CHC6781

Machine Vision

Sino-British Collaborative Education

Chengdu University of Technology

&

Oxford Brookes University

Dual Convolution Neural Networks of Ensemble learning with Attention Mechanism for Classification Task using Inception model and Res-Net model

Student number: 202018010314
Student Name: Aaron
Date: 143h Dec 2024

Class: Machine Vision

Content

Abstract

In this era of data-driven decision-making prevails, machine vision is emerging as a critical front-runner in the technological revolution, particularly in the field of medical diagnostics. This study leverages the formidable capabilities of Dual Convolution Neural Networks (CNNs), enhanced by an attention mechanism, to amplify the accuracy of image classification tasks. It introduces custom accuracy functions and cross-entropy loss functions. The ensemble framework taps into the unique feature detection strengths inherent to both employed models, while the attention mechanism deliberately focuses on significant regions within the images, resulting in a powerful and meticulously refined diagnostic tool. At the same time, it is hoped that in the future, convolutional neural networks can help people solve better medical problems.

Keywords: Dual Convolution Neural Networks, Ensemble Learning, Attention Mechanism

1. Introduction

This report is based on a CNN network to identify and classify whether malaria cells are infected. The aim is to use CNN network to build a unique algorithm to help doctors identify parasitic malaria cells and improve their work efficiency. This article uses a dataset of malaria cells from the Kaggle website, which includes a total of 27558 images[1].

This project uses a model that combines two CNN networks. Each CNN network has its own attention mechanism. One of the networks incorporates residual blocks for model adjustments. Through this technique, hyperparameters were fine-tuned, resulting in a well-tailored model. This model was then compared with others for evaluation.

2. Related Work

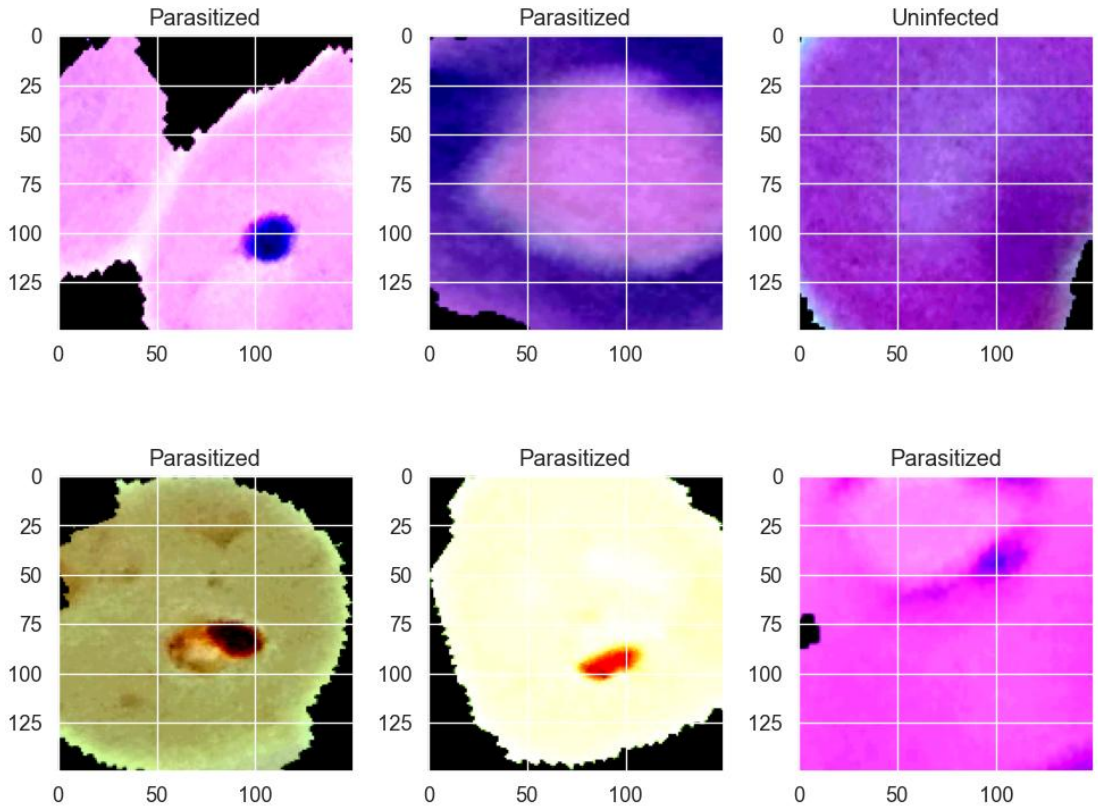
The detection of malaria parasites from blood smears is a challenging task that usually requires skilled medical personnel and is time-consuming. To overcome these limitations, several studies have proposed the use of deep learning techniques for malaria diagnosis. For instance, Cho, Y. S. developed an AI solution for malaria infection diagnosis using the CNN algorithm [2] . The model achieved high accuracy in classifying malaria-infected and non-infected cells, demonstrating the potential of machine learning-based diagnostic methods. Similarly, Ifeanyi, O. C. presented a mobile malaria disease detection system based on CNN, which allowed for the classification of patient blood smears using a mobile app[3].

This approach showed superior results in image classification and could provide a scalable solution for accurate malaria diagnosis. Furthermore, Nivaan, G. V. proposed a new model for detecting malaria-infected red blood cells among other normal and mutated cells, utilizing CNN and achieving high accuracy [4]. These studies highlight the effectiveness of CNN models in detecting malaria parasites and improving healthcare operational capabilities. Additionally, Jabbar, M. presented a Deep Convolutional Neural Network model for accurate malaria diagnosis, addressing issues related to model complexity and class imbalance[5]. The proposed model outperformed previous works and achieved reliable classification results. Overall, these studies demonstrate the potential of CNN models in malaria diagnosis and highlight their advantages over traditional methods, such as microscopy analysis, in terms of accuracy, processing time, and cost-effectiveness.

3. Material and method

3.1 Data set

To build and evaluate the model, the Malaria Cell Images dataset from Kaggle was used[1]. The dataset includes cells that have been parasitized and cells that remain uninfected. The classification of these two types of cells is shown in the figure below:



The downloaded data set from the website has undergone preliminary processing. The images were resized and subjected to data augmentation, ensuring uniformity in image format within the dataset for more accurate evaluation.

After processing, all the images were divided into 70% for the training set, 15% for the validation set, and 15% for the test set. The automatic segmentation of 27,558 images was implemented directly using Python code, and once this task was performed, the code for splitting the dataset was commented out.

3.2 Proposed Model

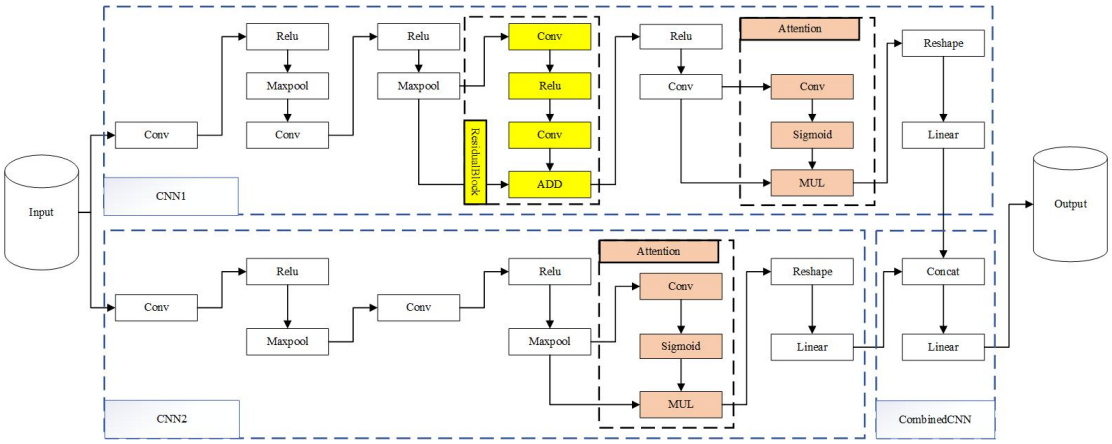


Figure 1. CNN model

In the provided figure, three neural network models are defined: AttentionModule, ResidualBlock, CNN1, CNN2, and CombinedCNN, which combines the features of the two CNN models.

Attention Module

This model is a simple attention mechanism layer designed to enhance the network's feature selection capability. It uses a 1x1 convolutional layer to generate an attention weight map (attention_map), and then employs a sigmoid activation function to limit these weights between 0 and 1. The attention weight map is then multiplied with the input feature map, allowing the network to emphasize significant features and suppress less important ones.

Residual-Block

This class defines a residual block that includes two convolutional 3x3 layers. The purpose of the residual block is to address the problems of vanishing or exploding gradients that may occur in deep networks. In this design, the output of the block is the sum of the input and output of the two convolutional operations. The ReLU nonlinear activation function is applied after each convolution.

CNN1

CNN1 is a convolutional neural network consisting of three convolutional layers, a global pooling layer, a residual block, and an attention mechanism layer. Finally, it has a fully connected layer that outputs 128 size features. This architecture incorporates some

common design principles from deep convolutional networks, such as the use of pooling layers to reduce the dimensionality of features, residual connections to preserve information flow, and attention mechanisms to focus on relevant features.

CNN2

CNN2 is similar to CNN1 but uses a different number of channels (64 instead of 32) after the second convolutional layer and incorporates an attention mechanism. This model also uses the attention mechanism before the fully connected layer, which has an output feature size of 128.

CombinedCNN

Finally, the CombinedCNN class combines the features of CNN1 and CNN2. It processes the input separately via CNN1 and CNN2, then concatenates their output features, and finally produces the classification result through a fully connected layer.

3.3 Model Evaluation

The evaluation function has been completed, and the Jupyter notebook includes visualization of ten key metrics: loss, accuracy, ROC-AUC curve, recall, precision, sensitivity, specificity, confusion matrix, f1-score, and the precision-recall curve.

In the process of evaluating the model, custom loss and accuracy functions were implemented, and their principles are as follows:

- **Loss function：**

calculate Softmax probabilities：

$$\text{max\_outputs} = \max(outputs, \text{dim} = 1, \text{keepdim} = True).values$$
$$\text{exp\_outputs} = \exp(outputs - \text{max\_outputs}) \ \text{\#avoiding exponential overflow}$$
$$\text{probs} = \frac{\text{exp\_outputs}}{\text{exp\_outputs.sum(dim=1, keepdim=True)}}$$

Figure 2

obtaining the probability of the target class：

$$\text{target\_probs} = \text{probs}[range(\text{batch\_size}), \text{targets}]$$

Figure 3

adding a constant to prevent logarithmic computation issues：

$$\text{eps} = 1 \times 10^{-8}$$
$$\text{target\_probs} = \text{torch.clamp(target\_probs, min=eps)}$$

Figure 4

calculate cross-entropy loss：

$$\text{loss} = -\log(\text{target\_probs}).mean()$$

Figure 5

- **Accuracy formula：**

$$\text{accuracy} = \frac{\text{correct predictions}}{\text{total predictions}}$$

Figure 6

The other indicators are as follows：

Recall = TP / (TP + FN)

Precision = TP / (TP + FP)

Sensitivity = TP / (TP + FN)

Specificity = TN / (TN + FP)

F1 Score = 2 * (Precision * Recall) / (Precision + Recall)

Confusion Matrix: TP (True Positives): The cases in which the actual class is positive and the model also predicts positive. TN (True Negatives): The cases in which the actual class is negative and the model also predicts negative. FP (False Positives): The cases in which the actual class is negative but the model predicts positive. FN (False Negatives): The cases in which the actual class is positive but the model predicts negative.

Precision-Recall Curve: The x-axis represents Recall. The y-axis represents Precision.

3.4 Parameter Tuning

Before Parameter Tuning:

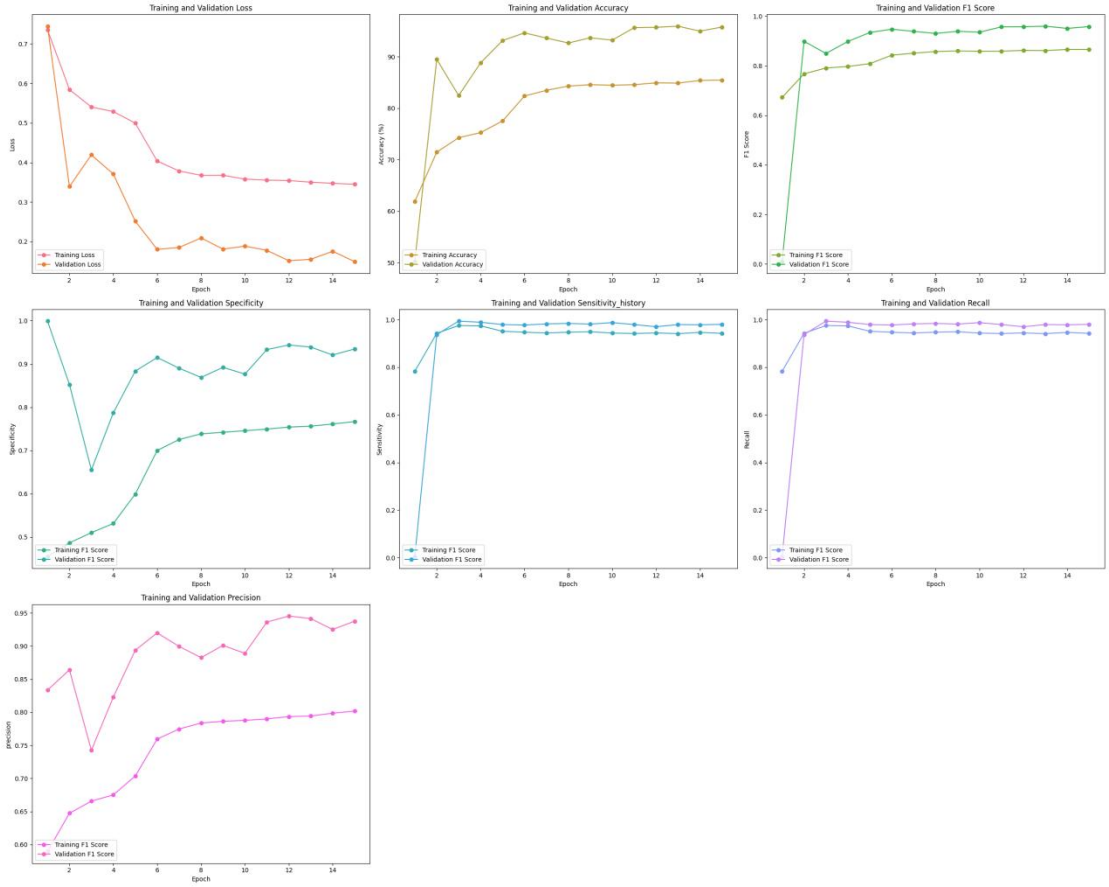


Figure 7

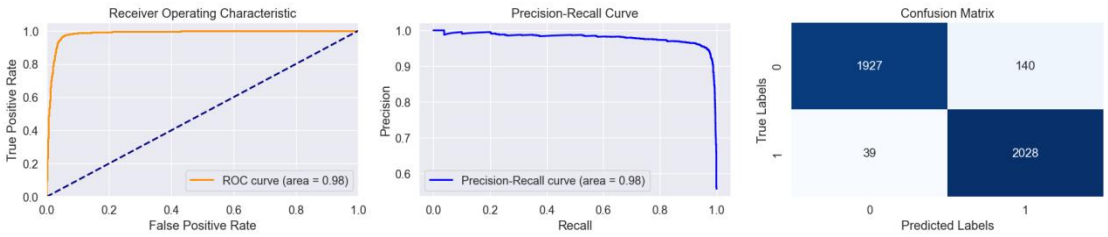


Figure 8

```
Accuracy: 0.9567005321722303
ROC-AUC: 0.984000192861819
Recall: 0.9811320754716981
Precision: 0.9354243542435424
F1-score: 0.9577331759149941
Confusion Matrix:
[[1927  140]
 [  39 2028]]
Precision-Recall AUC: 0.9792851609310605
Sensitivity: 0.9811320754716981
Specificity: 0.9322689888727624
```

Figure 9

In terms of training performance, the model demonstrates outstanding results, achieving high accuracy, precision, recall, and F1 scores. With an increase in training epochs, the

training loss steadily decreases, indicating that the model effectively learns features from the training data. Regarding validation performance, the model exhibits robust generalization capabilities, maintaining elevated levels of accuracy, precision, recall, and F1 scores. The significant improvement in the AUC-ROC value indicates the model's excellent ability to distinguish between different classes. Particularly noteworthy is the balanced performance in high specificity sensitivity and high specificity, further reinforcing its superiority.

However, a shortfall is evident in the model's training set accuracy and loss performance, which is notably inferior to that of the validation set. At this point, it is necessary to make certain parameter adjustments or examine whether there is any data leakage in the training set images.

After Parameter Tuning:

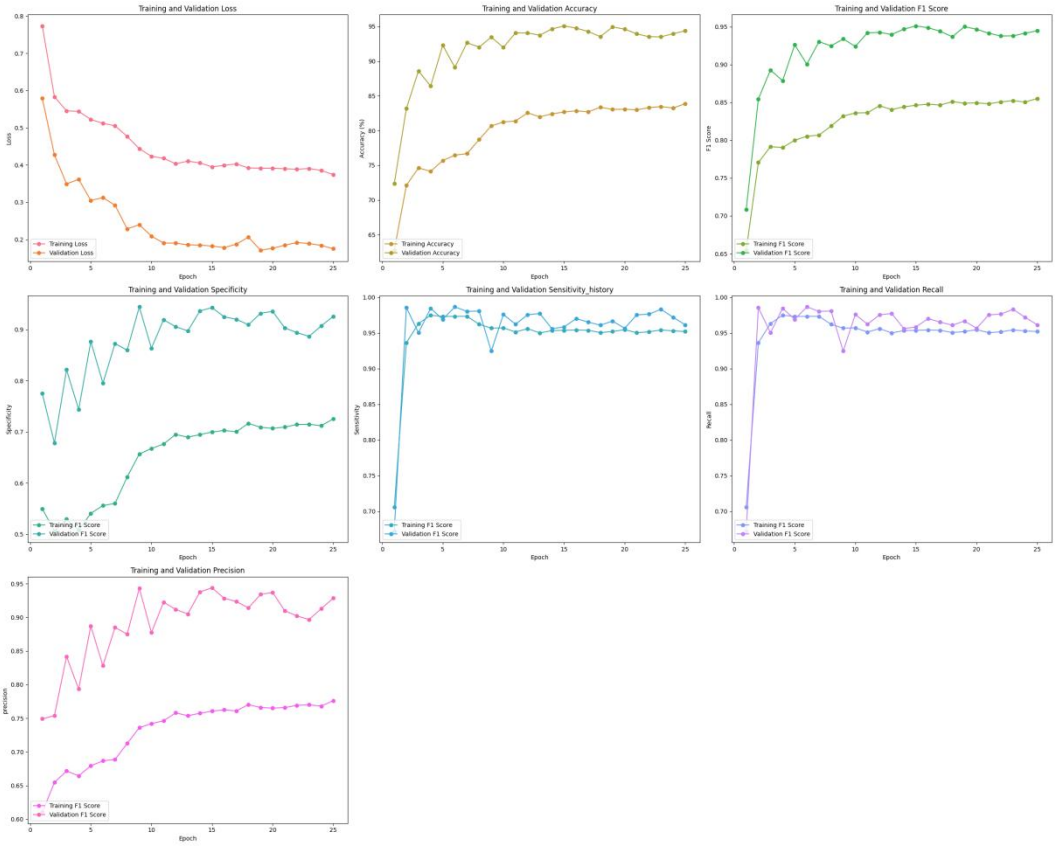


Figure 10

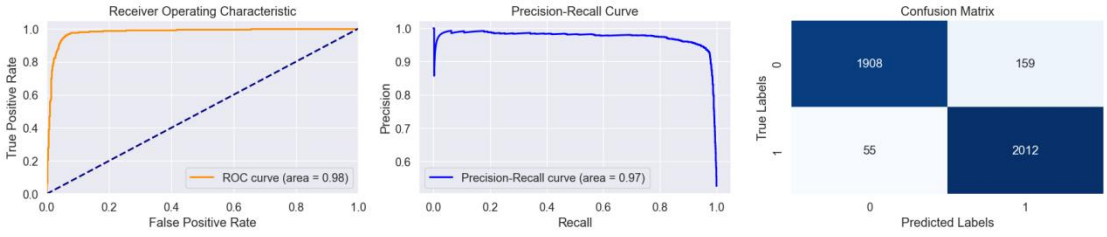


Figure 11

```
Accuracy: 0.9482341557813256
ROC-AUC: 0.9793946807118754
Recall: 0.9733913884857281
Precision: 0.9267618608935975
F1-score: 0.9495044832468146
Confusion Matrix:
[[1908  159]
 [  55 2012]]
Precision-Recall AUC: 0.9733333938498441
Sensitivity: 0.9733913884857281
Specificity: 0.9230769230769231
```

Figure 12

Recent attempts to adjust the model's parameters have failed to yield significant improvements; in fact, in some cases, they have even led to a deterioration in performance. The adjustment measures specifically involved setting the number of epochs to 25, coupled with the introduction of l2 regularization to the CNN2 network, in an effort to reduce the discrepancy between the training and validation set metrics and to prevent overfitting.

Analysis of the associated training data charts revealed a clear pattern: the performance of the model reached a plateau after 15 epochs, with subsequent rounds showing little to no substantial improvement. This trend is particularly evident in the data charts, where the line graphs progressively approximate a straight line, suggesting that the model's training capability on the current dataset is nearing its limit.

3.5 Execution Environment

The model discussed in this report was implemented on a Windows laptop equipped with the Nvidia GeForce RTX 3060 Laptop graphics card chip, with the CPU using AMD Ryzen 7 5800H. The RAM size is 16GB. This device ensures the smooth progress of the entire model research.

4. Experiment Result

Deep learning has become an increasingly popular tool across various fields, including medical applications. The use of different models, such as AlexNet, VGG-16, ResNet50, and Mobilenet_V2, has been explored in the medical domain to address different challenges and improve performance[6]. In this section, we will compare the CNN composite model constructed for this project against other types of models, including those created by different developers using the same dataset. Additionally, a relatively fair comparison will be made with pretrained models. This approach will demonstrate the performance and shortcomings of the model in question.

4.1 Direct Comparison

	Accuracy	Loss	Recall	ROC-AUC	F1-score	Sensitivity	Specificity	Precision	Precision-Recall
Provided	0.957	0.18	0.981	0.98	0.96	0.98	0.93	0.94	0.98
K. S. Gill et al.[7]	0.96	0.12	0.95	X	0.96	X	X	0.97	X
P. K. Maduri et al. [8]	0.82	0.47	X	X	X	X	X	X	X
W. R. W. M. Razin et.al.[9]	0.96	X	0.97	X	X	X	X	X	X
A. A. Alonso-Ramírez et al.[10]	0.98	X	0.98	X	0.97	X	X	0.97	X
K. M. F. Fuhad [11]	0.98	X	X	X	X	X	X	X	X

After parameter tuning, the VGG19 model exhibited superior performance[7], with its only drawback perhaps being the lengthy training time required due to the large volume of data. The CNN network by Maduri, P. K., et al. did not demonstrate exceptional performance,

which may be attributed to the overly simplistic structure of their model. However, they still provided evidence of CNN's positive impact on the medical field [8].

The YOLOv5 algorithm is very suitable for combining with CNN neural networks [9]. Although the parameters mentioned in the report are relatively small, it can still be seen that the performance of combining the two is not inferior to other models.

From the report by A. A. Alonso-Ramírez et al., [10]we learn that employing a CNN-BiLSTM architecture can greatly improve the outcomes of the data obtained. Their team's report has provided us with a training model of exceptional performance. There is also a growing trend of deploying models trained with the TensorFlow framework on mobile devices, which can assist people in quickly making judgments or identifying certain situations; [11]The model designed by K. M. F. Fuhad's team is another example of a model with robust performance.

4.2 Fair Comparison

In this fair comparison, VGG16, ResNet50, MobileNetV2, and AlexNet will be compared on the same dataset. Due to the complexity of the VGG16 model, and the dataset containing a total of 27,558 images, running one epoch takes 30 minutes on the devices under discussion. Therefore, we will run 8 epochs while keeping all other conditions as constant as possible.

	Accuracy	Loss	Recall	ROC-AUC	F1-score	Sensitivity	Specificity	Precision	Precision-Recall
Provided（epoch 10）	0.94	0.25	0.96	0.98	0.94	0.96	0.93	0.93	0.96
VGG16（8 epoch）	0.93	0.20	0.94	0.98	0.93	0.94	0.91	0.92	0.98
ResNet 50 (10 epoch)	0.96	0.18	0.97	0.98	0.96	0.97	0.94	0.94	0.98
Mobilenet_v2 (10 epoch)	0.90	0.25	0.93	0.96	0.91	0.93	0.88	0.88	0.96
AlexNet (10 epoch)	0.91	0.25	0.96	0.96	0.91	0.96	0.86	0.87	0.96

MobileNetV2, VGG16, Resnet50, and AlexNet are four distinct deep learning architectures, each with its own characteristics and application scenarios. MobileNetV2 is specifically designed for applications that require lightweight models, and it shows good performance when running smaller models[12]. Although VGG16 has a slower processing speed[13], its superior performance makes it suitable for applications that require real-time feedback. Resnet50 introduced the concept of residual blocks, which makes training deep networks easier and allows for faster convergence[14]. Finally, as one of the starting points for deep learning research, AlexNet continues to show excellent performance in large-scale image classification tasks, such as those on ImageNet[15].

Based on the table, we can see that all the models perform well, but MobileNetV2 and AlexNet have slightly lower accuracy compared to the other models, and both models' Precision and Specificity metrics are not very strong. The reason for this could be attributed to the simpler network architectures of these models.

5. Conclusion and future work

This report presents a training model that is based on a hybrid of two convolutional neural networks and demonstrates strong comprehensive performance in fitting image data. This model achieves an accuracy rate close to 96% for rice classification. Other metrics including loss, ROC-AUC curve, recall, precision, sensitivity, specificity, confusion matrix, f1-score, and precision-recall curve also show excellent performance. Overall, this model has successfully utilized residual blocks and attention mechanisms, and has also effectively implemented customized loss functions and precision functions.

However, there were still problems during the training of this model that could not be resolved through various parameter adjustments within the limited time available. Specifically, the validation and test set data were found to be quite similar, yet the training set consistently underperformed compared to the former two. This problem could be due to a portion of the training set being contaminated during dataset division, improper data preprocessing, or reverse overfitting. Comparative results obtained from various pre-trained models indicated that there was indeed some contaminated data in the training set. This was also mentioned in the report by K. M. F. Fuhad and his team, where they noted that more than 700 pictures of parasitized malaria cells were misclassified as uninfected[11]. In addition, the training data in the article by K. S. Gill et al. also showed that the training set's data was inferior to that of the validation set[7].

To solve or at least minimize this problem, future efforts should focus on improved categorization and cleansing of datasets, as well as a deliberate increase in model complexity to ensure that both training and validation sets reasonably represent the entire data distribution.

In the future medical field, the use of CNN models to assist doctors in diagnosis and treatment will become increasingly common and the technology behind it will also become more mature and reliable. The impact of these advances will be profound: from urban hospitals to remote clinics, CNN models will provide support in the medical decision-making process. This will democratize high-quality medical care and ensure that even patients in the most underserved areas can benefit from accurate diagnoses and effective treatments. In addition, by automating routine tasks and analyses, AI models will allow healthcare providers to reallocate more time to patient care and more complex cases, enhancing the overall healthcare delivery experience.

6. Reference

[1] ‘Malaria Cell Images Dataset’. Accessed: Dec. 13, 2023. [Online]. Available: https://www.kaggle.com/datasets/iarunava/cell-images-for-detecting-malaria

[2] ‘Healthcare | Free Full-Text | Applying Machine Learning to Healthcare Operations Management: CNN-Based Model for Malaria Diagnosis’. Accessed: Dec. 14, 2023. [Online]. Available: https://www.mdpi.com/2227-9032/11/12/1779

[3] ‘(PDF) Malaria disease detection system based on convolutional neural network (CNN)’. Accessed: Dec. 14, 2023. [Online]. Available: https://www.researchgate.net/publication/366182670_Malaria_disease_detection_system_based_on_convolutional_neural_network_CNN

[4] G. Nivaan, ‘Image Recognition of Malaria-infected Red Blood Cells among Other Normal and Cancer-Mutated Cells Using CNN’, *JINAV J. Inf. Vis.*, vol. 3, pp. 62–70, Jul. 2022, doi: 10.35877/454RI.jinav1552.

[5] ‘(PDF) Diagnosis of Malaria Infected Blood Cell Digital Images using Deep Convolutional Neural Networks’. Accessed: Dec. 14, 2023. [Online]. Available: https://www.researchgate.net/publication/358247947_Diagnosis_of_Malaria_Infected_Blood_Cell_Digital_Images_using_Deep_Convolutional_Neural_Networks

[6] ‘2311.08655.pdf’. Accessed: Dec. 14, 2023. [Online]. Available: https://arxiv.org/ftp/arxiv/papers/2311/2311.08655.pdf

[7] K. S. Gill, V. Anand, and R. Gupta, ‘An Efficient VGG19 Framework for Malaria Detection in Blood Cell Images’, in *2023 3rd Asian Conference on Innovation in Technology (ASIANCON)*, Aug. 2023, pp. 1–4. doi: 10.1109/ASIANCON58793.2023.10270637.

[8] P. K. Maduri, Shalu, S. Agrawal, A. Rai, and S. Chaubey, ‘Malaria Detection Using Image Processing And Machine Learning’, in *2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N)*, Dec. 2021, pp. 1789–1792. doi: 10.1109/ICAC3N53548.2021.9725557.

[9] W. R. W. M. Razin, T. S. Gunawan, M. Kartiwi, and N. Md. Yusoff, ‘Malaria Parasite Detection and Classification using CNN and YOLOv5 Architectures’, in *2022 IEEE 8th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA)*, Sep. 2022, pp. 277–281. doi: 10.1109/ICSIMA55652.2022.9928992.

[10] A. A. Alonso-Ramírez *et al.*, ‘Classifying Parasitized and Uninfected Malaria Red Blood Cells Using Convolutional-Recurrent Neural Networks’, *IEEE Access*, vol. 10, pp. 97348–97359, 2022, doi: 10.1109/ACCESS.2022.3206266.

[11] K. M. F. Fuhad, J. F. Tuba, M. 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, Art. no. 5, May 2020, doi: 10.3390/diagnostics10050329.

[12] T. B. Shahi, C. Sitaula, A. Neupane, and W. Guo, 'Fruit classification using attention-based MobileNetV2 for industrial applications', *PLOS ONE*, vol. 17, no. 2, p. e0264586, Feb. 2022, doi: 10.1371/journal.pone.0264586.

[13] '[PDF] Smart Pothole Detection Using Deep Learning Based on Dilated Convolution | Semantic Scholar'. Accessed: Dec. 14, 2023. [Online]. Available: https://www.semanticscholar.org/paper/Smart-Pothole-Detection-Using-Deep-Learning-Based-Ragab/2323dfe736ad51cf98a6763d2f275b6d3f1c1f46

[14] '[PDF] Go Wide, Then Narrow: Efficient Training of Deep Thin Networks | Semantic Scholar'. Accessed: Dec. 14, 2023. [Online]. Available: https://www.semanticscholar.org/paper/Go-Wide%2C-Then-Narrow%3A-Efficient-Training-of-Deep-Zhou-Ye/c898b685d9167fe48d5a6401e2007c59ee0bdacc

[15] '[PDF] 100-epoch ImageNet Training with AlexNet in 24 Minutes | Semantic Scholar'. Accessed: Dec. 14, 2023. [Online]. Available: https://www.semanticscholar.org/paper/100-epoch-ImageNet-Training-with-AlexNet-in-24-You-Zhang/fe436d3426eec6bee8072b2d442c09ccd23fe069