

Automated Malaria Cell Classification Using Deep Learning

Anjum Afrooz Shaik
Dept. of Applied Computing
Michigan Technological
University (of Aff.)
Houghton, Michigan

Venkata AnuSandeep Pamarthi
Dept. of Applied Computing
Michigan Technological
University (of Aff.)
Houghton, Michigan

Vijay Susmitha Bommini
Dept. of Applied Computing
Michigan Technological
University (of Aff.)
Houghton, Michigan

I. ABSTRACT

In the regions where malaria remains a significant public health challenge, the need for accurate and timely diagnosis is paramount. Conventional diagnostic methods, such as microscopic examination of blood smears, are often labor-intensive, time-consuming, and require skilled personnel, posing barriers to effective diagnosis, particularly in resource-limited settings. To address these challenges, this study investigates the potential of deep learning techniques in automated malaria cell classification. By harnessing the power of convolutional neural networks (CNNs), including a customized CNN architecture as well as pre-trained models like ResNet-50 and EfficientNet B2, we aimed to develop robust and accurate classifiers capable of distinguishing between infected and uninfected red blood cells in microscopic images. Through extensive experimentation and comparative analysis, we evaluated the performance of these models on unseen data.

Our results showcase the effectiveness of deep learning-based approaches in malaria cell classification, with the customized CNN achieving an impressive accuracy rate of approximately 94%. These findings underscore the transformative potential of artificial intelligence in revolutionizing malaria diagnosis, offering scalable solutions that can streamline the diagnostic process, reduce dependency on human expertise, and improve diagnostic accuracy. By enabling rapid and accurate identification of malaria-infected cells, these automated classification systems have the potential to enhance healthcare delivery, facilitate early intervention, and ultimately contribute to the global efforts aimed at reducing the burden of malaria. Moreover, the successful implementation of deep learning techniques in malaria diagnosis highlights their broader applicability in addressing other infectious diseases, paving the way for future advancements in disease detection and management.

A. Keywords: Malaria, Image Cell Classification, CNN, ReLu, Dropout, Pooling, fully connected, ResNet50, EfficientNetB2, Accuracy, DL

II. INTRODUCTION

Malaria, a mosquito-borne disease caused by Plasmodium parasites, remains a major worldwide health issue, disproportionately affecting vulnerable communities in tropical countries. Despite advances in both management and prevention, the malaria severity remains high, with millions of cases and hundreds of thousands of fatalities reported each year. In the year 2022 alone, it is estimated that nearly 249 million malaria cases and 608,000 deaths were reported across 85 countries, by World Health Organization (WHO) with African Region bearing the majority of the disease burden [1].

Malaria has an impact that goes beyond human health, affecting economics, society, and the healthcare system. Malaria holds a tremendous barrier to global public health initiatives, putting nearly 3.2 billion people at risk and costing billions of dollars in lost economic productivity each year [2].

The motivation to address malaria through automated cell classification using deep learning arises from the urgent need to enhance diagnostic capabilities, particularly in resource-constrained settings where traditional diagnostic methods may be limited. A timely and precise diagnosis is critical for commencing appropriate treatment, avoiding complications, and limiting transmission.

By leveraging advancements in deep learning and image analysis, automated classification systems offer the potential to revolutionize malaria diagnosis, enabling rapid and scalable identification of infected cells with high accuracy. Such technology holds promise for improving healthcare delivery, especially in regions heavily burdened by malaria, where access to trained medical professionals and laboratory facilities may be limited.

Furthermore, by augmenting existing efforts in malaria prevention and control, automated cell classification aligns with broader public health initiatives aimed at reducing the global malaria burden. Through collaboration with international partners, governmental agencies, and healthcare organizations, innovative approaches like automated cell classification contribute to the overarching goal of malaria elimination and improving health outcomes worldwide.

A. Background Knowledge

Machine learning, with neural networks at its core, has witnessed extensive exploration across diverse domains, including healthcare. Researchers have successfully employed neural networks for tasks such as stock price prediction, facial recognition, cancer type estimation, and music genre classification. Within medical imaging, Convolutional Neural Networks (CNNs) have demonstrated remarkable efficacy in detecting diabetic retinopathy, classifying skin lesions, and identifying tumors.

B. Related Work

The field of medical image classification, including malaria cell identification, has seen several state-of-the-art approaches. Notably, research has gravitated toward CNN-based models due to their ability to capture intricate patterns within images. Some noteworthy papers which explored the application of deep learning (DL) and convolutional neural networks (CNNs) for malaria detection from blood smear images include:

Raj et al. [3] proposed a DL-based approach utilizing CNNs, achieving high accuracy through different optimizers for training and validation. Minarno et al. [4] used the Inception-V3 architecture to classify malaria cell images, achieving a maximum accuracy of 97% with the RMSprop optimizer.

Silka et al. [5] achieved remarkable accuracy of 99.68% using a CNN architecture, emphasizing its potential for rapid and accurate diagnosis. Razin et al. [6] employed a DL strategy with CNN and the YOLOv5 algorithm, achieving an accuracy of 96.21%. Krishnadas et al. [7] demonstrated automation of malaria parasite identification using YOLOv5 and scaled YOLOv4.

Sifat et al. [8] developed an automated approach with CNN and VGG16 models, achieving segmentation accuracy of 97.67%. Sampathila et al. [9] utilized image segmentation and neural networks to detect malaria, achieving a training accuracy of 97.2%.

Li et al. [10] proposed a hybrid model combining CNNs with support vector machines, significantly improving prediction accuracy. Roy et al. [11] applied

image processing techniques to detect malaria parasites, achieving a detection rate of 90.0%. Nayak et al. [12] and Maduri et al. [13] employed ResNet and CNN models, respectively, for malaria diagnosis, achieving high accuracy in classifying infected and uninfected cells.

Francies et al. [14] and Setyawan et al. [15] described the development and application of YOLO and CNN techniques for malaria parasite classification, emphasizing rapid diagnosis. Hung and Carpenter [16] applied object detection algorithms for malaria-infected cell identification, achieving improved accuracy. Var and Tek [17] employed transfer learning with VGG19 to detect and classify malaria parasites from blood smear images.

Yang et al. [18] developed a DL system for mobile devices to detect malaria parasites in thick blood smear images. Iradukunda et al. [19] used extreme learning machine models to classify malaria-infected red blood cells. Prakash et al. [20] and Shekar et al. [21] proposed CNN-based approaches for malaria detection, achieving high accuracy and resistance to overfitting. Umer et al. [22] introduced a stacked CNN design for malaria detection without handcrafted features.

Joshi et al. [23] developed a CNN-based technique for malaria diagnosis, outperforming existing methods in accuracy. Paul and Bania [24] built CNN models for predicting malaria from RBC images, focusing on infected and uninfected cells' differentiation. Aimi et al. [25] used k-means clustering and color segmentation models to detect malaria parasites. Savkare and Narote [26] applied statistical classifiers like SVM to automatically detect malaria parasites using textural and morphological information.

These studies collectively highlight the effectiveness of DL and CNN-based approaches in malaria detection, emphasizing the need for accuracy, speed, and automation in diagnosis to improve patient outcomes.

III. DATASET

The dataset used in this analysis, comprises of images for malaria diagnosis, was sourced from the National Institutes of Health (NIH) website <https://ceb.nlm.nih.gov/repositories/malaria-datasets/> [27], a common repository for medical imaging datasets.

It consists of 27,558 images, with 13,779 images classified under parasitized and 13,779 Non-parasitized. The dataset contains a substantial number of examples for both training and testing. We explicitly used 24,804 images for training, 5,511 images for validation, and 2,756 for testing.

The data preprocessing steps followed involves concatenating separate data frames containing file paths

for parasitized and uninfected cell images, followed by shuffling and splitting into training and testing sets. During training, image augmentation techniques are applied using Keras' ImageDataGenerator, including rotation, shifting, shearing, zooming, and flipping, to enhance the model's ability to generalize.

Pixel normalization is implicitly applied during model training to scale pixel values between 0 and 1. The images are of size 135x135 pixels with 3 color channels (RGB).

The primary features utilized in this dataset are the images themselves, representing both parasitized and uninfected cells. These images serve as input data for training and testing the classification model.

Visualizations have been created to showcase sample images from the dataset and illustrate the distribution of classes in both the training and testing sets. Additionally, image augmentation techniques have been applied to enhance the training process. Furthermore, during model training, "311" and "78" batches are generated for the training and validation sets, respectively. These batches are utilized iteratively to train and validate the model.

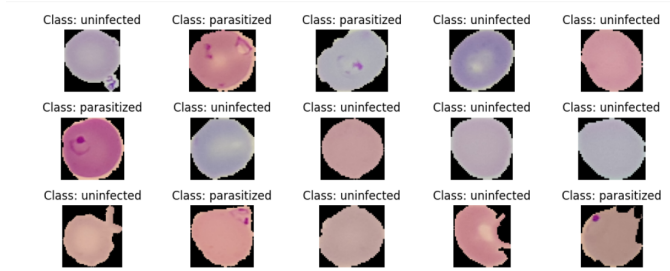


Fig. 1. Sample Images from the dataset along with class labels

IV. METHODS

In this section, we describe the learning algorithms utilized in our study, including Convolutional Neural Networks (CNNs) and Transfer Learning with pre-trained models such as ResNet50 and EfficientNetB2. We provide a brief overview of each algorithm's functionality, followed by relevant mathematical notations and descriptions of the training process.

A. Convolutional Neural Networks

Convolutional Neural Networks (CNNs) are a class of deep learning models specifically designed for processing structured grid data, such as images. A CNN typically consists of multiple layers, including convolutional layers, pooling layers, fully connected layers, activation layers, and dropout layers. During training, the CNN learns to automatically extract hierarchical features from

input images, which are then used for classification tasks.

- **Convolutional Layers** : The most important part of a CNN is its convolutional layer, which finds important features in an image using filters. These filters move across the image, adding up values to create feature maps. These maps then go through other layers to make the network smaller and deeper. Convolutional layers are vital for CNNs to understand images well, like for recognizing objects. Equation 1 helps calculate how many features come out of the convolution layer. The below given equation shows the calculation of the number of output parameters:

$$\text{No. of output parameters} = (((w \times h \times d) + 1) \times N) \quad (1)$$

- **Pooling Layers** : Pooling layers are frequently utilized in CNNs to decrease the size of feature maps generated by convolutional layers. Their primary purpose is to simplify network operations while retaining essential data in the feature maps. Two common pooling techniques are maximum pooling and average pooling. Maximum pooling selects the highest value within a specific region of the feature map, effectively reducing the size of feature maps and the computational load on the network. Conversely, average pooling computes the average value within the same region. Pooling helps prevent overfitting by reducing the number of parameters and computational resources required.

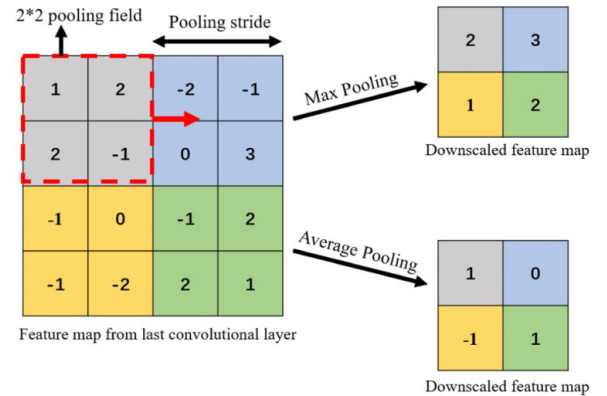


Fig. 2. Pooling Layer(2*2)

[29]

- **ReLU Activation** : ReLU (Rectified Linear Unit) layers play a crucial role in deep learning networks, particularly CNNs, by addressing the vanishing gradient problem. This activation function is widely

employed in deep networks to mitigate gradient vanishing issues by replacing all negative input values with zero. Mathematically, the ReLU function is represented by Equation 2, where 'i' signifies the input and 't' denotes the output. ReLU activation functions are straightforward to compute and have been demonstrated to enhance the performance of many deep learning models.

$$t = \max(0, i) \quad (2)$$

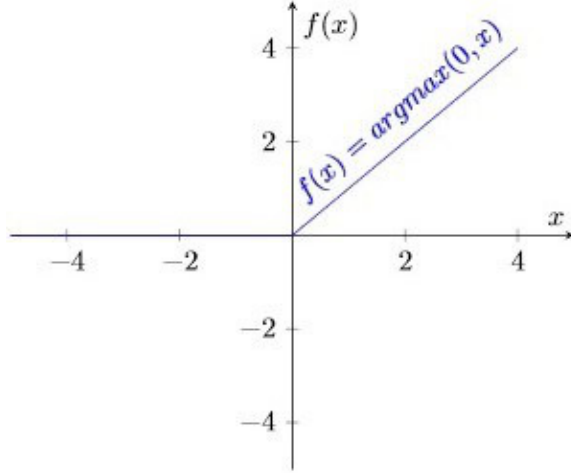


Fig. 3. ReLu Activation Function

[30]

- **Dropout** : The dropout layer is a regularization technique employed in deep learning neural networks to combat overfitting, depicted in Figure . During training, this layer randomly sets a predetermined portion of input units to 0, encouraging the network to learn diverse representations of the input data. This prevents the network from memorizing the training data excessively and promotes generalization to new inputs. Typically, the dropout rate, representing the proportion of input units set to 0, ranges between 0.5 and 0.8. During testing, all input units are utilized, and their activations are scaled by the inverse of the dropout rate to compensate for the reduced number of active units during training. Dropout is commonly employed to improve model performance and stability across various neural network architectures, including RNNs and CNNs

[30]

- **Fully Connected**: The fully connected (FC) layer serves as the core of a deep neural network (DNN). In this layer, every neuron is connected to each neuron in the layer above, as illustrated in Figure . Each neuron within an FC layer receives input from

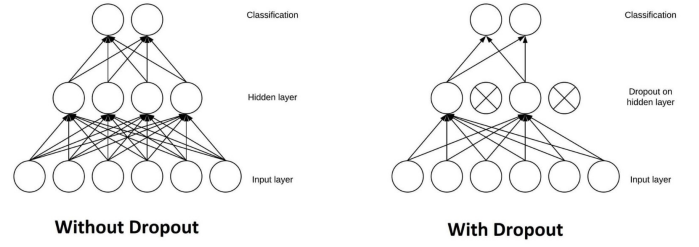


Fig. 4. Dropout Layer

the layer below, applies weights to these inputs, and then aggregates them. Subsequently, an activation function such as ReLU, Sigmoid, or Tanh is applied to the output of each neuron. These layers in deep learning models are instrumental in extracting high-level features from the raw input data by amalgamating various fundamental characteristics from the preceding layer. The output of the final FC layer can be utilized to derive class probabilities for classification tasks [28].

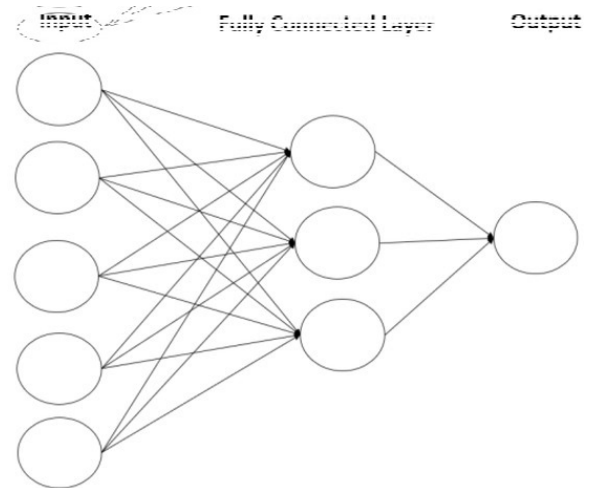


Fig. 5. Fully Connected Layer

[31]

B. Transfer learning with Pre - trained Models (ResNet50 and EfficientNetB2)

Transfer learning is a technique where a pre-trained model is leveraged as a starting point for a new task. In our study, we utilize the pre-trained ResNet50 and EfficientNetB2 models, which have been trained on large-scale image datasets. Transfer learning allows us to benefit from the rich feature representations learned by these models and adapt them to our specific classification task of detecting parasitized and uninfected cells.

ResNet-50, introduced in the paper "Deep Residual Learning for Image Recognition" by He et al., is renowned for its depth and effectiveness. It consists of 50 layers, including 48 convolutional layers, one MaxPool layer, and one average pool layer.

EfficientNet B2, on the other hand, is part of the EfficientNet family introduced in the paper "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks" by Tan et al. This series of CNNs are designed to achieve state-of-the-art performance while being computationally efficient.

EfficientNet B2 specifically strikes a balance between model size and accuracy, offering strong performance while remaining resource-efficient. Architecture is carefully crafted to achieve high performance in various computer vision tasks, including image classification, object detection, and image segmentation.

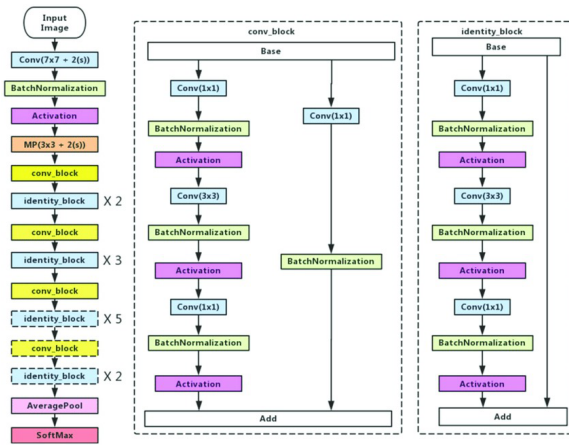


Fig. 6. ResNet Architecture

[?]

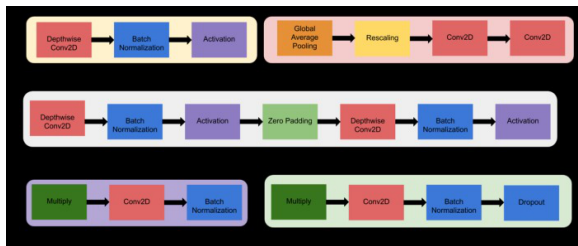


Fig. 7. EfficientB2 Modules Description

[?]

By employing these learning algorithms, incorporating activation layers and dropout for non-linearity and regularization, and utilizing the model architectures mentioned above, we aim to develop robust models capable

of accurately classifying parasitized and uninfected cells for malaria diagnosis.

V. EXPERIMENTS AND RESULTS

In our project, we employed various deep learning models, including ResNet50 and EfficientNetB2, to classify a dataset into infected and uninfected classes. Here, we present a combined analysis of the experimental setups, hyperparameters, performance metrics, results, and discussions for all models. We split the dataset into training, validation, and test sets. The training data was used with a generator, utilizing a batch size determined beforehand. For the ResNet50 and EfficientNetB2 models, we trained with 50 steps per epoch, covering approximately 16% of the total training data in each epoch. The Adam optimizer was employed for optimization during training.

Customized CNN : Our customized CNN is tailored for effective image classification, comprising several layers, each serving a distinct role. Initially, there are five convolutional layers, tasked with learning features from the images. These layers use different numbers of filters to extract patterns from the images, with output shapes varying according to the number of filters applied. Subsequently, max-pooling layers are employed to reduce the size of the features, facilitating computational efficiency, with output shapes being reduced compared to the input shapes.

Following this, fully connected layers analyze the features and generate predictions, with output shapes dependent on the number of neurons in each layer. Additionally, dropout layers are incorporated to prevent overfitting, promoting robust generalization to new data. In total, our CNN comprises 10 layers, encompassing convolutional and max-pooling layers. It is equipped with over 740,000 parameters, representing the values learned by the network during training to make accurate predictions.

ResNet50 : Our model is built upon the ResNet50 architecture, a pre-trained CNN. By leveraging ResNet50's pre-trained weights on the ImageNet dataset, we harness a robust foundation for feature extraction. The ResNet50 backbone processes input images, generating feature maps with dimensions (None, 5, 5, 2048), which encapsulate rich representations of the input data. On top of ResNet50, we add some custom layers tailored for our specific task. These layers reshape the extracted features and use them to make predictions. They include a Flatten layer to organize the data, followed by dense

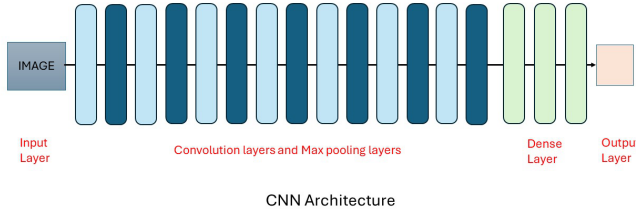


Fig. 8. Block diagram of Customized CNN model

layers for classification. During training, the model fine-tunes its parameters to improve its accuracy. It's trained to recognize patterns in batches of data, gradually improving its ability to classify images accurately. With a total of 36,695,425 parameters, including 13,107,713 trainable parameters and 23,587,712 non-trainable parameters inherited from ResNet50, our model exhibits robust performance.

EfficientNet B2 : The model is based on EfficientNetB2, a sophisticated architecture known for its efficiency in processing images. It's pre-trained on the ImageNet dataset, providing a strong foundation for feature extraction. The EfficientNetB2 backbone processes input images, producing feature maps with dimensions (None, 5, 5, 1408), which encapsulate crucial information about the images. To adapt the model to our specific task, we add custom layers on top of EfficientNetB2. These layers reshape the extracted features and use them for classification. A Flatten layer organizes the feature maps into a flat array, preparing them for further processing. The flattened feature vectors have a shape of (None, 35200). Next, dense layers are added for classification. During training, the model fine-tunes its parameters to improve its accuracy. It's trained to recognize patterns in batches of data, gradually improving its ability to classify images accurately. In total, the model has over 16 million parameters, which are adjusted during training to enhance its performance. The combination of EfficientNetB2's pre-trained features and custom layers makes our model effective for various image classification tasks.

RESULTS

A Comparative Analysis of Customized CNN, ResNet-50, and EfficientNetB2.

Customized CNN Performance: The customized CNN

exhibited fluctuating accuracy during training, starting at 50% and gradually improving to approximately 94% by the final epoch. Validation accuracy also displayed fluctuations but generally followed an increasing trend throughout training. The evaluation on unseen data revealed an accuracy score of approximately 94%, indicating the effectiveness of the model.

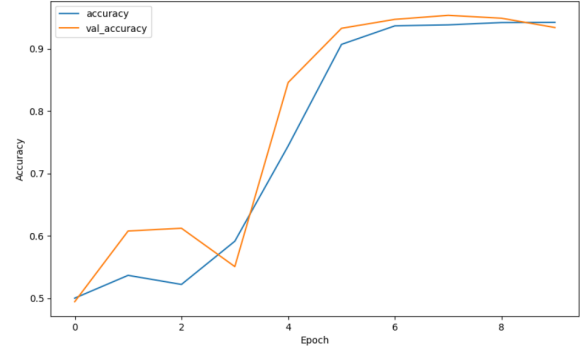


Fig. 9. Training vs Validation Accuracy (CNN)

ResNet-50 Performance: ResNet-50 demonstrated a higher initial accuracy of around 75%, steadily increasing to approximately 89% by the end of training. Validation accuracy exhibited consistent improvement over epochs, mirroring the training accuracy trend. Evaluation on unseen data yielded an accuracy score of around 89%, indicating respectable performance but slightly lower than the customized CNN.

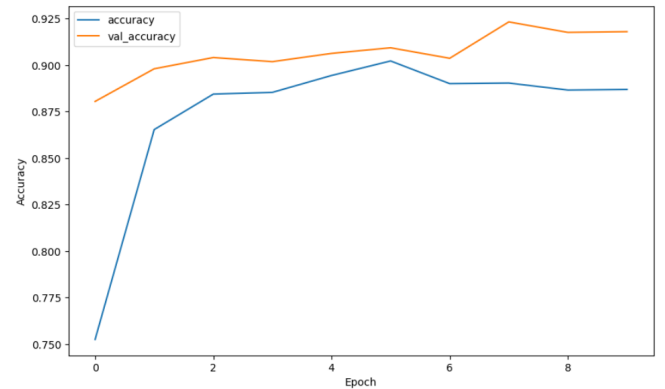


Fig. 10. Training vs Validation Accuracy (ResNet50)

EfficientNet B2 Performance: EfficientNet B2 started with a relatively high accuracy of approximately 80%, reaching around 91% accuracy by the final epoch. Validation accuracy showed a similar increasing trend, suggesting effective learning during training. Evaluation on unseen data resulted in an accuracy score of approxi-

mately 94%, indicating comparable performance to the customized CNN.

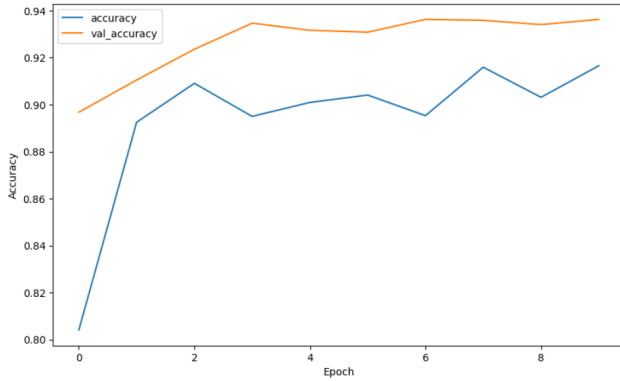


Fig. 11. Training vs Validation Accuracy (EfficientNetB2)

VI. CONCLUSION / FUTURE WORK

In this comparative analysis, we evaluated three different convolutional neural network (CNN) architectures for the task of automated cell classification in malaria diagnosis: a customized CNN, ResNet-50, and EfficientNet B2. Each model demonstrated varying levels of performance, shedding light on their efficacy and suitability for the task.

The customized CNN showcased robust performance, achieving a final accuracy of approximately 94% on unseen data. Despite some fluctuations during training, the model demonstrated consistent improvement and exhibited strong validation accuracy trends, indicating its effectiveness in learning relevant features for cell classification.

ResNet-50, leveraging its pre-trained weights and deep architecture, achieved an accuracy of around 89% on unseen data. While slightly lower than the customized CNN, ResNet-50 demonstrated steady improvement throughout training, suggesting its capability in capturing intricate features from cell images.

EfficientNet B2, known for its efficiency and effectiveness in image processing tasks, exhibited comparable performance to the customized CNN, with an accuracy score of approximately 94%. Its ability to leverage pre-trained features from the ImageNet dataset, combined with custom layers tailored for the task, contributed to its success in malaria cell classification.

Future Work:

Looking ahead, there are a few things we can do to make our malaria detection system even better. One idea is to improve the design of our models. We can try out

new ideas or even combine different models to see if they work better together. Another approach is to use techniques like data augmentation, which means making small changes to our existing data to help our models learn more effectively. We also need to make sure our models can handle different situations by testing them with a wide range of real-world examples.

Another important step is to work closely with doctors and healthcare workers to make sure our system fits into their everyday routines. This means listening to their feedback and making any necessary changes to make our system easy to use and understand. We should also keep testing our system in real-life situations to make sure it's working well and making a positive impact on people's lives. By doing this, we can make sure our system is helping as many people as possible, no matter where they live.

Lastly, we need to keep exploring new ways to take better pictures of cells and prepare the data for our models. This can help our models make even more accurate predictions, leading to better diagnosis and treatment of malaria. By working together and staying committed to improving our system, we can make a real difference in the fight against malaria.

VII. CONTRIBUTIONS

Anjum A Shaik Took charge of the project, developing the main idea and outlining the research goals. Planned the experimental setup, including data collection methods and model architecture. Played a pivotal role in analyzing the results and drawing conclusions. **Venkata Pamarthi** implemented and fine-tuned the deep learning models, focused on optimizing model performance. Experimented with various neural network architectures, tuning hyperparameters, and optimizing algorithms to achieve the best results. **Susmitha Bommini** dealt with handling the dataset, worked on data preprocessing, cleaning, and augmentation. Ensured the data was properly formatted and representative of the problem domain.

REFERENCES

- [1] <https://www.who.int/news-room/fact-sheets/detail/malaria>
- [2] <https://www.cdc.gov/parasites/malaria/index.html>
- [3] Raj M, Sharma R, Sain D. A deep convolutional neural network for detection of malaria parasite in thin blood smear images. In 10th international conference on communication systems and network technologies 2021 (pp. 510-4). IEEE.
- [4] Minarno AE, Aripa L, Azhar Y, Munarko Y. Classification of malaria cell image using inceptionV3 architecture. JOIV: International Journal on Informatics Visualization. 2023; 7(2):273-8.

- [5] Siłka W, Wieczorek M, Siłka J, Woźniak M. Malaria detection using advanced deep learning architecture. *Sensors*. 2023; 23(3):1-21.
- [6] Razin WR, Gunawan TS, Kartiwi M, Yusoff NM. Malaria parasite detection and classification using CNN and YOLOv5 architectures. In 8th international conference on smart instrumentation, measurement and applications 2022 (pp. 277-81). IEEE.
- [7] Krishnadas P, Chadaga K, Sampathila N, Rao S, Prabhu S. Classification of malaria using object detection models. *Informatics*. 2022; 9(4):1-18.
- [8] Sifat MM, Islam MM. A fully automated system to detect malaria parasites and their stages from the blood smear. In region 10 symposium 2020 (pp. 1351- 4). IEEE.
- [9] Sampathila N, Shet N, Basu A. Computational approach for diagnosis of malaria through classification of malaria parasite from microscopic image of blood smear. *Biomedical Research*. 2018; 29(18):3464-8.
- [10] Li D, Ma Z. Residual attention learning network and SVM for malaria parasite detection. *Multimedia Tools and Applications*. 2022; 81(8):10935-60.
- [11] Roy K, Sharmin S, Mukta RM, Sen A. Detection of malaria parasite in Giemsa blood sample using image processing. *International Journal of Computer Science and Information Technology*. 2018; 10(1):55-65.
- [12] Nayak S, Kumar S, Jangid M. Malaria detection using multiple deep learning approaches. In 2019 2nd international conference on intelligent communication and computational techniques 2019 (pp. 292-7). IEEE.
- [13] Maduri PK, Agrawal S, Rai A, Chaubey S. Malaria detection using image processing and machine learning. In 3rd international conference on advances in computing, communication control and networking 2021 (pp. 1789-92). IEEE.
- [14] Francies ML, Ata MM, Mohamed MA. A robust multiclass 3D object recognition based on modern YOLO deep learning algorithms. *Concurrency and Computation: Practice and Experience*. 2022; 34(1):e6517.
- [15] Setyawan D, Wardoyo R, Wibowo ME, Murhandarwati EE, Jamilah J. Malaria classification using convolutional neural network: a review. In sixth international conference on informatics and computing 2021 (pp. 1-9). IEEE.
- [16] Hung J, Carpenter A. Applying faster R-CNN for object detection on malaria images. In proceedings of the IEEE conference on computer vision and pattern recognition workshops 2017 (pp. 56-61).
- [17] Var E, Tek FB. Malaria parasite detection with deep transfer learning. In 3rd international conference on computer science and engineering 2018 (pp. 298-302). IEEE.
- [18] Yang F, Poostchi M, Yu H, Zhou Z, Silamut K, Yu J, et al. Deep learning for smartphone-based malaria parasite detection in thick blood smears. *IEEE Journal of Biomedical and Health Informatics*. 2019; 24(5):1427-38.
- [19] Iradukunda O, Che H, Uwineza J, Bayingana JY, Binimam MS, Niyonzima I. Malaria disease prediction based on machine learning. In international conference on signal, information and data processing 2019 (pp. 1-7). IEEE.
- [20] Prakash SS, Kovoov BC, Visakha K. Convolutional neural network based malaria parasite infection detection using thin microscopic blood smear samples. In second international conference on inventive research in computing applications 2020 (pp. 308-13). IEEE.
- [21] Shekar G, Revathy S, Goud EK. Malaria detection using deep learning. In 4th international conference on trends in electronics and informatics (48184) 2020 (pp. 746-50). IEEE
- [22] Umer M, Sadiq S, Ahmad M, Ullah S, Choi GS, Mehmood A. A novel stacked CNN for malarial parasite detection in thin blood smear images. *IEEE Access*. 2020; 8:93782-92.
- [23] Joshi AM, Das AK, Dhal S. Deep learning based approach for malaria detection in blood cell images. In region 10 conference 2020 (pp. 241-6). IEEE.
- [24] Paul A, Bania RK. Malaria parasite classification using deep convolutional neural network. In international conference on computational intelligence and computing applications 2021 (pp. 1-6). IEEE.
- [25] Aimi SAN, Yusoff M, Zeehaida M. Colour image segmentation approach for detection of malaria parasites using various colour models and k-means clustering. *WSEAS Transactions on Biology and Biomedicine*. 2013; 10(1):41-55.
- [26] 26. Savkare SS, Narote SP. Automatic detection of malaria parasites for estimating parasitemia. *International Journal of Computer Science and Security*. 2011; 5(3):310-5
- [27] <https://ceb.nlm.nih.gov/repositories/malaria-datasets/>
- [28] Shambhu, Shankar Koundal, Deepika Das, Prasenjit. (2023). Deep learning-based computer assisted detection techniques for malaria parasite using blood smear images. 10. 990-1015. 10.19101/IJATEE.2023.10101218.
- [29] Microstructure-informed deep convolutional neural network for predicting short-term creep modulus of cement paste - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Examples-for-max-and-average-pooling-layer_fig5_356816561 [accessed 27 Apr, 2024]
- [30] <https://www.baeldung.com/cs/ml-relu-dropout-layers>
- [31] Kalaycı, T.A.; Asan, U. Improving Classification Performance of Fully Connected Layers by Fuzzy Clustering in Transformed Feature Space. *Symmetry* 2022, 14, 658. <https://doi.org/10.3390/sym14040658>
- [32] Ji, Qingge Huang, Jie He, Wenjie Sun, Yankui. (2019). Optimized Deep Convolutional Neural Networks for Identification of Macular Diseases from Optical Coherence Tomography Images. *Algorithms*. 12. 51. 10.3390/a12030051.
- [33] Transfer Learning-Based Classification Comparison of Stroke - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/EfficientNetB2-modules-description-Agarwal-2020_fig3_364169637 [accessed 27 Apr, 2024]