Malaria Cell Image Recognition

Chuping Wang, Buyun Gao, Fanfei Xu, Wenquan Chen

June 2019

1 Abstract

In this paper, we present our project about implementing four neural network models: VGG16, Xception, ResNet, and traditional CNN to boost the process of detecting malaria-infected cells and compared the performance. By using the malaria cell image dataset from NIH and performing some data augmentation techniques, our models achieved a high accuracy over 97% on the testing set, which is of great significance in the diagnosis of malaria disease.

2 Introduction

Malaria is a blood disease caused by the Plasmodium parasites transmitted through the bite of female Anopheles mosquito. However, the accuracy of microscopic diagnosis depends on smear quality and observer's expertise as well as the endemic or resource constrained areas. Traditional microscopic examination could be arduous for large-scale diagnoses resulting in poor quality. By adopting deep learning techniques, the efficiency and accuracy of diagnosing malaria could be improved, which is of great significance for epidemic and resource-scarce areas.
Our image dataset comes from *National Institutes of Health*, known as NIH. The dataset is classified into two categories: parasitized and uninfected, containing 27,558 images in total. After applying data augmentation, we enlarged the dataset to 11,0232 by rotating the original image to different angles and enhanced the contrast of the RGB image to boost the feature extraction. The input size is maintained at 64*64*3, except for Xception, which uses the input of size 71*71*3. The output is the prediction of whether the image is infected with malaria or not, i.e. 0 for uninfected or 1 for infected.

3 Related Work

Studies on improving the efficiency of malaria infection diagnosis have been started years ago, and those studies have experimented methodologies to generate data from the blood sample. From the review written by M. Poostchi et.al(2018)(3), we read a comprehensive list of different approaches on data processing. From blood sample to image data, there are thick and thin smear, different staining, different microscopy including light microscopy, fluorescent microscopy and quantitative phase imaging. And to process the image data, methods like noise reduction, low image contrast, and uneven illumination have been applied. After that, the one image per blood cell is segmented. In our study, the NIH dataset images (4) are generated from Giemsa-stained thin blood smear slides with level-set based algorithm to detect, and segment the red blood cells.

Previous automated diagnosis has focused on traditional classification methods. In a study conducted by Y. Purwar et. al in 2011(5), they used k-means clustering and reached between 50 and 88 percent recall on over 150 samples; and from another study by D. Anggraini et. al(2) in the same year, a classification model based on Bayes Decision Theory has reached a 92.59 recall and F-1 score of 0.78. The difference of results between these two studies suggests that supervised learning is the more robust one.

We also learned the paper Transfer Learning with ResNet-50 for that Malaria Cell-Image Classification by S.B Reddy and D.S Juliet(1), which we later decided the convolution neural networks that we are going to compare. In this paper, the authors mentioned the gradient vanishing problem caused when choosing Sigmoid Function as the activation function in Neural Networks with deep depth. This problem is alleviated by the introduction of ResNet construction, which used the residual in studying, which we later examined its advantages in our study.

In the same paper, it also proposed their method of using pre-trained weights before the fully connected layers. S.B Reddy and D.S Juliet(1) proposed that they used a ResNet 50 layer with pre-trained weights for the ResNet50 layer to perform feature extraction. Using pretrained neural network before the fully connected layers instead is a common method applied in building Convolution Neural Network models, in this project, we adopted this idea and build three models by applying a fully connected layer in Keras with Xception, ResNet and VGG16 in the front, respectively.

4 Dataset and Features

4.1 Overview

Our dataset comes from National Institues of Health(4). It contains a total of 27,558 cell images of different sizes with equal instances of parasitized and uninfected cells, separated into two different folders named with uninfected and parasitized. The original dataset is as shown in Figure 1. Input features are image pixels containing three-channel RGB values. Features would be extracted and changed after the convolution operation in each convolutional layer, depending on the size of the kernel.

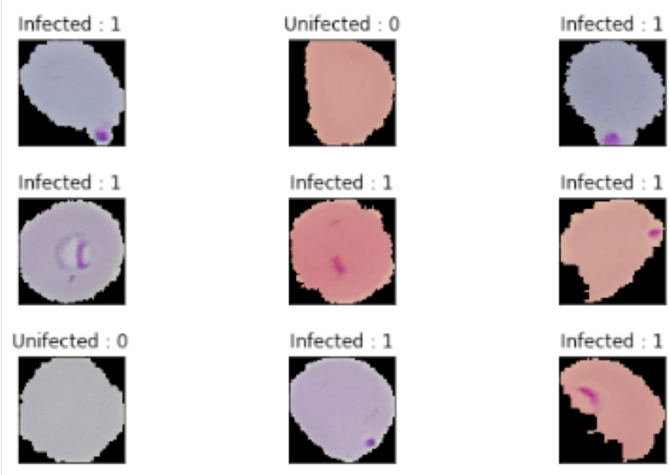


Figure 1: Original dataset images

4.2 Preprocessing

In order to boost the training process, the dataset is preprocessed in the following steps, including shaping, labeling, data augmentation, etc. We manually labeled the data with 1 for infected and 0 for uninfected. Then, by adopting data augmentation technique, such as rotating and flipping, we enlarged our dataset from 27,558 to 11,0232 images, in order to better train the model. We also processed all images to increase the contrast, for the purpose of boosting the feature extraction within the convolutional layers. All images then will be reshaped as 64*64 as input to the network. Our enhanced images are as the Figure 2 shows.

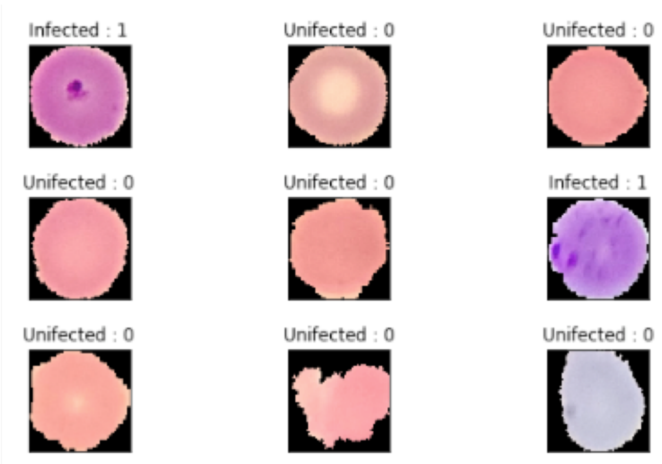


Figure 2: Enhanced dataset images

5 Method

We implemented three popular deep convolutional neural networks, VGG16, ResNet, Xception, along with one customized network in order to compare the performance. In the following subsections, each model would be briefly explained about its design and structures. Furthermore, for the purpose of comparison, we only used binary cross-entropy as the loss function, and the Adam optimizer as our optimization technique when reducing the loss on the loss function.

5.1 Model

5.1.1 CNN

Convolutional Neural Network, known as CNN, is the most typical method of deep learning. In 1980, the introduction of the neural sensor *neocognitron* marked the birth of the first initial convolutional neural network. It was also the first application of the receptive field concept in the field of artificial neural networks. The neural cognitive machine decomposed a visual pattern. A number of sub-patterns (features) are then entered into the layered hierarchically connected feature plane for processing. Then in 1988, the Shift-invariant neural network proposed to improve the function of the CNN, enabling it to be recognized even when the object is displaced or slightly deformed. Feed-forward convolutional neural network architecture is extended laterally and is connected in a feedback neural abstract pyramid (Neural abstraction pyramid) in. The resulting recurring convolution network allows for flexible incorporation of context information to iteratively resolve local ambiguities. In contrast to previous models, the highest resolution image output is produced. Finally, in 2005, a paper with GPU implementation of CNN appeared, which marked a more effective way to implement CNN. After the 2012 ImageNet contest, CNN stood out because of its high precision.

Figure 3 is our own CNN model. The specification is as Figure 4. We trained a basic CNN model using Keras. The result would be shown in next section.

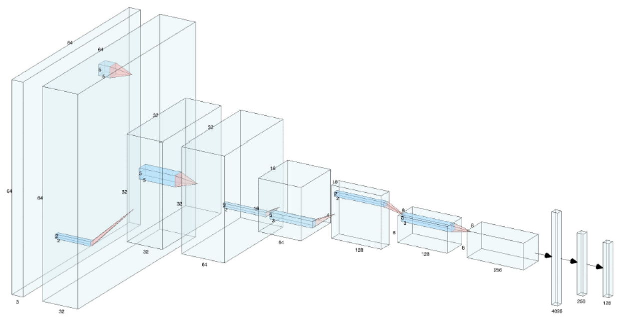


Figure 3: CNN model architecture

5.1.2 ResNet

ResNet (Residual Neural Network) was proposed by four Chinese scientists leading by Kaiming He from Microsoft Research Institute(6). Through the use of ResNet Unit, they successfully trained the 152-layer neural network and won the championship in ILSVRC2015. The error rate on top5 was 3.57%. At the same time, the parameter

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 60, 60, 32)	2432
max_pooling2d_1 (MaxPooling2	(None, 30, 30, 32)	0
conv2d_2 (Conv2D)	(None, 26, 26, 64)	51264
max_pooling2d_2 (MaxPooling2	(None, 13, 13, 64)	0
conv2d_3 (Conv2D)	(None, 11, 11, 128)	73856
max_pooling2d_3 (MaxPooling2	(None, 5, 5, 128)	0
conv2d_4 (Conv2D)	(None, 3, 3, 256)	295168
max_pooling2d_4 (MaxPooling2	(None, 1, 1, 256)	0
flatten_1 (Flatten)	(None, 256)	0
dense_1 (Dense)	(None, 256)	65792
dropout_1 (Dropout)	(None, 256)	0
dense_2 (Dense)	(None, 128)	32896
dropout_2 (Dropout)	(None, 128)	0
dense_3 (Dense)	(None, 1)	129

Total params: 521,537
Trainable params: 521,537
Non-trainable params: 0

Figure 4: CNN specifications

quantity is lower than VGGNet, and the effect is very outstanding. The structure of ResNet can accelerate the training of neural networks very quickly, and the accuracy of the model is also greatly improved. At the same time, ResNet is very popular and can even be used directly in the InceptionNet network.

The main idea of ResNet is to add a direct connection channel to the network as Figure 5 shows, which is the idea of the Highway Network. The previous network structure was a non-linear transformation of the performance inputs, while the Highway Network allowed a certain percentage of the output of the previous network layer to be preserved. The idea of ResNet is very similar to that of the Highway Network, allowing the original input to be passed directly to the later layers, as shown in the following figure.

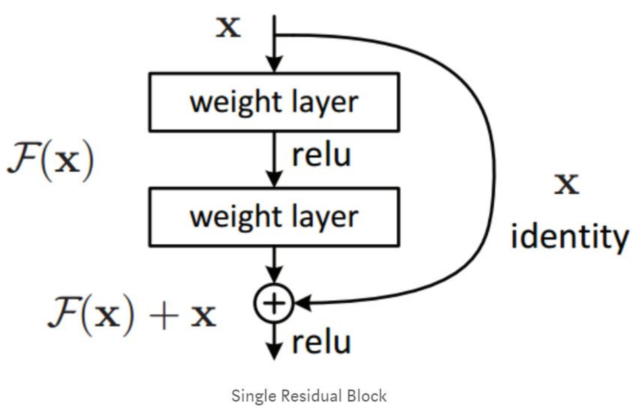


Figure 5: ResNet architecture

Traditional convolutional networks or fully connected networks have more or less information loss during information transmission, which also causes gradients to disappear or gradient explosions. These make it difficult to train deep network. ResNet solves this problem to a certain extent. By directly bypassing the input information to the output and protecting the integrity of the information, the entire network only needs to learn the part of the input and output differences, simplifying the learning objectives and difficulty.

5.1.3 VGG16

VGG16 is a convolutional neural network model proposed by K.Simonyan and A.Zisserman from the University of Oxford in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition"(7). The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes. It was one of the famous model submitted to ILSVRC-2014. It makes the improvement over AlexNet by replacing large kernel-sized filters (11 and 5 in the first and second convolutional layer, respectively) with multiple 3*3 kernel-sized filters one after another. VGG16 was trained for weeks and was using NVIDIA TITAN Black GPU's. Figure 6 shows the architecture of the VGG16 network.

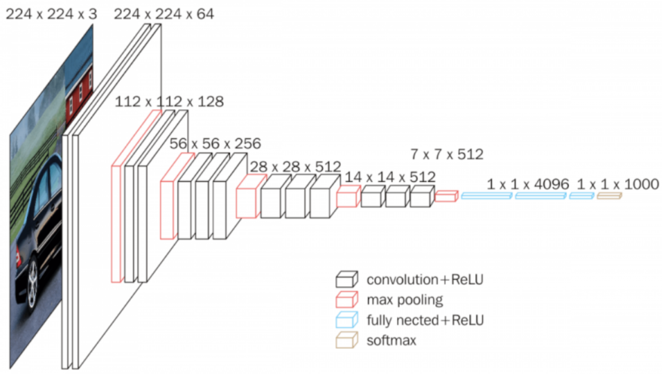


Figure 6: VGG16 architecture

5.1.4 Xception

Xception is a deep convolutional neural network architecture that involves Depthwise Separable Convolutions, developed by Google researchers(8). Google presented an interpretation of Inception modules in convolutional neural networks as being an intermediate step in-between regular convolution and the depthwise separable convolution operation (a depthwise convolution followed by a pointwise convolution). In this light, a depthwise separable convolution can be understood as an inception module with a maximally large number of towers. This observation leads them to propose a novel deep convolutional neural network architecture inspired by Inception when Inception modules have been replaced with depthwise separable convolutions. Xceptions models remain expensive to train, but are pretty good improvements compared with Inception. We implemented a Xception model in Keras, however, due to the very depth of the model, the training encountered with the overfitting problem.

5.2 Loss Function

For loss function, we choose the binary cross-entropy to train our model.

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i * \log(p(y_i)) + (1 - y_i) * \log(1 - p(y_i)) \tag{1}$$

5.3 Optimization

The goal of optimization is to reduce the loss function. The optimizer we chose is Adam optimization. The Adam optimization algorithm is an extension to stochastic gradient descent that has recently seen broader adoption for deep learning applications in computer vision and natural language processing. It can be used instead of the classical stochastic gradient descent procedure to update network weights iterative based in training data. Adam realizes the benefits of both AdaGrad and RMSProp to provide an optimization algorithm that can handle sparse gradients on noisy problems. It is also relatively easy to configure where the default configuration parameters do well on most problems. The default learning rate we chose is 0.0001.

6 Result

6.1 Overview

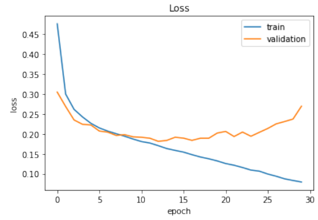
Model	Train Size	Test Size	Train Error	Test Error	Train Acc	Test Acc
CNN	70,548	22,047	0.0792	0.2794	97.23%	92.43%
VGG16	70,548	22,047	0.1062	0.1256	96.17%	95.35%
ResNet	70,548	22,047	0.0103	0.1504	99.65%	96.76%
Xception	70,548	22,047	0.0035	0.1410	99.89%	97.68%

The result is as the table shows. The training data size we chose is 70,548 and the testing data set we chose is 22,047. We can see the performance of each model based on its test error and accuracy. It is obvious to observe the difference in the performance. The Xception model is the best, reaching the accuracy of 99.89% on the training set and 97.68% on the testing set, while the CNN model we designed is relatively the worst with the accuracy of 97.23% on the training set, 92.43% on the testing set. However, it also can be noted that the over-fitting problem occurred in the ResNet model and the Xception model. We tried the dropout layer but the improvement was very limited. The confusion matrix and the graph would be demonstrated in the later subsections.

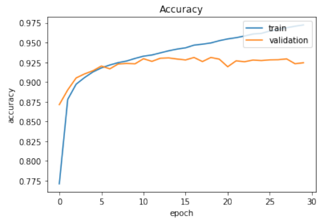
6.2 Confusion Matrix

6.2.1 CNN

The result of our own CNN model is shown in Figure 7. The confusion matrix for the CNN model is in Table 1. It performs good based on the graph and the matrix. And there is basically no overfitting problem according to the validation performance.



(a) loss



(b) accuracy

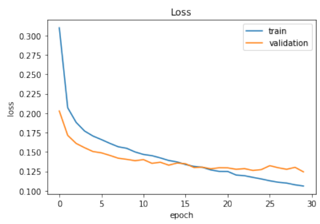
Figure 7: CNN result

	Predicted: NO	Predicted: YES
Actual: NO	TN = 10631	FP = 418
Actual: YES	FN = 1250	TP = 9742

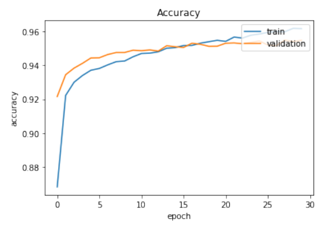
Table 1: CNN confusion matrix

6.2.2 VGG16

The result of VGG16 model is shown in Figure 8. The confusion matrix for VGG16 is in Table 2. From the confusion matrix we can see that VGG16 reduces the number of False-Negative largely, compared with the first CNN model. And there is barely overfitting problem occurred.



(a) loss



(b) accuracy

Figure 8: VGG16 result

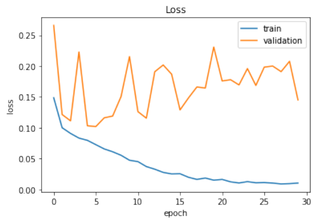
6.2.3 ResNet

The result of our own ResNet model is shown in Figure 9. The confusion matrix for the CNN model is in Table 3. From the graph, it can be obviously seen that the overfit-

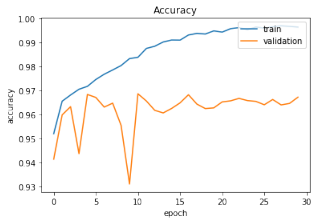
	Predicted: NO	**Predicted: YES**
Actual: NO	TN = 10497	FP = 556
Actual: YES	FN = 470	TP = 10524

Table 2: VGG16 confusion matrix

ting problem occurred, where the line of validation is very unstable but the performance on the training set is incredibly good. We believe it's the deep feature of the ResNet that caused the overfitting. We tried the dropout mechanism in the fully-connected layer but the performance was still not ideal. Compared with the VGG16 model, it did improve the performance on the testing set, including the reduction of False-Negative and False-Positive.



(a) loss



(b) accuracy

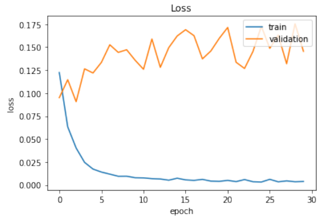
Figure 9: ResNet result

	Predicted: NO	**Predicted: YES**
Actual: NO	TN = 10710	FP = 346
Actual: YES	FN = 369	TP = 10622

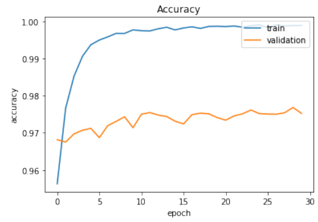
Table 3: ResNet confusion matrix

6.2.4 Xception

The result of our own Xception model is shown in Figure 10. The confusion matrix for the CNN model is in Table 4. Still, it is not difficult to find that the overfitting still occurred for this model since it's an extremely deep neural network. However, the validation performance shows that it's not as unstable as the ResNet model. To avoid overfitting, we also tried the dropout but the performance is not as good as expected. But compared with all former models, Xception reaches the highest accuracy on the training set and the testing set, yielding the largest number of True-Positive and True-Negative, as well as the smallest number of False-Positive and False-Negative.



(a) loss



(b) accuracy

Figure 10: Xception result

	Predicted: NO	**Predicted: YES**
Actual: NO	TN = 10894	FP = 179
Actual: YES	FN = 332	TP = 10642

Table 4: Xception confusion matrix

7 Conclusion

In this project, we implemented a customized convolutional neural network model and conducted several experiments of the latest CNN models on the problem of malaria cell image recognition, including VGG16, ResNet, and Xception invented by Google. These models have various features, leading to different performance. Generally speaking, deeper networks have better performance, for instance, ResNet and Xception. However, the very-deep feature indicates the higher probability of the problem of overfitting, which we encountered during our project when implementing ResNet and Xception. Even though, they still achieve way better performance than our initial CNN model (96.76% and 97.68%, compared with 92/43%). By applying advanced machine learning technique, the process of detecting and diagnosing malaria disease could be accelerated and boosted, which could be significant for poor areas.

8 Future Work

If we had more time for this project, there are many things we could still do to improve the performance and result. Since the major problem we encountered was the overfitting when training the deep networks, we would like to tackle this problem by doing more experiments. First of all, we would try different optimizer, learning rate, and loss function to see how the overfitting could be possibly affected. Secondly, we would apply more data augmentation techniques to enlarge the dataset because the larger data could reduce the fact of overfitting to some extent. Last but not least, finer regularization, as well as the dropout, should be done considering the current model.

Contribution

Fanfei Xu: Data preprocessing and augmentation, report writing.

Buyun Gao: CNN, ResNet training, report writing.

Chuping Wang: VGG16, Xception training, poster creating and report writing.

Wenquan Chen: Result collection and analysis, parameter tuning, report writing.

https://github.com/cpwang96/ECE228_FINAL_PROJECT

References

[1] A. S. B. Reddy and D. S. Juliet, "Transfer Learning with ResNet-50 for that Malaria Cell-Image Classification," 2019 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2019, pp. 0945-0949. doi: 10.1109/ICCSP.2019.8697909

[2] D. Anggraini, A.S. Nugroho, C. Pratama, I.E. Rozi, V. Pragesjvara, M. Gunawan, "Automated status identification of microscopic images obtained from malaria thin blood smears using Bayes decision: a study case in Plasmodium falciparum", International conference on advanced computer science and information system, IEEE (2011), pp. 347-352

[3] M. Poostchi, K. Silamut, R. J. Maude, S. Jaeger, G. Thoma, "Image analysis and machine learning for detecting malaria", Translational Research, Volume 194, 2018, Pages 36-55, ISSN 1931-5244

[4] S. Jaeger, Malaria Datasets, nih.gov, https://ceb.nlm.nih.gov/repositories/malaria-datasets/

[5] Y. Purwar, S.L. Shah, G. Clarke, A. Almugairi, A. Muehlenbachs Automated and unsupervised detection of malarial parasites in microscopic images Malar J, 10 (2011), p. 364

[6] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[7] Simonyan, K., Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[8] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1251-1258).