Malaria detection through RBC images using machine learning

1st Vivek Jain
2021218
IIIT Delhi
New Delhi, India
vivek21218@iiitd.ac.in

2nd Manan Garg
2021163
IIIT DELHI
New Delhi, India
manan21163@iiitd.ac.in

3nd Tushar
2020478
IIIT DELHI
New Delhi, India
tushar20478@iiitd.ac.in

I. Abstract

Malaria is a disease induced by a plasmodium parasite and transmitted by the bite of a female Anopheles mosquito infected with the parasite. More than 400 thousand individuals perish annually from malaria. The standard technique for detecting malaria involves preparing a blood smear and examining the stained blood smear under a microscope to detect the parasite genus Plasmodium, which heavily relies on the expertise of trained professionals. In this paper, we compare different feature extraction and classification methods on red blood cell smears from sampled cells to develop an accurate and reliable automated system for detecting malaria.

National Institutes of Health provided the dataset used in this study. The evaluation metrics consisting of accuracy, recall, precision, F1 score, and Area under the curve (AUC) were utilized to compare and select the best performing architecture.

II. Motivation

Malaria remains a global health concern with limited access to accurate diagnosis, especially in resource-limited areas. The disease claims the lives of more than 400,000 people annually. This project aims to use machine learning to automate the detection of malaria parasites in red blood cell (RBC) images. By leveraging machine learning, we can enhance accuracy, efficiency, scalability, and cost-effectiveness while continuously improving the diagnostic process. This project addresses the critical need for faster and more accessible malaria diagnosis.

The idea emerged from recognizing the challenges of manual malaria diagnosis and the potential of machine learning. Witnessing the impact of limited healthcare resources on accurate malaria detection sparked the concept. The advancement of machine learning inspired the vision to create an automated system for efficient, reliable, and accessible malaria diagnosis through RBC images. .

III. Introduction

Malaria, a fatal mosquito-borne disease, is caused by Plasmodium protozoan parasites that infect red blood cells and are transmitted through the bites of infected female Anopheles mosquitoes. The symptoms of an infected person are similar to those of the flu and may also include high fever, fatigue, shivers, septicemia, pneumonia, gastritis, enteritis, nausea, vomiting, migraines, and, in extreme cases, convulsions and coma, which can result in death. Malaria cannot travel from person to person, but it can be transmitted from mother to fetus and patients can become infected through blood transfusions or the sharing of instruments like syringes. Malaria is most prevalent in regions with mild, humid climates that are close to natural water resources and the territory of Anopheles mosquitoes. Blood films are commonly used for microscopic examination of blood cells to diagnose malaria. Every year, hundreds of millions of blood films are examined for malaria, which requires a trained microscopist to manually count parasites and infected red blood cells. Not only are accurate parasite counts crucial for malaria diagnosis, but also for testing for drug resistance, measuring drug efficacy, and classifying disease severity. For false-negative cases, this results in the unnecessary administration of antibiotics, a second consultation, lost work days, and in some cases, the progression of malaria to a severe form. For false-positive cases, a misdiagnosis necessitates the unnecessary use of antimalaria medications and the potential adverse effects associated with them, such as nausea, abdominal pain, diarrhea,and sometimes severe complications. Consequently, the F1 Score will serve as the primary metric for evaluating our model. The F1 Score is determined by the weighted average of Precision and Recall. Therefore, this Score takes into account both false positives and false negatives. F1 is typically more beneficial than accuracy in such situations, despite the fact that it is less intuitive to comprehend. We are attempting to automate the diagnostic system using various machine learning techniques that will assist human specialists in producing the most accurate diagnosis possible. We performed extensive preprocessing on the NIH malaria dataset and employed a few baseline classification models, including Decision tree, Logistic Regression, and Support Vector Machine. In addition, a number of preprocessing techniques, including various filters, global feature extraction, and local feature extraction, were applied to the dataset in order to optimize performance on an impartial test set.To compare the baseline models, we used a variety of evaluation metrics, including accuracy, recall, precision, F1 Score, and Area under curve (AUC).

IV. Literature Survey

Malaria is a significant global health concern that affects millions of people every year, primarily in developing nations. Early detection and treatment are among the most effective methods to manage malaria. Rapid diagnostic tests and microscopic examination of blood smears are commonly employed for malaria diagnosis, but they are time-consuming and prone to errors due to subjective interpretation by human experts. Machine learning (ML) techniques offer a promising solution to improve the accuracy and speed of malaria diagnosis by automatically analyzing rapid diagnostic tests (RDTs) and microscopic images. In recent years, researchers have utilized various ML algorithms for malaria detection using microscopic images. G.B. Saiprasath et al. (2019) compared the performance of five ML algorithms, including Decision Tree, Random Forest, Naïve Bayes, K-Nearest Neighbors, and Support Vector Machine, for malaria detection using microscopic images. The study found that Random Forest algorithm achieved the highest accuracy of 96.5. Mahdieh Poostchi et al. (2018) proposed an automated system for malaria detection using deep learning algorithms. The authors employed Convolutional Neural Networks (CNNs) to classify malaria-infected cells from microscopic images. The system achieved high accuracy, with a sensitivity of 96.6 Vijayalakshmi and Rajesh Kanna (2019) also presented a deep learning method for malaria detection utilizing microscopic images. The authors utilized a pre-trained CNN model, VGG-16, to classify infected and uninfected cells. This approach achieved an accuracy of 95.23 In a review article by Gaurav Kumar and Pradeep Kumar Bhatia (2014), different feature extraction techniques employed in image processing systems were discussed. The review emphasized that feature extraction is a crucial step in image processing systems as it reduces the image's dimensionality and enhances the ML algorithm's performance. Nagarajan Deivanayagampillai et al. (2017) proposed a feature extraction technique for melanoma detection in dermoscopic images. The authors used a combination of global and local features for classification, achieving an accuracy of 92.

V. Dataset

The dataset, NIH Malaria Dataset, was obtained from the National Institutes of Health (NIH). Our dataset consists of 2,7558 red blood cell stained specimens of the sampled cells that have been labelled. The dataset consists of 13779 images without infection and 13779 images with parasites. The regions of segmented red blood cells consist of three channels (RGB) that vary in size from 110 to 150 pixels and have a channel depth of three. Positive samples contained plasmodium, while negative samples lacked plasmodium but may have contained other substances, such as staining impurities. Later, the images were resampled to output dimensions of 64 x 64 with a channel depth of 3. Additionally, we computed the greyscale channel. Then, we created a list titled 'labels' to identify the uninfected as 0 and the parasitized as 1. After shuffling the Dataset, it is divided into two sets: training test and testing set with a ratio of 75:25.

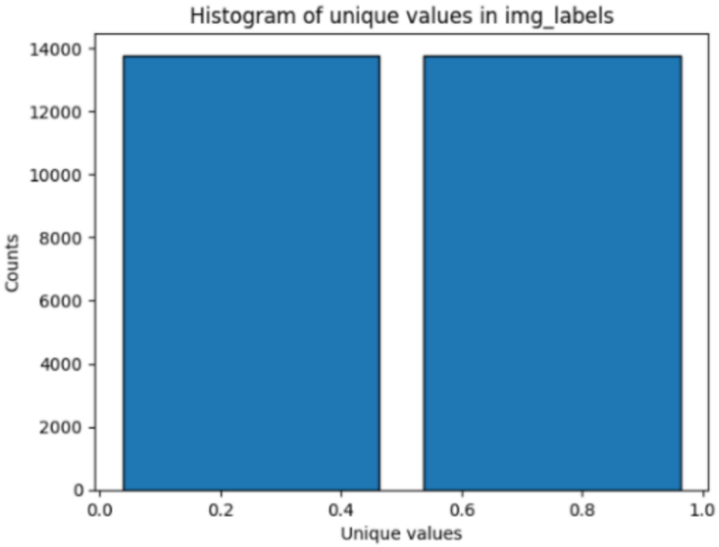

Fig. 1. Histogram depicting number of images from each class

VI. Methodology

We have utilized numerous filters, including edge detection algorithms Canny, Sobel, and Scharr, as well as Adaptive Histogram Equalization, also known as CLAHE. We will also use local feature extraction techniques such as SIFT (Scale-Invariant Feature Transform) and KAZE in addition to Global feature extraction, which detects features based on shape, texture, and color histogram. Then, we applied a few fundamental machine learning models, Decision Tree, Logistic Regression, and Support Vector Machine(SVM), Random Forest, KNN, AdaBoost and Naive Bayes to our processed data. We then used Ensembling on the top-performing models using various bagging and boosting techniques, such as Random Forest and Adaboost, grid search for hyperparameter optimization.

Preprocessing:
The unstructured pixel data in image patches will not directly aid in the classification process. Instead, we employ a representation that is unaffected by translation, rotation, and constant intensity offsets. The primary concern of the Plasmodium detection problem is the geometry of the objects within the input segments. We need a representation that is translation-, intensity-, rotation-, and size-invariant so that we scaled the images that were gathered in various sizes. Three edge detection filters, namely Canny, Sobel, and Scharr, have been utilized. Our dataset was preprocessed using edge detection because the Parasitized Cell Images contained purple-stained regions. As a result, we decided to use edge detection to enhance the image of the infected cell, which would aid us in feature extraction. Histogram Equalisation is

utilized for image segmentation. Still, simplistic histogram equalization generates a great deal of image noise because it considers the image's global contrast as well as its local contrast. Therefore, applying global equalization to our image may not yield optimal results. CLAHE, also known as Adaptive Histogram Equalization, was thus employed. It limits contrast and performs histogram equalization on tiny regions or tiles with high precision.

Global Feature Extraction:
On our filtered and unfiltered images, we then applied Global Feature Extraction, which extracted features based on shape, texture, and color histogram. Since we needed to calculate moments that are invariant to translation, scale, and rotation, we utilized OpenCV's Humoments to determine the Hu Moments of the structures within the input image. Haralick texture characteristics are derived from the co-occurrence matrix, which contains information about how image intensities in pixels with a particular position in relation to one another occur together. A color histogram depicts the distribution of colors within an image. A color histogram represents, for digital images, the number of pixels containing colors in each of a fixed list of color ranges that span the image's color space, the set of all possible colors.

Local Feature Extraction:
We applied two Local Feature Extraction techniques, SIFT and KAZE, to both filtered and unfiltered images. The Difference-of-Gaussians (DoG) operator is an approximation of Laplacian-of-Gaussians (LoG) (Tareen and Saleem, 2018). Feature points are detected by seeking for local maxima using DoG at varying image scales. The description method generates a total of 128 bin values by extracting a 16x16 neighborhood around each detected feature and further segmenting the region into subblocks. SIFT is insensitive to image rotations, scale, and limited affine variations. In addition to being invariant to rotation, scale, and limited affine, KAZE features are also more distinctive at differing scales at the expense of a moderate increase in computational time. KAZE's capabilities exploit non-linear scalespace via non-linear diffusion filtering, rendering blurred images locally adaptive to feature points, thereby reducing noise and preserving the boundaries of regions in subject photographs. Multiple scale levels are used to compute the scale normalized determinant of the Hessian Matrix for the KAZE detector. Using a moving window, the response maxima of a detector are extracted as feature points. Feature description explains rotation invariance by locating the prevalent orientation in a circular area surrounding each detected feature.

VII. Baseline Systems

We applied a few fundamental machine learning models to our filtered and unfiltered raw images with minimal preprocessing: Decision Tree, Logistic Regression, Support Vector Machine (SVM), Random Forest, KNN, and AdaBoost. The results of our models on raw data were abysmal, with the SVM model achieving the highest accuracy of 87.2 percent. This can be attributed to the low contrast of the image, the large number of features, and significant information not being given its due significance due to the lack of feature extraction.

Model	Accuracy	Precision	Recall	F1
Logistic	0.864	0.878	0.848	0.863
SVM	0.872	0.881	0.864	0.873
DT	0.726	0.706	0.787	0.744
RF	0.807	0.776	0.867	0.819
KNN	0.739	0.741	0.745	0.743
Ada Boost	0.726	0.706	0.787	0.744

Fig. 2. fig: Model Performance

On applying models to global features extracted from raw data, we obtained greater accuracy than with previous models, with SVM achieving a maximum of 87.1 percent. When we applied the classifiers to the images using the Canny Filter, SVM achieved a 86.3 percent accuracy. Filter for Detecting Edges The Canny filter outperforms its predecessors because the infected material is readily identifiable. However, it performs less well than other Edge Detection techniques because it reduces noise with a Gaussian filter and then applies Edge Detection to the prospective image, thereby diminishing the quality of the features. Using SVM to apply our baseline models to our CLAHE images yields accuracy of 85.2 percent. Simple histogram equalization generates a great deal of image noise because it evaluates the image's global contrast as well as its local contrast. Therefore, conducting global equalization on our image may not be optimal. In such instances, Adaptive Histogram Equalization, also known as CLAHE, can be utilized. It limits contrast and performs histogram equalization on tiny regions or tiles with high precision. With Sobel and Scharr filtered images, we achieve 89.4 percent and 90.9 percent accuracy respectively for SVM and 89.4 and 91.3 respectively for Logistic Regression. These Edge Detection Filters do not employ the noise detection technique and provide a higher-quality detection of patch edges that are parasitised. The accuracy of decision trees is diminished because they are ravenous, overgrown, and unpruned. Consequently, each tree has a high variance (tendency to overfit) but a low bias. They are susceptible to minor data perturbations: a small change can result in a drastically different tree. Consequently, we did not achieve favorable results with the decision tree and Naive Bayes when implementing our baseline models. Therefore, they were not an adequate baseline model. Also the accuracy on KNN, adaboost models wasn't as good as expected and Random forest provided a decent accuracy of 88.4 with sobel filter.

Methods	Logistic	SVM	DT	RF	KNN	NB	AB
raw	0.864	0.872	0.694	0.801	0.739	0.494	0.772
canny	0.839	0.863	0.777	0.859	0.830	0.731	0.837
clahe	0.827	0.852	0.590	0.750	0.797	0.672	0.801
scharr	0.894	0.894	0.785	0.876	0.851	0.762	0.859
sobel	0.913	0.909	0.792	0.884	0.853	0.784	0.862

Fig. 3. Model performance on various filters with global features

VIII. After Baseline Models

After observing the outcomes of the baseline models, we decided to attempt further improvement. Therefore, we extracted local features from our filtered and unfiltered images using SIFT and KAZE. SIFT produced the most accurate results when applied to unprocessed images with an SVM classifier, achieving 90.26 percent accuracy.

Feature	Logistic	SVM	RF
SIFT	0.894	0.903	0.906
KAZE	0.668	0.693	0.670

Fig. 4. Model performance with local features

When we applied both Local feature extraction techniques to the filtered images, the results were unimpressive. KAZE once again performed unfavorably in comparison to SIFT, but even SIFT performed comparatively unfavorably to its performance on unprocessed data. We discovered that KAZE was functioning inadequately because it was unable to extract many features from the data.

Feature	Logistic	SVM	RF
canny	0.820	0.817	0.808
clahe	0.869	0.881	0.879
scharr	0.811	0.817	0.807
sobel	0.798	0.805	0.801

Fig. 5. Filtered Images with Sift

Feature	Logistic	SVM	RF
canny	0.746	0.749	0.745
clahe	0.784	0.797	0.795
scharr	0.690	0.697	0.691
sobel	0.686	0.691	0.689

Fig. 6. Filtered Images with Kaze

Next, we performed Grid Search on our top-performing models, which were images filtered by Canny, Clahe, Sobel and Scharr, Global Feature Extraction on SVM classifier, and Local Feature Extraction from raw images on SVM classifier. For the grid search, we utilized polynomial and RBF kernels. On Sobel Filter with Global Feature extraction, we achieved a maximum precision of 91.91 percent.

	Accuracy	Precision	Recall	F1
SIFT	0.902	0.898	0.906	0.902
canny	0.869	0.865	0.877	0.871
clahe	0.865	0.864	0.869	0.866
scharr	0.906	0.923	0.889	0.906
sobel	0.919	0.938	0.899	0.918

Fig. 7. Grid Search on SVM

Then, we decided to utilize additional ensembling techniques, such as Bagging and Boosting, to determine if we could improve the accuracy of the same models. Random Forest Classifier, which fits multiple decision tree classifiers on various subsamples of the dataset and employs averaging to enhance predictive accuracy and prevent overfitting, was used for Bagging. Using Bagging, the accuracy of SIFT data remained the same, whereas Scharr and Sobel data became less precise.

Then, we conducted Boosting with both Logistic Regression and the Decision Tree using Adaboost. The AdaBoost classifier applies a classifier to the initial dataset. It then fits additional copies of the classifier on the same dataset, adjusting the weights of incorrectly classified instances so that subsequent classifiers prioritize complex cases. Using Boosting was ineffective because neither form of classifier produced superior results than Grid Search on Sobel.

	Accuracy	Precision	Recall	F1
SIFT	0.904	0.899	0.910	0.905
scharr	0.877	0.881	0.874	0.878
sobel	0.885	0.891	0.881	0.886

Fig. 8. Bagging with Random Forest

	Accuracy	Precision	Recall	F1
SIFT	0.878	0.920	0.828	0.872
scharr	0.853	0.856	0.853	0.855
sobel	0.840	0.854	0.825	0.839

Fig. 9. Adaboost with Logistic Regression

IX. Final Models

After applying traditional models such as SVM, RandomTree, AdaBoost,Decision tree, KNN, Naive bayes ,etc., we attained an accuracy of 91.9 percent, which is quite excellent, but it requires extensive preprocessing and hyperparameter

	Accuracy	Precision	Recall	F1
SIFT	0.876	0.923	0.820	0.868
scharr	0.852	0.849	0.861	0.855
sobel	0.853	0.842	0.872	0.857

Fig. 10. Adaboost with Decision Tree

tuning. Instead of relying on output that has been homogenized, we decided to use Deep Learning models to create a robust system that exploits the properties of images effectively. CNN was chosen because it is designed for image classification tasks, utilizes the convolution operation, and performs well with high-dimensional images. Each convolution task can acquire intelligent information, such as detecting an edge or a specific shape. Pooling reduces dimensionality even further. From then on, it has a fully connected layer that behaves like an ANN and aids in classification.

To implement CNN, we must initially arrange our dataset in a particular manner. We resized each image to 32x32 with three channels, normalized the data, and divided it 80:20 for training and testing purposes, respectively. We created DataLoaders with a collection size of 256 in the end. We chose Visual Geometry Group(VGG) as our CNN architecture because it makes higher use of the ”deep” in deep learning than its predecessors. It supplanted large kernels with multiple small kernels stacked on top of one another. This would enable us to obtain the same receptive field with fewer parameters, enabling us to go deeper.

We used the minimal version of the well-known VGG architecture by reducing the number of neurons in completely connected layers in order to reduce training overhead. Several convolutional layers, pooling, and bulk normalization are used to extract image features. The model is then classified using a handful of entirely interconnected layers. Relu activation is utilized to introduce non-linearity and to tackle the exploding/vanishing gradient problem. The dropout technique is used to reduce overfitting and generate the model. We implemented VGG11.

	Accuracy	Precision	Recall	F1
VGG11	0.998	0.997	0.999	0.998

Fig. 11. Training Performance of CNN

	Accuracy	Precision	Recall	F1
VGG11	0.940	0.923	0.961	0.941

Fig. 12. Testing Performance of CNN

X. Result and Analysis

The study explored various machine learning models for automated malaria detection using red blood cell smears. The results highlighted the importance of preprocessing steps and feature extraction techniques in enhancing model performance. Random Forests, when applied to preprocessed data, showed potential for malaria detection, but more sophisticated preprocessing may be needed for optimal performance. Logistic Regression demonstrated effectiveness, especially when used with appropriate preprocessing techniques. However, Support Vector Machine (SVM) emerged as the most promising model among the baselines, achieving a good accuracy. It showcased the potential of SVM in malaria detection, particularly when combined with effective preprocessing and feature extraction methods.

With various feature extraction techniques and hyperparameter optimization, we observed in our project how traditional machine learning models could yield high accuracy. When we employed Sobel Filter with Global Feature Extraction to the SVM Model, we obtained the highest F1 score of 0.918 for our conventional models. With a F1 score of 0.941 on testing the VGG11 CNN architecture, we saw that CNN architecture could be used to achieve even greater results.

XI. Future Scope

While we have achieved remarkable results on detecting malaria using deep learning techniques, there are several future scopes of work that could be considered for malaria detection.

Comparison with other models: While VGG models have shown good performance, it would be interesting to compare their results with other state-of-the-art models in the field of image classification, such as ResNet, Inception, or DenseNet. Transfer learning: Transfer learning is a technique where pretrained models are used as a starting point for new models trained on different datasets. It would be interesting to investigate the performance of transfer learning in the context of malaria detection and compare it with the performance of models trained from scratch.

XII. Conclusion

On the cell images, both traditional and deep learning models perform well for malaria detection. However, conventional models require a significant amount of feature extraction, processing, and engineering to produce accurate results. In contrast, deep learning models perform remarkably well. Always, there is a compromise between precision and power consumption. Deep networks are computationally intensive. Nonetheless, traditional architecture, when coupled with sound engineering behind the scenes, can generate excellent results, if not superior results than deep learning models.

References

1. Aimon Rahman, Hasib Zunair, M Sohel Rahman, Jesia Quader Yuki, Sabyasachi Biswas, Md Ashraful Alam, Nabila Binte Alam, and M. R. C. Mahdy. 2019. Improving malaria parasite detection from red blood cell using deep convolutional neural networks.

2. G.B. Saiprasath, N. Babu, J. ArunPriyan, R. Vinayakumar, V. Sowmya, and Dr Soman K. P. Performance comparison

of machine learning algorithms for malaria detection using microscopic images". IJRAR19RP014 International Journal of Research and Analytical Reviews, 6(1).

3. Gaurav Kumar and Pradeep Kumar Bhatia. 2014. A detailed review of feature extraction in image processing systems. In 2014 Fourth International Conference on Advanced Computing Communication Technologies, pages 5–12.

4.Mahdieh Poostchi, Kamolrat Silamut, Richard J. Maude, Stefan Jaeger, and George Thoma. 2018. Image analysis and machine learning for detecting malaria. Translational Research, 194:36–55. InDepth Review: Diagnostic Medical Imaging.

5.Nagarajan Deivanayagampillai, Suruliandi A, and Kavitha Jc. 2017. Melanoma detection in dermoscopic images using global and local feature extraction. International Journal of Multimedia and Ubiquitous Engineering,scopus, 12:19–27

6.Vijayalakshmi and Rajesh Kanna. 2019. Deep learning approach to detect malaria from microscopic images. Multimedia Tools and Applications, 79(21- 22):15297–15317

7.Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition.

8.Kirti Motwani, Abhishek Kanojiya, Cynara Gomes, ,Abhishek Yadav (2021). "Malaria Detection using Image Processing and Machine Learning" International Journal of Engineering Research Technology (IJERT)