2022 3rd International Conference on Electronic Communication and Artificial Intelligence (IWECAI) | 978-1-6654-7997-4/22/$31.00 ©2022 IEEE | DOI: 10.1109/IWECAI55315.2022.00087

Multi-CNN models with Pretraining for Binary Classification in Skin Cancer

Zhaojun Guo†
Data Science and Artificial intelligence
Ecole Supérieure d'Ingénieurs Léonard de Vinci
Paris, FRANCE
zhaojun.guo@edu.devinci.fr

Haobo Xu†,*
Viterbi School of Engineering
University of Southern California
Los Angeles, CA, 90007
*Corresponding author: haobox@usc.edu

Tianhao Yao†
UMich Joint Institute
Shanghai Jiao Tong University
Shanghai, China, 200240
yth990922@sjtu.edu.cn
†These authors contributed equally.

Abstract— **In that a myriad of people died because of skin cancer contemporarily and their diversity makes the distinguishing malignant or benign skin cancer even intricate, classifying them with small dataset training would be more challenging. In this paper, obtaining a higher accuracy of the binary classification problem for skin cancer by applying employed models with or without pretraining was tried. 3 proposed models were proposed and mainly popular models such as Inception3, Xception, VGG16, MobileNet, ResNet etc. Moreover, some popular models pretrained by utilizing the ImageNet dataset are worked on this study to test whether pretrained models can do well on this task or not. To achieve this stated goal, popular models are connected with the Dropout layer, the Global Average Pooling layers and the output layer sequentially to accommodate this requirement of the output. By controlling variables, multiple models were compared with different values of hyperparameters, such as number of epochs, batch size and image size. Through experiments, it is conspicuous that simpler models could converges on this small dataset rapidly than more complex models. Plus, pretraining cannot change the convergence of models and it would improve the performance of models which are already converged on this dataset.**

Keywords-component; Skin cancer classification, Pretrained model, Convolutional Neural Networks

I. Introduction

Cancer is pernicious to human's daily life, which may lead to death human. There is a myriad of various cancers that probably cause people to die. Skin cancer is one of the most prevalent cancers in the world. It is malignant tumors that are generated by the buildup of abnormal cells. Normally, new cells are created through division to replace older ones. However, some old cells may not die and some new cells are produced in an incorrect method. Due to the mutation of these cells, the out-of-control accumulation of abnormal cells in the epidermis forms a malignant tumor [1]. The main types of skin cancer are basal cell carcinoma (BCC), squamous cell carcinoma (SCC), melanoma, and Merkel cell carcinoma (MCC) [2].

Statistics show that 1.20 million people all over the world are died because of skin cancer in 2020 [3]. Moreover, an estimated 106,110 new cases of invasive melanoma and 101,280 cases of in situ melanoma will be diagnosed in the US, while 7,180 people will die from the disease in 2021 [4]. It is estimated that 5.4 million cases of basal cell carcinoma and squamous cell carcinoma are diagnosed annually in the United States among 3.3 million people. Some people are diagnosed with more than 1 skin cancer. The number of non-melanoma skin cancers has been growing for several years [5].

The two main causes of skin cancer are the sun's harmful ultraviolet (UV) rays and the use of UV tanning beds. The good news is that if skin cancer is caught early, dermatologists can treat it and high odds of eliminating it entirely [2]. Thus, the early detection of skin cancer is crucial to patients. Nevertheless, it increases the demands of a mass of sophisticated dermatologists with rich experiences. Therefore, it is necessary to diagnose and detect skin cancer automatically, which would lead us to computer-based systems. Traditional computer vision algorithms would extract features to detect skin cancer through an intricate and inconvenient strategy. Nowadays, multiple deep learning architectures are applied in the medical field to detect cancer cells, which results in extraordinary results of the detection of skin cancer using neural networks. Recently, multiple researchers had some contributions to the detection and diagnosing of skin cancer. Some researchers utilized the specific Convolutional Neural Network, You Only Look Once (YOLO), to extract features from the skin cancer and pass the obtained features to Fully Connected Network for classification by transfer learning [6]. Some scholars built their network for the classification of skin cancer images after removing noise and reducing some input images [7]. Numerous classifications are completed by constructing Convolutional Neural Networks using a large number of training images and testing images, such as the dataset called HAM10000 including over 10000 pictures [8]. Nonetheless, sometimes the dataset of skin cancer may not be readily acquired with high resolution so that the network could only extract features of skin cancer from a handful of blurry images. Furthermore, many pervasive Convolutional Neural Network models haven't be tried on small datasets.

978-1-6654-7997-4/22/$31.00 ©2022 IEEE
DOI 10.1109/IWECAI55315.2022.00087

Even if some studies had worked on binary classification of skin cancer with a small dataset, the accuracy after training was about 78.44, which is not too high eventually [9]. Taking account of the scarcity of fairly clear images sometimes, many popular CNN models were compared and also constructed three models by ourselves to obtain a comprehensive conclusion. The digital images for this purposed are collected from [10]. Plus, some classic pre-trained models were also applied to this dataset to get a conspicuous comparison.

II. Methodology

A. Dataset description and preprocessing

In this study, the "Skin Cancer: Malignant vs Benign" dataset provided on Kaggle [10] was utilized =. The dataset is a balanced dataset from the ISIA archive that consists of 1,800 pictures of benign skin moles and 1,800 pictures of malignant skin moles. The size of both malignant images and benign images is 224×224.

The preprocessing of our dataset contains several parts to generate more various skin cancer pictures because the dataset is too small to cause overfitting and images are moles. First, the dataset was normalized by dividing 255. In that our dataset is already balanced, it is unnecessary to balance it anymore. Second, pictures were flipped horizontally to generate more samples. Third, more images are produced by shifting 0.15 times of the total original size horizontally and vertically. Moreover, some images are rotated 45° randomly to obtained new samples. In addition, more samples are generated through zoom in original samples 0.5 times. Overall, some samples are displayed in the Fig. 1 after applying data augmentation and all the parameters used for preprocessing are shown in the Table I.

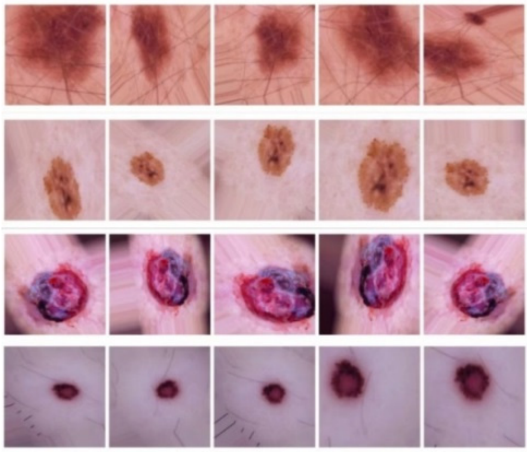

Figure 1. Visualization of images after data augmentation

TABLE I. PARAMETERS AND CORRESPONDING VALUE FOR PREPROCESSING

Parameters	Value
rescale	1./255
horizontal_flip	True
width_shift_range	0.15
height_shift_range	0.15
rotation_range	45
zoom_range	0.5

B. Approaches

In this study, multiple universal Convolutional Neural Network models were applied on our dataset including MobileNet, ResNet50V2, DenseNet121, DenseNet169, DenseNet201, NASNet-Mobile, Inception3, Xception, VGG16 and so on. To be compatible with the binary classification problem, one Global Average Pooling layer and the output layer would be connected behind one of those popular models. One more dropout layer with 0.5 rate was added before the Global Average Pooling layer sometimes to explore the difference.

Moreover, 3 proposed models were constructed for comparison to find which model can do well in our task. The first proposed model is the simplest model that contains 3 uniform convolutional layers and 3 uniform maxpooling layers, and after a flatten layer, there is 1 fully connected layer followed by the output layer, which is displayed in Fig. 2.

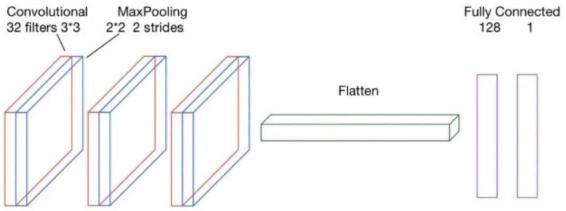

Figure 2. Architecture of the first proposed model

In the second proposed model, one more convolutional layer with maxpooling layer was added, and the number of filters is increasing so that the model is going deeper to extract features in higher dimensionality. The Fig. 3 indicates our second proposed model.

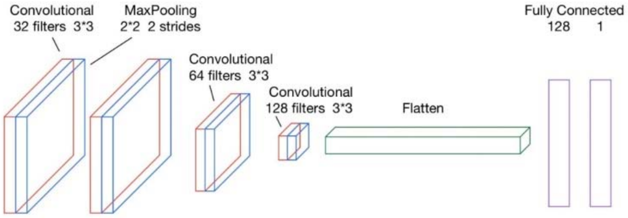

Figure 3. Architecture of the second proposed model

Furthermore, the inspiration of our third proposed model is from InceptionV3 model, which factorizes convolutions with large filter size and utilizes convolutions and maxpooling parallelly to reduce the size of feature maps, in order to decrease computational cost and avoid the representation bottleneck [11]. The third proposed model consists of 3 convolutional blocks used for downsampling, 2 inception blocks, and 1 upsampling block followed at the end. The Table 2 displays the whole architecture of the model, and the Fig. 4 and Fig. 5 indicates the structure of two inception blocks.

TABLE II. ARCHITECTURE OF THE WHOLE MODEL

Layer(type)	Related parameters of models	
	Output Shape	***Param #***
input(InputLayer)	(None, 224, 224, 3)	0
conv2d(Conv2D)	(None, 112, 112, 32)	896
conv2d_1(Conv2D)	(None, 112, 112, 32)	9248
conv2d_2(Conv2D)	(None, 112, 112, 64)	18496
max_pooling2d(MaxPooling2D)	(None, 112, 112, 64)	0
conv2d_3(Conv2D)	(None, 56, 56, 64)	36928
conv2d_4(Conv2D)	(None, 56, 56, 64)	36928
conv2d_5(Conv2D)	(None, 56, 56, 128)	73856
max_pooling2d_1(MaxPooling2D)	(None, 56, 56, 128)	0
conv2d_6(Conv2D)	(None, 28, 28, 128)	16512
conv2d_7(Conv2D)	(None, 14, 14, 256)	295168
conv2d_8(Conv2D)	(None, 14, 14, 256)	590080
max_pooling2d_2(MaxPooling2D)	(None, 7, 7, 256)	0
inception1(Inception1)	(None, 7, 7, 416)	951104
inception2(Inception2)	(None, 4, 4, 1184)	1481664
max_pooling2d_3(MaxPooling2D)	(None, 4, 4, 1184)	0
dropout(Dropout)	(None, 4, 4, 1184)	0
conv2d_9(Conv2D)	(None, 4, 4, 256)	303360
flatten(Flatten)	(None, 4096)	0
dense(Dense)	(None, 1)	4097

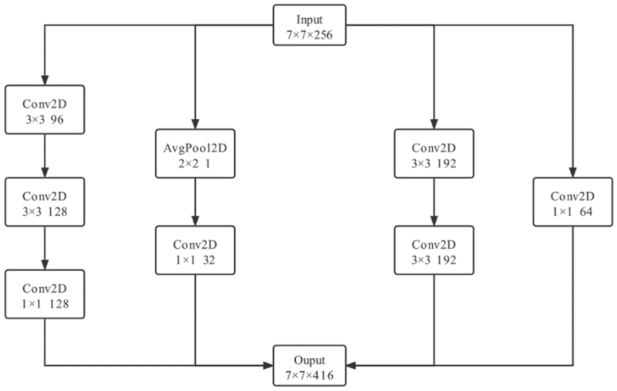

Figure 4. Architecture of two Inception Block1

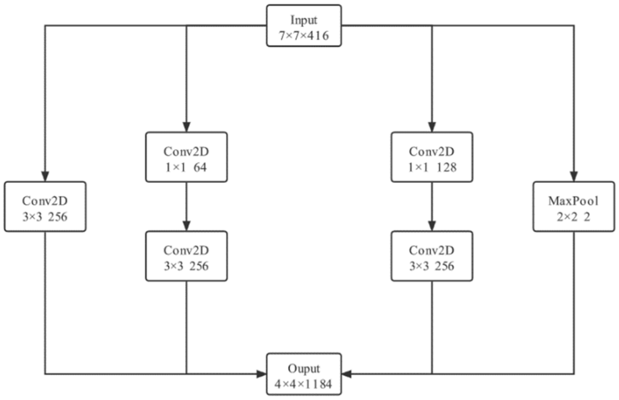

Figure 5. Architecture of two Inception Block2

During training, the control variable method on models mentioned before was applied to check the effect of different hyperparameters in the model, such as epochs, the size of inputs, and batch size. Different values for hyperparameters on different models were adjusted to acquire a comparison horizontally and vertically.

More importantly, another meaningful trial in our study is that using pretrain pervasive models to work on this binary classification problem of skin cancer. Because the ImageNet dataset created by Feifei Li [12] is a benchmark in image classification and it contains 14 million images which may not include pictures about skin cancer, the significance of pretraining popular models and applying them on skin cancer is that checking the effect of the pretrain dataset on the binary classification problem of skin cancer is meaningful. In other words, the desirable result is that pretrained models using the ImageNet dataset could perform better on the binary classification problem of skin cancer.

C. Implementation Details

Because the purpose of the study is classifying malignant images and benign images which is a binary classification problem, the "binary crossentropy" was used as the loss function. The Fig. 6 shows that the sigmoid function [13] is perfectly suitable to simulate probabilities and the Equation 1 displays the loss function where $\mathrm{y_i}$ denotes the labels (0 or 1) of our samples and $\mathrm{p(y_i)}$ denotes the probability of predicting the $\mathrm{i^{th}}$ sample as positive. Now that the probability falls down between [0,1], and the sum of probabilities of predicting 0 or 1 should be $1(\mathrm{p(y_i = 1) + p(y_i = 0) = 1})$, the sigmoid function was used to predict the probability of each label instead of predicting a discrete label. Then the logarithmic operation was used to penalize more on low probabilities and penalize less on high probabilities in the loss.

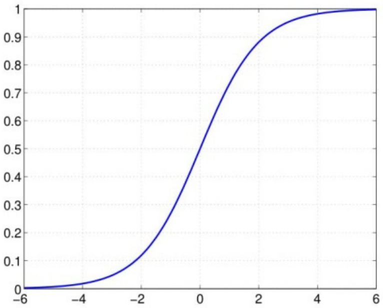

Figure 6. The sigmoid function

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

Besides, Adam (Adaptive Moment Estimation) optimizer was also usedto make our models learn parameters during backpropagation. It is a combination of the momentum algorithm and the RMSProp algorithm [14], which considers the effect of accumulated gradient and the effect of decayed sum of gradient squared respectively. In that Adam algorithm works well in recent years empirically, it was applied in the neural networks.

After settling the optimizer and loss function, the accuracy of different neural network models was chose as measurement. Due to the large number of experiments, the most conspicuous measurement for comparison among models is the accuracy of training and testing [15]. Thus, the accuracy was chose as the critical criteria. The Equation 2 shows the calculation of accuracy.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{2}$$

$\text{WhereTP} = \text{True positive}; \ \text{FP} = \text{False positive}; \ \text{TN} = \text{True negative}; \ \text{FN} = \text{False negative}.$

III. Results And Discussion

A. Classification performance of different models

The Table III represents the accuracy and loss for both training and testing of our first proposed model with different values of hyperparameters and fixed batch size 32.

TABLE III. Results of the first proposed model with different epochs and different image sizes

Number#	Performance of trained models					
	Epochs	*Image size*	*Training accuracy*	*Training Loss*	*Testing accuracy*	*Testing Loss*
1	5	128× 128	0.7922	0.4251	0.8000	0.3892
2	10	128× 128	0.8028	0.3898	0.8076	0.3635
3	5	224× 224	0.7952	0.4197	0.8091	0.4532
4	5	64× 64	0.7892	0.4267	0.8106	0.4001
5	10	64× 64	0.7979	0.4186	0.8000	0.3738

Then, the Table IV indicates the accuracy and loss for both training and testing of other two proposed models. The Table 5 presents the accuracy and loss for both training and testing of MobileNet model and MobileNetV2 model with different values of hyperparameters. Regarding other different popular neural network models, such as ResNet50V2, DenseNet121, DenseNet169, DenseNet201, NASNet-Mobile, Inception3, VGG16 and Xception, the batch size was fixed with 64 and image size with 224×224, and results are shown in Table 6.

TABLE IV. Results of proposed model 2 and proposed model 3.

Model	Performance of trained models						
	Batch size	*Epochs*	*Image size*	*Training accuracy*	*Training loss*	*Testing accuracy*	*Testing loss*
Proposed model 2	32	10	224 × 224	0.8009	0.4154	0.8221	0.4082
Proposed model 3	32	20	224 × 224	0.5461	0.6891	0.5455	0.6890

TABLE V. Results of MobileNet and MobileNetV2 with different values of hyperparameters.

Number#	Related parameters of trained models							
	Model	*Batch size*	*Epochs*	*Image size*	*Training accuracy*	*Training loss*	*Testing accuracy*	*Testing loss*
1	MobileNet(No dropout)	128	32	128 × 128	0.8604	0.2846	0.6485	0.6381
2	MobileNet(No dropout)	128	32	224 × 224	0.8790	0.2511	0.7470	0.8975
3	MobileNet(No dropout)	64	32	224 × 224	0.8658	0.2879	0.8045	0.5503
4	MobileNet(with dropout)	64	32	224 × 224	0.8642	0.2879	0.8182	0.3656
5	MobileNet(pretrained)	64	32	224 × 224	0.9518	0.1133	0.9030	0.2917
6	MobileNet(pretrained)	64	50	224 × 224	0.9727	0.0776	0.9121	0.3063
7	MobileNetV2	64	32	224 × 224	0.8635	0.2972	0.5455	0.8879
8	MobileNetV2(pretrained)	64	32	224 × 224	0.9314	0.1629	0.7742	1.3352

TABLE VI. Results of other popular neural network models and some of them are pretrained using the ImageNet dataset.

Model	Performance of trained models				
	Epochs	*Training accuracy*	*Training loss*	*Testing accuracy*	*Testing loss*
ResNet50V2	10	0.8127	0.3919	0.4515	1.3300
DenseNet121	10	0.8043	0.3962	0.5591	1.0786
DenseNet169	15	0.8248	0.3729	0.6864	2.2511
DenseNet201	10	0.8062	0.4133	0.7606	0.9976
NASNet-Mobile	20	0.8586	0.3025	0.4409	11.036
VGG16	10	0.5461	0.6891	0.5455	0.6890
Inception3	15	0.8297	0.3679	0.8212	0.4213
Inception3(pretrained)	15	0.8995	0.2285	0.8394	0.4671
Xception	15	0.8616	0.3123	0.8258	0.3360
Xception(pretrained)	15	0.9166	0.2001	0.8909	0.4083

B. Discussion

From the obtained results of multiple models, it can be discovered numerous interesting findings by comparing two results of models parallelly or vertically. When our first proposed model was applied on this small dataset, the results shown in Table III was obtained. First of all, our first proposed model could converge rapidly on this small dataset in 5 epochs. And by comparing the first row and the second row, it could conclude that the accuracy would gain a little growth if increasing the number of training times. Moreover, reducing the input size of the image would not help getting a higher accuracy, since when the image size was reduced from 224×224 to 64×64 or 128×128, the testing accuracy is changed from 0.8091 to 0.8000.

On the basis of the result from our first proposed model, the second and the third model were built. From the Table IV, it could derive that our second proposed model also converges on this dataset whereas the third proposed model cannot converge on this dataset in 20 epochs. It is not difficult to discover that simpler models could converge on this small dataset more swiftly than complex models, by comparing the first, second and third proposed model with fixed 32 batch size and 224×224 image size. Although training the third proposed model through 20 epochs, it seems that the complex model cannot converge readily on small dataset.

Moreover, when applying the MobileNet model on this problem, different values were changed for batch size and image size. By comparing the first row and the second row in Table V, the batch size for two models are the same whereas the image size for them are different. No matter the size of the image, it is hard to converge with 128 batch size. However, after changing the batch size to 64 and the image size remains as 224×224, the model converged well on the dataset and the testing accuracy is 0.8045. The difference is clearly shown through the comparison between the second row and the third row in Table V. Furthermore, the contrast between the third row and the fourth row is that adding one more dropout layer with 0.5 rate before the Global Average Pooling layer, and it is obvious that the model with dropout layer performs better. On the other hand, it is noticeable that the pretrained MobileNet model using the ImageNet dataset achieves an extraordinarily higher accuracy than the MobileNet model without pretraining, which could be distinguished through the fourth row and the fifth row. In this way, aggrandizing the number of epochs for the pretrained MobileNet model, and its testing accuracy reached 0.9121. Then, a more complex model called MobileNetV2 was tried. The results displayed as the seventh row and the eighth row of Table V implies that pretrained MobileNetV2 also get a higher accuracy on the testing dataset. Thus, it can be concluded that using the ImageNet dataset would improve the testing accuracy on this small dataset for skin cancer.

Last but not least, when the batch size was settled as 32 and image size as 224×224 on other classical neural network models, most of them cannot converge on this dataset as exhibited in Table VI. Now that the complexity of models is becoming larger, models are getting more difficult to learn the inputs in 10 epochs or 15 epochs. It demonstrates that more complex models are harder to converges on this skin cancer dataset again. Nonetheless, the Inception3 model and the Xception model converge well among those pervasive models, and their testing accuracy attained 0.8212 and 0.8258 respectively, which are acceptable by comparing to the result of MobileNet model shown as the fourth row in Table IV. Based on the experiments of Inception3 and Xception, they are pretrained using the ImageNet dataset, and their testing accuracy did improve apparently. Therefore, it also certifies that using the ImageNet dataset would improve the testing accuracy on this small dataset for skin cancer.

IV. Conclusion

In this paper, proposed models and popular neural network models on the small dataset attained from Kaggle are proposed, and pretrained model was tried to find whether pretrained model can do well on the collected dataset. Through experiments conducted on different models, it can conclude that simpler models could converges rapidly on this skin cancer dataset and complex models could not converges readily due to the small amounts of images or the diversity of skin cancers. Moreover, pretraining models using the ImageNet dataset on this skin cancer problem would improve the testing accuracy. Hence, if the model converges well on this dataset, pretraining is beneficial to training our models and obtain a higher performance for skin cancer. The testing accuracy of the simplest model with pretraining chose attained 0.9121, which is higher than other binary classification problem of skin cancer with small dataset particularly. In the future, increasing the number of epochs of complex models which has already converged on this dataset is needed and attempting other architectures of model to check whether there is a better performance.

References

[1] Skin Cancer 101. Skin Cancer Foundation. https://www.skincancer.org/skin-cancer-information/, 2021.

[2] What Is Skin Cancer. Centers for Disease Control and Prevention. https://www.cdc.gov/cancer/skin/basic_info/what-is-skin-cancer.htm, 2021.

[3] Cancer. World Health Organization. https://www.who.int/news-room/fact-sheets/detail/cancer, 2021.

[4] Cancer Facts & Figures. American Cancer Society. https://impactmelanoma.org/wp-content/uploads/2021/02/acs-skin-cancer-facts-and-figures-2021.pdf#:~:text=Invasive%20melanoma%20accounts%20for%20about%201%25%20of%20all,and%20use%20of%20health%20care%20may%20also%20contribute, 2021.

[5] Skin Cancer (Non-Melanoma): Statistics. Cancer.Net. https://www.cancer.net/cancer-types/skin-cancer-non-melanoma/statistics, 2021.

[6] N. Ruban, J. Tharun, N. Alex and R. Vijayarajan, "A Dermoscopic Skin Lesion Classification Technique Using YOLO-CNN and Traditional Feature Model", Arabian Journal for Science and Engineering, vol. 46, pp. 9797-9808, 2021.

[7] S. Md, H. Jahurul, M. Md and K. Md, "An enhanced technique of skin cancer classification using deep convolutional neural network with transfer learning models", Machine Learning with Applications, 100036, 2021.

[8] Skin Cancer MNIST: HAM10000. Kaggle. https://www.kaggle.com/kmader/skin-cancer-mnist-ham10000, 2018.

[9] N. Hemalatha, B. Nausheeda, K. Athul and Navaneeth, "Detection of Skin Cancer using Deep CNN", International Journal of Recent Technology and Engineering(IJRTE), pp. 2277-3878, Volume. 8, pp. 2277-3878, 2020.

[10] Skin Cancer: Malignant vs. Benign. Kaggle. https://www.kaggle.com/fanconic/skin-cancer-malignant-vs-benign, 2019.

[11] S. Christian, V. Vincent, I. Sergey, S. Jonathon and W. Zbigniew, "Rethinking the Inception Architecture for Computer Vision", 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[12] ImageNet. ImageNet. https://www.image-net.org/, 2020 Stanford Vision Lab, Stanford University, Princeton University, 2021.

[13] Sigmoid function. WIKIPEDIA. https://en.wikipedia.org/wiki/Sigmoid_function, 2021.

[14] A Visual Explanation of Gradient Descent Methods(Momentum, AdaGrad, RMSProp, Adam). towards data science. https://towardsdatascience.com/a-visual-explanation-of-gradient-descent-methods-momentum-adagrad-rmsprop-adam-f898b102325c, 2020.

[15] Accuracy and precision. WIKIPEDIA. https://en.wikipedia.org/wiki/Accuracy_and_precision, 2021.