

Skin Disease Detection using Machine Learning

Kritika Sujay Rao

Department of Computer Engineering
Vidyavardhini's College of Engineering And
Technology (Mumbai University)
Vasai, India

Pooja Suresh Yelkar

Department of Computer Engineering
Vidyavardhini's College of Engineering And
Technology (Mumbai University)
Vasai, India

Omkar Narayan Pise

Department of Computer Engineering
Vidyavardhini's College of Engineering And
Technology (Mumbai University)
Vasai, India

Dr. Swapna Borde

Department of Computer Engineering
Vidyavardhini's College of Engineering And
Technology (Mumbai University)

Abstract— Dermatology is the branch of bioscience that's involved with diagnosing and treatment of skin based mostly disorders. The immense spectrum of dermatologic disorders varies geographically and additionally seasonally because of temperature, humidness and alternative environmental factors. Human skin is one amongst the foremost unpredictable and tough terrains to mechanically synthesize and analyse because of its quality of unevenness, tone, presence of hair and alternative mitigating options. Though, many researches are conducted to find and model human skin victimisation (PC Vision techniques), only a few have targeted the medical paradigm of the matter. Due to lack of medical facilities available in the remote areas, patients usually ignore early symptoms which may worsen the situation as time progresses. Hence, there is a rising need for automatic skin disease detection system with high accuracy. Thus, we develop a multiclass deep learning model to differentiate between Healthy Skin Vs Skin suffering from a Disease and Classification of Skin Diseases into its main classes like MelanocyticNevi, Melanoma, Benign keratosis-like lesions, Basal cell Carcinoma, ActinicKeratoses, Vascular lesion and Dermatofibroma. We have used Deep Learning to train our model, Deep Learning is a part of Machine Learning in which unlike Machine Learning it uses large dataset and hence the number of classifiers is reduced substantially. The machine learns itself and divide the data provided into the levels of prediction and in a very short period of time gives the accurate results, thereby promoting and supporting development of Dermatology. The algorithm that we have used is Convolutional Neural Network (CNN) as it is one of the most preferred algorithm for image classification.

Keywords—*Dermoscopic images, Deep Learning, Data Enhancement, Convolutional Neural Network(CNN), Model Training, Testing and Evaluation.*

I. INTRODUCTION

Artificial Intelligence is taking over automation in all fields of application even within the healthcare field. In the past years these diseases have been a matter of concern due to the sudden arrival and the complexities which has increased life risks. These Skin abnormalities are very infectious and the require to be treated at earlier stages to avoid it from spreading. The majority of diseases is caused by unprotected exposure to excessive Ultraviolet Radiation(UR). Among all, benign type is considered to be less dangerous than malignant melanoma and can be cured with proper treatment, whereas the deadliest form of skin lesion is malignant

Melanoma. The survey results indicate that the back and lower extremity, trunk and upper extremity are heavily compromised regions of skin cancer. There are large instances of patients with age ranging from 30 to 60. Also, MelanocyticNevi, Carcinoma and Dermatofibroma are not prevalent below the age of 20years.

II. EXISTING TECHNOLOGY

A. Artificial Neural Network(ANN).

An artificial neuron network (ANN) is a statistical nonlinear predictive modelling method which is used to learn the complex relationships between input and output. The structure of ANN is inspired by the biological pattern of our brain neuron [2]. An ANN has three types of computation node. ANNs learn computation at each node through back-propagation. There are two sorts of data set trained and untrained data set which produces the accuracy by employing a supervised and unsupervised learning approach with different sort of neural network architectures like feed forward, back propagation method which uses the info set at a special manner. Using Artificial Neural Network, accuracy obtained in various researches is 80% which isn't optimum [2]. Also, ANNs require processors with parallel processing power. ANN produces a probing solution it does not give a clue as to why and how it takes place which reduces trust in the network

B. Back Propagation Network(BPN).

Back propagation, a strategy in Artificial Neural Networks to figure out the error contribution of each neuron after a cluster of information (in image recognition, multiple images) is processed. Back Propagation is quite sensitive to noisy and uproarious data. The BNN classifier achieves 75%-80% accuracy [2]. BNN is benefits on prediction and classification but the processing speed is slower compared to other learning algorithms [5] [2].

C. Support Vector Machine(SVM).

SVM is a supervised non-linear classifier which constructs an optimal n-dimensional hyperplane to separate all the data points in two categories [2]. In SVM, choosing an honest kernel function isn't easy. It requires long training time for large datasets. Since the final model is not easy to use we cannot make small calibrations to the model and it becomes difficult to tune the parameters used in SVMs. SVMs when compared with ANNs always give best results [3].

III. LITERATURE

Skin diseases are the 4th common cause of skin burden worldwide. Robust and Automated system have been developed to lessen this burden and to help the patients to conduct the early assessment of

the skin lesion. Mostly this system available in the literature only provide skin cancer classification. Treatments for skin are more effective and less disfiguring when found early and it is a challenging research due to similar characteristics of skin diseases. In this project we attempt to detect skin diseases .A novel system is presented in this research work for the diagnosis of the most common skin lesions (Melanocytic nevi, Melanoma, Benign keratosis-like lesions, Basal cell carcinoma, Actinic keratoses, Vascular lesion, Dermatofibroma). The proposed approach is based on the pre-processing, Deep learning algorithm, training the model , validation and classification phase. Experiments were performed on 10010 images and 93% accuracy is achieved for seven-class classification using Convolution Neural Networks (CNN) with the Keras Application API.

IV. DATASET

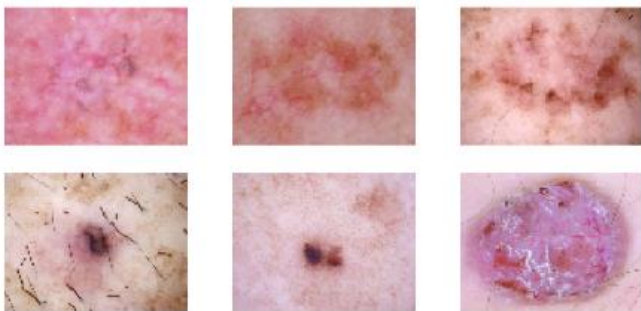


Fig. 1. Sample Data

The fig. above is the sample dataset which we have trained and tested.

V. IMPLEMENTATION (METHODOLOGY)

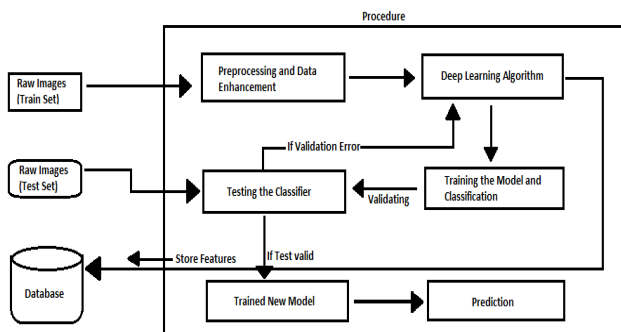


Fig.2. Procedure

To develop any ML-AI based system, be it this system; following steps are to be followed.

1. Data Gathering.

The proposed system has been assessed on dermatoscopic images which is collected from publicly available dataset based on Skin-Cancer-MNIST (Modified National Institute of Standards and Technology Database)-HAM10000. The number of options is endless. To save time and effort one can use publicly available data.

2. Data Preprocessing & Enhancement.

“Trash In- Good Out” is the basic motto in this step [6]. Validating your dataset with some basic profiling procedure will help speeding up the process, by slip-ups and grimy information [4]. AI algorithms don't give great outcomes when working with such information.

2.1. Data Cleaning.

Dirty data can cause confusion and results in unreliable and poor output. Hence first step in Data Pre-processing is Data Cleaning. Cleaning of data is done by filling in missing values, smoothing noisy data by identifying and/or removing outliers, and removing inconsistencies.

2.2. Data Transformation.

Data Transformation involves converting data from one format into another. It involves transforming actual values from one representation to the target representation.

2.3. Exploratory Data Analysis (EDA).

In this we explore different features of the dataset, their distributions and actual counts.

2.4. Label Encoding.

The dataset is labelled into 7 different categories:

1. MelanocyticNevi
2. Melanoma
3. Benign keratosis-like lesions
4. Basal cell carcinoma
5. ActinicKeratoses
6. Vascular lesions
7. Dermatofibroma

3. Training.

For this we have to divide the data into training set and testing set. This division can be in any ratio. Also, the batch size and number of epochs has to be decided beforehand.

4. Model Building.

We have used Convolutional Neural Network (CNN). A **Convolutional Neural Network (CNN or ConvNet)** is a category of deep neural networks, where the machine learns on its own and divide the data provided into the levels of prediction and in a very short period of time gives the accurate results [2]. A Convolutional Neural Network (CNN) is an algorithm in deep learning which consist of a combination of convolutional and pooling layers in sequence and then followed by fully connected layers at the end as like multilayer neural network [2]. CNN stands out among all alternative algorithms in classifying images. Crucial characteristics are Sparse Connectivity, Shared Weights and Pooling Feature so as to extract the best features. Also, the use of Graphical Processing Units (GPUs) have shrivelled the training time of deep learning methods. Giant databases of labelled data and pre-trained networks are now publicly available.

The figure below shows the difference between Sparse and Dense Connectivity.

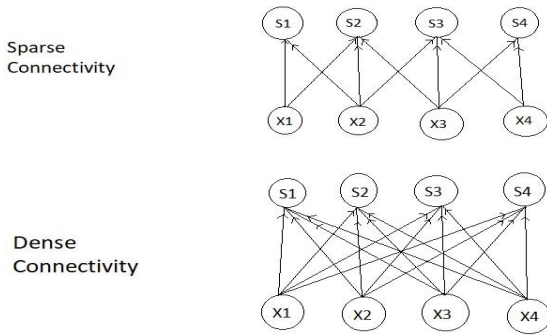


Fig.3. Sparse and Dense Connectivity

4.1. Explanation.

We have used **Keras Sequential API**, where you have just to add one layer at a time, starting from the input. **Conv2D** layer, a set of learnable features. The number of filters used here is thirty two. Each filter transforms a part of the image which is defined by the kernel size using the kernel filter. Transformed images are the filter maps. Next important layer is the pooling layer which simply acts as a down sampling filter. We have Max pooling, **MaxPool()** picks the **maximal worth** among set of two neighbouring pixels. This layer is employed to scale back (to cut back) machine value and additionally reduce overfitting to some extent. Combining both the above layers, CNN gets the ease to combine local features and learn global features. Activation Function **relu** is used to add non-linearity to the network. We use a regularization method, where a proportion of nodes in the layer are randomly ignored (setting their weights to zero) for each training sample i.e. the **Dropout** function. This improves in generalizing the network. Now, to convert the final feature maps into a one single 1D vector we need to flatten them, thus **Flatten** Layer is used. This flattening step is needed so that you can make use of fully connected layers after some of the above layers. It combines all the found native options of the previous convolutional layers. In the last layer, **Dense()** is used which gives the net output distribution of likelihood of every category. Once layers have been added, we need to set up a score function, a loss function and a proper optimization algorithm. We define **binary cross entropy** as our loss function which will actually measure the error rate between the observed labels and predicted labels. Next most important is the optimizer. **Adam Optimizer** has advantage as it involves functions of other optimizers as well. Adam is a well known and popular algorithm in the field of learning models. Next is the metric function which is used to evaluate the performance of the system, metric **accuracy** is used. **Learning Rate(LR)** is another important term. It is an annealing method. Ideally one should have a decreasing Learning Rate during rate so as to have minimal loss. **ReduceLROnPlateau** is used, the name itself means reduce the LR so as to reach the global minimum of loss function.

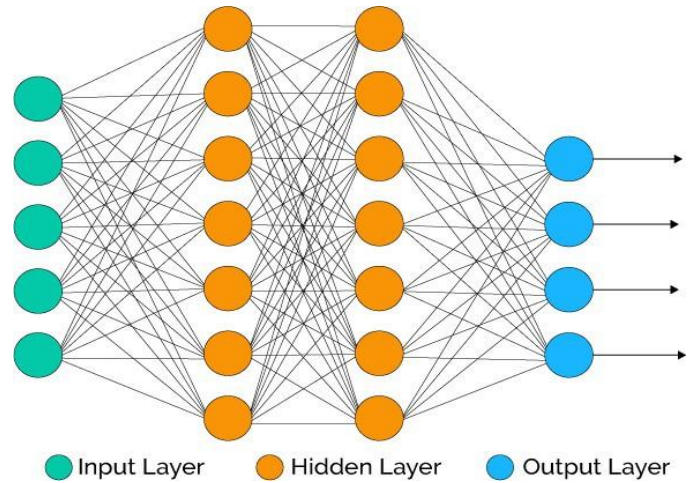


Fig.4. Fully Connected Network

5. Model Evaluation.

“**More the accuracy, better is the model**”. Every model is evaluated based on the accuracy achieved and the loss obtained. There are two accuracies involved: Validation accuracy And Test accuracy. Before this Validation set is different from Train set i.e. Validation set is independent from the Train set, Validation set is used for selecting parameters. Just for an instance if your model has 90% train accuracy and 89% validation accuracy then your model is expected to have 89% accuracy on new data.

6. Graphical Analysis.

This involves plotting Histogram and the Confusion Matrix. Confusion Matrix involves TP, TN, FP, FN [2]. The set of decision made by classification algorithm contains correct decision (true positive) and incorrect decision (false positive). False negative are those decision which is declared as negative by classification algorithm but actually they are positive. True negative are those decision which is correctly identified as negative by the classification algorithm.

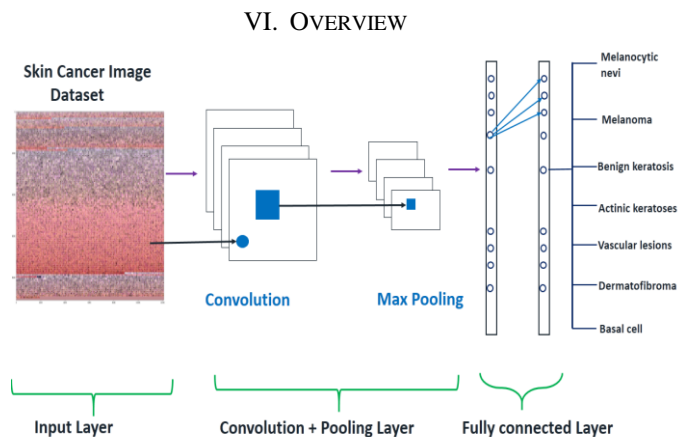


Fig.5. Convolution Neural Network (CNN)

VII. RESULTS

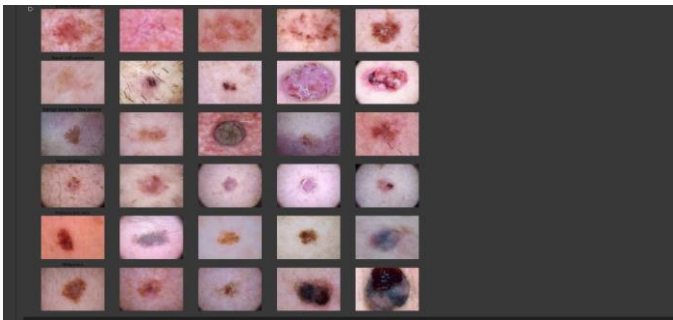


Fig.6. Streamed Images

```

[ ] tile_df = pd.read_csv(os.path.join(base_skin_dir, 'MIM10000_metadata.csv'))
[ ] tile_df['path'] = tile_df['image_id'].map(lambda id: os.path.join(path, id))
[ ] tile_df['cell_type'] = tile_df['dx'].map(lambda dx: lesion_type_dict.get(dx, 'unknown'))
[ ] tile_df['cell_type_idx'] = pd.Categorical(tile_df['cell_type']).codes
[ ] tile_df.sample(5)

   lesion_id  image_id  dx  dx_type  age  sex  localization  path  cell_type  cell_type_idx
0  8619  HAM_000422  ISIC_0027654  nv  follow_up  40.0  male  trunk (content/HAM10000_images_part_1/ISIC_0027654.jpg)  Melanocytic.nevi  4
1  9667  HAM_000518  ISIC_0028008  nv  consensus  55.0  male  upper extremity (content/HAM10000_images_part_1/ISIC_0028008.jpg)  Melanocytic.nevi  4
2  7009  HAM_0006145  ISIC_0031713  nv  histo  60.0  female  back (content/HAM10000_images_part_2/ISIC_0031713.jpg)  Melanocytic.nevi  4
    
```

Fig.7. Model Summary

```

Layer (type)          Output Shape         Param #
-----
conv2d (Conv2D)       (None, 75, 100, 32)  896
conv2d_1 (Conv2D)     (None, 75, 100, 32)  9248
max_pooling2d (MaxPooling2D) (None, 37, 50, 32)  0
dropout (Dropout)     (None, 37, 50, 32)  0
conv2d_2 (Conv2D)     (None, 37, 50, 64)  18496
conv2d_3 (Conv2D)     (None, 37, 50, 64)  36928
max_pooling2d_1 (MaxPooling2D) (None, 18, 25, 64)  0
dropout_1 (Dropout)   (None, 18, 25, 64)  0
conv2d_4 (Conv2D)     (None, 18, 25, 128)  73856
max_pooling2d_2 (MaxPooling2D) (None, 9, 12, 128)  0
dropout_2 (Dropout)   (None, 9, 12, 128)  0
Flatten (Flatten)     (None, 13824)  0
dense (Dense)         (None, 512)  7075008
dropout_3 (Dropout)   (None, 512)  0
dense_1 (Dense)       (None, 7)  3591
Total params: 7,721,415
Trainable params: 7,721,415
Non-trainable params: 0
    
```

Fig.8. Epoch-50

```

[ ] loss, accuracy = model.evaluate(x_test, y_test, verbose=1)
[ ] loss_y, accuracy_y = model.evaluate(x_valdata, y_valdata, verbose=1)
[ ] print("Validation accuracy = %f ; loss = %f" % (accuracy_y, loss_y))
[ ] print("Test accuracy = %f ; loss = %f" % (accuracy, loss))
[ ] model.save("model.h5")

2083/2083 [-----] - 211 10ms/sample - loss: 0.1506 - acc: 0.9335
882/882 [-----] - 8s 10ms/sample - loss: 0.1543 - acc: 0.9369
Validation accuracy = 0.936941 ; loss_y = 0.15436
Test accuracy = 0.933528 ; loss = 0.150595

[ ] plot_model_history(history)

/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:9: MatplotlibDeprecationWarning: Passing the minor parameter of set_ticks() positionally is deprecated if name == 'major'.
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:17: MatplotlibDeprecationWarning: Passing the minor parameter of set_ticks() positionally is deprecated if name == 'major'.
    
```

Fig.9. Graphical Plotting for Epoch-50

```

[ ] epochs = 1
[ ] batch_size = 10
[ ] history = model.fit_generator(
[ ] datagen.flow(x_train, y_train, batch_size=batch_size), class_weight=class_weights,
[ ] stop_per_epoch=train_steps // batch_size,
[ ] epochs=epochs,
[ ] validation_data=(x_valdata, y_valdata),
[ ] validation_steps=valdata_steps // batch_size,
[ ] callbacks=[learning_rate_reduction])

Epoch 1/2
720/721 [-----] - ETA: 0s - loss: 0.1703 - acc: 0.9409epoch 1/2
882/721 [-----] - 9s 11ms/sample - loss: 0.1401 - acc: 0.9375
721/721 [-----] - 33ms 466ms/step - loss: 0.1702 - acc: 0.9409 - val_loss: 0.1514 - val_acc: 0.9375
Epoch 2/2
720/721 [-----] - ETA: 0s - loss: 0.1679 - acc: 0.9411epoch 1/2
882/721 [-----] - 9s 11ms/sample - loss: 0.1426 - acc: 0.9374
721/721 [-----] - 33ms 466ms/step - loss: 0.1678 - acc: 0.9412 - val_loss: 0.1519 - val_acc: 0.9371

[ ] loss, accuracy = model.evaluate(x_test, y_test, verbose=1)
[ ] loss_y, accuracy_y = model.evaluate(x_valdata, y_valdata, verbose=1)
[ ] print("Validation accuracy = %f ; loss = %f" % (accuracy_y, loss_y))
[ ] print("Test accuracy = %f ; loss = %f" % (accuracy, loss))

2083/2083 [-----] - 20s 10ms/sample - loss: 0.1402 - acc: 0.9328
882/882 [-----] - 8s 10ms/sample - loss: 0.1547 - acc: 0.9371
Validation accuracy = 0.937121 ; loss_y = 0.154717
Test accuracy = 0.932815 ; loss = 0.140198
    
```

Fig.10. Epoch-2

```

ax1[1].set_title('train_loss')
ax1[1].set_ylabel('loss')
ax1[1].set_xlabel('epoch')
ax1[1].set_xticks(np.arange(1, len(model_history.history['loss'])+1), len(model_history.history['loss'])/10)
ax1[1].legend(['train', 'val'], loc='best')
plt.show()

[ ] plot_model_history(history)

/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:9: MatplotlibDeprecationWarning: Passing the minor parameter of set_ticks() positionally is deprecated if name == 'major'.
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:17: MatplotlibDeprecationWarning: Passing the minor parameter of set_ticks() positionally is deprecated if name == 'major'.
    
```

Fig.11. Graphical Plotting for Epoch-2

```

[ ] import cv2
[ ] test_image = cv2.imread('image1.jpg')

[ ] import matplotlib.pyplot as plt
[ ] plt.imshow(test_image)
[ ] plt.show()

cv2.imread('image1.jpg') at 0x7f3f3c3282b0

[ ] test_image = np.expand_dims(test_image, axis=0)
[ ] t1 = np.resize(test_image, (75,100,3))
[ ] t2 = np.expand_dims(t1, axis=0)

[ ] print(model.predict(t2))
[ ] print(model.predict_classes(t2))

[[ 0.  0.  0.  1.  0.  0.]]
    
```

Fig.12. Testing for an Image with Detection TABLE I.

EPOCH-50

Sr. No.	Evaluation		
	Metric/Parameter	Testing	Validation
1.	Accuracy	93.35%	93.35%
2.	Loss	15.65%	15.65%

TABLE II.

EPOCH-2

Sr. No.	Evaluation		
	Metric/Parameter	Testing	Validation
1.	Accuracy	93.28%	93.28%
2.	Loss	16.01%	16.01%

VII. DISCUSSION

The proposed system aims in automatic computer-based detection of skin diseases so as to reduce life risks. This has been no doubt a challenging task owing to the fine-grained variability in the appearance of skin.

VIII. CONCLUSION

Skin Diseases are ranked fourth most common cause of human illness, but many still do not consult doctors. We presented a robust and automated method for the diagnosis of dermatological diseases. Treatments for skin are more effective and less disfiguring when found early. We should point out that it is to replace doctors because no machine can yet replace the human input on analysis and intuition. Researches in European Society of Medical Oncology have shown for the first time that form of AI or ML is better than experienced dermatologists. In this a brief description of the system and the implementation methodology is presented.

IX. ACKNOWLEDGEMENT

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

We are heartily thankful to our internal guide Dr. Swapna Borde for noble guidance and moral support. Their valuable suggestion timely advise, inspired us towards sustain efforts for our project work.

We would like to express our gratitude towards our parents and members of Vidyavardhini's College of Engineering & Technology for their kind co-operation and encouragement which helped us in completion of this project.

Our thanks and appreciation also goes to our colleagues in developing the project and people who have willingly helped us out with their abilities.

X. REFERENCES.

- [1] Shamsul Arifin, M., Golam Kibria, M., Firoze, A., Ashraf Amini, M., & Hong Yan. (2012). Dermatological disease diagnosis using color-skin images. 2012 International Conference on Machine Learning and Cybernetics. doi:10.1109/icmlc.2012.6359626).
- [2] Jana, E., Subban, R., & Saraswathi, S. (2017). Research on Skin Cancer Cell Detection Using Image Processing. 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICIC). doi:10.1109/icic.2017.8524554.
- [3] Mhaske, H. R., & Phalke, D. A. (2013). Melanoma skin cancer detection and classification based on supervised and unsupervised learning. 2013 International Conference on Circuits, Controls and Communications (CCUBE). doi:10.1109/ccube.2013.6718539.
- [4] Alfed, N., Khelifi, F., Bouridane, A., & Seker, H. (2015). Pigment network-based skin cancer detection. 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). doi:10.1109/embc.2015.7320056.
- [5] Lau, H. T., & Al-Jumaily, A. (2009). Automatically Early Detection of Skin Cancer: Study Based on Nural Netwok Classification. 2009 International Conference of Soft Computing and Pattern Recognition. doi:10.1109/socpar.2009.80.
- [6] Dubal, P., Bhatt, S., Joglekar, C., & Patii, S. (2017). Skin cancer detection and classification. 2017 6th International Conference on Electrical Engineering and Informatics (ICEEI). doi:10.1109/iceei.2017.8312419