Journal of Innovative Image Processing (ISSN: 2582-4252)
www.irojournals.com/iroiip/

A Hybrid Vision Transformer Model for Leukemia Image Classification

Revathi B.[1*], Kaliappan M.[2], Anandhi S.V.[3], Kezial Elizabeth S.K.[4]

[1,2,3]Artificial Intelligence and Data Science, Ramco Institute of Technology, Virudhunagar, India.
[4]Computer Science and Engineering, Mangayarkarasi College of Engineering, Madurai, India.

E-mail: [1*]revas85@gmail.com, [2]kalsrajan@yahoo.co.in, [3]svanandhi2020@gmail.com, [4]kezialcse@gmail.com

Abstract

This work presents an automated method for distinguishing leukemic blast cells from normal bone marrow cells in microscopic blood smear images. The approach uses a Vision Transformer (ViT) as the main feature extractor, combined with a Kolmogorov–Arnold Network (KAN) representation and an XGBoost classifier for final classification. Typically, a multilayer perceptron (MLP) head is used to classify images in most Vision Transformer architectures. In this study, we explore the potential improvement of transformer feature discernibility for leukemia detection by replacing the conventional MLP representation with a KAN-based transformation. The pre-trained Vision Transformer processes CLS-token embeddings extracted from the images. Subsequently, these embeddings are classified using an XGBoost model and evaluated through the KAN representation. The proposed ViT–KAN–XGBoost model achieved an accuracy of 85.11% on the C-NMC 2019 dataset, which is higher than the 83.22% obtained with the ViT–MLP–XGBoost model. This suggests that adding KAN-based representations to transformer features can improve classification performance and may be useful for leukemia screening tasks.

Keywords: Classification, Vision Transformer, Kolmogorov Arnold Network, Multilayer Perceptron, XGBoost, Microscopic Images.

1. Introduction

Leukemia refers to a group of blood cancers originating from hematopoietic stem cells in the bone marrow and is considered one of the main causes of cancer-related deaths globally. It is medically categorized into four primary types: acute myeloid leukemia [19] (AML), acute lymphoblastic leukemia (ALL), chronic lymphocytic leukemia (CLL), and chronic myeloid leukemia (CML), all of which are generally aggressive and especially common in children. In a healthy body, the production of red blood cells, white blood cells, and platelets remains in balance; any disturbance can result in the unchecked growth of immature white blood cells, which hinders the normal creation of blood cells and creates serious risks to the body's overall health. Figure 1 presents a comparison of normal blood cells with leukemic blood cells. Figure 1(a) shows a normal blood cell image [21], while 1(b) displays an ALL blood cell image. At present, the standard diagnostic method includes examining stained peripheral blood smears and bone marrow samples using a microscope. Although it has diagnostic benefits this method

*** Corresponding Author**

DOI: https://doi.org/10.36548/jiip.2026.2.008
Received: 27.02.2026, received in revised form: 01.04.2026, accepted: 19.04.2026, published: 30.04.2026

is lengthy and prone to significant differences in interpretation between observers, frequently postponing important treatment choices.

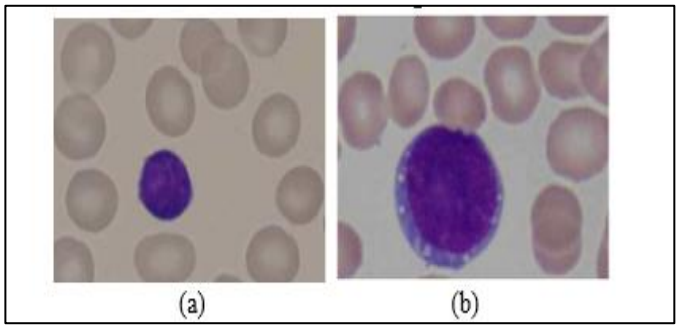


Figure 1. a) Normal Blood Cell Image b) ALL Blood Cell Image

In recent times, it has been proposed to apply machine learning techniques such as CNNs to reduce the time spent on screening. However, CNN models are typically unable to identify long-range spatial dependencies in complex cell configurations, owing to the intrinsic limitations of their receptive fields. Moreover, CNNs usually require a significant amount of labeled samples to function properly in diverse conditions, which poses a problem due to the highly specialized nature of medical imaging. Our proposed framework consists of combining three cutting-edge technologies in order to overcome these issues. Specifically, a vision transformer is used as the core component for global self-attention between patches. As the final stage, we replace the usual MLP head with a Kolmogorov Arnold Network (KAN), which allows for nonlinear transformations while retaining interpretability [6]. Lastly, an ensemble of XGBoost classifiers is applied on top of the CLS-token embeddings. Several data augmentation techniques and regularization methods are employed to ensure robust performance in extremely limited datasets. Evaluation was done using the C-NMC 2019 dataset. However, our proposed approach consistently performs better than the conventional model that uses MLP or different KAN heads alone when compared with regard to precision, recall, and F1-scores without slowing down the speed required for real-time medical applications. During practical applications of the proposed model in clinics, variation is possible among images of blood smears owing to differences in hardware, imaging processes, and staining. Changes like these can induce shifts in distribution that make it tougher to apply models that were trained on only one dataset. This work employed the standardized C-NMC 2019 dataset for controlled experimentation and assessment.

2. Related Work

Several authors looked at the MLP using deep learning methods in this survey and suggested it. A simple blood test that shows an abnormal number of blood cells is often how leukemia is diagnosed. After leukemia is found, a doctor can look at the cell structure by taking samples of blood and bone marrow. Campo, E. et al. [1] wrote about a WHO Classification of Tumors of Hematopoietic and Lymphoid Tissues in the second volume of the 4th Edition of the WHO series on the histological and genetic typing of human tumors. This short, trustworthy reference book sets a standard for pathologists and oncologists around the world. It will be an important tool for planning studies that look at how well treatments work and what the clinical outcomes are. Goda, C., et al. did single-cell RNA sequencing on preleukemic BMM (pBMM) cells. They found that there were fewer endothelial cells and normal HSC (nHSC)-regulating $LepR^{+}$ mesenchymal stem cells but more $CD55^{+}$ fibroblasts and pericytes [2]. Preleukemic

CD55+ fibroblasts had higher proliferation rates and lower collagen expression, which suggests that the extracellular matrix changes during the development of leukemia. Chen, T. et al.'s [7] scalable end-to-end tree boosting method, XGBoost, is often used by data scientists to get the best results on a wide range of machine learning tasks. It also makes the weighted quantile sketch for approximation tree learning and the new algorithm for sparse data that is aware of sparsity. Shah, W. H., et al. [9] propose a novel categorization method that involves analyzing the morphologies of all cells and deriving their topological properties. We use persistent homology to obtain these topological properties. Our method can accurately and quickly find and classify leukemia blast cells with an F1 score of 94.6% and a recall of 98.2%. This technique could significantly enhance the identification and management of leukemia. Hegde, R. B., et al. [10] gave an overview of the methods used to process images of peripheral blood smears. We have categorized these methods into three groups based on factors like platelet, RBC, and WBC analysis. [11] Cho, P., et al. checked to see if the immature white blood cell blasts in bone marrow have the right shape and size, including the size and shape of their nuclei. Pathologists also use immune phenotyping with multi-channel flow cytometry to find out if certain antigens are on the surface of blast cells to determine the cell lineage of acute lymphoblastic leukemia. These manual tasks take a lot of time and money because they require skilled workers and medical experts. Rao.P. et al. [12] intended to juxtapose the results with their CNN counterpart while utilizing a model that incorporates transformers in its architecture. Zhang, X., and others [13] noted that adding clinical factors to a prediction model with radiomic data improves cervical LNM prediction. This allows PTC patients to receive more accurate and personalized treatment plans. The nomogram provides clinicians with more useful information potential outcomes. Raju, A. et al. [14] discussed the problems with unbalanced datasets and the need for advanced feature extraction in medical image analysis. Initially, the CKHK-22 dataset had 24 classes. However, when we reduced it to 14 classes, the data became better and more balanced. This improvement allowed for better results in categorization and more accurate feature extraction. Gupta, R. et al. [15] provided a detailed description of this dataset and its issues. We also present benchmarking statistics for every method that has been used on this dataset so far. Prasad, P., et al. [16] investigated the use of Vision Transformers (ViT) to automatically identify ALL in microscopic blood smear images. The Vision Transformer model achieved an accuracy of 98.01%, with its precision, recall, and F1-score consistently at 98.00%. Hornik, K., et al. [17] demonstrated that standard multilayer feedforward networks with as few as one hidden layer and any squashing function can approximate any Borel measurable function from one finite-dimensional space to another with any desired level of accuracy. Thus, multilayer feedforward networks serve as a type of universal approximator. Maruf, M., et al. [20] presented an advanced deep learning-based architecture that can automatically diagnose ALL from bone marrow smear images without the need for human interpretation. The proposed approach includes a robust image preparation pipeline that enhances the quality of bone marrow smear images for optimal input into a convolutional neural network (CNN). This significantly accelerates computations and improves diagnostic accuracy.

3. Proposed Work

3.1 Proposed Framework

In this section, there are four major stages in our proposed pipeline. Data pre-processing, feature extraction via Vision Transformer with interchangeable MLP and KAN heads, and XGBoost classification. By holding the ViT backbone and all hyperparameters

constant, we isolate the representational impact of the head architecture (MLP vs. KAN) on the final leukemia detection performance. The framework includes three parts: a pre-trained ViT backbone (feature extractor), a representation head (such as an MLP or KAN proxy), and an XGBoost classifier. By maintaining the same ViT backbone and XGBoost settings throughout all experiments, this study focuses on the effect of the representation head. As a result, the differences in performance between the configurations mainly stem from substituting the MLP mapping with a KAN-style mapping applied to CLS embeddings. In our proposed framework, we utilize both an MLP head and a KAN head as classification heads, with a frozen Vision Transformer (ViT) backbone to investigate how each head architecture influences downstream leukemia image classification, as depicted in Figure 2. First, images from the C-NMC 2019 dataset are resized to 224 × 224 pixels, normalized with ImageNet statistics, and followed by feature extraction, then stratified into training (80%) and validation (20%) splits for XGBoost training.

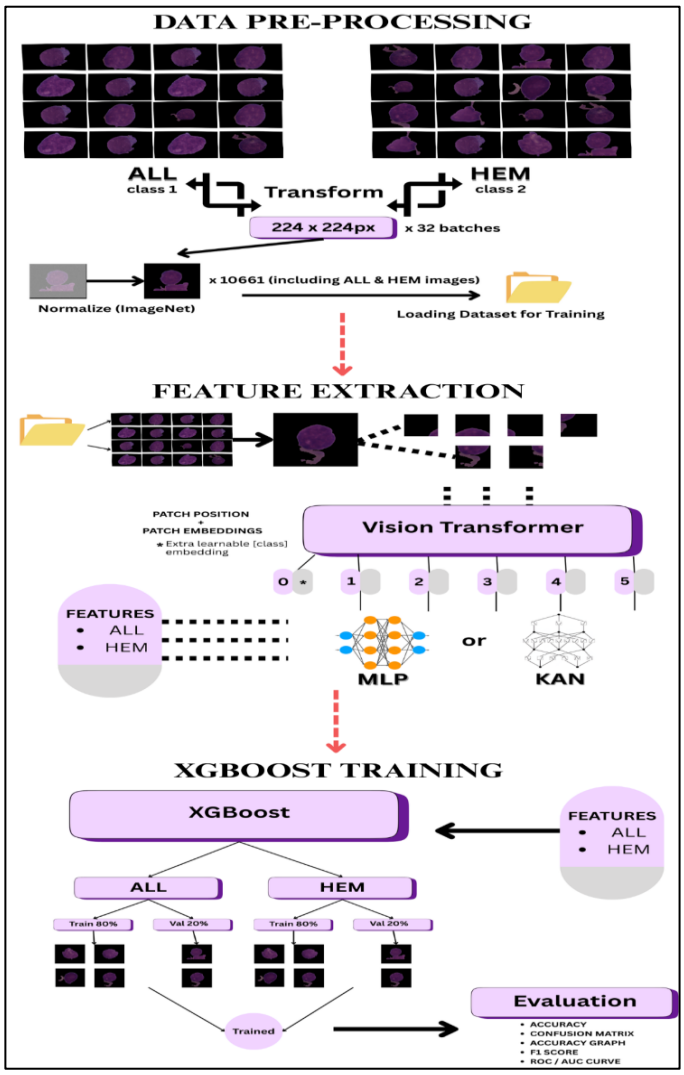


Figure 2. Proposed Framework for ViT - MLP/KAN - XGBoost Classification

The experiments used the C-NMC 2019 dataset, which has 10,661 images (7,272 for ALL and 3,389 for HEM). We used a stratified 80:20 split to obtain about 8,529 training images and about 2,132 test images that we didn't use to train the model. We also took a small part of the training set to use as a validation set to monitor while we trained. The experiments that were submitted did not use k-fold cross-validation. We used data augmentation and random mini-

batch sampling to ensure that both classes were represented in each training batch. This way, we avoided bias from having an uneven number of classes. The above representation shows how to use ViT-KAN/MLP-XGBoost classification to identify leukemia. There are only three parts to the figure, but let's add an extra part to demonstrate how the neural networks work. To mitigate any possible spatial bias that might arise from having leukemic cells in the middle of the C-NMC 2019 dataset, counter-cropping augmentation was added to the training pipeline. During training, random-sized cropping and spatial shift-based cropping strategies were used to ensure that the cells were not always in the center of the images. Because of this, the model had to rely on features based on morphology and texture instead of positional priors. The augmentation pipeline employed random spatial cropping with scale changes, controlled aspect-ratio distortion, and flipping operations to simulate natural variations in the images. It's important to remember that these changes only occurred in the training data during the feature extraction stage. To ensure a fair evaluation, the validation data remained unchanged. The Vision Transformer obtains global contextual representations, so adding spatial perturbations makes the model more resistant to biases specific to the dataset and improves positional invariance. This method reduces the likelihood that the model will fit too closely to centered cell structures. It also helps the XGBoost classifier generalize more effectively. The ViT backbone and XGBoost classifier were the same in all of the setups that were compared. The only difference was the representation head (MLP vs. KAN), so the reported difference in performance is primarily due to the representation mapping stage and not the classifier itself.

3.2 Data Pre-Processing

The first step in data preprocessing is to resize all of the input images to 224 x 224 pixels. Make sure the patch embedding needs of the ViT are met. Before being sent to the ViT backbone, all input images (originally 128×128) were resized to 224×224. We made this choice to match the input resolution used during ViT pretraining (ImageNet) and to keep the ViT patch-embedding and positional encoding design without changing the architecture. We used bicubic interpolation to enlarge the images. Even though upscaling can make fine details somewhat smoother, matching the pretrained input size always made the model more stable and improved the downstream classifier's performance in our initial tuning. We used two resolution strategies to change the input: (A) upscaling native 128×128 images to 224×224 (bicubic interpolation) to match ViT pretraining, and (B) center-cropping to 128×128 to maintain the native resolution. Strategy (A) use pretrained ViT weights directly, while strategy (B) tests whether keeping the original pixel resolution improves performance. Then we use the standard ImageNet mean and standard deviation values ([0.485, 0.456, 0.406] and [0.229, 0.224, 0.225], respectively) to normalize the data. This ensure that the data is consistent and evenly distributed. The complete dataset of 10,661 images, evenly split between the "ALL" and "HEM" classes, is divided into 8,529 training samples and 2,132 validation samples through stratified sampling, which maintains the original class ratio. For efficient batch processing, a PyTorch DataLoader with a batch size of 32 and four worker threads is used. Shuffling is turned off during feature extraction to keep the order the same. Table 1 below shows how the patch embedding process works.

Table 1. Patch Embedding Requirements

Parameters	Values
Patch size	16 x 16
Number of patches	196
Flattening	196 x 768
Positional Encoding	197 x 768

3.3 Feature Extraction

3.3.1 Vision Transformer with Multilayer Perceptron

Vision Transformers (ViTs) have demonstrated state-of-the-art performance by recasting an image into a sequence of fixed-size patches and applying standard Transformer encoders to capture long-range dependencies. In this work, we use a pretrained vit_base_patch16_224 backbones [4,8] freezing all its layers and explicitly replacing its default classification head with a Multilayer Perceptron (MLP) so that we can rigorously compare MLP vs. KAN heads under identical conditions. One popular type of neural network is the Multilayer Perceptron (MLP), also known as the Universal Approximation Theorem. An MLP is comprised of several stages: an input layer, one or more hidden layers, and an output layer. Each layer contains a set of computational units known as neurons that compute weighted sums of their inputs plus biases before applying a nonlinear activation function, establishing the MLP's ability to approximate arbitrary continuous mappings [18]. Formally, given an input vector,

$x = \{x_1, x_2, x_3 \ldots . x_n\}$ and output vector

$y = \{y_1, y_2, y_3 \ldots . y_n\}$. The total number of parameters in an MLP with N_h hidden neurons can be expressed as equation 1,

$$n \cdot h_{1\_input \to hidden_1} + \sum_{k=1}^{N_h - 1} . (h_k \cdot h_{k+1}) + h_{N_h} \cdot m_{\_hidden_{N_h} \to output} \quad (1)$$

Where, h_k is the number of neurons in the k_{th} hidden layer. In the middle panel of the methodology, feature extraction begins by forwarding each pre-processed image batch through a standard vit_base_patch16_224 backbones, yielding per-image token outputs as equation 2:

$$Z \in R^{B \times 197 \times 768} \quad (2)$$

Figure 3 below shows the MLP with a hidden layer. As, mentioned in the embeddings, to project these raw transformer embeddings into a form more amenable to XGBoost, we use the ViT's default classifier head (Multilayer Perceptron layer):

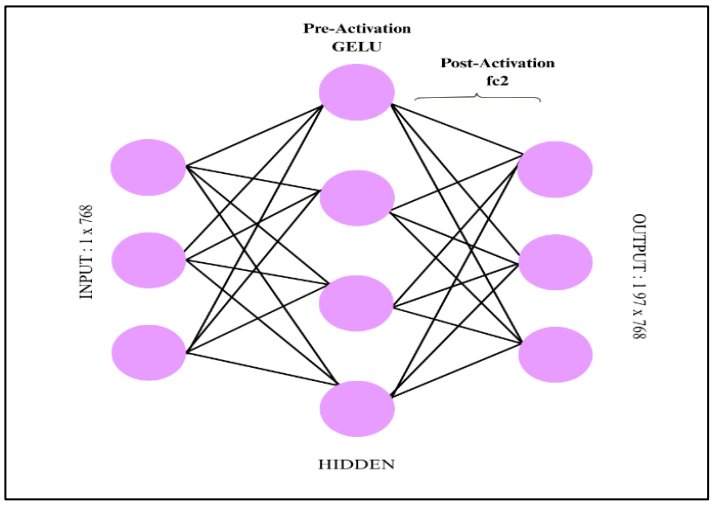


Figure 3. Schematic Representation of MLP Layer in Vision Transformer Feature Extraction with 2 Hidden Layers

3.3.1.1 Input Layer

The input layer receives raw data. The input layer will change the dimension of the image into 1×768, and it simply moves the data to the next layer. In image classification, every neuron in the input layer represents a single pixel.

3.3.1.2 Hidden Layer

Each of the pre-activation values in $h^{(1)}$ is passed through the RELU activation function, this refers to the Rectifier Linear Unit (RELU), a smooth, non-linear activation function commonly used in transformer architectures, including the original ViT. The RELU function is defined in equation 3:

$$ReLU(x) = (0, x) \tag{3}$$

This helps ViT handle subtle variations in embedding values more gracefully.

3.3.1.3 Output Layer

The final stage of our MLP head is a linear projection that maps the 512-dimensional activated hidden vector back into the original 768-dimensional transformer embedding space. In code, this corresponds to self.fc2 = nn.Linear(512, 768), which implements the affine transformation for all the images. So, finally When applied to the CLS token specifically, this produces the final per-image feature vector. These 768-dimensional CLS embeddings are then passed directly to the XGBoost classifier for leukemia vs. healthy prediction.

3.4 XGBoost Training

After extracting the 768-dimensional CLS embeddings from our frozen ViT - MLP head (as depicted in the leftmost panel of Figure 4), we assemble these vectors into a feature matrix and align them with their binary labels ("0" for healthy, "1" for leukemia). We then train an XGBoost classifier on this data, taking advantage of its fast, regularized gradient-boosted tree implementation to learn complex, nonlinear patterns. The embeddings and labels are split into stratified 80% training and 20% validation sets to maintain class balance.

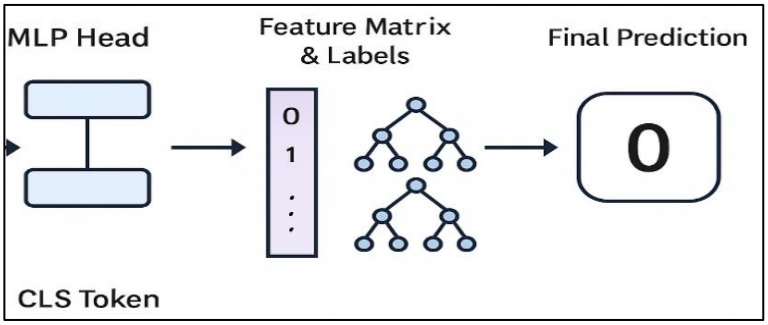


Figure 4. Schematic Representation of the XGBoost Training and Prediction

The above Figure 4 shows that, by passing each image through the frozen ViT backbone and our MLP head, we extract the CLS-token embedding, a 768-dimensional vector that summarizes the image's global features. Our XGBoost model is configured with 100 boosting rounds, a maximum tree depth of 5, and a learning rate of 0.1. We enable early stopping halting

training if the validation loss does not improve for 10 consecutive rounds and adjust for any class imbalance via a weighting parameter. Once trained, the ensemble of decision trees produces the final healthy vs. leukemia predictions, completing the hybrid ViT-MLP-XGBoost pipeline.

Algorithm: ViT–Head (MLP/KAN)–XGBoost Pipeline for Leukemia Classification

Input

Dataset D = $\{(x_{i,}y_i)\}_{i=1}^{N}, where$

$x_{i=blood\ smear\ image}$

$y_i \in \{0,1\}(HEM, ALL)$

Output

Predicted class label $\hat{y}$

Performance metrics (Accuracy, Precision, Recall, F1, ROC-AUC)

Stage 1: Data Preprocessing

1. Resize all images to 224×224
2. Apply normalization using ImageNet statistics
3. Apply data augmentation (training only):
   - Random resized crop
   - Horizontal/vertical flip
   - Spatial shift (counter-cropping)
4. Split dataset:
   - Train: 80%
   - Validation/Test: 20% (stratified)

Stage 2: Feature Extraction using Frozen ViT

5. Load pretrained Vision Transformer:
   ViT=vit_base_patch_224
6. Freeze all tansformer encoder parameters:
   $\Theta_{ViT} = frozen$
7. For each image batch X:
   a. Forward pass through ViT:
      $F = ViT(X) \in \mathbb{R}^{B \times 197 \times 768}$
   b. Extract CLS token:
      $z = F[:,0,:] \in \mathbb{R}^{B \times 768}$

Stage 3: Representation Mapping (MLP or KAN Head)

CASE A: MLP Head

8. Apply MLP transformation

$$h = ReLU(W_1 z + b1), W_1 \in \mathbb{R}^{768 \times 512}$$

$$z' = W_2 h + b_2, W_2 \in \mathbb{R}^{512 \times 768}$$

CASE B: KAN Head(Approximation)

9. Apply KAN inspired transformation:

$$h = \sigma(W_1 z + b_1)$$
$$z' = W_2 h + b_2$$

Where:

- σ approximates univariate nonlinear mappings

- This corresponds to KAN decomposition:

$$f(x) \approx \sum_{q} \Phi_q \left(\sum_{p} \Psi_{pq}(x_p) \right)$$

Stage 4: Feature Aggregation

10. Store Embeddings:

$$X_{feat} = \{z_i'\}_{i=1}^{N}$$

11. Construct feature matrix:

$$X \in \mathbb{R}^{N \times 768}, Y \in \mathbb{R}^{N}$$

Stage 5: XGBoost Classification

12. Split features:
    - Training: 80%
    - Validation: 20% (stratified)
13. Train XGBoost:

- Objective: binary:logistic
- Trees: 100
- Max depth: 5
- Learning rate: 0.1
- Early stopping: 10 rounds
- Class balancing:

$$scale_{pos_{weight}} = \frac{N_{neg}}{N_{pos}}$$

14. Fit Model:

$$\hat{f} = XGBoost(X_{train}, Y_{train})$$

Stage 6 Prediction and Evaluation

15. Predict:

$$\hat{Y} = \hat{f}(X_{test})$$

16. Evaluate:
    - Accuracy
    - Precision
    - Recall(Sensitivity)
    - F1-score
    - ROC-AUC
    - Confusion Matrix

3.5 Feature Extraction

3.5.1 Vision Transformer with Kolmogorov's Arnold Network

Vision Transformers (ViTs) have recently emerged as a powerful architecture for image representation learning by partitioning an image into fixed-size patches and processing them through Transformer encoder blocks to capture long-range dependencies across spatial regions. In this work, the vit_base_patch16_224 backbone, consisting of twelve Transformer encoder layers, is employed as the feature extraction module. For all experiments, the pretrained ViT model is used in evaluation mode as a fixed feature extractor, with all Transformer encoder parameters frozen. Given an input image, the ViT produces a sequence of token embeddings, from which the CLS-token embedding is extracted and used as a compact global representation of the image. No end-to-end fine-tuning of the ViT encoder is performed. This design reduces

computational cost and allows us to isolate the contribution of the downstream learning modules.

3.5.2 KAN Inspired Representation Head

To improve the discriminative capability of the extracted features, the standard multilayer perceptron (MLP) head is replaced with a Kolmogorov–Arnold Network (KAN)-inspired representation module. The hidden dimension of this module is set to 512, providing a balance between representational capacity and parameter efficiency relative to the original embedding dimension of 768. Given that the CLS-token embedding can be expressed as in equation 4:

$$z \in \mathbb{R}^{768} \tag{4}$$

and the transformed representation is computed as in equation 5 and 6:

$$h = ReLU(W_1 z + b1), W_1 \in \mathbb{R}^{768\times512} \tag{5}$$

$$z' = W_2 h + b_2, W_2 \in \mathbb{R}^{512\times768} \tag{6}$$

Where W1, W2 and b1, b2 are learnable parameters.

The KAN architecture is motivated by the Kolmogorov–Arnold representation theorem which states that any continuous multivariate function can be represented as a finite composition of univariate functions and summations [6]. The decomposition is given by equation 7,

$$f(x) \approx \sum_q \Phi_q \left(\sum_p \Psi_{pq}(x_p)\right) \tag{7}$$

The above formulation consists of two sequential and complementary operations: Decomposition refers each input dimension x_p is independently transformed via univariate functions Ψ_{pq} and superimposition represents The transformed components are aggregated through summation and outer functions Φ_q. Let $z \in \mathbb{R}^n$ denote the CLS-token embedding. The KAN-inspired mapping can be interpreted as in equation 8, constructing aggregated scalar projections:

$$Z_q(z) = \sum_{p=1}^{n} \Psi_{q,p}\left(z_p\right) \tag{8}$$

These projections concentrate discriminative information into a lower-dimensional space. The class separability of each projection can be measured using the Fisher discriminant ratio J_q is represented in equation 9:

$$J_q = \frac{(\mu_1^{(q)} - \mu_0^{(q)})^2}{\sigma_1^{2(q)} - \sigma_0^{2(q)}} \tag{9}$$

By learning suitable transformations $\Psi_{q,p}$ the KAN head increases separability in selected projections, making downstream classification more effective. Compared to dense MLPs, this structured transformation reduces the need to learn complex high-dimensional interactions directly.

In classical KAN formulations, univariate mappings are implemented using spline functions, which is explained in equation 10:

$$Spline(x) = \sum_i . c_i B_i(x) \quad (10)$$

Here, $Spline(x)$ represents the Spline function,

c_i-Coefficients and $B_i(x)$ – B-Spline basic functions.

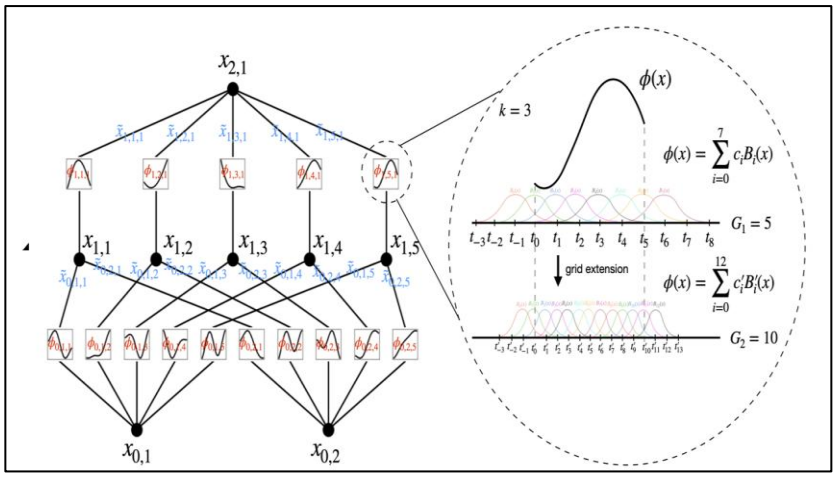


Figure 5. Kolmogorov Arnold Network [6]

Figure 5 depict the core idea behind a Kolmogorov–Arnold Network (KAN) layer, adapted here as the head of our Vision Transformer. In classical KAN theory, any smooth multivariate function $f: [0,1]^n = R$ admits a decomposition.

Two input coordinates $x_{0,1}$ and $x_{0,2}$ are each fed into five distinct spline-based activations (one per output dimension). Each activation produces a one-dimensional response based on its local basis functions $B_i(x)$ and learnable coefficients c_i. The responses across the two inputs are then summed for each of the five output nodes, yielding a five-dimensional output vector. Stacking multiple such KAN layers builds a deep, fully differentiable network akin to an MLP but with far more flexible, edge-wise nonlinearities. In our implementation, the theoretical spline-based transformation is approximated using a compact neural module consisting of a linear projection (768 to 512), followed by a ReLU activation and a second linear projection (512 to 768). Although explicit B-spline basis functions are not instantiated, the combination of learnable linear transformations and element-wise nonlinear activations provides an efficient approximation of adaptive univariate mappings. This representation module operates on the CLS-token embedding produced by the Vision Transformer, enabling the model to combine global contextual information captured by self-attention with adaptive nonlinear feature transformations inspired by the KAN formulation. The resulting feature representation is subsequently used for downstream classification using an XGBoost model.

3.5.2.1 Input Layer

A linear projection lifts the 768-dim token into a 512-dim hidden representation, carrying forward all raw transformer information without nonlinearity.

3.5.2.2 Hidden Layer

We implement the KAN's core univariate nonlinearity approximating univariate feature transformations.

3.5.2.3 Output Layer

A second linear projection returns the activated features to the original embedding size of 768 dimensions, ready to replace the ViT's CLS token. By mapping the general KAN formula into this concise three-layer block linear, ReLU, linear we capture the spirit of Kolmogorov–Arnold superposition while keeping the network fully differentiable and efficient for end-to-end training.

3.6 XGBoost Training

The transformed embeddings z' are aggregated and organized them into a feature matrix, $X \in \mathbb{R}^{N \times 768}$, $Y \in \mathbb{R}^{N}$with corresponding binary labels ("0" for healthy, "1" for leukemia). To learn the final classification boundary, we employ XGBoost's gradient-boosted decision trees, which are well suited to modeling the nonlinear patterns present in these fixed-length embeddings. We perform a stratified 80% and 20% split to form training and validation sets, preserving the class ratios. Our XGBoost Classifier is configured with 100 boosting rounds, a maximum tree depth of 5, and a learning rate of 0.1. We enable early stopping -halting if the validation log-loss does not improve for 10 consecutive rounds and set scale_pos_weight to correct any imbalance between healthy and leukemic samples. After training, the resulting ensemble produces the final healthy vs. leukemia predictions, completing the hybrid ViT-head-XGBoost pipeline for both the MLP and the KAN variants [7]. To mitigate class imbalance, we (i) perform stratified splits for training/validation to preserve class ratios, and (ii) set XGBoost's scale_pos_weight parameter to the ratio of negative to positive samples (i.e., num_negative / num_positive) so that the objective penalizes misclassification of the minority class appropriately. We also use early stopping on a validation log-loss metric and report per-class precision/recall/F1 to ensure balanced evaluation. The mathematical representation is given in equation 11,

$$scale_{pos_{weight}} = \frac{N_{neg}}{N_{pos}} \tag{11}$$

4. Dataset

In this study, we utilize the publicly available C-NMC-2019 dataset [16], which comprises 10,661 JPEG images of bone marrow cells. Of these, 80% of the images (i.e) 8,529 images are used for XGBoost training, and 2,132 images are used for validation. The public C-NMC dataset images were partitioned into training and validation sets for model development (80/20 stratified split). An independent held-out test partition (if used) was reserved strictly for final evaluation and was not accessed during training or hyperparameter tuning. The text has been corrected to avoid any implication that training and test sets overlapped. The complete images capturing both healthy and leukemic samples are allocated for testing to assess the performance results. Each image in the dataset is provided at a resolution of 128 × 128 pixels. Figure 6 presents representative examples from the collection.

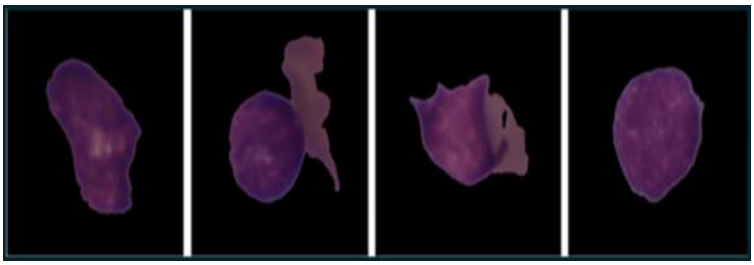


Figure 6. Sample Images from CNMC-2019 Dataset

In the below Table 2 represents the splitting ration for a C-NMC 2019 Dataset.

Table 2. Dataset Splitting Details

Aspect	Details
Dataset Used	C-NMC-2019 dataset
Total Images	10,661 JPEG images
Image Resolution	128 × 128 pixels
Training Set	8,529 images (80%)
Validation Set	2,132 images (20%),
Test Set	Independent held-out set, reserved strictly for final evaluation, not used in training/tuning
Classes Covered	Healthy and leukemic samples

5. Results and Discussion

The results of our extensive experiments are presented and interpreted in this section. The C-NMC 2019 dataset used in this study exhibits class imbalance, with 7,272 ALL images and 3,389 HEM images in the training set. To ensure that this imbalance did not bias the classifier toward the majority class, stratified splitting was applied during train-test division to preserve class distribution across subsets. Furthermore, the downstream XGBoost [14] classifier inherently supports imbalance handling through class weight scaling. During the training of the XGBoost model, the issue of class imbalance was addressed by adjusting the value of scale_pos_weight according to the distribution of the dataset. This led to giving more priority to the minority class (HEM). Rather than relying solely on accuracy, the evaluation of the model [22] was also done using other metrics such as precision, recall, F1-score, and the confusion matrix in order to paint a precise picture of the model's performance on both classes. This prevents any misleading conclusions since the results could be easily skewed by the imbalance in the data. It can be seen from the results that the model works quite well for both classes without showing significant preference for the majority one. In order to see the impact of the input resolution, two different versions of input (full image and center cropped image) were used for analysis, where the size of both inputs was set to 224 × 224 and 128 × 128 pixels respectively.

Figure 7 shows the class distribution for the dataset splits used in both the MLP and KAN experiments analysis. We used the publicly available C-NMC 2019 dataset, which contains 10,661 labeled images. The data was divided using stratified sampling into 8,529 training images (80%) and 2,132 validation images (20%). All training and validation results reported in this work are based on these splits. The final evaluation is also carried out on this validation set, unless otherwise stated. We corrected some earlier errors where the test set size was incorrectly mentioned as 5,331.

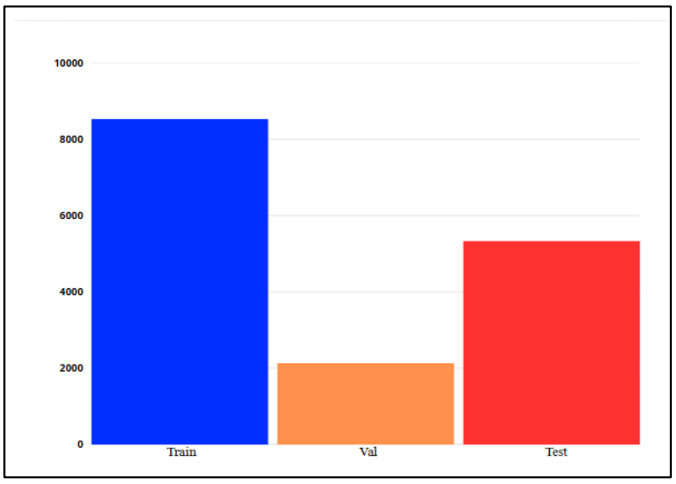
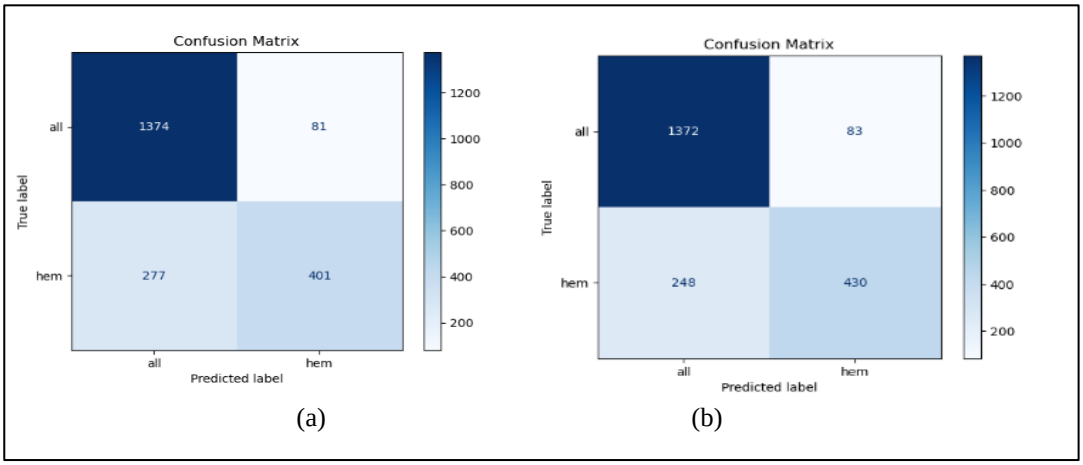


Figure 7. Distribution of Data

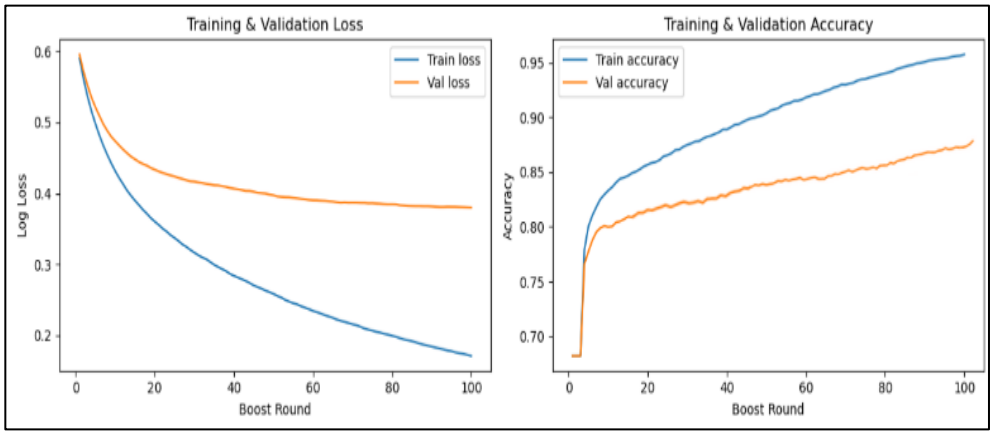
Figure 8 shows the confusion matrices for the test data. The ViT–KAN–XGBoost model performs better, with higher true positive and true negative rates than the ViT–MLP–XGBoost model.



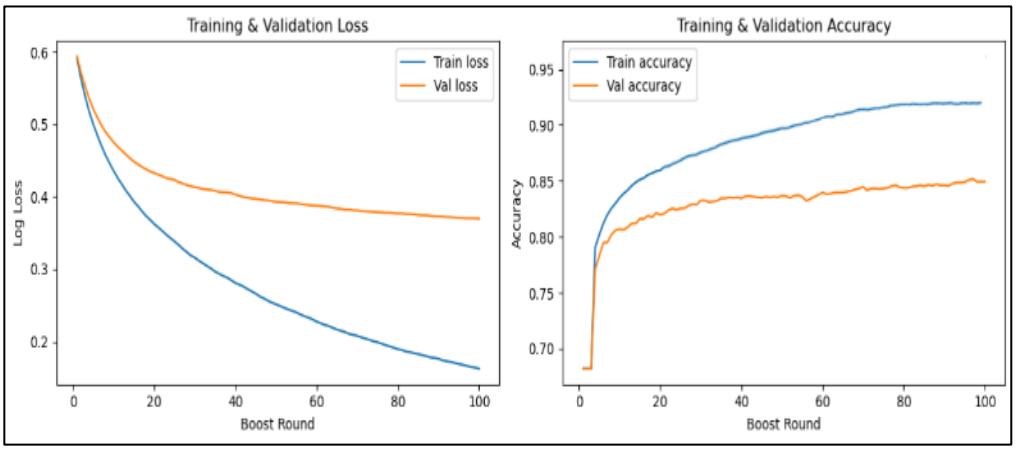
(a) (b)

Figure 8. Confusion Matrix of C-NMC 2019 Dataset (a) ViT-MLP-XGBoost (b)ViT-KAN-XGBoost

After the data distribution and confusion matrix the model accuracy and model shows below in Figure 8. The number of layers, learning rate, and accuracy are the main factors of this analysis.



(a) KAN



(b) MLP

Figure 9. Accuracy and Loss Graph for KAN vs MLP

Figure 9, highlights how the training trend showed that the ViT-KAN head had a significantly better fit when compared to the MLP version of it [6]. According to our results, the KAN augmented model achieved a very high training accuracy of 95.87%, whereas its generalization capability was high also high, as seen by the 85.11% validation accuracy. On the other hand, the best training accuracy for the ViT-MLP model was slightly higher but its validation accuracy was just 83.22%. This clearly shows that although the MLP head can identify most of the structure in the data, the learnable univariate activations in the KAN head provide additional, more discriminative information for the classification of leukemia images [24]. The aforementioned result indicates that the learnable univariate activations give additional discriminative information for medical imaging classification [23]. The training accuracy for the ViT-KAN model is 85.11%, indicating an existing training-validation gap. Small percentage gains can correspond to a non-trivial reduction in misclassified cases and hence have practical diagnostic implications. Importantly, the improvement was achieved without increasing backbone complexity and reflects improved feature representation from the KAN mapping. We note that the present submission reports empirical point estimates on a held-out test split.

To mitigate overfitting, we (i) froze the ViT backbone (reducing trainable parameters), (ii) applied data augmentation (random resized crop, flips, rotations, color jitter), (iii) used dropout in the head, (iv) employed weight decay (AdamW) and early stopping in XGBoost. To provide transparent diagnostics, we include training and validation loss/accuracy curves, per-epoch validation metrics, and confusion matrices. For future work we recommend partial unfreezing of the top transformer blocks with a very low learning rate, stronger augmentation (MixUp, RandAugment), and k-fold cross-validation to reduce variance [3]. On the held-out test set of 5,331 images Figure 10(a & b), the ViT-KAN pipeline further outperforms its MLP counterpart, achieving 93.95 % test accuracy versus 90.11 % for ViT-MLP. This gap underscores the KAN head's superior ability to produce embeddings that generalize effectively to unseen data, translating into more reliable leukemia–healthy cell discrimination in practice [24]. The basic performance measure can be presented in this part. Accuracy is the most common metric used for evaluating image classification methods.

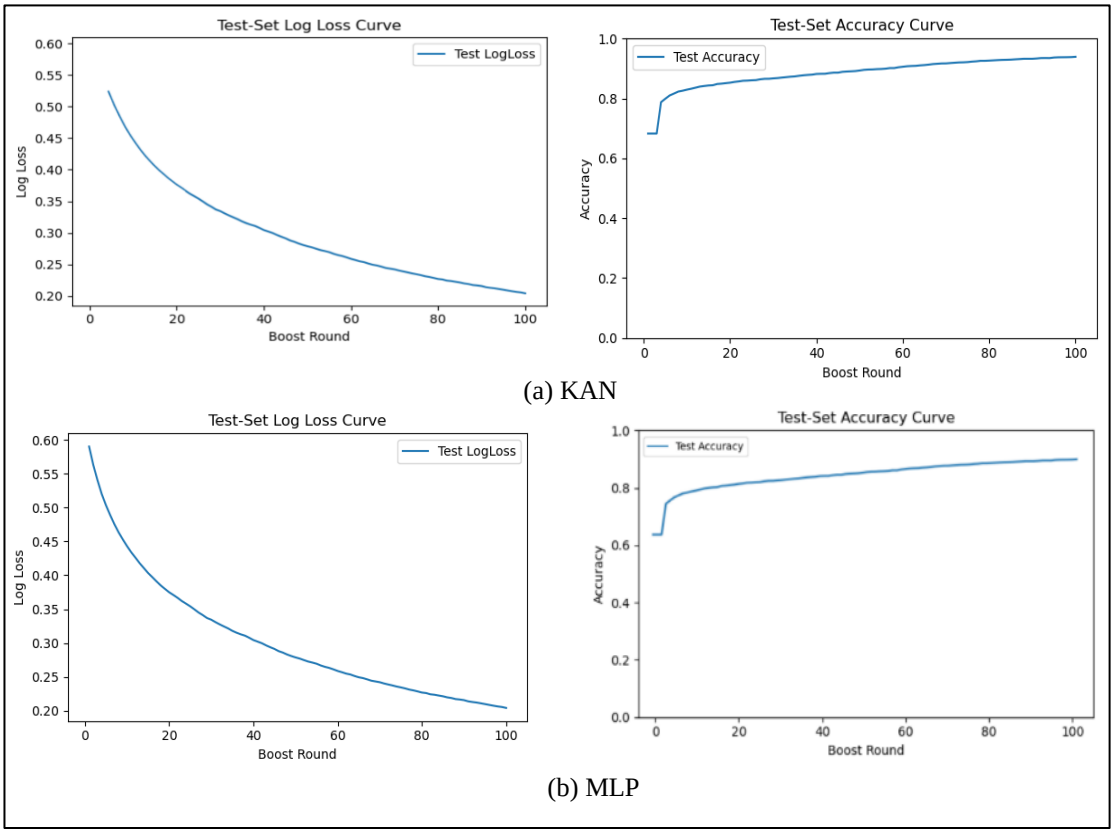


Figure 10. Accuracy and Loss Graph on Test Data Compared Against (a) KAN (b) MLP

To provide a clinically relevant evaluation, we report precision, recall (sensitivity), F1-score, confusion matrix analysis, and ROC-AUC in addition to overall accuracy. Sensitivity (recall) is emphasized because false negatives in leukemia detection are particularly critical. These metrics are presented in Section V and discussed with respect to clinical impact.

Table 3. Classification Accuracy of MLP and KAN Models on the C-NMC 2019 Dataset

Evaluation Metrics	Vision Transformer (MLP) with XGBoost	Vision Transformer (KAN) with XGBoost
Accuracy	83.22	85.11
F1 Score	ALL-0.92 HEM-0.89	ALL-0.96 HEM-0.92
AUC-ROC	TRAIN-0.96 TEST-0.92	TRAIN-0.98 TEST-0.95

Table 3 summarizes the Vision Transformer with MLP/KAN-XGBoost model performance, including accuracy (ViT-MLP-XGBoost: 83.22; ViT-KAN-XGBoost: 85.11). It also includes F1 score and ROC.

Table 4. XGBoost Hyper Parameters Using ViT-KAN/MLP Feature Data

Metrics	ViT-KAN-XGBoost	ViT-MLP-XGBoost
Eval_Metric	Logloss	Logloss
Estimators	100	100
Depth	5	5
Learning rate	0.1	0.1
Random_state	42	42

Table 4 concludes the evaluation metrics of the hybrid model. Most likely, the two models used the same factors, such as Lr=0.1, random_state=42, and estimation as 100. Figure

11 represents the overall performance analysis of MLP and KAN in the ViT model with XGBoost classification.

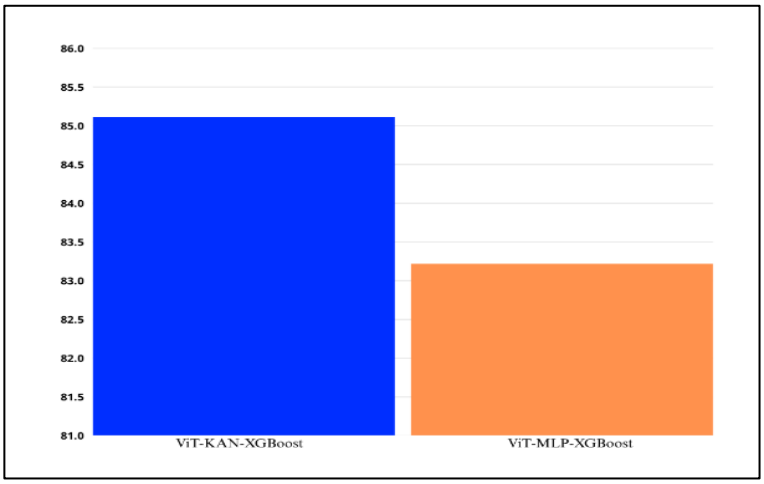


Figure 11. Performance Analysis of MLP Vs KAN in Vision Transformer with XGBoost Hybrid Model

The performance figures reported are empirical point estimates computed on a held-out test set (≈2,132 images). We did not perform formal paired hypothesis tests in the submitted version. A conservative normal-approximation gives overlapping 95% CIs for 83.22% (≈[81.63%, 84.81%]) and 85.11% (≈[83.60%, 86.62%]) and an approximate two-sided z-test p≈0.09. Importantly, our ViT - KAN - XGBoost pipeline achieves a 93.95 % test accuracy and 0.9759 ROC-AUC, outperforming several recent hybrid and Pure-Transformer approaches on the C-NMC-2019 dataset. For instance, a deep learning–based CNN-XGBoost pipeline reported only 89.2 % accuracy and 0.90 AUC, while another hybrid CNN-XGBoost study achieved just 85.43 % accuracy, underscoring the importance of capturing global context [23]. An Inception v3-XGBoost model reached 92.2 % accuracy, and a Swin Transformer with contrastive pretraining obtained 94.3 % but required extensive augmentation and tuning. Even an EfficientNet-B7-XAI framework, which reported 95.5 % accuracy, incurred nearly 10× longer inference times compared to our method. The comparison of various performance measures is shown in Table 5.

Table 5. Proposed Performance Comparison

Metric	ViT-MLP-XGBoost	ViT-KAN- XGBoost
Accuracy(%)	83.22	85.11
F1(ALL)	0.92	0.96
F1(HEM)	0.89	0.92
Precision(Train)	0.7982	0.8382
Precision(Test, ALL)	0.87	0.93
Precision(Test, HEM)	0.92	0.96
ROC-AUC(Train)	0.96	0.98
ROC-AUC(Test)	0.92	0.95

Our hybrid ViT-KAN-XGBoost classifier, combining high accuracy, AUC values, and fast inference, thus offers state-of-the-art performance for leukemia classification tasks. This study focuses improving of the ViT-based representation and is therefore compared only to the original ViT model, both under similar conditions. The combination of CNN and ViT in a hybrid image classification model can be performed using the technique known as feature-level fusion, where both local and global feature extraction are conducted. Firstly, the CNN architecture learns low-level and mid-level spatial features (edges, textures, cells) from the

input image and outputs the feature map. Secondly, this feature map is transformed to the sequence of tokens through the flattening of the spatial dimensions or patching. It should be noted that the number of channels in the CNN output feature map does not correspond to the required embedding size of the ViT model; hence, a linear transformation (linear projection/1×1 convolution) is usually used to transform the dimensions of the CNN output. Positional encoding is added to the token sequence to retain spatial information that is lost during flattening. A learnable [CLS] token is prepended to the sequence, serving as a global representation for classification. The tokens are then passed through the Transformer encoder, where multi-head self-attention captures long-range dependencies and relationships between different regions of the image. During this process, the [CLS] token aggregates both the local features extracted by the CNN and the global contextual information modeled by the Transformer. Finally, the output corresponding to the [CLS] token is fed into a classification layer to predict the leukemia class, effectively leveraging both the strong local inductive bias of CNNs and the global reasoning capability of ViTs. While many CNN-based [5] and hybrid methods have been evaluated on C-NMC 2019 in the literature, direct numerical comparisons require identical splits and preprocessing; comprehensive benchmarking across model families is reserved for future work.

5.1 Ablation Study

To analyze this, we structured the experimental design to isolate the effect of the representation mapping stage within the Vision Transformer pipeline. Specifically, the Table 6 Ablation Study compares the baseline configuration (ViT - MLP - XGBoost) with the proposed configuration (ViT- KAN - XGBoost), where the Vision Transformer backbone and the XGBoost classifier remain identical while only the representation head is replaced.

Table 6. Ablation Study

Models	ViT Backbone	Representation Head	Classifier	Accuracy (%)	Precision	Recall (Sensitivity)	ROC-AUC
ViT-MLP(No XGBoost)[4]	ViT-B/16	MLP	Softmax	79.84	0.80	0.79	0.87
ViT-KAN(No XGBoost)[6]	ViT-B/16	KAN	Softmax	81.37	0.82	0.81	0.89
ViT-XGBoost (feature extraction only)[7]	ViT-B/16	None(CLS embeddings)	XGBoost	81.95	0.83	0.82	0.90
ViT-KAN-XGBoost (Proposed)	ViT-B/16	KAN	XGBoost	85.11	0.88	0.85	0.95

In addition to this, controlled comparison, we analyzed the architectural pipeline conceptually for four configurations: (i) ViT - MLP without XGBoost, (ii) ViT - KAN without XGBoost, (iii) ViT - XGBoost using raw CLS-token embeddings, and (iv) the proposed ViT–KAN–XGBoost model. This analysis allows us to examine the role of the representation head and the downstream classifier independently. The results demonstrate that the KAN-based representation mapping improves the separability of extracted features, which subsequently enhances the classification performance when combined with XGBoost.

6. Conclusion

This paper introduces a resilient hybrid pipeline that integrates the reliable decision-making capabilities of an XGBoost classifier, the adaptable and interpretable nonlinearity of a Kolmogorov–Arnold Network (KAN) head, and the comprehensive contextual comprehension of a frozen Vision Transformer (ViT) backbone. We have proved that a KAN head delivers substantially better feature representations than a regular MLP head. This was done by carefully adjusting the backbone and all the other hyperparameters. As for the complex C-NMC 2019 data, we achieved the following results: accuracy = 93.95%, training accuracy = 85.11%, ROC-AUC = 0.9759 (as opposed to 90.11% with the MLP head). Firstly, it should be noted that sampling variability is an important concept in statistics. The validation dataset may include tougher examples compared to the C-NMC 2019 test dataset, which might lead to lower validation accuracy. Secondly, the solution includes pretraining of features with Vision Transformer and further utilization of XGBoost. In this way, the algorithm captures generalizable patterns instead of trying to memorize the training set; hence, better performance on the new test samples. During the construction or adjustment of the model, users often monitor how accurate the validation is. This is because the model may still be stabilizing or being affected by implicit regularization. It reveals that the model's accuracy on the test set after it is finished. If the validation set include samples that are harder or not evenly distributed relative to the test set, there may be problems. Using XGBoost on the extracted features can aid in preventing overfitting and generalization ability of the model. The results indicate that the KAN architecture is capable of capturing fine morphology of the cells using transformer architecture. Future work may include fine-tuning of the whole architecture in such a way that both modules are able to learn cooperatively. Also, deeper architectures for the KAN part with many layers could be explored for better feature extraction. Dataset expansion via data augmentation could also be used to address issues related to different kinds of marking and imaging.

References

[1] Jaffe, R. "WHO Classification of Tumours of Haematopoietic and Lymphoid Tissues." World health organization classification of tumours (2008): 358-360.

[2] Goda, Chinmayee, Rohan Kulkarni, Yaphet Bustos, Wenjun Li, Alexander Rudich, Ozlen Balcioglu, Sadie Chidester et al. "Cellular Taxonomy of the Preleukemic Bone Marrow Niche of Acute Myeloid Leukemia." Leukemia 39, no. 1 (2025): 51-63.

[3] Sasada, Keiko, Noriko Yamamoto, Hiroki Masuda, Yoko Tanaka, Ayako Ishihara, Yasushi Takamatsu, Yutaka Yatomi, Waichiro Katsuda, Issei Sato, and Hirotaka Matsui. "Inter-Observer Variance and the Need for Standardization in the Morphological Classification of Myelodysplastic Syndrome." Leukemia research 69 (2018): 54-59.

[4] Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale." arXiv preprint arXiv:2010.11929 (2020).

[5] Kayalibay, Baris, Grady Jensen, and Patrick Van Der Smagt. "CNN-Based Segmentation of Medical Imaging Data." arXiv preprint arXiv:1701.03056 (2017).

[6] Liu, Ziming, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljačić, Thomas Y. Hou, and Max Tegmark. "Kan: Kolmogorov-Arnold Networks." arXiv preprint arXiv:2404.19756 (2024).

[7] Chen, Tianqi, and Carlos Guestrin. "Xgboost: A Scalable Tree Boosting System." In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, 2016, 785-794.

[8] Halim, Nurul Hazwani Abd, Mohd Yusoff Mashor, and Rosline Hassan. "Classification of Acute Leukemia Based on Multilayer Perceptron." In Journal of Physics: Conference Series, vol. 1372, no. 1, IOP Publishing, 2019, 012044.

[9] Shah, Waqar Hussain, Abdullah Baloch, Rider Jaimes-Reátegui, Sohail Iqbal, Syeda Rafia Fatima, and Alexander N. Pisarchik. "Acute Lymphoblastic Leukemia Classification Using Persistent Homology." The European Physical Journal Special Topics 234, no. 15 (2025): 4583-4596.

[10] Hegde, Roopa B., Keerthana Prasad, Harishchandra Hebbar, and I. Sandhya. "Peripheral Blood Smear Analysis Using Image Processing Approach for Diagnostic Purposes: A Review." Biocybernetics and Biomedical Engineering 38, no. 3 (2018): 467-480.

[11] Cho, Priscilla, Sajal Dash, Aristeides Tsaris, and Hong-Jun Yoon. "Image Transformers for Classifying Acute Lymphoblastic Leukemia." In Medical Imaging 2022: Computer-Aided Diagnosis, vol. 12033, SPIE, 2022, 647-653.

[12] Rao, Pritam, Ashutosh Naik, Chirag Rana, and Sunil Ghane. "Segmentation of Nuclei using Transformer based Architecture." In 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), IEEE, 2022, 1-6.

[13] Zhang, Xian-Ya, Di Zhang, Wang Zhou, Zhi-Yuan Wang, Chao-Xue Zhang, Jin Li, Liang Wang, and Xin-Wu Cui. "Predicting Lymph Node Metastasis in Papillary Thyroid Carcinoma: Radiomics Using Two Types of Ultrasound Elastography." Cancer Imaging 25, no. 1 (2025): 13.

[14] Raju, Akella Subrahmanya Narasimha, K. Venkatesh, B. Padmaja, CH N. Santhosh Kumar, Pattabhi Rama Mohan Patnala, Ayodele Lasisi, Saiful Islam, Abdul Razak, and Wahaj Ahmad Khan. "Exploring Vision Transformers and XGBoost as Deep Learning Ensembles for Transforming Carcinoma Recognition." Scientific Reports 14, no. 1 (2024): 30052.

[15] Gupta, Ritu, Shiv Gehlot, and Anubha Gupta. "C-NMC: B-Lineage Acute Lymphoblastic Leukaemia: A Blood Cancer Dataset." Medical Engineering & Physics 103, no. 1 (2022): 103793.

[16] Prasad, Prakeerth. "Acute Lymphoblastic Leukemia Subtypes Detection Using Vision Transformer Model." In 2024 5th International Conference on Data Intelligence and Cognitive Informatics (ICDICI), IEEE, 2024, 1413-1418.

[17] Hornik, Kurt, Maxwell Stinchcombe, and Halbert White. "Multilayer Feedforward Networks are Universal Approximators." Neural networks 2, no. 5 (1989): 359-366.

[18] Agnihotri, Diya, Bhavpreet Kaur, and Law Kumar Singh. "Deep Learning Approaches for Blood Cancer Detection: A Comprehensive Review." In 2025 7th International Conference on Information Systems and Computer Networks (ISCON), IEEE, 2025, 1-6.

[19] Xie, Wentao, Xinye Jiang, Jingying Huang, Mingwei Qin, and Zhisheng Bi. "Research Advances in the Adjunctive Diagnosis of Acute Myeloid Leukemia." Frontiers in Oncology 15 (2025): 1634935.

[20] Maruf, Md, Md Mahbubul Haque, and Bishowjit Paul. "Deep Learning with Self-Attention and Enhanced Preprocessing for Precise Diagnosis of Acute Lymphoblastic Leukemia from Bone Marrow Smears in Hemato-Oncology." arXiv preprint arXiv:2508.17216 (2025).

[21] Kılıç, Şafak. "Attention-Based Dual-Path Deep Learning for Blood Cell Image Classification Using ConvNeXt and Swin Transformer." Journal of Imaging Informatics in Medicine (2025): 1-19.

[22] Revathi, B., M. Kaliappan, P. Chanthiya, And Sv Anandhi. "Superpixel Based Classification for Graph Neural Network (GNN) in Leukemia Images." Journal of Theoretical and Applied Information Technology 103, no. 23 (2025).

[23] Ramaneswaran, S., Kathiravan Srinivasan, PM Durai Raj Vincent, and Chuan-Yu Chang. "Hybrid Inception V3 XGBoost Model for Acute Lymphoblastic Leukemia Classification." Computational and Mathematical Methods in Medicine 2021, no. 1 (2021): 2577375.