IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Traitement du Signal
Vol. 43, No. 1, February, 2026, pp. 467-476
Journal homepage: http://iieta.org/journals/ts

Multi-Scale Transformer and Explainable AI Framework for Automated Detection of Circulating Tumor Cells

Kavitha Sandanam[1*], Raghuraman Sivalingam[2]

[1] Department of Electronics and Communication Engineering, Velammal Engineering College, Chennai 600066, India
[2] Department of Electrical and Electronics Engineering, Velammal Engineering College, Chennai 600066, India

Corresponding Author Email: kavithasandanam2408@gmail.com

https://doi.org/10.18280/ts.430133

Received: 18 May 2025
Revised: 2 December 2025
Accepted: 21 January 2026
Available online: 28 February 2026

Keywords:
Circulating Tumor Cells, Multi-Scale Swin Transformer, Aquila Optimizer, Explainable Boosting Machine

ABSTRACT

Cancer remains one of the most widespread and life-threatening diseases worldwide. Every year millions of new cases diagnosed with cancer. According to the World Health Organization (WHO), cancer accounted for nearly 10 million deaths in 2023. One of the key biomarkers for early cancer detection is Circulating Tumor Cells (CTCs). An image processing-based detection approach is a promising solution for early detection and diagnosis. In this work, a three-fold CTC classification approach is proposed. Initially, the important feature from CTC images is extracted using the Multi-Scale Swin Transformer (MS-Swin-T). Then, the relevant feature is extracted using Aquila Optimizer (AO). Finally, the feature from optimizer is classified using an Explainable Boosting Machine (EBM) classifier. Experimental results on the data set show that the proposed MS-Swin-T shows higher precision, accuracy, recall and F-Score rate than previously proposed models. This approach holds significant potential for automated CTC detection and contributes to early cancer diagnosis and precision medicine applications.

1. INTRODUCTION

Cancer is the major cause of death worldwide, with millions of new cases diagnosed each year. Early detection of cancer diseases supports timely treatment and increases patient survival rates [1, 2]. One of the key biomarkers for early cancer detection is CTCs—tumor cells that detach from primary or metastatic tumors and enter the bloodstream. The early diagnosis and analysis of CTC is used for prevention and nursing response. But finding its presence is a difficult task due to its extreme rarity (as low as 1 CTC per billion blood cells) [3].

Different techniques have been developed for CTC detection. It can be categorised into biological, physical, and image-based approaches. Biological methods based on immunomagnetic separation and molecular markers to isolate and identify CTCs. Physical methods use size differences and density between CTCs with filtration techniques. Image-based methods use microscopic imaging and computational techniques for automated CTC identification [4].

Recently, the arrival of Artificial Intelligence (AI) techniques has received greater attention due to their accuracy. AI models can be classified into two categories: Machine Learning and Deep Learning [5]. In ML models, the feature learning and classification processes are processed independently. Conversely, in Deep Leraning (DL) models, the learning and categorisation are carried out in the architecture itself. The well-known ML models are catboost, Decision Trees and Random Forests (RF). The well-known DL models are Convolutional Neural Networks (CNNs) and Transformer-based architectures. However, these models have several limitations: limited spatial awareness, high computational complexity, lack of feature selection mechanisms and limited model interpretability. Conventional CNN models failed to capture long-range dependencies and to detect morphological differences between CTCs and normal blood cells. Deep CNN-based architectures require large datasets and extensive computational resources for processing them. Many DL models extract a high volume of redundant features which can lead to overfitting and decreased efficiency.

Based on the drawbacks of existing DL models, there is a need for an advanced, efficient and interpretable DL model. In this work, a novel threefold model is proposed for CTC classification. For improved feature extraction, the MS-Swin-T is proposed. Then, optimized with AO. The classification is done with Explainable Boosting Machines. It can overcome the limitations of existing DL models.

2. RELATED WORK

Liang et al. [6] proposed a CNN-RNN model for blood cell image classification. The proposed model captures long-term dependencies in features and uses transfer learning with pre-trained ImageNet weights to overcome the limitations of existing CNNs.

Guo et al. [7] developed a CNN-based model for detecting CTCs in peripheral blood using imFISH images. They collected data from 776 patients and applied transfer learning

to increase model performance.

Yanagisawa et al. [8] developed a VGG16-based CNN to classify anticancer drug sensitivity based on cell morphology. The proposed model achieves 80% accuracy for the detection of drug effects at the single-cell level. A hybrid CNN-based deep learning model is proposed by Soto-Ayala et al. [9] to automate blood cell analysis for cancer detection. Their model achieved high accuracy and a lower false negative rate compared to existing methods.

Ciurte et al. [10] proposed a boosting technique-based learning model for automatic CTC detection in blood using unstained cell microscopy. Experimental results on 263 dark-field microscopy images show that the boosting model achieved 92.87% sensitivity and 99.98% specificity.

A hybrid model which combines CNN and support vector machine (SVM) is proposed by Park et al. [11] for CTC classification without immunofluorescence staining. The hybrid model extracted four key morphological characteristics and achieved classification results with over 90% sensitivity and specificity.

In Alexander et al.'s study [12], the authors proposed a three-level detection model for CTC classification. Initially, the Cytokeratin stains are identified using RetinaNet. Mask-RCNN is applied for nuclei detection. Finally, Otsu thresholding is used for CD-45s identification. The overall accuracy achieved by the model is 97.72%.

Shehta et al. [13] introduced an ensemble model for skin lesion categorization in dermoscopy images. The variants of ConvNeXt models, like small, medium, and high, are used for final lesion classification. The ensemble model achieved an overall classification accuracy, sensitivity, and specificity of 96%, 83.1%, and 96.8%, respectively.

A modified AlexNet-based CTC classification model is proposed by Kisanuki et al. [14] from fluorescence microscopy images. In preprocessing, a series of filters is applied for noise removal. Then, the modified AlexNet-based model is used for classification. Experimental results on 5040 images show that the AlexNet model achieves higher accuracy.

Chen et al. [15] proposed a hybrid model for white blood cell (wbc)classification. The hybrid model involves a pre-trained ResNet and a ConvNet for feature learning. In addition, a spatial and channel attention module is added to the hybrid model for accurate classification.

Similarly, Rao et al. [16] proposed an optimisation-based model for WBC classification. The feature extraction from WBC images is done by MobileNetV3. Then, the important features are identified using the Cuckoo Search optimizer. Finally, the types of cells are classified by the ShuffleNetV2 model. Results on the Raabin-WBC dataset show that the proposed model achieves 95.6% accuracy with minimal memory requirement overhead.

Likewise, Krishna Prasad et al. [17] developed a three-fold model for WBC classification. For noise removal, a modified median filter is used. Then, for region of interest extraction, Color Balancing Binary Threshold is applied. The final WBC classification is done by Optimal DCRNet.

Wang et al. [18] developed a DL based classification model for bone marrow cells (BMC) images. Initially, the watershed algorithm is used to remove stained impurity cells. A dual-channel convolutional block attention network is used for feature extraction. Finally, it classifies BMC cells into the types of myeloblasts and monoblasts. Experimental results show that the proposed model achieved 96.8% macro F1-score on the BMC-1 dataset and 87.49% macro F1-score on the BMC-2 dataset.

A hybrid transformer and CNN model-based Cervical cytology image classification is proposed by Fang et al. [19]. For feature discrimination, a Deep Integrated Feature Fusion (DIFF) block is added between the convolutional layers. Results on the cervical image data set show that the DIFF DIFF-based model achieves higher accuracy than existing models.

Üzen and Firat [20] proposed a new multipath DL architecture for WBC classification. It integrates ConvMixer and Swin Transformer for feature capturing. Swin Transformer uses self-attention to learn global features. The spatial relationship between pixels is identified using ConvMixer.

Batool and Byun [21] constructed a DL model based on Modified EfficientNet-V2 for the classification of acute lymphoblastic leukaemia and normal cells in WBC images. The EfficientNet-B3 model uses depthwise separable convolutions to improve computational efficiency and achieve strong classification results.

To classify cervical cancer types, Pacal and Kılıcarslan [22] proposed a modified model based on CNN and ViT-based models. In addition, they applied synthetic image generation to improve data diversity and collaborative detection to increase the classification prediction rate.

A hybrid loss function with label smoothing for a DL model is introduced by Chen et al. [23] for cervical cell classification. The proposed loss function is applied in the ShuffleNetV2 model and validated on the SIPaKMeD dataset.

3. PROPOSED MODEL

In this work, a novel threefold DL architecture is proposed for CTC categorisation. The conventional Swin Transformer is modified with multi-scale operation in patch embedding and window attention. The extracted feature is optimized using AO. The optimized features are classified using the Explainable Boosting Machines model.

3.1 Swin Transformer

The Swin Transformer is a hierarchical vision transformer designed for efficient high-resolution image processing. It applies self-attention globally for improved feature extraction. The working of the Swin Transformer involves three main processes: Window-based self-attention, Shifted windows and Hierarchical structure. Window-based self-attention is computed within fixed-size local windows. The shifted windows strategy is used for cross-window feature interaction without high computational cost.

3.2 Key mechanisms in Swin Transformer

3.2.1 Patch splitting and embedding

The input image $X \in \mathbb{R}^{H \times W \times 3}$ is divided into non-overlapping patches of multiple sizes P1×P1, P2×P2, …PN×PN which forms a sequence of feature vectors. These patches are linearly projected to obtain an initial feature representation.

$$F_{patch,i} = W_{p,i}X + b_{p,i} \quad (1)$$

where, $W_{p,i}$ is the projection matrix for the i-th patch size, $b_{p,i}$

is the corresponding bias term, $F_{patch,i}$ represents the extracted patch features for scale i. The structure of the proposed Swin architecture is given in Figure 1.

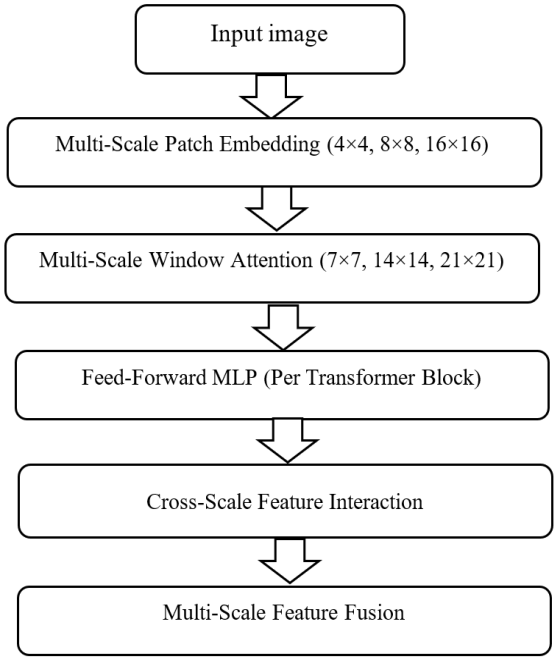


Figure 1. Swin Transformer architecture

The final multi-scale feature representation is obtained by concatenation and transformation:

$$F_{patch} = W_c \left[F_{patch,1}, F_{patch,2} \ldots..F_{pathc,N} \right] + b_c \tag{2}$$

where, W_c and b_c are trainable parameters for feature fusion.

3.3 Multi-scale window-based self-attention

In standard Swin Transformer, self-attention is applied within fixed-size windows. However, in MS-W- MSA Swin Transformer applies self-attention within non-overlapping windows of size M×M. In the standard Swin Transformer, self-attention is applied within fixed-size windows. However, in MS-W-MSA, multiple window sizes W1×W1, W2×W2, ...WM×WM are used, leading to multi-scale spatial attention:

$$MS-W-MSA(Q,K,V) = \sum_{j=1}^{M} W_j Softmax\left(\frac{Q_j K^T{}_j}{\sqrt{d}} + B_j \right) V_j \tag{3}$$

where, j indexes the window scale, Q, K, V are the query, key, and value parameters, B_j is the correlated positional bias specific to each window, d_k is the scaling factor, W_j are learnable weights to adaptively merge information across scales. The final multi-scale attention output is obtained via weighted summation.

3.4 Shifted windows mechanism

To enable feature interactions across windows, windows are shifted by M/2 pixels before the next transformer block. It is used for neighboring patches interaction.

$$F_{SW-MSA} = MS-W-MAS(Shift(F_{prev})) \tag{4}$$

where, Shift(.) is a predefined shifting function.

3.5 Cross-scale feature interaction and fusion

The extracted multi-scale features from different patch and window sizes are aggregated using a weighted summation mechanism:

$$F_{merged} = \sum_{i=1}^{N} \sum_{j=1}^{M} W_{ij} F_{MSA,ij} \tag{5}$$

where, W_{ij} are learnable parameters ensuring adaptive cross-scale feature fusion.

The final feature representation is passed through a Feed-Forward Network (FFN) with GELU activation:

$$F_{FFN} = GELU\left(W_1 F_{merged} + b_1\right) W_2 + b_2) \tag{6}$$

where, W_1, W_2 are trainable matrices, b1, b2 are bias terms.

In our proposed multi-scale window attention mechanism (MS-W-MSA), multiple window sizes $W_1 \times W_1, W_2 \times W_2, \ldots, W_M \times W_M$ are used to capture different spatial scales within the self-attention layers. The selection of these window sizes is guided by the task's need to capture both fine-grained details and broader contextual information. This choice is not arbitrary but designed to enhance performance in CTC detection. The learning strategy for the adaptive weights W_j is handled through a Softmax normalization mechanism which ensures balanced attention across all window scales. This mechanism prevents any single scale from being overly emphasized or ignored. It is used for a robust fusion of multi-scale information. Moreover, the use of a SW-MSA ensures that neighboring patches can interact and increase the model's ability to detect spatial relationships across the entire input. Thus, the multi-scale window sizes and their corresponding learnable weights are designed to optimize feature extraction from both local and global perspectives.

3.6 Feature selection using Aquila Optimizer (AO)

Once features are extracted from MS-Swin-T, AO is applied to select the most informative features. Feature optimization is crucial in DL models to enhance performance with reduced computational complexity. It is also used to eliminate redundant or irrelevant information. By optimizing the selected features, models become more interpretable and computationally efficient**.**

3.6.1 AO model

The Aquila is the well known bird of prey in the Northern region. While hunting, the male Aquila is the bird that can hunt its prey solo. It can utilize its sharp talons and speed to hunt its prey like squirrels, hares, rabbits and also the many birds and animals. Most likely Aquila diet is the ground squirrels. The Aquila has used four steps for hunting with several distinct differences. The ability of Aquila is cleverer and quicker to move a back-and-forth motion in its hunting strategy and depending on the situation. There are several steps taken for

hunting methods of Aquila which are given in the following:

The first step is a heavy soar with a vertical stoop which is used to choose its prey. The second step is a contour flight with a short glide attack done by Aquila. The Aquila catches the prey that cannot run or fly from it. The third step is a slow descent attack by a low flight. The Aquila is bending low towards the ground, and next attacks gradually on the prey. It chooses its victim and lands on the neck and back of the prey that can be tried to penetrated. The different step is walking and grabbing the prey by walking on the surface and grabbing to pull its target. This strategy is used to grab a young or large prey like a deer or sheep in its region.

By observing the hunting behaviour of Aquila, it is clear that it is a more skilful and intelligent hunter and probably after humans. The AO is developed based on this strategy to apply to the TransUnet segmentation model. The AO can be derived with several steps which are given in the following.

(i) Solutions Initialization

The initialization is the basic step which is a population-based model that randomly places a population of candidate solutions (X'). The position is randomly initiated as follows:

$$A'_{ij} = r \times \left(UB'_j - LB'_j\right) + LB'_j,\ \text{i=1,2…N}_\text{j}\text{=1, 2…D} \qquad (7)$$

where, A' indicates a set of present candidate solutions, A'_i indicates positions of the i^{th} solution, r indicates a random number, UB'_j indicates a j^{th} upper bound and LB'_j represents the j^{th} lower bound.

(ii) AO based Mathematical derivation

The AO method is derived from its four-step hunting behaviour that can be transferred from the exploration stage to the exploitation stage. Using these stages, the hunting behaviours are in the condition if $\text{t}' \le (23) * \text{T}'$.

Step 1: High soar with the vertical stoop (A1)

The AO is broadly explored from a high soar to recognise the search space area. Aquila's behaviour of high soar with the vertical stoop is expressed as below:

$$A_1'\left(t'+1\right) = A'_{best}\left(t'\right) \times \left(1 - \frac{1}{T}\right) + \left(A'_M\left(t'\right) - A'_{best}\left(t'\right) * r\right) \qquad (8)$$

Eq. (8) expressed the first step of the search model $A1$. The $A'_{best}(t')$ denotes the best fitness until t'^{th} iteration to predict an approximate location of prey. The $1 - \frac{1}{T}$ can be controlled an exploration through several iterations. $A'_M(t')$ indicates the locations of the present solution of mean value that is connected at t'^{th} iteration is expressed in the below Eq. (9). In the above Eq. (8), the random value among a 0 and 1. t' and T' denotes a present iteration and the higher quantity of iteration.

$$A'_M\left(t'\right) = \frac{1}{N}\sum_{i=1}^{N} A'_i\left(t'\right),\ j' = 1,2,\ldots,D \qquad (9)$$

Step 2: Narrowed exploration (X'_2)

When the prey is identified from high, it can circle the aimed prey, observe the land, and then attack which is named contour flight with random movement. The Aquila selected a target area of prey narrowly in preparation for the attack. This narrow strategy of hunting is expressed in Eq. (10):

$$A'_2\left(t'+1\right) = A'_{best}\left(t'\right) \times levy\left(D\right) + A'_R\left(t'\right) + \left(y' - x'\right) * r \qquad (10)$$

where, $A'_2(t'+1)$ indicates a next iteration solution of t' by a second step search strategy (A'_2), $A'_R(t')$ indicates a random result occupied in the range between zero ti N at the t^{th} iteration. *Levy* (D) denotes a Levy flight parameter that is expressed using Eq. (11):

$$levy\left(D\right) = s \times \frac{u' \times \sigma'}{|v|'} \qquad (11)$$

where, s denotes fixed values as 0.01, u and v represent the random numbers among 0 and 1. σ' expressed in the following Eq. (12):

$$\sigma' = \left(\frac{r'(1+\beta') \times \sin\left(\frac{\pi\beta}{2}\right)}{\frac{r'(1+\beta')}{2} \times \beta \times 2^{\left(\beta - \frac{1}{2}\right)}}\right) \qquad (12)$$

where, β' indicates a tuning parameter. The Eqs. (13)-(17) below, y' and x' denoted the spiral structure in the search is expressed as:

$$y' = r' \times \cos\left(\theta\right) \qquad (13)$$

$$x' = r' \times \sin\left(\theta\right) \qquad (14)$$

where,

$$r' = r'_1 + U' \times D'_1 \qquad (15)$$

$$\theta = -\omega \times D'_1 + \theta_1 \qquad (16)$$

$$\theta_1 = \frac{3 \times \pi}{2} \qquad (17)$$

where, r'_1 is valued between one to twenty for the constant search cycles, D'_1 indicates an integer number from 1 to the search space length (D'), U' indicates a low value fixed to 0.00565 and ω denotes a minimum value fixed to 0.005.

Step 3: Expanded exploitation (X3)

If the prey region is marked accurately, then Aquila is ready for attack. This attack is named as slow descent attack by low flight which is the third step strategy of Aquila. Where, the optimizer executes the chosen place of the target to reach closer place of the target. This step is expressed in Eq. (18):

$$\begin{aligned} &A'_3\left(t'+1\right) = \\ &A'_{best}\left(t'\right) - A'_M\left(t'\right) \times \propto' - r + \left(\left(UB' - LB'\right) \times r + LB'\right) \times \delta' \end{aligned} \qquad (18)$$

where, $A'_3(t'+1)$ denotes the next iteration solution of t' *derivate* by the third search approach ($A3$). $A'_M(t')$ indicates a current solution mean value at t' *th* iteration. $\propto'$ and δ' are the exploitation parameters to a minimum value between zero to one.

Step 4: Narrowed exploitation (A4)

If the Aquila is so close to the prey, the land and attack by walking which is the fourth step of the Aquila strategy named as walk and grab prey. This step is expressed in Eq. (19):

$$\begin{aligned} &A'_4\left(t'+1\right) = \\ &QF' \times A'_{best}\left(t'\right) \times \left(K'_1 \times A'\left(t\right) \times r\right) - K'_2 \times levy\left(D\right) + r \times K'_1 \end{aligned} \qquad (19)$$

where, $A_4'(t'+1)$ indicates a next iteration solution of t' by $A4$. QF' represents the quality function of search strategies equilibrium given in Eq. (20) and then. K_1' indicates a different motion to track the prey in elopes that is expressed in Eq. (21). K_2' indicates a decreasing value from 2 to 0 of AO flight slope from the initial position to the final position that is given in Eq. (22):

$$QF'(t') = t^{2\times r-1/(1-T)^2} \tag{20}$$

$$G_1^{'} = 2\times r-1 \tag{21}$$

$$G_2^{'} = 2\times(1-\frac{t}{T}) \tag{22}$$

Pseudocode for feature selection using the Aquila Optimizer

```
Initialize Population A' with N candidate feature subsets
Set maximum iterations T and initialize best solution X_best
For each iteration t' in range(T):
    Compute fitness for each candidate solution
    Update A_best with the best-performing subset
    If t' ≤ (2/3) * T: # Exploration Phase
        # High Soar with Vertical Stoop (Eq. (1))
        # Narrowed Exploration (Eq. (2))
        Else: # Exploitation Phase
        # Expanded Exploitation (Eq. (3))
            # Narrowed Exploitation (Eq. (4))
    # Update population with new feature subsets
    Replace worst solutions in A' with A1, A2, A3, A4 based on fitness
    Evaluate new solutions and update A_best
Terminate when convergence criteria is met
Return Abest (optimal feature subset)
```

The AO for feature selection begins with the initialization phase where a set of candidate feature subsets is generated randomly. Each subset represents a potential solution containing a selected group of features. The fitness function $f(\mathrm{A})$for a feature subset X can be expressed as:

$$f(\mathrm{X}) = \frac{1}{1+\text{Classification Error}(A)} \tag{23}$$

where,

$$\text{Classification Error}(A) = \frac{FP+FN}{TP+TN+FP+FN} \tag{24}$$

Here, TP, TN, FP, and FN represent the number of true positives, true negatives, false positives, and false negatives, respectively. The aim of the optimizer is to reduce the classification error by selecting the optimal subset of features.

During the exploration phase, AO applies two strategies: high soar and narrowed exploration which help in identifying promising feature subsets by encouraging diverse searches across the feature space. This phase prevents premature convergence and ensures a broad search for optimal solutions. Following this, the exploitation phase is introduced to refine the selected features further. AO uses expanded exploitation and narrowed exploitation techniques to fine-tune the feature subsets by focusing on the most informative features while reducing redundancy.

Throughout the process, AO continuously updates the best solution keeping track of the optimal feature subset with the highest classification performance. The optimization process terminates when the convergence criteria are met or after a predefined number of iterations. At the end of the process, the most informative and compact feature subset is selected.

3.6.2 EBM-based classification

EBM is a variant of cyclic gradient boosting Generalized Additive Model designed to overcome the limitations of existing ML models like RF and catboost models. It resembles a tree-based structure to achieve higher accuracy. It can be expressed as follows:

$$g\left(E\left[y\right]\right) = \beta_0 + \sum f_j(x_j) \tag{25}$$

where g denotes the link function. It can be used to configure the model as a regression or a classification. Compared to other models, it learns features using bagging and gradient boosting. For effective boosting, the round-robin fashion is applied in feature learning. This round robin fashion is used to overcome the effects of co-linearity and identify important features based on the best feature function fj. In addition, it can detect pairwise interaction automatically as follows:

$$g\left(E\left[y\right]\right) = \beta_0 + \sum f_j(x_j) + \sum f_{i,j}(x_i, x_j) \tag{26}$$

This interaction increases EBM accuracy further. EBM is an additive model where each feature contributes to classifications in a modular way.

4. RESULT AND DISCUSSION

The training and testing images are collected from open source websites. It consists of 13472 normal images and 13000 CTC images. The proposed model is coded in Python. The entire data set is divided into training and test set images. The test set includes 4118 normal images and 3934 CTC images. The MS-Swin-T is trained using a standard AdamW optimizer with a learning rate of 1e-4. The batch size is set to 32. The model is trained for 100 epochs. For the AO, the population size is set to 50 candidates with a maximum iteration count of 100. The weight decay parameter was set to 1e-5. For the EBM classifier, a tree depth is set to 5 with a learning rate of 0.05.

The visualization of the dataset image is shown in Figure 2.

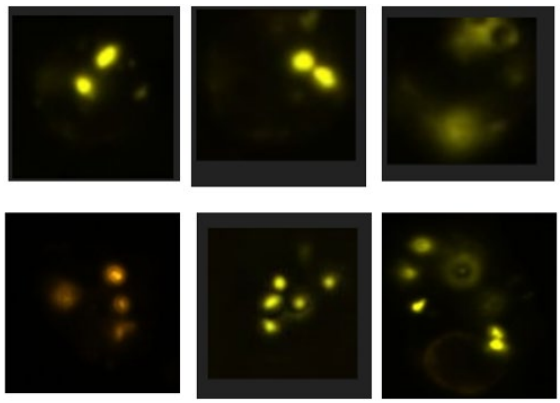


Figure 2. Visualization of input images

The results of proposed models can be assessed using the parameters of Recall, Accuracy, F1Score and precision.

The relationship between the classification error of a model and the Number of Features Selected is given in Figure 3. Each blue diamond on the plot represents a certain number of features that were used to train and evaluate the model which results in a corresponding classification error.

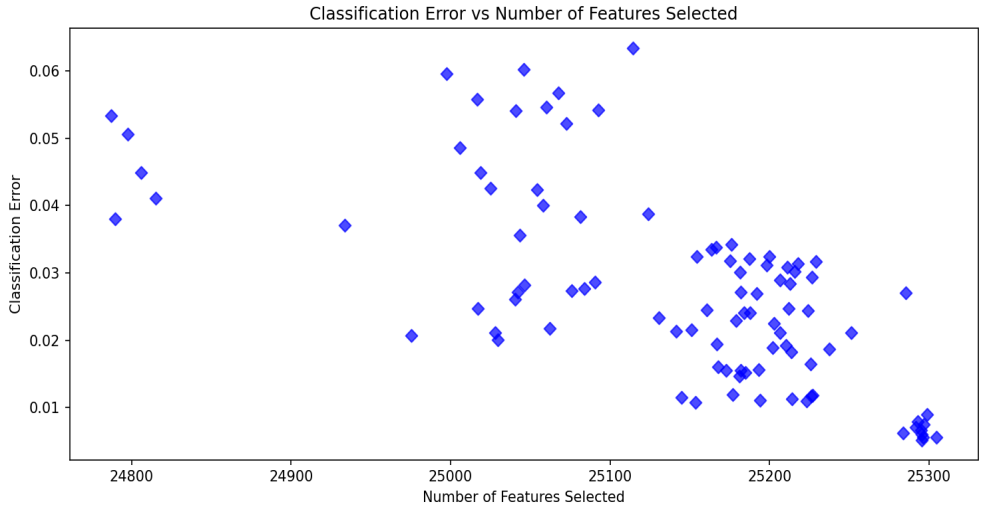
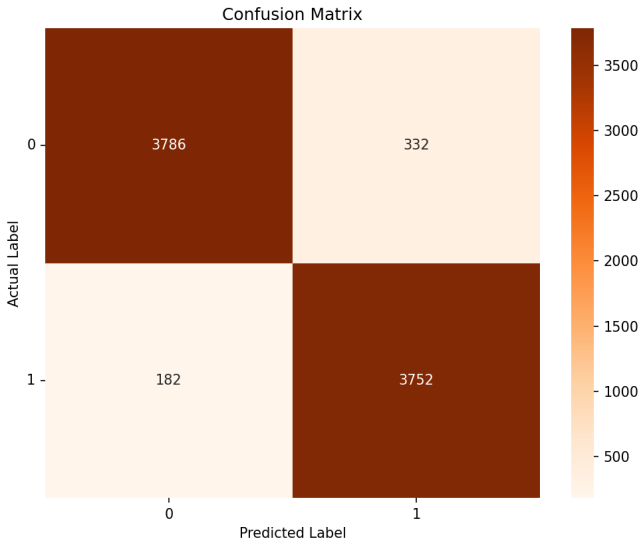


Figure 3. Number of features versus classification error

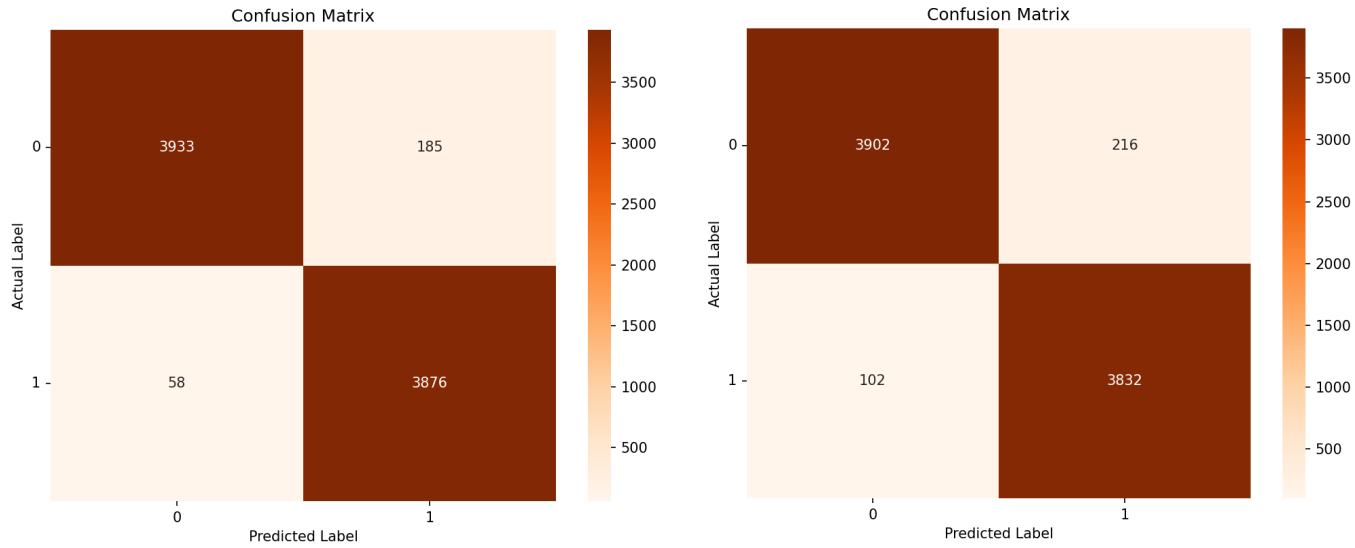
Figure 4. AO fitness function evaluation

The fitness function optimization over a number of iterations is given in Figure 4. This axis represents the progression of the optimization process. This x-axis represents the progression of the optimization process. The fitness function is set to classification error. The y-axis represents the value of the fitness function being minimized. The decreasing nature of the fitness function indicates that the AO algorithm is successfully learning or finding optimal features that improve the accuracy of the CTC classification.

The ablation study of the proposed model is given in Table 1. The models are evaluated based on Precision, Recall, F1-Score, and Accuracy. The Full Model (MS-Swin-T + AO + EBM) achieves the highest accuracy of 98% with higher precision and recall rates. The inclusion of AO and EBM improves the model's ability to learn complex patterns with increased accuracy rates.The accuracy drops slightly to 96.8% when AO is removed. It is observed that AO contributes to further refining the model's predictive capabilities. The precision and recall values show a minor decrement.

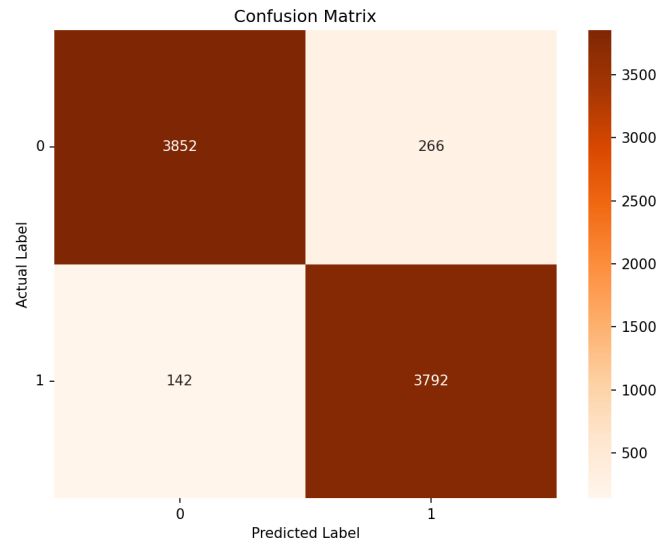


(a) Proposed Model

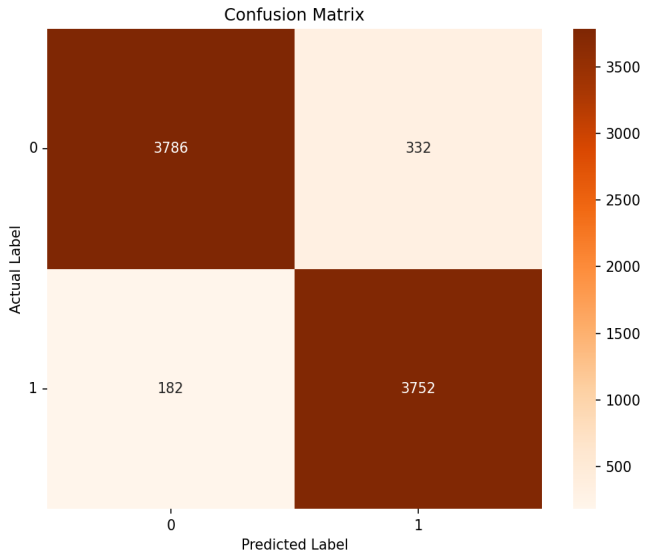


b) MS-Swin -T + EBM

c) MS-Swin-T +AO



d) MS-Swin-T



e) CNN Model

Figure 5. Confusion matrix of the models

Table 1. Ablation study of the proposed model

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Sensitivity (%)	Specificity (%)	AUC (%)
Full Model (MS-Swin-T + AO + EBM)	98.0	97.8	97.9	97.9	97.5	98.3	98.5
MS-Swin-T + EBM (No AO)	96.8	97.1	96.3	96.7	96.2	97.0	97.1
MS-Swin-T + AO	96.0	95.3	95.5	95.4	94.8	96.4	96.2
MS-Swin-T Only	94.0	93.6	94.1	93.8	93.1	94.5	94.0
Baseline CNN Model	93.4	92.8	93.0	92.9	92.3	93.5	93.2

Similarly, the accuracy further decreases to 96% when EBM is excluded. It is observed that EBM performs an important role in enhancing decision boundaries and improving generalisation. Using only MS-Swin-T, without any additional optimizations, results in an accuracy of 94%. Finally, the Baseline CNN Model which lacks the advanced transformer-based architecture and achieves the lowest accuracy of 93.4%. The confusion matrix obtained for the proposed model is given in Figure 5.

To validate the performance further, the MS-Swin-T model is validated on EGAD00001003601 associated with the Direct Detection of Early-Stage Cancers using Circulating Tumor DNA project. This dataset consists of 550 samples. After augmentation, the performance of the model is assessed. The model achieves an accuracy of 95% which proves the model's suitability for the different dataset environments. The results are given in Table 2.

To solve the class, the SMOTE (Synthetic Minority Over-sampling Technique is used. The use of SMOTE and class weight adjustments led to an improvement of 1.3% in accuracy, 1.3% in precision, and 1.6% in recall, respectively. The results are given in Table 3.

In addition, a 10-fold cross-validation is implemented to provide a more reliable evaluation. The 10-fold cross-

validation approach resulted in a 0.7% increase in accuracy. It proves the model's generalization ability compared to the initial 70:30 split. The results are given in Table 4.

Table 2. Performance on EGAD00001003601 dataset

Metric	MS-Swin-T Model
Accuracy	95.00%
Precision	94.50%
Recall	95.50%
F1-Score	95.00%

Table 3. Model performance with SMOTE

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
No Class Balancing	96.8	97.1	96.3	96.7
With SMOTE + Class Weights	98.1	98.4	97.9	98.1

The computational assessment of the MS-Swin-T model is shown in Table 5. The use of multi-scale operation minimally increases training time, but the inference speed remains within acceptable clinical values and confirms the fitness for real-world applications.

Table 4. Model performance on 10-fold cross-validation

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
70:30 Train-Test Split	96.8	97.1	96.3	96.7
10-Fold Cross-Validation	97.5	97.8	97.2	97.5

Table 5. Computational assessment of the MS-Swin-T

Model	Training Time (hrs)	Inference Time (ms)	Memory Consumption (GB)
MS-Swin-T	12	142	16
Baseline Swin Transformer	10	130	13
CNN-based Model	6	20	8
VGG16 Model	9	100	10
EfficientNet	8	90	13

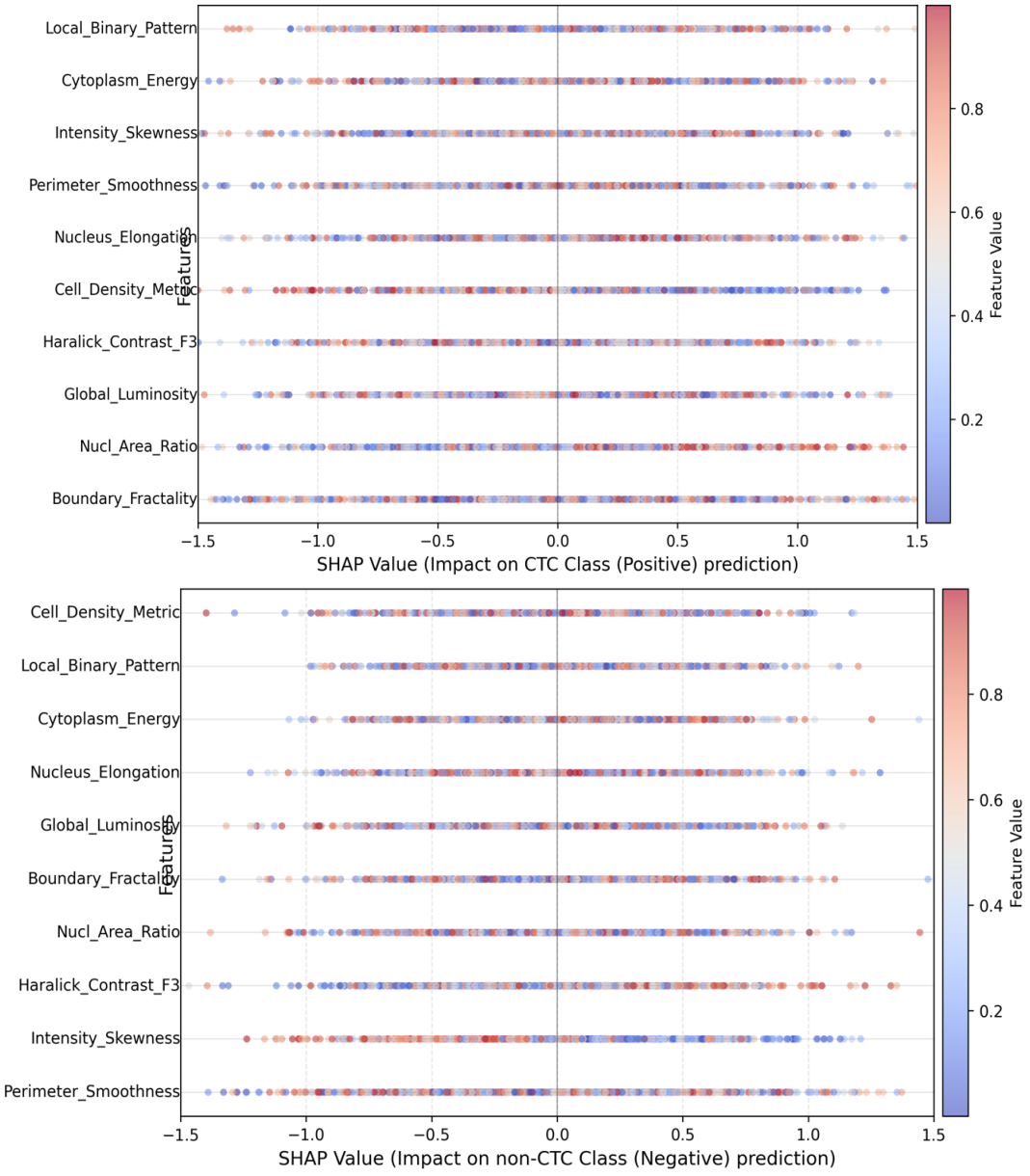


Figure 6. SHAP plot of the MS-Swin-T

To interpret the predictions of a proposed model, the SHAP (Shapley Additive exPlanations) summary of this model is plotted as shown in Figure 6. It shows the global impact of Each plot features a y-axis listing the various image-derived features and an x-axis representing the feature's SHAP Value. Here, 'Cell_Density_Metric' and 'Local_Binary_Pattern' are highly important, with high values (red dots) of both features strongly contributing to a high prediction for the non-CTC Class.

The statistical t-test is conducted to analyse the performance of the model. In Table 6, all p-values are less than 0.05 which indicates that the differences between the proposed and the baseline models are statistically significant across all metrics.

Table 6. Statistical test analysis

Metric	Full Model	Baseline CNN Model	P-Value	Significance ($\alpha = 0.05$)
Accuracy	98.0	93.4	0.001	Significant
Precision	97.8	92.8	0.008	Significant
Recall	97.9	93.0	0.003	Significant
F1-Score	97.9	92.9	0.025	Significant
Sensitivity	97.5	92.3	0.001	Significant
Specificity	98.3	93.5	0.004	Significant
AUC	98.5	93.2	0.004	Significant

Table 7. Comparison with other models

Liang et al. [6]	CNN-RNN	91.8
Ciurte et al. [10]	Boosting Classifiers	92
Guo et al. [7]	Novel Convolutional Neural Network (CNN)	92
Yanagisawa et al. [8]	VGG16 model	90
Soto-Ayala et al. [9]	Convolutional Neural Networks and Transfer Learning	93
Kohei	Improved Alexnet.	91
Chen et al. [23]	Hybrid Resnet and Densenet	93
Rao et al. [16]	MobilenetV3-ShufflenetV2	93.8
Alexander et al. [12]	Mask-RCNN	94
Park [11]	CNN+SVM	93
Batool and Byun [21]	EfficientNetB3 Model Based on Depthwise Separable Convolutions	95
Krishna Prasad et al. [24]	Fine Optimal Deep Convolution Residual Network (Fine Optimal DCRNet)	94
Wang et al. [18]	Dual-Channel Convolutional Block Attention Network	95
Fang et al. [19]	Deep Integrated Feature Fusion (DIFF) block-based DL	95.7
Üzen and Fırat [20]	Swin Transformer with Convmixer	95.8
Proposed	MS-Swin-T + AO + EBM	98

The performance comparison of the proposed model with other models is given in Table 7. A CNN-RNN hybrid model achieved an accuracy of 91.8%. A novel CNN and the well-known VGG16 model demonstrated accuracies of 92% and 90%, respectively. CNNs with transfer learning show an accuracy of 93%. An improved AlexNet reached an accuracy of 91% and a hybrid of ResNet and DenseNet achieved 93%. The integration of MobileNetV3 and ShuffleNetV2 yielded an accuracy of 93.8%.

Mask-RCNN and ConvNeXt models both achieved an accuracy of 94. Combining CNNs with SVMs also resulted in a 93% accuracy. The adoption of efficient convolutional strategies, as seen in the EfficientNetB3 model using depthwise separable convolutions achieves a higher accuracy of 95%. Similarly, a fine- Optimal DCRNet reached 94%. The incorporation of attention mechanisms like Dual-Channel Convolutional Block Attention Network achieved 95% and innovative feature fusion techniques, such as the DIFF block-based deep learning approach further improved performance to 95.7%. More recent transformer-based models like the Swin Transformer combined with ConvMixer reached an accuracy of 95.8%. Notably, the proposed method, MS-Swin-T integrated with Aquila Optimizer (AO) and Ensemble Boundary Margin (EBM) significantly outperformed the other models and achieved a remarkable accuracy of 98%.

5. CONCLUSION

Detection of CTCs is critical for early cancer detection and treatment monitoring. Existing models failed to capture long-term dependencies and were not based on feature optimizations. In this work, a novel DL model is proposed to overcome the existing model drawbacks. It involves MS-Swin-T based feature extraction and EBM based classifications. In addition, the feature optimization is carried out with Aquila Optimizer. MS-Swin-T is used to extract local and global spatial dependencies of CTC images. The use of EBM further increases the classification accuracy of CTC.

REFERENCES

[1] Aceto, N., Bardia, A., Miyamoto, D.T., Donaldson, M.C., et al. (2014). Circulating tumor cell clusters are oligoclonal precursors of breast cancer metastasis. Cell, 158(5): 1110-1122. https://doi.org/10.1016/j.cell.2014.07.013

[2] Pantel, K., Alix-Panabières, C. (2010). Circulating tumour cells in cancer patients: Challenges and perspectives. Trends in Molecular Medicine, 16(9): 398-406. https://doi.org/10.1016/j.molmed.2010.07.001

[3] Horimoto, Y., Tokuda, E., Murakami, F., Uomori, T., et al. (2018). Analysis of circulating tumour cell and the epithelial mesenchymal transition (EMT) status during eribulin-based treatment in 22 patients with metastatic breast cancer: A pilot study. Journal of Translational Medicine, 16(1): 287. https://doi.org/10.1186/s12967-018-1663-8

[4] Lannin, T.B., Thege, F.I., Kirby, B.J. (2016). Comparison and optimization of machine learning methods for automated classification of circulating tumor cells. Cytometry Part A, 89(10): 922-931. https://doi.org/10.1002/cyto.a.22993

[5] Tsuji, K., Lu, H., Tan, J.K., Kim, H., Yoneda, K., Tanaka, F. (2020). Detection of circulating tumor cells in fluorescence microscopy images based on ANN classifier. Mobile Networks and Applications, 25(3): 1042-1051. https://doi.org/10.1007/s11036-018-1121-0

[6] Liang, G., Hong, H., Xie, W., Zheng, L. (2018). Combining convolutional neural network with recursive neural network for blood cell image classification. IEEE Access, 6: 36188-36197. https://doi.org/10.1109/ACCESS.2018.2846685

[7] Guo, Z., Lin, X., Hui, Y., Wang, J., Zhang, Q., Kong, F. (2022). Circulating tumor cell identification based on deep learning. Frontiers in Oncology, 12: 843879.

https://doi.org/10.3389/fonc.2022.843879
[8] Yanagisawa, K., Toratani, M., Asai, A., Konno, M., et al. (2020). Convolutional neural network can recognize drug resistance of single cancer cells. International Journal of Molecular Sciences, 21(9): 3166. https://doi.org/10.3390/ijms21093166
[9] Soto-Ayala, L.C., Cantoral-Ceballos, J.A. (2021). Automatic blood-cell classification via convolutional neural networks and transfer learning. IEEE Latin America Transactions, 19(12): 2028-2036. https://doi.org/10.1109/TLA.2021.9480144
[10] Ciurte, A., Selicean, C., Soritau, O., Buiga, R. (2018). Automatic detection of circulating tumor cells in darkfield microscopic images of unstained blood using boosting techniques. PloS One, 13(12): e0208385. https://doi.org/10.1371/journal.pone.0208385
[11] Park, J., Ha, S., Kim, J., Song, J.W., Hyun, K.A., Kamiya, T., Jung, H.I. (2024). Classification of circulating tumor cell clusters by morphological characteristics using convolutional neural network-support vector machine. Sensors and Actuators B: Chemical, 401: 134896. https://doi.org/10.1016/j.snb.2023.134896
[12] Alexander, E., Leong, K.W., Laine, A.F. (2021). Automated multi-process CTC detection using deep learning. arXiv preprint arXiv:2109.12709. https://doi.org/10.48550/arXiv.2109.12709
[13] Shehta, A.I., Nasr, M., El Ghazali, A.E.D.M. (2025). Blood cancer prediction model based on deep learning technique. Scientific Reports, 15(1): 1889. https://doi.org/10.1038/s41598-024-84475-0
[14] Kisanuki, K., Guangxu, L., Kamiya, T. (2022). Classification of CTC on fluorescence image based on improved Alexnet. In 2022 22nd International Conference on Control, Automation and Systems (ICCAS), Jeju, Korea, pp. 550-553. https://doi.org/10.23919/ICCAS55662.2022.10003905
[15] Chen, H., Liu, J., Hua, C., Feng, J., Pang, B., Cao, D., Li, C. (2022). Accurate classification of white blood cells by coupling pre-trained ResNet and DenseNet with SCAM mechanism. BMC Bioinformatics, 23(1): 282. https://doi.org/10.1186/s12859-022-04824-6
[16] Rao, B.S.S., Rao, B.S. (2023). An effective WBC segmentation and classification using MobilenetV3–ShufflenetV2 based deep learning framework. IEEE Access, 11: 27739-27748. https://doi.org/10.1109/ACCESS.2023.3259100
[17] Krishna Prasad, P.R., Reddy, E.S., Chandra Sekharaiah, K. (2024). An intelligent white blood cell detection and multi-class classification using fine optimal DCRNet. Multimedia Tools and Applications, 83(31): 75825-75853. https://doi.org/10.1007/s11042-024-18455-x
[18] Wang, Z., Zheng, R., Zhu, X., Luo, W., He, S. (2024). Classification of bone marrow cells based on dual-channel convolutional block attention network. IEEE Access, 12: 96205-96219. https://doi.org/10.1109/ACCESS.2024.3427320
[19] Fang, M., Fu, M., Liao, B., Lei, X., Wu, F.X. (2024). Deep integrated fusion of local and global features for cervical cell classification. Computers in Biology and Medicine, 171: 108153. https://doi.org/10.1016/j.compbiomed.2024.108153
[20] Üzen, H., Fırat, H. (2024). A hybrid approach based on multipath Swin Transformer and ConvMixer for white blood cells classification. Health Information Science and Systems, 12(1): 33. https://doi.org/10.1007/s13755-024-00291-w
[21] Batool, A., Byun, Y.C. (2023). Lightweight EfficientNetB3 model based on depthwise separable convolutions for enhancing classification of leukemia white blood cell images. IEEE Access, 11: 37203-37215. https://doi.org/10.1109/ACCESS.2023.3266511
[22] Pacal, I., Kılıcarslan, S. (2023). Deep learning-based approaches for robust classification of cervical cancer. Neural Computing and Applications, 35(25): 18813-18828. https://doi.org/10.1007/s00521-023-08757-w
[23] Chen, W., Shen, W., Gao, L., Li, X. (2022). Hybrid loss-constrained lightweight convolutional neural networks for cervical cell classification. Sensors, 22(9): 3272. https://doi.org/10.3390/s22093272