Soumya Nayak (nayak.soumya17@gmail.com
Downloaded: 9/6/2019 4:41:46 AM
IP Address: 136.233.11.130

Chapter 7
Bio-Medical Image Processing:
Medical Image Analysis for Malaria With Deep Learning

Rasmita Lenka
KIIT University (Deemed), India

Koustav Dutta
KIIT University (Deemed), India

Ashimananda Khandual
https://orcid.org/0000-0002-9357-9749
College of Engineering and Technology, India

Soumya Ranjan Nayak
https://orcid.org/0000-0002-4155-884X
Chitkara University, India

ABSTRACT

The chapter focuses on application of digital image processing and deep learning for analyzing the occurrence of malaria from the medical reports. This approach is helpful in quick identification of the disease from the preliminary tests which are carried out in a person affected by malaria. The combination of deep learning has made the process much advanced as the convolutional neural network is able to gain deeper insights from the medical images of the person. Since traditional methods are not able to detect malaria properly and quickly, by means of convolutional neural networks, the early detection of malaria has been possible, and thus, this process will open a new door in the world of medical science.

DOI: 10.4018/978-1-7998-0066-8.ch007

Copyright © 2020, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.

Soumya Nayak (nayak.soumya17@gmail.com
Downloaded: 9/6/2019 4:41:46 AM
IP Address: 136.233.11.130

INTRODUCTION

Application of Digital Image Processing and Deep Learning for analyzing the occurrence of Malaria from the Medical Reports is an important poverty. This approach is helpful in quick identification of the disease from the preliminary tests which are carried out in a person affected by Malaria. The combination of Deep Learning has made the process much advanced as the Convolution Neural Network is able to gain deeper insights from the medical images of the person. Since, traditional methods are not able to detect malaria properly and quickly, so by means of Convolution Neural Networks, the early detection of malaria has been possible and thus, this process will open a new door in the world of medical science. Automatic In this chapter, the main problems regarding the disease malaria is being discussed as well as the problems faced due in the quick diagnosis of malaria due to absence of advanced medical techniques is being brought into light. Most importantly, the solution to handle such a big problem using Deep Learning algorithms (Antony et al., 2016) is being provided in this chapter. The techniques to handle the problem and possible benefits by the use of Deep Learning (Kumar et al., 2013) technique is also being discussed.

BACKGROUND OF RELATED WORK

There are many components that make an area susceptible to an infectious disease outbreak. We'll the primary constituents below

- **Poverty Level**: When assessing the risk of infectious disease outbreak, we typically examine how many people in the population or at or below poverty levels. The higher the poverty level, the higher the risk of infectious disease, although some researchers will say the opposite — that malaria causes poverty. Whichever the cause we all can agree there is a correlation between the two.
- **Access to Proper Healthcare**: Regions of the world that are below poverty levels most likely do not have access to proper healthcare. Without good healthcare, proper treatment, and if necessary, quarantine, infectious diseases can spread quickly.
- **War and Government**: An area of the world that either has a corrupt government or is experiencing civil war will also have *higher* poverty levels and *lower* access to proper healthcare. Furthermore, if may be impossible for a corrupt government to provide emergency medical treatment or issue proper quarantines during a massive outbreak.

Soumya Nayak (nayak.soumya17@gmail.com
Downloaded: 9/6/2019 4:41:46 AM
IP Address: 136.233.11.130

- **Disease Transmission Vectors**: A disease vector is an agent that carries the disease and spreads it to other organisms. Mosquitoes are notorious for carrying malaria.

Once infected, a human can also be a vector and can spread malaria through blood transfusions, organ transplants, sharing needles/syringes, etc.

Furthermore, warmer climates of the world allow mosquitoes to flourish, further spreading disease. Without proper healthcare, these infectious diseases can lead to endemic proportions.

MAIN FOCUS OF THE CHAPTER

Here, in this section, the existing methods of the tests being conducted for detection of Malaria (Fabio, Gonzalez & Romero, 2009) and the problems faced due the existing methods are being discussed.

Tests for Malaria

T There are a handful of methods to test for Malaria (Hartl, 2004), but the two I most frequently have read about include:

1. Blood smears
2. Antigen testing (i.e., rapid tests)
    a. First, a blood sample is taken from a patient and then placed on a slide.
    b. The sample is stained with a contrasting agent to help highlight malaria parasites in red blood cells
3. clinician then examines the slide under a microscope and *manually counts* the number of red blood cells that are Infected.

According to the official WHO malaria parasite counting protocol, a clinician may have to manually count up to 5,000 cells, an extremely tedious and time-consuming process.

In order to help make malaria testing a faster process (Mitiku, Mengistu & Gelaw, 2000) in the field, scientists and researchers have developed antigen tests for Rapid Diagnosis Testing (RDT).

Soumya Nayak (nayak.soumya17@gmail.com
Downloaded: 9/6/2019 4:41:46 AM
IP Address: 136.233.11.130

For RDTs, a small device that allows both a blood sample and a buffer to be added. Internally, the device performs the test and provides the results While RDTs are significantly faster than cell counting they are also much less accurate.

An ideal solution would, therefore, need to combine the *speed* of RDTs with the *accuracy* of microscopy.

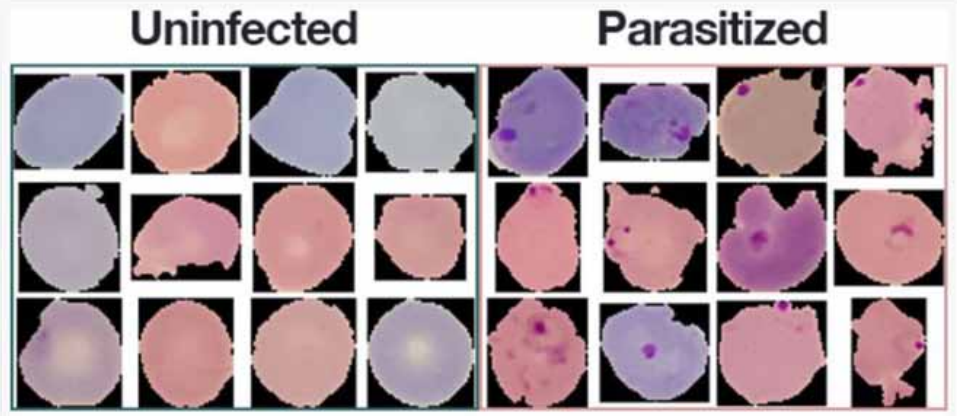
SOLUTIONS AND RECOMMENDATIONS

Thus, Using Deep Learning we will be tackling the same problem and thus bring out a solution which will monitor all the requirements as well as help in using the application in any moment of time. The malaria dataset we will be using in deep learning and medical image analysis tutorial is the exact same dataset that Rajaraman (at NIH -United States National Institutes of Health). used in their 2018 publication as shown in Figure 1.

The Convolution Neural Network Architecture used by us is Res-Net Architecture. The main problem with the approach taken by Rajaraman et al. is that it took 24 hrs to train the model. The Convolution Neural Network Architecture (Mitiku, Mengistu & Gelaw, 2000) makes use of Input Layers, Convolution Layers, Zero Padding Layers, Drop out Layers, Batch Normalization Process, Flatten Layers, Dense Layers and Output Layers with appropriate Activation Functions applied in between the layers while passing the Inputs. The complete Convolution Neural Network Architecture of the Res-Net Model is given below in Figure 2.

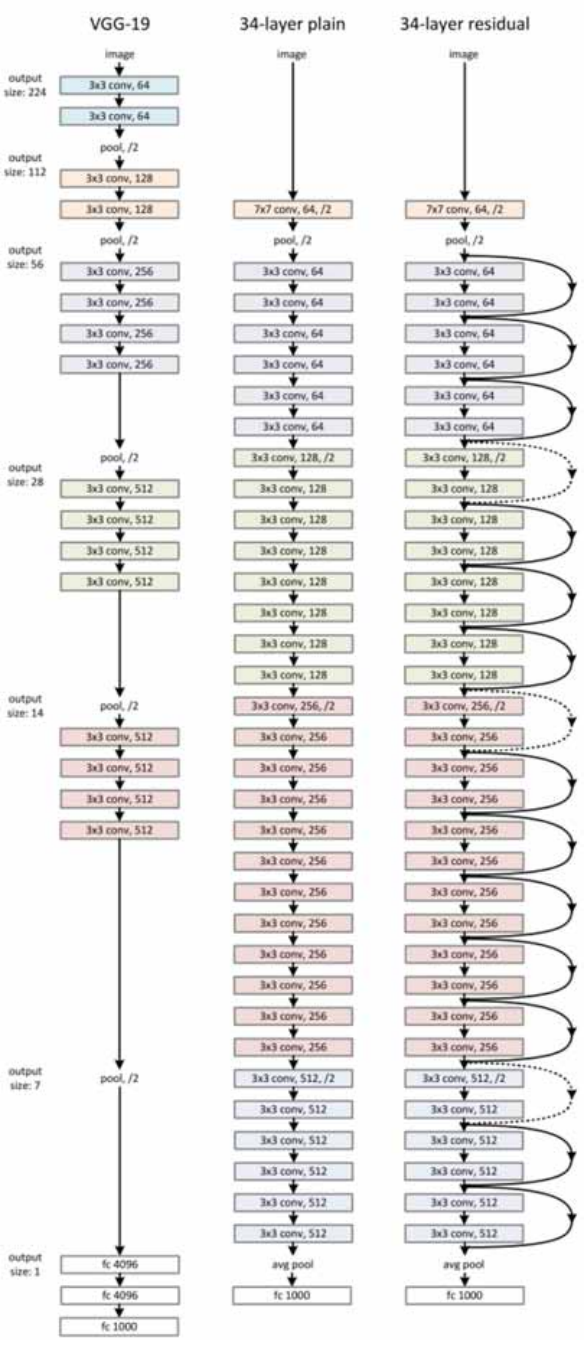
The dataset consists of 27,588 images belonging to two separate classes:

Figure 1. Data set image



Soumya Nayak (nayak.soumya17@gmail.com
Downloaded: 9/6/2019 4:41:46 AM
IP Address: 136.233.11.130

Figure 2. Convolution neural network architecture of the res-net model



Soumya Nayak (nayak.soumya17@gmail.com
Downloaded: 9/6/2019 4:41:46 AM
IP Address: 136.233.11.130

- **Parasitized:** Implying that the region contains malaria.
- **Uninfected:** Meaning there is no evidence of malaria in the region.

The number of images per class is equally distributed with 13,794 images per each respective class.

The code is given in below

```
# set the matplotlib backend so figures can be saved in the background
importmatplotlib
matplotlib.use(“Agg”)
# import the necessary packages
fromkeras.preprocessing.image importImageDataGenerator
fromkeras.callbacks importLearningRateScheduler
fromkeras.optimizers importSGD
frompyimagesearch.resnet importResNet
frompyimagesearch importconfig
fromsklearn.metrics importclassification_report
fromimutils importpaths
importmatplotlib.pyplot asplt
importnumpy asnp
importargparse
# construct the argument parser and parse the arguments
ap = argparse.ArgumentParser()
ap.add_argument(“-p”,”--plot”,type=str,default=”plot.png”,
help=”path to output loss/accuracy plot”)
args = vars(ap.parse_args())
# define the total number of epochs to train for along with the
# initial learning rate and batch size
NUM_EPOCHS = 50
INIT_LR = 1e-1
BS = 32
defpoly_decay(epoch):
# initialize the maximum number of epochs, base learning rate,
# and power of the polynomial
maxEpochs = NUM_EPOCHS
baseLR = INIT_LR
power = 1.0
# compute the new learning rate based on polynomial decay
alpha = baseLR * (1 - (epoch / float(maxEpochs))) ** power
# return the new learning rate
```

Soumya Nayak (nayak.soumya17@gmail.com
Downloaded: 9/6/2019 4:41:46 AM
IP Address: 136.233.11.130


```
returnalpha
# determine the total number of image paths in training, validation,
# and testing directories
totalTrain = len(list(paths.list_images(config.TRAIN_PATH)))
totalVal = len(list(paths.list_images(config.VAL_PATH)))
totalTest = len(list(paths.list_images(config.TEST_PATH)))
# initialize the training training data augmentation object
trainAug = ImageDataGenerator(
rescale=1 / 255.0,
rotation_range=20,
zoom_range=0.05,
width_shift_range=0.05,
height_shift_range=0.05,
shear_range=0.05,
horizontal_flip=True,
fill_mode="nearest")
# initialize the validation (and testing) data augmentation object
valAug = ImageDataGenerator(rescale=1 / 255.0)
# initialize the training generator
trainGen = trainAug.flow_from_directory(
config.TRAIN_PATH,
class_mode="categorical",
target_size=(64,64),
color_mode="rgb",
shuffle=True,
batch_size=BS)
# initialize the validation generator
valGen = valAug.flow_from_directory(
config.VAL_PATH,
class_mode="categorical",
target_size=(64,64),
color_mode="rgb",
shuffle=False,
batch_size=BS)
# initialize the testing generator
testGen = valAug.flow_from_directory(
config.TEST_PATH,
class_mode="categorical",
target_size=(64,64),
color_mode="rgb",
```

Soumya Nayak (nayak.soumya17@gmail.com
Downloaded: 9/6/2019 4:41:46 AM
IP Address: 136.233.11.130


```
shuffle=False,
batch_size=BS)
# initialize our ResNet model and compile it
model = ResNet.build(64,64,3,2,(3,4,6),
(64,128,256,512),reg=0.0005)
opt = SGD(lr=INIT_LR,momentum=0.9)
model.compile(loss="binary_crossentropy",optimizer=opt,
metrics=["accuracy"])
# define our set of callbacks and fit the model
callbacks = [LearningRateScheduler(poly_decay)]
H = model.fit_generator(
trainGen,
steps_per_epoch=totalTrain // BS,
validation_data=valGen,
validation_steps=totalVal // BS,
epochs=NUM_EPOCHS,
callbacks=callbacks)
# reset the testing generator and then use our trained model to
# make predictions on the data
print("[INFO] evaluating network...")
testGen.reset()
predIdxs = model.predict_generator(testGen,
steps=(totalTest // BS) + 1)
# for each image in the testing set we need to find the index of the
# label with corresponding largest predicted probability
predIdxs = np.argmax(predIdxs,axis=1)
# show a nicely formatted classification report
print(classification_report(testGen.classes,predIdxs,
target_names=testGen.class_indices.keys()))
# plot the training loss and accuracy
N = NUM_EPOCHS
plt.style.use("ggplot")
plt.figure()
plt.plot(np.arange(0,N),H.history["loss"],label="train_loss")
plt.plot(np.arange(0,N),H.history["val_loss"],label="val_loss")
plt.plot(np.arange(0,N),H.history["acc"],label="train_acc")
plt.plot(np.arange(0,N),H.history["val_acc"],label="val_acc")
plt.title("Training Loss and Accuracy on Dataset")
plt.xlabel("Epoch #")
plt.ylabel("Loss/Accuracy")
```

Soumya Nayak (nayak.soumya17@gmail.com
Downloaded: 9/6/2019 4:41:46 AM
IP Address: 136.233.11.130


```
plt.legend(loc="lower left")
plt.savefig(args["plot"])
```

RESULTS AND DISCUSSION

Overall, the entire training process took *only 54 minutes* (*significantly* faster than the 24-hour training process of NIH's method). At the end of the 50th epoch we are obtaining:

- **96.50% accuracy** on the *training* data
- **96.78% accuracy** on the *validation* data
- **97% accuracy** on the *testing* data

There are a number of benefits to using the ResNet-based model we are training for medical image analysis.

Thus, our model is a complete end-to-end malaria classification system.

Unlike NIH's approach which leverages a multiple step process of (1) feature extraction from multiple models and (2) classification, we instead can utilize only a *single, compact model* and obtain comparable results.

Speaking of compactness, our serialized model file is only 17.7MB. Quantizing the weights in the model themselves would allow us to obtain a model < 10MB (or even smaller, depending on the quantization method) with only slight, if any, decreases in accuracy.

Our approach is also faster in two manners.

First, it takes less time to train our model than NIH's approach Our model took only 54 minutes to train while NIH's model took ~24 hours.

>Secondly, our model is faster in terms of both (1) forward-pass inference time and (2) significantly fewer parameters and memory/hardware requirements.

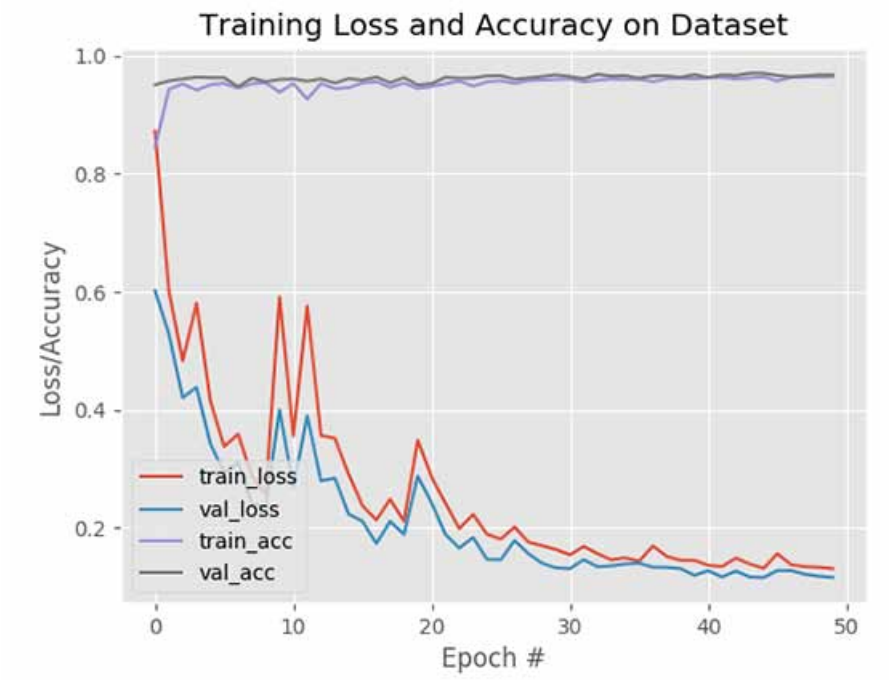
Considering the fact that NIH's method requires pre-trained networks for feature extraction. Each of these models accepts input images that have input image spatial dimensions in the range of 224×244, 227×227, and 299×299 pixels.

Our model requires only 64×64 input images and obtains near identical accuracy.

Based on our results we can see that we have created an automatic malaria classifier that is not only *more* accurate but significantly smaller, requiring less processing power as well. Given below is the Visualized final result:

Soumya Nayak (nayak.soumya17@gmail.com
Downloaded: 9/6/2019 4:41:46 AM
IP Address: 136.233.11.130

Figure 3. Graph between accuracy and epoch



FUTURE RESEARCH DIRECTIONS

This Research work of combining Deep Learning based Model in the case of Medical Imaging will be more modified by us and we will further focus on handling more complex diseases like Liver Cirosis, for MRI Segmentation and in Lungs Imaging which will thus open a new door in the field of Medical Science in handling different diseases.

CONCLUSION

Thus, our technique of using Residual Network based Deep Learning in case of Malaria Analysis has proved to be much better both taking into consideration the training time required during the process and also with respect to the accuracy level attained. Thus, our model and technique is robust and thus, much useful in analysis of Malaria .

Soumya Nayak (nayak.soumya17@gmail.com
Downloaded: 9/6/2019 4:41:46 AM
IP Address: 136.233.11.130

ACKNOWLEDGMENT

The authors are sincerely thankful to the Department of Computer Science and Engineering, KL University Vijayawada. And we are also thankful to all the authors of the references.

REFERENCES

Antony, J., McGuinness, K., Connor, N. E. O., & Moran, K. (2016). *Quantifying radiographic knee osteoarthritis severity using deep convolutional neural networks*. arxiv:1609.02469

Barbu, A., Lu, L., Roth, H., Seff, A., & Summers, R. M. (2016). *An analysis of robust cost functions for CNN in computer-aided diagnosis Comput*. Methods Biomech. Biomed. Eng. Imag. Visual.

Dasa, Ghosha, Palb, Maitib, & Chakraborty. (2013). Machine learning approach for automated screening of malaria parasite using light microscopic images. *Micron (Oxford, England)*, *45*(February), 97–106. PMID:23218914

Fabio, G. D., González, A., & Romero, E. (2009, April). A semi-automatic method for quantification and classification of erythrocytes infected with malaria parasites in microscopic images. *Journal of Biomedical InformaticsVolume*, *42*(2), 296–307. doi:10.1016/j.jbi.2008.11.005 PMID:19166974

Hartl, D. (2004). The origin of malaria: Mixed messages from genetic diversity. *Nature Reviews. Microbiology*, *2*(1), 15–22. doi:10.1038/nrmicro795 PMID:15035005

Mitiku, K., Mengistu, G., & Gelaw, B. (2000). The reliability of blood film examination for malaria at the peripheral health unit. *Tropical Medicine & International Health*, *5*, 3–8. PMID:10672199

Ruberto, C. D., Dempster, A., Khan, S., & Jarra, B. (2002). Analysis of infected blood cell images using morphological operators. *Image and Vision Computing*, *20*(2), 133–146. doi:10.1016/S0262-8856(01)00092-0

Soumya Nayak (nayak.soumya17@gmail.com
Downloaded: 9/6/2019 4:41:46 AM
IP Address: 136.233.11.130

KEY TERMS AND DEFINITIONS

Antigen: A toxin or other foreign substance which induces an immune response in the body, especially the production of antibodies.

Convolution: A function derived from two given functions by integration which expresses how the shape of one is modified by the other.

Malaria: An intermittent and remittent fever caused by a protozoan parasite which invades the red blood cells and is transmitted by mosquitoes in many tropical and subtropical regions.

Neural Network: A computer system modeled on the human brain and nervous system.

Normalize: Multiply (a series, function, or item of data) by a factor that makes the norm or some associated quantity such as an integral equal to a desired value (usually 1).