

Manuscript received June 17, 2024; revised August 10, 2024; accepted August 15, 2024; date of publication October 2024
Digital Object Identifier (DOI): <https://doi.org/10.35882/jeeemi.v6i4.484>

Copyright © 2024 by the authors. This work is an open-access article and licensed under a Creative Commons Attribution-ShareAlike 4.0 International License ([CC BY-SA 4.0](https://creativecommons.org/licenses/by-sa/4.0/)).

How to cite: Dilip Kumar Baruah, Kuntala Boruah, "Early Detection Of Canine Babesia From Red Blood Cell Images Using Deep Ensemble Learning", Journal of Electronics, Electromedical Engineering, and Medical Informatics, vol. 6, no. 4, pp. 509-523, October 2024.

Early Detection Of Canine Babesia From Red Blood Cell Images Using Deep Ensemble Learning

Dilip Kumar Baruah^{ORCID}, Kuntala Boruah^{ORCID}

Department of Computer Application, Sibsagar University, Sivasagar-785665, India

*Corresponding author: kuntala17@gmail.com (Kuntala Boruah)

ABSTRACT Artificial intelligence-assisted medical diagnosis is enhancing accuracy with the contribution of several state of the art technologies such as Deep Learning (DL), Machine Learning (ML) and Image Processing (IP). From the detection of diseases to the selection of proper treatment plans, AI-powered assistance is effectively employed by healthcare professionals. Despite these advancements, the application of AI in animal healthcare is lagging behind, presenting a significant scope for AI adoption in veterinary medical diagnostics. This study addresses this gap by focusing on the automated diagnosis of canine Babesia infection, a parasitic disease that affects red blood cells (RBC). Our research contributed by developing a labeled dataset of microscopic images of red blood cells of infected and uninfected cases. During this work, four AI models are developed for automated classification: a custom Convolutional Neural Network (CNN), two pre-trained models (VGG16, DenseNet121) and a hybrid model (DenseNet121 + Support Vector Machine (SVM)). The performance of these models was 96.88%, 94%, 96.37% and 95.50% respectively. To further enhance the accuracy, a weighted average ensemble technique was employed. The ensemble model achieved an improved accuracy of 97.75%, demonstrating its potential. The enhanced performance of the ensemble model highlights the effectiveness of our method, significantly outperforming traditional methods and providing veterinarians with an efficient early diagnosis tool. This study is one of the few to address disease detection from microscopic images in animals using the potential of Artificial Intelligence.

INDEX TERMS Canine Babesiosis, CNN, Deep Learning, Machine Learning, Transfer Learning

I. INTRODUCTION

Babesia species such as Babesiacanis, Babesiagibsoni, Babesiaroi and Babesiavogeli invade mammalian red blood cells (RBCs) causing anemia and posing a serious threat to dogs across the world [1]-[4]. Babesiosis is primarily spread through the bites of infected ticks with evidence of direct animal-to-animal transmission. The traditional diagnostic process involves microscopic examination of Giemsa or Wright's stained peripheral blood smear. However, other diagnostic tests such as the Rapid Antigen Test (RAT), ELISA (enzyme-linked immunosorbent assay) and molecular assays like PCR (polymerase chain reaction) [4] are seldom used. AI is utilized in several medical problems for its ability to provide accurate, consistent and rapid diagnostic results, enhancing the efficiency and reliability of disease detection. In human disease diagnosis, AI-powered applications are already demonstrating high accuracy in various diagnostic tasks. However, the application

of AI in animal healthcare remains underexplored, presenting a significant opportunity for innovation.

One of the most successful applications of CNNs is in the field of quantitative microscopy. Convolutional Neural Network (CNN) has enabled models to learn by emulating the human visual systems' functionality and structure. In several paradigms such as intelligent medical treatments [5], speech recognition [6], face detection [7], object detection [8], self-driving or autonomous cars [9], handwritten character recognition [10], cancer detection [11], malaria parasite detection [12], [13] and in various other fields CNNs' performances are attaining quite promising results.

For this study, we collaborated with veterinary pathology laboratories to create a custom dataset. We build and retrain two (02) pre-trained CNN models, a custom CNN model and one hybrid model. Initially VGG16 [14] and DenseNet121 [15] were deployed in a non-trainable setting, with all weights imported as configured during training on the

ImageNet dataset. During the transfer learning approach, the top layers after the flattened layer were removed and replaced with customized neural networks for our desired binary classification. An ensemble approach was adopted by combining the predictions of these four models, which is less susceptible to errors or biases present in any single model. The successful development of such a system could result in a rapid, cost-effective, automated system that requires minimum human intervention and accurately assists pathologists. Through the integration of these fields, we fill a fundamental void in veterinary diagnostics by demonstrating how deep ensemble learning may greatly improve animal wellness.

The findings of this research works are expected to contribute in the following ways:

- a. As there is no dataset available in public domain for babesia infected RBC images, the dataset generated for this research will yield a new repository for upcoming researchers.
- b. The results of this study will lead to a new way for diagnosing babesia that is fast, accurate, automated and cost effective using deep learning strategies
- c. Attract several researchers to work on problems related to animal healthcare and diagnostics

The remainder of the paper is structured as follows: Section 2 briefly discusses related works published in recent years. Section 3 covers the materials and methods used in this study, highlighting the data acquisition and pre-processing steps, along with a discussion on the theoretical background of the strategies used, including an overview of CNNs and the transfer learning approach. Section 4 and 5 proposes the new ensemble learning scheme and presents the results and discussion. Finally, the paper concludes with the findings and future scope.

II. RELATED WORK AND PROBLEM STATEMENT

Deep Neural Network models are nonlinear methods that are highly flexible, able to learn complex relationships between variables and approximate any mapping function. Due to this flexibility, these models have high variance, which can be addressed by ensemble deep learning approaches. These approaches involve training multiple deep models for the problem and combining their predictions. Various researchers have utilized ensemble learning in combination with deep learning techniques. Given the potential benefits, it is crucial to explore how strategies employed in other fields could be adapted to veterinary contexts. Therefore, we have chosen to review publications that utilize deep ensemble learning strategies in diverse application areas. Following are the most significant model proposed in last few years using ensemble learning strategies (shown in TABLE 1).

III. MATERIAL AND METHODS

A. PROPOSED METHODOLOGY

The proposed approach seeks to create a model with high classification accuracy for distinguishing between infected and uninfected cells. Initially, four different models are examined separately. We take into account models from various categories: two pre-trained models are deployed, one self-created custom CNN model (trained from scratch) and one hybrid model that uses DL for feature extraction and ML for classification (DenseNet121+SVM). Each model is used for the classification on the same dataset and the accuracies are recorded. To enhance accuracy and robustness, we applied average ensemble learning by combining the predictions of all four models. Later, we implemented weighted ensemble learning. The appropriate weight for which the weighted ensemble model delivers the highest accuracy is determined by grid search. FIGURE 1 demonstrates the proposed classification technique.

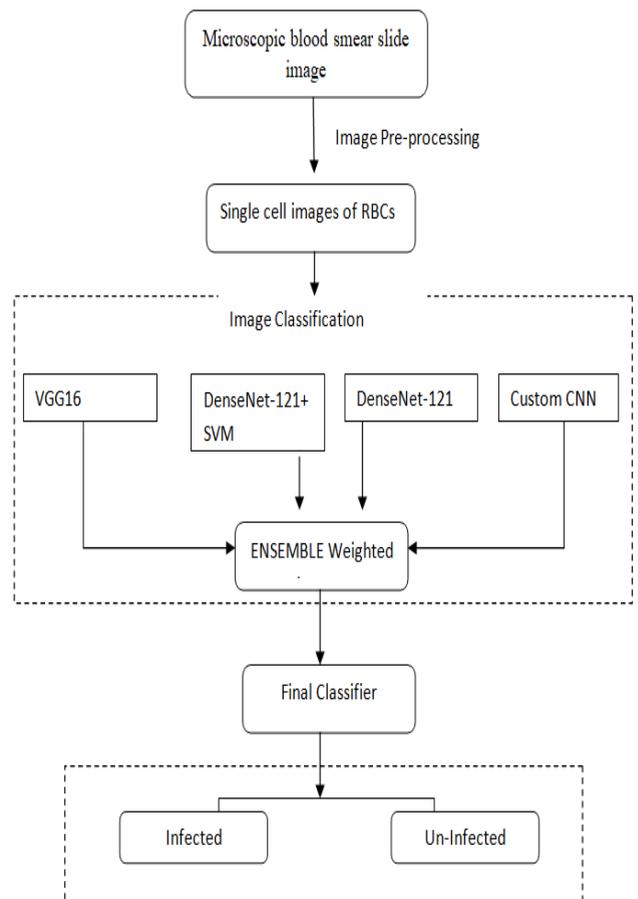


FIGURE 1. The proposed deep ensemble model for classification

A. DATA ACQUISITION

Data collection for this work is carried out with the help of laboratories that have expertise in animal pathology, such as the laboratory of the Veterinary Clinical Complex of the College of Veterinary Science, Khanapara, Assam. Suspected canine babesiosis blood samples are collected at the laboratory and blood smears are prepared on glass slides as per standard

protocol. Each microscope has a mounted camera that can be used for capturing microscopic slide images. These slides are observed under a microscope (SENSI i4000, Labovision) under the oil immersion objective (100X). In the case of babesia infection, a pear-shaped, round, oval, pleomorphic or signet ring-shaped spot[32] is visible in the red blood cell, which is used as an indicator. The appearance of the spot indicates an infected cell.

TABLE 1
Summary of recent models

Author(s)	Proposed Method	Datasets
Díaz et al. (2009)[16]	Quantification and classification of malaria infection	450 malaria images Specificity of 99.7% and sensitivity of 94%.
Anggraini et al. (2011)[17]	Novel image processing algorithm to detect malaria	Giemsa-stained peripheral blood Samples
Das et al. (2015)[18]	Image characterization and classification framework for malaria-infected stage detection.	Microscopic images of thin blood smears Specificity: 97.29%-98.64%, Sensitivity: 100%, PPV: 99.46%-99.73%, Overall accuracy: 96.73%-96.84%.
Liang et al. (2016)[19]	Robust ML CNN-based image analysis for malaria diagnosis.	27,578 single cell images accuracy of 97.37%.
Bashir et al. (2017)[20]	Feature-based malaria detection using ANN classifier.	Centre for Disease Control (CDC) website.
Rajaraman et al. (2018)[21]	Evaluation of pre-trained CNN-based DL models as feature extractors for classifying parasitized and uninfected cells.	National Library of Medicine (IRB#12972)
Kora et al.(2019)[22]	Meta-learning ensemble method fusing baseline DL models with 2 tiers of meta-classifiers.	Arabic tweets, AJGT, IMDB review, SemEval 2017 Task 4, COVID19 fake news detection, ArSarcasm
Fuhad et al., (2020)[13]	CNN-based model for malaria diagnosis from blood smear images using autoencoder.	Microscopic blood smear images Accuracy of 99.23%.
Livieris et al. (2021)[23]	Two ensemble prediction models exploiting WCNNs using Bagging and Boosting.	30 datasets from UCI Machine Learning Repository (Dua and Karra Taniskidou)
Alharbi et al., (2021)[24]	Sentiment analysis DL-based model to predict opinion polarity and sentiments.	LABR,HTL,RES, PROD,ArTwitter, ASTD Accuracy of 81.11% to 94.32%
Mohammadi et al. (2022)[25]	Four DL models (CNN, LSTM, BiLSTM, GRU) combined using stacking ensemble with logistic regression as meta-learner.	SemEval2016(SE-ABSA 2016) Increase in accuracy of aspect-based prediction by 5% to 20% compared to basic DL methods.
Bhuiyan et al (2023)[26]	Ensemble learning-based DL model using VGG16, VGG19, and DenseNet201 for malaria parasite identification.	Red blood cell images accuracy of 97.92%
Jameela et al. (2022)[27]	Automated diagnosis to evaluate parasitemia using CNN models (ResNet50, ResNet34, VGG-16, VGG-19).	N/A
Alnussairi & Ibrahim (2022)[28]	Deep CNN to improve malaria diagnosis accuracy using VGG19, ResNet50, and MobileNetV2.	NIH Malaria Dataset
Madhu et al. (2023)[29]	Deep-learning approach to distinguish between malaria-parasitized and uninfected cells.	N/A
Wang et al. (2023)[30]	Deep learning algorithm based on YOLOv7 for identifying and classifying Plasmodium species in thin blood smears.	Thin blood smears
Jorwal et al. (2024)[31]	DL model for malaria detection using blood smear images.	Blood smear images

B. IMAGE PRE-PROCESSING

For our experiment, instead of processing the whole slide images, we decided to crop single red blood cell images from the slides. This approach can highly reduce processing costs and optimize resource utilization. To crop the blood cells from the microscopic slide images, various image pre-processing techniques are applied, such as resizing the image to a standard size of 512 x 512 and converting the image to grayscale. The noise of the image is removed using Gaussian blur, adaptive thresholding to convert the grayscale image to a binary image. Otsu's method automatically determines the optimal threshold value. Finally, morphological operations are performed to improve the quality of the binary image, particularly in separating connected components and removing noise. A morphological opening operation is performed, which is a combination of two morphological operations called erosion

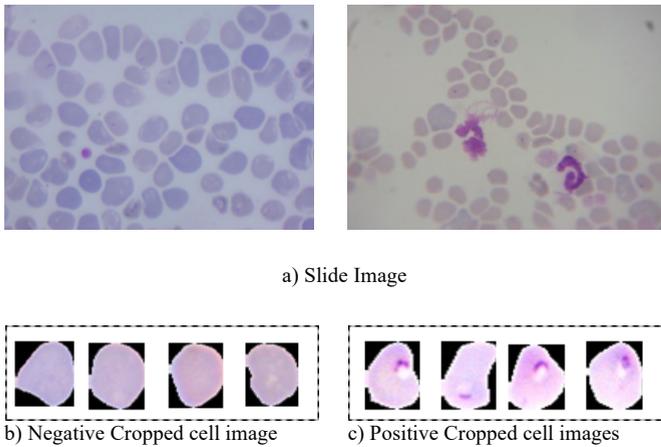


FIGURE 2. Blood Cells Images(a) Slide Image; (b) Negative Cropped cell image; (c) Positive Cropped cell images

followed by dilation. Next, we find the contours that detect the boundaries of the cells in the binary image. The contours create individual masks, and these masks are then applied to the original image, allowing the extraction of distinct cells. FIGURE 2 (a),(b),(c) show the samples of a slides, and Negative and Positive cropped cells respectively.

The process saves each cropped cell as a separate image in an output folder.

C. DATA AUGMENTATION

Data augmentation is used to increase the number of images in the dataset and to provide generalization in the training process by reducing the problem of overfitting. It involves applying various transformations to the existing data to create new, slightly altered versions of the original data. For this work, we have included rotation (randomly between -10 and 10 degrees), zooming images with a probability of 0.3, flipping (left to right, top to bottom) with a probability of 0.5, shearing (randomly between -10 and 10 degrees), adjusting brightness and contrast (both are randomly chosen between 0.5 and 1.5 with a probability of 0.5). Also, images are flipped left-right and top-bottom, along with a shear and zoom

probability of 0.2. These will help to increase the diversity of the training dataset to improve robustness and model generalization. For this experiment, we decided to set the augmented dataset to 4000 images saved in separate folder. The dimensions of the augmented images are 128x128 pixels each.

D. CONVOLUTIONAL NEURAL NETWORK(CNN) AND TRANSFER LEARNING APPROACH

Convolutional Neural Networks (CNNs) have gained tremendous attention for their ability to automatically learn hierarchical features from images, which can solve various computer vision-based problems such as image classification, segmentation, and object detection. The CNN architecture is designed to mimic the functionality of the human visual system. There are several convolutional blocks[33] in a CNN architecture, and each block contains convolutional filters that extract features from the input images[34]. These filters are followed by pooling and Normalization layers to reduce the spatial dimensions and normalize the activations[35],[36]. FIGURE 3 presents a basic CNN architecture and Eq. (1), (2), (5)[37], Eq. (3)[38], Eq. (4), Eq. (6)[39] represents the basic mathematical model for CNN architecture. The convolution operation between an input image and filter is given by Eq. (1).

$$O(x, y) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} I(x+i, y+j) \cdot F(i, j) \quad (1)$$

where $O(x, y)$ is the output feature map, $I(x+i, y+j)$ is the input image, $F(i, j)$ is the filter, m, n are the dimensions of the filter.

The **Rectified Linear Unit (ReLU)** activation function is defined as Eq. (2)

$$ReLU(x) = \max(0, x) \quad (2)$$

where x is the input to the activation function, $\max(0, x)$ is a mathematical operation that returns the larger of the two values if x is less than or equal to zero, $\max(0, x)$ returns 0 if x is greater than zero, $\max(0, x)$ returns $0x$

1. POOLING OPERATION (MAX POOLING)

Pooling helps in managing spatial dimensions, enhance computational efficiency and improve generalization abilities in Convolutional Neural Networks (CNNs). The pooling layer summarizes the features present in a region of the feature map generated by a convolution layer. It reduces the spatial dimensions (height and width) of feature maps, which decreases the number of parameters, lessens the computational load and reduces the risk of overfitting. Max Pooling selects the maximum value from a defined region of the feature map. It produces a down sampled feature map, retaining the most prominent features while discarding less important information, effectively downsampling the input (Eq. (3)).

$$p_{j,m} = \max(h_{j,(m-1)N+r}) \quad (3)$$

where m -th max-pooled band is composed of j related filters $p_m, N \in \{1, \dots, R\}$ is a pooling shift.

2. FLATTENING

Flattening is the process of converting the multi-dimensional output of the convolutional and pooling layers into a single one-dimensional vector. The feature maps from the pooling and convolutional and are flattened into a one-dimensional vector that are fed to the next layer i.e. Fully Connected Layer as input which performs high-level reasoning based on the extracted features. To make a final decision regarding the input data, it enables the network to integrate and make use of all the spatial features that were learned throughout the convolutional layers. The fully connected layers take the flattened vector and process it to predict the final output as shown in Eq. (4).

$$FlattenedSize = O_{width} \times O_{height} \times O_{channels} \quad (4)$$

where O_{width} is the width of the feature map, O_{height} is the height of the feature map and $O_{channels}$ is the number of channels.

3. FULLY CONNECTED LAYER

Dense layers or fully linked layers are an essential part of convolutional neural networks (CNNs) and other neural network architectures. Usually located at the end of the network design these layers are in charge of merging the features that were retrieved previously in order to produce final predictions, such as classifying an image or producing a regression output.

Fully connected layers are vital, particularly in classification tasks, where the objective is to categorize the input image into different classes according to the features learned. In a fully connected layer, each neuron receives input from every neuron in the preceding layer, combining these inputs in various ways to make predictions (Eq. (5)).

A set of logits is produced by the last fully connected layer, also referred to as the output layer, following the convolutional, pooling, and fully connected layers. Each of these real-valued logits represents the network's score for a particular class. These logits are converted into probabilities using the Softmax method. The function successfully highlights the class with the highest projected score by emphasizing the greatest logits and suppressing smaller ones. A probability distribution is the Softmax function's output. The probability that an input image belongs to a specific class is represented by each value. It's an essential part of the final layer in many classification-based CNN architectures (Eq. (6)).

$$Softmax(z) = \frac{e^{z_i}}{\sum_{j=1}^n e^{z_j}} \quad (6)$$

where Softmax function $z = [z_1, z_2, \dots, z_n]$, z_i is the score of class i , e is the Euler's number.

After the last convolutional block, the complex feature maps are flattened into a one-dimensional array and fed into fully connected (dense) layers. These dense layers may consist of varying numbers of layers and neurons, depending on the complexity of the task. The final output layer has a number of neurons equal to the number of classes in the classification problem. The output of the model is given in terms of the probability of each class, with the highest probability chosen as the prediction result. The advantage of using CNN over Neural Network is in its reduced processing overhead. As all deep learning (DL) models are data-hungry, convolutional neural networks (CNNs) are no exception and require a massive

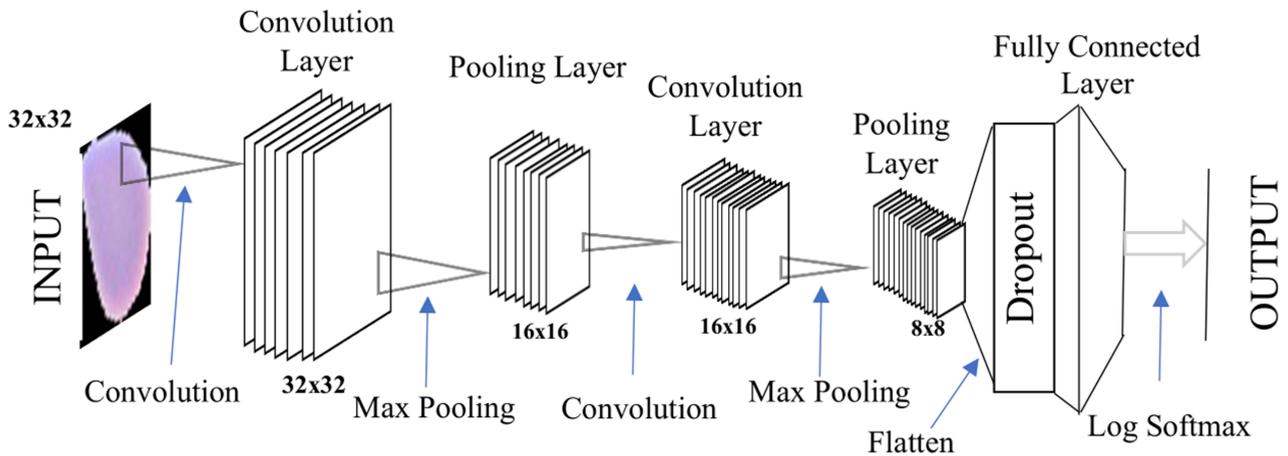


FIGURE 3. Basic CNN-architecture

$$y = f(x.w) \quad (5)$$

where w is the weight matrix, x is the input vector, f is the activation function, y is the output vector

4. SOFTMAX ACTIVATION (FOR CLASSIFICATION)

amount of meticulously labeled data for training. In most scenarios, it is difficult to obtain such annotated data. Even if such a huge amount of data is available, training a CNN from scratch often requires massive processing power and time, which is challenging in resource-constrained environments. This problem can be addressed by adopting a transfer learning approach. Instead of training a CNN from scratch, pre-trained

CNN models can be used to fine-tune weights. In this strategy, the already trained weights on large datasets can be directly used for our models. Several renowned pre-trained models, such as VGG16, VGG19, AlexNet, DenseNet, ResNet, Inception etc., are trained on extensive datasets like ImageNet. Depending on the requirement, one can freeze and unfreeze any number of convolutional blocks. These pre-trained models capture low-level features like edges and textures, which are generally applicable across different visual recognition tasks. The fully connected layers are replaced with custom neural network layers with any number of layers and neurons. The output layer should have neurons equal to the number of classes in the classification problem. By using a transfer learning approach, a model can deliver high accuracy with a limited dataset size and converge faster, resulting in faster training.

E. ENSEMBLE LEARNING

In ensemble approaches the stability and prediction capacity of the final model are increased by combining individual models using bagging[40], boosting[41] or stacking[42] strategies. To arrive at the final choice, the predictions of multiple simple models are merged, rather than learning a single sophisticated model[43],[44]. When compared to the accuracy of the individual models, it yields a composite prediction whose final accuracy is higher. These methods can be divided into two groups: sequential ensemble methods and parallel ensemble methods. In sequential ensemble methods, base learners are generated consecutively. The basic motivation of sequential methods is to use the dependence between the base learners; by giving higher weight to previously mislabeled examples, the overall performance of the model can be boosted. Parallel ensemble methods are applied when the base learners are generated in parallel, such as in random forests.

Deep ensemble techniques involve creating a collection of multiple deep learning models, each trained independently on the same dataset (or slightly modified versions of the dataset). These individual models are then combined to make a final prediction. A weighted average ensemble leverages predictions from multiple models, but instead of simply voting or averaging the predictions, each model's contribution to the final prediction is weighted according to its perceived accuracy. FIGURE 4 demonstrates the working of Ensemble learning. The performances of the models are quantified with the help of metrics called accuracy, recall, precision, specificity and F1-Score. Each of which are depicted in Eq.(7), (8), (9), (10), (11)[45], [46] respectively.

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN} \quad (7)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (8)$$

$$\text{Sensitivity (Recall)} = \frac{TP}{TP+FN} \quad (9)$$

$$\text{Specificity} = \frac{TN}{TN+FP} \quad (10)$$

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (11)$$

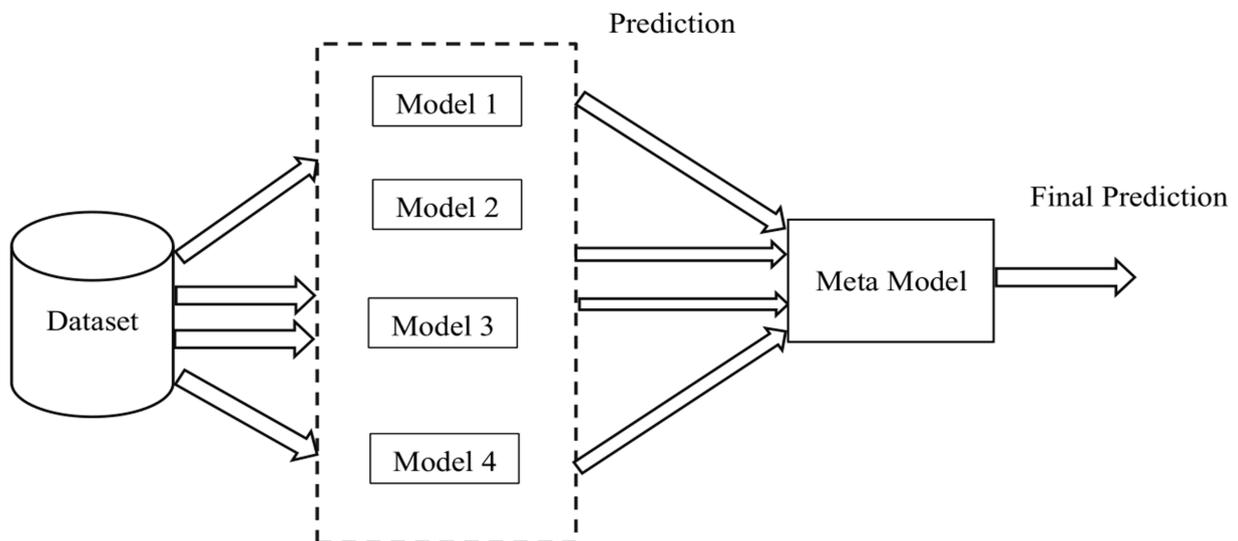
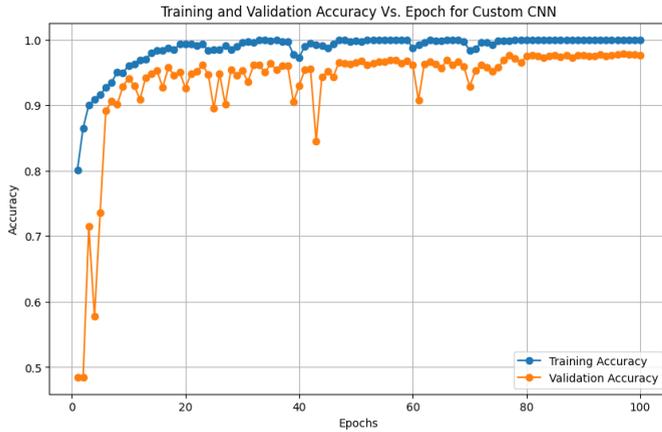


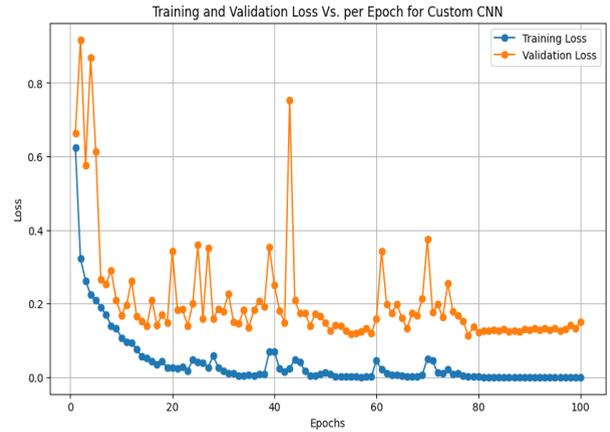
FIGURE 4. Working of Ensemble Learning

TP (True Positive) is the number of instances that are positive and correctly predicted as positive. TN (True Negative) is the number of instances that are negative and correctly predicted as negative. FP (False Positive) is the number of instances that are actually negative but incorrectly predicted as positive. And FN (False Negative) is the number of instances that are actually positive but incorrectly predicted as negative.

activities. The testing and training are performed using T4 Tesla GPU and an Intel Xeon 2.20 GHz CPU with 16 GB RAM. The platform provided by TensorFlow with Keras 2.3.1 API is used for implementing neural networks. Python 3 is used for all programming in this project. For the training 80% of total images are used whereas for testing 20% is reserved. After employing augmentation 4000 images are generated out of which 3200 images i.e. 80% of total images are used for training whereas 800 images (20%) are reserved for testing or validation.

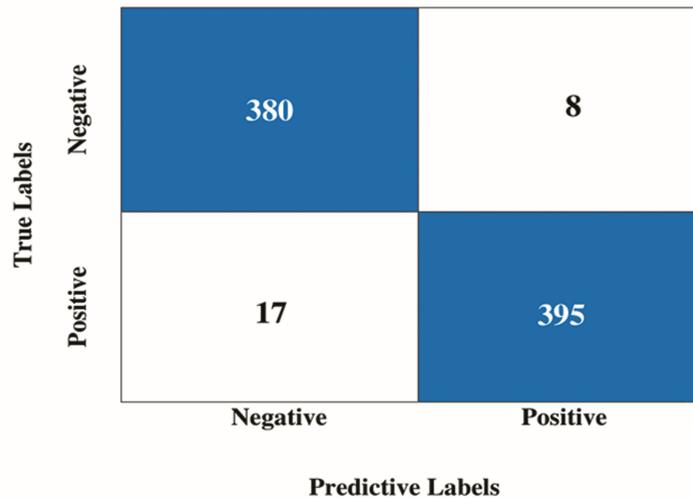


(a)



(b)

Confusion Matrix



(c)

FIGURE 5. Graph of Custom CNN: (a) Loss graph of Custom CNN, (b) Accuracy graph of Custom CNN, (c) Confusion Matrix of Custom CNN

IV. RESULTS

A. EXPERIMENTAL SETUP

As the experiments were conducted in a resource-constrained environment, the personal computer resources were only used for support purposes. We completely relied on the Google cloud environment called Google Colab Pro for all training and testing

B. EXPERIMENT 1: CUSTOM CNN MODEL FOR CLASSIFICATION

A custom CNN model is designed with a total of three convolutional blocks. The input shape of the model is set to (128,128,3). The first convolutional layer has 32 filters, followed by MaxPooling and BatchNormalization. The second convolutional layer has 64 filters, and the third convolutional

layer consists of 128 filters. The fully connected layer is fed the flattened output of the third convolutional block after a dropout of 50%. In the proposed architecture, one dense layer is added at the top. Dense layer 1 has 256 nodes and the output layer has 2 nodes (indicating a binary classification problem). The model has 7,633,090 (29.1 MB) parameters, of which 7,632,642 are trainable and 448 are non-trainable. The model achieves a high accuracy of 96.88% in 100 epochs. The precision, recall, and F1-score of the model are 96%, 98% and 97% respectively. The 'Adam' optimizer is used with a learning rate of 0.0001, and the loss function is `sparse_categorical_crossentropy`. The model summary is shown in TABLE 2. The loss and accuracy graphs for training and testing are demonstrated in FIGURE 5(a) and FIGURES 5(b) respectively. Performance of the model is recorder in TABLE 4. From the confusion matrix shown in FIGURE 5(c), it is prominent that the model is able to correctly classify 775 images out of 800 validation images. 8 numbers of negative images are misclassified as positive image whereas 17 positive images are wrongly predicted as negative.

TABLE 2
Model Summary for the proposed Custom CNN model

Layer	Size	Kernel Size	Stride	Act
Convolutional 1	128x128x32	5x5	1	ReLU
Pooling 1	63x63x32	3x3	2	-
Normalization	63x63x32	-	-	-
Convolutional 2	63x63x64	5x5	1	ReLU
Pooling 2	31x31x64	3x3	2	-
Normalization	31x31x64	-	-	-
Convolutional 3	31x31x128	5x5	1	ReLU
Pooling 3	15x15x128	3x3	2	-
Normalization	15x15x128	-	-	-
Fully Connected	256	-	-	ReLU
Output	2	-	-	Softmax

C. EXPERIMENT 2: CLASSIFICATION USING TRANSFER LEARNING

In the pre-trained models, VGG16 and DenseNet121, the top layers are removed and replaced with a fully connected layer. The input size is set to (128,128,3), which is the same as the input image dimensions. The dense network consists of two layers. The first layer has 256 nodes, and the activation function used is ReLU. The output layer has 2 nodes, and the activation function used is SoftMax. The models are compiled with a learning rate of 0.0001 using the Adam optimizer, and `sparse_categorical_crossentropy` is used as the loss function. The accuracy of the VGG16 model is 94% for 100 epochs, and similarly, DenseNet121 achieves 96.37% accuracy for 100 epochs. The loss and accuracy graphs for training and testing of VGG16 and DenseNet121 are demonstrated in FIGURE 6 (a), FIGURE 6(b), FIGURE6 (c) and FIGURE6 (d) respectively. FIGURE 7(a) and FIGURE 7(b) shows the confusion matrix of VGG16 and DenseNet121. Table 4 depicts the performance of the two models.

D. EXPERIMENT 3: HYBRID MODEL (DENSENET121 AND SVM)

Using a hybrid model that combines deep learning (DL) and machine learning (ML) techniques offers several advantages. One significant benefit is the ability to leverage the strengths of both approaches: DL's capacity to automatically extract intricate and high-dimensional features from raw data, and ML's efficiency in making predictions with lower computational overhead. This approach helps environment with limited computational resources. For instance, while Convolutional Neural Networks (CNNs) can handle the complex task of feature extraction from large datasets, traditional ML classifiers can then be used to process these extracted features, providing robust predictions with reduced training time. Additionally, this hybrid approach can mitigate the need for extensive labeled datasets. Overall, hybrid models offer a balanced solution, optimizing both feature extraction and predictive accuracy.

In our previous work, we already examined several CNN+ML combinations, and the performances were recorded. We used four pre-trained models: VGG16, ResNet50, DenseNet121 and Inception-V3 for feature extraction purposes, and we used several ML models such as Support Vector Machine (SVM), Random Forest (RF), Extreme Gradient Boost Classifier (XGBoost) and Adaptive Boosting (AdaBoost) for classification. In total, 16 combinations of models are evaluated on the same dataset. The hybrid models along with the accuracy are shown in TABLE 3.

TABLE 3
Performance of different hybrid models

SL.NO	Hybrid Model	Accuracy (%)
1	VGG16 + RF	83.82
2	VGG16 + SVM	86.46
3	VGG16 + AdaBoost	82.30
4	VGG16 + XGBoost	81.95
5	ResNet50 + RF	82.70
6	ResNet50 + SVM	81.57
7	ResNet50 + AdaBoost	81.95
8	ResNet50 + XGBoost	83.83
9	DenseNet121 + RF	91.35
10	DenseNet121 + SVM	95.50
11	DenseNet121 + AdaBoost	92.48
12	DenseNet121 + XGBoost	90.92
13	Inception-V3+ RF	92.10
14	Inception-V3+ SVM	92.20
15	Inception-V3+ AdaBoost	91.35
16	Inception-V3+ XGBoost	90.97

It is clear that DenseNet121 with SVM is able to deliver highest accuracy of 95.50%, therefore in our work we decided to consider this model for ensemble.

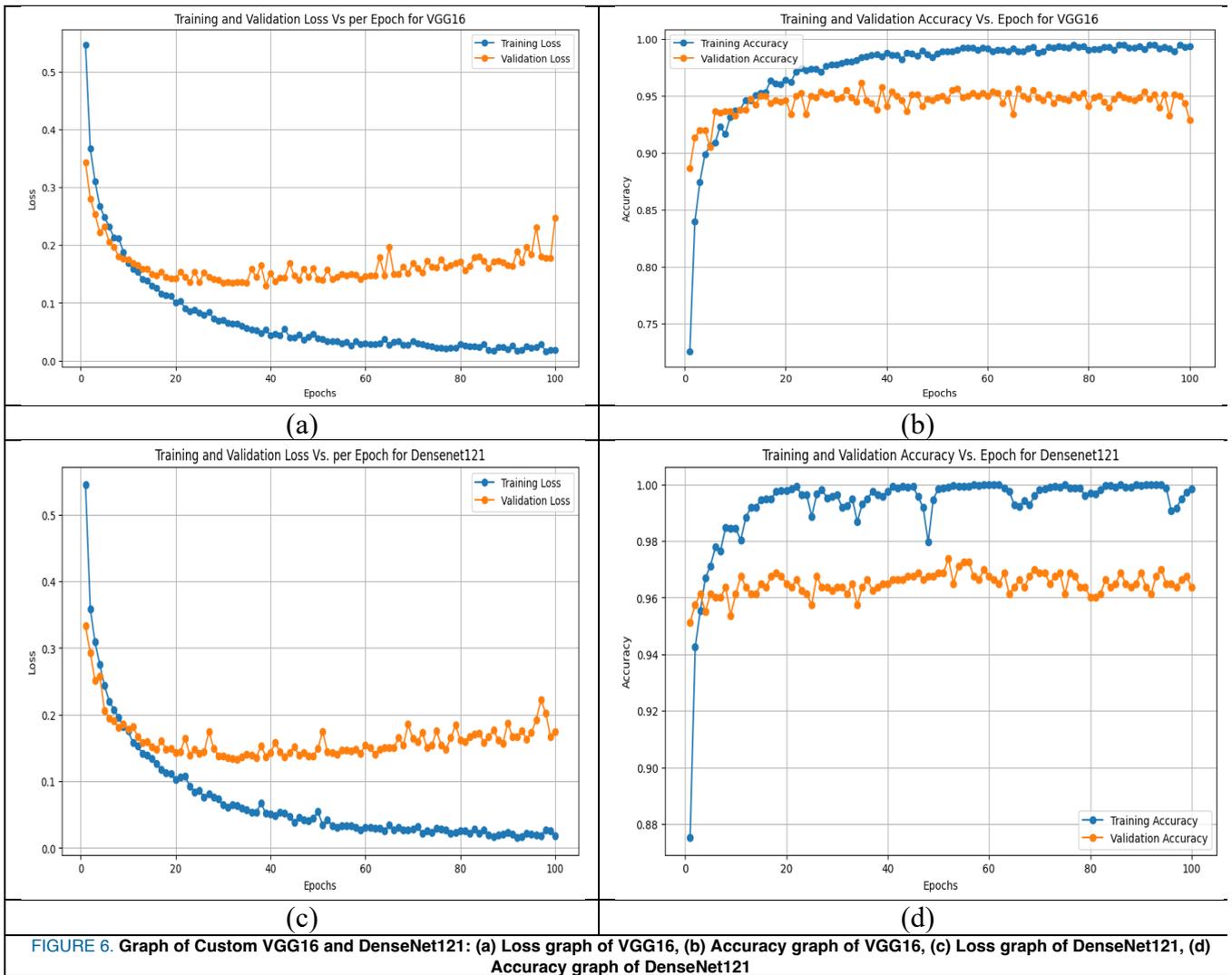


FIGURE 6. Graph of Custom VGG16 and DenseNet121: (a) Loss graph of VGG16, (b) Accuracy graph of VGG16, (c) Loss graph of DenseNet121, (d) Accuracy graph of DenseNet121

Confusion Matrix of DenseNet121 with SVM Hybrid Model is shown in FIGURE 8. To compare the performances of all the CNN based model the graphs of training loss, training accuracy, validation loss, validation accuracy are plotted against epochs is illustrated in FIGURE 9(a) – FIGURE 9(d).

A. EXPERIMENT 3: ENSEMBLE OF MODELS FOR CLASSIFICATION

Individually, four models were investigated to detect Babesia parasites from microscopic images. All of the models showed decent performance, each with certain advantages over the others. An ensemble learning strategy was employed to combine the advantages of all the models. Initially, an average ensemble learning method was implemented, achieving 96.97% accuracy. To further increase accuracy, a weighted average ensemble was employed, achieving 97.75% accuracy with weights of 0.2, 0.1, 0.2, and 0.1. These weights were applied to the predictions of the respective models: 0.2 to the custom CNN model, 0.1 to the VGG16 model, 0.2 to the DenseNet121 model, and 0.1 to the hybrid model (DenseNet121+SVM). TABLE 4 shows the

performance achieved by the different models. The weight of each model was calculated using Eq. (12)[47].

$$W_i = \frac{C_i}{\sum_{i=1}^n C_i} \quad (12)$$

W_i : The weight of the i^{th} classifier in the ensemble.

C_i : The confidence or performance score of the i^{th} classifier.

n : The total number of classifiers in the ensemble.

After that, probability of parasitized class is determined using the formula given in the Eq. (13)[47].

$$\text{Prediction Probability} = \sum_{i=1}^n P_i * W_i \quad (13)$$

where P_i =Probability assigned by the i^{th} classifier
 W_i = i^{th} classifier's weight

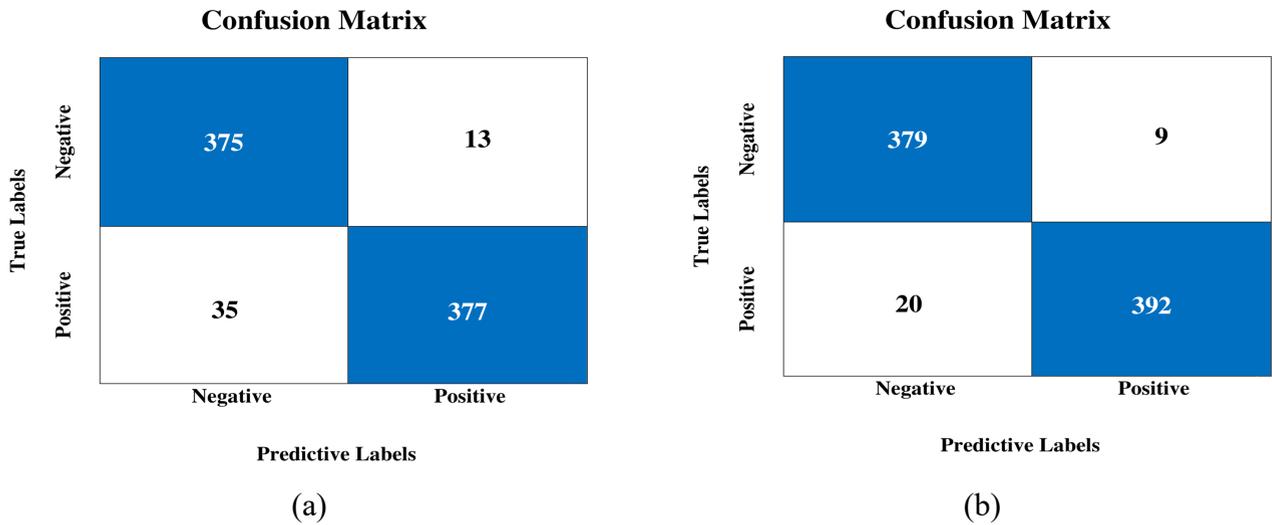


FIGURE 7. Confusion Matrix of Custom VGG16 and DenseNet121: (a) Confusion Matrix of VGG16, (b) Confusion Matrix of DenseNet121.

Where W_i represents the weight of i th model. 'i' is ranging from 0 to 'n', 'n' being the total numbers of models participating. P_i is the probability assigned by i th model. The confusion matrix of Average Ensemble and Weighted Ensemble is shown in FIGURE 10(a) and FIGURE 10(b). From the confusion Matrix it is clear that the performance of weighted ensemble for $w_1=0.2$, $w_2=0.1$, $w_3=0.2$, $w_4=0.1$ demonstrated better performance over the Average Ensemble. In average ensemble model the misclassification is of 25 images out of total 800 images. Misclassification of positive cell image is higher than negative

TABLE 4

Result performance of different models

Model	Precision		Recall		F1 Score		ACCURACY (%)
	Infected		Infected		Infected		
	YES	NO	YES	NO	YES	NO	
VGG16	0.95	0.96	0.96	0.95	0.95	0.96	94.00
Custom CNN	0.96	0.98	0.98	0.96	0.97	0.97	96.88
DenseNet1 21+ SVM	0.95	0.96	0.96	0.95	0.95	0.96	95.50
DenseNet1 21	0.95	0.98	0.98	0.95	0.96	0.96	96.37
Average Ensemble	0.96	0.98	0.98	0.96	0.97	0.97	96.97
Weighted Average Ensemble	0.97	0.99	0.98	0.97	0.98	0.98	97.75

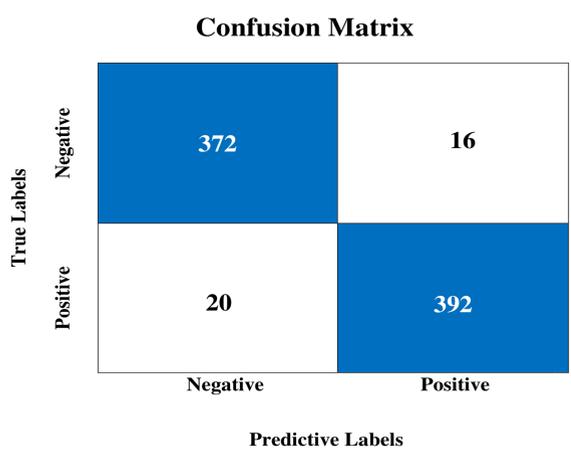


FIGURE 8: Confusion Matrix of DenseNet121 with SVM Hybrid Model

images. The weighted ensemble model showed improvement by reducing the miss classification rate. 12 positive cells images are misclassified as negative whereas 6 negative cells are misclassified as positive. The performance of the ensemble models is recorded in TABLE 4.

V. DISCUSSION

There are six (06) experiments described in the prior sections for classification of RBC images using VGG16, DenseNet121, Custom CNN, hybrid model, Average Ensemble and Weighted Average Ensemble along with their findings depicted in TABLE 4. The visual comparison of precision, recall and f1-score of evaluation results of all six experiments are shown in FIGURE11-FIGURE 13.

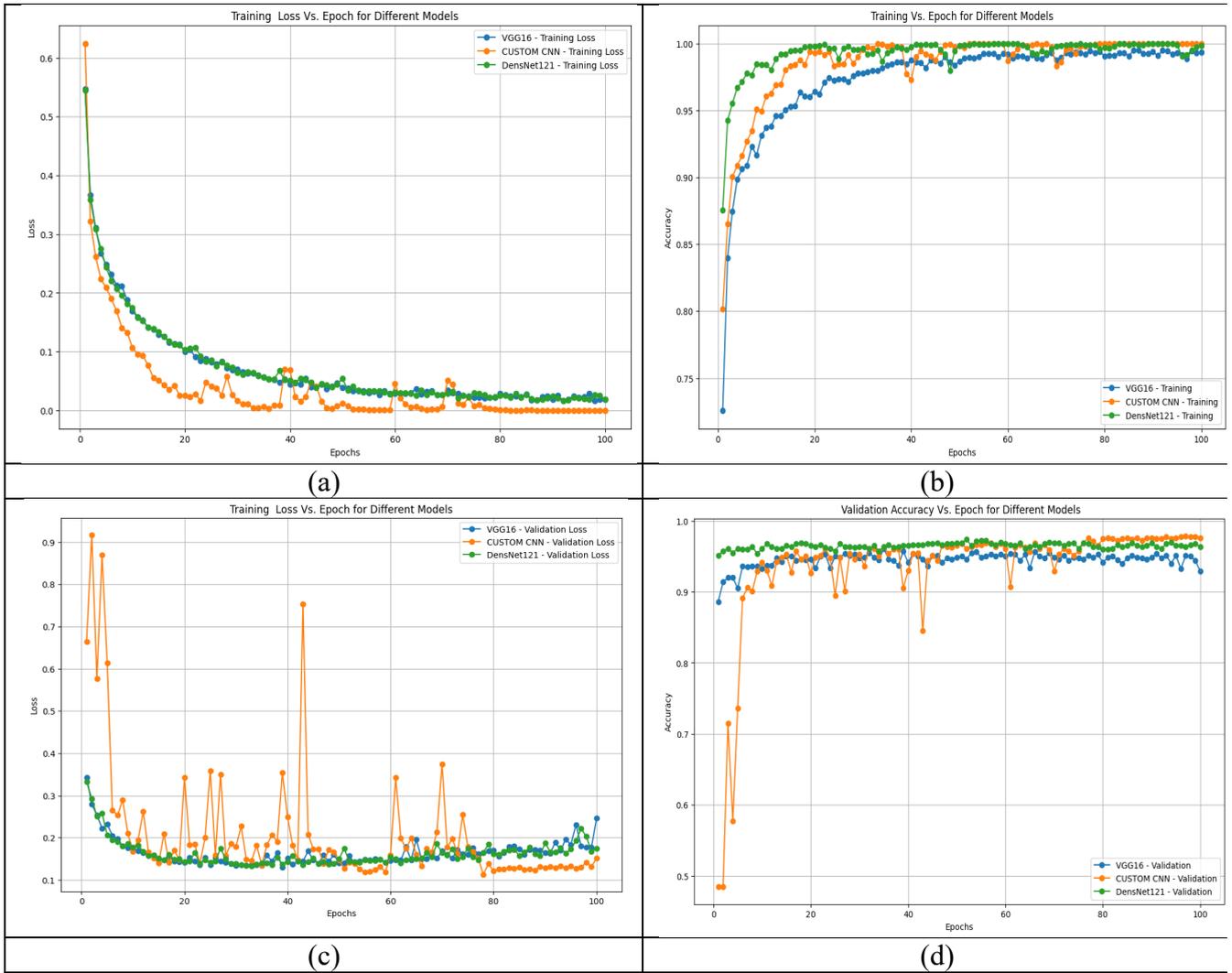


FIGURE 9. Training loss and Accuracy graph of three CNN based models: (a) Train loss of Custom CNN, VGG16 and DenseNet121, (b) Train accuracy of Custom CNN, VGG16 and DenseNet121, (c) Validation loss of Custom CNN, VGG16 and DenseNet121, (d) Validation accuracy of Custom CNN, VGG16 and DenseNet121

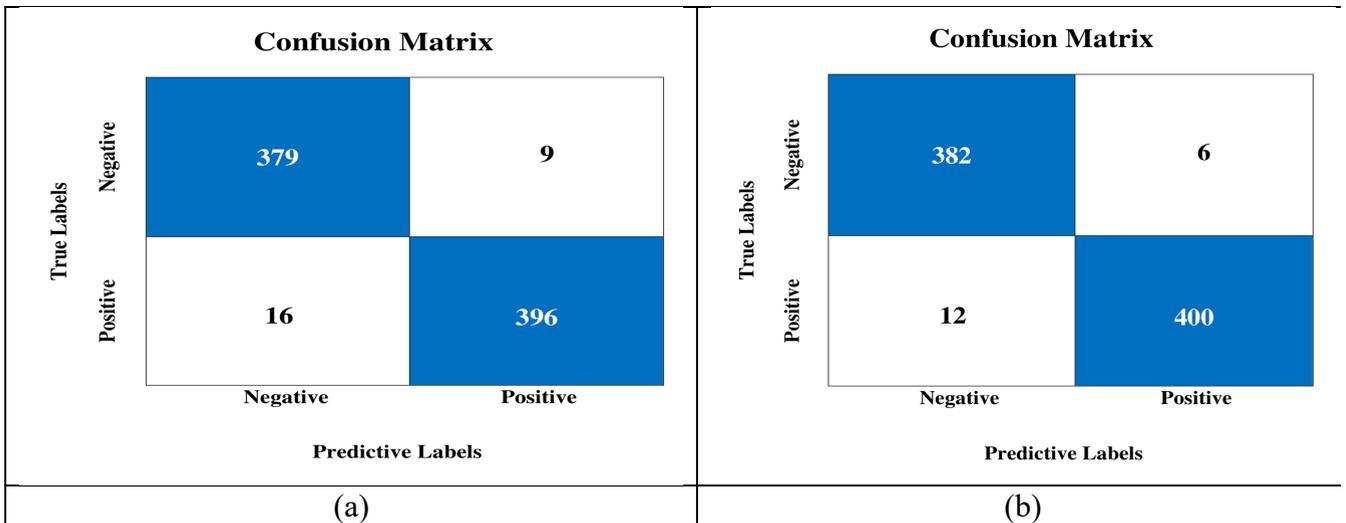


FIGURE 10. Confusion Matrix of Ensemble: (a) Confusion Matrix of Avg Ensemble, (b) Confusion Matrix of Weighted Ensemble

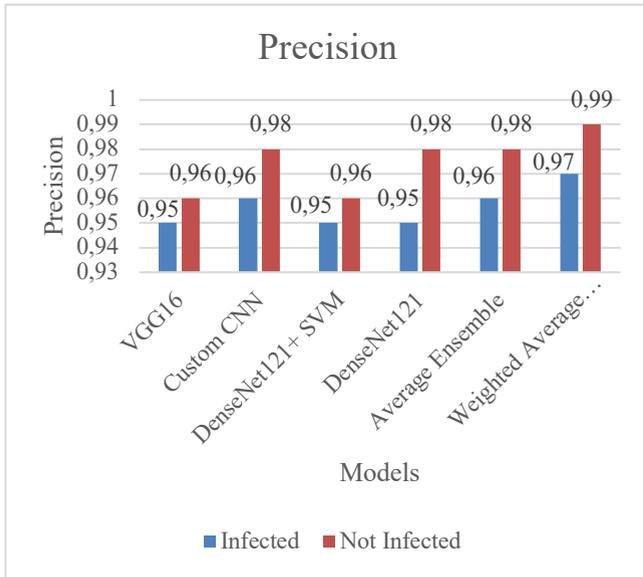


FIGURE 11. Comparison of Precision of all six experiments

The Weighted Average Ensemble model demonstrates the highest precision for both “Infected” and “Not Infected” classifications. All the models show high precision scores for “Not Infected” cases with the Weighted Average Ensemble achieving the highest scores whereas VGG16 model demonstrates lower precision (shown in FIGURE 11).

FIGURE 12 shows the comparison of Recall of all six experiments. Most models demonstrate high recall scores for “Not Infected” classifications, indicating their ability to identify most of the true positive cases.

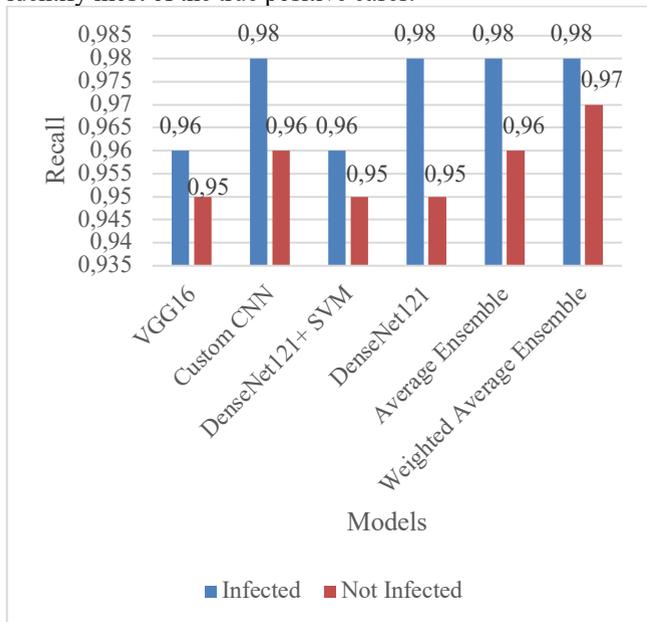


FIGURE 12. Comparison of Recall of all six experiments

FIGURE 13 visually represents the F1 Score of all the experiments. Weighted Average Ensemble emerges as the most robust model with the highest F1 scores, indicating it effectively balances precision and recall for both infected

and non-infected cells. Custom CNN and Average Ensemble are also top performers, offering a

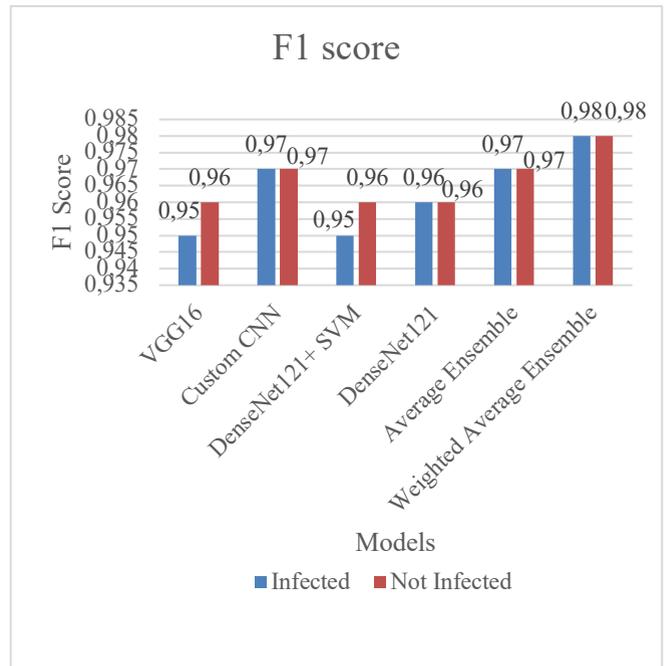


FIGURE 13. Comparison of F1-Score of all six experiments

reliable balance, particularly in detecting infected cells, which is crucial in medical diagnostics. VGG16’s lower F1 scores suggest it may not be the best choice for this task, as it struggles to balance precision and recall effectively.

The Weighted Average Ensemble model outperforms all other models, achieving the highest accuracy of 97.75%. The Average Ensemble model follows closely with an accuracy of 96.97%. Among individual models, the Custom CNN and DenseNet121 demonstrate strong performance with accuracies of 96.88% and 96.37%, respectively. DenseNet121 combined with SVM shows a slightly lower accuracy of 95.50%, while VGG16 has the lowest accuracy at 94.00%. The Weighted Average Ensemble model is the most recommended for the task of detecting and classifying infected and non-infected blood cells due to its superior balance of precision, recall, and F1 score. Custom CNN also suffers from underfitting issue. This comparison aims to assess the performance achieved in this study with previous studies. As of now we are unable to find much work in the field of microscopic level disease detection using AI in animal parasitic disease detection, we are considering malaria parasite detection in human medicinal research because of its close resemblance. The results of the comparison are presented in TABLE 5. Data is a key component of any AI-based model, and the availability of ready-to-use data in the public domain is crucial for attracting upcoming researchers. Currently, there is no publicly available image dataset for Babesia, so this work makes an effort to generate an authentic labeled dataset by collecting digital slide samples from the Veterinary Clinical Complex, College of Veterinary Science, Khanapara, Assam. To maintain diversity in the dataset, we collected samples

from different breeds and age groups of canines. However, it might still lack diversity since all samples were collected from the same location. Therefore, future research should focus on

TABLE 5
Comparison of accuracy result with previous research

Author	Technique	Accuracy
Nurcahyati et. al. [48] Year:2024	Gray Level Co-Occurrence Matrix (GLCM) and CNN	87%
	CNN with AlexNet	92%
Bhuiyan et al. (2023)[26]	Ensemble learning-based DL model using VGG16, VGG19, and DenseNet201	97.92%
Sohaib Asif et. al.[49]; 2024	MozzieNet	96.73%
Proposed Model	Average Ensemble	96.97
Proposed Model	Weighted Average Ensemble	97.75

developing a larger and more diverse dataset. Additionally, instead of classifying a single parasitic infection, future model should be upgraded to identify multiple parasitic infections (multiclass classification). Despite these limitations, the proposed Weighted Average Ensemble model has achieved quite good results compared to other models.

To enhance the quality of life for animals and their owners, early intervention is critical when any disease symptoms arise. Rather than relying solely on a veterinarian's diagnosis, our model offers an advanced disease detection capability. Early diagnosis can significantly reduce the mortality risk and assist veterinarians in managing a higher volume of cases more efficiently, thereby reducing both costs and time. Accurate diagnosis enables targeted treatments, reducing the need for broad-spectrum medications that could contribute to drug resistance. Timely detection is particularly vital for zoonotic diseases, which pose a threat to human health as well. This research advocates for the integration of technology into animal healthcare.

VI. CONCLUSION

This paper aims to bridge the gap between AI-assisted diagnostics in veterinary medicine and the human healthcare system. We developed an image dataset of Babesia-infected blood smear slides in collaboration with several nearby laboratories. We deployed multiple CNN approaches, including a custom CNN model, pre-trained models, and a hybrid model (using CNN for feature extraction and machine learning for classification). Each classifier was evaluated and compared, showing good performance but also demonstrating some biases.

To enhance classifier performance, we proposed an ensemble learning-based deep neural network. We used both average ensemble and weighted average ensemble approaches, deriving

appropriate weights through a grid search strategy. The proposed average ensemble learning model achieved an accuracy of 96.97%, while the weighted average ensemble learning model delivered a slightly higher accuracy of 97.75%.

We recognize the importance of diverse and extensive datasets for training deep learning models. To expand our image dataset, we are collaborating with several veterinary clinics from various regions. Additionally, we plan to explore other machine learning techniques, such as semi-supervised learning and unsupervised learning, to enhance detection accuracy and minimize false positives and negatives. Unsupervised learning models trained on unlabeled data will allow us to utilize un-annotated datasets effectively. A significant focus will also be on integrating the AI system with existing clinical tools, offering a seamless and user-friendly interface for veterinarians to assist in diagnostics and treatment planning.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

RESEARCH FUNDING

This research received no external funding.

REFERENCES

- [1] P. Kelly, L. Koster, and R. Lobetti, "Canine babesiosis: a perspective on clinical complications, biomarkers, and treatment," *VMRR*, p. 119, Apr. 2015, doi: 10.2147/VMRR.S60431.
- [2] K. Bhattacharjee and P. C. Sarmah, "Prevalence of haemoparasites in pet, working and stray dogs of Assam and North-East India: A hospital based study," *Vet World*, vol. 6, no. 11, pp. 874–878, Nov. 2013, doi: 10.14202/vetworld.2013.874-878.
- [3] R. Laha, M. Das, and A. Sen, "Morphology, epidemiology, and phylogeny of Babesia: An overview," *Trop Parasitol*, vol. 5, no. 2, p. 94, 2015, doi: 10.4103/2229-5070.162490.
- [4] P. Vishwakarma and M. K. Nandini, "Overview of Canine Babesiosis," in *Veterinary Medicine and Pharmaceuticals*, S. Oppong Bekoe, M. Saravanan, R. Kwame Adosraku, and P. K. Ramkumar, Eds., IntechOpen, 2020. doi: 10.5772/intechopen.82243.
- [5] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 33, no. 12, pp. 6999–7019, Dec. 2022, doi: 10.1109/TNNLS.2021.3084827.
- [6] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, no. 3, pp. 328–339, Mar. 1989, doi: 10.1109/29.21701.
- [7] H. Jiang and E. Learned-Miller, "Face Detection with the Faster R-CNN," in *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, Washington, DC, DC, USA: IEEE, May 2017, pp. 650–657. doi: 10.1109/FG.2017.82.
- [8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.
- [9] I. Sonata, Y. Heryadi, L. Lukas, and A. Wibowo, "Autonomous car using CNN deep learning algorithm," *J. Phys.: Conf. Ser.*, vol. 1869, no. 1, p. 012071, Apr. 2021, doi: 10.1088/1742-6596/1869/1/012071.
- [10] Y. LeCun et al., "Backpropagation Applied to Handwritten Zip Code Recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, Dec. 1989, doi: 10.1162/neco.1989.1.4.541.
- [11] Z. Wang et al., "Breast Cancer Detection Using Extreme Learning Machine Based on Feature Fusion With CNN Deep Features," *IEEE*

- Access, vol. 7, pp. 105146–105158, 2019, doi: 10.1109/ACCESS.2019.2892795.
- [12] Z. Liang *et al.*, “CNN-based image analysis for malaria diagnosis,” in *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, Shenzhen, China: IEEE, Dec. 2016, pp. 493–496. doi: 10.1109/BIBM.2016.7822567.
- [13] K. M. F. Fuhad, J. F. Tuba, Md. R. A. Sarker, S. Momen, N. Mohammed, and T. Rahman, “Deep Learning Based Automatic Malaria Parasite Detection from Blood Smear and its Smartphone Based Application,” *Diagnostics*, vol. 10, no. 5, p. 329, May 2020, doi: 10.3390/diagnostics10050329.
- [14] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” Apr. 10, 2015, *arXiv: arXiv:1409.1556*. Accessed: May 04, 2024. [Online]. Available: <http://arxiv.org/abs/1409.1556>
- [15] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” 2016, doi: 10.48550/ARXIV.1608.06993.
- [16] G. Díaz, F. A. González, and E. Romero, “A semi-automatic method for quantification and classification of erythrocytes infected with malaria parasites in microscopic images,” *Journal of Biomedical Informatics*, vol. 42, no. 2, Art. no. 2, Apr. 2009, doi: 10.1016/j.jbi.2008.11.005.
- [17] D. Anggraini, A. S. Nugroho, C. Pratama, I. E. Rozi, Aulia Arif Iskandar, and Reggio Nurtanio Hartono, “Automated status identification of microscopic images obtained from malaria thin blood smears,” in *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics*, Bandung, Indonesia: IEEE, Jul. 2011, pp. 1–6. doi: 10.1109/ICEEI.2011.6021762.
- [18] S. Das, S. Biswas, A. Paul, and A. Dey, “AI Doctor: An Intelligent Approach for Medical Diagnosis,” in *Industry Interactive Innovations in Science, Engineering and Technology*, vol. 11, S. Bhattacharyya, S. Sen, M. Dutta, P. Biswas, and H. Chattopadhyay, Eds., in *Lecture Notes in Networks and Systems*, vol. 11, Singapore: Springer Singapore, 2018, pp. 173–183. doi: 10.1007/978-981-10-3953-9_17.
- [19] Z. Liang *et al.*, “CNN-based image analysis for malaria diagnosis,” in *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, Shenzhen, China: IEEE, Dec. 2016, pp. 493–496. doi: 10.1109/BIBM.2016.7822567.
- [20] A. Bashir, Z. A. Mustafa, I. Abdelhameid, and R. Ibrahim, “Detection of malaria parasites using digital image processing,” in *2017 International Conference on Communication, Control, Computing and Electronics Engineering (ICCCCEE)*, Khartoum, Sudan: IEEE, Jan. 2017, pp. 1–5. doi: 10.1109/ICCCCEE.2017.7867644.
- [21] S. Rajaraman *et al.*, “Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images,” *PeerJ*, vol. 6, p. e4568, Apr. 2018, doi: 10.7717/peerj.4568.
- [22] R. Kora and A. Mohammed, “Corpus on Arabic Egyptian tweets.” Harvard Dataverse, 2019. doi: 10.7910/DVN/LBXV9O.
- [23] I. E. Livieris, L. Iliadis, and P. Pintelas, “On ensemble techniques of weight-constrained neural networks,” *Evolving Systems*, vol. 12, no. 1, Art. no. 1, Mar. 2021, doi: 10.1007/s12530-019-09324-2.
- [24] A. Alharbi, M. Kalkatawi, and M. Taileb, “Arabic Sentiment Analysis Using Deep Learning and Ensemble Methods,” *Arab J Sci Eng*, vol. 46, no. 9, pp. 8913–8923, Sep. 2021, doi: 10.1007/s13369-021-05475-0.
- [25] A. Mohammadi and A. Shaverizade, “Ensemble deep learning for aspect-based sentiment analysis,” *IJNAA*, vol. 12, no. Special Issue, Art. no. Special Issue, Jan. 2021, doi: 10.22075/ijnaa.2021.4769.
- [26] M. Bhuiyan and M. S. Islam, “A new ensemble learning approach to detect malaria from microscopic red blood cell images,” *Sensors International*, vol. 4, p. 100209, 2023, doi: 10.1016/j.sintl.2022.100209.
- [27] T. Jameela, K. Athota, N. Singh, V. K. Gunjan, and S. Kahali, “Deep Learning and Transfer Learning for Malaria Detection,” *Computational Intelligence and Neuroscience*, vol. 2022, pp. 1–14, Jun. 2022, doi: 10.1155/2022/2221728.
- [28] M. H. D. Alussairi and A. A. Ibrahim, “Malaria parasite detection using deep learning algorithms based on (CNNs) technique,” *Computers and Electrical Engineering*, vol. 103, p. 108316, Oct. 2022, doi: 10.1016/j.compeleceng.2022.108316.
- [29] G. Madhu, A. W. Mohamed, S. Kautish, M. A. Shah, and I. Ali, “Intelligent diagnostic model for malaria parasite detection and classification using imperative inception-based capsule neural networks,” *Sci Rep*, vol. 13, no. 1, p. 13377, Aug. 2023, doi: 10.1038/s41598-023-40317-z.
- [30] G. Wang, G. Luo, H. Lian, L. Chen, W. Wu, and H. Liu, “Application of Deep Learning in Clinical Settings for Detecting and Classifying Malaria Parasites in Thin Blood Smears,” *Open Forum Infectious Diseases*, vol. 10, no. 11, p. ofad469, Nov. 2023, doi: 10.1093/ofid/ofad469.
- [31] S. Jorwal, Ankit, A. Tibrewal, K. Saurav, and S. Agarwal, “Malaria Parasite Detection Using Deep Learning,” in *Proceedings of the International Conference on Machine Learning, Deep Learning and Computational Intelligence for Wireless Communication*, E. S. Gopi and P. Maheswaran, Eds., in *Signals and Communication Technology*, Cham: Springer Nature Switzerland, 2024, pp. 387–397. doi: 10.1007/978-3-031-47942-7_33.
- [32] D. Huber *et al.*, “Microscopic and molecular analysis of *Babesia canis* in archived and diagnostic specimens reveal the impact of anti-parasitic treatment and postmortem changes on pathogen detection,” *Parasites Vectors*, vol. 10, no. 1, p. 495, Dec. 2017, doi: 10.1186/s13071-017-2412-1.
- [33] F. Sultana, A. Sufian, and P. Dutta, “Advancements in Image Classification using Convolutional Neural Network,” in *2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN)*, Kolkata, India: IEEE, Nov. 2018, pp. 122–129. doi: 10.1109/ICRCICN.2018.8718718.
- [34] A. Saxena, “An Introduction to Convolutional Neural Networks,” *IJRASET*, vol. 10, no. 12, pp. 943–947, Dec. 2022, doi: 10.22214/ijraset.2022.47789.
- [35] M. Ahmadi, S. Vakili, J. M. P. Langlois, and W. Gross, “Power Reduction in CNN Pooling Layers with a Preliminary Partial Computation Strategy,” in *2018 16th IEEE International New Circuits and Systems Conference (NEWCAS)*, Montreal, QC: IEEE, Jun. 2018, pp. 125–129. doi: 10.1109/NEWCAS.2018.8585433.
- [36] S. Liu and W. Deng, “Very deep convolutional neural network based image classification using small training sample size,” in *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, Kuala Lumpur, Malaysia: IEEE, Nov. 2015, pp. 730–734. doi: 10.1109/ACPR.2015.7486599.
- [37] N. Ketkar and J. Moolayil, “Convolutional Neural Networks,” in *Deep Learning with Python*, Berkeley, CA: Apress, 2021, pp. 197–242. doi: 10.1007/978-1-4842-5364-9_6.
- [38] H. Gholamalizadeh and H. Khosravi, “Pooling Methods in Deep Neural Networks, a Review,” Sep. 16, 2020, *arXiv: arXiv:2009.07485*. Accessed: Sep. 12, 2024. [Online]. Available: <http://arxiv.org/abs/2009.07485>
- [39] X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, and M. Parmar, “A review of convolutional neural networks in computer vision,” *Artif Intell Rev*, vol. 57, no. 4, p. 99, Mar. 2024, doi: 10.1007/s10462-024-10721-6.
- [40] L. Breiman, “Bagging predictors,” *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996, doi: 10.1023/A:1018054314350.
- [41] Y. Freund and R. E. Schapire, “Experiments with a New Boosting Algorithm,” *International Conference on Machine Learning*, 1996, [Online]. Available: <https://api.semanticscholar.org/CorpusID:1836349>
- [42] P. Smyth and D. H. Wolpert, “Stacked Density Estimation,” *Neural Information Processing Systems*, 1997, [Online]. Available: <https://api.semanticscholar.org/CorpusID:463205>
- [43] B. Lakshminarayanan, A. Pritzel, and C. Blundell, “Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles,” 2016, doi: 10.48550/ARXIV.1612.01474.
- [44] E. Ismanto, A. Fadlil, A. Yudhana, and K. Kitagawa, “A Comparative Study of Improved Ensemble Learning Algorithms for Patient Severity Condition Classification,” *Journal of Electronics, Electromedical Engineering, and Medical Informatics (JEEEMI)*, vol. Vol 6, no. No 3, Jul. 2024.
- [45] M. Yildirim and A. Cinar, “Classification of Alzheimer’s Disease MRI Images with CNN Based Hybrid Method,” *ISI*, vol. 25, no. 4, pp. 413–418, Sep. 2020, doi: 10.18280/isi.250402.
- [46] Ahmad Badruzzaman and Aniati Murni Arymurthy, “A Comparative Study of Convolutional Neural Network in Detecting

- Blast Cells for Diagnose Acute Myeloid Leukemia,” *j.electron.electromedical.eng.med.inform*, vol. 6, no. 1, pp. 84–91, Jan. 2024, doi: 10.35882/jeemi.v6i1.354.
- [47] K. M. T. Ahmed, Z. Rahman, R. Shaikh, and S. I. Hossain, “Malaria Parasite Detection Using CNN-Based Ensemble Technique on Blood Smear Images,” in *2023 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, Chittagong, Bangladesh: IEEE, Feb. 2023, pp. 1–4. doi: 10.1109/ECCE57851.2023.10101524.
- [48] I. Nurcahyati, T. Saragih Hamonangan, A. Farmadi, D. Kartini, and M. Muliadi, “Classification of Lung Disease in X-Ray Images Using Gray Level Co-Occurrence Matrix Method and Convolutional Neural Network,” vol. Vol 6, no. No 4, pp. 332–242, Oct. 2024.
- [49] S. Asif, S. U. R. Khan, X. Zheng, and M. Zhao, “MOZZIENET: A deep learning approach to efficiently detect malaria parasites in blood smear images.” *Int J Imaging Syst Tech*, vol. 34, no. 1, p. e22953, Jan. 2024, doi: 10.1002/ima.22953.

About the Author



Bioinformatics and IoT.

Dr. Kuntala Boruah is currently an Assistant Professor in the Department of Computer Applications at Sibsagar University and former Assistant Professor at Assam Rajiv Gandhi University of Cooperative Management (ARGUCOM), Sivasagar, Assam, India. She received her B.Sc. in Physics from Gauhati University in 2008 and her MCA from Dibrugarh University in 2011. She earned her Ph.D from Tezpur University in 2019. Her research interests include DNA Computing, Deep Learning,



(IoT).

Mr. Dilip Kumar Baruah received his BCA and MCA degree from Dibrugarh University in 2009 and 2012 respectively. He qualified the National Eligibility Test (NET) in January 2019. Currently, he is a research scholar at Assam Rajiv Gandhi University of Cooperative Management (ARGUCOM) in Sivasagar, Assam, India. His research interests include Deep Learning, Biomedical Imaging, and the Internet of Things