

Malaria Cell-Image Classification using InceptionV3 and SVM

Marada Amrutha Reddy
Dept of Computer Science and Engineering
GITAM University
Visakhapatnam, India

Ganti Sai Siva Rama Krishna
Dept of Computer Science and Engineering
GITAM University
Visakhapatnam, India

Teki Tanoj Kumar
Dept of Computer Science and Engineering
GITAM University
Visakhapatnam, India

Abstract— Malaria is a lethal illness spread by female anopheles mosquito bites. The bite carries parasites of the plasmodium group. In 2019, an estimated 229 million cases of malaria were reported globally, according to the WHO (World Health Organization). In the same year, an estimated 409 000 people died from malaria. If diagnosed early and treated promptly, the infection will not spread. The disease is diagnosed via a microscopic examination of the patient's blood sample. The sample is thinly spread as a smear, with the cell images serving as the visual criteria. Diagnosis is a time-consuming process that needs the assistance of an expert. To avoid incorrect findings caused by human error, various machine learning and deep learning methods have been developed. In this research, we built a transfer learning model using inception-v3 and SVM classifier. The achieved results show that a model with Inception-V3 as feature extractor and SVM classifier gave an accuracy score of 94.8 percent.

Keywords— Classification; Deep Learning; Transfer Learning; Inception-V; Support Vector Machine

I. INTRODUCTION

Malaria is a disease that affects the red blood cells and can be fatal. Female anopheles mosquitoes harbouring plasmodium parasites bite people and transmit the disease. It is a prevalent disease with a high risk of being fatal. It is regarded as a major health problem all over the world due to its lethal character. There are 5 different species of malaria which are capable of causing infection, two of which are dangerous: *P. falciparum* and *P. vivax*. In 2019, an estimated 229 million cases of malaria were reported globally, according to the WHO [1]. Children aged under 5 years are the most vulnerable group affected by malaria; in 2019, they accounted for 67% (274 000) of all malaria deaths worldwide [1]. The disease is diagnosed via microscopic examination of cell images which are collected via a patient's blood sample. The process of examination is carried out manually and is prone to human error. To avoid erroneous conclusions, various machine learning and deep learning models have been developed. Recently, convolutional neural network has proven to provide models with high accuracy scores for classifying images. Shah D. et al [2] proposed a CNN classifier using sigmoid activation function which acquired an accuracy score of 95%. Banerjee T et al [3] proposed a deep convolutional neural network known as Falcon to detect the parasitic cells from blood smeared slide images of Malaria Screener. The model acquired an accuracy score of

95.2 Vijayalakshmi A et al [4] proposed a methodology to classify malaria cell images using a VGG19-SVM model with 93.1 classification accuracy. Reddy, A. S. B. et al [5] built a model to classify the images using resnet-50 with a train accuracy score of 95.91%. In this work, we used a transfer learning model to extract features which are sent to the SVM classifier for classification. Transfer learning is quite popular in the fields of image processing and natural language processing. In transfer learning, a previously trained model on a humongous dataset with millions of instances is reused on a smaller dataset with similar properties. The use of previously learnt knowledge makes it computationally efficient. The proposed model unifies inception-v3 model that uses previously learnt weights to extract features. It is then unified with SVM classifier to classify the images.

II. DATA COLLECTION

There are 27,560 cell images in the dataset. Half of the photos are parasitized, while the other half are unaffected. Positive samples contained *Plasmodium* and negative samples contained no *Plasmodium* but other types of objects including staining artifacts/impurities [6]. This dataset can be downloaded from the National Library of Medicine's official website (NLM). It's also available in Kaggle and Tensorflow datasets. The images that are stored in the dataset are taken from patients who are chosen at random. An instance image of each class label namely parasitized and uninfected is shown below.

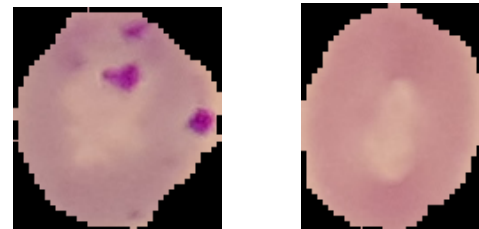


Fig. 1. Image of parasitized and uninfected cell

III. DEEP LEARNING

Deep learning models try to replicate or emulate how the human brain works. The fundamental building block of Deep Learning is the Perceptron which is a single neuron in a Neural Network [7]. It allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction [8]. To put it another way, deep learning uses artificial neural

networks to solve problems. Every artificial neural network contains an input layer, multiple hidden layers, and an output layer. There are nodes in each layer, and all of the nodes between the layers are linked and have weights. The figure below shows the fundamental structure of an artificial neural network:

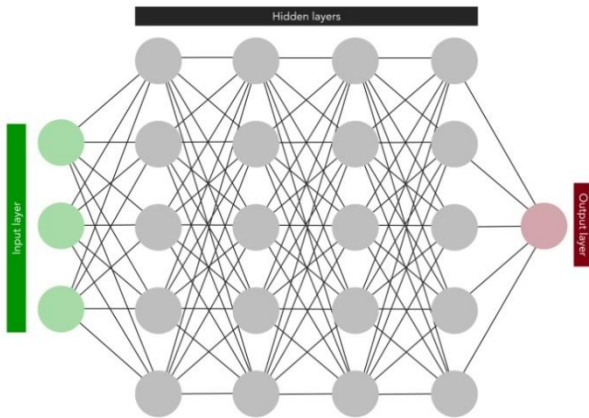


Fig. 2. Structure of an artificial neural network

Feed forward neural networks, convolutional neural networks and recurrent neural networks are the three most popular forms of neural networks. In a feed forward neural network, information flows in one direction along connecting pathways, from the input layer via the hidden layers to the final output layer [9]. Convolutional neural networks are used to process image data. It was inspired by biological processes in which the connectivity pattern between neurons resembles the organization of the animal visual cortex. [10]. Recurrent neural networks can keep the record of all the input values that are viewed by the network and also current input, which is value hidden at each layer network depend on all the previously seen inputs [11]. These are extremely useful when working with time series data. Deep learning models can be used in all types of learning namely supervised, unsupervised and reinforcement learning. Supervised learning entails learning a mapping between a set of input variables X and an output Variable Y and applying this mapping to predict the outputs for unseen data [12]. Regression and classification are included in supervised learning. Deep Learning is being applied in various domains for its ability to find patterns in data extract features and generate intermediate representations. [13]. It can be used in practically any field and aids in the achievement of superior results.

IV. TRANSFER LEARNING

Transfer learning is the reuse of a pretrained model on a new and similar problem. In transfer learning, a machine exploits the knowledge gained from a previous task to improve generalization about another but related task [14]. Transfer learning's key benefit is that it reduces training time and can produce better outcomes even when there isn't a lot of data. It is computationally efficient.

A. Inception-v3

Inception V3 is a Deep Convolutional Network and is Google's 3rd version of Deep Learning Architecture series and it is trained on a dataset containing 1000 different classes that are derived from the original ImageNet dataset that had 1million classes of images [15]. The building blocks of

inception v3 are convolutions, max pooling, concats, dropouts, fully connected layers and average pooling. Batch norm is also used throughout the model and applied to activation inputs [16]. It has 48 layers. The architecture of Inception v3 is depicted in the diagram below.

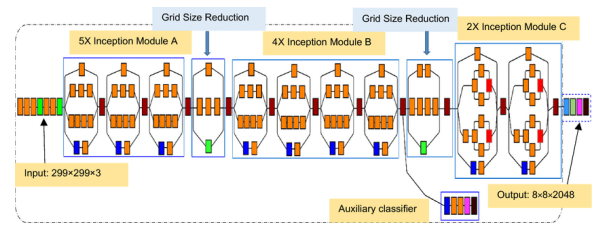


Fig. 3. Architecture of Inception-v3

V. SUPPORT VECTOR MACHINE (SVM)

The basic concept of SVM methods is to place an optimal class in the space of original attributes separating the hyperplane [17]. The support vector machine contains inputs x_1, x_2, \dots, x_i and the desired output y and set of weights used to the optimization of the maximizing margin but in most cases, it will be difficult to separate between classes by line so SVM uses mathematical functions are kernels which process mapping objects to transform the complex curve [18]. SVM can also be used in deep learning models as a classifier. This is done by using l2 kernel regularizer, linear and softmax activations with hinge and squared hinge as loss functions. If the classification problem has two class labels, activation is set to 'linear' and the loss to 'hinge'. For a multi-class classification problem, the activation is set to softmax and the loss is set to 'squared hinge'.

VI. PROPOSED MODEL

The proposed model receives images, produced by microscopic inspection of a patient's blood sample as input. Each image is of RGB (Red, Green, and Blue) color model. The images are augmented using imageDataGenerator and sent to the model. The model then detects if the cell image is parasitized or uninfected. The model is built using pre-trained weights and transfer learning as a feature extractor. The feature extraction is done by freezing the model up to the bottleneck layer and by omitting the last fully connected layer. The last layer is excluded to ensure that the feature extraction is generalized. We defined our own fully connected dense layer using an l2 regularized linear support vector machine classifier. The dense layer is then added on top of the pre-trained feature extractor. When using a support vector machine as a classifier, the activation functions used are linear and softmax. As the problem at hand has two class labels, linear activation function is used. Softmax is well-known for its ability to solve multi-class classification problems. We used the l2 regularization penalty on the weight vectors in the fully connected layer to avoid overfitting. The classifier is trained using hinge loss function. The proposed approach architecture is depicted below:

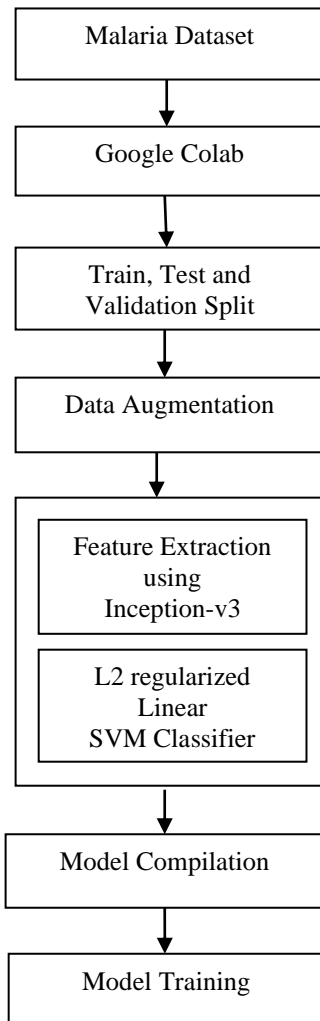


Fig. 4. Proposed model

VII. EXPERIMENT

The dataset is imported into Google Colab from Kaggle website. The images are loaded as a zip file and are unzipped. The environment used to build the model is set using Keras and Tensorflow 2.5.0. The experiment was carried out on a windows 10 computer with 8 GB RAM. To speed up the model training, we used Google Colab’s hardware accelerator.

Colab offers two different kinds of hardware accelerators namely GPU and TPU. The code was executed using GPU as the hardware accelerator.

A. Splitting the Data:

The Malaria dataset contains a total of 27,560 cell images with equal instances of parasitized and uninfected cells. It has 13779 parasitized and 13779 uninfected cell instances. The dataset is split into 70 train, 15 validation and 15 test set. There are 19290 cell images in train set after the split. The validation and test sets each have 4134 images. The sample images are shown below:

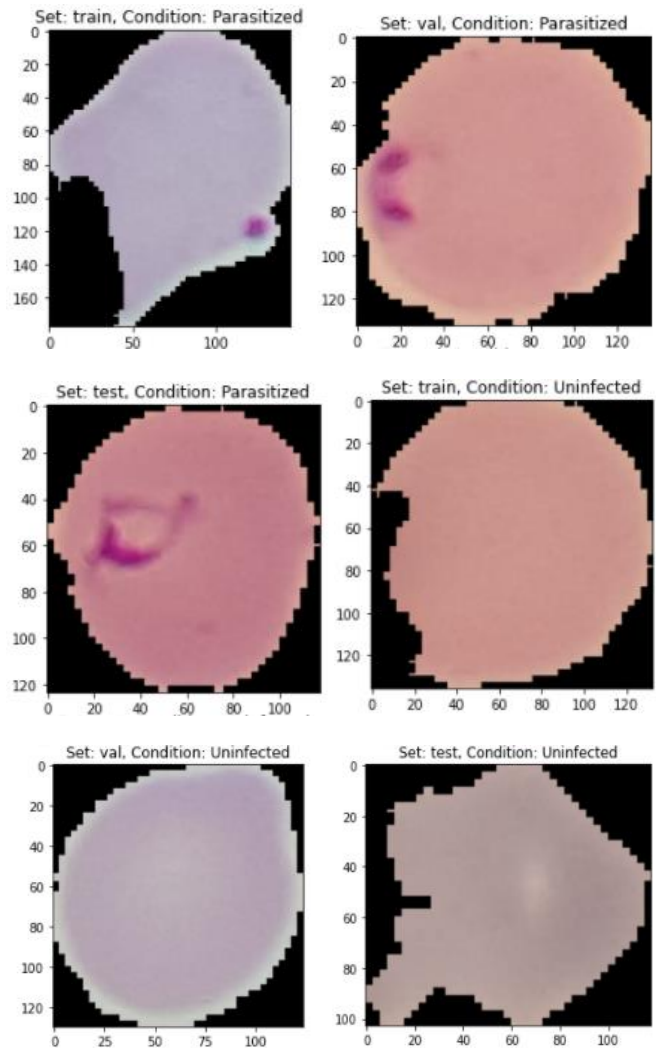


Fig. 5. Images of dataset split

B. Data Augmentation:

The images are augmented using the imageDataGenerator class provided by Keras. The image pixels are rescaled to a range of 0-1. The shear range and zoom range are both set to 0.2. The horizontal flip is set to true. When data augmentation is used, the images are transformed. It ensures that the images are taken at random in each batch and no same images are taken when the model is being trained.

C. Feature Extraction using Transfer Learning:

The default input image size for inception-v3 is 299*299 pixels. As the model is only being used for feature extraction, the size could be set to any square size which is not less than 75 pixels. In this study, the input image size is set to 150*150 pixels. The feature extraction is done using inception-v3 transfer learning model by freezing it till its bottleneck layer. The bottleneck layer is a layer which is right below the fully connected layer. The bottleneck layer for inception v3 is mixed7. The model is frozen with pre-trained weights and omitted the last dense layer. The model summary of inception-v3 as feature extractor is shown below:

mixed9_1 (Concatenate)	(None, 3, 3, 768)	0	activation_87[0][0] activation_88[0][0]
concatenate_1 (Concatenate)	(None, 3, 3, 768)	0	activation_91[0][0] activation_92[0][0]
activation_93 (Activation)	(None, 3, 3, 192)	0	batch_normalization_93[0][0]
mixed10 (Concatenate)	(None, 3, 3, 2048)	0	activation_85[0][0] mixed9_1[0][0] concatenate_1[0][0] activation_93[0][0]

 Total params: 21,802,784
 Trainable params: 0
 Non-trainable params: 21,802,784

Fig. 6. Model summary of Inception-V3 as feature extractor

D. Adding SVM classifier:

A fully connected layer is built using L2-regularizer linear SVM classifier. In inception v3, the fully connected layer has 1024 hidden units and uses relu activation function. A dropout value of 0.7 is used to regularize the layer. A final dense layer with SVM classifier is then added to the model. The Keras implementation of SVM uses a kernel regularizer. The default value of kernel regularizer is 0.01. We conducted the experiment with kernel regularizer set to 0.01. The activation function is set to linear as the problem at hand has two class labels. The fully connected layer with SVM classifier and the model summary are shown below:

```
# Flatten the output layer to 1 dimension
x = layers.Flatten()(last_output)
# Add a fully connected layer with 1,024 hidden units and ReLU activation
x = layers.Dense(1024, activation='relu')(x)
# Add a dropout rate of 0.7
x = layers.Dropout(0.7)(x)
# Add a final SVM layer for binary classification
x = layers.Dense(1, kernel_regularizer = l2(0.01), activation='linear')(x)
# Configure and compile the model
model = Model(pre_trained_model.input, x)
model.compile(loss='hinge',
              optimizer= 'adam',
              metrics=['acc'])
```

Fig. 7. Fully connected dense layer with SVM classifier

flatten (Flatten)	(None, 37632)	0	mixed7[0][0]
dense (Dense)	(None, 1024)	38536192	flatten[0][0]
dropout (Dropout)	(None, 1024)	0	dense[0][0]
dense_1 (Dense)	(None, 1)	1025	dropout[0][0]

 Total params: 47,512,481
 Trainable params: 38,537,217
 Non-trainable params: 8,975,264

Fig. 8. Model summary of Inception V3 with SVM classifier

E. Model compilation and training:

The model is compiled with loss, optimizer and metrics as parameters. The loss is set to hinge loss with adam optimizer. The metrics used is accuracy. To train the model, the batch size is set to 64. The model went through 50 epochs. The steps per epoch are 302.

VIII. RESULTS

Each epoch took 103 seconds of time for execution. The model was trained through 50 epochs. Within 10 epochs, the model reached a training accuracy of 93 percent. The training accuracy and validation accuracy had a stable increase with each epoch and converged during the later epochs. We recorded the accuracy and loss during each epoch of the training process. It is observed that the train and validation accuracy values at each epoch are nearly close to one another. The training loss is slightly higher than the validation less. With each epoch, the train loss and the validation loss gradually decreased. The accuracy and loss graphs are shown below.

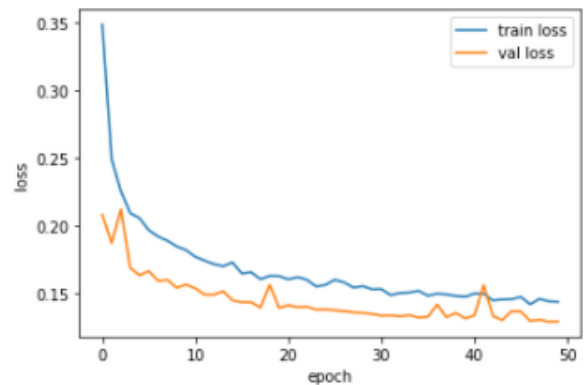


Fig. 9. Loss Graph

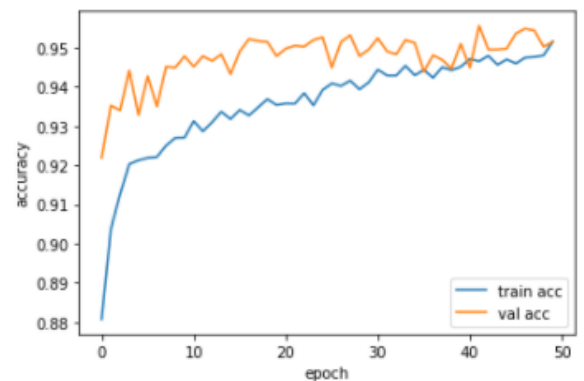


Fig. 10. Accuracy Graph

We have tabulated the train accuracy, validation accuracy, train loss and validation loss obtained after 50 epochs with the SVM classifier below:

TABLE I. EVALUATION SCORES

Model	Metrics		
	Dataset-split	Loss	Accuracy
Inception-v3 with SVM classifier	Training Dataset	0.1294	95.15
	Validation Dataset	0.1367	95.14

XI. CONCLUSION

In this study, we were able to successfully build a transfer learning model using inception-v3 and SVM classifier to detect Malaria. The test accuracy of the model is 94.8. The accuracy of the model can be further improved by training the model for more number of epochs, hyper tuning the

parameters and fine tuning the model. Even with high number of epochs, SVM classifier was faster to execute. Other than inception-v3, maybe using VGG19 for feature extraction might yield better accuracy.

REFERENCES

- [1] <https://www.who.int/news-room/fact-sheets/detail/malaria>
- [2] Shah, D., Kawale, K., Shah, M., Randive, S., & Mapari, R. (2020, May). Malaria Parasite Detection Using Deep Learning:(Beneficial to humankind). In *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)* (pp. 984-988). IEEE.
- [3] Banerjee, T., Jain, A., Sethuraman, S. C., Satapathy, S. C., Karthikeyan, S., & Jubilson, A. (2021). Deep Convolutional Neural Network (Falcon) and transfer learning-based approach to detect malarial parasite. *Multimedia Tools and Applications*, 1-15.
- [4] Vijayalakshmi, A. (2020). Deep learning approach to detect malaria from microscopic images. *Multimedia Tools and Applications*, 79(21), 15297-15317.
- [5] Reddy, A. S. B., & Juliet, D. S. (2019, April). Transfer learning with ResNet-50 for malaria cell-image classification. In *2019 International Conference on Communication and Signal Processing (ICCS)* (pp. 0945-0949). IEEE.
- [6] Rajaraman S, Antani SK, Poostchi M, Silamut K, Hossain MA, Maude RJ, Jaeger S, Thoma GR. 2018. Pre-trained convolutional neural networks as feature extractors toward improved malaria parasite detection in thin blood smear images. *PeerJ* 6:e4568 <https://doi.org/10.7717/peerj.4568>
- [7] Pamina, J., & Raja, B. (2019). Survey on deep learning algorithms. *International Journal of Emerging Technology and Innovative Engineering*, 5(1).
- [8] İçöz, B., Karayaglı, S., & Kanlıtepe, S. DEEP LEARNING. Ankara Yıldırım Beyazıt University.
- [9] Jha, G. K. (2007). Artificial neural networks and its applications. *IARI, New Delhi*, girish_iasri@rediffmail.com.
- [10] Deepthi, T. H., Gaayathri, R., Shanthosh, S., Gebin, A. S., & Nithya, R. A. (2019). Firearm Recognition Using Convolutional Neural Network.
- [11] Thomas, M., & Latha, C. A. (2018). Sentimental analysis using recurrent neural network. *International Journal of Engineering & Technology*, 7(2.27), 88-92.
- [12] Cunningham, P., Cord, M., & Delany, S. J. (2008). Supervised learning. In *Machine learning techniques for multimedia* (pp. 21-49). Springer, Berlin, Heidelberg.
- [13] Sk, S., Jabez, J., & Anu, V. M. (2017). The Power of Deep Learning Models: Applications. *Networks*, 33.
- [14] Ajagunsegun, T., & Kaur, P. (2021) "Detection of Motorcyclists without Helmet using Convolutional Neural Networks", *International Journal of Advanced Trends in Computer Applications (IJATCA)*
- [15] Nelli, A, Nalige, K., Abraham, R., & Manohar. R.(2020). "Landmark Recognition using Inception-v3", *International Research Journal of Engineering and Technology (IRJET)*, 7(5)
- [16] Chugh, G., Sharma, A., Choudhary, P., & Khanna, R. (2020). "Potato Leaf Disease Detection Using Inception V3", *International Research Journal of Engineering and Technology (IRJET)*, 7(1).
- [17] Publication, I. A. E. M. E. (2020). Performance Analysis Of Dimensionality Reduction Techniques In Eeg Signal Classification. IAEME Publication. <https://doi.org/10.34218/IJARET.11.10.2020.120>
- [18] Publication, I. A. E. M. E. (2021). USING SUPPORT VECTOR MACHINE (SVM) FOR TIME AND COST OVERRUN ANALYSIS IN CONSTRUCTION PROJECTS IN EGYPT. IAEME Publications. <https://doi.org/10.34218/IJCIET.12.3.2021.002>