Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 44 No. 6 (2023)

Automated System for Chromosome Karyotyping Detection

Akshan Krithick P.[1], Abinesh P.[2], Hemalatha R.[3]

[1, 2, 3] *Department of Computer Science and Engineering, St. Joseph's College of Engineering, Tamil Nadu, India*

Abstract:- In this research, we present a system aimed at identifying and categorizing chromosomes within the context of karyotyping. Leveraging the YOLOv8(You Only Look Once) object detection framework, our approach focuses on training the system to recognize and classify individual chromosomes through exposure to numerous images containing these genetic structures. The developed system offers fast operation, reducing the time required for chromosome analysis, and high accuracy, minimizing inherent errors in manual analysis. This reduces the result delivery time in clinical applications. In the development process, we made use of a dataset comprising annotated chromosome images as the training material for our YOLO model. Through fine-tuning, we achieved a Mean Average Precision(mAP) of 80.7% for an Intersection over Union(IoU) threshold of 0.5 suggesting remarkable precision and recall rates, minimizing misclassifications while consuming less amount of resources, ensuring reliable chromosome detection and classification that can be established even on resource-constrained devices. The output obtained from the automated system can be used to confirm chromosomal abnormalities in the input samples. The potential applications of our system empower medical professionals with an improved understanding of genetic conditions thereby resulting in a more accurate diagnosis and accelerating genetic research.

Keywords*: Chromosome analysis, Karyotyping, Medical diagnosis, Object detection, YOLO.*

1. Introduction

The karyogram shows that an average human has 46 chromosomes in total—23 from each parent. Homologous chromosomes arranged in a karyogram are pairs with the same size and structure. Humans have 22 pairs of autosomes. Autosomes are home to the genes that determine every aspect of human biology, except sex. We have two autosomes from each class, class 1 through class 22. The biggest chromosome is class 1, while the smallest is class 21. There are 44 autosomes. The two chromosomes that make up the remainder of the chromosomal count in humans are known as sex chromosomes, or chromosomes X and Y which are responsible for determining gender. They are positioned in a karyogram following autosomes. A person is considered feminine if they have two X chromosomes and masculine if they have both X and Y chromosomes. Doctors must complete years of training to identify genes and chromosomes with a low rate of mistakes due to their intricate nature. As most chromosomes are normal, it is more difficult for trainees to understand how aberrant chromosomes look in chromosomal abnormality diagnostics due to the unbalance of normal and abnormal data. Therefore, for doctors to become experts in this subject, they require more data and years of training. In recent years, automation of genetic diagnostics has become a prominent area of study.

2. Structure of Neural Networks

In instances where deep learning is used, the process that occurs is almost the same, which would go as follows: the algorithm acquires the data, and this data is subjected to non-linear transformations. Learning is done through the transformations and the output is acquired as a model. This carries on for several layers, through a multitude of trials, till a reliable and accurate output is arrived at.

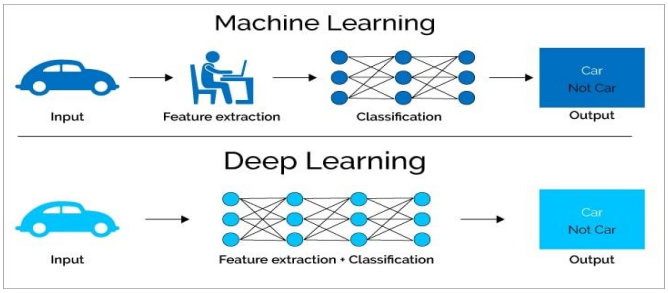

Fig. 1: Machine Learning vs Deep Learning

3. YOLO Algorithm

The computer vision community first learned about YOLO through a study published in 2015 by Joseph Redmon and several others under the heading "You Only Look Once: Unified, Real-Time Object Detection."

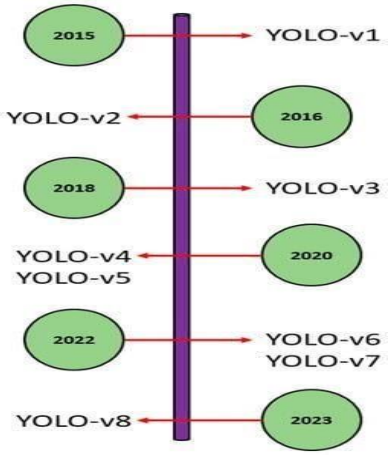

Fig. 2: YOLO evolution

By redefining object detection as basically, the research reframed the problem as a single-pass regression issue, working from image pixels to bounding boxes and class probabilities. Through the ability to anticipate several bounding boxes and class probabilities at once, the suggested method based on the "unified" notion improved speed and accuracy.

From 2016 to current, 2024, the YOLO family has seen remarkable evolution. Even though Joseph Redmon, the original creator, stopped with YOLO-v3, the "unified" concept's efficacy and potential have been further expanded by several others, with YOLO-v8 being the most recent addition to the YOLO family.

The fundamental idea put out by YOLO-v1 was overlaying a picture with a s×s grid cell. Each cell predicts bounding boxes and class probabilities for all objects whose center falls within the cell or whose bounding boxes significantly overlap with the cell. This allowed other cells to ignore that object if it made more than one appearance.

$$Confidence\ score=(object)*IoUtruthpred$$

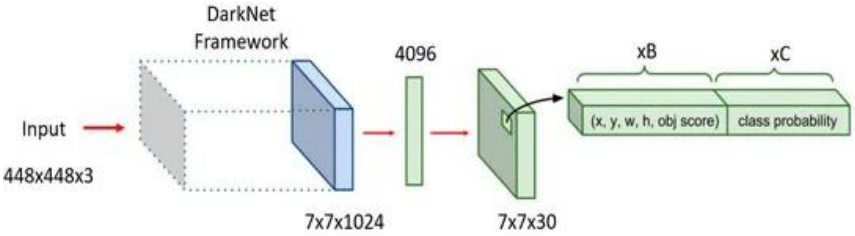

Fig. 3: YOLO-v1 preliminary architecture.

Joseph Redmon unveiled YOLO-v2 in 2016, capable of detecting over 9000 object categories. The goal was to keep the amazing speed factor of the original YOLO while eliminating or at least reducing the inefficiencies that were noticed. Several improvements were reported as a result of applying different strategies. With the internal design, batch normalization was included to enhance model convergence and speed up training. With its debut, other regularisation methods—like dropout, which lessens overfitting—were no longer necessary. The fact that only adding batch normalization increased the mAP over the original YOLO by 2% serves as a barometer for its efficacy.

YOLO-v3 is built on YOLO-v2 by incorporating skip connections to address the issue of losing fine-grained features. It also introduced multi-scale detection. YOLO-v4 was designed to be lightweight and run in real time. It experimented with different backbones and feature aggregation techniques. It also introduced several innovations including bag-of-freebies and bag-of-specials. YOLO-v5 is similar to YOLO-v4 in that it focuses on combining various techniques for better performance. It was the first YOLO architecture written in PyTorch. It also introduced automated anchor box learning.

YOLO-v6 is designed for industry and comes in various sizes. It is anchor-free and uses a decoupled head. It also uses a two-loss function. YOLO-v7 focuses on improving accuracy while maintaining speed. It introduced architectural reforms including E-ELAN and compound model scaling [7].

4. YOLO-V8

In January 2023, Ultralytics (who had previously published YOLO-v5) officially announced the introduction of YOLO-v8, the newest member of the YOLO family. While features are constantly being added to the YOLO- v8 repository and a formal paper release is imminent, preliminary comparisons between the newcomer and its predecessors are provoking.

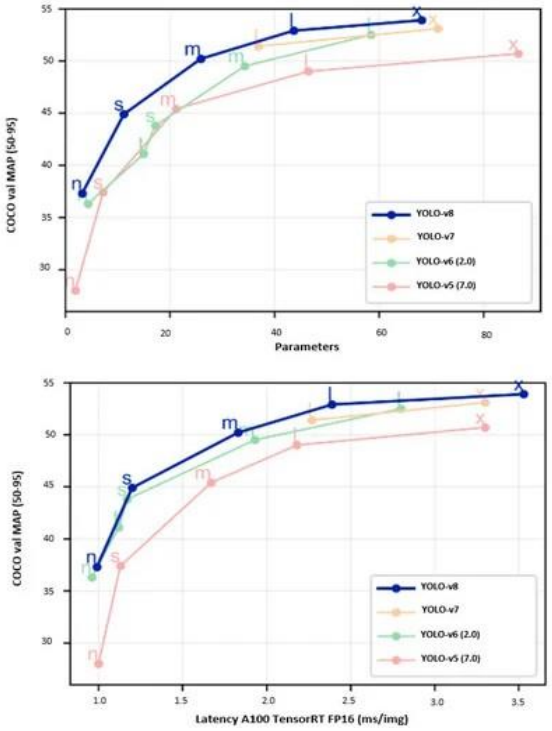

Fig. 4: YOLO-v8 comparison with predecessors.

Fig.4. shows that all YOLO-v8 models provide improvement as compared to the previous versions, trained on 640 picture resolution, and throughput with the same parameters, suggesting hardware-efficient, architectural changes. With YOLO-v5 exhibiting remarkable real-time performance and Ultralytics having published the benchmarking results for both YOLO-v8 and YOLO-v5, the main focus of YOLO-v8 will be on limited edge deployment at high-inference speed.

5. Problem Definition

Karyotyping involves the manual examination of hundreds of chromosomes in each cell. Through deep learning, this is automated, resulting in a significant amount of time and effort being saved while detecting and classifying chromosomes in images, quickly and accurately, which is especially valuable in clinical settings where rapid diagnosis is essential.

6. Objectives

To develop an accurate and efficient automated system for chromosome karyotyping detection using the YOLO object detection framework, with the primary goals of improving the speed, precision, and consistency of chromosome analysis.

7. Literature Survey

Understanding current research, scholarly publications, and pertinent articles about concurrent data access, locking systems, and techniques for ensuring data consistency while using a shared database were the main goals of the project's literature assessment.

The work in the article "Chromosome classification for karyotype composing applying shape representation on wavelet packet transform" written by L. V. Guimaraes, A. Schuck, and A. Elbern [1], addresses the challenge of automating the karyotyping process, which involves classifying chromosomes into specific groups. Karyotypes are images containing seven groups of chromosomes arranged in pairs, and manual interaction has traditionally been required due to the random disposition of chromosomes in photos. The authors highlight the advancements in digital image processing techniques that aim to automate karyotyping. Various methods have been proposed, considering characteristics such as the chromosomal band distribution or color. Within this framework, the study suggests a unique way to classify chromosomes according to their form. The suggested technique entails creating a signature from the chromosomal shape. Next, this signature is broken down using its best basis and the wavelet packet transform. Chromosomes can be classified into one of the seven groups (A, B, C, D, E, F, and G) in the karyotype by comparing the wavelet packet coefficients that match their signature. The results presented in the paper indicate that the proposed method successfully achieves the classification of chromosomes based on their shapes. By leveraging the wavelet packet transform, the authors contribute to the automation of the karyotyping process, potentially reducing the need for manual intervention in chromosome classification. The paper provides valuable insights into the application of digital image processing techniques for improving the efficiency and accuracy of karyotype analysis.

The article "Chromosome Classification with Convolutional Neural Network Based Deep Learning," by W. Zhang et al. [2] presents a convolutional neural network (CNN) based deep learning technique to automate the labor-intensive and time-consuming procedure of karyotyping for genetic disease detection. Sorting chromosomes into 23 categories is the goal, which is a big step toward having completely automated karyotyping. The suggested technique is validated on a dataset with 4,830 chromosomes after being trained and tested on a sizable dataset with 10,304 chromosome pictures. The CNN-based deep learning network astoundingly attained an astounding accuracy of 92.5%, exceeding the results of three additional techniques that have been published in the literature. The paper also presents a new statistic, the "proportion of well-classified karyotype," intended to evaluate how well the suggested approach will work in real-world scenarios involving medical practitioners. Notably, the outcomes show a 91.3% success rate on this parameter, indicating that the suggested categorization approach may be a useful aid for medical professionals in the diagnosis of genetic illnesses. The automation of karyotyping has advanced significantly as a result of this research, which may improve the efficiency and precision of diagnosing genetic disorders.

To classify chromosomes, the research "Automatic Chromosome Classification using Deep Attention Based Sequence Learning of Chromosome Bands," by M. Sharma, Swati, and L. Vig [3], proposes the Residual Convolutional Recurrent Attention Neural Network (Res-CRANN) utilizing the property of the band sequence. A set of feature vectors that are extracted from the feature maps produced by the convolutional layers of Residual Neural Networks (ResNet) are fed into Recurrent Neural Networks (RNN) which can be used to train Res-

CRANN end-to-end. An attention mechanism is then used to further categorize the RNN output sequences into one of the 24 labels.

In the domain of chromosomal analysis, the paper by J. Zhang et al. [4] "Chromosome Classification and Straightening Based on an Interleaved and MultiTask Network," contributes a novel technique for classifying and aligning chromosomes, addressing key aspects of karyotyping crucial for detecting chromosomal abnormalities. The proposed approach unfolds in three stages: Initially, multi-scale characteristics are learned via an interleaved network. chromosomal joint detection is then achieved by feeding high-resolution data into a convolutional neural subnetwork, and the chromosomal type and polarity classification is carried out by a multi-layer perceptron subnetwork. In the third step, curved chromosomes are straightened using identified joints, improving the readability of banding data.

In the realm of human chromosome classification, the paper "Human chromosome classification using Competitive Neural Network Teams (CNNT) and Nearest Neighbour Algorithm," by S. Gagula-Palalic and M. Can [5], devises a novel strategy utilizing Competitive Neural Network Teams (CNNT) and Nearest Neighbour algorithms. There are 22 pairs of autosomes in a human cell, which correspond to 22 different classes that need to be determined. Competitive Neural Network Teams (CNNTs), a painstakingly arranged committee of 462 basic perceptrons, are the basis of the unique classification technique that is suggested. With training, every CNNT perceptron can distinguish between two classes, yielding a total of 22 × 21 learning machines. A deliberate strategy is used to set dummy perceptrons to zero for chromosomes that belong to the same class. Each learning machine's output is captured in a 22x22 decision matrix that represents the testing data's outcome.

The study conducted by W. Ding et al.'s paper [6], "Chromosome Karyotype Classification Using FasterRCNN and the Segmentation and Enhancement Preprocessing Model," discusses Prenatal testing for chromosomal abnormalities as essential to ensure the proper survival of fetuses. The intricacy and labor-intensiveness of analyzing chromosomal karyotype images provide formidable obstacles to medical diagnosis. In response, the study suggests a preprocessing model that combines feature augmentation and object segmentation. This model, when paired with a deep learning network architecture, creates an automated classification model for the detection of chromosomal karyotypes.

The preprocessing model focuses on chromosomal karyotype photos through pixel-level extraction and feature improvement to give the ensuing deep learning network more comprehensible data. The study builds a classification recognition network, performs algorithmic analysis of chromosomal karyotype preprocessing, and uses the network's detection findings to confirm the efficacy of the preprocessing model. The proposed automatic analysis model based on deep learning has the potential to offer accurate reference information for medical practitioners, thereby reducing the repetitive diagnostic workload. This research contributes to the field by addressing the challenges associated with chromosome karyotype analysis, emphasizing the integration of preprocessing techniques with deep learning networks to enhance the interpretability and efficiency of chromosomal abnormality detection in prenatal screening.

8. Methodology Modules

DATA COLLECTION

Dataset Size:

Training set:

The training set consists of 453 images.

Test Set:

There is a test set containing 22 images.

Validation Set:

A validation set is present with 43 images. NC: 24. Names: ['1', '10', '11', '12', '13', '14', '15', '16', '17', '18', '19', '2', '20', '21', '22', '3', '4', '5', '6', '7', '8', '9', 'X', 'Y']

The numerical labels and class names are necessary for properly annotating the images during the labeling process, configuring the model's output layer, and interpreting the model's predictions during evaluation. The annotated chromosome image dataset we use from Roboflow is well-structured and suitable for a multi-class image classification task focused on identifying specific types of chromosomes.

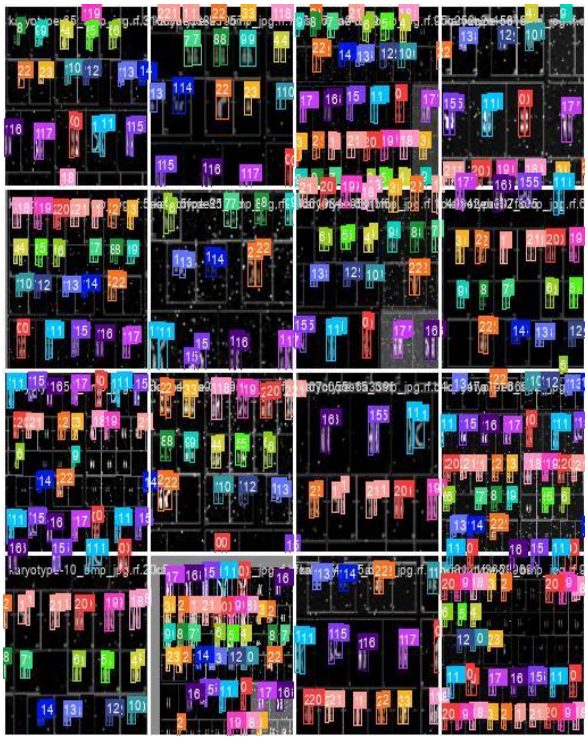

Fig. 5: Train batch

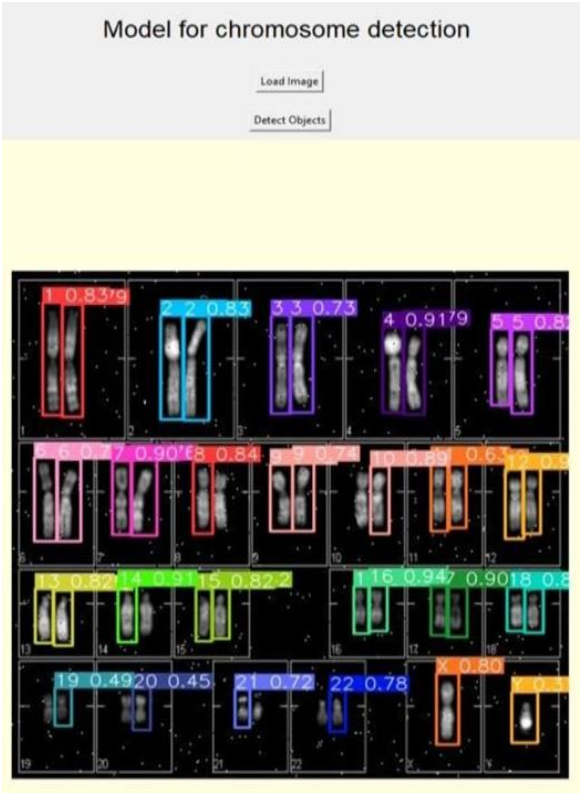

Fig. 6: Chromosome Detection

9. Model Training

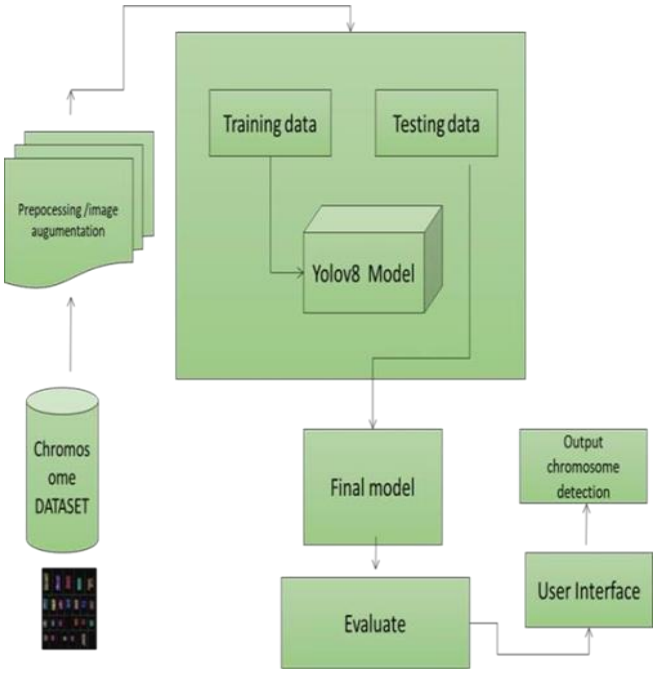

Fig. 7: Workflow

Yolov8 Architecture

A novel loss function, anchor-free detection head, and backbone network are characteristics of YOLOv8. **Fig.8.** outlining the improved head structure and the backbone of YOLOv8 was contributed by Github user RangeKing. RangeKing found the following modifications when comparing this graphic to a similar analysis of YOLOv5:

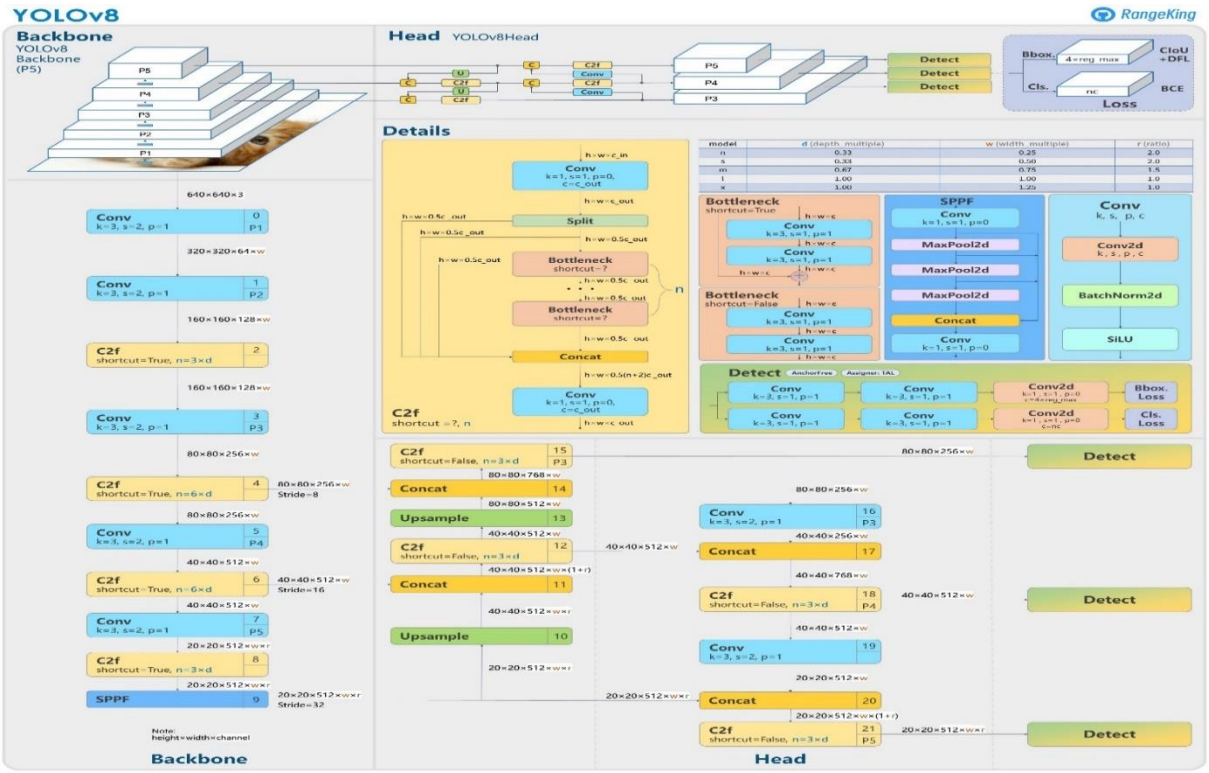

Fig. 8: Yolov8 Architecture

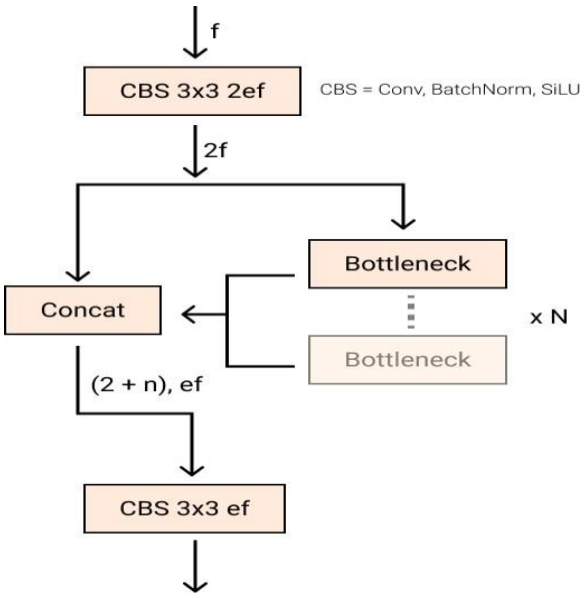

Fig. 9: The C2f module

The C2f module was used in place of the C3 module. Only the output of the final Bottleneck was utilized in C3, however in C2f all of the outputs from the Bottleneck (the two 3x3 convolutions with residual connections) are concatenated. This enhances feature extraction compared to C3, which only used the final output. In the Backbone, they swapped out the original 6x6 Conv block for a 3x3 Conv block. A comparison of the two model backbones shows that they removed two of the Conv (No. 10 and No. 14 in the YOLOv5 configuration) for efficiency. In the bottleneck, they swapped out the original 1x1 Conv with a 3x3 Conv block.

Anchor-free bounding boxes

As stated in the blog article on YOLOv8 by RoboFlow, an Ultralytics partner, YOLOv8 is an anchor-free object detection model, unlike previous YOLO versions which relied on predefined anchor boxes. These anchors helped the model predict bounding boxes efficiently, but YOLOv8 takes a different approach. It directly predicts the center coordinates, width, and height of bounding boxes, potentially offering improved accuracy and speed compared to earlier versions. YOLOv8 performs bounding box predictions pixel-wise. Additionally, due to the absence of anchors, it is end-to-end differentiable and offers context-aware prediction.

Stopping the Mosaic Augmentation before the end of training

YOLOv8 uses a somewhat altered version of the supplied images at every training epoch. We refer to these adjustments as augmentations. One such, mosaic augmentation involves stitching four photos and feeding them as input. This forces the model to learn the identities of the objects in new positions, partially blocking each other through occlusion. YOLOv8 provisions halting this procedure during the last training epochs as it has been demonstrated that utilizing this for the full training regime might negatively impact prediction accuracy. This makes it possible to perform the best training pattern without using up the entire run.

10. Model Testing

After training a YOLOv8 model, it is important to evaluate its performance on a testing set to assess its accuracy. The testing set is a different dataset that the model hasn't encountered during training, and is used to gauge how well the model can generalize to fresh information. For evaluating the performance of a trained model, two plots are common; the accuracy plot and the loss plot.

Accuracy Plot: An accuracy plot demonstrates how well the model performs on the testing set as the training epochs increase. The accuracy is usually measured as the mAP. As the model is trained, its accuracy on the testing set generally increases. The mAP might eventually plateau due to factors like early stopping, indicating the model has reached its optimal performance.

Loss Plot: A loss plot tracks the loss values on each training epoch on the training and testing sets. The difference between the model's projected and actual outputs is often used to calculate the loss. If loss decreases as the epoch increases, it indicates that the model is improving its ability to make predictions.

11. Results and Discussion

1. **Epoch**: This column represents the training epoch or iteration number. An epoch is one complete pass through the entire training dataset.
2. **train/box_loss**: This is the loss resulting from using bounding box estimates during training. It measures how well the model predicts the locations of objects (in this case, chromosome regions).
3. **train/cls_loss**: This loss pertains to the classification aspect of the model. It measures how well the model can classify the objects or regions it identifies, distinguishing between chromosomes. It is computed using the Cross-Entropy Loss function.
4. **train/dfl_loss**: DFL stands for Distribution Focal Loss, used to address issues of class imbalance in detecting objects of different classes. It aims at downplaying the influence of easily classified examples and emphasizes the learning process for less frequent classes
5. **metrics/precision(B)**, **metrics/recall(B)**, **metrics/ mAP50(B)**, **metrics/mAP50-95(B)**: These columns show various metrics that are commonly used in object detection tasks, evaluating precision, recall, Mean Average Precision(mAP) at an IoU(Intersection over Union) threshold of 0.5, and mAP over a broader range of IoU thresholds (50-95). It accounts for the performance of the model for all classes in an object detection task
6. **val/box_loss, val/cls_loss, val/dfl_loss**: These are the validation losses for bounding box predictions, classification, and class imbalance. These losses represent distinct aspects that contribute to the model's overall loss.
7. **lr/pg0**, **lr/pg1**, **lr/pg2**: These columns represent the learning rates for different parameter groups in the optimization process, often used in deep learning when using techniques like learning rate schedules.

Overall, the results table is a record of how well the model performed during training and validation across different epochs. It helps assess whether the model is improving, overfitting, or converging to a suitable solution for the chromosome detection task. The losses and metrics provide insights about various aspects of model performance, such as object localization, and classification accuracy.

Confidence curve

In our model, each detected object comes with a confidence score that represents the model's confidence in its prediction. This score is typically used to filter out low-confidence detections and improve the precision of the model. **Fig.10.** shows the F1 score at different confidence thresholds. The F1 score is the harmonic mean of precision and recall.

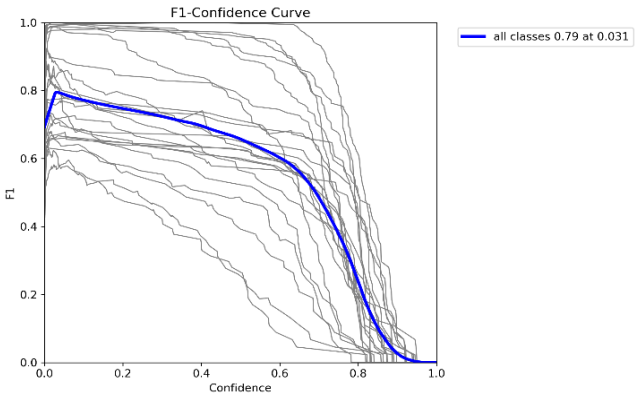

Fig. 10: f1 confidence

Confusion matrix

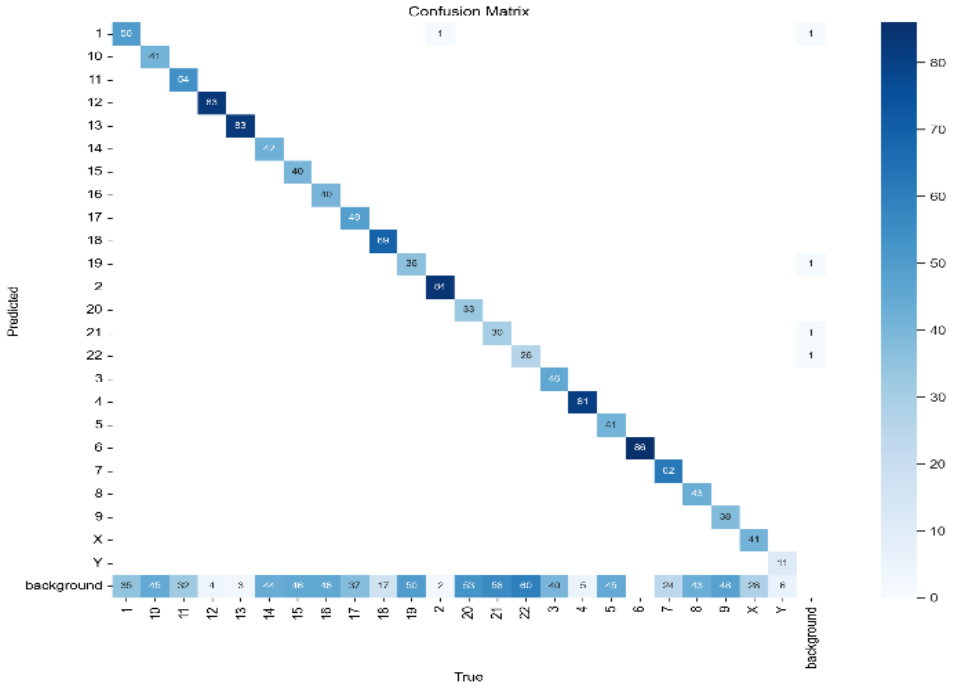

Fig. 11: Confusion matrix without normalization

In a non-normalized confusion matrix, the values in each cell represent the raw count of predictions made by the model. It shows the actual numbers representing true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

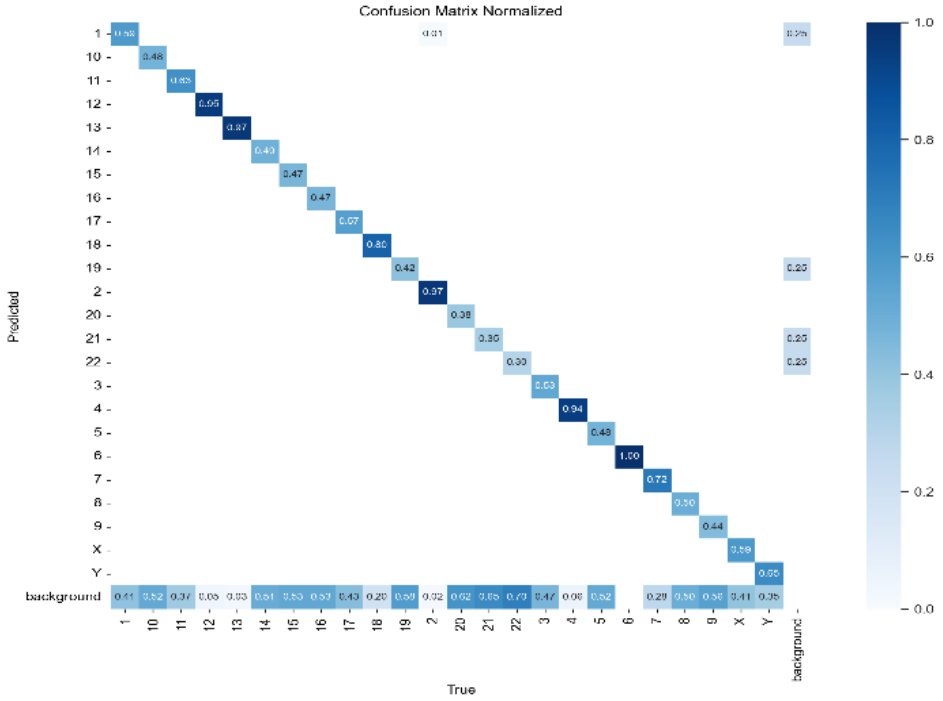

Fig. 12: Confusion matrix with normalization

In a normalized confusion matrix, the values in each cell are normalized to show the proportion or percentage of predictions relative to the total number of instances in each class. It is particularly useful when you want to compare models across datasets with different class distributions.

Labels

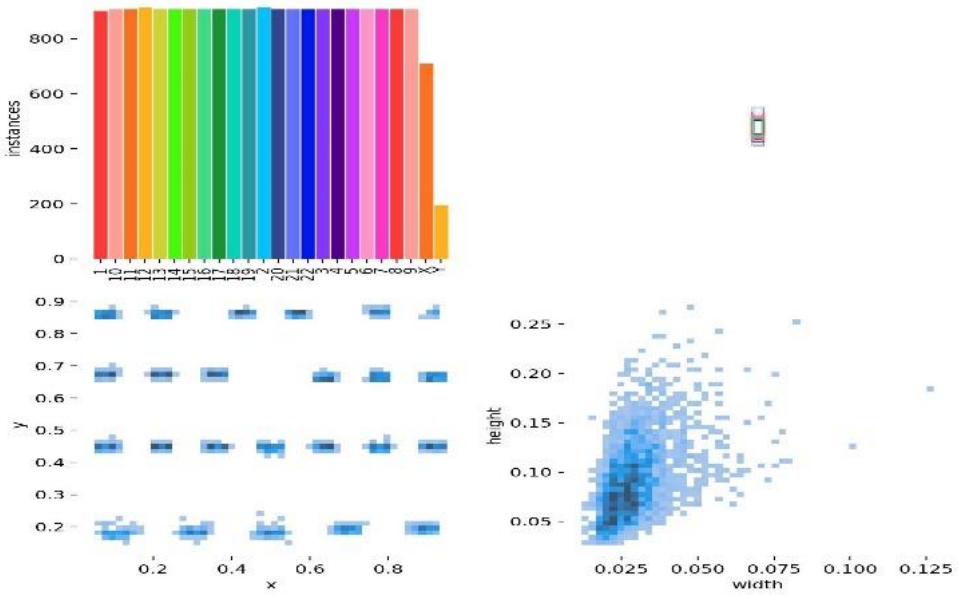

Fig. 13: Labels

Annotation is a manual process where we mark the regions of interest in each image. For chromosome detection, this typically means drawing bounding boxes around areas of the image where types of chromosomes are present. The markings, in the form of bounding boxes, serve as labels that guide the model's learning. The model learns to associate specific visual features with corresponding chromosome types, enabling it to detect and classify them in unseen images.

Mean Average Precision

It is a valuable metric for evaluating the model's effectiveness. The Average Precision (AP) values calculated for each class are averaged to obtain the mAP, representing the overall performance of the model across all the classes. **Fig.14.** shows the trade-off between precision and recall, for an IoU threshold of 0.5.

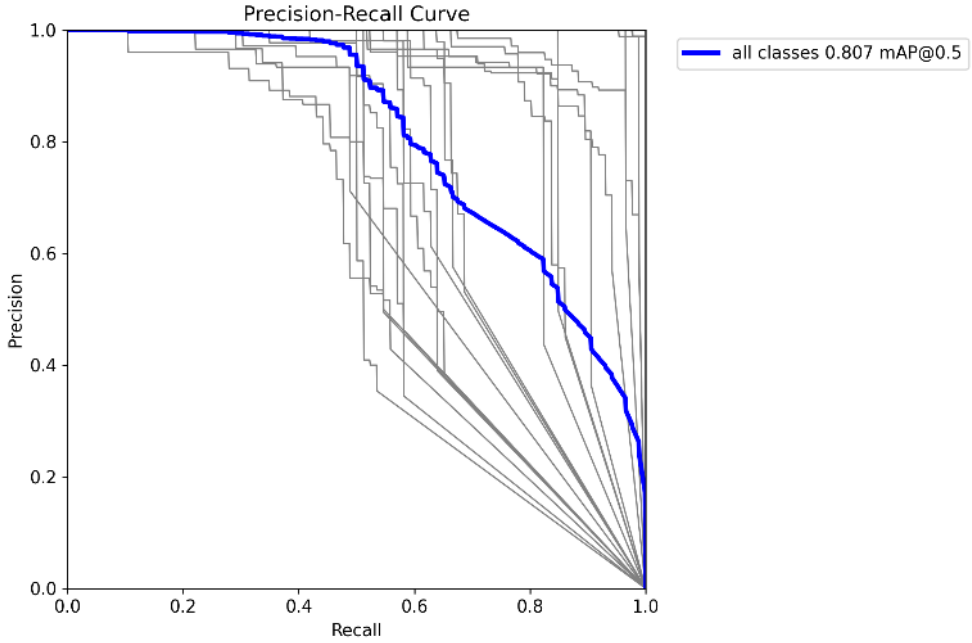

Fig. 14: Precision-Recall Curve

Evaluation

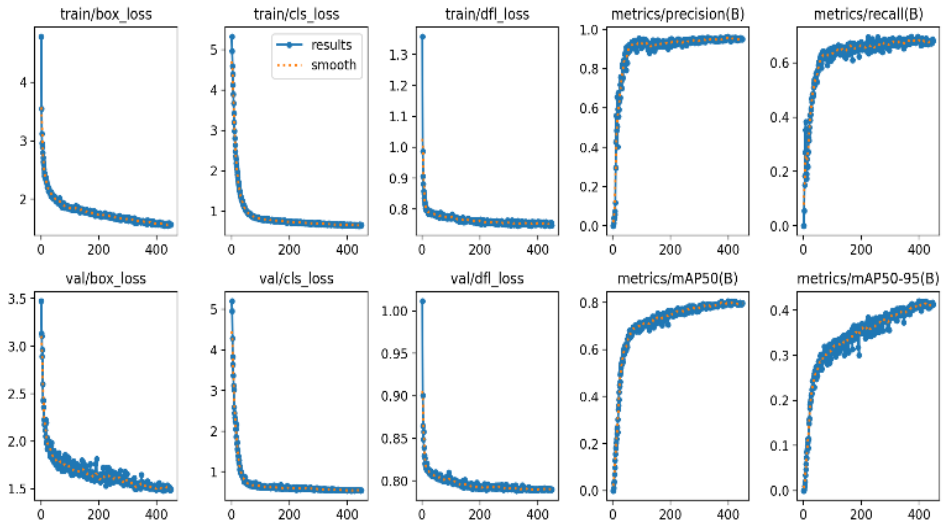

Fig. 15: Results

Fig.15. denotes the change in loss and relevant performance metrics across different epochs

12. Conclusion

Leveraging the YOLOv8 object detection framework, a potent tool in computer vision, we trained the system to recognize and classify individual chromosomes through exposure to a diverse set of annotated images. Our model has a mAP of 0.807 at an IoU threshold of 0.5 on all classes. For reproducibility, the hyperparameters we had explicitly tuned for the training process are imgsz= 100, epochs=500, device='cpu', and iou= 0.5 and the model used is pre-trained YOLOv8n. It is worth noting that our model is computationally inexpensive and has lesser training time when compared to models that have the imgsz parameter set to a higher resolution or use a bigger model of YOLOv8. When the same training process was conducted using the slowest but most accurate pre-trained YOLOv8x model instead of the smallest and fastest pre-trained YOLOv8n, the mAP on all classes with an IoU threshold of 0.5 was noted to be 0.83 but the training time lasted about 6 times as much. When we made use of the pre-trained YOLOv8x model with the imgsz parameter set to 320, the model resulted in a 0.987 mAP at an IoU threshold of 0.5 on all classes and an f1 score of 0.98 but the training time was nearly 9 times as such compared to our initial model even while the number of epochs was set to just 100. The trade-off between accuracy and training speed is discernible. Future work alluding to this research involves making use of a customized YOLOv8 architecture by modifying the depth, width, and number of maximum channel parameters while also making modifications to the blocks in the model architecture, based on the size of the object to be detected in the image. The implications of our work extend across the fields of clinical genetics and research, providing medical professionals with a reliable and computationally efficient tool for better understanding genetic conditions and facilitating precise diagnoses.

References

[1] L. V. Guimaraes, A. Schuck and A. Elbern, "Chromosome classification for karyotype composing applying shape representation on wavelet packet transform," Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE Cat. No.03CH37439), Cancun, Mexico, 2003, pp. 941-943 Vol.1, DOI: 10.1109/IEMBS.2003.1279921.

[2] W. Zhang et al., "Chromosome Classification with Convolutional Neural Network Based Deep Learning," 2018 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Beijing, China, 2018, pp. 1-5, DOI: 10.1109/CISP- BMEI.2018.8633228.

[3] M. Sharma, Swati and L. Vig, "Automatic Chromosome Classification using Deep Attention Based Sequence Learning of Chromosome Bands," 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 2018, pp. 1-8, DOI: 10.1109/IJCNN.2018.8489321.

[4] J. Zhang et al., "Chromosome Classification and Straightening Based on an Interleaved and Multi-Task Network," in IEEE Journal of Biomedical and Health Informatics, vol. 25, no. 8, pp. 3240-3251, Aug. 2021, DOI: 10.1109/JBHI.2021.3062234.

[5] S. Gagula-Palalic and M. Can, "Human chromosome classification using Competitive Neural Network Teams (CNNT) and Nearest Neighbor," IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI), Valencia, Spain, 2014, pp. 626- 629, DOI: 10.1109/BHI.2014.6864442.

[6] W. Ding, L. Chang, C. Gu, and K. Wu, "Classification of Chromosome Karyotype Based on Faster-RCNN with the Segmentation and Enhancement Preprocessing Model," 2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Suzhou, China, 2019, pp. 1-5, DOI: 10.1109/CISP-BMEI48845.2019.8965713.

[7] Hussain, M. YOLO-v1 to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection. Machines 2023, 11, 677. https://doi.org/10.3390/ machines11070677

[8] Gert de Graaf, Frank Buckley, and Brian G. Skotko, “Estimation of the number of people with Down syndrome in the United States,” article in Genetics in medicine: official journal of the American College of Medical Genetics, September 2016, DOI:10.1038/gim.2016.127.

[9] S. Saranya and S. Lakshmi, "Classification of Chromosomes to Diagnose Chromosomal Abnormalities using CNN,” 2023 International Conference on Artificial Intelligence and Knowledge Discovery in Concurrent Engineering (ICECONF), Chennai, India, 2023, pp. 1-5, DOI: 10.1109/ICECONF57129.2023. 10083710.

[10] Granlund, G. H. Identification of human chromosomes by integrated density profiles. IEEE Transactions on Biomedical Engineering, 23: 182-192, 1976.

[11] Granum, E. Application of statistical and syntactic methods of analysis to classification of chromosome data. *Pattern Recognition Theory and Applications*. J. Kittler, K.S. Fu, and L.S. Pau (Eds.) D. Reidel, 1982, pp. 373–397.

[12] Groen, F.C.A., ten Kate T.K., Smeulders, A.W. M., and Young I. T. Human chromosome classification based on local band descriptors. *Pattern Recognition Letters*, 9:211-222, 1989.

[13] Gregor, J., and Thomason, G. Hybrid pattern recognition using Markov networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:651–656, 1993.

[14] Jennings, A. M., and Graham, J. A neural network approach to automatic chromosome classification. *Physics in Medicine and Biology*, 38:959–970, 1993.

[15] Errington, P. A., and Graham, J. Application of artificial neural networks to chromosome classification. *Cytometry*, 14:627–639, 1993.

[16] Rumelhart, D. E., Hinton, G. E., and Williams, R. J. Learning internal representations by error propagation. *Parallel Distributed Processing: Explorations in the Microstructures of Cognition. Vol. 1*. D.E. Rumelhart and J.L. McClelland (Eds.) MIT Press, 1986, pp. 318–362.

[17] Piper, J. Variability and bias in experimentally measured classifier error rates. *Pattern Recognition Letters*, *13:685–692*, 1991.