Article

A Privacy-preserving Approach to Effectively Utilize Distributed Data for Malaria Image Detection

Amer Kareem [*] , Haiming Liu , Vladan Velisavljevic

Posted Date: 5 October 2023

doi: 10.20944/preprints202310.0163.v1

Keywords: Malaria images; machine learning; federated learning; privacy preserving; medical image detection

Article

A Privacy-Preserving Approach to Effectively Utilize Distributed Data for Malaria Image Detection

Amer Kareem [1], Haiming Liu [2] and Vladan Velisavljevic [3]

[1] University of Bedfordshire; amer.kareem@study.beds.ac.uk
[2] University of Southampton; h.liu@soton.ac.uk
[3] University of Bedfordshire; vladan.velisavljevic@beds.ac.uk
* Correspondence: amer.kareem@hotmail.com

Abstract: Malaria is one of the life-threatening disease caused by the Anopheles mosquitoes affecting the human red blood cells. It is the global health concern which is quite common in tropical as well as sub-tropical parts of the World. Therefore, it is an important to have an effective computer aided system in place for early detection and treatment. State of the art machine learning techniques are used to detect the infected malaria cells from the pool of images. As the visual heterogeneity of the malaria dataset is highly complex and dynamic, therefore higher number of images are needed to train the machine learning models effectively. However, hospitals as well as medical institutions do not share the medical image data for collaboration due to general data protection regulation (GDPR) and data protection act (DPA). To overcome this collaborative challenge, our research is inspired to use the real-time medical image data while using the framework of federated learning (FL) framework that will ensure the data privacy while mutual collaboration of hospitals and medical institutes. We have used the state of the art machine learning models that include the Resnet50 and densenet in a federated learning framework. We have experimented both models in different settings and our preliminary results showed that the densenet model performed better in accuracy (75%) in contrast to resnet50 (72%) while considering 8 clients, while the trend is observed common in 4 clients with the similar accuracy of 94% and 6 client showed that the densenet model performed quite well with the accuracy of 92% while resnet50 achieving only 72%.

Keywords: Malaria images; machine learning; federated learning; privacy preserving; medical image detection

1. Introduction

During study of related work, we have observed that, over recent years, advancements in the field of artificial intelligence (AI) have brought a great revolution in the field of medical sciences. It has been demonstrated as an effective way of deployment to detect diseases through CXR, city scan, ultrasound, and other mediums. Researchers are using artificial intelligence for diagnosis and detection. Improving computer vision in AI increases the research interest in medical diagnosis. Regarding medical image detection, AI techniques such as convolution neural networks (CNN) have been used to classify CXR, whether the disease is present or not. While considering AI in the medical field, significant amounts of research has been done that includes abnormal pattern recognition [10,11], biometric detection [12], trauma valuation [13], and diabetes detection [14]. The importance of medical image detection has brought us to work in devising effective methods for detecting diseases including pneumonia, malaria and brain tumor. It can be illustrated in the next section of scope and motivation.

2. Significance of the Research

Considering the potential of proposed research work, the medical imaging industry is in a huge need to alter the procedures for disease detection. The following are some of the important points with regard to the significance of our research.

1. **Improved Security and Compliance**: The proposed research work provides huge potential considering the privacy and security of medical imaging. The research follows the guidelines as per GDPR and DPA for data security that will be a great revolution in the medical industry.
2. **Enhanced diagnostic capacity:** The research framework that constitutes the hybrid model that will ensure the security of data and the accuracy efficiency of disease detection that will ultimately result in early diagnosis and treatment.
3. **Facilitating Collaboration**: Research will promote an innovative culture that allows the mutual collaboration of hospitals and medical institutes to be joined to achieve the improved advancements.
4. **Benchmark for Potential Innovation**: Based on the study analysis, it will guide the future scope of innovation in medical imaging for researchers. The proposed research can be a benchmark for the future development of the idea that can be mutually beneficial for medical institutes.
5. **Scalable and Flexible Framework**: The research illustrates the use of a CNN-based pre-trained model on the federated learning framework that is highly scalable towards the multiple of type of medical images. Provides a robust and enhanced solution for medical image detection.
6. **Economic Influence**: Research will bring important changes to ease the economic impact such as early detection. It will ultimately bring about early disease detection saving cost and resources in the medical industry.
7. **Global extent and convenience:** Using the FL framework as in the proposed research will ensure data privacy, allowing data from diverse sources to enhance machine learning models' learning capability.

It can be seen from the significance of the proposed research work that the industrial implications are massive. It reflects the current situation in the medical field and the future scope that allows medical institutes to collaborate in a single platform for improved diagnoses and early treatment. The research will also bring academic institutes together to develop an innovative solution to medical image detection by collaborating with the image data of the medical centre in a privacy-preserving manner.

3. Literature Review

This section demonstrates the different state of the art techniques that are used for medical image detection. We have critically analysed the previous work and highlighted the limitations as well as reason for the selection of CNN-based models.

Researchers have used the CNN pre-trained model of Resnet-50 for the detection of the Diabetic Retinopathy which is the major cause of blindness in the diabetic patients [1]. The researcher's have followed the optimal steps for image pre-pre-processing and augmentation. In the experiment, dataset of 3762 images have been used, among them 1855 was healthy ones and 1907 are the infected from the Eyepacs. The authors have compared the work with the other state-of-the-art literature and find out that the performance of the ResNet50 is effective when the image pre-processing is enhanced as the model performance can vary depending on the input images. In the machine learning context with regards to the image detection, the author has demonstrated the outcome of the experiments while achieving the accuracy of 0.9802 in the binary classification. It gives the clear view of using the dataset having better augmentation and data cleaning to let the model perform effectively.

Another experiment was performed while comparing the model of ResNet50 and VGG16 for the detection of the Covid-19 [2]. In the experiment the dataset of "COVID-19 Radiography Database" was used that was obtained from the Kaggle. The dataset constitutes of the images over 10k, among them 3600 were infected. After performing the data cleaning and data pre-processing, the authors have individually used the resnet50 and VGG16 model. The results of the experiments have demonstrated that the performance of resnet50 stands out in contrast to the VGG16. The achieved accuracy on the experiment for restnet50 was 88% while on the other hand, the achieved accuracy for the VGG16 was 85% for the detection of the covid-19. In the same experiment, the precision achieved is 100% for restnet50 while 84% for the VGG network. The authors have suggested that the overall accuracy of the model performance can be enhanced while using the hyperparameter tunning.

A research was conducted while using the restnet50 model on the covid 19 dataset [3]. The CT scan images were used for the experiment. The dataset constituting of over 5K CT scan images were utilised. The researchers have realised that the quite larger number of datasets are similar to the pre-training dataset, therefore the CNN based models can be less effective. To cope up with this challenge, researchers have urged to perform fine tuning of the model on the training dataset to enhance the performance of the model and it also reduces the time consumption of the model training. While using the same optimiser, the authors have achieved the accuracy of 88% on the normal CNN model with the default tuning, however when the authors have used the similar optimiser on the tuned CNN model, the accuracy went up by 6% that is 94%. The experiment performed by the authors have clearly distinguish the use of different hyperparameter tuning to effectively increase the model accuracy for the disease prediction and also improve the time consumption. As per, our proposed model for using the real-time data where the amount of data and time required to train the model will be challenging, it is important to define the model into the state of fine tuning to get the best possible outcome.

The researchers have proposed the use of enhanced method for tuberculosis (TB) detection from the CXR while using the densenet model [4]. The experiment was performed while using the wider framework of the densenet (WDnet) which was based on the Convolutional Block Attention Module (CBAM). In the experiment, the dataset was used from the multiple repositories and combined together. The combination of the repositories have produced the images upto 5k. Among the image dataset, 1094 was classed as infected images while the remaining were classed as normal. While comparing the model with other literature, the researchers have demonstrated the evaluation of the experiments by producing the accuracy of 98.80%. This research has highlighted the use of different epochs to understand the best possible outcome of the experiment for medical image detection.

The study was conducted that have highlighted the importance of using the deep learning approach for the effective medical image detection [5]. In the literature research, authors have analysed the different machine learning models for the medical image classification and used the benchmark CT scan dataset to compare the results. The authors have raised the concern of challenges in using the deep learning model for training the dataset. Also, it has been mentioned in the research that the new dataset for the pre-trained model could be complex, therefore it is important, we understand the lacking and use the effective approach to get the least false positive or negative results. This research gives us the understanding of using the deep learning model for larger amount of dataset to understand the maximum patterns in the data especially in the medical images. In our proposed solution of using the real time data in the federated learning framework gives the clear way forward to utilise the constant stream of data from different hospitals and medical institutions, that would eventually help to make deep learning models more capable of recognising the different diseases and effective classification accuracy.

The experiment was conducted to analyse the performance of the neural network in the detection of Sars-Cov2 virus [6]. The research was done while using the CNN model of VGG-16, VGG-19, ResNet 50, Inception v3, DenseNet, XceptionNet, and MobileNet v2. The dataset used in the experiment consists of 1252 Covid and 1229 Non-Covid CT scan images. While comparing the different CNN models, the authors have defined the use of proposed CNN model that has produced the accuracy of 92%. The main idea of the research was to identify the different model performance, do the alteration on the CNN pre-trained model and produce the customised CNN approach for effective classification. The results of the experiment have been demonstrated in the paper that gives the clear picture of the use of customised CNN model. Although, the customised CNN approach has produced in the classification of the Sars-Cov2 virus, however the proposed model of the authors could be challenging when followed up with the higher number of images. The current experiment demonstrates the limited number of images. While considering our proposed method of using the real-time data in the federated learning framework, it posses limitation, the idea to use the model that is pre-trained and scalable. The author's given experiment is effective in the limited number of dataset, however higher number can adversely impact on the performance of the model.

In an another experiment [7], the researchers have performed experiments for the detection of the pneumonia disease while using the resnet. In the experiment, authors have used the different version of the resnet to compare the results. The researchers have used the optimised attention mechanism to enhance the performance of the model that includes better extract of channel and spatial feature from the features map. The publicaly available dataset was used for the experiment that comprises of 5800 images among which 3875 was infected with pneumonia while others was normal images i.e., 1341. The researchers have used the Convolutional Block Attention Module (CBAM) with the resent models that has resulted in the overall effectiveness of disease detection. The epoch of 30 was used in the experiment. While using the attention mechanism, resent50, resnet101 and resnet152 have produced the accuracy of 90%, 92% and 94% respectively while taken by each epoch is 22s, 21s and 28s. The result demonstrates that, although the performance of the resnet152 with the attention mechanism performs well in terms of accuracy, however time consumption on the other hand has been increased, while resent50 has slightly less accuracy in contrast, the time consumption on each epoch is quite less. This experiment gives us the understanding of using the model that is capable of producing the effective classification on the detection of the disease and least possible time of each epoch. While considering the larger dataset as in case of real-time, the time element plays an essential role as the higher training time can adversely impact the overall architecture.

Another interesting experiment was performed that has demonstrated the performance of aggregated deep learning models that includes ResNet-34, VGG-19, DenseNet 121, and DenseNet 161 [8]. The experiment was conducted on the detection of the Knee Osteoarthritis. The dataset used in the experiment has been available publicly on Kaggle that constitute of 9786 images in total which is classed into 4 different grades. The experiment was conducted individually on the models as well as the with the ensemble approach. The proposed ensemble approach has produced the accuracy of 98%. The authors have used the fine-tuned model to ensembles and evaluate the results. Although, authors have used the effective approach of ensemble for the detection of the Knee Osteoarthritis. The limitation involves the model overfitting issues where the different models have their own capabilities to input and process the data. It is important to ensure that model performance is maintained by feeding with the different quantity and variety of the dataset as in case of real-time stream of data.

The research [9] was carried out using block chain technology in a federated learning environment. In the proposed study, a novel approach of weight modification was used to train local models from different data sources. The concept of federated learning falls under single point failure; therefore, to cope up with this challenge researchers have used the blockchain in collaboration that involves the training of the dataset from the different sources based on nodes and ledgers. As utilising the blockchain is immutable, the history of all events is preserved. In the multidisease classification, the researchers have achieved an accuracy of 88.10%. The experiment has demonstrated the use of federated learning in conjunction with blockchain technology for medical image classification; however, the use of blockchain ledger in the federated learning architecture slows the process of training the model and aggregating at the central server. Therefore, in real time, to effectively use the federated learning architecture, the collaboration of blockchain technology possesses limitations. Our proposed study is based on utilising the solely federated learning architecture and adjusting hyperparameters as well as adding optimiser to effective medical image classification.

4. Methodology

4.1. Research Model Design

The conventional system adopted by medical institutions and hospitals abides by data protection and privacy laws. This are why data is not shared with other institutions or any third party due to GDPR. Therefore, in our FL framework, data privacy is ensured while model training is performed in multiple medical institutes and hospitals without sharing data. In medical image detection, this approach is effective as it helps train the ML model from various medical centres; in this way, a significant amount of heterogeneous real-time data is used, which ultimately helps the ML model for effective training. The framework of using FL with the CNN-based pre-trained models is selected based on previous work on medical image detection. As mentioned in the literature review, the performance of the ML model is enhanced if the model is trained with a larger amount of data. In other words, fewer data produces ineffective performance in ML model training and vice versa. Therefore, a collaborative way of using the dataset's multiple sources is required, which can be possible by mutual collaboration. In this way, the model learns the patterns of the image data and helps to form an effective model for detecting diseases from pool of dataset. The image below illustrates our research model design:

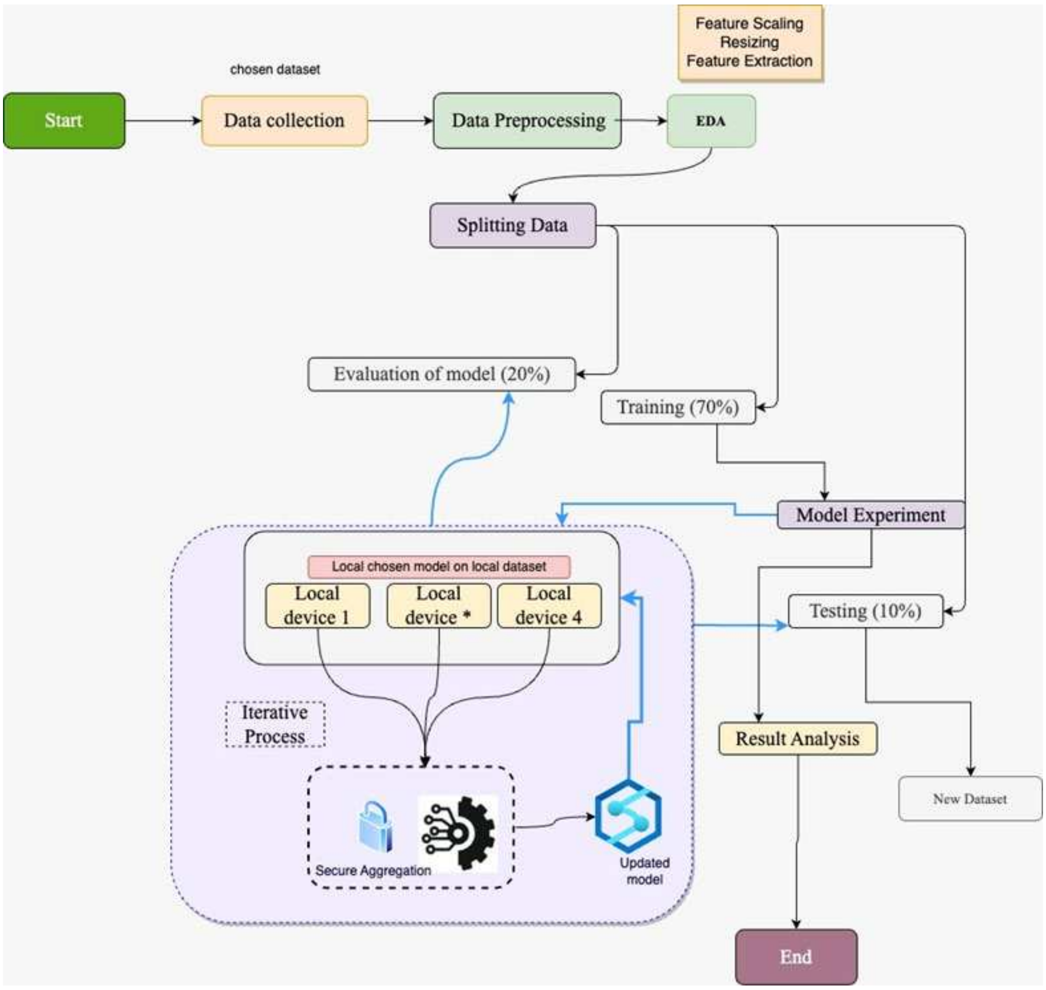


Figure 1. Proposed Framework.

As elaborated in the figure above, our research workflow follows the data collection. The collected dataset was pre-processed by resizing and removing any irregularities. Once the data are pre-processed, it is split into training, validation, and testing with a ratio of 70%, 20%, and 10%, respectively. The training dataset is used for applying the machine learning model after processing which is sent across the participants for model training in the above-mentioned federated learning framework. Once the model is locally trained on the local devices, the individual trained model is shared with the FL server (central server), forming an updated model after aggregation. Secure aggregation is in place in this model design to ensure the security of the trained model. The performance of the model is evaluated with the validation and testing data set. Training the ML model on the individual devices and the FL server is an iterative process to get more updates and correspondence from the dataset on the client side. As our research framework is based on FL, let us explore the fundamental concepts behind FL.

We have used FL architecture in our research to demonstrate the privacy-preserving approach that allows for mutual collaboration between hospitals and medical institutions. A standard approach is required to agree by participating parties that includes the model framework, loss, and activation functions. In general, the FL architecture can be illustrated as follows:

$$\min_{w\in\mathbb{R}^d} \ell(x,y;w) \text{ where } \ell(x,y;w) \overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} \ell(x_i,y_i;w) \tag{1}$$

To understand the architecture of FL, let us take hospitals 'C' comprising the dataset 'Di = nc', 'n' shows the quantity of data, and the FL architecture in the individual hospital can be illustrated as follows:

$$\ell(x,y;w) = \text{ where } L_c(x_c,y_c;w) = \frac{1}{n_c}\sum_{i\in D_i} \ell(x_i^c,y_i^c;w) \tag{2}$$

The FL global server inputs the model that consists of different parameters for medical image detection. During the individual cycle with the global server, random participants are considered. Generate the one-to-one link with the server. The local participant downloads the model from the local server that calculates the average gradient based on the loss 'fc' which can be demonstrated based on the weight 'wt' that constitutes the learning rate 'η'. In this way, local participant keeps updating and at the same time keeps updating the central server. The aggregated model receiving the updates from the participants can be calculated as follows:

$$w_{t+1} \leftarrow w_t - \eta\nabla\ell(x,y;w)$$

$$w_{t+1} \leftarrow w_t - \eta\sum_{c=1}^{C}\frac{n_c}{n}\nabla L_c(x_c,y_c;w) \tag{3}$$

$$w_{t+1} \leftarrow w_t - \eta\frac{n_c}{n}f_c \tag{4}$$

For every hospital c, $w_{t+1}^c \leftarrow w_t - \eta f_c$, then

$$w_{t+1} \leftarrow w_t - \sum_{c=1}^{C}\frac{n_c}{n}w_{t+1}^c \tag{5}$$

The various sizes of the data set at each round help to improve the model leaning capability in terms of detection w_{t+1}^c which can be quite useful for the skewed data set and can be illustrated in the following equation:

$$w_{t+1} \leftarrow w_t - \sum_{c=1}^{C}\frac{n_c}{n}\alpha_{t+1}^c w_{t+1}^c \tag{6}$$

The concept of increasing the weight parameter can result in better performance. The averaging of the weight from the distinct client tends to improve the learning capabilities of the model for image detection.

4.2. Configuration of Models in Federated Learning

In the configuration setting of our models in federated learning framework, we have used 'return_df()' function that is capable of processing the '.png' images file directories based on distinct datasets and spread equally across the participating clients, i.e., client 4,6 and 8. The individual file

'.png' keeps track of the file location. The return function is capable of doing an equal distribution of available data labels to the clients. For example, if we have 5 clients and 100 images, then the return function will ensure the distribution of 20 images to individual clients.

In our setting, we have also used custom data loader, which is efficient as it avoids loading the full dataset in RAM instead it loads on demand basis. This is an effective approach, especially on the larger datasets. While using the data loading function, i.e., 'return_dataloader()', the DataLoader object is outputted as below:

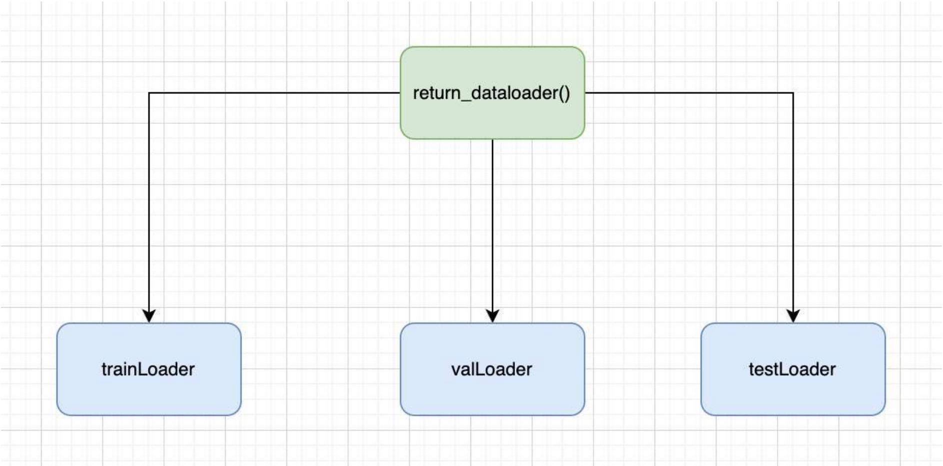


Figure 2. Data loader function.

1. **TrainLoader**: It is the type of DataLoader that determines the training section of the dataset. It is quite useful in iterating on the transformed images as well as in iterating on the labels over the batches. Variability is also ensured, as it is involved in the reshuffling of the training data prior to every epoch.
2. **valLoader**: It shows the validation section of the dataset. Data iteration takes place over batches on validation images, as well as labels. It is not involved in the data reshuffling; therefore, the order stays intact throughout the epochs.
3. **testLoader**: It is the type DataLoader that demonstrates the test part of the data set. The data iteration takes place over the batched over the test images as well as labels. Unlike TrainLoader, it is not involved in data reshuffling.

DataLoaders are depending on the percentage allocated for the training, validation, and the testing data. In general, DataLoader is quite efficient in memory consumption and can be effectively used in the training validation and testing stages of ML models. It eventually helps to make the data ready for PyTorch-based models.

We have also used the 'train_model()' function, which is involved in the following stages:

1. **Initialise the variable and list**: The initial step involved in the tracking of the model state, validation loss, and the accuracy for the individual epoch.
2. **Epoch loops**: The train model function also loops over a certain number of epochs.
3. **Phase loops**: The function also performs the loops over the training as well as the validation stage of individual epochs.
4. **Set Models Mode:** The function also keeps the mode to ‘train’ if the model is in training mode. In this way certain features including dropout as well as batch normalisation are activated. Similarly, during the validation stage, these parameters are disabled.
5. **Batch loops**: The function also loops with the data batches.
6. **Forward pass**: The function transfers the input label to the corresponding device via model and performs loss calculation.

7. **Backward pass and optimisation:** When using the train model function, if the training is zero, then the gradient of the model is calculated by using the optimisers also known as backward pass.
8. **Statistics calculation**: In the train_model function, the prediction is calculated, and the model run loss as well as accuracy is updated.
9. **Epoch Statistic Calculation:** During the training phase, by the end of every epoch, the loss and accuracy are calculated by the end of every epoch. While in the vase of validation stage, the precision, accuracy, F1-score is calculated while showing the confusion matrix.

Several numbers of iterations are performed by using the while loop. The 'while loop' in general is involved in higher number of continuous iterations, however, in our case, it runs at one time based on the rounds:

- In our case, we have used clients 4, 6 and 8 to train the model individually on the clients while using the 'train_model' function, and in this way, model accuracy and loss are given out.
- The weights from the individual clients are kept in 'w_local' lists, similarly, the accuracy and loss of the model are kept on their respective lists.
- In the federated learning framework, the 'fed_avg' function is used to take the average weight of all models to form the global model.
- The mean loss and accuracy is also calculated on the participants.
- The mean average of all the weights is referred back towards the model on individual devices. It allows every client to receive the similar model update.
- The weight average of the models is stored as the 'fed_model_client'.
- The output is displayed with the round phase, loss, as well as the accuracy.
- Alterations in 'validation_loss' as well as accuracy from the last round are displayed, too.

In this configuration setup of federated learning, we have demonstrated that ML models are locally trained on participating clients, the weight of individual clients is aggregated to form a global model which is shared acres the clients. In this approach, data stays local to client, and only the weight average model is shared to central server. In these data, data privacy is retained.

10. **Return best model**: Once all corresponding epoch rounds are accomplished, the one with the least validation loss loads up the model.

At every epoch, initially the model is trained with the training dataset, and once the training is done, the model performance is observed with the validation data. The confusion matrix including the accuracy and loss function is demonstrated at each stage to understand the performance of the model. Once all the epochs are completed, the function determines the epoch with the least validation loss, which ultimately selects the corresponding model state. It is a quite useful method that helps us to understand the state of the model in best terms of its performance.

4.3. Data Gathering

The mobile application was designed by lister hill national center for biomedical communication knows as "malaria screener" which was designed specifically for the individual who works with microscope not having massive resources. The mobile application was linked to the microscope, where the camera was used to capture pictures. The malaria pictures were marked by professionals from the Mahidol Oxford Tropical Research Institute, which is based in Bangkok, Thailand. The data set comprises 27,560 images which contain both infected and unaffected [15]. The data set was gathered from the different locations based on the types of malaria as follows:

1. Malaria falciparum blood samples were taken from 150 patients at the Chittaging Medical College Hospital, Bangladesh.
2. The vivax malaria samples were taken from the same location as above from 150 patients and also 50 from healthy individuals.
3. Malaria samples from falciparum were also taken from a similar location in Bangladesh: 148 patients and 45 healthy individuals.

4. Vibrax malaria samples were also taken from Bangkok, Thailand, 171 patients.
5. In addition, blood cell samples of falciparum malaria were collected from Bangladesh, i.e., 150 patients and other 50 healthy individuals.

Why this data set. We have used the malaria dataset for several reasons, some of the main reasons are as follows:

1. **Real-time data, non-synthetic:** In the experiments, synthetic data can be quite useful for training the ML models, however, the lab-based model has a limitation to real-time complexity and variance of the disease. The data collected from the real-time provides stronger robust and efficient data for training the models. As in our experiments, we are targeting the real-time data; therefore, the given dataset is collected from the patients in real time, which helps to model generalisation.
2. **Geographical Relevancy**: Another significant impact of using this data set involves it's correspondence with the geographic relevance where malaria is the serious health issue, i.e., Bangladesh and Thailand. The geographic location of the data helps to make the effective model prediction based on the provided data. The model customisation facility based on geographic location can help achieve higher performance. Similarly, it assists in nonspecific regions for malaria detection as well.
3. **Reliable dataset**: Another important aspect of using these data is the reliability, as the data have been collected from the endorsed hospitals that maintain the standards. Therefore, higher reliability is essential for better model performance.
4. **Diverse Images**: The diverse malaria image collection helps to understand the variations of the causing parasites, which ultimately helps to train the ML model effectively.
5. **Significant Global Impact**: Malaria is one of the serious diseases that affect millions of individuals each year and has major health consequences. Therefore, the reflexion of malaria disease constitutes the main global consideration.

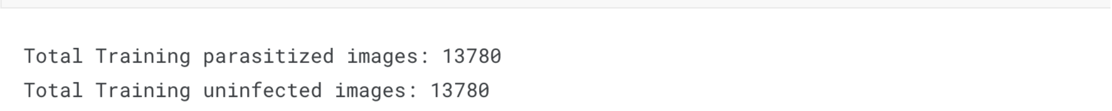
The selection of the malaria data set fulfils our goal of diagnosing the disease that has a significant impact globally. The work done in the field will help to improve public health and avoid the maximum serious consequences of the disease.

4.4. Exploratory Data Analysis (EDA)

To perform the EDA, we have randomly selected the training dataset to visualise the data diversity. We have taken 15 images individually from the training data set and the validation data set with their own labels. The visualisation of the training data set has helped us to understand the data while observing it's performance during the validation stage. The EDA on the malaria dataset is quite important as it helped to ensure the correct labels on the images, as well as it also aids in understanding the data variation and complexities, so the models can be adjusted according while training.

4.4.1. Number of Available Data Sets

The dataset consititue of following distribution which is class balanced as well:



```
Total Training parasitized images: 13780
Total Training uninfected images: 13780
```

The individual data set that is available on the training and validation dataset is 13,780. The count of dataset based on the infected and unaffected can be visualised as follows:

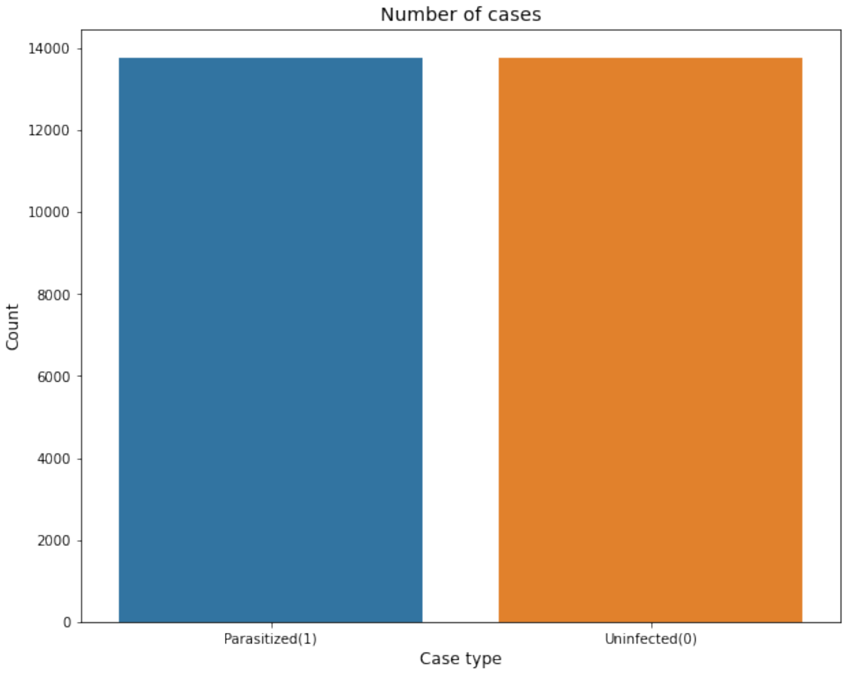


Figure 3. Malaria dataset Distribution.

The total count of 27,560 images is available when combining the training and validation datasets. The large number of images helps our models to perform an effective training from the variety of images. The balance of the data set helps to achieve better model training that is essential for correct disease prediction.

4.4.2. How Parasitised and Nonparasitised Cells Look Like

When observing the malaria dataset, the main element that makes the difference between the two classes is the 'dots' that is visible in the parasitised (infected) class. The dots are actually the malaria parasite. The parasite that causes malaria is known as the plasmodium that directly targets red blood cells of the body. Under the microscope, the infected part of the cell is seen as small dots.

In our dataset, we see the ‘ting dots’ as parasite in the cells, which is one of the visual side of detecting malaria. The tiny dots also make a good difference between the healthy cells in contrast to those affected by the malaria parasite. Healthy cells lack these dots that make up non-malaria cells. The difference between healthy cells and malaria cells can be identified in the following visualisation.

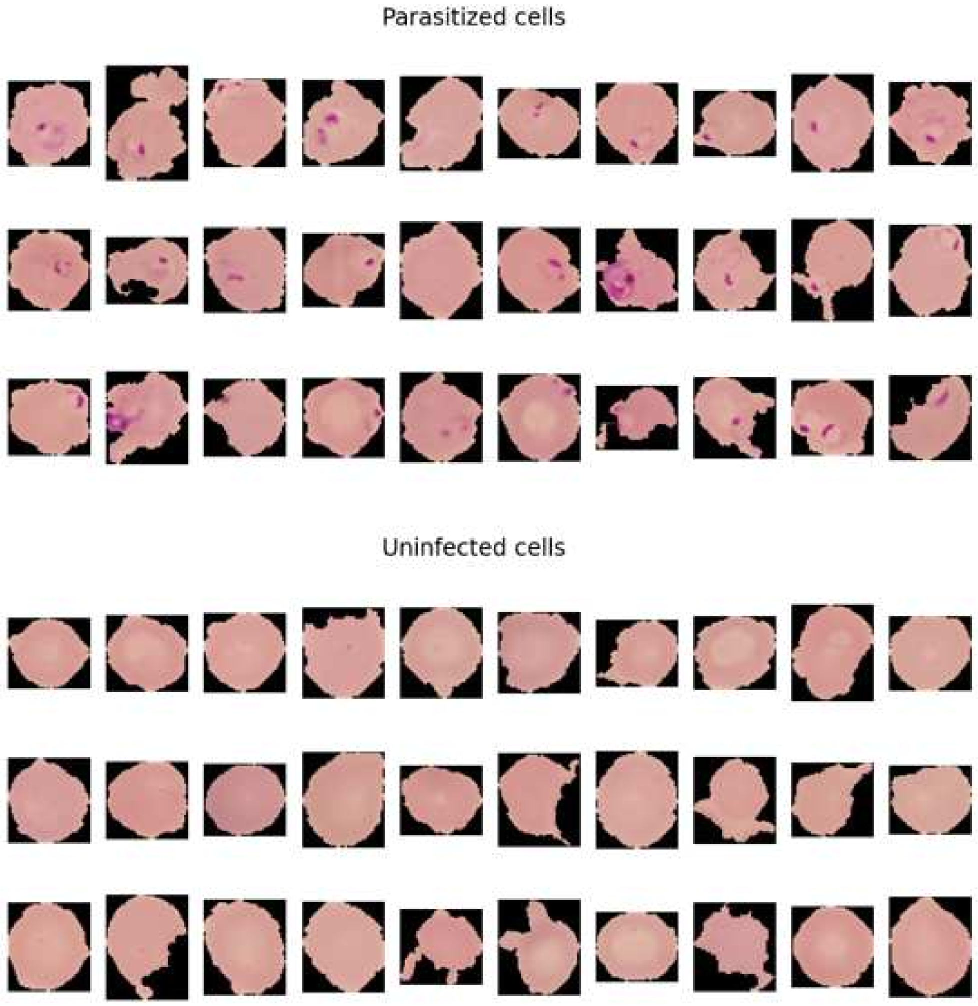


Figure 4. The distinguish between the cell among the two classes help models, i.e., resent50 and densenet, to learn the patterns from the cells. Therefore, the ML models classify cells with the presence of dots as parasitised and those without dots as uninfected.

5. Data Preparation

Data preparation is an initial step to get the data ready for machine learning modelling. Therefore, the images must be changed in the form and shape that the computer could understand. The following are the steps involved in our data preparation.

- The initial stage involves reading the images from the directory.
- Decoding of the image content that involves converting into grid form as RGB.
- Conversion of images into float point tensor.
- Rescaling the tensors into the form that allows the scale range from 0 and 255 to be 0s and 1s as the CNN models takes in the smaller inputs.

We have performed the above steps using the tensorflow tool known as the ImgeDataGenerator. It helps to form the images in the required form for CNN models. The sampling of images also takes place with a similar tool. Afterwords, we have used the technique known as the flow-from-directory, which helps to input the images, adjusting the value, and also rescale the size of the images.

5.1. Visualize the Training Images

We have taken the 15 random images from the training dataset as well as from the validation dataset which can be visualised as follows with respect to their labels:

Figure 5. 15 random images of the sample training images.

Figure 6. 15 images from the sample validation images.

6. Malaria Experiments and Results

6.1. FL_Densenet and FL_Resnet50 (4 Clients)

The figure below illustrates the confusion matrix for densenet and resnet50 with 4 clients:

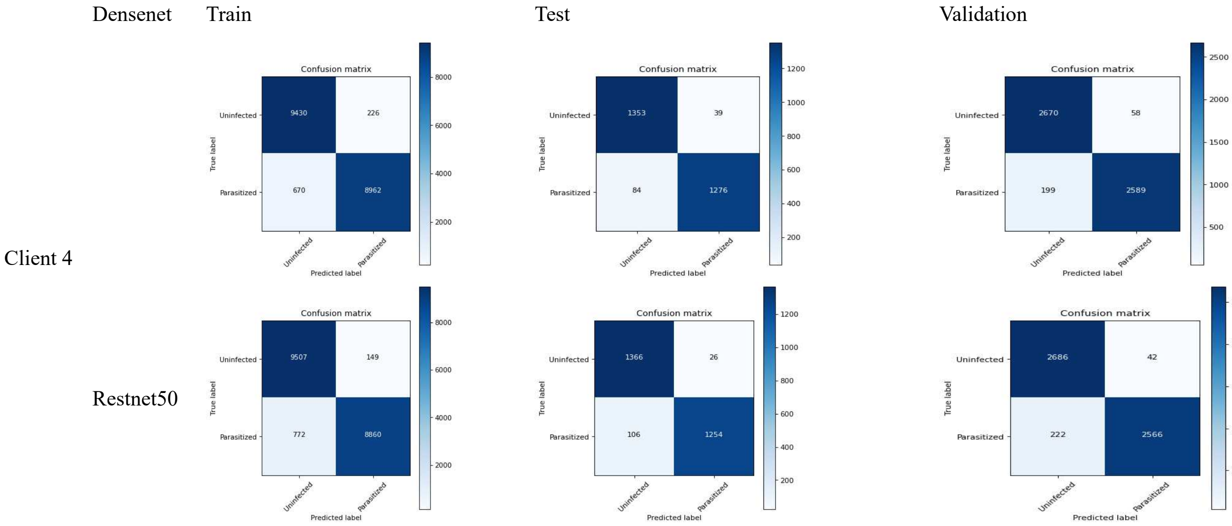


Figure 7. the confusion matrix for densenet and resnet50 with 4 clients.

In terms of performance of model towards the unseen data, let us consider the test data performance analysis with critical discussion:

1. **Accuracy**: The overall correct malaria predictions out of all predictions, the accuracy can be calculated as follows:

- DenseNet: (1276 + 1353)/(1276 + 1353 + 39 + 84) = 0.9463 or 94.63%
- ResNet50: (1254 + 1366)/(1254 + 1366 + 26 + 106) = 0.9486 or 94.86%

ResNet50 has slightly higher accuracy than DenseNet.

2. **Precision**: It shows the accuracy of the positive prediction of malaria and can be observed in the following equation:

- DenseNet: 1276/(1276 + 39) = 0.9703 or 97.03%
- ResNet50: 1254/(1254 + 26) = 0.9796 or 97.96%

ResNet50 has slightly higher precision.

3. **Recall (sensitivity):** It involves the fraction of positive predictions which is correctly determined:

- DenseNet: 1276/(1276 + 84) = 0.9382 or 93.82%
- ResNet50: 1254/(1254 + 106) = 0.9220 or 92.20%

DenseNet has slightly higher recall.

4. **Specificity**: It involves the fractions of negative prediction that are correctly determined:

- DenseNet: 1353/(1353 + 39) = 0.9720 or 97.20%
- ResNet50: 1366/(1366 + 26) = 0.9813 or 98.13%

ResNet has higher specificity.

5. **F1 Score**: The weighted average of precision and recall of both models can be calculated as follows:

- DenseNet: 2 * (0.9703 * 0.9382)/(0.9703 + 0.9382) = 0.9540 or 95.40%
- ResNet50: 2 * (0.9796 * 0.9220)/(0.9796 + 0.9220) = 0.9501 or 95.01%

DenseNet has a slightly higher F1 score.

It can be observed from the above analysis that Densenet has higher recall as well as F1-score, however, on the other hand esnet50 stands out with accuracy, precision, and specificity.

To determine the selection of model based on the above results, it depends on the use case in a given situation. In the case, if we need to decrease false negative values, for example classify if someone doesn't have malaria, but actually has it, the selection of densenet is preferred, as it has a higher recall rate. In the other case, if we need to reduce false positive values, in other words, if the model tells someone to have malaria, however in reality they do not, then resnet50 could be a better option due to better precision and specificity. Resnet50 is also leads slightly in accuracy.

It is worthwhile to understand the importance of model selection based on the metric; however, the selection mostly relies on the use case scenario. In real life, it is good to have a model with few false positive values (including some further testing) instead of missing true positive results that can neglect the disease treatment. Therefore, in this case the model with higher recall rate should be prioritised.

6.2. FL_Densenet and FL_Resnet50 (6 Clients)

The following confusion matrix is achieved by using 6 clients in the malaria dataset:

1. **Accuracy**: (TP+TN)/(TP+FP+FN+TN)

- Densenet: (688+1272)/(688+0+168+1272) = 0.9213 (92.13%)
- Restnet50: (666+1269)/(666+3+703+1269) = 0.7250 (72.50%)

2. **Precision**: TP/(TP+FP)

- Densenet: 688/(688+0) = 1 (100%)
- Restnet50: 666/(666+3) = 0.9955 (99.55%)

3. **Recall**: TP/(TP+FN)

- Densenet: 688/(688+168) = 0.8037 (80.37%)
- Restnet50: 666/(666+703) = 0.4864 (48.64%)

4. **F1-score**: 2*(precision*recall)/(precision+recall)

- Densenet: 2*(1*0.8037)/(1+0.8037) = 0.8911 (89.11%)
- Restnet50: 2*(0.9955*0.4864)/(0.9955+0.4864) = 0.6530 (65.30%)

5. **Specificity**: TN/(TN+FP)

- Densenet: 1272/(1272+0) = 1 (100%)
- Restnet50: 1269/(1269+3) = 0.9976 (99.76%)

The above results shows the better performance of densenet in contrast to resnet50 while comparing accuracy, recall, f1-score and specificity. Resnet50 also constitutes less precision, while the the the difference is quite minor. The overall comparison shows that there are less false negative and non false positive in contrast. The detailed analysis shows that the densenet model stands out in terms of malaria classification; however, it is also important to understand the certain requirement of the use case. In the case where higher false positives are required, the use of densenet could be a good choice, as it constitutes higher precision. Overall, densenet performed well in the malaria case study.

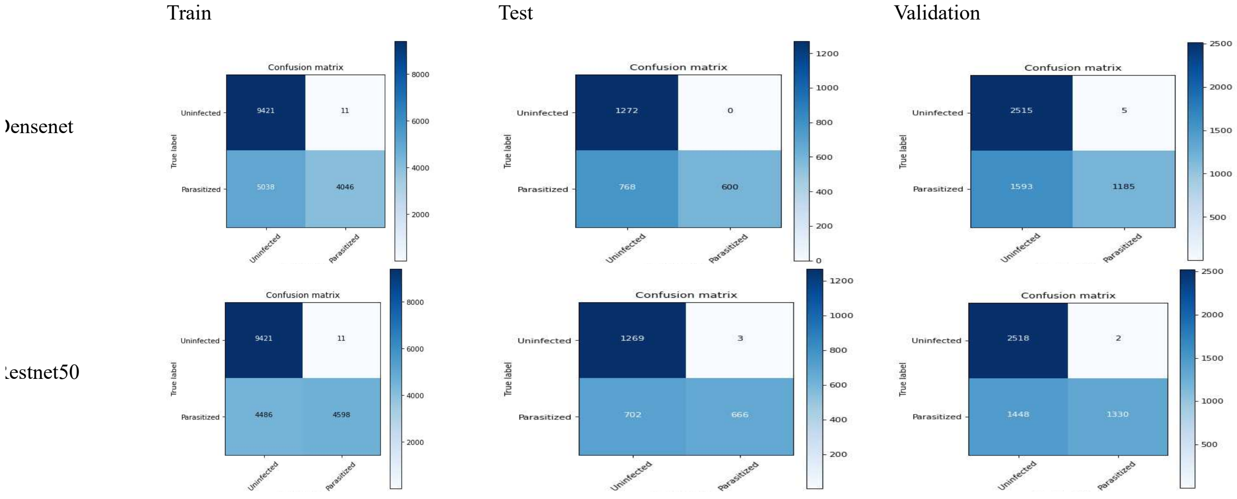


Figure 8. confusion matrix of FL_Densenet and FL_Resnet50 (6 Clients).

6.3. FL_Densenet and FL_Resnet50 (8 Clients)

We have evaluated the models performance based on the following matrix:

1. **Accuracy:** (TP+TN)/(TP+FP+FN+TN)

- Densenet: (1308+712)/(1308+552+68+712) = 0.7504 (75.04%)
- Restnet50: (666+1269)/(666+3+703+1269) = 0.7250 (72.50%)

The accuracy result shows that the densenet performs well in accuracy as compared to densenet demonstrating the fewer errors. On the other case, the only accuracy is not enough to justify the model performance in the case where the rate of false positive is massively different from that of false negative.

2. **Precision:** TP/(TP+FP)

- Densenet: 1308/(1308+552) = 0.7033 (70.33%)
- Restnet50: 666/(666+3) = 0.9955 (99.55%)

In the consideration where the cost of false positive is higher, which means the prediction of malaria disease while it's reality is not, then resent50 is the best choice.

3. **Recall** (sensitivity): TP/(TP+FN)

- Densenet: 1308/(1308+68) = 0.9506 (95.06%)
- Restnet50: 666/(666+703) = 0.4864 (48.64%)

In the case where it is necessary to determine the actual malaria case while risking the false alarm, the densenet model is effective due to the higher recall value. The use of a recall matrix could be ideal in the health care sector where disease identification is crucial.

4. **F1-Score**: 2*(Precision*Recall)/(Precision+Recall)

- Densenet: 2*(0.7033*0.9506)/(0.7033+0.9506) = 0.8087 (80.87%)
- Restnet50: 2*(0.9955*0.4864)/(0.9955+0.4864) = 0.6530 (65.30%)

Determines the relationship between precision and recall values. In other words, f1-score plays an essential role where the false positive as well as the false negative are of equal importance. In our case, the densenet model is the best selection for f1-score.

5. **Specificity** (True Negative Rate): TN/(TN+FP)

- Densenet: 712/(712+552) = 0.5635 (56.35%)
- Restnet50: 1269/(1269+3) = 0.9976 (99.76%)

Specificity is quite useful in the cases where it is required to determine the negative cases that do not constitute malaria. In our case, resnet50 performs a higher specificity score, which is useful in the situation where it requires one to avoid unnecessary treatment.

6.4. Significance Test

The t-test suggests whether the difference between the methods really makes the difference based on the critical differences or it is just a coincidence. Therefore, we have used our CNN models along with FL to conduct pairwise tests. The p-value reflects whether the difference between the models is larger, in case if the values go lower than 5%, it will result in rejection. The p-test conducted on the CNN models individually on the FL framework reflects the performance based on each model. In other words, t-test shows the mean difference of the chosen parameter between the two models. The positive t-values indicates whether the 1st group have any difference with the 2nd group and negative values reflects that the 2nd group has major differences with the 1st one. Similarly, the p-value, as in our case is standard scale of 0.05, so if the results are below this value it indicates the significant difference between the models. The following table shows the t-stats and p-values of our results:

Table 1. T-stats and p-values of different models on malaria dataset.

Metrics	Model Name	T-Stats	P-Value
Accuracy	FL_DENSENET And FL RESNET50	11.45726	0
Precision	FL_DENSENET And FL RESNET50	−0.09118	0.92746
Re-call	FL_DENSENET And FL RESNET50	0.09637	0.92335
F1-Score	FL_DENSENET And FL RESNET50	0.25665	0.79778

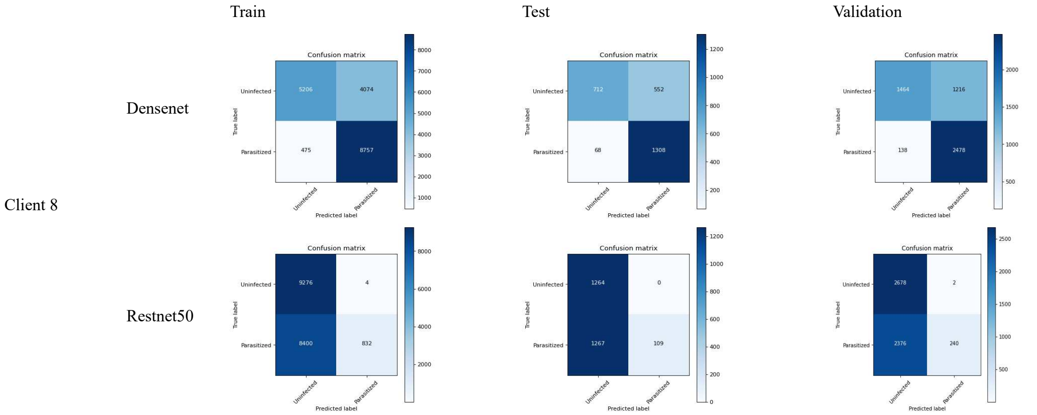


Figure 9. Confusion matrix of FL_Densenet and FL_Resnet50 (8 Clients).

7. Summary

While observing the above results, it can be seen that the densenet model performs better in terms of accuracy, recall, and f1-score while on the other hand, resnet50 do well in precision and specificity. Based on the models, the following situations can be taken into account:

- In the case where it is necessary to determine the actual malaria disease as much as possible, densenet is preferred due to higher recall.
- In the case where it is required to determine the actual malaria disease, and in reality, it is malaria, then resnet50 is preferred due to it's higher precision.
- In the case where it is required to determine the balancing among the false positive as well as false negative, then densenet performs well due to higher recall.
- In the case where it is required to correctly determine the higher negative cases, that is, no-malaria, resnet50 performs well due to it's effective specificity results.

In general, densenet might be the best choice if the accuracy of predicting the malaria cases is crucial, however, the resnet50 model avoids false alarm and also helps in right detection of nonmalaria cases.

References

1. Ali, S.; Raut, S. Detection of Diabetic Retinopathy from fundus images using Resnet50," 2023 2nd International Conference on Paradigm Shifts in Communications Embedded Systems, Machine Learning and Signal Processing (PCEMS), Nagpur, India, 2023, pp. 1–5. https://doi.org/10.1109/PCEMS58491.2023.10136073.
2. S. S. M; Rao, M.D.S.; Rani, M.S.; Durga, K.; Kranthi, A. Covid-19 X-Ray Image Detection using ResNet50 and VGG16 in Convolution Neural Network," 2022 IEEE Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI), Gwalior, India, 2022; pp. 1–5. https://doi.org/. https://doi.org/10.1109/IATMSI56455.2022.10119261.
3. Jing, S.; Kun, H.; Xin, Y.; Juanli, H. Optimization of Deep-Learning Network Using Resnet50 Based Model for Corona Virus Disease (COVID-19) Histopathological Image Classification. In Proceedings of the 2022 IEEE International Conference on Electrical Engineering, Big Data and Algorithms (EEBDA), Changchun, China, 2022, pp. 992–997. https://doi.org/. https://doi.org/10.1109/EEBDA53927.2022.9744883.
4. Huy, V.T.Q.; Lin, C.-M. An Improved Densenet Deep Neural Network Model for Tuberculosis Detection Using Chest X-Ray Images. *IEEE Access* **2023**, *11*, 42839–42849. https://doi.org/10.1109/ACCESS.2023.3270774
5. S and A. L. R. P J. Study of Deep Learning Approaches for Diagnosing Covid-19 Disease using Chest CT Images," 2023 7th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2023, pp. 263–269. https://doi.org/. https://doi.org/10.1109/ICCMC56507.2023.10083640.
6. Chowdhury, D.; et al. Detection of Sars-Cov-2 from human chest CT images in Multi-Convolutional Neural Network's environment," 2023 11th International Conference on Internet of Everything, Microwave Engineering, Communication and Networks (IEMECON), Jaipur, India, 2023, pp. 1–7. https://doi.org/10.1109/IEMECON56962.2023.10092370.
7. Zhou, Z.; Liu, Y.; Wang, Q.; Toe, T.T. Detection of Pneumonia Based on ResNet Improved by Attention Mechanism," 2023 IEEE 3rd International Conference on Power, Electronics and Computer Applications (ICPECA), Shenyang, China, 2023, pp. 859–863. https://doi.org/10.1109/ICPECA56706.2023.10076216.
8. Tariq, T.; Suhail, Z.; Nawaz, Z. Knee Osteoarthritis Detection and Classification Using X-Rays," in IEEE Access, vol. 11, pp. 48292–48303, 2023. https://doi.org/. https://doi.org/10.1109/ACCESS.2023.3276810.
9. Amelia Jiménez-Sánchez, Mickael Tardy, Miguel A. González Ballester, Diana Mateus, Gemma Piella, Memory-aware curriculum federated learning for breast cancer classification, Computer Methods and Programs in Biomedicine, Volume 229, 2023, 107318, ISSN 0169-2607. https://doi.org/10.1016/j.cmpb.2022.107318.
10. Jakaite, L.; Schetinin, V.; Maple, C. Bayesian Assessment of Newborn Brain Maturity from Two-Channel Sleep Electroencephalograms. *Comput. Math. Methods Med.* **2012**, *2012*, 1–7. https://doi.org/10.1155/2012/629654.
11. Jakaite, L.; Schetinin, V.; Maple, C.; Schult, J. Bayesian Decision Trees for EEG Assessment of newborn brain maturity', in 2010 UK Workshop on Computational Intelligence (UKCI), Colchester, United Kingdom, Sep. 2010, pp. 1–6. https://doi.org/10.1109/UKCI.2010.5625584.
12. Schetinin, V.; Jakaite, L.; Nyah, N.; Novakovic, D.; Krzanowski, W. Feature Extraction with GMDH-Type Neural Networks for EEG-Based Person Identification. *Int. J. Neur. Syst.* **2018**, *28*, 1750064. https://doi.org/10.1142/S0129065717500642.
13. Schetinin, V.; Jakaite, L.; Jakaitis, J.; Krzanowski, W. Bayesian Decision Trees for predicting survival of patients: A study on the US National Trauma Data Bank', Computer Methods and Programs in Biomedicine, **2013**, *111*, 602–612. https://doi.org/10.1016/j.cmpb.2013.05.015.
14. Swapna, G.; Vinayakumar, R.; Soman, K.P. Diabetes detection using deep learning algorithms. ICT express, **2018**, *4*, 243–246.
15. Available online: https://www.kaggle.com/datasets/iarunava/cell-images-for-detecting-malaria.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.