Group 16. Automatic Classification of Human Chromosome: a Systematic Review

1st Mahmoud Hilani

mahmoudhilani18@gmail.com

***Abstract*—This project attempts to classify chromosome images using deep learning techniques, transfer learning, and modern vision transformers. Four chromosomes, 6, 7, 8, and 9, were picked for classification to maintain valuable learning while also limiting dataset size. Data analysis was performed to make sure the data was clean, and how complex the learning required is, showing that the data is not linearly separable through PCA. A straightening algorithm and data augmentation were applied to the data, while preventing cross-leakage between training, validation, and test splits. Fivefold cross-validation was used for the main runs to produce evaluation metrics. Three models were picked: a CNN trained from scratch, a ResNet-50 model, and a DINOv2 model. Two forms of experiments were run, one with raw images with augmentation and the other with straightened, augmented images added on top. Results show that the ResNet-50 was the most stable overall, showing the least variance in test accuracy and macro-F1. The CNN starts off randomly but learns important features well, showing the best results when only raw images are introduced. The DINOv2 performed the best in raw and straightened images combined but showed signs of instability when data is limited, sometimes collapsing to one class. Ethical considerations were made, with missing patient-level data being the main concern, potentially giving optimistic results.**

I. INTRODUCTION

Chromosomal analysis or karyotyping is a cytogenetic procedure used to examine an organism's complete set of chromosomes. In humans, a typical somatic cell contains 46 chromosomes arranged into 23 pairs: 22 pairs of autosomes and one pair of sex chromosomes [1]. Karyotyping is used for detecting genetic abnormalities like Down syndrome, Edwards syndrome, chronic myelogenous leukemia, and Turner syndrome by analysing chromosome number, size, shape, banding pattern, and centromere position [2]. Karyotyping is carried out by culturing cells, separating the chromosomes from the cell nucleus during metaphase, when chromosomes are most visible, and labeling the cells on a slide to enable microphotography [2]. Staining techniques such as G-banding use Giemsa dye to produce the characteristic dark and light bands that help specialists distinguish individual chromosomes.

Although valuable, karyotyping is not a simple procedure. Considerable manual time and effort is required for segmenting out and classifying the chromosomes in images from a cell culture [2]. To help reduce manual labor, continuous research has been done to try and automate key processes such as chromosome type recognition [3]. More modern approaches such as next-generation sequencing (NGS) and optical genome mapping (OGM) have appeared, but karyotyping remains a staple in detecting large-scale numerical and structural chromosome abnormalities, such as aneuploidies, balanced translocations, Robertsonian translocations and low-level mosaicism, and marker chromosomes [4]. Its cost-effectiveness and suitability for low-resource settings lead to continued use. AI-guided karyotyping addresses the limitations of traditional karyotyping with systems that streamline and standardize the image capture and analysis process with high efficiency, using models trained on large chromosome image datasets to support laboratory professionals [5].

This project focuses on classifying isolated chromosome images by type. Although a normal human karyotype contains 23 pairs, the image-classification task is defined at the level of chromosome type. Therefore, each cropped chromosome image is assigned to one of the possible chromosome categories, such as chromosomes 1-22, X, or Y, rather than to a physical pair. This makes the task suitable for supervised image classification while still reflecting a real step in automated karyotyping.

The aim of this study is to evaluate how well deep learning and transfer learning can classify isolated chromosome images accurately using a limited image dataset. A convolution neural network is trained from scratch, while pretrained models such as ResNet-50 and DINOv2 are evaluated to test whether learned visual representations improve classification performance. The models are compared using accuracy, macro-F1 score, confusion matrices, and cross-validation results, with attention given not only to overall performance but also to class-level errors and the visual features the models appear to use to make sure models are learning features relevant to karyotyping.

II. RELATED WORKS

Early automated karyotyping studies treated the workflow as a sequence of computer-vision tasks,

including chromosome segmentation, feature extraction, classification, and karyogram assembly [6], [7], [8], [**?**]. These included simpler forms of machine learning such as multi layer perceptron neural networks, medial axis-based feature extraction, multiplex fluorescent in-situ hybridization, and Naive Bayes classification. They were designed as multi-stage computer-vision pipelines.

More recent work implements deep-learning models, learning discriminative chromosome representations from image data. For example, Lin et al. [9] use the same G-band chromosome dataset (CIR-Net) used in this project, which was made publicly available by them. Their main model is based on Inception-ResNet, adapted for 24-class chromosome classification, and introduces chromosome data augmentation (CDA) to increase the diversity of the limited training data. In CIR-Net, the original images are first separated into a test set and a temporary development set. CDA is then applied to the development set, and the augmented images are randomly assigned to either the training or validation set. This avoids direct leakage into the test set, but it may allow augmented versions of the same original chromosome image to appear in both training and validation. As a result, validation performance may be optimistic, since the validation set is not fully independent at the source-image level. They reported a classification accuracy of 95.98%, showing the potential of CNN-based architectures for this task.

Qin et al. [10] uses a local/global CNN to capture whole chromosome shape and local discriminative regions. Global features can mean overall length, general shape, and centromere position while local features can be local banding patterns, short chromosome segments near the centromere, and fine texture differences that may be lost when only considering the full image. The varifocal mechanism shifts the attention from the whole chromsome to useful local features. This study performs classification only on similar chromosome classes, and visually similar chromosome classes may differ only in subtle local patterns. Whole image shape alone might be insufficient. The model is trained on a relatively large amount of data, containing 87831 images. Their global branch achieved 97.8% type-classification accuracy, while the full Varifocal-Net improved performance to 98.9%.

Li et al. [3] proposed SR-DenseNet, short for Segmentally Recalibrated Dense Convolutional Network. It is a chromosome-specific CNN architecture. They combine dense connections, squeeze-and-excitation blocks, segmental recalibration, and model fusion. Features from early convolutional layers are reused by later layers to help the model preserve chromosome features. With Squeeze-and-Excitation blocks, the model can learn which feature channels are more important. Segmental calibration lets the model emphasize useful chromosome features at different abstraction levels. Lastly, they combine multiple recognition models to reduce classification error. They evaluated on 2 public datasets with G and Q-bands, with their best G-band result reaching 99.01% accuracy.

D'angelo & Nanni [1] also use the same CIR-Net dataset, but classify with ResNet-50, Swin Transformer, augmentation, straightening, and ensembles. They use the entire dataset and classify all 24 labels. They used five-fold cross-validation and reported precision, recall, accuracy, and F1-score. Their results showed that augmentation was important for improvement. A single ResNet-50 without augmentation achieved 88.08% accuracy, while ResNet-50 with CDA reached 97.59%. The best CNN ensemble, combining CDA, straightening, and feature-transform augmentation, achieved 98.56% accuracy, while the Swin Transformer ensemble achieved 99.63%. However, the authors mentioned that the dataset does not provide karyotype identifiers, so random image-level splits may overestimate generalization compared with stricter evaluation.

Chen et al. [11] tested whether two-step transfer learning would outperform standard ImageNet transfer learning. They first trained a model from scratch, used pretraining on the target dataset, then trained on an intermediate dataset, followed by the target dataset again. The two datasets were BioImLab (Q-band) and CIR (G-band), which is the same dataset family as CIR-Net. They evaluated 11 model families, comparing both transfer learning and vision transformers. Key results mentioned that two-step transfer learning improved Macro-F1 by about 0.8 to 3.3 percentage points for modern architectures such as ConvNeXt, Swin Transformer, ViT, and MobileNetV3 on the Q-band dataset. Meanwhile, the G-band dataset showed small gains of 0.1 to 0.7 percent since performance was already near saturation. This is relevant to this study because it suggests that pretrained visual backbones may offer diminishing returns on the CIR dataset, especially when the dataset is relatively clean and class-discriminative features are already learnable from limited training data.

III. Dataset Overview

This study uses Lin et al.'s CIR-Net dataset (https://github.com/CloudDataLab/CIR-Net (accessed on 15 April 2026)). It has 2986 images, which represent 65 different karyotypes, of which 32 are male and 33 are female. With the exclusion of labels 8, 12, 16, and 19, which correspond to just 129 chromosomes, the dataset has 130 chromosomes for every label from 1 to 22. 98 type X chromosomes, defined as 23, and 32 type Y chromosomes, defined

as 24, make up the sex chromosomes. Pixels in the grayscale, 8-bit color depth images range in intensity from 0 (black) to 255 (white). It is 224 × 224 pixels in size, with a black backdrop and grayscale chromosomes. Figure 1 shows samples from the dataset.

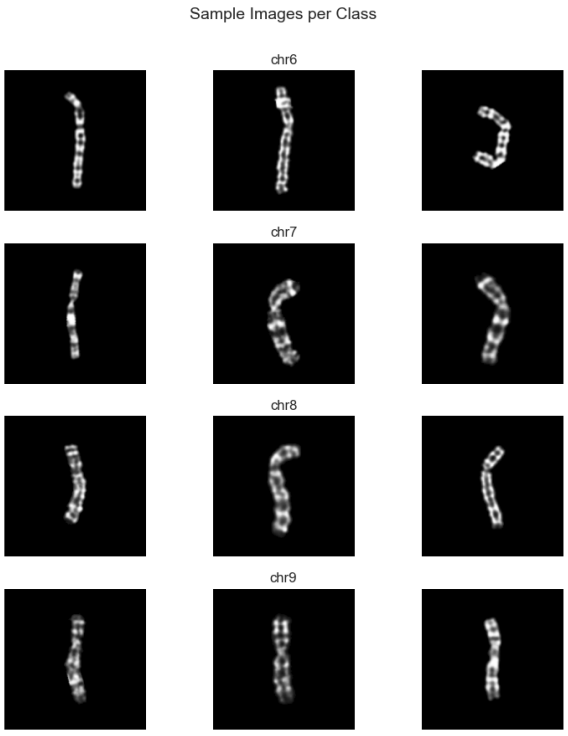


Figure 1. Samples of chromosomes 6, 7, 8, and 9

A limitation of the dataset is that the released files do not provide patient, metaphase, or karyotype identifiers. As a result, the train, validation, and test splits in this study were performed at image level using stratified sampling. This matches the limitation noted by D'Angelo and Nanni [1] for the same dataset. Therefore, chromosomes originating from the same karyotype may be distributed across different folds. This could make the evaluation slightly optimistic because images from the same case may share staining, acquisition, or preparation characteristics. A future evaluation would require metadata that allows all images from the same karyotype or patient to be kept within a single split.

IV. Exploratory Data Analysis

The dataset was first inspected using exploratory data analysis and prepared. This study only focuses on classes 6 to 9. Picking 4 classes allows each class to have a sufficient amount of images for training. Similar chromosome classes were picked to enable more valuable and complex learning. This produced a working dataset of 519 grayscale TIFF images, each with size 224 by 224 pixels. The class distribution was almost balanced, with 130 images for chr6, chr7, and chr9, and 129 images for chr8.

To calculate image-level statistics, a threshold value of 15 was used to separate the chromosome pixel values from the background. Features such as mean intensity, intensity standard deviation, foreground ratio, bounding-box area ratio, aspect ratio, centroid position, and estimated orientation were computed. With these features, measureable morphology (the form of organisms) can be identified between chromosome classes before training.

The EDA showed a decrease in mean intensity, foreground ratio, and bounding-box area from chr6 to chr9, showing that there are some measureable differences to be learned. Figure 2 graphs these features for visual comparison. Some outliers were identified but kept as this is real data that could appear in a lab. Duplicate and near-blank samples were checked for and none were identified.

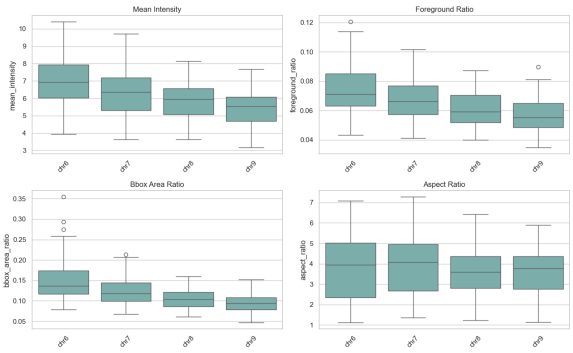


Figure 2. Mean intensity, foreground ratio, bounding-box area ratio, and aspect ratio for chromosomes 6, 7, 8, and 9

PCA was also used as a diagnostic to examine whether the selected classes were separable in flattened pixel space. The first two principal components explained 26.19% of the variance, while the first ten explained 50.25%. The PCA projection showed substantial overlap between the four classes, with a silhouette score of -0.0628 and nearest-centroid accuracy of 0.3006. This indicates that simple linear structure in raw pixel space is not sufficient for reliable classification, supporting the use of nonlinear visual models.

PCA was also used to examine whether the selected classes were linearly separable. The first two explained 26.19% while the first ten explained 50.25%. The four classes significantly overlap in the first two principal components, as seen in Figure 3. Although the overlap between individual samples suggests that linear separation in raw pixel space is limited, the class centroids show that chr8 and chr9 are closest in this projection, whereas chr6 is slightly more out.



Figure 3. PCA class-centroid plot

Morphology features were associated with the first four principal components in order to comprehend the PCA structure. PC2 is closely linked to aspect ratio and bounding-box fill ratio, but PC3 is more closely linked to foreground ratio and mean intensity, as seen in Figure 4. This implies that rather than being only driven by class identification, the dominant variance in the raw photos is also influenced by size, compactness, and location.

V. Data Preparation

A. Projection arm straightening

Chromosome images in the dataset vary in orientation and curvature. Since chromosome classification can depend on banding pattern, centromere position, and overall morphology, large variation in pose may make the classification task harder. To explore whether pose normalization could help, a projection-based chromosome straightening method was implemented. The aim was to reduce curvature while preserving the chromosome foreground and keeping the output image on the original 224 by 224 canvas.

As seen on Figure 2 and Figure 12, chromosomes can vary in orientation and curvature. Large variation in pose may make the classification procedure more difficult because chromosomal categorization can depend on banding pattern, centromere position, and overall shape [12]. A projection-based chromosomal straightening technique inspired by Javan Roshtkhari and Setarehdan was used to minimize curvature while maintaining the output image on the original 224 by 224 canvas with the chromosomal foreground.

Using a fixed intensity threshold, chromosomal foreground is separated from the black backdrop. After estimating the foreground pixels' dominant orientation, it looks through potential rotations to find a stable vertical axis. By searching for a low-width valley that provides a reasonable balance between the two chromosomal arms, a bend row is chosen from the foreground projection profile. Around this bend row, the chromosome is then divided into upper and lower arms. The two arms are then recombined at the bend point and centered on the original image canvas after each arm has been individually rotated to reduce its vertical projection width. Figure **??** shows the average image per class before applying straightening, and Figure 2 shows the average image per class after.

Straightening can distort chromosomal shape or remove banding information, so the method is quality controlled. If foreground is lost, the chromosome is disconnected, an unrealistic aspect ratio is given, or horizontal artifacts are introduced, the method falls back to the original image. This way, only accepted transformed images are added to the data.

B. Data Splitting and Augmentation

For evaluation, stratified image splitting was performed. Initial experiments contained a single stratified 70/15/15 split, while the main comparison was performed using five-fold cross-validation. In each fold, the test set was held out while the remaining data was split to reserve 15% of the full dataset for validation, leaving the rest for training. Validation was used for model selection and early stopping.

These split definitions are consistent across all comparisons, which include the raw data only, the straightened images only, and both together. Importantly, the split happens before the addition of straightened images, so raw and straightened images stayed in the same train, validation, or test partition. This prevents cross-leakage which would produce optimistic results.

Chromosome data augmentation was only applied to the training data. One augmented image is created per original training sample (including straightened images) using rotations selected in increments of 5 in ranges of -15 to 15 degrees, excluding 0 degrees. Additionally, translations of up to 4% of the image size are added.

Splitting could not be done at patient or case level because the CIR-Net dataset doesn't provide patient, metaphase, or karyotype identifiers. As in D'Angelo and Nanni's study, images originating from the same karyotype may therefore be distributed across different folds, which remains a limitation of the evaluation protocol [1].

C. Model Input Preparation

After straightening, splitting, and augmentation, images got converted to tensors for training. Each image was scaled from the original 8-bit pixel range $[0, 255]$ to $[0, 1]$, which is represented as a single-channel tensor for the custom CNN and ResNet-50

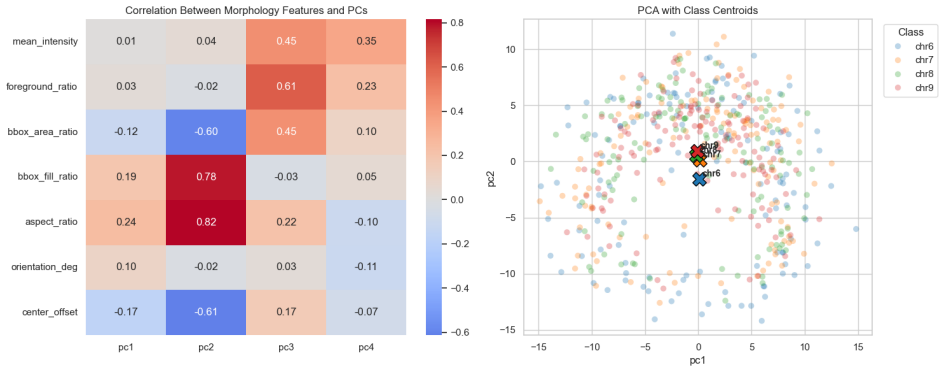


Figure 4. PCA morphology correlation heatmap

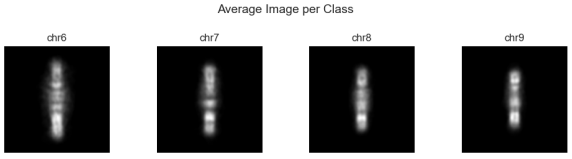


Figure 5. Average image per class before straightening.

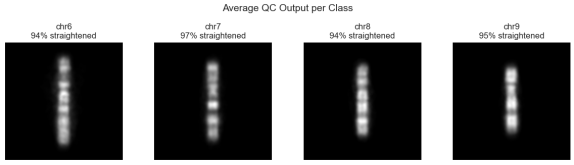


Figure 6. Average image per class after straightening.

experiments. Class labels were encoded as integers corresponding to the chromosome number.

For ResNet-50, the pretrained architecture was adapted to accept grayscale input. The first convolutional layer expects three-channel RGB images, so its pretrained weights were averaged across the RGB channel dimension to represent a single-channel convolution. For DINOv2, the images were repeated across three channels to match the expected input. Then, images were normalized using the ImageNet mean and standard deviation because the pretrained model expects inputs with the same channel-wise scaling used during pretraining.

VI. Modeling

Three models were compared. A compact CNN trained from scratch, ResNet-50, a pretrained ImageNet model, and DINOv2, a fully fine-tuned ViT-S/14 model.

A. CNN

The purpose of this model was to test how well information could be learned directly without the help of pretrained visual features. This helps identify if using transfer learning offered a clear advantage. Transfer learning is often beneficial with limited training data, but this is a constrained domain. Images contain single grayscale chromosomes on a simple background, with discriminative cues being banding patterns, size, shape, and foreground structure. Because of this, a custom CNN can learn task-specific features that compete with pretrained models.

Four convolutional blocks with channel widths of 32, 64, 128, and 256 were used. Two 3×3 convolutional layers with padding were included in each block, followed by batch normalization and ReLU activation. A max-pooling layer was applied after each block to reduce spatial resolution while increasing feature depth across the network. Adaptive average pooling reduced the feature map to a single vector after the last convolutional block. Finally, dropout and a fully connected layer were applied for four-class classification.

Cross-entropy loss and the AdamW optimizer with a learning rate of 1×10^{-3} and weight decay of 1×10^{-4} were used to train the model. With a maximum of 50 epochs and an early stopping patience of 8 epochs, the batch size was 32. Validation macro-F1 was used to pick the model, and the held-out test fold was used to assess the optimal validation checkpoint.

B. ResNet-50

As a robust convolutional transfer-learning baseline, ResNet-50 was chosen. ResNet architectures are now common backbones for image recognition tasks [13], and residual networks leverage shortcut connections to make deep convolutional models easier to optimize. In this effort, the pretrained ResNet examines whether, following grayscale modification and fine-tuning, general natural-image properties can transfer to G-band chromosomal morphology.

ImageNet pretrained weights were used to initialize the model. The first convolution was modified by averaging the pretrained RGB filters over the channel

Algorithm 1 QC-gated projection-arm chromosome straightening

Require: Grayscale chromosome image I, foreground threshold τ
Ensure: Straightened image I_s or original image I
Create foreground mask $M \leftarrow I > \tau$
Remove isolated foreground noise from M
Compute foreground area A
if A is too small **then**
    return I ▷ insufficient foreground
end if
Estimate dominant foreground orientation θ
Generate candidate global rotations around θ
$bestSplit \leftarrow \emptyset$
for each candidate rotation r **do**
    Rotate image and mask by r
    Crop rotated image to foreground bounding box
    Compute row-wise foreground projection
    Smooth the projection profile
    Search for a bend row b with low projection width and balanced arms
    if b gives the best projection score **then**
        Store rotated image, mask, bend row, and projection score
    end if
end for
if no stable bend row is found **then**
    return I ▷ unstable projection split
end if
Split foreground mask into upper arm M_u and lower arm M_l using bend row b
if either arm has too little foreground **then**
    return I ▷ unstable arm split
end if
for each arm $a \in \{upper, lower\}$ **do**
    Generate candidate arm rotations from -90° to 90°
    for each candidate arm rotation q **do**
        Rotate arm image and mask by q
        Crop to arm foreground
        Compute vertical projection width
    end for
    Select arm rotation with minimum projection-width score
end for
Align upper and lower arms at the bend marker
Combine arms into a single reconstructed chromosome strip
Centre reconstructed strip on a blank canvas with the original image size
Compute output quality metrics:
    foreground retention, component count, continuity, aspect ratio, row-width spike, arm balance
if output fails any quality-control check **then**
    return I ▷ fallback to raw image
else
    return I_s ▷ accepted straightened image
end if

dimension and using the result as a one-channel convolution because the chromosomal images are single-channel grayscale images. The last fully connected layer was swapped out for a new classifier that matched the chromosomal classes.

Cross-entropy loss, AdamW optimization, weight decay of 10^{-4}, cosine learning-rate scheduling, a learning rate of 10^{-4}, batch size of 16, and early stopping on validation macro-F1 with patience of 10 epochs have all been used to fine-tune the model after experimenting with different values.

C. DINOv2

As an exploratory pretrained vision-transformer baseline, DINOv2 was used. To the best of our knowledge, the examined karyotyping literature did not identify DINOv2 as a baseline for G-banded chromosomal classification. Custom CNNs, ResNet-style convolutional models, Inception-based networks, or task-specific chromosomal topology models are more commonly used in modern automated karyotyping studies. Therefore, adding DINOv2 examines the ability of a modern self-supervised foundation vision model to transfer to chromosome morphology without being explicitly pretrained on cytogenetic images.

Since chromosomal classification relies on visual signals including global shape, centromere position, relative arm length, and local G-band texture, this decision makes sense. DINOv2's patch-based transformer architecture can model both local image regions and longer-range spatial relationships, and it is trained via self-supervised learning to generate general-purpose visual representations. For chromosomes, where the class signal is dispersed throughout the entire item rather than concentrated in a single local feature, these characteristics might be helpful. Furthermore, recent bioimage research suggests that DINOv2-style representations may be helpful outside of natural-image domains [14], and DINOv2 has demonstrated strong transfer ability across downstream vision tasks [15].

In order to match the expected DINOv2 input format, the grayscale images were replicated across three channels and normalized using ImageNet statistics. The backbone was reinforced with a lightweight classification head that included layer normalization, a hidden linear layer, GELU activation, dropout, and a final linear classifier. AdamW was used for fine-tuning, weight decay of 10^{-4}, gradient clipping, and label smoothing of 0.05 were also used. To prevent harmful updates to the pretrained representation, the classifier head employed a learning rate of 10^{-3}, while the backbone used a lower learning rate of 10^{-5}. Eight batches were employed for training, with a maximum of sixty epochs and early stopping on validation macro-F1 with a patience of twelve epochs.

Occasional collapse-like behavior during five-fold cross-validation, where the model predicted only one class or obtained validation macro-F1 at the single-class baseline for a fold, was a weakness observed during development. This was most apparent in unstable folds where validation loss remained high and validation macro-F1 either remained close to 0.10 or abruptly declined after first improving. This instability is probably caused by fine-tuning a large pretrained transformer on a tiny chromosomal dataset, where the pretrained representation may be disrupted by early backbone updates before the randomly initialized classification head has learned a stable decision boundary.

A five-epoch head warm-up was implemented to lower this risk. The DINOv2 backbone was frozen throughout this warm-up, and only the classification head was trained; all backbone layers were unfrozen after the warm-up. In order to give the model more time to recover from unstable early epochs, DINOv2 was also given a longer early-stopping patience than the CNN and ResNet baselines. Longer patience windows were not rigorously assessed, however development runs indicated that this behavior can recover with more epochs. Consequently, rather than being a fully optimized hyperparameter setup, the DINOv2 training configuration should be viewed as an useful stabilization technique.

VII. Evaluation and Results

A. Evaluation Protocol

Five-fold stratified cross-validation was used for the primary model comparison. For early stopping and model selection, a validation set was generated from the remaining data in each fold while a test partition was held out. The test fold was used to calculate the final test measurements. The mean and standard deviation for each of the five folds are presented in the results. Accuracy was used to measure overall correctness, and macro-F1 was used to give equal weight to each chromosome class. Although the four-class dataset was balanced, macro-F1 remains useful because it reveals whether a model performs consistently across classes rather than relying on strong performance for only some labels.

B. Overall Model Performance

1) Raw images: The results of the five-fold cross-validation are displayed in Table I. All three models performed well, with mean macro-F1 values more than 0.90. At 0.9326 accuracy and 0.9323 macro-F1, the custom CNN had the highest mean test accuracy and macro-F1. ResNet-50 had the lowest standard deviation across folds, but it was somewhat lower with 0.9229 accuracy and 0.9232 macro-F1. DINOv2 had the most fold-to-fold variance, achieving 0.8999 accuracy and 0.9012 macro-F1.

Importantly, even though it had the highest mean score, the custom CNN was more unstable across folds. The most consistent results were achieved by ResNet-50, showing that the backbone transferred to the chromosomal images reliably. Due to the collapsed fold, DINOv2 was less stable overall despite being competitive in several folds. This could be because it was challenging to fine-tune a larger self-supervised transformer model on a tiny four-class dataset, or it needed more time to train.

One collapsed fold had a significant impact on DINOv2's mean result. Only 0.7308 accuracy and 0.7369 macro-F1 were achieved in fold 2, whereas the other four had accuracy and macro-F1 above 0.93. As a sensitivity check, removing this fold would raise the mean accuracy of DINOv2 from 0.8999 to 0.9422 and macro-F1 from 0.9012 to 0.9422, while the standard deviation would decrease from roughly 0.095 to 0.014. Nonetheless, these results were kept to show the unstable nature on the model on this data.

The models got confused in different areas, shown by the confusion matrices in Figure 7. These patterns align with the EDA, which revealed overlapping morphology and only slight variations in foreground area and intensity among the chosen chromosomal classes.

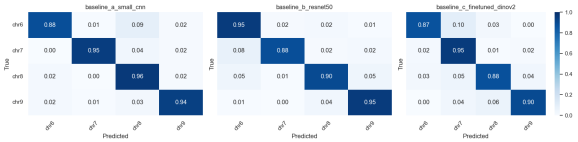


Figure 7. Mean normalized confusion matrices for raw images with train-only CDA.

2) Raw and Straightened Images: This run adds straightened images plus augmentation to the total data. This gives the models more data to train on, while also giving them more testing data. Straightened images can be included in validation and test sets, but without augmentation. Only straightened chromosomes that passed quality control are added to avoid duplicates.

Compared to the raw-only run, all models showed improved performance. The custom CNN reached a mean test accuracy of 94.66% and macro-F1 of 94.67%. ResNet-50 achieved 96.15% accuracy and 96.14% macro-F1, while DINOv2 achieved the highest mean score with 96.34% accuracy and 96.33% macro-F1. ResNet-50 remained as the most stable model.

DINOv2 showed the largest improvement, showing the best results. It still showed initial collapse but recovered as epochs progressed, reaching optimal performance. This confirms the initial claim of this model performing better when more data is intro-

duced. The training curves for raw and straighened data are shown in Figures 8 and 9, while results for both runs are compared in Table **??**.

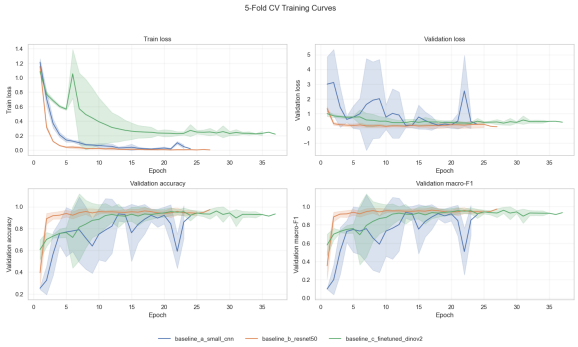


Figure 8. Training curves for the raw and straightened images experiment.

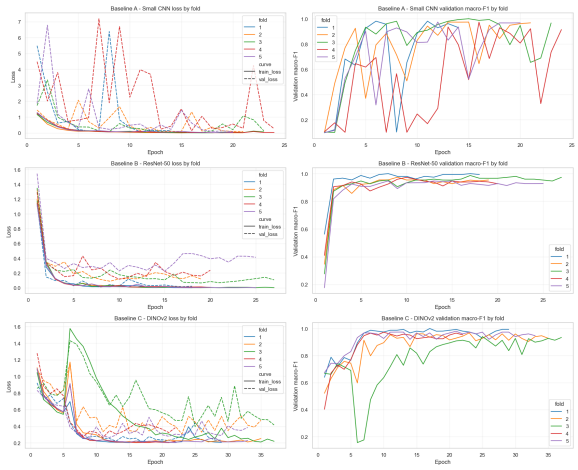


Figure 9. Loss and validation macro-F1 curves for the raw and straightened images experiment.

Table I
FIVE-FOLD CROSS-VALIDATION RESULTS FOR RAW-ONLY AND RAW+STRAIGHTENED EXPERIMENTS.

Model	Input Setting	Accuracy	Macro-F1
Custom CNN	Raw	0.9326	0.9323
Custom CNN	Raw + Straightened	0.9466	0.9467
ResNet-50	Raw	0.9229	0.9232
ResNet-50	Raw + Straightened	0.9615	0.9614
DINOv2	Raw	0.8999	0.9012
DINOv2	Raw + Straightened	0.9634	0.9633

C. Training Behaviour

For each of the three models, the training curves had unique optimization behavior. ResNet-50 rapidly converged, with validation accuracy and macro-F1 rising quickly and training loss greatly dropping in the early epochs. Its low standard deviation in the final test findings is consistent with its validation curves being robust over folds.

During validation, the custom CNN showed more fold-to-fold variation and improved more gradually. This is to be expected since the model had to learn directly from the small dataset after being trained from random initialization. Nevertheless, it eventually achieved robust validation and test performance, demonstrating that there is sufficient visual structure in the data for a smaller CNN to lean meaningful representations.

The most unpredictable training behavior was displayed by DINOv2. Its curves displayed greater unpredictability, and one fold resulted in much worse test performance even if it attained superior validation performance in several folds.

Both raw image and raw with straightened image runs showed similar learning patterns, so only graphs for raw data are shown.



Figure 10. Training curves for the raw-image CDA experiment.

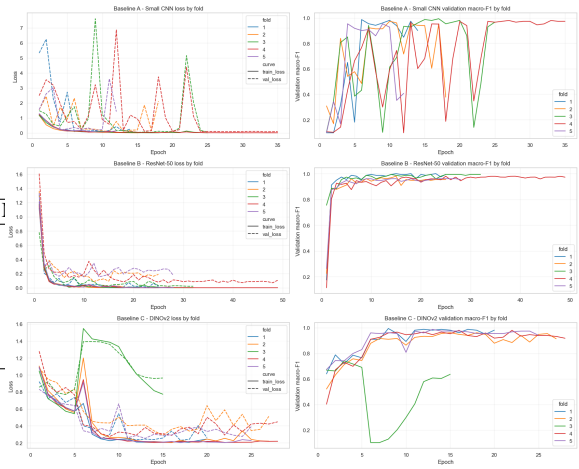


Figure 11. Loss and validation macro-F1 curves for the raw-image CDA experiment.

D. Visual Error and Attention Analysis

Qualitative visual diagnostics were created for first fold runs in order to gain a deeper understanding of model behavior. Patch-token similarity to the class token was used to examine DINOv2, whereas Grad-CAM was employed for the bespoke CNN and ResNet-50 models. These visuals made it easier to determine whether the models were concentrating on background or chromosomal areas.

High activation was found on the chromosomal body in the Grad-CAM maps, especially near curved ends, bends, and highly banded regions. This implies that the CNN-based algorithms made predictions based on local banding cues and observable chromosomal shape. The usage of patch-level transformer features is consistent with the DINOv2 patch-similarity maps' wider activation along the chromosomal shape.

The confusion-matrix patterns are consistent with the error instances. A number of chromosomes that were incorrectly classified were compact, curved, or had banding patterns that visually resembled those of nearby classes.

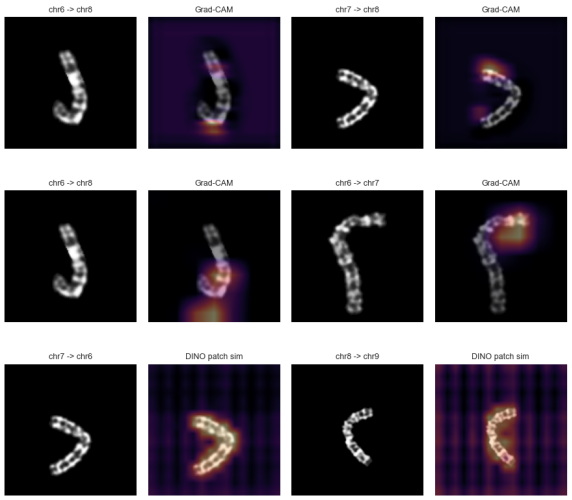


Figure 12. Qualitative attention diagnostics for selected fold-1 examples.

VIII. Ethical Considerations

Dataset representation is the main ethical concern. Isolated chromosomal images are included in the CIR-Net dataset, but patient, metaphase, laboratory, and acquisition metadata are not included. This makes it more difficult to evaluate model generalization across staining conditions, microscope settings, patients, and lab procedures. Performance may be optimistic if images from the same karyotype match visual characteristics across folds because the train-test split is image-level rather than patient-level. External validation on data from different labs and patient-level splitting would be necessary for a quality assessment [16]. This study only picked 4 chromosomes, and its performance shouldn't be directly generalized to a full 24 class classification.

Because chromosome images are derived from human biological samples, privacy and consent are also important [17]. Cytogenetic data is still medically sensitive even if the released information includes cropped images rather than directly identifiable patient records. Consent, anonymization, and institutional approval should be followed.

Lastly, accountability and interpretability are essential [5]. Patch-similarity and Grad-CAM visualizations can be used to examine whether models take chromosomal regions into account, but they do not fully explain model choices. For clinical usage, high accuracy is not enough on its own. Models should be assessed for robustness to unusual morphology, anomalies, and class-specific errors. In this study, the model is viewed as an experimental classifier for automated karyotyping research, not as a clinically deployable diagnostic system.

IX. Conclusion and Future Works

This project explored deep learning, transfer learning and vision transformers in chromosome image classification. Chromosomes 6, 7, 8 and 9 were used to create a task with enough samples per class while maintaining visual similarity between classes. Exploratory data analysis showed measurable morphological differences between the selected chromosomes but PCA diagnostics showed that the classes were not linearly separable in flattened pixel space.

The experiments compared a custom CNN trained from scratch, a ResNet-50 pretrained on ImageNet, and a DINOv2 based model. Two pre-processing settings were tested: raw images with augmentation only on the training set, and raw and straightened images with augmentation. The custom CNN performed best on average in the raw-image setting, showing that smaller models can still learn effectively. All architectures were stable, with ResNet-50 showing the lowest variances of accuracy and macro-F1 across folds. DINOv2 showed the best performance when straightened images were introduced but was more unstable, with one collapsed fold in raw images and one recovered fold in total images. This shows that fine-tuning a larger transformer-style model is more sensitive when data is limited.

The absence of patient, metaphase, or karyotype identifiers in the CIR-Net dataset is a drawback of this study. Because evaluation was done at the image level, the stated performance may be optimistic if images from the same original karyotype occur in different folds. Therefore, rather than being clinically

validated systems, the models should be interpreted as experimental classifiers.

If appropriate metadata becomes available, future research should evaluate the models using splits at the patient or karyotype levels. Generalization should be tested using external validation on data from various labs, staining conditions, and acquisition settings. Calibration, top-k accuracy, uncertainty estimation, and failure-case detection could be further investigated. Lastly, implementing modern deep learning techniques may improve the straightening pipeline.

REFERENCES

[1] M. D'Angelo and L. Nanni, "Deep-learning-based human chromosome classification: Data augmentation and ensemble," *Information*, vol. 14, no. 7, 2023. [Online]. Available: https://www.mdpi.com/2078-2489/14/7/389

[2] 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 (CVPR) Workshops*, July 2017.

[3] J. Li, B. Chen, X. Sun, T. Feng, and Y. Zhang, "Banded chromosome images recognition based on dense convolutional network with segmental recalibration," *Sheng Wu Yi Xue Gong Cheng Xue Za Zhi*, vol. 38, no. 1, pp. 122–130, Feb. 2021.

[4] A. Chebly, "Cytogenetics in the genomics era: why karyotyping still matters," *BMC Medical Genomics*, vol. 19, no. 1, p. 10, 2026. [Online]. Available: https://doi.org/10.1186/s12920-025-02296-0

[5] L. S. Rosenblum, J. Holmes, and A. F. Taghiyev, "The emergence of artificial intelligence-guided karyotyping: A review and reflection," *Genes*, vol. 16, no. 6, 2025. [Online]. Available: https://www.mdpi.com/2073-4425/16/6/685

[6] L. J. van Vliet, I. T. Young, and B. H. Mayall, "The athena semi-automated karyotyping system," *Cytometry*, vol. 11, no. 1, pp. 51–58, 1990.

[7] B. LERNER, H. GUTERMAN, I. DINSTEIN, and Y. ROMEM, "Human chromosome classification using multilayer perceptron neural network," *International Journal of Neural Systems*, vol. 06, no. 03, pp. 359–370, 1995. [Online]. Available: https://doi.org/10.1142/S012906579500024X

[8] 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. [Online]. Available: https://www.sciencedirect.com/science/article/pii/003132039500042X

[9] C. Lin, G. Zhao, Z. Yang, A. Yin, X. Wang, L. Guo, H. Chen, Z. Ma, L. Zhao, H. Luo, T. Wang, B. Ding, X. Pang, and Q. Chen, "Cir-net: Automatic classification of human chromosome based on inception-resnet architecture," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 19, no. 3, pp. 1285–1293, 2022.

[10] Y. Qin, J. Wen, H. Zheng, X. Huang, J. Yang, N. Song, Y.-M. Zhu, L. Wu, and G.-Z. Yang, "Varifocal-net: A chromosome classification approach using deep convolutional networks," *IEEE Transactions on Medical Imaging*, vol. 38, no. 11, pp. 2569–2581, 2019.

[11] T. Chen, C. Xie, W. Zhang, Y. Li, W. Ke, T. Li, X. Huang, and K. Li, "Imagenet pre-training and two-step transfer learning in chromosome image classification," *Scientific Reports*, vol. 16, no. 1, p. 7572, 2026. [Online]. Available: https://doi.org/10.1038/s41598-026-38662-w

[12] 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. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167865508000342

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[14] T. Moutakanni, C. Couprie, S. Yi, M. Doron, Z. S. Chen, N. Moshkov, E. Gardes, M. Caron, H. Touvron, A. Joulin, P. Bojanowski, W. M. Pernice, and J. C. Caicedo, "Cell-dino: Self-supervised image-based embeddings for cell fluorescent microscopy," *PLoS Computational Biology*, vol. 21, no. 12, p. e1013828, 2025.

[15] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "Dinov2: Learning robust visual features without supervision," 2024. [Online]. Available: https://arxiv.org/abs/2304.07193

[16] Code of Federal Regulations, "42 CFR § 493.1276 – Standard: Clinical cytogenetics," https://www.law.cornell.edu/cfr/text/42/493.1276, 2026, accessed: 2026-05-04.

[17] S. Hallaj, A. Heinke, F. G. P. Kalaw, N. Gim, M. Blazes, J. Owen, E. Dysinger, E. S. Benton, B. A. Cordier, N. G. Evans, J. Li-Pook-Than, M. P. Snyder, C. Nebeker, L. M. Zangwill, S. L. Baxter, S. McWeeney, C. S. Lee, A. Y. Lee, B. Patel, and AI-READI Consortium, "Navigating open data sharing and privacy in the age of clinical ai research: from reidentification to pseudo-reidentification," *EClinicalMedicine*, vol. 91, p. 103729, 2025.