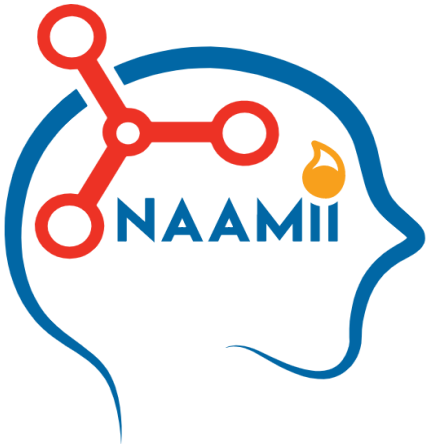
NepAl Applied Mathematics and Informatics Institute for research (NAAMII)



Performance Evaluation of VGG19 and LeNet5 for Detection of Malaria from Cell Images

By
Manoj Nepali

May 2023

TABLE OF CONTENTS

1. INTRODUCTION

Malaria is caused by the plasmodium parasite, and this parasite is transmitted to a healthy person by the bite of a mosquito carrying the parasite. Millions of people get infected by malaria every year. For example, in the year 2019, there were 229 million cases of malaria all over the world, out of which 409,000 people died. A lot of research is being done by different countries and WHO to fully eradicate malaria from the world. Many countries have been able to achieve this, but there are a lot of countries where malaria is still a very serious health problem.

Various efforts are being made to diagnose malaria easily and efficiently [1, 2]. The traditional method involves procuring the blood sample of the patient and taking it to the laboratory where a skilled and experienced health professional is required to discriminate between the healthy and parasitic red blood cells (RBCs). This process of identification is a very tedious and time-consuming task. So, to reduce the cost of blood tests and lab technicians for testing the malaria parasites in the blood sample, it is advised to use a cost-effective digitized approach in one or other forms to detect the parasitic cell to diagnose malaria.

Hence, Deep learning based malaria infected cell detection can be very effective and accurate for the malaria detection problem.

2. EXPERIMENTAL DESIGN

Once the model is initialized, the training is done with varying ratios of the amount of data for training and validation.

Normalization was done using the rescale parameter in both train_datagen and test_datagen, where the pixel values of the images are scaled down to a range of 0-1. This was an important step as it helps in faster convergence during the training process and makes the model more robust to different lighting conditions and color variations in the images.

Data augmentation was done to increase the amount of training data by artificially creating new images from the existing ones. This was done to prevent overfitting of the model and make it more generalized. In the given code, the shear_range, zoom_range, and horizontal_flip parameters are used to apply random shearing, zooming, and horizontal flipping to the input images during training.

I created the train, and validation set from the raw data pulled from kaggle.

The optimizer “Adam” is used with the default learning rate (i.e., 0.001) for the custom CNN (i.e., the proposed model

3. MODELS

VGG19 and LeNet5 are two popular convolutional neural network architectures used for image classification tasks.

VGG19:

VGG19 is a convolutional neural network (CNN) architecture that was developed by the Visual Geometry Group at the University of Oxford. It has 19 layers and is known for its simplicity and effectiveness. The architecture consists of a series of convolutional layers, followed by max pooling layers, and finally a few fully connected layers. The convolutional layers learn various features of the input image, while the fully connected layers perform classification based on these features. The VGG19 architecture has been trained on the ImageNet dataset, which contains over 1 million images in 1000 categories.

Transfer learning:

Transfer learning is a technique where a pre-trained model is used as a starting point for a new task. In our case, we used the VGG19 model that was pre-trained on the ImageNet dataset as a starting point for our malaria detection task. We added a few additional layers to the model and trained it on the malaria dataset. This allowed us to leverage the knowledge that the model had learned from the ImageNet dataset, which can save a lot of time and computational resources.

Hyperparameters:

- Learning rate: This is a parameter that controls how much the weights of the model are updated during training. A higher learning rate can lead to faster convergence but may result in unstable training. We used a learning rate of 0.01, which is a common value for transfer learning tasks.

- Batch size: This is the number of training examples that are processed in each training iteration. A larger batch size can lead to faster training, but may also require more memory. We used a batch size of 2000, which is a relatively large value due to the large size of the malaria dataset.
- Optimizer: This is the algorithm used to update the weights of the model during training. We used the Adam optimizer, which is a popular choice for deep learning tasks. It adapts the learning rate during training and can converge faster than other optimizers.

LeNet5:

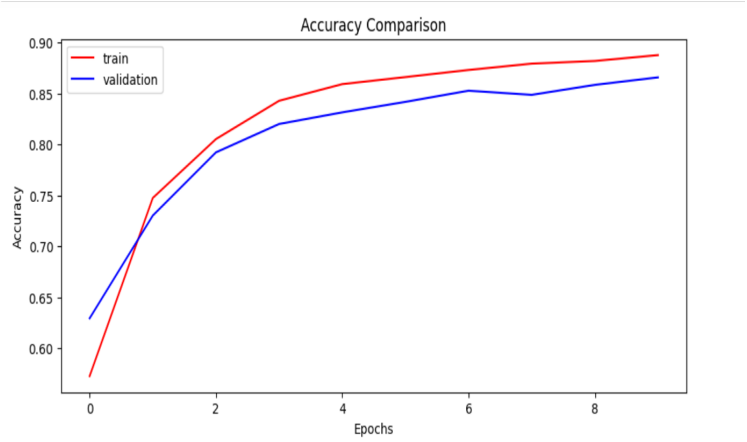
LeNet-5 is another CNN architecture that was developed by Yann LeCun and his colleagues in the 1990s. It was originally designed for handwritten digit recognition, but has since been applied to other tasks. The architecture consists of a series of convolutional layers, followed by max pooling layers, and finally a few fully connected layers. It is a relatively small network compared to modern architectures like VGG19, but can still achieve good performance on certain tasks.

Hyperparameters:

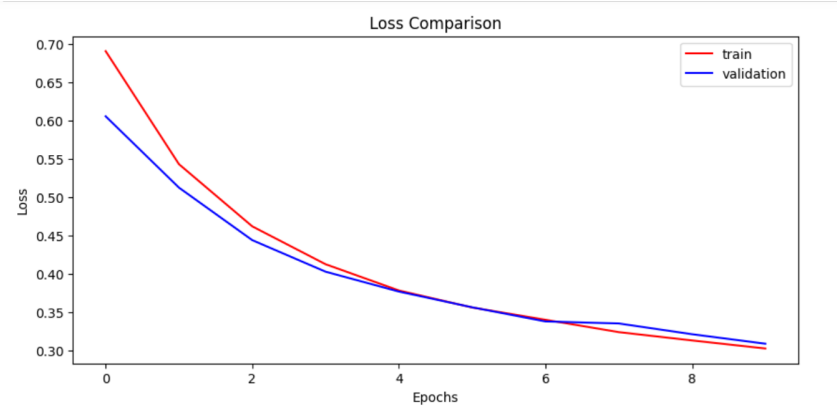
- Learning rate: We used a learning rate of 0.01, which is the same as the value we used for VGG19.
- Batch size: This is the number of training examples that are processed in each training iteration. A larger batch size can lead to faster training, but may also require more memory. We used a batch size of 2000, which is a relatively large value due to the large size of the malaria dataset.
- Optimizer: We used the Adam optimizer, which is the same as the optimizer we used for VGG19.

4. TRAINING INFORMATIONS

This is the Comparison of accuracy on training and validation set of **VGG-19** model.

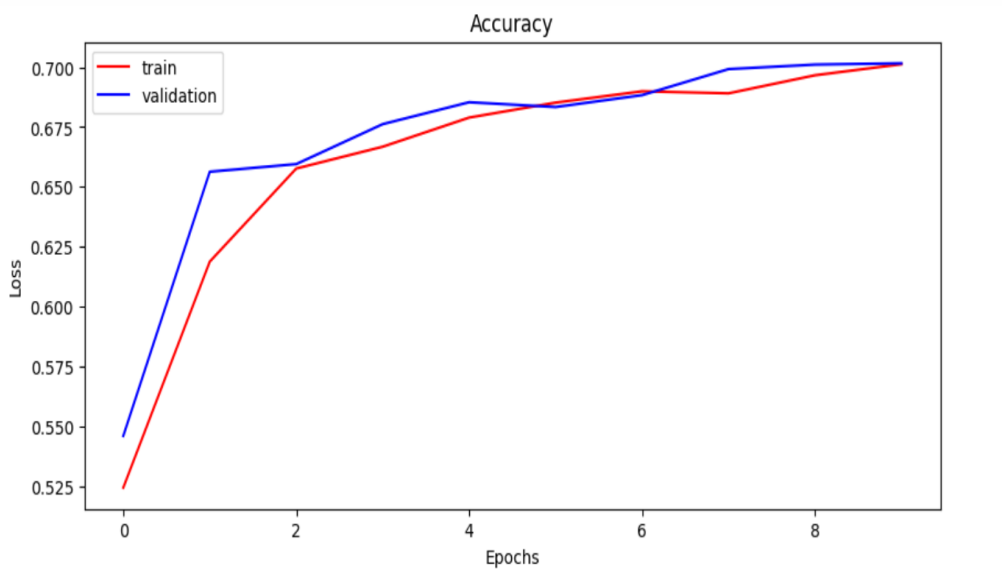


This is the Comparison of loss on training and validation set of **VGG-19** model.



From the above two graphs of loss and accuracy on training and validation I can clearly see there is no issue of overfitting. As there is no significant difference in accuracy on training and validation sets. And the loss is minimum in both the sets.

This is the Comparison of accuracy on training and validation set of **LeNet-5** model.



Same is the case for LeNet-5 as well.

5. EVALUATION METRICS

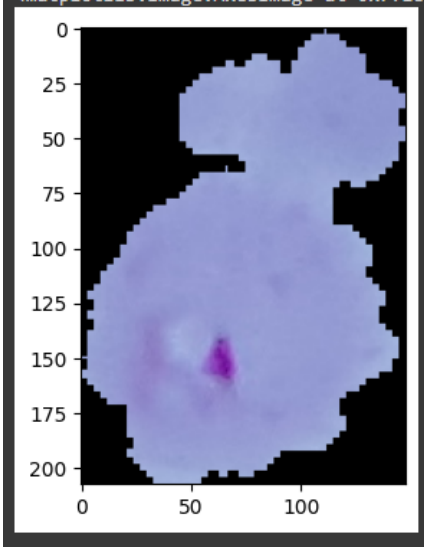
The label of the target class of the data was balanced so accuracy was also the great option for the evaluation metrics along with recall. The use of precision isn't justifiable for the medical report so it could be discarded. Hence, I focused on the loss, accuracy, recall, f1-score for the evaluation of my classification models but I have also evaluated the precision just for the sake of reference. I applied these metrics in the validation set. I have compared these metrics in the comparison section of the report in detail.

6. EXAMPLES

Random images was given to the models and the output be:

```
test_img = cv2.imread('/content/cell_images/Parasitized/C100P61ThinF_IMG_20150918_144104_cell_163.png')
plt.subplot(1 , 2 , 2)
plt.imshow(test_img)
```

```
<matplotlib.image.AxesImage at 0x7f1efc2225c0>
```




```
test_img.shape
```

```
(208, 148, 3)
```

```
test_img = cv2.resize(test_img,(98,98))
test_input = test_img.reshape((1,98,98,3))

pred = model.predict(test_input)
if(pred == 1):
  print("The Predicted Class is: Uninfected")
else:
  print("The Predicted Class is: Infected")
```

```
1/1 [==============================] - 1s 647ms/step
The Predicted Class is: Infected
```

Here our models predict the infected cell of the given input image.

7. COMPARISONS

Compare the accuracy, precision, recall, and F1-score of the two models.

Comparison of two models on the basic of f1 score, accuracy, recall,precision				
Models	Accuracy	Recall	Precision	F1-score
VGG19	0.87	0.96	0.81	0.87
LeNet5	0.70	0.80	0.67	0.72

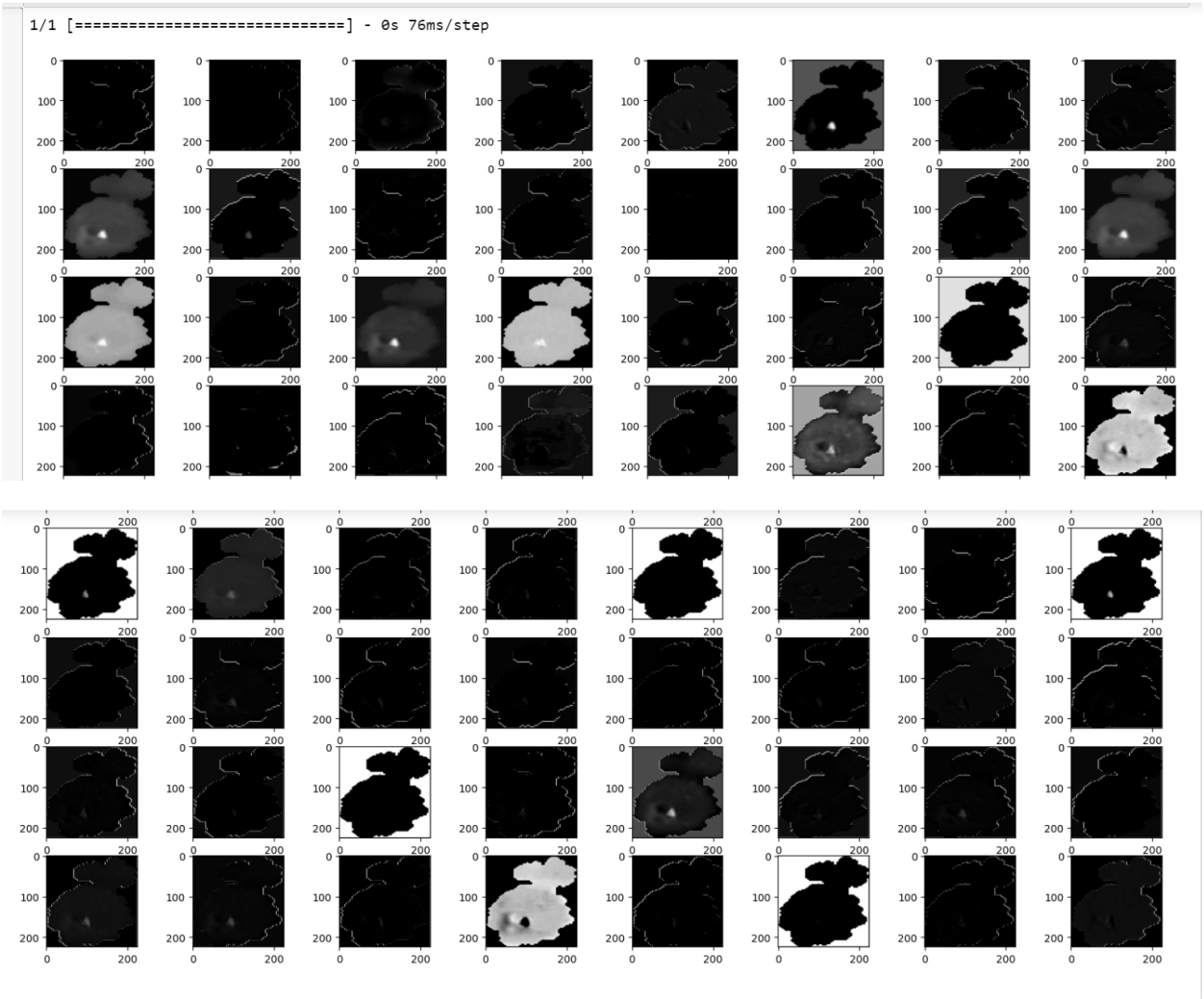
There are significant differences in the performance of the VGG19 and LeNet5 models. The VGG19 model has higher accuracy, recall, precision, and F1-score compared to the LeNet5 model. The VGG19 model was able to correctly classify 87% of the images and had an F1-score of 0.87, while the LeNet5 model correctly classified only 70% of the images and had an F1-score of 0.72. The VGG19 model performed better in identifying positive cases (infected cells) with a recall of 0.96, while the LeNet5 model had a recall of 0.80. The main difference between the two models is the number of layers and their complexity. VGG19 has 19 layers, while LeNet5 has only 5 layers. VGG19 also uses smaller filter sizes, which helps in capturing more fine-grained features. These differences in the model architecture could have led to the difference in their performance.

8. DISCOVERY / MODEL INTERPRETATION

A study published in the journal BMC Bioinformatics compared the performance of several deep learning models, including LeNet5 and VGG19, on a malaria image dataset. The study found that Lenet5 had a higher accuracy than VGG19 on this dataset. But in my comparison I found that in this malaria image dataset VGG-19 performed well with an accuracy of 87% compared to LeNet5's 70%. I have to further dig deeper on how and why I got these results.

To further interpret the models we use Class Activation Map(CAM) to visualize the regions of the input image that are most important for a given classification decision made by a convolutional neural network.

By examining the Class Activation Maps, we can get insights into how the models make their predictions. We can observe which regions of the image contribute the most to the classification decision and how the models focus on specific features or patterns to classify the images correctly. For example, we can see if the models focus on the parasite or the red blood cells to make the classification decision.



Possible Improvements:

1. We could have improved our model with use of other normalization techniques like batch normalization.
2. We could have used the ReLU activation function in our hidden layers.
3. We used other sophisticated models other than LeNet-5.
4. In LeNet-5 we could have decreased the batch size as this is a very simple CNN model.