A Transformer-based Algorithm for Automatically Diagnosing Malaria Parasite in Thin Blood Smear Images Using MobileViT

Abdolreza Marefat
Islamic Azad University
Javad Hassannataj Joloudari (javad.hassannataj@birjand.ac.ir)
University of Birjand
Maryam Rastgarpour
Islamic Azad University

Research Article

Keywords: Malaria diagnosis, Vision transformers, Deep learning, MobileViT2, Convolutional Neural Networks

Posted Date: June 26th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3067927/v1

Abstract

Based on the report provided by the World Health Organization (WHO), malaria has proved to be a life-threatening disease whose cases reached 241 million in 2020 globally. However, diagnosing malaria in the early stages of infection can be very fruitful for ameliorating this disease. The standard way of diagnosing malaria is by examining the blood cell images by professionals. Despite medical technology development, this is not feasible in many underdeveloped areas due to the lack of such experts. Thus, researchers interested in computer-aided decision-making, specifically deep learning, have focused on atomizing the diagnosis of malaria recently. The performance of transformer-based models combined with convolutional neural networks motivated us to propose an approach based on MobileViT for atomizing the process of diagnosing malaria. To achieve this, the model was trained on blood cell images collected from a publicly available dataset. Evaluated on 27,560 samples, the proposed classifier achieves an accuracy of 98.37% on average using 10-fold cross-validation. Among 2756 test samples, the model achieves 34 false negatives at least and 48 ones at most. Due to the medical nature of our problem, this is significant because the model's miss-cases of actual positive malaria-infected samples are low, making the accuracy and recall of the model 98.37% and 98.38%, respectively. To our knowledge, this is the first study that applies a transformer-based model to a problem with superior performance. In addition, it is a lightweight and mobile-friendly neural network which can be utilized in mobile applications.

1. Introduction

Malaria is a dangerous illness that is caused by a parasitic protozoan of the genus Plasmodium parasites. It can be transmitted through the bites of infected female mosquitos and can cause severe diseases in human beings [1, 2]. The known five variants of malaria are P.falciparum, P.vivax, P.malariae, P.ovale, and P.knowlesi, among which P.falciparum is the most menacing one [2]. According to the World Health Organization (WHO), the number of diagnosed malaria infection cases worldwide reached 241 million in 2020 and the estimated number of death cases stood at 627000 in the same year [3].

Moreover, although malaria can cause health-related severe ramifications for infected people, it is utterly curable given that it is diagnosed in the early stages of infection [2]. To do so, the most reliable modality is to manually examine the microscopic slides. This approach is often not feasible due to the necessity of specialized human resources, which is a common problem in many underdeveloped regions where malaria is considerably more dominant compared with other areas [4]. These problematic scenarios have sparked researchers' interest to propose automatic approaches for malaria diagnosis. Specifically, deep learning (DL)-based algorithms have been amongst the most widely-used techniques employed in the literature [5]. DL is a sub-domain in machine learning inspired by the biological brain and imitates its information processing functionality by amalgamating a massive number of computational units called neurons. These neurons reside in a set of stacked layers, which are famously called Deep Neural Networks (DNN) [6, 7]. In recent years, different architectures have been advanced by researchers. Amidst these architectures, the Convolutional Neural Networks (CNN) have proved to be significantly effective in

pattern recognition and computer vision [8, 9]. CNNs belong to a specific class of DNNs whose main advantage over the common DNNs is their capability to reduce the computational cost by shared learnable weights and also their robustness to translation variance. These have made CNNs an excellent choice for processing images in various machine learning fields [7]. However, more recently, Vision Transformer (ViT)-based models, have claimed to be more powerful in a variety of visual challenges such as image classification, content-based image retrieval, instance segmentation, and image recognition. Unlike models based on CNNs, ViT models learn representations in shallow and deep layers, with more similarities between them. Another advantage of ViT models over CNN ones is the fact that intermediate representations learned by ViT models, in case of the existence of large amounts of data, is better in terms of quality. In addition, ViT-based models are more informative of the spatial structure of the input data than CNNs and this makes them a better choice for harnessing the global pattern of the data.

Our study proposes a transformer-based model for classifying infected cell image samples from uninfected ones. Our algorithm is based on MobileViT. It is a lightweight model which utilizes the might of self-attention in its architecture. To the best of our knowledge, this is the first study that applies vision transformers for malaria-infected blood cell image classification and achieves superior performance compared with previous works. Our main contributions are itemized as follows:

- We aim to demonstrate the effectiveness of a proposed deep learning-based approach based on MobileViT for predicting malaria infection from the red blood cells image samples.
- The proposed algorithm is lightweight yet performant and it can be embedded in mobile-based applications.
- We prove the reliability of the classifier by evaluating the model on a publicly available dataset and achieving high accuracy of 98.37.
- Grad-CAM method of visualization is applied so as to have a complete understanding of the proposed method’s attention modules’ behavior.

The rest of the paper is outlined as the following. Section 2 reviews different approaches put forward in the literature. Section 3 elaborates the details of the proposed framework. Section 4 demonstrates our achieved results and explains the discussion. Finally, section 5 includes the conclusion and future work.

2. Related work

Hitherto, a considerable number of approaches, based on deep neural networks, have been utilized for malaria diagnosis from smear images. In this section, we present a summary of such works.

To begin with, deep learning-based algorithms have shown significant performance in a variety of health-related domains [10]. Thus, researchers have been interested in applying CNN models to diagnose malaria. Liang et al. [11] adopted a transfer learning-based approach to diagnose malaria. Their proposed custom CNN model to automatically classify thin blood smear image cells showed better results by achieving 97.37% accuracy. In [12], they trained several CNN models and opted for employing ResNet and

VGG16 as the best feature extractors. They used transfer learning and proved the efficiency of this technique as a powerful tool for malaria diagnosis. Rahman et al. investigated 3 different approaches (custom CNN, VGG16, and a CNN backbone integrated with the SVM classifier) for diagnosing malaria infection. They used 5-fold cross-validation in order to assess these models and concluded that data augmentation has a significant impact on the final output of the models. They achieved 96.29%, 97.77%, and 94.77% for the custom CNN, VGG16, and CNN + SVM, respectively. A CNN-based image classification method was created by Shah et al. [13] and evaluated using a dataset of labeled images. The accuracy of their results is 94.77%. They list pertinent computational resource limits and make the case that more powerful computers might enhance the results. By fusing ideas from dense and residual networks with attention processes, Quan et al. [14] developed a unique and compact CNN-based model. Attentive Dense Circular Net was the name of the suggested technique. The results were contrasted with similar research in the literature, such as DenseNet121 or DPN92. The findings demonstrate better performance with 97.47% as opposed to 90.94% and 87.88%, respectively, reported by DenseNet121 and DPN92. The suggested model also offers excellent accuracy and a quick rate of convergence.

In [15], the authors proposed an ensemble approach, in which ResNet is used as the backbone for malaria diagnosis. In their approach, they, fine-tuned the ResNet18 architecture and then introduced three randomly initialized networks, including a random vector functional link, Schmidt neural network, and extreme learning machine, as the classifier. They achieved 95.73%, 94.79%, 96.68%, and 95.69% for their accuracy, sensitivity, specificity, and F1-Score, respectively.

3. Methods

This section comprises a description of our methodology and the dataset used in this work. Further, the detailed evaluation of the proposed approach, including different metrics in classification tasks and attention visualization using the Grad-CAM method, is included. Grad-CAM can help us understand the inner workings of the classifier better.

3.1. Dataset

In this study, we have used a publicly available dataset1 provided by the USA National Institutes of Health (NIH). This dataset includes 27,558 thin blood smear images, half of which are positive samples meaning that they are infected or parasitized, and the other half are negative samples which means they are uninfected. In this dataset, the red blood cell micrographs were collected from Giemsa-stained thin blood smear slides [12]. The total number of patients, whose blood cells are used for this dataset, is 150 (100 infected individuals and 50 healthy ones). These people were treated at Chittagong Medical College Hospital, and each obtained sample was manually labeled in the Mahidol Oxford Tropical Medicine Research Unit in Bangkok by trained specialists. Figure 1 demonstrates some samples from the dataset.

3.2. Preprocessing

For the model to be properly trained on the images, we need to normalize them. Generally, a colorful image has three channels named Red, Green, and Blue (RGB). Each of these channels has an integer pixel intensity value between 0 (complete black) and 255 (complete white). Before feeding the input images to the model, we normalize the RGB values, in that they have a float value between 0.0 and 1.0.

Further, we have used Data Augmentation (DA) techniques for preventing the model from overfitting. Overfitting is the occurrence of a situation in which the model learns from the training data so well but cannot demonstrate satisfactory performance in unseen data [16]. In this case, the model starts to learn some intricate patterns in each sample that are not necessarily generalizable to the others [17]. By applying different transformations to the images, we randomly alter them, and this helps the model to be more robust to new cases and generalize better. Figure 2 depicts some of these techniques used in our training pipeline as a way of DA.

3.3. MobileNetV2

MobileNetV2 [18] is a novel deep neural network whose design is parameterized to be tailored in low resource and constrained environments while delivering high accuracy results. In addition to following the usage of Depth-wise Separable Convolution, introduced in [19], Sandler et al. improve the effectiveness of MobileNets by introducing Inverted Residuals and Linear Bottlenecks. These are elaborated in the following.

Depth-wise Separable Convolutional (DSC) networks are a type of convolutional network which have been ubiquitous in many applications [20–24]. DSC networks have two main privileges over the standard convolutional layers: 1) They contain less trainable parameters which should be optimized during the training phase, and consequently, this leads to a low chance of overfitting in the network. 2) They are less computationally intensive and require fewer computational resources [25]. Figures 3 (a) and (b) illustrates the difference between DSC and normal convolutional layers. As is observed in Fig. 3 (a), in normal convolutional layer *N* kernels of size $Dk * Dk * M$ is applied on a tensor of size $Di * Di * M$ and the output is $Do * Do * 1$, meaning that the convolution operation is done for all the existent channels. Meanwhile, in DSC (Fig. 3b) the convolution operation is applied to a single layer at each step, making the kernel shapes $Dk * Dk * 1$ for each one of the channels in input data. In addition to this, in the Point-wise convolution, $a1 * 1$ kernel is applied to each channel so as to sustain the required depth for the output tensor.

Furthermore, Inverted Residuals are in an inverted order of shapes of convolutional layers. A standard residual block has a wide -> narrow -> wide structure. In this type of structure, the input has a large number of channels which are further compacted with 1*1 convolution with the aim of being integrated or added. In inverted residuals, the structure is completely reversed, meaning that its order of layers is in narrow -> wide -> narrow style. In this architecture, the input is firstly widened by $1 * 1$ convolution, then a

$3 * 3$depthwise convolution, followed by $1 * 1$convolution layer, is used to reduce the number of channels. This is also done to add the input and the output.

Moreover, linear bottlenecks are introduced by the authors of [18] due to the challenge which inverted residual blocks mount. In linear bottlenecks, the last layer of the convolution has a linear output before being added to the initial activations.

3.4. Vision Transformer

Transformer-based models are amongst the most prominent deep neural networks which have been adopted in a wide range of applications such as Natural Language Processing (NLP), Computer Vision (CV), and Speech Recognition [26]. At the core of this family of architectures lies the transformer module, which uses the self-attention mechanism for processing the input. Transformers were initially introduced in [27] in the context of NLP for language translation and further applied for vision tasks [28] in [29].

The original ViT architecture is depicted in Fig. 4. As is observed in this figure, following the type of input in NLP which is always sequential and time-dependent, in ViT, we patchify an image so as to make it sequential data. Then, these smaller chunks of the input image, which play the role of tokens in NLP, are flattened in order to provide a sequence of linear embeddings which will be fed to the transformer. Later, the network learns to model global dependencies among these visual tokens with stacked transformer blocks shown in Fig. 4.

Although ViT-based models have become pervasive due to their state-of-the-art performance in a large number of challenges, they have some drawbacks. In stark contrast to CNN-based architectures, they are not equipped with inductive bias, which is intrinsic in CNNs [30]. Compared with CNNs, when the training data is small or mid-level size, ViTs deliver unsatisfactory results and overfitting. However, they show significant success in vision challenges when pre-trained on large datasets and fine-tuned for the different downstream tasks [29].

3.5. MobileViT Block

MobileViT Block is a component used in MobileViT's architecture that utilizes a transformer in a specific way. Mehta et al. have introduced this block and refer to them as Transformers as Convolutions. Given an input tensor with the shape $H * W * C$, an $n * n$ standard convolution is applied by MobileViT and followed by a point-wise convolution. This results in a tensor of shape $H * W * d$. Then, the tensor is split into non-overlapping patches [29] of size $h * w * d$. As a result, each of the patches becomes unfolded, forming intermediate-level embeddings of shape *P*N*d* where $P = w * h$ and $N = H * W/P$. Lastly, the transformer is applied to these embeddings. Figure 5 demonstrates this procedure.

3.6. Proposed Network

Our proposed algorithm is based on MobileViT which is a lightweight, general-purpose, and mobile-friendly transformer-based network. MobileViT belongs to a class of deep neural networks that follows the philosophy of lightweight CNNs. It combines the power of CNN-based networks in learning spatially local representations and Vision Transformers (ViT) in learning global features. The architecture of MobileViT is demonstrated in Fig. 6. As is seen in this figure, it comprises different components, which are elaborated in previous parts of this section.

3.7. Training

This section provides our training and evaluation procedure. Figure 7 presents all the steps in our approach.

As is seen in Fig. 7, in order to have a more reliable training and validation process for our proposed solution, we have used the cross-validation method. It is a statistical method for validating and comparing learning algorithms. In this method, we build two segments of the data, one for training the model and the other for validation of the model [31]. This method is employed such that each sample in the data has a chance of being validated. The most basic form of CV is K-fold Cross Validation (KCV), in which the data is split into K folds with approximately the same number of samples. Then, the model is trained on K-1 folds and evaluated on the remaining single fold. The main advantage of KCV is that our estimation of the model's performance can be more accurate and viable since the model's performance in classifying all the samples can be monitored and there is no. This means that the biasedness of the model can be revealed in the evaluation process. In this paper, we use 10 CV for splitting the dataset. Figure 8 depicts the procedure of splitting with 10 folds.

3.8. Evaluation

In this study, we have opted for different metrics for evaluating our proposed model. These metrics are itemized as follows:

- **Confusion Matrix**: It is a matrix that shows the summary of prediction results on a classification problem [32]. Figure 9 depicts one such matrix for a binary classification problem.

As is seen in Fig. 9, this matrix contains four indicators, namely True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). These are explicated as follows:

- TP is the number of samples that belong to the infected class and are correctly classified by the classifier as infected or positive.

- TN shows the number of samples without any infection and they are correctly classified as uninfected or negative by the classifier.
- FP is the number of samples that belong to the uninfected class but are wrongly classified as infected or positive by the classifier.
- FN is the number of samples from the infected class which are classified as uninfected or negative by the classifier.
- **Accuracy**: Accuracy is calculated by dividing TP plus TN by the total number of FPs, FNs, TPs, and TNs, as shown in (1).

$$(1) Acc = \frac{TP+TN}{FP+FN+TP+TN}$$

- **Precision**: As is shown in (2), precision is calculated by dividing TP by the total number of TPs and FPs [33].

$$(2) Precision = \frac{TP}{TP+FP}$$

- **Sensitivity or Recall**: It is also called True Positive Rate (TPR) [34] and is calculated using formula (3).

$$(3) sensitivity = \frac{TP}{TP+FN}$$

- **F1-Score**: It is the harmonic mean of precision and recall, as is shown in (4).

$$(4) F1 - Score = \frac{2TP}{2TP+FP+FN}$$

- **AUC-ROC**: Area Under Curve - Receiver Operator Characteristics (AUC-ROC) is one of the most important metrics used for evaluating any classification model. ROC is a curve depicting the probability of prediction by the classifier [35]. AUC is the area that is under this curve and signifies the separability power of the classifier. Its range is from 0 to 1, and the higher the AUC-ROC for a classifier, the better the classifier is at distinguishing the different classes.
- **Specificity**: It is also called True Negative Rate (TNR) and is calculated using formula (5).

$$(5) Specificity = \frac{TN}{TN+FP}$$

- **False Positive Rate**: It is calculated using formula (6).

$$(6) FPR = 1 - Specificity$$

- **False Negative Rate**: It is calculated using formula (7).

$$(7) FNR = 1 - sensitivity$$

1. https://lhncbc.nlm.nih.gov/LHC-publications/pubs/MalariaDatasets.html

4. Results and discussion

4.1. Experimental setup

The following tools, shown in Table 1, have been used for the implementation of our proposed algorithm.

Table 1
Experimental setup.

Programming language	**Python 3.7**
Deep learning library	Pytorch 1.9
CPU	Intel® Core™ i7-10700 CPU @ 2.90GHz × 16
GPU	GeForce GTX 1060

Further, we have trained the model for 128 epochs in each fold. The model uses a reduction algorithm for the learning rate of the optimizer named ReducedLROnPlateau for the epoch, after which there exists no improvement. This technique is required for pushing the learning boundaries of the model to a higher level, in that after a certain patience level (set to 10 in this study), when the model has stopped learning, the callback function provided in the implementation reduces the learning rate. The initial learning rate, the minimum learning rate, and all the other hyperparameters have been detailed in Table 2.

Table 2
Hyperparameters setting.

Image size	**256**
Batch size	32
Optimizer	Adam
Initial learning rate for optimizer	0.0001
Minimum learning rate for the optimizer	0.000001
Drop-out rate for transformer module	0.4
Number of attention heads	12
Head dimension	64
Patch size (width, height)	(4, 4)
Last layer	Sigmoid
Loss function	Binary Cross Entropy

We aim to demonstrate the efficiency of our proposed approach by performing extensive experiments and illustrating our results based on the previously introduced metrics. Additionally, we compare the performance of the proposed method with other recent works in the literature.

4.2. Classification performance

As stated before, we used the 10-fold cross-validation method to train and evaluate our proposed algorithm. In respect of training history, we have provided Fig. 10 and Fig. 11 to illustrate accuracy vs. epochs and loss vs. epoch curves, respectively.

Based on Fig. 10, the value of training accuracy starts from 87% in epoch #1 and climbs to 99% in epoch #128. Meanwhile, the value of validation accuracy in epoch #1 begins at 89% and after some oscillation ends at 99%. Furthermore, as is depicted in Fig. 11, the value of training loss and validation loss is 0.28 and 0.15 in epoch #1, respectively. In the last epoch, training loss decreases to 0.009, and the validation loss becomes 0.016. It is worth mentioning that, due to the adaptation of 10-fold cross-validation we have to obtain the average of the mentioned values for all 10 rounds of training.

Moreover, the confusion matrix for each 1 to 10-fold is demonstrated in Fig. 12 (a)-(j).

As is observed from Fig. 12 (a)-(j), the highest number of TPs happened in folds 5 to 10, correctly classifying 1354 samples. Meanwhile, the highest number of TNs happened in fold 10, where it achieved 1366 samples correctly classified as uninfected. The lowest number of FNs, 34, occurs in fold 10 and the highest in fold 8, reaching up to 48.

Moreover, Table 3 details our results in each fold and also on average.

Table 3
Classification results based on different metrics for each 10-fold.

Fold	Accuracy	Precision	Recall	F1-Score	TNR	FPR	FNR	AUC-ROC
1	98.15	98.15	98.16	98.15	97.50	2.5	1.18	99.91
2	98.44	98.44	98.46	98.44	97.46	2.54	0.59	99.90
3	98.04	98.04	98.05	98.04	97.29	2.71	1.18	99.85
4	98.00	98.00	98.02	98.00	97.21	2.79	1.18	99.83
5	98.62	98.63	98.64	98.62	97.43	2.57	0.15	99.91
6	98.66	98.67	98.68	98.66	97.50	2.50	0.15	99.81
7	98.37	98.39	98.39	98.37	96.93	3.07	0.15	99.85
8	98.19	98.21	98.21	98.19	96.57	3.43	0.15	99.89
9	98.55	98.56	98.57	98.55	97.29	2.71	0.15	99.84
10	98.69	98.70	98.71	98.69	97.57	2.43	0.15	99.82
Average	98.37	98.38	98.39	98.38	97.28	2.73	0.50	99.86

Table 3 shows that fold #10 is the best iteration that the model has done. Based on the metrics reported in this table, we can see that the model has delivered 98.69% for accuracy, 98.70% for precision, 98.39% for recall, and 98.69% for f1-score. The highest value for AUC-ROC happens in fold 9, where our classifier achieves 0.9984 for this metric. In addition, fold 4 is the worst fold, in which the model scores the lowest in terms of the target metrics.

Table 4 shows the prediction time of our proposed method for 32 (the chosen batch size) samples of cell images and its total number of trainable parameters.

Table 4
Inference time and the number of trainable parameters.

Number of Samples	32
Prediction Time (seconds)	0.089
Number of learnable parameters in the model	5,224,688

In Table 4, it can be observed that the time taken by the model for classifying 100 samples is 0.089 seconds. Unfortunately, we did not have access to the other research works' implementation so as to obtain their prediction time and this eliminates the opportunity for a detailed comparison between our proposed method and others concerning the models' complexity.

Furthermore, Fig. 13 illustrates the ROC curve for our proposed method.

4.3. Gradient-weighted Class Activation Mapping (Grad-CAM)

In order to further validate our proposed approach, we are interested in presenting Gradient-weighted Class Activation Mapping (Grad-CAM) [36] visualization for the model. Grad-CAM is a technique for making a large class of CNN-based models more explainable and transparent by using the gradients of different classes on which the model is trained to produce localization maps highlighting the salient regions of the input image. That is, by applying this technique, we can easily acquaint ourselves with the regions on which the model focuses when predicting a particular class. Figure 14 (a)-(d) shows some TP, TN, FP, and FN samples and their Grad-CAM output in two modes, namely heatmap and superimposed.

Based on Fig. 14 (a)-(d), we can see the attention of the model in different samples. Figure 14 (a) shows that the model mainly focuses on the stained parts of the sample image. This behavior is expected since the samples should be predicted as positive (infected). In Fig. 14 (b), it can be observed that the model has a holistic focus on the sample; thus, it predicts the negative samples correctly. In Fig. 14 (c) and Fig. 14 (d), the model finds some other regions in the images, leading to the wrong classification.

The world's most important parasitic infection is known to be malaria [37]. Malaria's eradication from temperate areas was due to the development of highly efficient, residual insecticide Dichlorodiphenyltrichloroethane, also known as DDT. Nevertheless, this success became halted on account of the costly operations and resistance of the societies to take hygienic measurements such as iteratively spraying their environments [38].

Malaria's symptoms are manifold, ranging from irregular fever, chills, headache, and malaise. However, these signs of infection might be hidden due to nearly two weeks of the incubation period, which is common in malaria-infected cases. An accurate diagnosis is made by examining the microscopic films of thin and thick blood cells, some of which may be stained in the case of an infected patient. These samples provide the scientific communities with the opportunity of adopting machine learning-based approaches to atomize the process of malaria diagnosis. For instance, in [39], Das et al. took the effort of collecting thin peripheral blood smear samples, and after some preprocessing steps, tried to extract and use image-based features with the object of feeding to Naive Bayes and SVM to classify the acquired samples. In another research work, Rosado et al. [40] presented an approach based on SVM to classify the parasitized and uninfected samples with the help of mixing geometric, color, and texture features. Although these works are effective to some extent, the obligation of obtaining attuned extracted features can be detrimental to the performance of algorithms. As a result, DL-based methods have paved the way to enhance the efficiency of such methodologies further.

In the era of DL, a variety of research works employed various deep architectures to atomize the procedure of diagnosing malaria. In Table 5, we have presented a review of these works in comparison to our own proposed approach. The central point of this comparison is that the power of transformer-based

architectures has not been investigated yet. Consequently, we are motivated to apply a ViT-based model to the problem at hand. The family of ViT architectures has proved to be more efficacious in most computer vision tasks, albeit with some requirements and necessities. The most vital factor in training such architectures is a large amount of data. This is a problem, especially in terms of malaria, since publicly available datasets with a sufficient number of samples are scarce. In order to address the issue, we opted for a customized version of MobileViT, which has the optimal need for data and uses both transformer and CNN for learning the patterns.

Our findings prove the effectiveness of our approach. First of all, the metrics, provided in the evaluation section, demonstrate that the model has consistency with regard to the recall and precision of the proposed classifier. Moreover, the comparison detailed in Table 5 shows that the approach has superior performance. Obtaining an F1-Score of 98.38 and an AUC value of 99.86 signifies the powerful separability made by the decision boundaries within the proposed deep classification model. Regarding the other metrics, FPR and FNR are deemed cardinal types of error in medical contexts [41]. As is depicted in Table 5, the FPR value for our proposed approach is 2.73 on average. Compared with other works, the proposed method has a better performance in terms of FNR. However, the FPR is slightly worse than [42] and [3].

In addition to this, the usage of the Grad-CAM visualization method depicts that the proposed model primarily attends to the stained areas in infected samples, proving its estimated behavior in that the model has a proper understanding of the infected cell samples. On the other hand, when the target sample is healthy and normal, the model behaves in a way to pay attention to the whole sample structure. Furthermore, the computational cost of the proposed method is thoroughly suitable for low-resource environments. This makes the approach more preferable since it can be embedded in mobile applications. As is shown in Table 5, this can be verified by the inference time of the model for 100 samples which are included in Table 5.

Table 5
Comparison of our proposed method with other research work.

Reference	Architecture	Train-Test	Accuracy	F1-Score	AUC-ROC
Liang et al. [11]	Custom CNN	10 CV	97.37	97.36	Not Reported
Rajaraman et al. [12]	ResNet50	5 CV	95.70	95.70	99.00
Rahman et al. [43]	VGG16	5 CV	97.77	97.09	99.38
Shah et al. [13]	Custom CNN	80 – 20	94.77	94.81	Not Reported
Quan et al. [14]	ADCN	5 CV	97.47	94.34	Not Reported
Yang et al. [42]	Custom CNN	5 CV	97.26	80.81	98.39
Marques et al. [3]	EfficientNetB0	5 CV	98.29	98.28	0.9976
Zhu et al. [15]	ResNet-Based Output Ensemble	5 CV	95.73	95.69	Not Reported
Proposed	MobileViT	10 CV	98.37	98.38	99.86

Furthermore, the advantages of our proposed method can be summarized as follows:

- Our proposed method has achieved superior performance over the other research works.
- The algorithm is evaluated using KCV, proving the stability of our results and preventing biasedness.
- DA techniques have been employed, and this helped the proposed method be more generalizable to other related datasets.
- Our method is based on MobileViT, meaning that it is completely tailored for low-resource mobile applications.
- The proposed method proves the usefulness of the ViT-based model on relatively small datasets of malaria cell images.
- Grad-CAM is used to verify further and validate the robust behavior of the model.

Our work has some limitations. Firstly, the approach is not evaluated in more datasets since, as mentioned before, such a vast dataset is not accessible to us. Secondly, the training process can be done better. The frequent oscillation of accuracy and loss curves during training and validation supposedly originates from the small batch size, and this is because we do not have access or essential funding for better hardware such as GPU.

The limitations of our proposed algorithm can be summarized as follows:

- Due to the scarcity of publicly available datasets, our approach has not been yet evaluated on more realistic datasets.
- The lack of access to more powerful hardware, especially GPU, spoils the quality of the training process.

5. Conclusion and future work

This work presented a deep learning-based method for automatically classifying blood cell images that are either infected by malaria or uninfected. This study aims to alleviate the labor-intensive and time-consuming conundrum of manually diagnosing malaria which is frequently done by trained practitioners. Our proposed algorithm can facilitate such procedures, especially because it can be embedded in mobile-friendly environments which are not rich in terms of computational resources. An extensive evaluation of our proposed network proves its outstanding performance regarding the accuracy of 98.38%. In future studies, we aim to evaluate our method on larger and more realistic datasets with closer distribution to real-world scenarios which happen in the laboratory.

Declarations

Author Contributions

JHJ designed the study. AM performed the implementation of the approach. AM and JHJ performed the literature review and the methodology. AM and JHJ did the discussion. AM, JHJ, and MR edited the final version of the article. MR supervised the project. JHJ co-supervised the study. All authors have read and approved the final manuscript.

Funding

None.

Data availability and access

The dataset which is used in this study is publicly available at https://lhncbc.nlm.nih.gov/LHC-publications/pubs/MalariaDatasets.html

Ethics approval

Not applicable.

Consent to participate

Not applicable.

Competing Interests

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. May Z, Aziz SSAM (2013) Automated quantification and classification of malaria parasites in thin blood smears. In: 2013 IEEE International Conference on Signal and Image Processing Applications, IEEE, pp 369–373
2. Vijayalakshmi A (2020) Deep learning approach to detect malaria from microscopic images. Multimedia Tools and Applications 79(21):15297–15317
3. Marques G, Ferreras A, de la Torre-Diez I (2022) An ensemble-based approach for automated medical diagnosis of malaria using EfficientNet. Multimedia tools and applications :1–18
4. Dong Y, Jiang Z, Shen H, Pan WD, Williams LA, Reddy VV, Benjamin WH (2017) Bryan AW Evaluations of deep convolutional neural networks for automatic identification of malaria infected cells. In: 2017 IEEE EMBS international conference on biomedical & health informatics (BHI), IEEE, pp 101–104
5. Poostchi M, Silamut K, Maude RJ, Jaeger S, Thoma G (2018) Image analysis and machine learning for detecting malaria. Translational Res 194:36–55
6. Caterini AL, Chang DE (2018) Deep neural networks in a mathematical framework. Springer
7. Sriporn K, Tsai C-F, Tsai C-E, Wang P (2020) Analyzing malaria disease using effective deep learning approach. Diagnostics 10(10):744
8. Abiodun OI, Jantan A, Omolara AE, Dada KV, Umar AM, Linus OU, Arshad H, Kazaure AA, Gana U, Kiru MU (2019) Comprehensive review of artificial neural network applications to pattern recognition. IEEE Access 7:158820–158846
9. Feng X, Jiang Y, Yang X, Du M, Li X (2019) Computer vision algorithms and hardware implementations: A survey. Integration 69:309–320
10. Litjens G, Kooi T, Bejnordi BE, Setio AAA, Ciompi F, Ghafoorian M, Van Der Laak JA, Van Ginneken B, Sánchez CI (2017) A survey on deep learning in medical image analysis. Med Image Anal 42:60–88
11. Liang Z, Powell A, Ersoy I, Poostchi M, Silamut K, Palaniappan K, Guo P, Hossain MA, Sameer A (2016) Maude RJ CNN-based image analysis for malaria diagnosis. In: IEEE international conference on bioinformatics and biomedicine (BIBM), 2016. IEEE, pp 493–496
12. Rajaraman S, Antani SK, Poostchi M, Silamut K, Hossain MA, Maude RJ, Jaeger S, Thoma GR (2018) Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images. PeerJ 6:e4568
13. Shah D, Kawale K, Shah M, Randive S, Mapari R (2020) Malaria parasite detection using deep learning:(Beneficial to humankind). In: 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), IEEE, pp 984–988
14. Quan Q, Wang J, Liu L (2020) An effective convolutional neural network for classifying red blood cells in malaria diseases. Interdisciplinary Sciences: Computational Life Sciences 12(2):217–225

15. Zhu Z, Wang S, Zhang Y (2022) ROENet: A ResNet-Based Output Ensemble for Malaria Parasite Classification. Electronics 11(13):2040
16. Dietterich T (1995) Overfitting and undercomputing in machine learning. ACM Comput Surv (CSUR) 27(3):326–327
17. Ying X (2019) An overview of overfitting and its solutions. In: Journal of physics: Conference series, vol 2. IOP Publishing, p 022022
18. Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L-C (2018) Mobilenetv2: Inverted residuals and linear bottlenecks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4510–4520
19. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:170404861
20. Chollet F, Xception (2017) : Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1251–1258
21. Kaiser L, Gomez AN, Chollet F (2017) Depthwise separable convolutions for neural machine translation. arXiv preprint arXiv:170603059
22. Bai L, Zhao Y, Huang X (2018) A CNN accelerator on FPGA using depthwise separable convolution. IEEE Trans Circuits Syst II Express Briefs 65(10):1415–1419
23. Kamal K, Yin Z, Wu M, Wu Z (2019) Depthwise separable convolution architectures for plant disease classification. Comput Electron Agric 165:104948
24. Khan ZY, Niu Z (2021) CNN with depthwise separable convolutions and combined kernels for rating prediction. Expert Syst Appl 170:114528
25. Zhang R, Zhu F, Liu J, Liu G (2019) Depth-wise separable convolutions and multi-level pooling for an efficient spatial CNN-based steganalysis. IEEE Trans Inf Forensics Secur 15:1138–1150
26. Lin T, Wang Y, Liu X, Qiu X (2021) A survey of transformers. arXiv preprint arXiv:210604554
27. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. Advances in neural information processing systems 30
28. Tay Y, Dehghani M, Bahri D, Metzler D (2020) Efficient transformers: A survey. ACM Computing Surveys (CSUR)
29. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, Dehghani M, Minderer M, Heigold G, Gelly S (2020) An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:201011929
30. Xu Y, Zhang Q, Zhang J, Tao D (2021) Vitae: Vision transformer advanced by exploring intrinsic inductive bias. Adv Neural Inf Process Syst 34:28522–28535
31. Refaeilzadeh P, Tang L, Liu H (2009) Cross-validation. Encyclopedia of database systems 5:532–538
32. Grandini M, Bagli E, Visani G (2020) Metrics for multi-class classification: an overview. arXiv preprint arXiv:200805756

33. Hossin M, Sulaiman MN (2015) A review on evaluation metrics for data classification evaluations. Int J data Min Knowl Manage process 5(2):1
34. Koyejo OO, Natarajan N, Ravikumar PK, Dhillon IS (2014) Consistent binary classification with generalized performance metrics. Advances in neural information processing systems 27
35. Bradley AP (1997) The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recogn 30(7):1145–1159
36. Selvaraju RR, Cogswell M, Das A, Vedantam R, Parikh D, Batra D (2017) Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE international conference on computer vision, pp 618–626
37. White NJ (1996) The treatment of malaria. N Engl J Med 335(11):800–806
38. Greenwood B, Mutabingwa T (2002) Malaria in 2002. Nature 415(6872):670
39. Das DK, Ghosh M, Pal M, Maiti AK, Chakraborty C (2013) Machine learning approach for automated screening of malaria parasite using light microscopic images. Micron 45:97–106
40. Rosado L, Da Costa JMC, Elias D, Cardoso JS (2016) Automated detection of malaria parasites on thick blood smears via mobile devices. Procedia Comput Sci 90:138–144
41. Abdar M, Zomorodi-Moghadam M, Das R, Ting I-H (2017) Performance analysis of classification algorithms on early detection of liver disease. Expert Syst Appl 67:239–251
42. Yang F, Poostchi M, Yu H, Zhou Z, Silamut K, Yu J, Maude RJ, Jaeger S, Antani S (2019) Deep learning for smartphone-based malaria parasite detection in thick blood smears. IEEE J biomedical health Inf 24(5):1427–1438
43. Rahman A, Zunair H, Rahman MS, Yuki JQ, Biswas S, Alam MA, Alam NB, Mahdy M (2019) Improving malaria parasite detection from red blood cell using deep convolutional neural networks. arXiv preprint arXiv:190710418

Figures

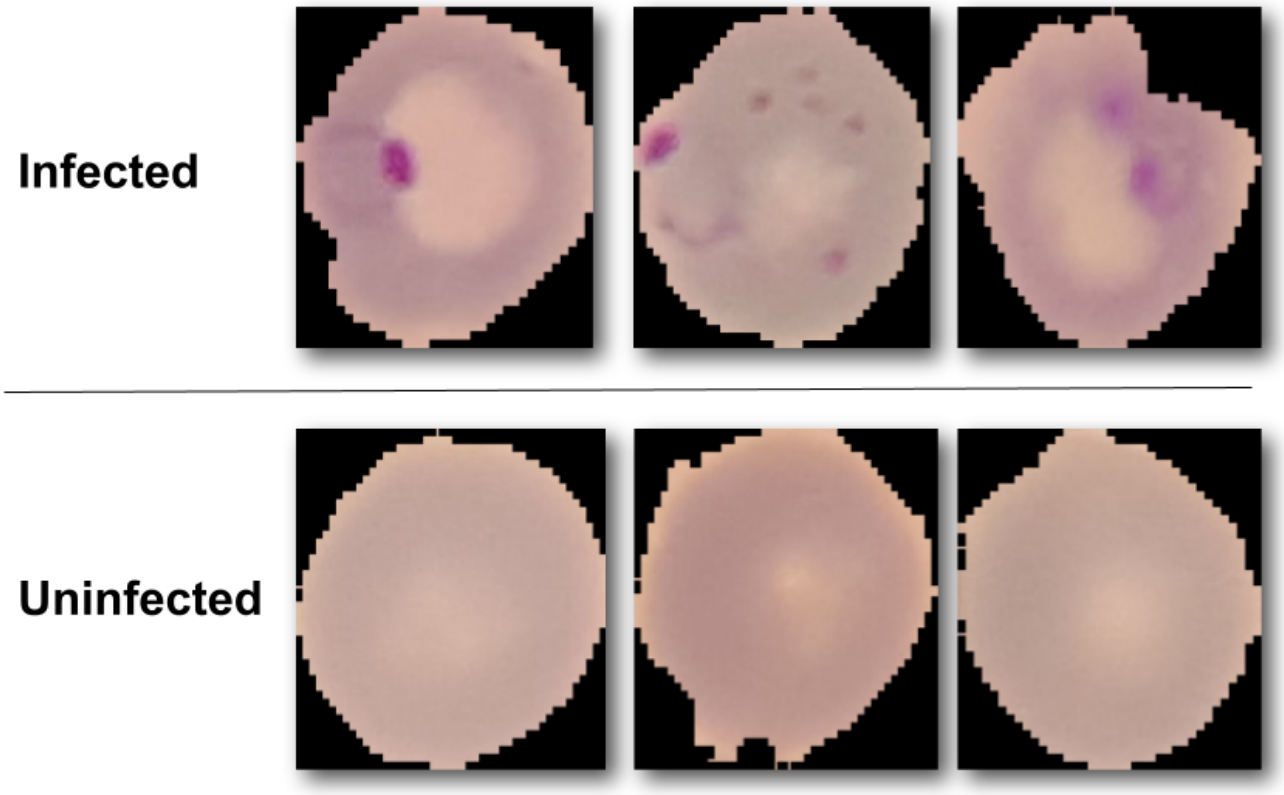

Figure 1

Samples of data from both classes.

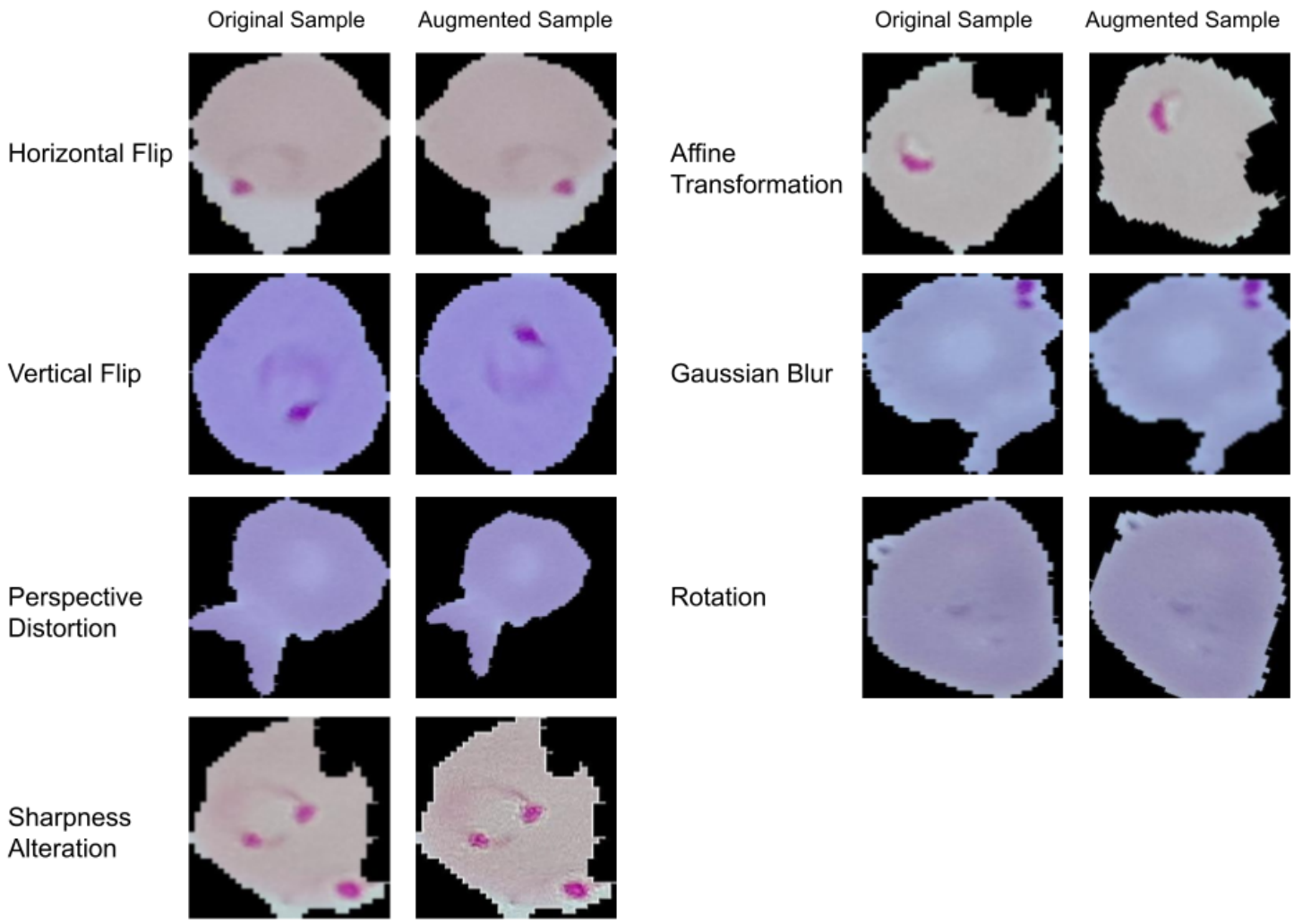

Figure 2

Data augmentation techniques.

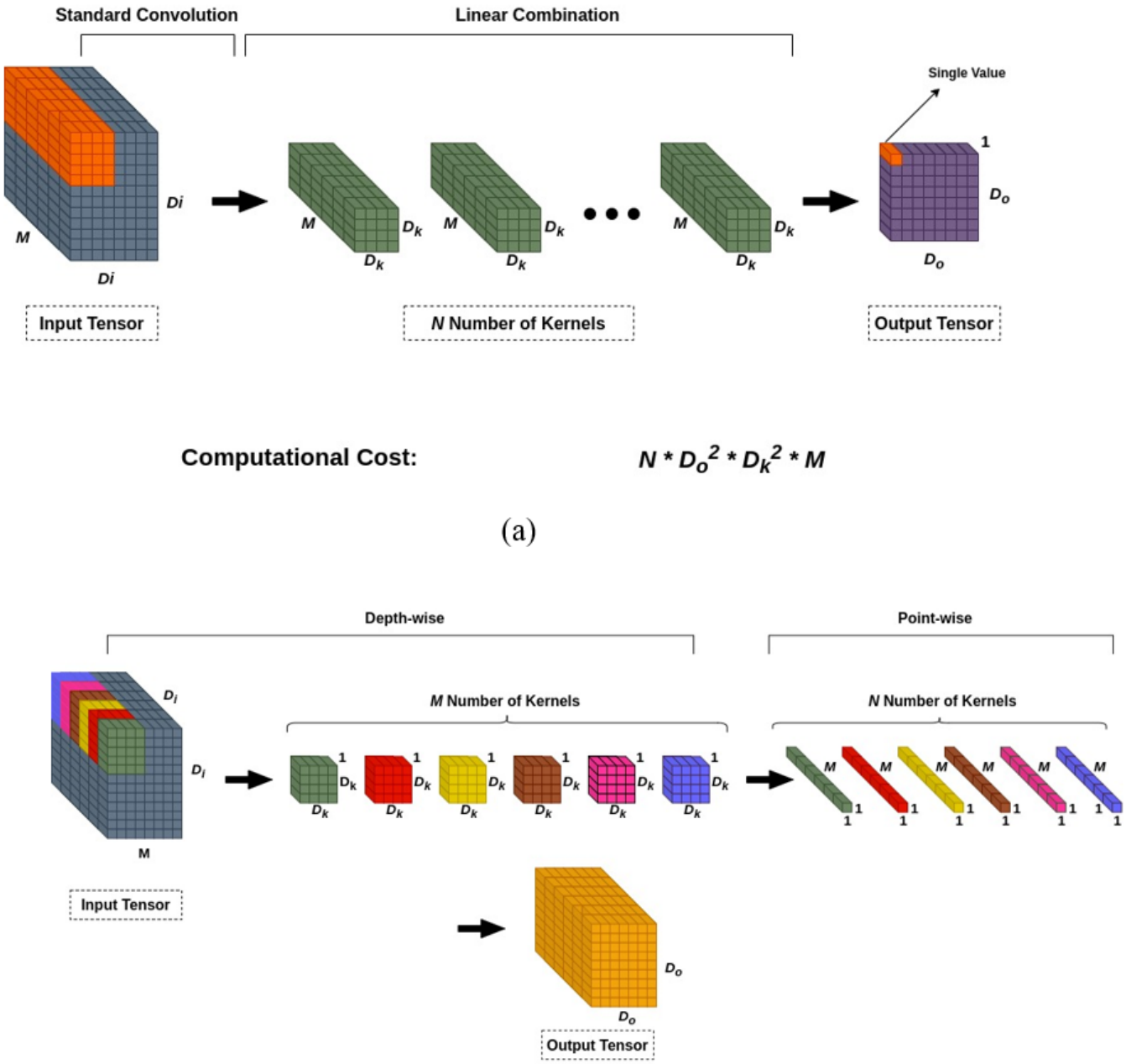

Figure 3

(a) Standard CNN, (b) DSC structure.

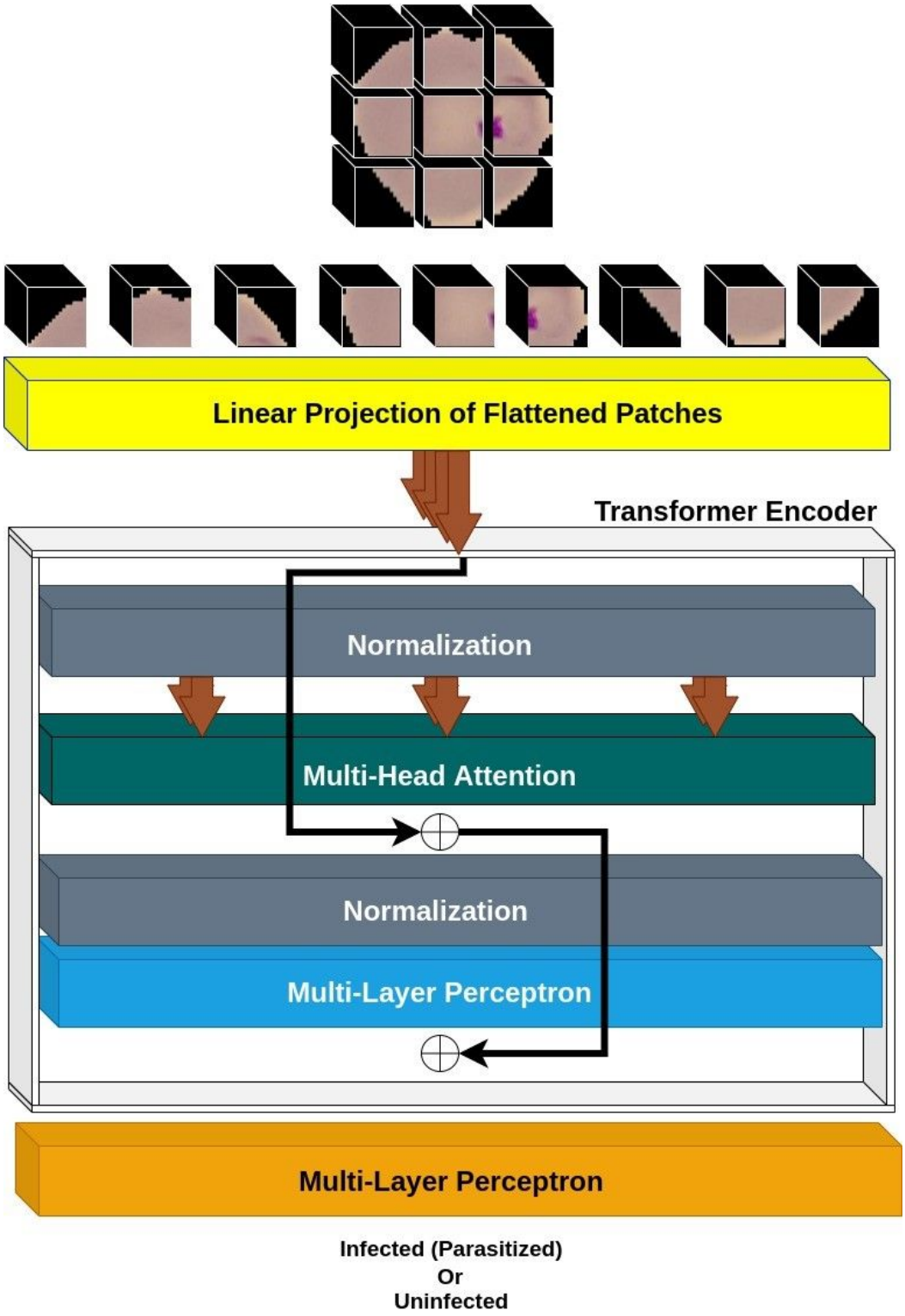

Figure 4

Structure of ViT base model.

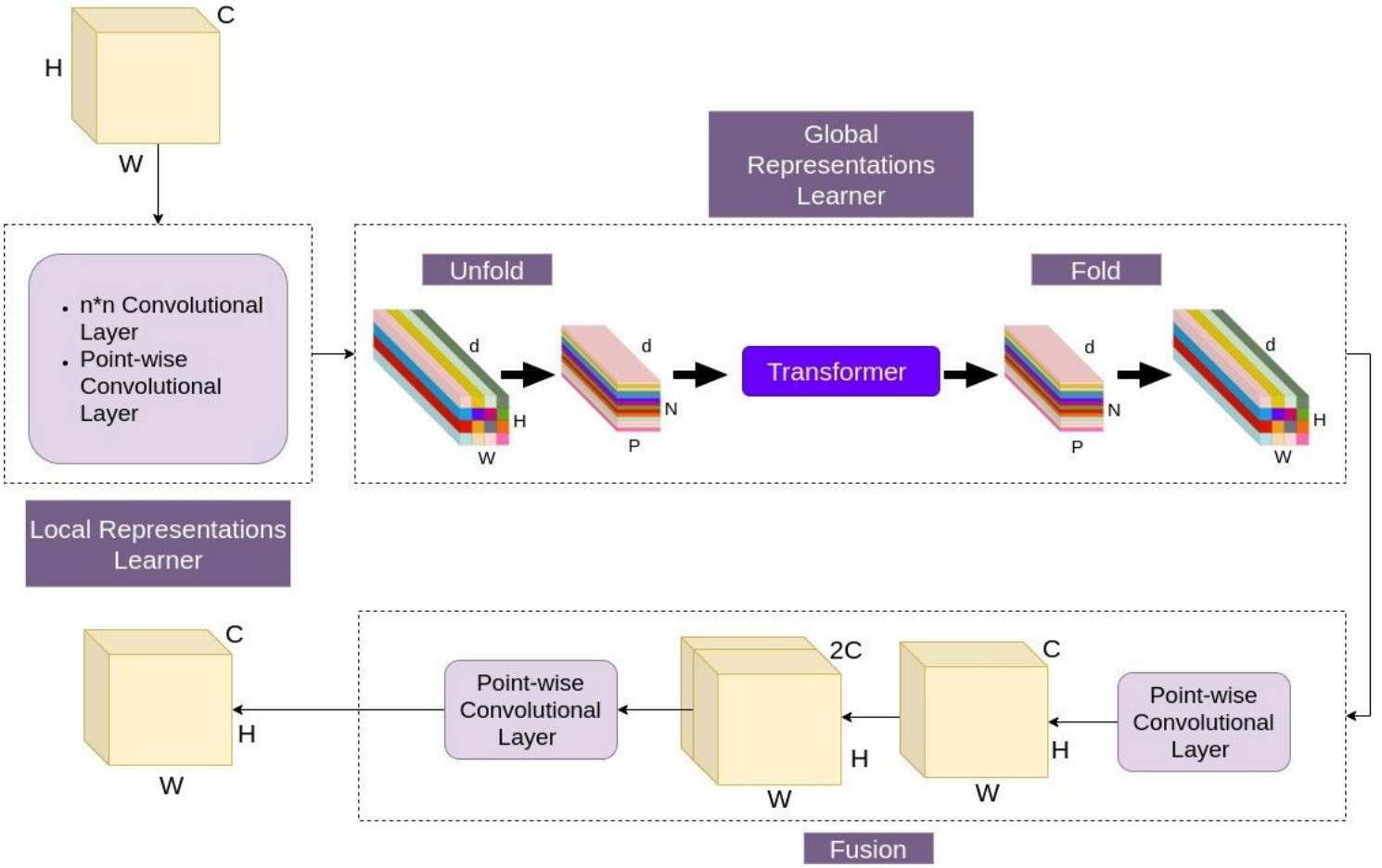

Figure 5

Structure MobileViT block.

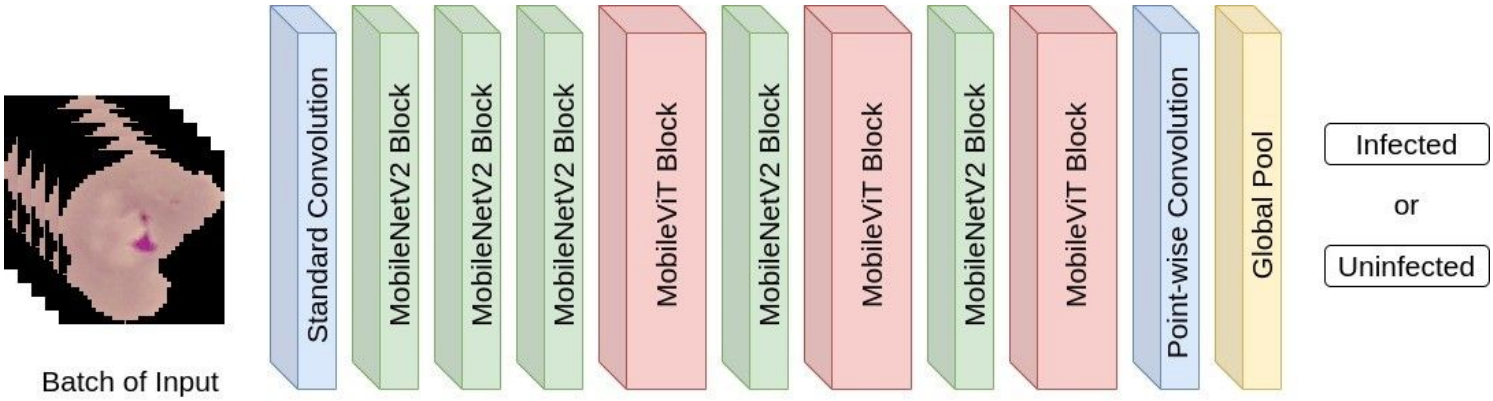

Figure 6

Architecture of MobileViT.

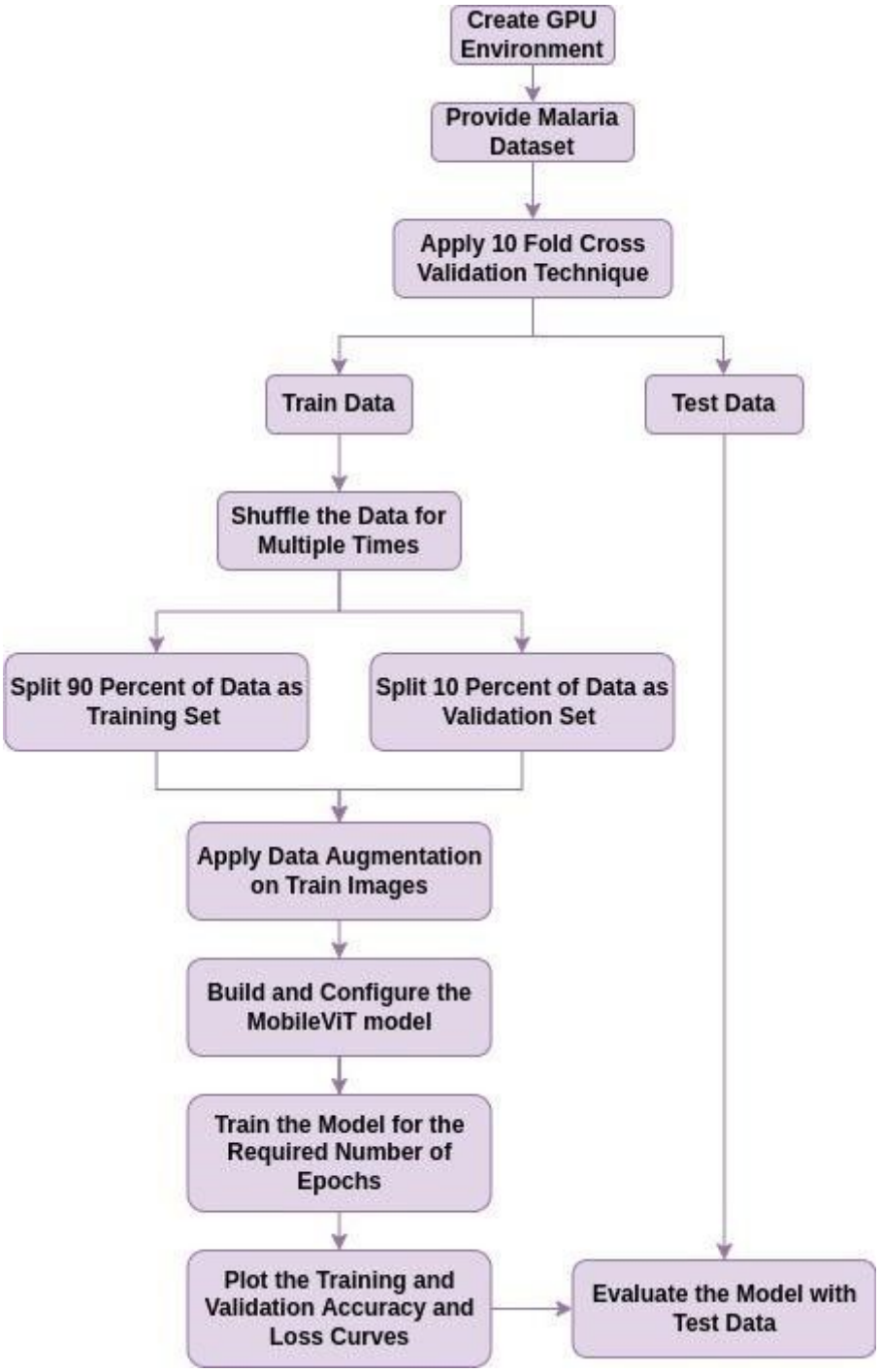

Figure 7

Pipeline of the proposed method.

Figure 8

10-fold cross-validation process of splitting training and testing data.

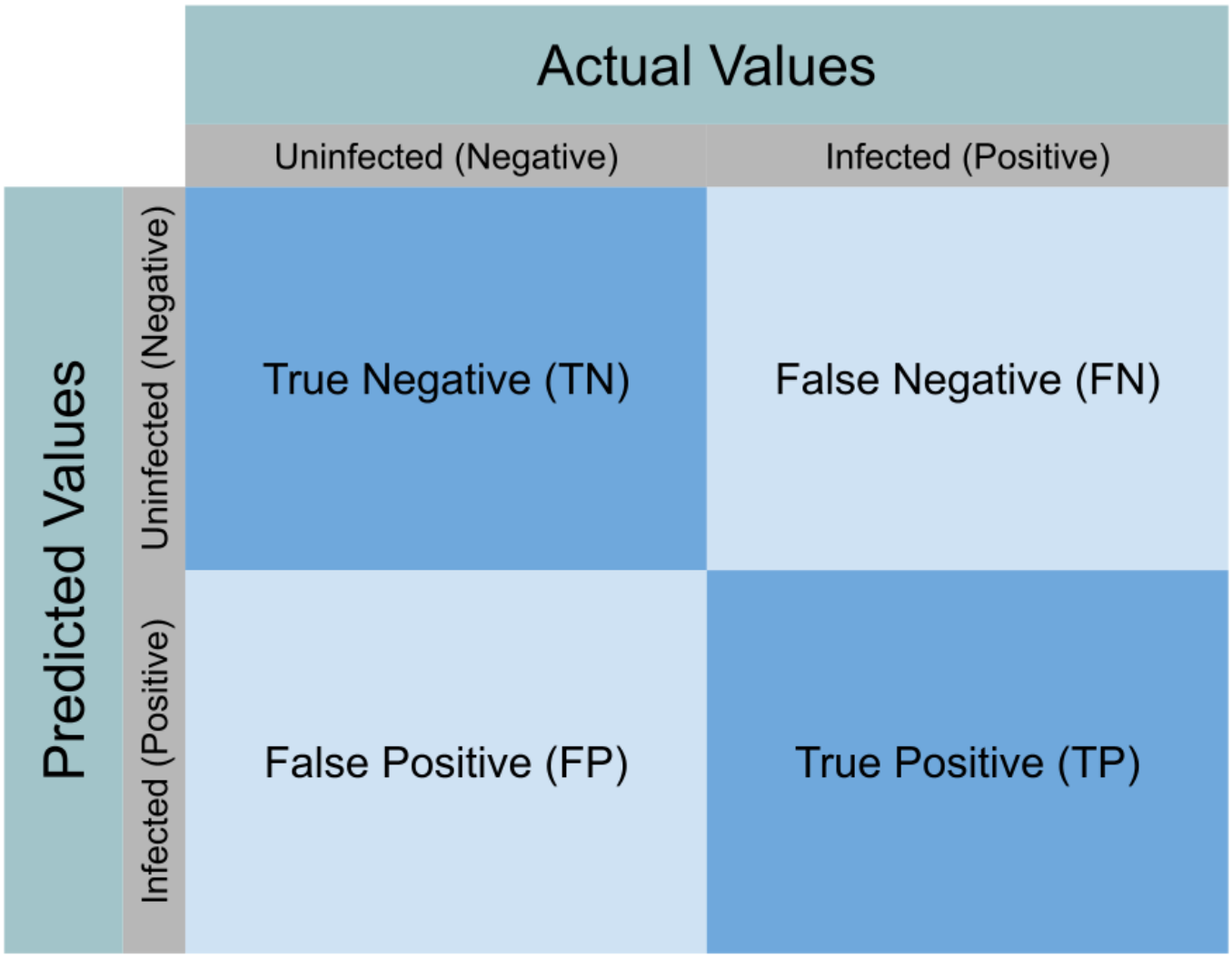

Figure 9

A confusion matrix.

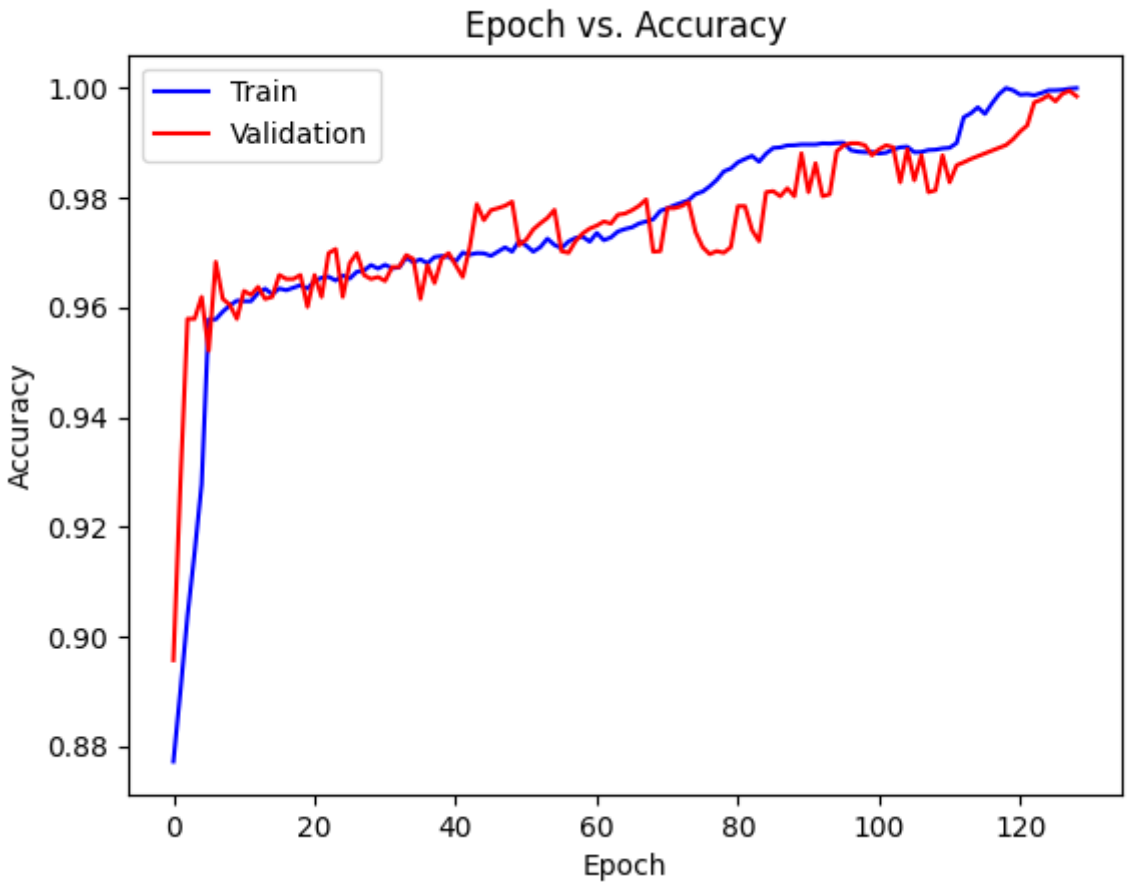

Figure 10

Training and validation accuracy vs. epoch.

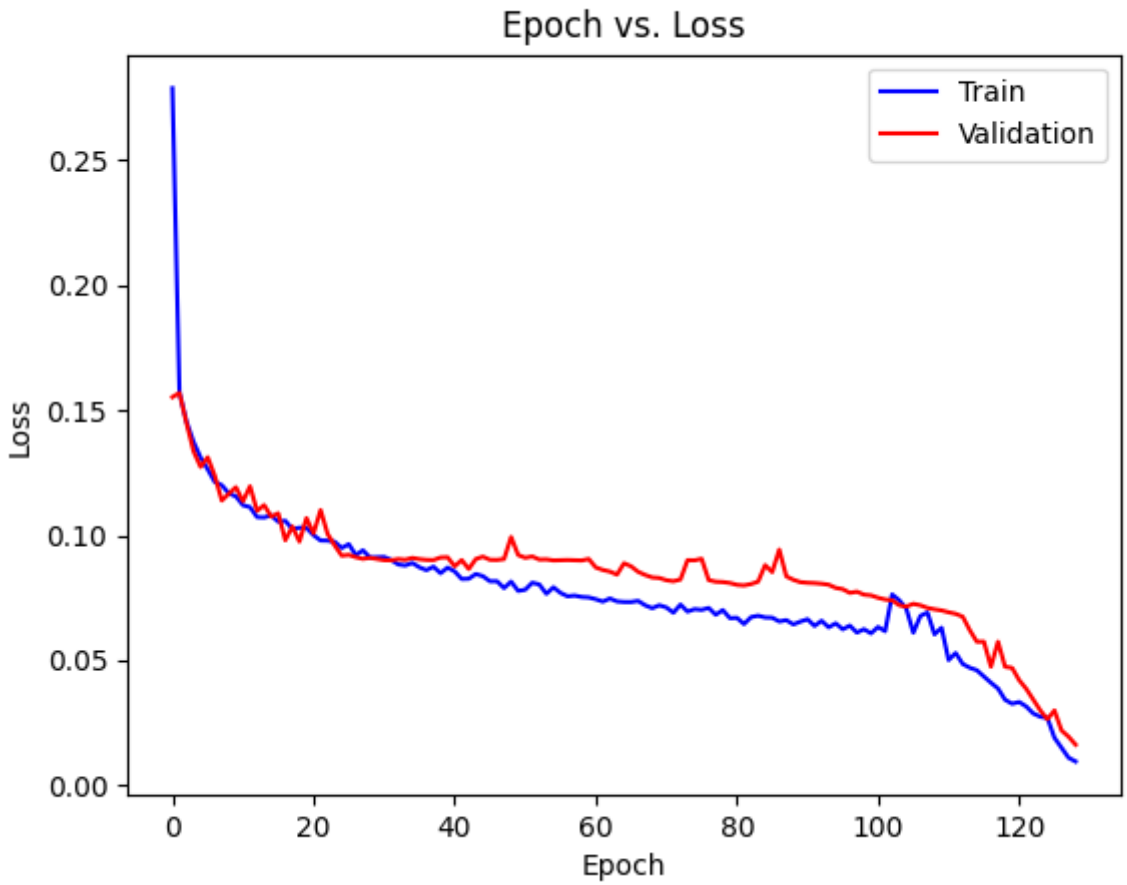

Figure 11

Training and validation loss vs. epoch.

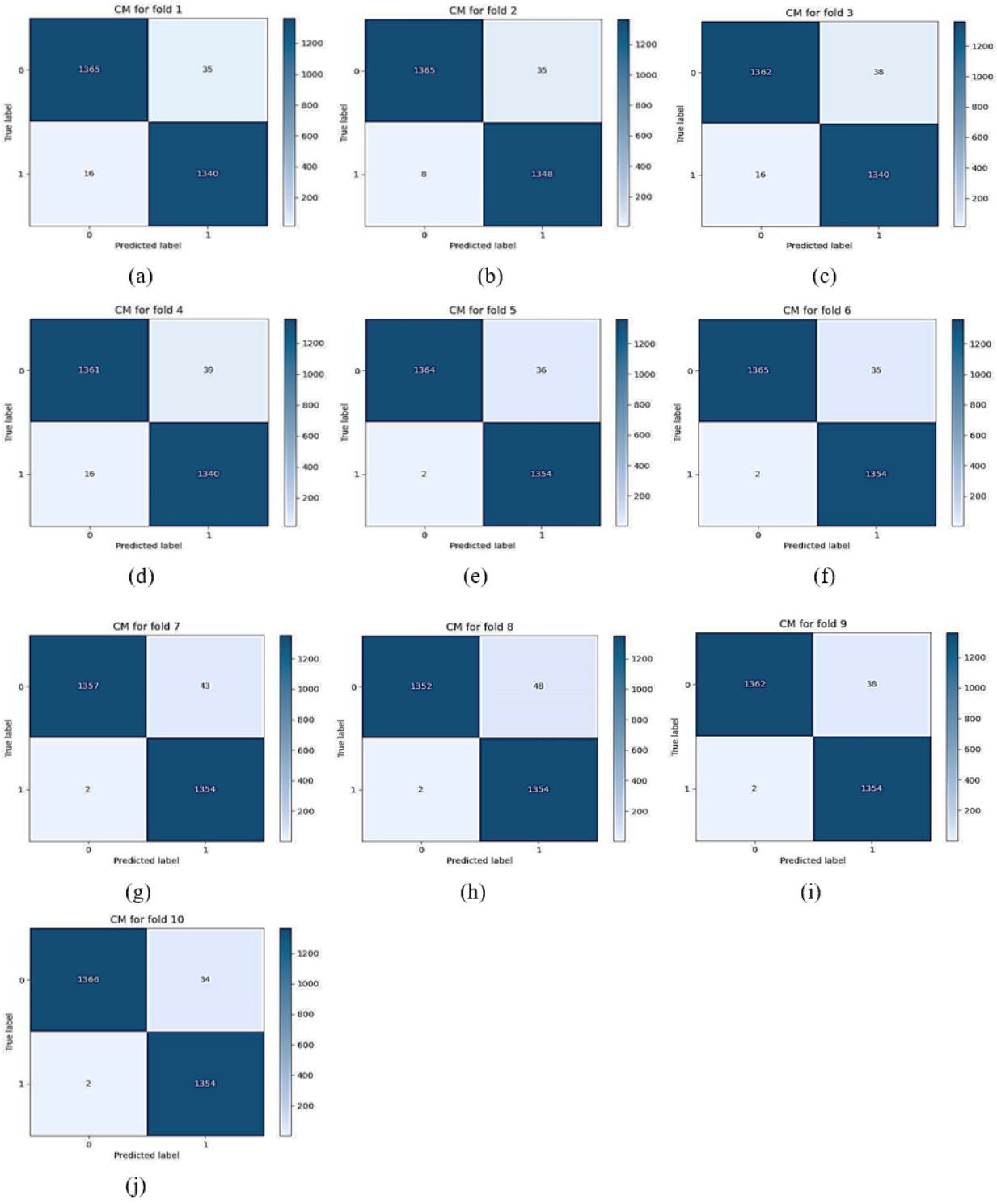

Figure 12

(a)-(j). Confusion Matrices for folds 1 to 10.

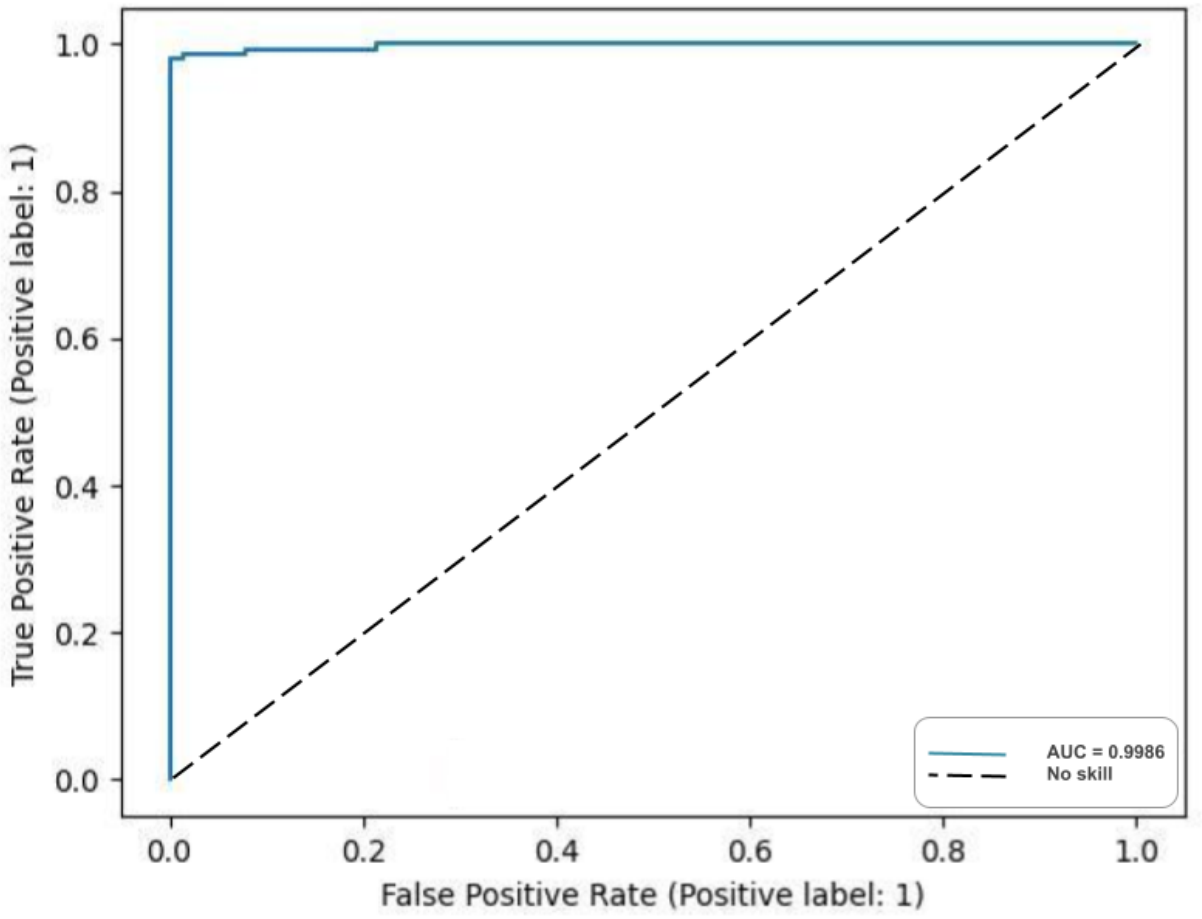

Figure 13

The ROC curve for the proposed model.

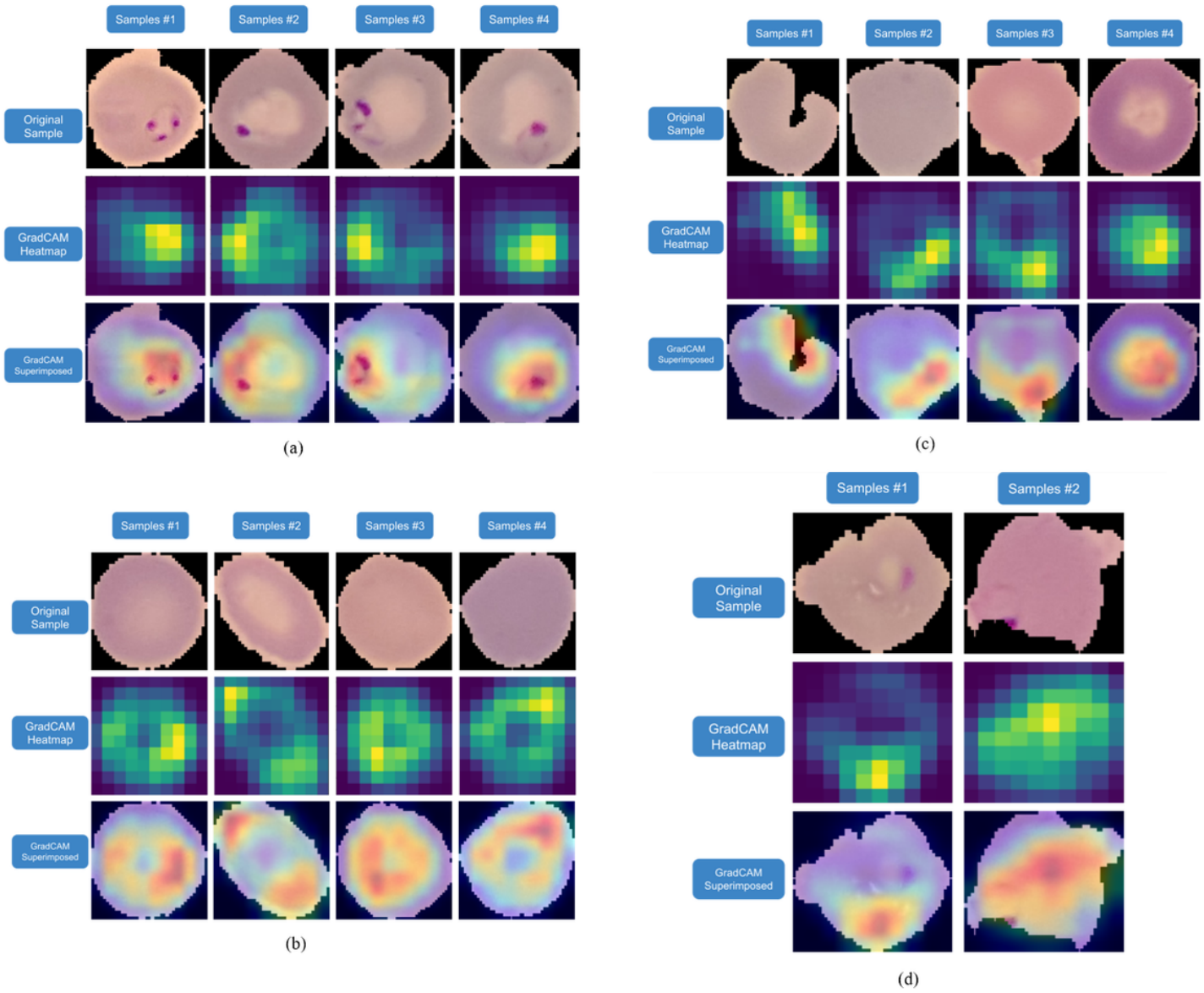

Figure 14

(a)-(d). (a) TP Samples, (b) TN Samples, (c) FP Samples, (d) FN Samples.