38-616: Neural Networks and Deep Learning for Scientists
Homework 2

Name: Janvi Umesh Mirchandani
Andrew ID: jmirchan

I . Introduction

Convolutional neural networks (CNNs) are a specialized form of deep neural networks that excel in image classification tasks, as they can learn spatial hierarchies of features from input images using multiple convolution layers.

Plasmodium parasites are responsible for causing malaria which has been a significant burden on the healthcare system worldwide, particularly in impoverished nations. Effective diagnostic techniques are crucial for successfully treating and preventing malaria. Traditional diagnostic methods like staining blood smears can be expensive and labor-intensive, making it challenging to provide widespread access to diagnosis. To overcome these challenges, researchers have begun developing automated screening methods using machine learning technologies. These methods use photographs of blood cells to successfully identify malaria.

In this report, we provide a brief background on CNNs, and implementation of CNN with various pre-trained models using PyTorch with modifications. Our goal is to develop an effective and efficient diagnostic method for malaria that can be easily implemented in resource-limited settings.

II. Dataset Description

The dataset provided consists of images for training, testing, and making predictions on the hidden_test images. Each image in the dataset contains only one cell, which simplifies the classification task. The training set consists of 22,067 labeled images classified as 1 (infected) or 0 (uninfected), whereas the testing set has 2,757 images. The hidden_test set also contains 2,757 images, which are used for making predictions. Examples of images from the training, testing, and hidden_test sets are given below.

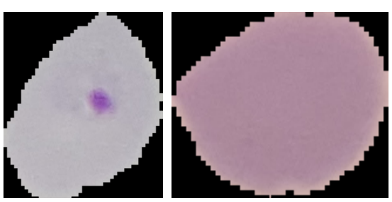

Figure 1. Image in Train Set

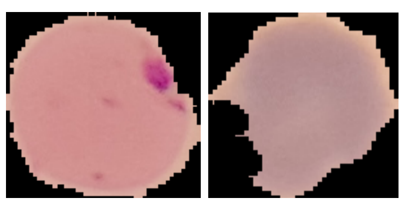

Figure 2. Image in Test Set

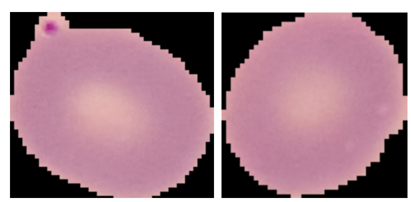

Figure 3. Image in Hidden_Test

In the above figures, the cells without the purple circles indicate that the Plasmodium genus responsible for malaria is absent. Whereas those with the purple circles indicate the presence of Plasmodium genus responsible for malaria.

III. Background of Convolution Neural Networks

CNNs are a specialized type of neural network that are utilized for image and speech recognition purposes. The main component of a CNN is the convolutional layer, which reduces the dimensionality of images without losing their information. To achieve this, the input image is divided into small sub-images, and a filter is applied to each sub-image with a step length that determines the distance between the sub-images. This greatly reduces the dimensionality of the image. Next, the pooling layer is applied, which computes either the average or maximum value of the result from the convolution layer. This helps to preserve small features in a few pixels that are crucial for the task solution. Finally, a fully connected layer is used to link the sub-images back together for classification. This layer is similar to those used in regular neural networks but can be used in conjunction with the convolutional and pooling layers to effectively classify images.

Batch normalization is another popular method used in CNNs, where the inputs to each layer are normalized to improve generalization and training stability. Batch normalization normalizes the mean and variance of inputs to each layer, thereby reducing the internal covariate shift. To expand the size of the training dataset and improve model generalization, CNNs use data augmentation to create additional examples of the existing data. Common data augmentation methods include random rotations, translations, flips, etc. For multi-class classification tasks, CNNs generally employ cross-entropy loss as their preferred loss function. The cross-entropy loss measures the difference between the true labels and the predicted probability for each class. Adam and stochastic gradient descent (SGD) are popular optimization techniques for CNNs. These methods update the weights of the model by utilizing the gradient of the loss function with respect to the weights.

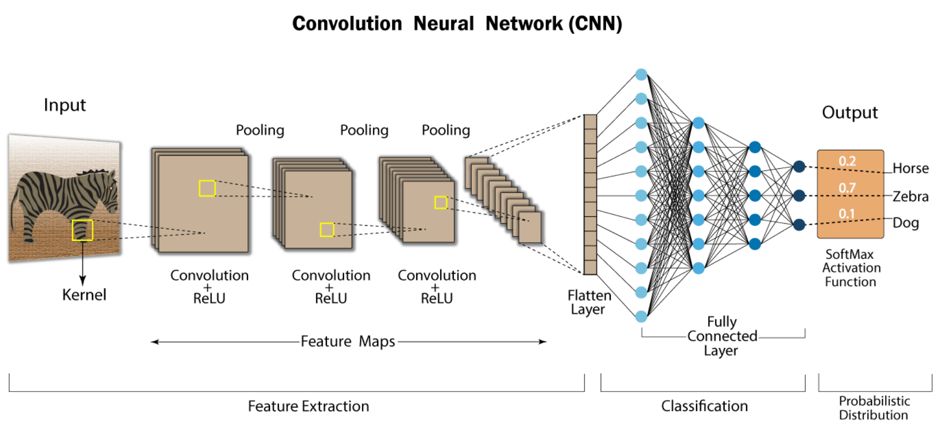

Figure 4. Basic CNN Architecture

In this malaria classification problem, we will explore machine learning models that employ stochastic gradient descent (SGD) optimizer, a learning rate of 0.001, data augmentation methods (rotation by 30 degrees, horizonal and vertical flip by 50%, resizing and normalization) to enhance the model's performance, and various combinations of convolutional and pooling layers using pre-trained models such as VGG, ResNet, and DenseNet.

IV. Architecture of the models

a) VGG (VGG16 and VGG19):

VGG16 and VGG19 are both convolutional neural network (CNN) models were introduced by the Visual Geometry Group (VGG) at the University of Oxford in 2014. They are commonly used for image classification tasks and have achieved state-of-the-art performance on various benchmark datasets.

The VGG16 model consists of 16 layers, including 13 convolutional layers, followed by 3 fully connected layers. The convolutional layers have a fixed kernel size of 3x3 and are stacked on top of each other, which results in a very deep network architecture. The VGG19 model, on the other hand, has 19 layers, with 16 convolutional layers and 3 fully connected layers.

Both models use max pooling layers with a stride of 2x2 after every two or three convolutional layers, which helps to reduce the spatial dimensions of the feature maps and control overfitting. They also use the Rectified Linear Unit (ReLU) activation function, which helps to introduce non-linearity into the model and improve its ability to learn complex patterns in the data.

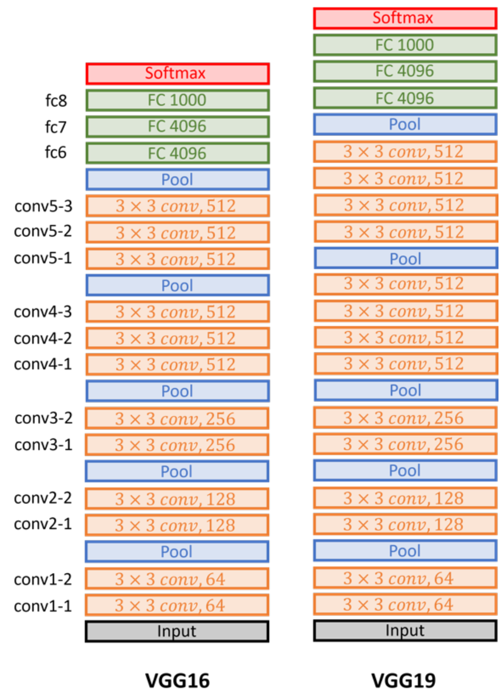

Figure 5. VGG16 and VGG19 Architecture

b) <u>ResNet (ResNet50 and ResNet101):</u>

ResNet50 and ResNet101 are both convolutional neural network (CNN) models that are part of the ResNet (Residual Network) family of models, which were introduced by Microsoft Research in 2015. ResNet models have been widely used for image classification tasks and have achieved state-of-the-art performance on various benchmark datasets.
ResNet50 and ResNet101 differ in the number of layers and the number of parameters in the model. ResNet50 has 50 layers, including 4 convolutional layers followed by 46 residual blocks, while ResNet101 has 101 layers, including 4 convolutional layers followed by 98 residual blocks.

In a residual block, the input to the block is passed through two or three convolutional layers, with batch normalization and ReLU activation functions applied after each layer. Instead of passing the output of the final layer directly to the next block, the output is added to the input to the block, creating a shortcut connection. This shortcut connection allows the network to bypass certain layers and learn more directly from the input, which helps to improve the flow of gradients during training and improve the overall performance of the network.

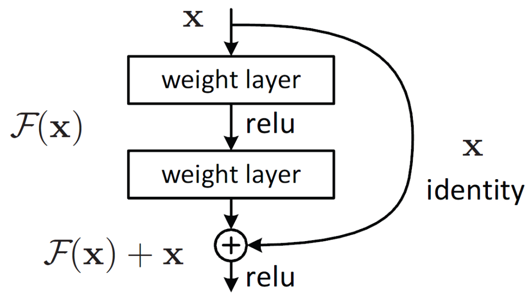

Figure 6. Residual block

Both ResNet50 and ResNet101 also use max pooling layers with a stride of 2x2 after the first convolutional layer to reduce the spatial dimensions of the input. They also use global average pooling to reduce the feature maps to a single vector, which is then fed into a fully connected layer for classification.

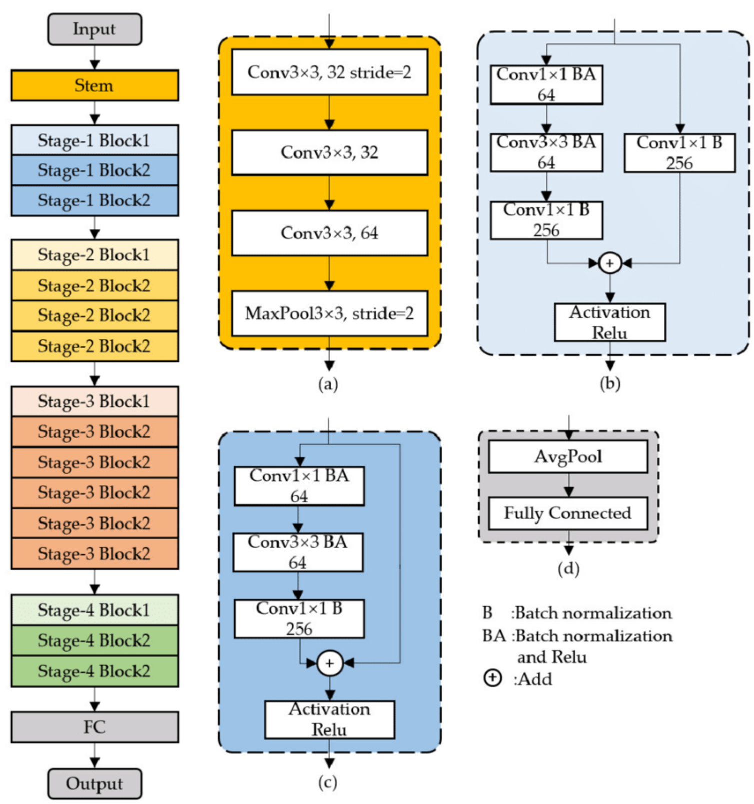

Figure 7. ResNet50 Architecture

ResNet models were trained on the ImageNet dataset, which contains millions of labeled images, and achieved very high accuracy rates on this dataset. They have also been used as a starting point for transfer learning, where pre-trained models are fine-tuned on smaller, specialized datasets for specific image classification tasks.

c) **<u>DenseNet (DenseNet121 and DenseNet201):</u>**

DenseNet121 and DenseNet201 are both convolutional neural network (CNN) models that are part of the DenseNet (Densely Connected Convolutional Network) family of models, which were introduced by researchers at Facebook AI Research in 2017. DenseNet models have been widely used for image classification tasks and have achieved state-of-the-art performance on various benchmark datasets.

DenseNet121 and DenseNet201 differ in the number of layers and the number of parameters in the model. DenseNet121 has 121 layers, including 4 dense blocks, while DenseNet201 has 201 layers, including 4 dense blocks as well. Dense blocks are a key feature of DenseNet models that help to improve feature reuse and reduce the number of parameters in the model.

In a dense block, each layer receives the feature maps of all preceding layers in the block as input, concatenated together. This allows each layer to reuse the feature maps learned by all preceding layers, which helps to improve the flow of gradients during training and reduce the number of parameters needed to represent the model.

Both DenseNet121 and DenseNet201 also use batch normalization and ReLU activation functions after each convolutional layer and use max pooling layers with a stride of 2x2 after the first convolutional layer to reduce the spatial dimensions of the input. They also use global average pooling to reduce the feature maps to a single vector, which is then fed into a fully connected layer for classification.

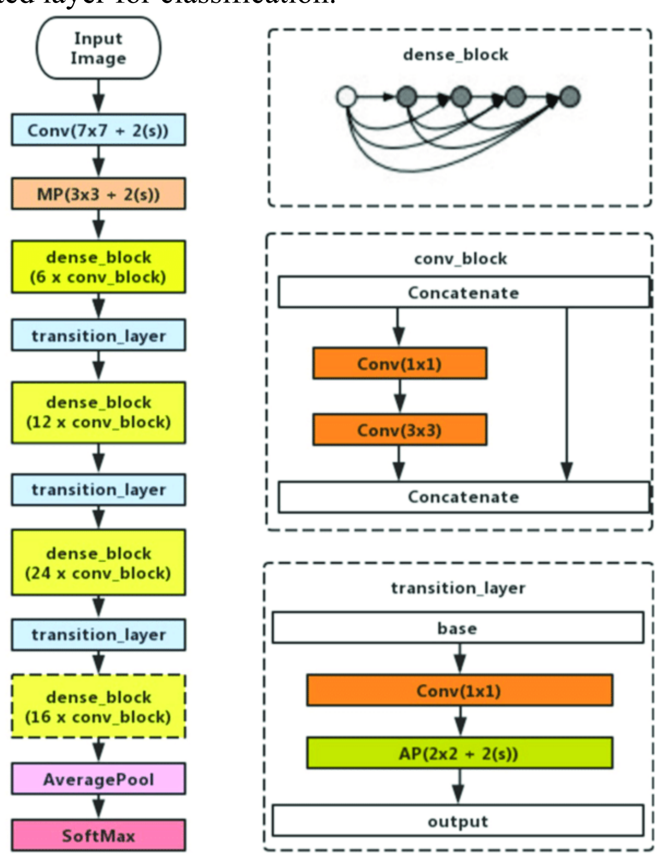

Figure 8. DenseNet121 Architecture

DenseNet models were trained on the ImageNet dataset, which contains millions of labeled images, and achieved very high accuracy rates on this dataset. They have also been used as a starting point for transfer learning, where pre-trained models are fine-tuned on smaller, specialized datasets for specific image classification tasks.

V. Results

In Table 1 given below, we can see the performance of 6 deep learning models compared based on their accuracy rates in the classification task using various data augmentation methods and optimization techniques. Based on the results, DenseNet201 achieved the highest Kaggle accuracy of 96.516% followed by ResNet101 with an accuracy rate of 96.008%.

MODEL	TEST ACCURACY	KAGGLE ACCURACY
VGG16	95.791	95.238
VGG19	96.226	95.137
ResNet50	96.444	95.936
ResNet101	96.335	96.008
DenseNet121	96.626	95.863
DenseNet201	96.589	96.516

Table 1. Accuracies of the CNN models

VI. Analysis and Conclusion

This study emphasizes on the significance of standardizing and augmenting data to improve the precision of deep learning models for classification tasks.

Among the models, DenseNet201 and ResNet101 achieved the highest accuracies, with 96.516% and 96.008% test accuracy, respectively. ResNet50 also performed well with 95.936% test accuracy followed by DenseNet121 with an accuracy of 95.863%. VGG16 and VGG19 achieved lower accuracies, but still showed good performance with test accuracies of 95.238% and 95.137%, respectively. The Kaggle accuracy results are slightly lower than the test accuracy results for all models, but generally follow the same ranking order. The high Kaggle accuracy scores of DenseNet201 and ResNet101 suggest that these models have a good generalization performance and can make accurate predictions on unseen data.

The results suggest that deeper models with more layers can improve accuracy. DenseNet201 and ResNet101 are deep residual networks that use skip connections to overcome the vanishing gradient problem during training. This makes them effective in learning complex features from images, leading to high accuracy in image classification tasks. In addition, DenseNet201 has a larger number of layers and parameters compared to the other models, allowing it to capture more intricate details in the images. On the other hand, VGG16 and VGG19 have a simpler architecture compared to the other models, which may explain their relatively lower Kaggle accuracy scores. However, it is important to note that all models achieved high accuracy scores in the test dataset, suggesting that they are still effective for image classification tasks.

The choice of architecture can also have a considerable impact on the model's performance, with DenseNet201 giving the most promising results for this specific dataset. These results depict the importance of selecting the appropriate model, optimization, and data preprocessing techniques in deep learning model development.

Stochastic gradient descent (SGD) with a learning rate of 0.001 was utilized for all models and proved effective in achieving high accuracy rates without overfitting. This was likely due to the large size of the training dataset (22,067 images) compared to the validation dataset (2,757 images), indicating that the models are capable of accurately classifying new data. The accuracy of all models was improved through the use of data augmentation techniques such as rotation, flipping, and normalization. Further research could explore other augmentation methods and architectures to enhance model accuracy.

Each model has its own strengths and weaknesses, and their performance depends on the specific requirements of the malaria dataset and genus Plasmodium detection task at hand. For instance, while DenseNet201 performed well in this classification task, it may not be the best option for other problems. Additionally, some models may require more time and computational resources to train, which can be a disadvantage in certain situations. Therefore, it is essential to consider the specific requirements and limitations of the task at hand when selecting a model.

VII. References

Das D K, Ghosh M, Pal M, et al. Machine learning approach for automated screening of malaria parasite using light microscopic images[J]. Micron, 2013, 45: 97-106.

Rajaraman S, Antani S K, Poostchi M, et al. Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images[J]. PeerJ, 2018, 6: e4568.

Tangpukdee N, Duangdee C, Wilairatana P, et al. Malaria diagnosis: a brief review[J]. The Korean journal of parasitology, 2009, 47(2): 93.