Implementation of Convolutional Neural Networks in Skin Cancer Classification Using Django

Nur Nafi'Iyah

July 12, 2020

Implementation of Convolutional Neural Networks in Skin Cancer Classification Using Django

Nur Nafi'iyah

Informatics Engineering Study Program, Faculty of Engineering, Lamongan Islamic University, Jalan Veteran 53A Lamongan, Indonesia
mynaff26@gmail.com, nurnafiiyah.19051@mhs.its.ac.id

Abstract. Skin diseases can be clearly seen by yourself or others. Although this disease is clearly visible on the skin, sometimes we are very worried, for example if this skin disease is not a mild disease. So there are some people if you experience skin diseases directly and will quickly go to a dermatologist, both to check complaints and symptoms experienced. This skin is very protective of the body especially from the sun, so that if something unexpected happens it can result in death. One example of a deadly skin disease is skin cancer or cancerous tumor. In this study the classification of skin cancer will be Banign and Malignant with CNN algorithm. Where the dataset used was 3297 skin cancer images taken from the Kaggle website. We use 2 different CNN architectures. The first architecture has 6,427,745 parameters, and the second is 2,797,665. With these two architectures, the accuracy values range from 70-90%, the first model has an accuracy value of 93%, and the second model has 74%. We did training for many times, each time we did 10 epoches of repetition, and every epoch of 100-200 iterations.

1. Introduction

IoT (Internet of Things) technology has helped create an effective health system [1]. So that doctors can provide services anywhere without reducing the diagnosis results significantly [1]. Deep learning is a neural network model that can help in doing good computing [1]. The Convolution Neural Network (CNN) method is used to detect four types of computer vision-based skin diseases [2]. The layers used are the convolution layer, activation screen, pooling screen, fully connected screen, and softmax classification [2]. CNN is also used to diagnose breast cancer. Where the experimental results show that CNN can be used for classification of breast cancer with an accuracy of 91.3% [3].

Related to CNN, several studies that carry out diagnoses in the health sector have used computers. So that the CNN algorithm can be used as an alternative in diagnosing without reducing the role of the doctor himself. For example, research in diagnosing cervical cancer using CNN [4]. The computer-based system results in diagnosing cervical cancer into 7 types and its accuracy is 91.2% to 99.5% [4].

Skin cancer is a dangerous type of skin disease but can be cured if detected early [5]. Artificial intelligence is able to classify skin cancers whose accuracy resembles a dermatologist's diagnosis [5]. The algorithm used to classify skin cancer is CNN. The CNN technique proved significant in classifying dermoscopis melanoma with a sensitivity value of 95% [5].

In the world of leather industry, the process of identifying skin types is very important in order to be able to control the quality of the products produced. Examination of product results so that it can be known according to the suitability of skin types manually will be subject to errors due to human error. So we need a computer vision system in order to be able to classify leather sheets based on product quality. Several algorithms were tested to classify skin sheets starting from the extraction of texture and classification features, one of them with multilayer neural networks [6].

With some previous research, we will classify skin cancer using the web-based CNN algorithm. The purpose of this study will be to classify Banign and Malignant skin cancers with neural networks. In order to make early detection of skin cancer, and can reduce the death rate.

2. Literature Review

There are types of leather sheets studied in the industrial world such as Figure 1.

Figure 1. (a) Types of Raw Hides and (b) Wet Blue Leather [6]

Where the input images such as Figure 2 are classified using multilayer neural network algorithms such as Figures 2 and 3.

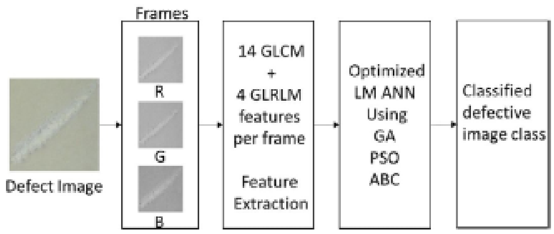


Figure 2. Khwaja Research Steps [6]

In Javeria Amin's research, using the architecture of Alexnet and VGG-16 as a pretrain [7]. Where the image of a diseased skin is carried out training with the architecture of Alexnet and VGG-16 so that it can be classified into Banign and Malignant (see Figure 4) [7]. Melanoma type skin cancer is a very fatal cancer [7]. To extract features from skin images using PCA, and wavelet transforms [7]. The algorithm that has recently provided good and reliable accuracy is CNN. Convolution nerve tissue (CNN) is very good at classifying skin lesions and analyzing images [8]. Diagnosis using a computer with the CNN algorithm can help in the doctor's performance. The framework of the diagnosis of computer-based skin lesions combines the image of segmented skin lesions and classifies skin lesions into multi-grade [8]. Classification with convolution neural networks (CNN) there is an acceleration of learning (transfer learning), where this process uses a network architecture that is already available. The transfer learning architecture uses the pretension Inception-v3, resNet-50, Inception-ResNet-v2, and DenseNet-201 [9]. Classification of skin lesions is a process caused by limitations in the characteristics of dermoscopic images during the process of capture or sampling. Skin lesions have several types, including malignant cancers such as melanoma, benign cancers such as nevi, BCC, and SCC [10].

Convolution nerve tissue can be used as a classification of skin lesions in the dermatological field [11]. Analyzing the image and the process of segmenting and extracting features of skin lesions must be examined carefully. CNN using the architecture of rapid learning (transfer learning) is used to classify skin lesions [11]. CNN is an efficient and accurate method for the analysis of skin disorders [12]. And dermatologists need an effective system to facilitate diagnosis with reliable abilities [12].

Early detection of skin cancer is very important and can prevent death, and several types of skin cancer, carcinoma and melanoma [13]. A reliable automatic melanoma screening system (early detection) is a system that can diagnose using a computer-based algorithm. The CNN algorithm can be used as a screening and detecting malignant skin lesions early [14]. And the CNN process must require a dataset of images along with the type of skin lesions as machine learning. And the types of skin lesions studied by Balazs, including melanoma, nevus, and seborrheic [14]. Segmentation of skin lesions is an important process in dermoscopic image using a computer [15]. Many segmentation methods take the features of skin lesions, one of which is by convolution nerve tissue. The CNN network architecture that is often used for segmentation is (FCN-8s and U-Net) [15]. Computerized convolution nerve tissue (CNN) can distinguish melanoma and nevi based on dermoscopic images [16-17]. A total of 11,444 dermoscopic images were used as CNN training datasets. And the results CNN can be used as a tool for skin doctrines in classifying skin lesions in dermoscopic images [17]. Skin cancer is one type of cancer that is often experienced by whites [18]. A good algorithmic approach for the classification or diagnosis of skin lesions is CNN pretrain [18]. Automatic diagnosis system for early detection of skin cancer has a very good effect [18-19]. It is proven that the process of treating patients that are detected early can be treated quickly. So in order to be able to make a computerized diagnostic system based on dermoscopic imagery, it must perform several complete stages. The first stage is segmenting the skin lesions and taking dermascopic features. These features are used as a reference for learning convolution neural networks [19].

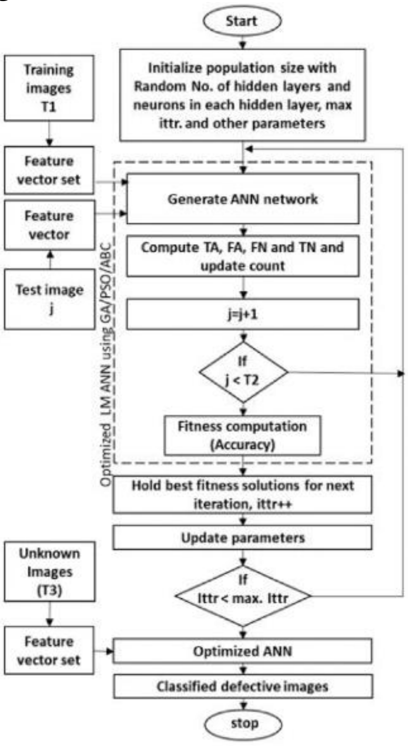


Figure 3. Steps in Training and Testing Neural Network Algorithms [6]

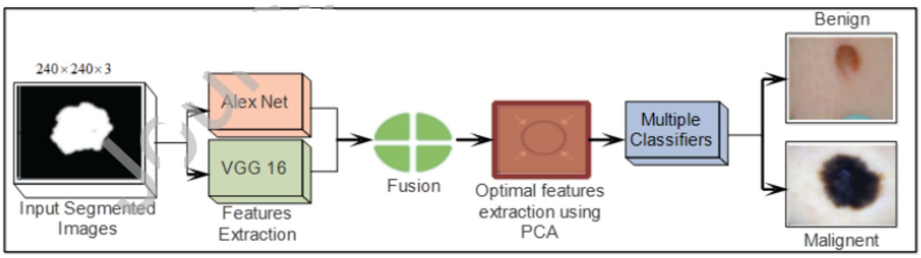


Figure 4. Classification of Skin Cancer Process [7]

	Management decision			Binary classification		
	Sensitivity	Specificity	Accuracy	Sensitivity	Specificity	Accuracy
Raters level I						
All (*N* = 96)	89.0%	80.7%	84.0%	83.8%	77.6%	**80.1%**[a]
Beginner (*n* = 17)	85.7%	76.1%	**79.9%**[a]	80.0%	67.7%	**72.6%**[a]
Skilled (*n* = 29)	89.7%	79.1%	83.3%	84.3%	75.9%	**79.3%**[a]
Expert (*n* = 40)	91.1%	84.1%	**86.9%**[b]	86.2%	82.9%	84.2%
Raters level II						
All (*N* = 96)	94.1%	80.4%	**85.9%**[b]	90.6%	82.4%	**85.7%**[b]
Beginner (*n* = 17)	92.9%	74.7%	82.0%	89.0%	76.0%	81.2%
Skilled (*n* = 29)	94.7%	79.2%	85.4%	90.9%	81.2%	85.1%
Expert (*n* = 40)	94.8%	84.4%	**88.5%**[b]	91.8%	86.6%	**88.7%**[b]
CNN	95.0%	76.7%	84.0%	95.0%	76.7%	84.0%

Figure 5. Classification of Skin Lesions Results [8]

Melanoma is a type of deadly skin cancer [14] [20]. So we need a computer-based system that has a good learning algorithm. Image-based skin cancer detection consists of processes of image improvement, segmentation, extraction of interesting features from images and classification of skin lesions [20]. One good learning algorithm is the convolution neural network (CNN). CNN can be used to identify malignant tumors on the skin surface with a sensitivity value of 93.3% [20].

3. Research Methods

This study classifies Banign and melanoma skin cancers as shown in Figure 6. Where 3297 images were downloaded from the Kaggle dataset, with divisions such as Table 1. And the sample dataset used was as Table 2, with 224x224 input image size with color image type.

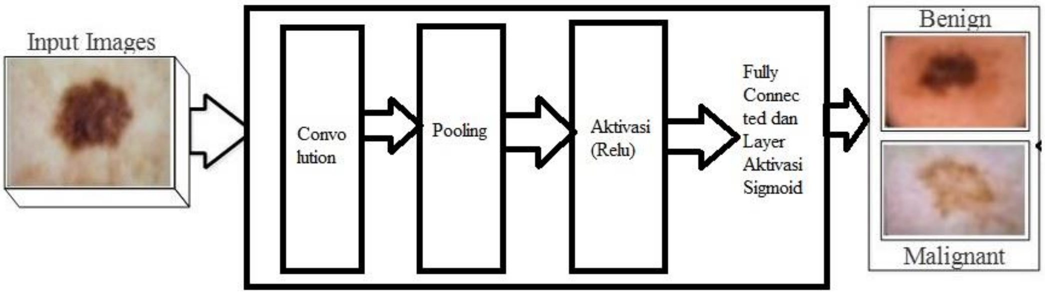


Figure 6. Classification of Skin Cancer with CNN

Table 1. Dataset of Skin Cancer Image

Dataset	Train	Test	Summary
Banign	1440	360	1800
Malignant	1197	300	1497
Summary	2637	660	3s297

Table 2. Sample Dataset

Banign	Malignant
	
	

4. Results and Discussion

We classified skin cancer into 2 classes, Banign and Malignant [7]. We did the classification using 2 CNN architectural models. The first CNN architectural model has a parameter value of 6,427,745, with an architecture such as Figure 7. There are 2 times 2D convolution screens, 2 times pooling screens (using max pooling), and Sigmoid activation functions. 2D convolution is to multiply the input image with the kernel or filter. The multiplication process of each pixel image will be multiplied by a filter, see the illustration of the multiplication or the convolution process in Figure 8. Where the goal of 2D convolution is to take the maximum features. Next, the pooling screen is a screen to determine the value of the best features (see Figure 9).

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 224, 224, 16)      448
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 112, 112, 16)      0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 112, 112, 32)      4640
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 56, 56, 32)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 100352)            0
_________________________________________________________________
dense_1 (Dense)              (None, 64)                6422592
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 65
=================================================================
Total params: 6,427,745
Trainable params: 6,427,745
Non-trainable params: 0
```

Figure 7. First CNN Architecture

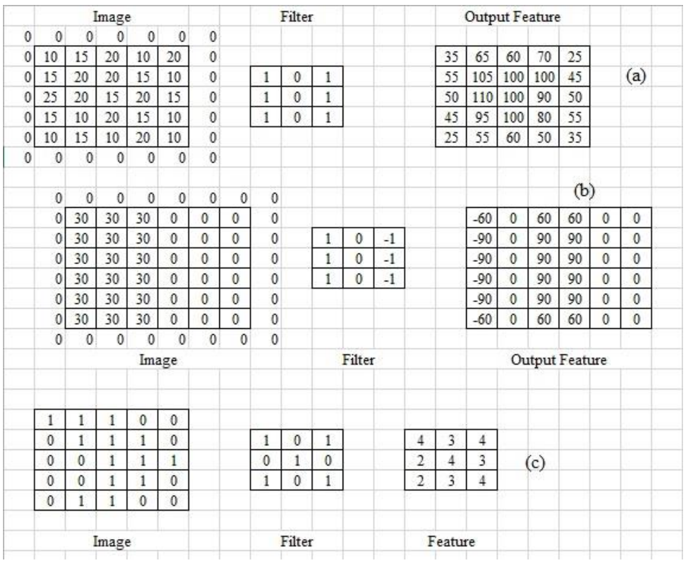


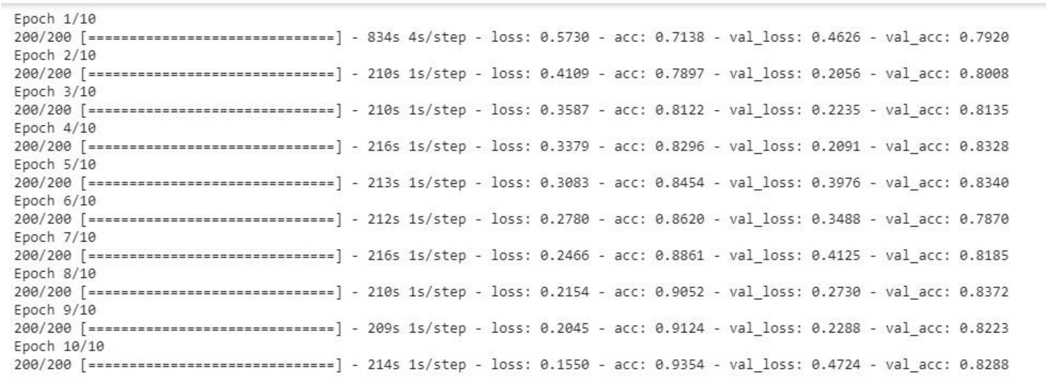
Figure 8. Convolution Screen Illustration (a), (b) Output Feature Image Size Same as Input Image (c) Output Feature Image Size Is Smaller

The pooling screen takes the best features by means of the maximum value of each image size or the average value of the image size (Figure 9). And the last screen is a screen to classify types of cancer (Banign and Malignant) using the Sigmoid function (Equation 1).

$$y = \frac{1}{(1+e^{-x})} \quad (1)$$

The first CNN architecture training process was carried out many times, with as many as 10 epochs and each epoch consisting of 200 iterations (Figure 10). With accuracy values ranging from 85-95% (Figure 11).

Figure 9. Pooling Screen Illustration



```
Epoch 1/10
200/200 [==============================] - 834s 4s/step - loss: 0.5730 - acc: 0.7138 - val_loss: 0.4626 - val_acc: 0.7920
Epoch 2/10
200/200 [==============================] - 210s 1s/step - loss: 0.4109 - acc: 0.7897 - val_loss: 0.2056 - val_acc: 0.8008
Epoch 3/10
200/200 [==============================] - 210s 1s/step - loss: 0.3587 - acc: 0.8122 - val_loss: 0.2235 - val_acc: 0.8135
Epoch 4/10
200/200 [==============================] - 216s 1s/step - loss: 0.3379 - acc: 0.8296 - val_loss: 0.2091 - val_acc: 0.8328
Epoch 5/10
200/200 [==============================] - 213s 1s/step - loss: 0.3083 - acc: 0.8454 - val_loss: 0.3976 - val_acc: 0.8340
Epoch 6/10
200/200 [==============================] - 212s 1s/step - loss: 0.2780 - acc: 0.8620 - val_loss: 0.3488 - val_acc: 0.7870
Epoch 7/10
200/200 [==============================] - 216s 1s/step - loss: 0.2466 - acc: 0.8861 - val_loss: 0.4125 - val_acc: 0.8185
Epoch 8/10
200/200 [==============================] - 210s 1s/step - loss: 0.2154 - acc: 0.9052 - val_loss: 0.2730 - val_acc: 0.8372
Epoch 9/10
200/200 [==============================] - 209s 1s/step - loss: 0.2045 - acc: 0.9124 - val_loss: 0.2288 - val_acc: 0.8223
Epoch 10/10
200/200 [==============================] - 214s 1s/step - loss: 0.1550 - acc: 0.9354 - val_loss: 0.4724 - val_acc: 0.8288
```

Figure 10. The First CNN Architecture Training Process

Next we made a second CNN architecture with a smaller number of parameters 2,797,665 (which consisted of a screen like Figure 12). And the training process is carried out many times with as many as 10 epochs, and each epoch consists of (100-200 iterations) (Figure 13). The accuracy of the second model shows a lower value, although iterates more, because the number of parameters is lower (Figure 14)

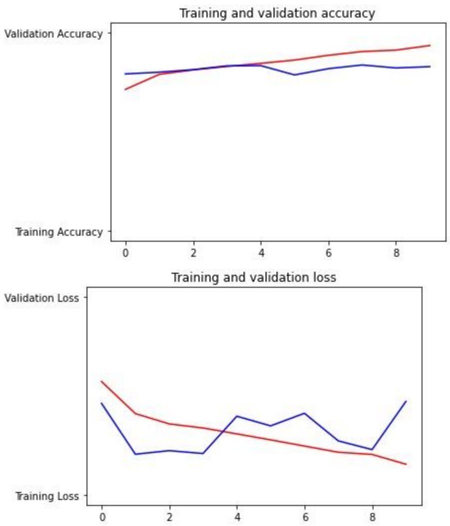


Figure 11. First CNN Architecture Test Results

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 222, 222, 32)	896
activation_1 (Activation)	(None, 222, 222, 32)	0
max_pooling2d_1 (MaxPooling2	(None, 111, 111, 32)	0
conv2d_2 (Conv2D)	(None, 109, 109, 32)	9248
activation_2 (Activation)	(None, 109, 109, 32)	0
max_pooling2d_2 (MaxPooling2	(None, 54, 54, 32)	0
conv2d_3 (Conv2D)	(None, 52, 52, 64)	18496
activation_3 (Activation)	(None, 52, 52, 64)	0
max_pooling2d_3 (MaxPooling2	(None, 26, 26, 64)	0
flatten_1 (Flatten)	(None, 43264)	0
dense_1 (Dense)	(None, 64)	2768960

Figure 12. Second CNN Architecture

```
Found 2637 images belonging to 2 classes.
Found 660 images belonging to 2 classes.
Epoch 1/10
200/200 [==============================] - 177s 883ms/step - loss: 0.7196 - acc: 0.5764 - val_loss: 0.9156 - val_acc: 0.6530
Epoch 2/10
200/200 [==============================] - 174s 871ms/step - loss: 0.5992 - acc: 0.7151 - val_loss: 0.5599 - val_acc: 0.7455
Epoch 3/10
200/200 [==============================] - 174s 870ms/step - loss: 0.5791 - acc: 0.7375 - val_loss: 0.4542 - val_acc: 0.7750
Epoch 4/10
200/200 [==============================] - 178s 890ms/step - loss: 0.5354 - acc: 0.7481 - val_loss: 0.5458 - val_acc: 0.7865
Epoch 5/10
200/200 [==============================] - 173s 865ms/step - loss: 0.5430 - acc: 0.7717 - val_loss: 0.4222 - val_acc: 0.7820
Epoch 6/10
200/200 [==============================] - 173s 865ms/step - loss: 0.5372 - acc: 0.7601 - val_loss: 0.3090 - val_acc: 0.7980
Epoch 7/10
200/200 [==============================] - 178s 888ms/step - loss: 0.5604 - acc: 0.7535 - val_loss: 0.4849 - val_acc: 0.7770
Epoch 8/10
200/200 [==============================] - 172s 861ms/step - loss: 0.5115 - acc: 0.7561 - val_loss: 0.1936 - val_acc: 0.8115
Epoch 9/10
200/200 [==============================] - 173s 865ms/step - loss: 0.5625 - acc: 0.7742 - val_loss: 0.7115 - val_acc: 0.7900
Epoch 10/10
200/200 [==============================] - 173s 864ms/step - loss: 0.5590 - acc: 0.7810 - val_loss: 0.2350 - val_acc: 0.7945
```

Figure 13. Second CNN Training Process

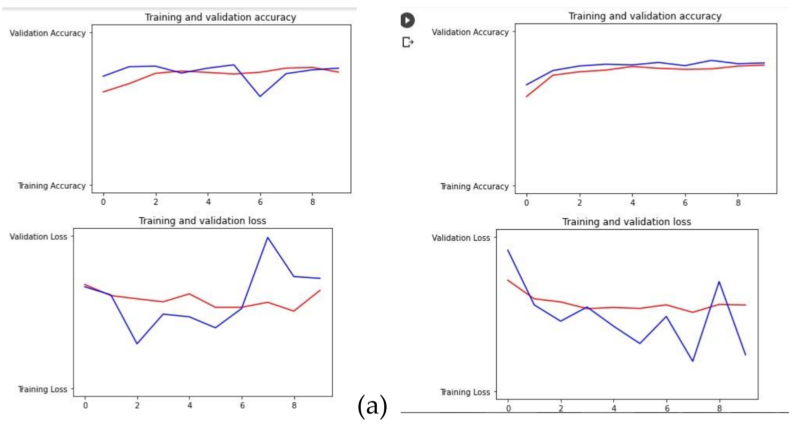
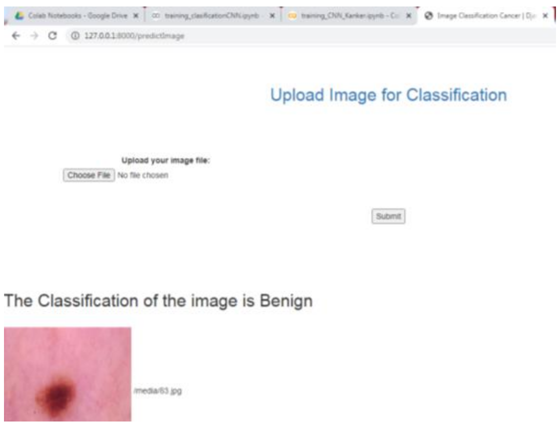
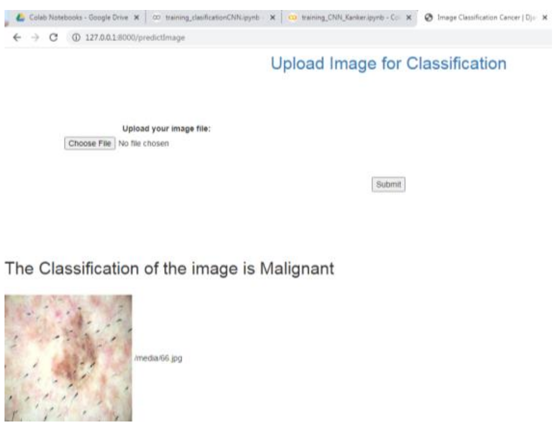


Figure 14. Second CNN Test Results (a) 100 Iteration (b) 200 Iteration

Furthermore, the results of the training model are stored in the form (h5) in the form of a weight vector model and network architecture. The model (h5) is used for web-based classification testing. And the web makes it easy for users to conduct skin cancer classification tests (Banign and Manignant) see Figure 15.



(a)



(b)

Figure 15. Web Display (a) Banign Classification Test Results (b) Manignant

5. Conclusion

The trial results showed that the parameters that were 6,427,745 were able to classify skin cancer with the highest accuracy of 93%. The parameters of 2,797,665 are able to classify skin cancer with the highest accuracy of 73%. The number of parameters determines the results of the classification accuracy (Banign and Malignant). The number of parameters is determined architectural arrangement (CNN layers).

6. References

[1] Douglas de A. Rodrigues, Roberto F. Ivo, Suresh Chandra Satapathy, Shuihua Wang, Jude Hemanth, Pedro P. Rebouc¸ as Filho, "A new approach for classification skin lesion based on transfer learning, deep learning, and IoT system.," *Pattern Recognition Letters,* pp. 1-8, 2020.

[2] T. Shanthi, R.S. Sabeenian, R. Anand, "Automatic diagnosis of skin diseases using convolution neural network," *Microprocessors and Microsystems,* pp. 1-8, 2020.

[3] Rui Yan, Fei Ren, Zihao Wang, Lihua Wang, Tong Zhang, Yudong Liu, Xiaosong Rao, Chunhou Zheng, Fa Zhang, "Breast cancer histopathological image classification using a hybrid deep neural network," *Methods,* pp. 1-9, 2019.

[4] Ahmed Ghoneim, Ghulam Muhammad, M.Shamim Hossain, "Cervical cancer classification using convolutional neural networks and extreme learning machines," *Future Generation Computer Systems,* pp. 643-649, 2020.

[5] Titus J. Brinker, Achim Hekler, Alexander H. Enk, Carola Berking, Sebastian Haferkamp, Axel Hauschild, Michael Weichenthal, Joachim Klode, Dirk Schadendorf, Tim Holland-Letz, Christof von Kalle, Stefan Fro, Bastian Schilling, Jochen S. Utikal, "Deep neural networks are superior to dermatologists in melanoma image classification," *European Journal of Cancer,* vol. 119, pp. 11-17, 2019.

[6] Khwaja Muinuddin Chisti Mohammed, Srinivas Kumar S, Prasad G, "Defectivetextureclassificationusingoptimizedneuralnetwork structure," *Pattern Recognition Letters* , vol. 135, pp. 228-236, 2020.

[7] Javeria Amin, Abida Sharif, Nadia Gul, Muhammad Almas Anjum, Muhammad Wasif Nisar, Faisal Azam, Syed Ahmad Chan Bukhari, "Integrated Design of Deep Features Fusion for Localization and," *Pattern Recognition Letters,* pp. 3-8, 2019.

[8] H. A. Haenssle, C. Fink, F. Toberer, J. Winkler, W. Stolz, T. Deinlein, R. Hofmann-Wellenhof, A. Lallas, S. Emmert, "Man against machine reloaded: performance of a market-approved convolutional neural network in classifying a broad spectrum of skin lesions in comparison with 96 dermatologists working under less artificial conditions," *Annals of Oncology,* vol. 31, no. 1, pp.

137-143, 2020.

[9] Mohammed A. Al-masni, Dong-Hyun Kim, Tae-Seong Kim, "Multiple Skin Lesions Diagnostics via Integrated Deep Convolutional Networks for Segmentation and Classification," *Computer Methods and Programs in Biomedicine,* pp. 2-25, 2020.

[10] Yanhui Guo, Amira S. Ashour, "Neutrosophic multiple deep convolutional neural network for skin dermoscopic image classification," pp. 269-285.

[11] Katharina Sies, Julia K. Winkler, Christine Fink, Felicitas Bardehle, Ferdinand Toberer, Timo Buhl, Alexander Enk, Andreas Blum, Albert Rosenberger, Holger A. Haenssle, "Past and present of computer-assisted dermoscopic diagnosis: performance of a conventional image analyser versus a convolutional neural network in a prospective data set of 1,981 skin lesions," *European Journal of Cancer,* vol. 135, pp. 39-46, 2020.

[12] M. Hajabdollahi, R. Esfandiarpoor, P. Khadivi, S.M.R. Soroushmehr, N. Karimi, S. Samavi, "Simplification of Neural Networks for Skin Lesion Image Segmentation Using Color Channel Pruning," *Computerized Medical Imaging and Graphics,* pp. 2-23, 2020.

[13] Ni Zhang, Yi-Xin Cai, Yong-Yong Wang, Yi-Tao Tian, Xiao-Li Wang, Benjamin Badami, "Skin Cancer Diagnosis Based on Optimized Convolutional Neural Network," *Artificial Intelligence In Medicine,* pp. 2-19, 2019.

[14] B. Harangi, "Skin lesion classification with ensembles of deep convolutional neural network," *Journal of Biomedical Informatics,* vol. 86, pp. 25-32, 2018.

[15] Fengying Xie, Jiawen Yang, Jie Liu, Zhiguo Jiang, Yushan Zheng, Yukun Wang, "Skin lesion segmentation using high-resolution convolutional neural network," *Computer Methods and Programs in Biomedicine,* pp. 3-20, 2019.

[16] Achim Hekler, Jochen S. Utikal, Alexander H. Enk, Axel Hauschild, Michael Weichenthal, Roman C. Maron, "Superior skin cancer classification by the combination of human and artificial intelligence," *European Journal of Cancer,* vol. 120, pp. 114-121, 2019.

[17] Roman C. Maron, Michael Weichenthal, Jochen S. Utikal, Achim Hekler, Carola Berking, Axel Hauschild, Alexander H. Enk, "Systematic outperformance of 112 dermatologists in multiclass skin cancer image classification by convolutional neural networks," *European Journal of Cancer,* vol. 119, pp. 57-65, 2019.

[18] Amirreza Mahbod, Gerald Schaefer, Chunliang Wang, Georg Dorffner, Rupert Ecker, Isabella Ellinger, "Transfer learning using a multi-scale and multi-network ensemble for skin lesion classification," *Computer Methods and Programs in Biomedicine,* vol. 193, pp. 1-9, 2020.

[19] Mansoureh Pezhman Pour, Huseyin Seker, "Transform domain representation-driven convolutional neural networks for skin lesion segmentation," *Expert Systems With Applications,* pp. 3-18, 2019.

[20] V. Srividhya, K.Sujatha, R.S. Ponmagal, G.Durgadevi, L. Madheshwaran, "Vision based Detection and Categorization of Skin lesions using Deep Learning Neural networks," *Procedia Computer Science,* vol. 171, p. 1726–1735, 2020.