Malaria Infected Cell Detection

1st Anubhav Shrimal
M.Tech CSE (MT18033)
IIIT Delhi
New Delhi, India
anubhav18033@iiitd.ac.in

2nd Vrutti Patel
M.Tech CSE (MT18020)
IIIT Delhi
New Delhi, India
vrutti18020@iiitd.ac.in

***Abstract*—Our aim is to build and compare various machine learning models to classify a given cell image as uninfected or infected by malaria parasite. We show an in depth analysis of various features like HOG, LBP, SIFT, SURF, pixel values with feature reduction techniques PCA, LDA along with normalization techniques such as z-score and min-max over different classifiers such as Naive Bayes, SVM, XGBoost, Bagging, AdaBoost, K-Nearest Neighbors, Random Forests and compare their performance by tuning different hyper-parameters. We evaluate the performance of these classifiers on metrics such as Accuracy, Precision, Recall, F1 score and ROC.**

***Index Terms*—SIFT, HOG, LBP, Min-max normalization, Z-score normalization, PCA, LDA, Naive Bayes, Bagging, AdaBoost, SVM, XGBoost, KNN, RF, CNN**

I. Motivation & Use Case

Hundreds of thousands of people die every year due to malaria majorly in the underdeveloped or developing countries due to delayed diagnostics and unavailability of specialized doctors in this field. The goal of our project is to automate the detection of malaria parasite in a given blood sample image accurately. It is a challenge in Computer Vision and Machine Learning to handle sensitive cases like detecting cancerous cell and classifying whether a person is suffering from a disease or not. This project is a good example of solving such issues and can be extended to other diseases as well.

II. Literature Review

In-depth explanation about malaria and how it is detected and diagnosed in real life is explained in paper [3] so that machine learning technique which automates the detection of malaria is close to the real world practise and gives better results. It is a survey as the authors have summarized different ways to obtain cell images with or without malaria parasites like light microscopy, binocular microscopy and so on. Various preprocessing techniques to enhance blood smear samples such as noise reduction, improving contrast and ways to do this. Different segmentation techniques like otsu thresholding, clustering, watershed, hough transform etc. Types and ways of feature computation like on the basis of color, texture or morphology and finally numerous classification techniques for supervised as well as unsupervised learning to detect malaria infected cells from images. Authors have cited various papers to support all the above methods.

In paper [1] the authors presents the evaluation of a color segmentation technique, based on standard supervised classification algorithms. They have implemented four different algorithms - K Nearest Neighbors, Naive Bayes, Support Vector Machine (SVM) and Multi Layer Perceptron (MLP) with different color spaces - RGB, normalized RGB, HSV and YCbCr. They have compared the results on the basis of F-score and inferred that all the algorithms are able to identify the uninfected cell images with a higher score as compared to parasitized cell images. The best performance for both classes is given by KNN classifier with normalized RGB color space and SVM classifier with YCrCb color space.

Paper [2] used the dataset of blood samples stained with giemsa so the authors were able to extract color histogram, granulometry, gradient and flat texture features after preprocessing data and used this features instead of just pixel value from the images which gives only color information as done in [1]. This features were given as input to SVM, nearest mean (NM), KNN, 1-NN, and Fishers linear discriminant classifiers. The results were compared by accuracy and precision.

As a preprocessing step staining variation has been removed from peripheral blood smear images, impulse noise has been reduced, Erythrocytes are segmented using marker controlled watershed algorithm and textural and morphological features have been extracted and trained on bayesian classifier in paper [4]. The proposed approach in paper [5] includes a preprocessing step to correct luminance differences, segmentation technique using the normalized RGB color space to classify pixels as erythrocyte or background followed by an Inclusion-Tree representation that structures the pixel information into objects. Then the classification process which identifies infected erythrocytes using a trained bank of classifiers. Average sensitivity and specificity are reported to evaluate the results.

III. Dataset Used

The dataset used is Malaria-dataset downloaded from https://ceb.nlm.nih.gov/repositories/malaria-datasets/. The dataset contains **27,558** images in total and has 2 classes, Paracitized and Uninfected (13,780 images each). Each image belongs to either one of the classes and is an image of a blood smears which either does or does not contain the malarial parasite. Images are of varying sizes and are resized to same size for training and testing purposes. The dataset is split into **90:10 Training-Testing** ratio. The training dataset is further split into **80:20 Training-Validation** set.

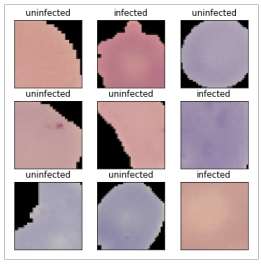


Fig. 1: Dataset Visualization

IV. Proposed Solution

A. Hand Crafted Feature Engineering

Different combinations of feature sets were used, some of which are shown in Table II & III. Following the **Ugly Duckling Theorem** many other combinations were tried to get the best feature set. Evaluation was performed with different classifiers following the **No Free Lunch Theorem** and model parameters were varied using **Grid Search** to find the best parameters. Dimensionality reduction technique were applied to preserve only important information from the features and reduce computational complexity. In PCA, number of components were preserved using **Elbow method** over variance of PCA projected data as shown in Fig. 3.

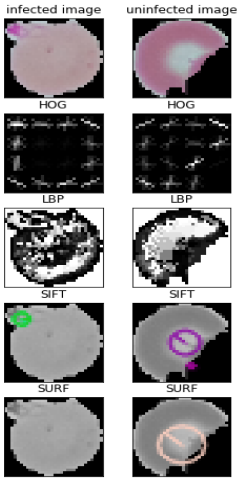


Fig. 2: Feature visualization

- **Feature Extraction**: Extract texture information from HOG and LBP, Spatial information from KAZE, SIFT and SURF and pixel information from color histogram, RBG and grayscale image.
- **Preprocessing**: Feature normalization by Min-Max and Z-score to bring features on a similar scale.
- **Feature selection**: To select proper combination of features and including only those features which performs better than random classifier.
- **Dimensionality Reduction**: PCA or LDA was applied to project the features with max separation. In PCA number of components were selected by plotting the variance over projected data Fig. 3.
- **Classification**: Different classifiers were trained and tested with different parameters and feature combinations.
- **Prediction and Evaluation Metrics**: Metrics such as accuracy, Precision, Recall, F1-score, ROC curve, confusion matrix were calculated to compare the performance of classifiers.

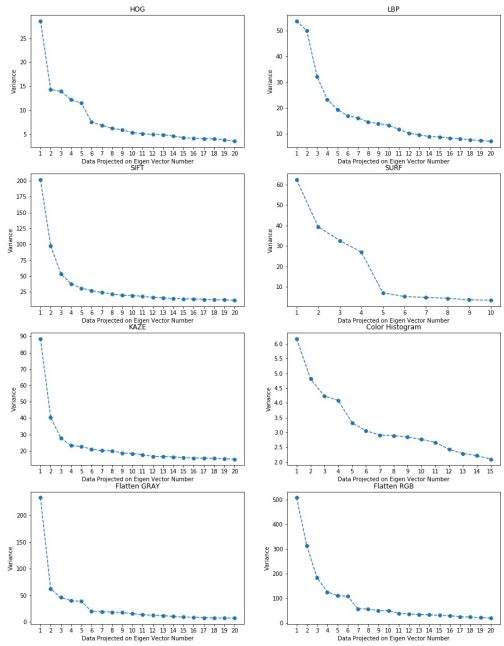


Fig. 3: Variance of PCA projected Z-score normalized data

B. Convolutional Neural Network implemented from Scratch

The input image size used is 64x64. We have applied RandomCrop and horizontal flips as image transformations. We created a CNN architecture from scratch with 5 Convolution layers, 3 Max-Pooling layers and Fully Connected (FC) layers of dimensions 500, 100 and 2. Activation function used is ReLU. Dropout of 0.2 is added in between FC layers. The optimizer used is Adam with a learning rate of 0.001 with Cross Entropy Loss function. The model was trained for 20 epochs.

V. Evaluation Metric

ROC curve of handcrafted features with different classifiers is plotted to analyse the performance of various classifiers on

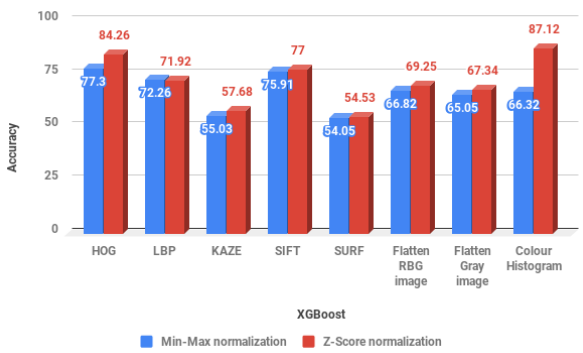


Fig. 4: Comparison between Min-Max & Z-score normalization

different dimensionality reduction techniques.

PCA reduced data includes HOG, LBP, Color Histogram, SIFT and RGB image as features and Fig. 6a is ROC over different classifiers for this features. LDA reduced data includes HOG, LBP, Color Histogram and SIFT as features and Fig. 6b is ROC over different classifiers for this combination of feature set. As the problem is 2 class classification LDA preserves only 1 feature irrespective of the feature type. So it can be inferred that PCA tends to perform better as compared to LDA on most of the classifiers.

TABLE I: Accuracy, Precision, Recall and F1-score on CNN

Accuracy	Precision	Recall	F1-score
88	93.2	82.8	87.7

VI. Observations and Results

- Table II shows bad features which are close to random in classification for example KAZE features.
- It can be seen from Table II that color histogram individually gives best accuracy but it was observed that precision contributed more as compared to recall in this accuracy. As our problem is malaria cell detection we need to focus more on improving recall rather than precision or accuracy. Hence, different features are combined to improve recall.
- Features were said to be bad because of close to random accuracy i.e. no differentiating capability.
- Table III shows that XGBoost on PCA projected feature set (HOG, LBP, Color Hist, SIFT & RGB) gave the best metric scores because boosting methods learn for misclassified data as well and XGB parameters (regularization, gradient descent) help learn better.

VII. Inferences

- Z-score normalization gave better accuracy than min-max normalization Fig. 4.
- Naive Bayes though gives good precision, performs poorly on infected class as it can be seen that recall value is too low.
- AUC for ROCs of uninfected class show that the trained models are able to differentiate well.
- Confusion Matrix of CNN shows that both the classes are classified reasonably well.
- The training and validation loss tells that the CNN model is generalized well over the data and is not under or over fitting.

VIII. Individual Contributions

1) Anubhav Shrimal:
   - Implemented feature extraction pipeline.
   - Applied PCA over different feature sets, number of components selection and combinations of various features to find the best feature set.
   - Implemented CNN model.
   - Created PCA variance plot & feature visualizations.
   - Created the tables for classifier comparisons and good or bad features.
   - Created report & poster.
2) Vrutti Patel:
   - Implemented classification pipeline.
   - Applied LDA over different feature sets, combinations of various features to find the best feature set and normalization techniques.
   - Implemented Grid Search to find the best model parameters for all the classifiers.
   - Created ROC, Pipeline diagram & Bar plot.
   - Created the tables for classifier comparisons and good or bad features.
   - Created report & poster.

References

[1] Daz, Gloria & Gonzlez, Fabio & Romero, Eduardo. (2007). Infected Cell Identification in Thin Blood Images Based on Color Pixel Classification: Comparison and Analysis. 4756.812-821.10.1007/978-3-540-76725-1_84.

[2] Malihi, L., Ansari-Asl, K. & Behbahani, A. (2013). Malaria parasite detection in giemsa-stained blood cell images. 2013 8th Iranian Conference on Machine Vision and Image Processing (MVIP), 360-365.

[3] Poostchi, Mahdieh & Silamut, Kamolrat & Maude, Richard & Jaeger, Stefan & Thoma, George. (2018). Image analysis and machine learning for detecting malaria. Translational Research. 194.10.1016/j.trsl.2017.12.004.

[4] Das, D.K., Ghosh, M., Pal, M., Maiti, A.K., & Chakraborty, C. (2013). Machine learning approach for automated screening of malaria parasite using light microscopic images. Micron, 45, 97-106.

[5] Gloria Daz, Fabio A. Gonzlez, and Eduardo Romero. 2009. A semi-automatic method for quantification and classification of erythrocytes infected with malaria parasites in microscopic images. J. of Biomedical Informatics 42, 2 (April 2009), 296-307. DOI: https://doi.org/10.1016/j.jbi.2008.11.005

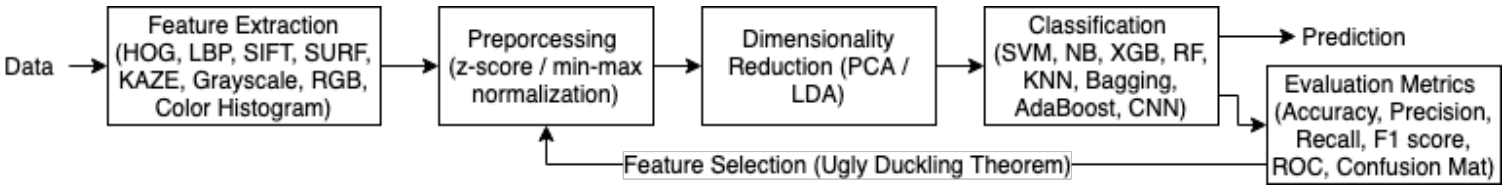
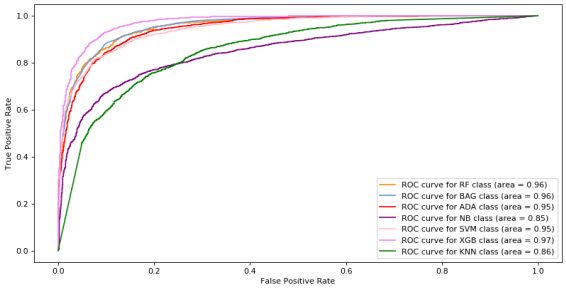
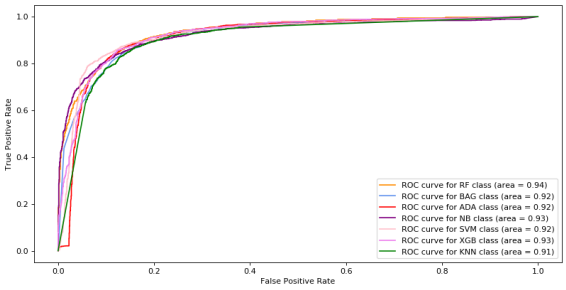


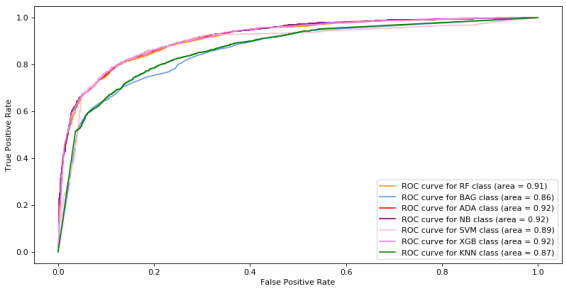
Fig. 5: Training Pipeline



(a) PCA reduced



(b) LDA reduced



(c) LDA on PCA

Fig. 6: ROC curves for classifiers trained over feature projections of HOG, LBP, SIFT and Color Histogram

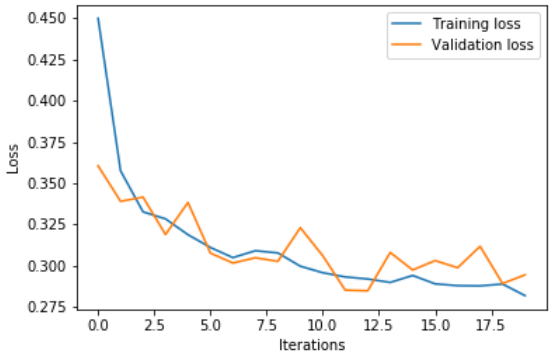


Fig. 7: Training and Validation Loss for CNN

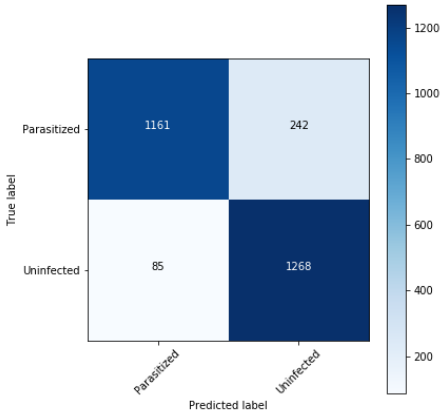


Fig. 8: Confusion Matrix for CNN scratch implementation

TABLE II: Good and Bad features on the basis of Accuracy on Random Forest classifier

Good Features			Bad Features		
Feature	**Number of components**	**Accuracy in RF**	**Feature**	**Number of components**	**Accuracy in RF**
HOG	324	**86.43**	KAZE	2048	60.52
PCA HOG	10	69.97	PCA KAZE	10	58.48
LDA HOG	1	**83.44**	PCA SURF	5	54.78
LBP	1024	71.49	PCA Gray	6	67.68
PCA LBP	10	72.19	LDA Gray	1	65.89
LDA LBP	1	68.11			
PCA SIFT	5	76.03			
Color Hist	512	**94.73**			
PCA Color Hist	10	75.35			

TABLE III: Comparing various classifiers with different feature sets over Accuracy / Recall / Precision / F1-score

Features\Classifiers		**RF**	**BAG**	**ADA**	**NB**	**SVM**	**XGB**	**KNN**
PCA (45 components): HOG(10), LBP(10), Color histogram(10), SIFT(5) and flatten RGB(10)	Accuracy	88.57	89.13	87.48	66.68	87.02	**91.49**	77.64
	Recall	88.37	89.59	86.38	42.24	85.16	**90.59**	72.27
	Precision	88.77	88.83	88.38	82.94	88.52	**92.30**	81.07
	F1-score	88.57	89.21	87.37	55.97	86.81	**91.44**	76.42
LDA (4 components): HOG, LBP, Color histogram, and SIFT	Accuracy	86.48	85.46	86.50	85.14	**86.82**	86.07	85.07
	Recall	86.06	84.89	85.34	81.86	**87.78**	85.16	85.21
	Precision	86.85	85.94	87.44	**87.69**	86.19	86.81	85.05
	F1-score	86.46	85.41	86.38	84.67	**86.98**	85.98	85.13
LDA PCA combined features (1 component)	Accuracy	83.60	76.91	83.80	82.17	83.65	**83.83**	79.34
	Recall	86.06	76.66	**86.70**	75.75	85.02	86.11	79.42
	Precision	82.09	77.15	82.02	**87.01**	82.81	82.42	79.38
	F1-score	84.03	76.90	**84.30**	80.99	83.90	84.22	79.40