R&D Project Report : Malaria Detection with Deep Learning

Amey Patil
IIT Bombay
ameypatil@cse.iitb.ac.in

Guide: Prof. Suyash Awate

Abstract

Malaria is a life-threatening disease caused by parasites that are transmitted to people through the bites of infected female Anopheles mosquitoes[1]. The most widely used method (so far) is examining thin blood smears under a microscope, and visually searching for infected cells. The patients' blood is smeared on a glass slide and stained with contrasting agents to better identify infected parasites in their red blood cells. The aim of this project is to come up with accurate and efficient technique to detect Malaria from the microscopic images of thin blood smears. The code is available to public at: Github code link.

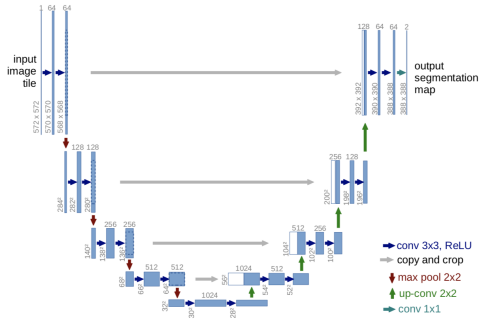

Figure 1. Unet architecture

1. Literature Survey

Malaria Detection from red-blood-cell images is a quite popular problem from past. There exist many techniques to classify the cell into infected or healthy cells. Running an SVM type of classifier on the manually selected feature sets of the cell images is one of the most successful classic Machine Learning technique used for the problem[2]. Then as the Deep Learning era came, models like VGG, AlexNet are being used for Malaria Detection with very good performance. Though the these successful approaches exist for classifying rbc into infected or healthy, we still have to manually give the bounding box of the cell from the image[3]. So this project focuses on the task of detection rather than just classification. The rest of the report will elaborate all the methods tried during the course of the project with model selected, training methods used(both successful and failed) and results for the problem.

2. Introduction

2.1. Dataset

Dataset used for training purpuses contains 1328 images, which contained images of stained blood cells in various stages of infection. The labels were provided as bounding boxes and each bounding box was either a 'red blood cell', 'trophozoite', 'ring', 'schizont', 'leukocyte' or 'gametocyte'. Fos simplicity of the basic models in the project we consider all the labels other than 'red blood cell' as 'infected'(label 1) and 'red blood cell' as 'not infected'(label 0). Every image in the dataset contains on an average around 74 cells, so the dataset contains 97580 bounding boxes for red blood cells and infected red blood cells. Malaria can be caused by very few infected cells in the blood, so usually the blood sample will contain very few infected cells than the normal red blood cells. Statistcally the dataset, contining 97580 bounding boxes, almost 96.6% of the bounding boxes are of normal red blood cells and only 3.4% bounding boxes has infected cells. This property of the dataset make training procedure very hard as the dataset is extremely skewed.

2.2. Unet

To start things we decided to use U-Net (Convolutional Networks for Biomedical Image Segmentation)[4] as the basic model. The U-Net architecture is illustrated in Figure 1. The network consists of a contracting path and an expansive path. The contracting path is similar to conventional convolutional networks. It consists of repeated application of 2 unpadded 3x3 convolutions, followed by a

downsampling using a max-pooling layer of kernel size 2x2 and stride 2. At each downsampling step, the number of features is doubled. In each step of the expansive path, we upsample the feature map followed by a 2x2 convolution that halves the number of feature channels. We concatenate the corresponding feature map from the contracting path, cropping the map as we do so to remove the extra border pixels in the expansive path. Finally, a 1x1 convolution is applied to the 64-component feature vector to the desired number of classes. The model is very good for segmentation tasks as proven for other problems. Main advantage of the model is it restores most information from the original image to the prediceted segmentation because of the concatenation of the previous feature maps from the downsampling layers to upsample layers. The training, testing and other experiments are done with the model are described in the Unet sections.

2.3. Classifier

The classifier model, as illustrated in figure 2, is writtern from scratch using convolutional layers and max-pooling layers. The model consist 3 blocks of convolutional layer(kernel size = 3), followed by LeakyReLu layer, Dropout layer BatchNorm layer and finally a 2D MaxPool layer(kernel size = 2). The last block does not have a MaxPool layer instead a single fully connecte layer followed by sigmoid activation to the output. The training, testing and other experiments are done with the model are described in the classifier section.

3. Unet for all cells

Unet is an architecture designed for the segmentation task on medical images. The malaria detection problem, though not exactly segmentation, can be thought of as a segmentation problem with labels as pxiel labels. So we use Unet architecture for pixelwise classification of the input image.

3.1. Training procedure

First to just to try the model we trained Unet to predict all masks labeles as cells(infected+normal rbcs). The images in the dataset are labeled pixelwise from the available bounding box labels as each pixel in the bounding box of the cell(infected or normal) is labeled as 1 and all background pixels are labeled 0. The Unet model used for training is same model as described in paper[4] and illustrated in figure 1, with image containing cells as input to to model and image of same size with pixel values as the predicted label to that corrosponding pixel in the input image as infected or normal rbc. The loss function used for the training is Binary Cross Entropy Loss(BCELoss) applied on every pixel prediction given the label for each pixel. The optimizer used is Adam optimizer provided in PyTorch. The only data augmentation used is Random Horizontal Flip of

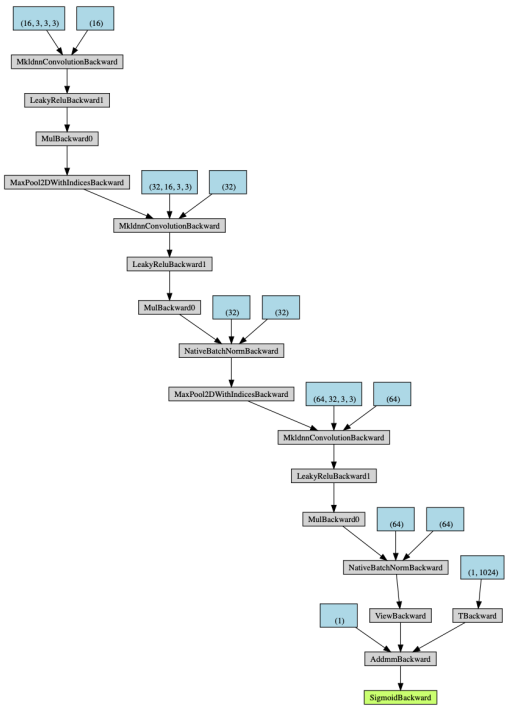

Figure 2. Classifier model architecture

Type	IOU score	Pixelwise Accuracy
Training	0.7901	0.9125
Test	0.7464	0.9031

Table 1. Accuracy for Unet predicting all cells

input image with probability 0.5. The Unet model is large with total of 31043523 parameters with 23 convolutional layers. hence though the original image size in the dataset is 1200x1600, we take random crop of shape 512x512 for the training purpose as the model with reduced image size itself takes around 8-9GB space on the GPU. Also the batch size used for training is 1, which is also recommended in the paper[4]. The dataset is already provided as train and test seperately. The train data is furter split into 2 parts 85% for training and 15% for validation. For selecting hyperparametes we make use of validation set and training graphs. The learining rate used in training is 0.001, the threshold used after sigmoid layer for prediction is 0.5 and model is trained for 10 epochs on the entire training dataset.

3.2. Testing and Results

For tesing the whole input image of oiginal size(1200x1600) is passed through Unet and not any crop. The accuray metrics used for measuring performance are intersection over union(iou) score and pixelwise accuracy are decribed in table 1. As expected, Unet has excellent performance for the task this simple. Some of the

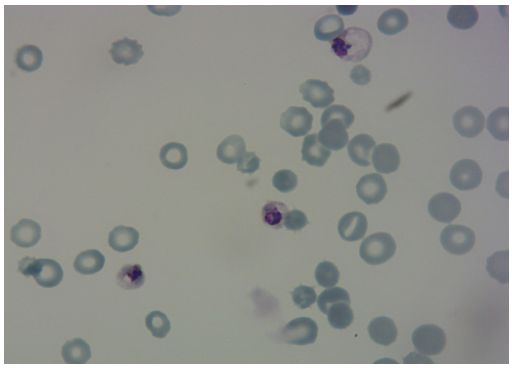

Figure 3. Input image

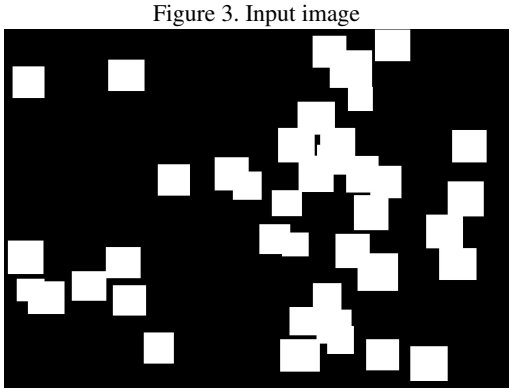

Figure 4. Given Label

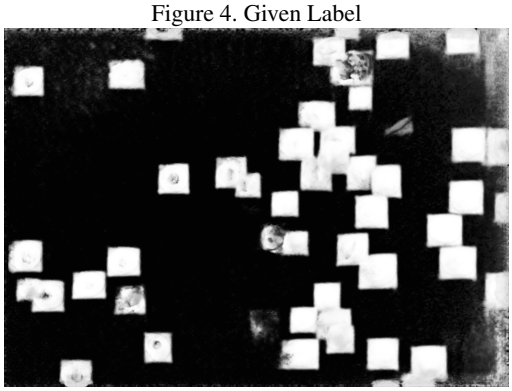

Figure 5. Predicted mask

results are interesting, for example, in the figure 3,4,5, at the bottom left of the image, label for the cell is missing in the dataset, but Unet still predicts the cell label correctly. Also it is noticable that unet learns the mapping to the square boundaries of each cell, as the cell boundaries have very large effect in convolutions as the pixel vlaues in image changes and is captured easily. But given the square bounday labels in the training, the Unet actually predicts bounding boxes for the input image in testing also. The detailed results are available on this Drive Link.

	IOU score				
Trained on	Full dataset	stain 0	stain 1	stain 2	stain 3
Full dataset	0.7901	0.7904	0.7976	0.8032	0.7159
Stain set 0	0.7125	0.7613	0.7350	0.7557	0.6861
Stain set 1	0.7398	0.7523	0.7697	0.7463	0.6945
Stain set 2	0.7406	0.7542	0.7531	0.7602	0.9698
Stain set 3	0.7034	0.7103	0.7412	0.7345	0.7102

Table 2. Accuracy variation for different stain sets

3.3. Stain variation in the images

The dataset contains images with slightly different stains. Actually some images contain portions where no cell exists but it contains colours in random shapes simply due to stain in the blood sample. But the problem is Unet predicts that portion with some proability as cell(some pixels in the region). So to check the effect of stain on the images, we manually created subsets of data with 4 different categories of stain(stain set 0-3) in the images, each set containing around 200-300 images from data and trained 4 different models on the different stain sets. The IOU score is taken as the measure of performance and the scores are as shown in table 2. The IOU values are highest when the full dataset is chosen for training for each stain set testing. This shows that the model generalises on the data well enough. Also the stain set 3 contains images with random shaped colour stains in it, but the iou for stain set 3 checked on stain set 3 is lower than model score trained on full dataset. This indicates that random stain structures are not captured by specificaly training that stain. The scores for indvidual stain trined models is more or less similar to the model trained on full dataset, difference is mainly arising due to training on limited data. Though trained on single stain each stain set trained model has good performance on full dataset, so generalisability of dataset is reliable and stain variation can be ignored. So now on we use full dataset itself for training without stain normalization.

4. Unet for only infected cells

After the experiment with the Unet detecting all cells was sucessful, the net obvious thing to try is detecting only infected cells from the image.

4.1. Training procedure

The labels for each image in dataset are created by taking bounding boxes for image and labelling infected red blood cells with 1 and rest all the pixels with label 0. If there are overlapping boxes, the label 1 bounding box is given preference and pixel is labeled 1. The Unet model used for training is the same model as described in paper[4] and illustrated in figure 1, with image containing cells as input to to model and image of same size with pixel values

Type	IOU score	Pixelwise Accuracy
Training	0.4993	0.9855
Test	0.3536	0.9749

Table 3. Accuracy for Unet predicting all cells

as the predicted label to that corrosponding pixel in the input image as infected or normal rbc and background. The loss function used for the training is Weighted Binary Cross Entropy Loss(BCELoss) applied on every pixel prediction given the label for each pixel. The optimizer used is Adam optimizer provided in PyTorch. The loss function uses different weights for different labels.

$$loss = -w*y*log(sigmoid(x))-(1-y)log(1-sigmoid(x))$$

Here w is the weight for the positive labels. As the infected cells are very rare in the dataset(only 3.4%) this weighted loss function had to be used. Of the several variants tried for training the best results were found on fixing the $w = 8$ and using data augmentation as Random Horizontal Flip. Most relevent varients tried was calculating w as the ratio of positive labels in the training batch with all cells in found in the batch. But this dynamic weighting factor has bad effects on training.

The Unet model is large with total of 31043523 parameters with 23 convolutional layers. Hence though the original image size in the dataset is 1200x1600, we take random crop of shape 512x512 for the training purpose as the model with reduced image size itself takes around 8-9GB space on the GPU. Also the batch size used for training is 1, which is also recommended in the paper[4]. The dataset is already provided as train and test seperately. The train data is furter split into 2 parts 85% for training and 15% for validation. For selecting hyperparametes we make use of validation set and training graphs. The learining rate used in training is 0.0003, the threshold used after sigmoid layer for prediction is 0.5 and model is trained for 10 epochs on the entire training dataset.

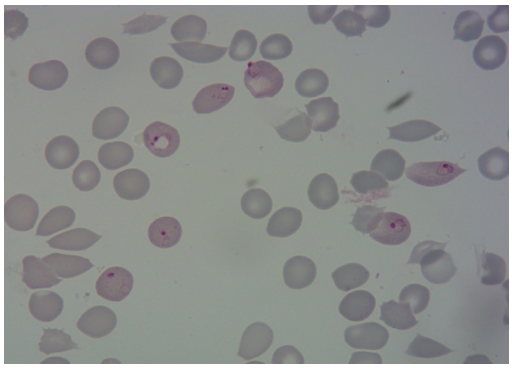

Figure 6. Input image

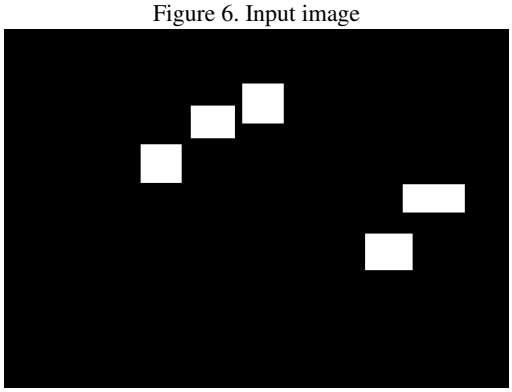

Figure 7. Given Label

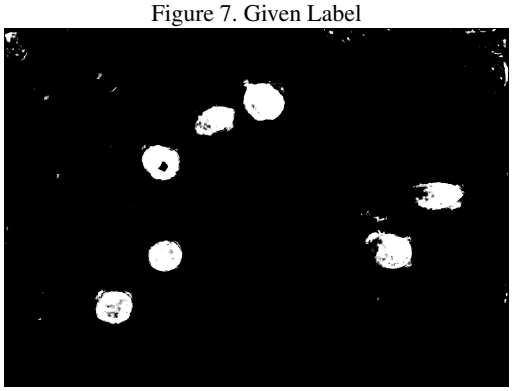

Figure 8. Predicted mask

4.2. Testing and Results

For tesing the whole input image of oiginal size(1200x1600) is passed through Unet and not any crop. The accuray metrics used for measuring performance are intersection over union(iou) score and pixelwise accuracy are decribed in table 1. Clearly the accuracy values are not satisfactory. But important thing notices in the results is the model does not misses most of the infected cells, which means the model predicts false positives abundantly. But this does not mean that the model is weighted on positive labels too much as lowering the weight misses the opportunity to balance data and also accuracy terms drops too low if weighting factor is kept lower. As seen, the pixel wise accuracy value is significantly higher but only IOU score has low value. This is because the image has mostly background(including normal rbcs) pixels and very few infected cell pixels. So the main misclassification done by model actually at random pixels. Model mostly labels a few pixels around non infected cells and boundary pixels as label 1. This is kind of unavoidable after certain extent as Unet moslty retains all the segmentation information from input image directly, so as cells(and their boundaries) are mostly differentiating features for the model, we are trying to classify cells with very minute differences(rings or colour) to be infected when other features of both the cells(infected and normal) are same. This is a little

overexpectation from the model and so we have to settle with some false positives in the end. While detecting infected cells, having false positives is always better than having false negatives, so weight factor is chose such that it maximises the iou considering false negatives are minimum(No infected cells are missed in detection). Also the prediction of rectangular bounding boxes could not be enforced during the training so the model predicts the cell shapes as the mask. This also contibutes to lowering the iou score. So to sum up, the model does a good job detecting infected cells but as the data and model are not consistent the accuracy values are not that high. The detailed results are available on this Drive Link.

Type	Training data	Test data
Accuracy	0.98	0.99
Precision	0.891878	0.900271
Recall	0.924240	0.925189
F1-score	0.906312	0.911357
MCC score	0.903972	0.909048
True Positives	3987	984
True Negatives	123383	30868
False Positives	479	110
False Negatives	331	84

Table 4. Performance measures for classifier model

5. Classifier

Unet has good performance on the Malaria dataset, but in certain cases, the prediction made by Unet is not totally reliable. So we learnt a classifier which will work as a validator for the Unet predictions. A classifier a simple model written from scratch with convolution layers, maxpool layers, dropout layer and fully connected layer illustrated in figure 2.

5.1. Training procedure

The training of classifier model is a little different from classic classifier training methods mostly because of 2 main reasons 1) Skewed dataset and 2) Application of classifier on Unet prediction. The dataset is very skewed in nature with 96.6% normal rbc boxes and 3.4% infected rbcs. So data augmentations/ data sampling/ loss function has to be changed. For this purpose we added 2 extra hypeparameters λ_1 and λ_2, λ_1 is used as weight in loss function and λ_2 is used as a probability to skip sample in sampling process. While sampling trining examples from the dataset, we skip negative examples with probability λ_2, i.e if random number generated is less than λ_2 the negative example is skipped, positive example is always picked up. This attempts to make data sampled for training to be less skewed that in the dataset. The optimizer used is Adam optimizer provided in PyTorch. For weighting loss function following objective is used.

$$loss = -\lambda_1 * y * log(sigmoid(x)) - (1-y) log(1 - sigmoid(x))$$

This loss function makes it possible to learn the skewed training data. The optimum values of λ_1 and λ_2 is $1/0.8$ and 0.9 respectively During sampling the crop from image is selected as bounding box information is given, but the crop is selected after randomly shifting it in bot X and Y direction with gaussian probability with mean at 0 and std as hallf of the crop size. This is done as an augmentation to the training data. More importantly this shifted crop training is useful when the classifier is used on Unet prediction in sliding window pattern, then the pixels around the cell center should also be predicted as the cell label and so shifted instances in training help model learn that mapping. The dataset is already provided as train and test seperately. The train data is furter split into 2 parts 90% for training and 10% for validation. For selecting hyperparametes we make use of validation set and training graphs. The learining rate used in training is 0.0003, the threshold used after sigmoid layer for prediction is 0.5 and model is trained for 10 epochs with batch size 200 on the entire training dataset. The input image size is 128x128(the mean size of bounding boxes), every crop is selected and brought to this shape. Finally due to sampling change and weighted loss function, the threshold for the sigmoid predictions had to change as the predictions come out to be near 1 and hence after hyperparameter tuning on validation set, optimal value for threshold is 0.90.

5.2. Testing and Results

Tesiting of the model is also done on shifted samples as that is the case found in actual application. As the dataset is skwed, the direct accuracy metric is not enough and so 1) preision, 2) recall, 3) F1-score and 4) MCC score are used and are as shown in table 4. The values are optimum and are satisfactory as F1-score and MCC values are 0.9. Also the model with given set of hyperparameters minimizes False Negatives. The detailed results are available on this Drive Link.

6. Unet + Classifier

We had trained Unet for detecting only infected cells but the problem with it was it was also classifying extra pixels as infected which lowered accuracy. But Unet cell detection was working excellent, so we add classifier onto the Unet all cell detection model. Now we pass input image from Unet to predict all cells(infected or normal rbcs) and then run classifier on the predicted mask, cropping image taking center as positively predicted pixel in Unet mask and marking it positive or negative given classifier output.

iou score	accuracy	FP count	FN count	Total count
0.4158	0.9818	41	34	453

Table 5. Performance for combined Unet and classifier model

6.1. Testing

Testing procedure is slightly heavy computationally, as Unet predicts mask and then classifier has to do a sliding window pass on the mask. We sample the crops from the masked image and make batch of 500 cropped images sharing GPU memeory making the process slightly efficient. The iou score and accuracy is calculated from the thresholded mask obtained from Unet followed by classifier with infected cell label image on full dataset. The detailed results are available on this Drive Link.

6.2. Results

Though the not the most efficient method, the combined model achieves higher accuracy than using just Unet for detecting infected cells. But for some cases the results are very vauge, whcih might be because of random stains in images. So though the iou scores are not as good as expected the count of False positives is low. This approach can be further made more accurate with some technique to get bounding boxes from predicted mask, like non maximal suppression. The manual count is done on a set of images with nice stain and distinguishable mask(mask where the different objects can be identified). The accuracy terms are given in table 5.

7. Conclusion

We developed a method, though not very efficient, quite accurate for malaria detection. Since detection is a different kind of problem than segmentation, Unet did not give the best performance. Still Unet performs very well when trained on enough data and right weighted loss function with approximately balanced dataset. The iou score is very tough to raise as the label contains bounding boxes of rectangular shapes and Unet tries to segment cells in cell shape exactly. The FP and FN counts are very good considering the model complexity. For bounding box prediction, we can use RCNNN type of methods which are very sucessful and efficient for object detection task. We tried Faster RCNN(most suitable architecture) for the problem, but unfortunately due to lack of time we could not get the desired results. We believe that Faster RCNN when trained with right measures will perform better than Unet as the bounding box regression will always give better iou than iou obtained with segmentation mask.

References

[1] World Health Organization: World Malaria Report (2018) 1

[2] Dhanya Bibin, Madhu S. Nair, P. Punitha *Malaria Parasite Detection From Peripheral Blood Smear Images Using Deep Belief Networks* 1

[3] Zhaohui Liang, Andrew Powell, Ilker Ersoy *CNN-based image analysis for malaria diagnosis* 1

[4] Ronneberger O., Fischer P., Brox T.: U-Net: Convolutional Networks for Biomedical Im- age Segmentation (2015), arXiv:1505.04597 [cs.CV] 1, 2, 3, 4