficiently. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2016 IEEE Conference on*, pages 1380–1389. IEEE, 2016. 2

[16] A. Savitzky and M. J. E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36(8):1627–1639, 1964. 5

[17] D. Schlesinger, F. Jug, G. Myers, C. Rother, and D. Kainmüller. Crowd sourcing image segmentation with iastaple. *arXiv preprint arXiv:1702.06461*, 2017. 3

[18] A. Sorokin and D. Forsyth. Utility data annotation with amazon mechanical turk. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on*, pages 1–8. IEEE, 2008. 2