IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

COMPARISON OF CONVOLUTION NEURAL NETWORK ARCHITECTURES FOR ACUTE LYMPHOBLASTIC LEUKEMIA DETECTION

Vibhav L. Dhuri, Dr. Nitesh B. Guinde
Student, Associate Professor
Electronics and Telecommunication Engineering,
Goa College of Engineering, Goa, India

Abstract: Acute Lymphoblastic Leukemia (ALL) is the most widely recognized malignant growth in youngsters and grown- ups. The identification of leukemia in the prior stage is significant before it spreads into the circulatory systems and other fundamental organs. For a considerable length of time, the finding of leukemia has been finished by experienced administrators and it is a tedious undertaking for pathologists. Programmed recognition of Acute lymphoblastic leukemia disease in tiny pictures is difficult because of the complicated structures. To handle this issue, a programmed and powerful indicative framework is required for early identification and treatment. Right now, machine learning-based implementation of Transfer learning for classification is proposed to recognize leukemic cells and ordinary cells. The purpose of this paper is to compare three types of Convolution Neural Network (CNN) architectures and to find which of them is suitable for our detection application. These architectures are MobileNet, VGG16 and ResNet50. All the methods are done using Google Colab which is a free cloud service to create machine learning model, Tensorflow a machine learning library, openCV, keras and other libraries to accurately detect Acute Lymphoblastic Leukemia.

Index Terms—Machine learning; Acute Lymphoblastic Leukemia (ALL); Google Colab; Tensorflow; openCV; keras; CNN; MobileNet, VGG16, ResNet50,transfer learning.

I. INTRODUCTION

Cancer begins when cells of body start to grow quickly. The leukemia cells begin in bone marrow and grow in the blood cells and cause deadly infection. Symptoms are same as in other common diseases and hence diagnosis is difficult. Principally, there exist 4 kinds of leukemia which are Acute Lymphoblastic Leukemia (ALL), Acute Myeloid Leukemia (AML), Chronic Lymphocytic Leukemia (CLL) and Chronic Myeloid Leukemia (CML). Machine learning offers chances to progress diagnosis. ALL creates countless non matured white blood cells in bone marrow. There are many errors in detection of ALL by the pathologists when there is a huge amount of workload or if they work for long hours, it’s human tendency to make errors if they work for a long hours. But it is not the same in case of computers. To handle this issue we need to find an accurate and fast model for the ALL cells detection.

In reference paper [1] it shows one diagnostic procedure for ALL which is microscopic inspection of peripheral blood. In reference paper [2] their system has improved the classification accuracy .It uses openCV and ski image for image processing to extract relevant features from blood image .Classification is carried out using various classifiers: CNN, FNN, SVM and KNN. CNN gives the highest accuracy of 98.33%. Similarly, reference paper [3] presents that the running times highly depend on input data size. From obtained results GPU speedups can be expected for most of the algorithms from 3.6 times to 15 times. The Reference paper [6] concluded that it is faster to train a CNN on Colaboratory’s accelerated runtime than using 20 physical cores of a Linux server. Here it is possible to use GPU without paying or setting up for an expensive service. Colab’s performance is equivalent to dedicated hardware, given similar resources.

In this paper, earlier architectures are used to classify ALL cells and Normal (NML) cells, which will help in comparing the CNN architectures performance. Thus it will help in selecting an automatic-system (architecture) that will aid in the detection of ALL cancer and this will be very useful to Experts in medical field. The rest of the paper is organized as follows: section II gives the literature review; section III focuses on basic introduction about CNN; section IV gives basic information about transfer learning and the architectures used for comparison; section V demonstrates the methodology used to detect cancerous cells and information about the dataset; section VI shows the results for the compared models and finally conclusions are drawn in section VII.

II. LITERATURE REVIEW

Reference paper [16] is based on ResNet50 architecture in which the performance of residual networks was observed. The models were trained on the 1.28 million training images, and evaluated on the 50,000 validation images. The performance of the networks was evaluated using top-1 error rate. ResNet50 has decreased error rate for deeper networks and also alleviates the effect of vanishing gradient problem.

Reference paper [17] is based on MobileNets which use depthwise separable convolution. In this work some of the important design decisions leading to an efficient model was investigated and then demonstrated on how to build smaller and faster MobileNets using width multiplier and resolution multiplier by trading off a reasonable amount of accuracy to reduce size and latency. They further compared different MobileNets to popular models demonstrating superior size, speed and accuracy characteristics.

Reference paper [18] is based on VGG16 which experimented with different level of depths in convolutional neural networks (ConvNets). They presented the improvement in error reduction with respect to the depth. More non-linear rectification layers (ReLUs) makes the decision function more discriminative. Though depth was increased, the number of parameters in ConvNets was not greater than previously proposed shallow networks.

In reference paper [4], the result demonstrates that the loss value of the Adam and RMSProp optimizers was lower than the Adagrad and Proximal Adagrad optimizers. The classification accuracy using Adam optimizer is 95.8. In reference paper [5], Tensorflow was used which is one of the most popular libraries to classify MNIST dataset. In this paper the effects of multiple activation functions on classification results was compared. The functions used were Rectified Linear Unit (ReLu), Hyperbolic Tangent (tanH), Exponential Linear Unit (eLu), sigmoid, soft plus and soft sign. Based on the advantages, the above three models were selected for comparison for application of detecting ALL cancerous cells.

III. CONVOLUTION NEURAL NETWORKS

Convolution Neural Networks consists of many layers to solve the classification problem. The Fig. 1 shows CNN layers. The different layers are:

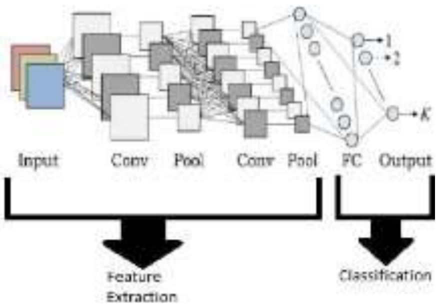


Figure 1 convolution neural network layers

A. Input

This layer takes the input image and gives it to other layers for further processing of the image to other layers.

B. Convolution

This layers has some filters which are then multiplied (convolution process) with each pixel block of the image to find the features of the image. This process helps in the testing phase.

C. ReLU (rectified linear unit)

This function converts negative outputs of convolution layer to zero which helps in speedy training.

D. Pooling

It reduces image size and the parameters hence only important parameters and large values from each window are taken.

E. Fully Connected (FC)

These are the layers where all the inputs from one layer are connected to every activation unit of the next layer.

F. Softmax

This is second last layer which gives probabilities between 0-1 for each class.

IV. TRANSFER LEARNING

Transfer learning is the idea of using a pre-trained model and its weight and utilizing knowledge acquired by the above model to solve related tasks as shown in Fig. 2.

Figure 2 difference between traditional ML and transfer learning

The Reasons for using pre-trained models: It takes huge amount of time for training a CNN from scratch and hence by using a pre-trained model the learning process is improved. Generally, it requires large computation power for processing huge data's. Pre-trained model used for solving problem is ResNet50.These models were downloaded and integrated with some new models or functions which would take input images and give better output results as compared to simple Convolution Neural Networks (CNN).

A. VGG16

VGG16 is CNN architecture which won first place in image localization task and second in Image classification at Image Net Large Scale Visual Recognition Challenge (ILSVR) competition in 2014.Instead of having large number of hyper parameter it focused on having convolution layer of 3×3 filters with stride of 1 and same padding and 2×2 max pooling layers. At the end it has 2 FC layers followed by a softmax for output. Input image requirement was of size 224×224×3(RGB). There are approximately 138 million parameters. Fig. 3 shows VGG16 architecture.

Figure 3 VGG16 architecture

B. Microsoft ResNet Model

Main motivation of Residual Network (ResNet) is the skip connection as shown in Fig. 4.

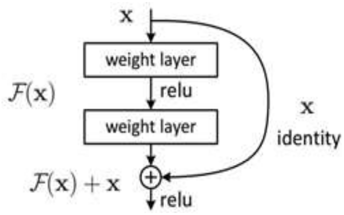


Figure 4 ResNet building block

Deep neural network often suffer from vanishing gradient problem i.e. as the model back propagates the gradient gets smaller and smaller. Tiny gradients can make learning intractable.

The skip connection in the Fig. 4 is labeled as x. It allows network to learn the identity function, which allows it to pass the input through the block without passing through the other weight layers. This allows it to stack additional layers and build deep neural networks. ReLU activations are used in ResNet. ResNet50 has 50 layers and has won first place in ILSVRC 2015 classification competition with top error rate of 3.57%.

The input data is trained on pre-trained weights and layers learning through back propagation are dense and unfrozen layers (fine-tuned layers). Few layers, for example in case of ResNet50 such as the batch normalization layers are fine-tuned because this will help the mean and variance of the dataset match the mean and variance from the pre-trained weights. The activation function used is softmax and it performs better on classification task.

C. MobileNet

It has lightweight architecture and uses depthwise separable convolutions which basically means it performs single convolution on each color channel rather than combining all three and flattening. Separate single filters are used for each input channel. The pointwise convolution then applies a 1×1 convolution to combine the outputs of the depth wise convolution. This drastically helps to reduce computation and model size. There are 30 layers in MobileNet. It has low maintenance, therefore performs with high speed. It has Size is 17mb. The Model is shown in Fig. 5. Its convolution layer has stride 2. It has depthwise layer with stride 2. Its pointwise layer doubles the number of channels.

Type / Stride	Filter Shape	Input Size
Conv / s2	3 × 3 × 3 × 32	224 × 224 × 3
Conv dw / s1	3 × 3 × 32 dw	112 × 112 × 32
Conv / s1	1 × 1 × 32 × 64	112 × 112 × 32
Conv dw / s2	3 × 3 × 64 dw	112 × 112 × 64
Conv / s1	1 × 1 × 64 × 128	56 × 56 × 64
Conv dw / s1	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 128	56 × 56 × 128
Conv dw / s2	3 × 3 × 128 dw	56 × 56 × 128
Conv / s1	1 × 1 × 128 × 256	28 × 28 × 128
Conv dw / s1	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 256	28 × 28 × 256
Conv dw / s2	3 × 3 × 256 dw	28 × 28 × 256
Conv / s1	1 × 1 × 256 × 512	14 × 14 × 256
[illegible]	[illegible]	[illegible] × 512
[illegible]	[illegible]	[illegible] × 512
[illegible]	[illegible]	[illegible] × 512
Conv / s1	1 × 1 × 512 × 1024	7 × 7 × 512
Conv dw / s2	3 × 3 × 1024 dw	7 × 7 × 1024
Conv / s1	1 × 1 × 1024 × 1024	7 × 7 × 1024
Avg Pool / s1	Pool 7 × 7	7 × 7 × 1024
FC / s1	1024 × 1000	1 × 1 × 1024
Softmax / s1	Classifier	1 × 1 × 1000

Figure 5 MobileNet architecture

V. METHODOLOGY

The methodology used to classify the cancerous cells from non-cancerous cells is given in the Models and, the in detail description about the methodology and data is given in the Experiment.

A. Models

The inputs for models are RGB with size 224×224.The image will enter the models with pre-trained weights and the last layer has a FC layer of softmax activation for classification. Fig. 6 shows the experiment flow.

B. Experiment

In our experiment every model uses Google colab with an environment of keras 2.3.1 with Tensorflow 2.2.0. Online colab GPU used for our models is Nvidia K80s.

C. Data Collection

The dataset contains of 6,176 images of both cancerous and non-cancerous cells. The dataset is available on official website of The Cancer Imaging Archieve (C-NMC) dataset. The difference between cancerous and non-cancerous cell is shown in Fig. 7. Dataset was already segmented.

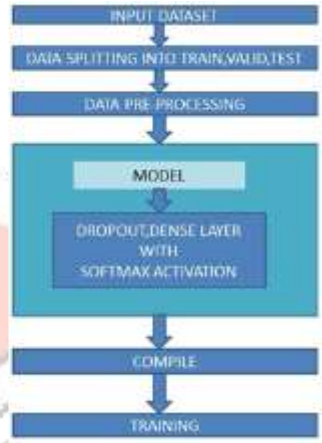


Figure 6 Model flow

D. Splitting Data

Total number of cancerous and non-cancerous cell images in dataset was 6,176. The split was done with 97.7, 1.13 and 1.13% for train, valid and test respectively. After splitting there were 6036 images for cancerous and non-cancerous cell images with 3018 for each class, 70 images for validation with 35 for each class and 70 images for test with 35 for each class.

E. Data Preprocessing

The required size of input image for our models was 224×224. Therefore images were resized to 224×224. All preprocessing function such labeling the data set, resizing, shuffling were done using image generator and flow from directory functions from keras. The batch size chosen was 32 and hence steps per epoch were selected to 189. We get steps per epoch by just dividing the total number of training samples divided by the batch size. For compiling the model, the optimizer, loss function and the metrics have to be calculated. These parameters help the model to work well on the data. Optimizer sets the learning rate of a neural network [13]. The optimizer used for these models is Stochastic Gradient Descent (SGD). SGD optimizer performs better than many other optimizers [14]. Choosing a loss function is a difficult task [12]. The loss function used to find the loss for our models is Categorical-Cross Entropy, with learning rate of 0.0001 and momentum for SGD optimizer set 0.9.

F. Training the Model

Neural Network learns through back propagation for reducing the output error. In case of transfer learning, pre-trained models top layers are not frozen therefore they learn through back propagation, whereas other layers are frozen. Weights get updated during back propagation called as fine tuning [13]. The number of epochs for training is 100.

Figure 7 Image of cancerous and non-cancerous cells

global_average_pooling2d_1 (	(None, 1024)	0
dropout_1 (Dropout)	(None, 1024)	0
dense_1 (Dense)	(None, 2048)	2099200
dropout_2 (Dropout)	(None, 2048)	0
dense_2 (Dense)	(None, 2)	4098

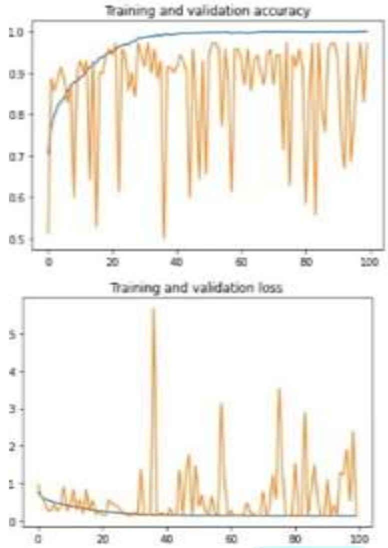
Total params: 5,332,162
Trainable params: 5,310,274
Non-trainable params: 21,888

VI. RESULT

A. MobileNet

The model which was used for MobileNet is shown in Fig. 8. Here only the later half of the model is given i.e. the added layers.

Fig. 9 shows the training accuracy and validation accuracy and also training loss and validation loss. Here we got training accuracy of 0.99 and validation accuracy of 0.97.Training loss of 0.13 and validation loss of 0.15was obtained. Fig. 10 gives the output result for the testing data. This is the confusion matrix.



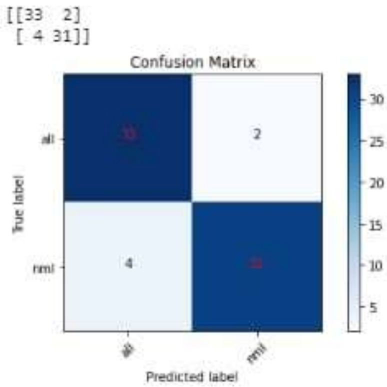


Figure 10 Confusion matrix of MobileNet

Figure 8 MobileNet model

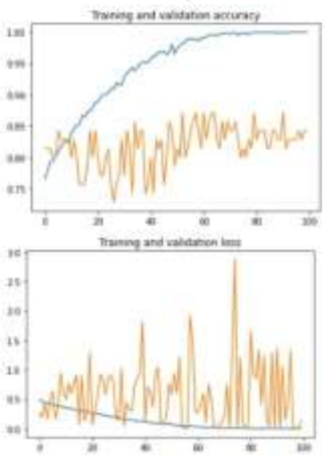
Layer (type)	Output Shape	Param #
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2359808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2359808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2359808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
fc1 (Dense)	(None, 4096)	102764544
fc2 (Dense)	(None, 4096)	16781312
dropout_1 (Dropout)	(None, 4096)	0
dense_1 (Dense)	(None, 2048)	8390656
dropout_2 (Dropout)	(None, 2048)	0
dense_2 (Dense)	(None, 2)	4098

Total params: 142,655,298
Trainable params: 127,940,610
Non-trainable params: 14,714,688

B. VGG16

Fig. 11 shows the used model for VGG16. Fig. 12 shows the training accuracy and validation accuracy and also training loss and validation loss. Here we got training accuracy of 1 and validation accuracy of 0.84.Traning loss of 0.0023 and validation loss of 0.12 was obtained. Fig. 13 gives the output result for the testing data .This is the confusion matrix.

Figure 9 Accuracies and losses of MobileNet



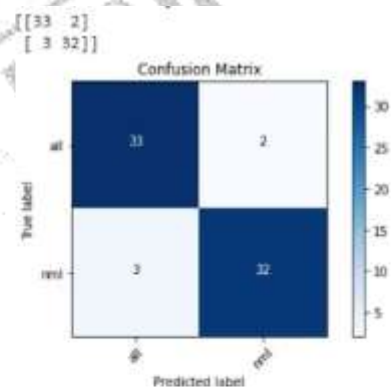


Figure 11 VGG16 model

Figure 12 Accuracies and losses of VGG16

Figure 13 Confusion matrix VGG16

C. ResNet50

The model which was used for ResNet50 is shown in Fig. 14. Here in below Figure only the added layers for information purpose was given. Fig. 15 shows the training accuracy and validation accuracy and also training loss and validation loss. Here we got training accuracy of 0.84 and validation accuracy of 0.60 .Training loss of 0.35 and validation loss of 0.70 was obtained. Fig. 16 gives the output result for the testing data. This is the confusion matrix. Results based on final testing accuracy on the testing data were done and tabulated in table 1.These results were obtained from the confusion matrix by dividing the correct predictions by the total number of test samples.

Figure 14 ResNet50 model

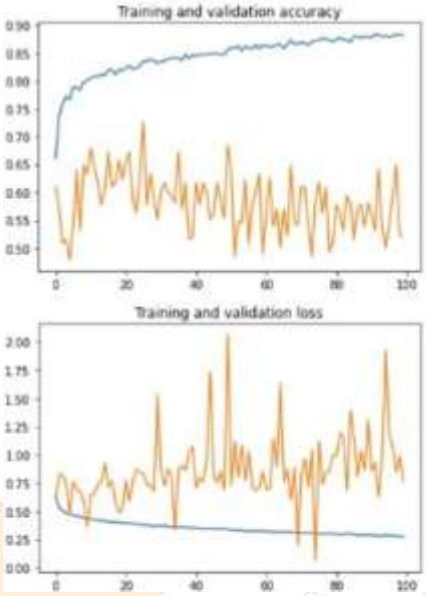


Figure 15 Accuracies and losses ResNet50

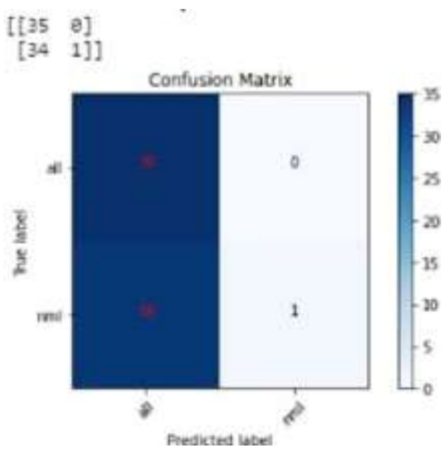


Figure 16 Output result for the testing data

VII. CONCLUSION

In our project we created a model for detection of Acute Lymphoblastic Leukemia with accuracy of 92.8%. We used different architectures of CNN in tensor flow using Google colab. So we found from table 1 that VGG16 gives a great accuracy for detection of ALL cancer out of the other models. Although the VGG16 model has large in size as compared to MobileNet but the main aim is to detect the cancerous cells with minimum error, so the slight increase in accuracy is also taken into consideration. Further other pre-trained models such as Inception, AlexNet, NasaNet can be used to find whether there is an increase in accuracy for ALL detection. Also large and cleaner dataset can be utilized to improve on the accuracy factor.

Table 1 Comparison for methods on same testing data

Method	Test accuracy
MobileNet	91.4%
VGG16	92.8%
ResNet50	51.4%

REFERENCES

[1]Labati, R. D., Piuri, V., & Scotti, F. (2011). ”All-IDB: The acute lymphoblastic leukemia image database for image processing.” 2011 18th IEEE International Conference on Image Processing.

[2]Rajpurohit, S., Patil, S., Choudhary, N., Gavasane, S., & Kosamkar,
P. (2018). Identification of Acute Lymphoblastic Leukemia in Micro- scopic Blood Image Using Image Processing and Machine Learning Algorithms. 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI) .

[3]Demirovic, D., Skejic, E., & Serifovic-Trbalic, A. (2018). “Performance of Some Image Processing Algorithms in Tensorflow”. 2018 25th International Conference on Systems, Signals and Image Processing (IWSSIP).

[4]Arwa Mohammed Taqi ,Ahmed Awad , Fadwa Al-Azzo , Mariofanna Milanova.”The Impact of Multi-optimizers and Data Augmentation on TensorFlow Convolutional Neural Network Performance.” 2018 IEEE Conference on Multimedia Information Processing and Retrieval.

[5]F. Ertam and G. Aydin,”Data classification with deep learning using Tensorflow,” 2017 International Conference on Computer Science and Engineering (UBMK),Antalya, pp.755-758.

[6]T. Carneiro, R. V. Medeiros Da N o´Brega, T. Nepomuceno, G. Bian, V. H.
C. De Albuquerque and P. P. R. Filho, ”Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications,” in IEEE Access, vol. 6, pp. 61677-61685, 2018.

[7]F. Ertam and G. Aydın, ”Data classification with deep learning using Tensorflow,” 2017 International Conference on Computer Science and Engineering (UBMK), Antalya, 2017, pp. 755-758.

[8]N. R. Gavai, Y. A. Jakhade, S. A. Tribhuvan and R. Bhattad, ”Mo- bileNets for flower classification using TensorFlow,” 2017 International Conference on Big Data, IoT and Data Science (BID), Pune, 2017, pp. 154-158.

[9]Xiaoling Xia, Cui Xu and Bing Nan, ”Inception-v3 for flower clas- sification,” 2017 2nd International Conference on Image, Vision and Computing (ICIVC), Chengdu, 2017, pp. 783-787.

[10]D. P. Mishra, T. K. Tripathy and S. Chakraborty, ”CNN for Butterfly Classification,” 2018 International Conference on Recent Innovations in Electrical, Electronics & Communication Engineering (ICRIEECE), Bhubaneswar, India, 2018, pp. 200-205.

[11]A. D. Andronescu, D. I. Nastac and G. S. Tiplica, ”Skin Anomaly Detection Using Classification Algorithms,” 2019 IEEE 25th Interna- tional Symposium for Design and Technology in Electronic Packaging (SIITME), Cluj-Napoca, Romania, 2019, pp. 299-303.

[12]Rosasco L, Vito ED, Caponnetto A, Piana M, Verri A. Are loss functions all the same?. Neural Computation. 2004 May 1;16(5):1063-76.

[13]Reyes AK, Caicedo JC, Camargo JE. Fine-tuning Deep Convolutional Networks for Plant Recognition. CLEF (Working Notes). 2015 Sep 8;1391.

[14]Keskar NS, Socher R. Improving generalization performance by switching from adam to sgd. arXiv preprint arXiv:1712.07628. 2017 Dec 20 [15]Pascanu R , Mikolov T. Understanding the exploding gradient problem.
CoRR,abs/1211.5063.2012 Nov;2.

[16]K. He, X. Zhang, S. Ren and J. Sun, ”Deep Residual Learning for Image Recognition,” 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[17]@mischoward2017mobilenets,title=MobileNets: Efficient Convolutional Neural Networks forMobile Vision Applications,author=Andrew G. Howard and Menglong Zhu and Bo Chen and Dmitry Kalenichenko and Weijun Wang and Tobias Weyand and Marco Andreetto and Hartwig Adam, year=2017,eprint=1704.04861,archivePrefix=arXiv,primaryClass=cs.CV

[18] @misc{simonyan2014deep, title={Very Deep Convolutional Networks for Large-Scale Image Recognition},
author={Karen Simonyan and Andrew Zisserman},
year={2014},eprint={1409.1556},archivePrefix={arXiv},primaryClass={cs.CV}}