An Efficient Deep Learning Approach to Detect Skin Cancer

Ashfaqul Islam
Department of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
ashfaqul.islam@g.bracu.ac.bd

Daiyan Khan
Department of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
daiyan.khan@g.bracu.ac.bd

Rakeen Ashraf Chowdhury
Department of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
rakeen.ashraf.chowdhury@g.bracu.ac.bd

Md. Ashraful Alam
Department of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
ashraful.alam@bracu.ac.bd

Md. Tanzim Reza
Department of Computer Science and Engineering
BRAC University
Dhaka, Bangladesh
md.tanzim.reza@g.bracu.ac.bd

***Abstract*—In our research, we tackled the issues caused by difficulties in diagnosing skin cancer and distinguishing between different types of skin growths, especially without the use of advanced medical equipment and a high level of medical expertise of the diagnosticians. To do so, we have implemented a system that will use a deep-learning approach to be able to detect skin cancer from digital images. This paper discusses the identification of cancer from seven types of skin lesions from images using CNN with Keras Sequential API. We have used the publicly available HAM10000 dataset, obtained from the Harvard Dataverse. This dataset contains 10,015 labeled images of skin growths. We applied multiple data pre-processing methods after reading the data and before training our model. For accuracy checks and as a means of comparison we have pre-trained data. These transfer learning models include ResNet50, DenseNet121, and VGG11. This helps identify better methods of machine-learning application in the field of skin growth classification for skin cancer detection. Our model achieved an accuracy of over 97% in the proper identification of the type of skin growth.**

***Index Terms*—cancer detection; convolutional neural networks; image classification; deep learning; machine learning algorithms**

I. Introduction

Each year, millions of people around the world are affected by cancer. Research shows that early and accurate diagnosis can have a major effect on improving mortality rates from cancer. Diagnosis sometimes requires a high level of expertise and is often diagnosed completely wrong or as false-negative. Human diagnosis is prone to error, a deep-learning-based computerized diagnostic system should be considered.

Machine Learning has a subclass called Deep Learning. All of these algorithms take input data and extract information from them to classify the dataset and give predictions often along with the accuracy of the predictions. Deep learning algorithms excel at predicting results using layers and nodes especially when supervised learning is used. Supervised learning means teaching the deep learning algorithm by dividing the dataset into training and testing data.

There are about 2 to 3 million cases of skin cancer occurring globally each year with 1 out of every 3 growths being diagnosed as skin cancer. The report also states that the occurrence of cancer is increasing with each passing day [1].

Cancer treatment success depends on quick and proper diagnoses and early start of treatment. Non-melanoma cancer patients will often have a mortality rate of less than 5 percent. According to [2], the early diagnoses and treatment also apply to melanoma but since the nature of melanoma is more severe the mortality rates will be much higher.

The required number of oncologists will increase by about 40 percent whereas the number of actual oncologists will increase by only about 25 percent. The study also states that there needs to be an increase in the productivity of cancer specialists to cope with the growing demands, otherwise, the lack of doctors will start having diverse effects on the growing number of patients [3].

Although dermatologists have hands-on experience, traditional skin cancer diagnosis methods require time and might produce erroneous results. A deep learning convolutional neural network-based system, that can detect skin cancer and the type from digital image inputs, can be developed and trained with a large and diverse dataset of images of non-cancerous and cancerous skin deformities. This will help oncologists work more efficiently and accurately and may help avoid a number of cases where complications might arise from false or inaccurate results.

We have used the HAM10000 dataset (Human Against Machine with 10000 training images), which contains 10,015 images of skin lesions and growths.

We divided the dataset into an 80:20 training and testing set. We normalized the train and test sets by subtracting from their respective mean values and then dividing by their value of standard deviation. We normalized the train and test sets by subtracting from their respective mean values and then dividing by their value of standard deviation. All values from 0-255 are normalized to 0-1.

For building our model, we have used the Keras Sequential API. Models with this API are built by adding layers upon layers. Our model includes layers like Conv2D, MaxPool2D, Flatten and Dense. For yielding better results and learning rate, we have used functions like ReduceLROnPlateau, Adam optimizer, and Sparse Categorical Crossentropy.

Our model allowed us to achieve an accuracy of around 97%, where the model was run for 20 epochs.

II. Proposed Method

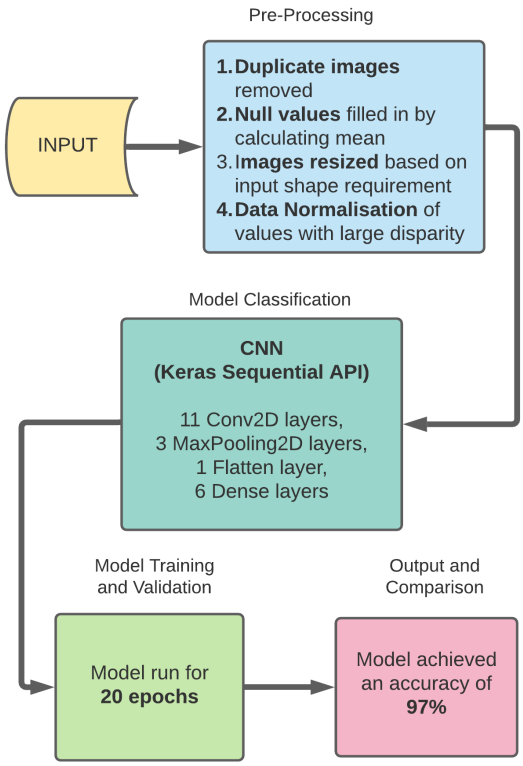

Fig. 1. Workflow of our proposed method

A. Dataset Description and Technical Terms

The HAM10000 repository contains data from multiple population categories using varied data collection methods over a period of about 20 years. Due to this large variation in data collected, many processes of "cleaning" and preparing the data were undertaken to ensure that the images could be used. The HAM10000 dataset contains a total of 10,015 dermatoscopic images. The images were either confirmed by histopathology, follow-ups, in-vivo confocal microscopy, or via expert consensus.

Each image in the dataset is classified using seven (7) unique identifying categories according to their respective medical diagnoses under the category type of "dx":

- ***akiec*** Actinic Keratoses and Intraepithelial Carcinoma (aka Bowen's Disease)
- ***bcc*** Basal Cell Carcinoma
- ***bkl*** Benign keratosis (generic class including seborrheic keratoses, solar lentigo and lichen-planus like keratoses)
- ***df*** Dermatofibroma
- ***mel*** Melanoma
- ***nv*** Melanocytic Nevi
- ***vasc*** Vascular Skin Lesions

B. Libraries Required

Multiple code-based libraries were required for the pre-processing of our dataset, the libraries we have used are as follows:

1) Python: numpy, matplotlib, seaborn, PIL,pandas, datetime, os, cv2,itertools, tqdm, glob

2) Pytorch: torch, torch.autograd,torch.utils.data, torchvision

3) Scikit-Learn: sklearn.metrics,sklearn.model_selection

4) Keras: keras.preprocessing.image, keras.callbacks, tensorflow.keras.models, tensorflow.keras.layers, tensorflow

C. Reading and Processing Data

The dataset contains a compiled CSV file (hmnist_28_28_RGB.CSV) of all the RGB data of the images, we have used this data for our model as we were aiming for our implementation to work even with very low resolution colored images. After this, we have labeled those images from 0 to 6 based on the type of cancer they represent by comparing the image file name with the data in the metadata.CSV file

D. Data Cleaning

Often, especially large datasets, have missing or null values. These have to be filled, and we have searched for and filled in these values with their mean. Duplicate images might create a bias in the outcome and yield unexpected results, hence, we have checked for duplicates and used only those images that are unique. We found 4501 unique images from our database.

```
unduplicated    5514
duplicated      4501
Name: duplicates, dtype: int64
```

Fig. 2. Check for image duplicates in our dataset

III. Model Training

A. Loading and Resizing of Images

TensorFlow cannot work with the images of the dataset directly as they have a much larger dimension (450x600x3) than what it can handle. Hence, for our model, we have used the compiled RGB data of the provided CSV file within the dataset.

B. Normalization

We normalized the train and test sets by subtracting from their respective mean values and then dividing by their value of standard deviation. All values from 0-255 are normalized to 0-1. Normalization is only necessary when features have a wide range of values, like in our case.

C. Data Augmentation

As Melanocytic nevi accounts for over half of the training data, we can observe that there is a severe class imbalance in the data. We believe we can enhance our data to tackle this challenge. To avoid the problems caused by over-fitting, we must artificially expand our HAM 10000 dataset. We can significantly increase the size of our current dataset. To reproduce the variations, the idea is to modify the preparation instructions with minor changes. Information increase strategies are procedures that vary the cluster depiction while keeping the mark the same by modifying the preparation information. Grayscales, horizontal and vertical flips, random cropping, color jitters, translations, rotations, and other common augmentations are only a few examples.

D. Train-Test Split

We divided the dataset into an 80:20 training and testing set. The training dataset's goal is to provide "ground truth" data for our calculations. Regardless, the test dataset is used to assess how successfully our computation was produced using the preparation dataset. Because the calculation will almost certainly "know" the normal yield, we cannot just repeat the training dataset in the testing step, which defeats the purpose of evaluating the computation.

E. Label-Encoding

There are seven different classes of skin cancer types on the labels, ranging from 0 to 6. The following is the indexing for data classification purposes:

Label name	nv	mel	bkl	bcc	akiec	vasc	df
Index value	0	1	2	3	4	5	6

F. Splitting Training and Validation Split

The training set was divided into two parts: a small fraction (20%) served as a validation set for the model, while the remainder (80%) was utilized to train the model. To avoid over-fitting, a validation dataset is utilized. We're not changing the network's weights using this data set; instead, we're confirming that any increase in correctness over the train set is indeed an improvement over data that the system doesn't already have, or at the very least haven't been exposed to. (i.e. validation data set). If the accuracy of the train set of data improves while the correctness of the validation set of data declines, your neural network is over-fitted, and you should cease training.

- We have re-scaled the images, which simply means to change the range of pixel values from [0, 255] to [0, 1].
- We rotated the photographs at random angles between 0 and 10 degrees, which resulted in some blank pixels that needed to be filled in. The default value for the fill mode argument is "nearest," which simply fills the empty area with the nearest pixel values. This is done so that our model may learn more when it looks at similar images from different perspectives.
- We have also shifted the images, both vertically and horizontally as realistically, images fed to the model will not always be centered
- We have also sheared the images randomly upto 20%

IV. Our Model

For building our model, we have used the Keras Sequential API. Models with this API are built by adding layers upon layers. We need to specify the input shape because the models need to know what shape of the input they will get. For our model, we have chosen not to use any transfer learning methods, instead, we opted to train our model from the ground up.

1) ***Conv2D****:* The first layer is the Conv2D layer. Conv2D is a kernel that has dimensions that are smaller than the images that pass through it. This is so that the layer can go across the image. Conv2D gives us an integer value that helps determine the total number of output filters we will receive from convolution. 32 filters in Conv2D are the most common number but we have utilized Conv2D layers that go up to 128 filters. These layers basically alter and transform the image.

2) ***MaxPool2D****:* Another layer is the pooling layer (MaxPool2D) which usually comes after a convolutional layer. The layer basically allows us to downsample (pool) images. This layer compares pixels from the input images and chooses the maximum value from them to create the downsampled image. This method is used to aid in the extraction of smooth and sharp features. These features may include edges, points, etcetera.

3) ***Flatten****:* The Flatten layer converts the images into a one-dimensional array so that each pixel of data can be connected to the next layer. It brings together all the features that were extracted in the layers together. Next, we add and pass data through dense layers. Once the images pass through the previous convolutional layers, we do not know exactly how the outputs turned out.

4) ***Dense****:* The Dense layers function is to find the connection between all the features that were fed to it without any input parameters beyond convoluted layers.

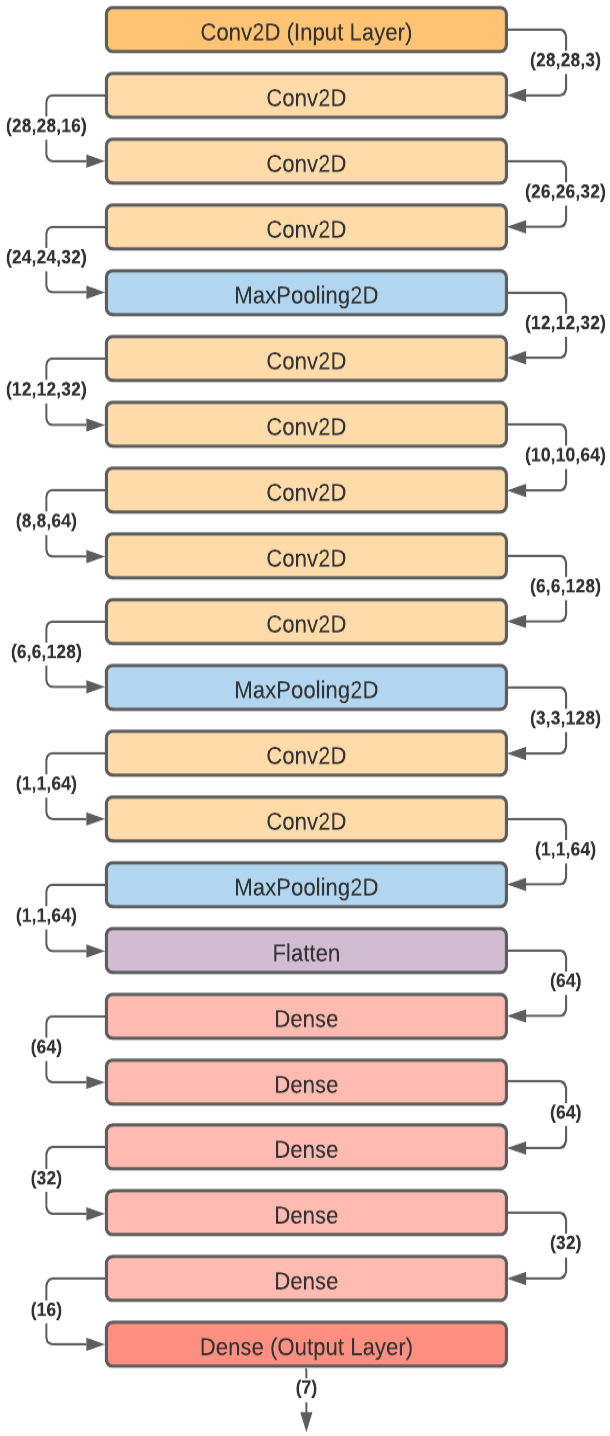

Fig. 3. Architecture of our model

A. *Annealing*

ReduceLROnPlateau: Reduce the learning rate when a metric no longer improves. Models often benefit from lowering the learning rate by a factor of 2-10 when learning becomes static. If no progress is noticed after a 'patience' number of epochs, this callback watches the quantity and slows down the learning rate.

We observed that after about the 19^{th} or 20^{th} epoch, the learning rate reduction came into play and this yielded better results.

B. *Optimizer*

We have used the **Adam** optimizer because we wanted our model to use an adaptive learning-rate optimisation mechanism.

C. *Loss Function*

Sparse Categorical Crossentropy: We used Sparse Categorical Crossentropy as our loss function, due to our classes being mutually exclusive.

V. Convolutional Neural Network

Convolutional neural networks are designed to detect patterns in pixel images. They are typically used to evaluate data that consists of a grid-like structure. CNNs are usually composed of three layers. The first layer is the convolution layer, which is usually done in the pooling layer. The second layer is the kernel, which is the representation of the receptive field. According to the stride, the kernel moves over the image's height and breadth. The activation map is a 2D representation of the picture that results from this process. The activation map shows the kernel's reaction at each picture location.

VI. Observations

Our model allowed us to achieve an accuracy of around 97%, where the model was run for 20 epochs. To compare our results with that of previously set models, we looked to other known algorithms, mainly ResNet50, DenseNet121 and VGG11. We ran the algorithms with pre-trained data from ImageNet, through the three aforementioned algorithms (using 10 epochs only due to the fact that these models are slower than our model) and recorded the accuracy of these compared to that achieved by our model. For comparison and evaluation, graphs of loss and accuracy, confusion matrices as well as precision, support and F1 scores of all the algorithms were recorded. We ascertained that the accuracy achieved by our model is higher, compared to the other 3 models used as correspondence. This will allow it to be implemented in real-world applications in the future, with intended efficacy to aid in applicable fields to work alongside human efforts.

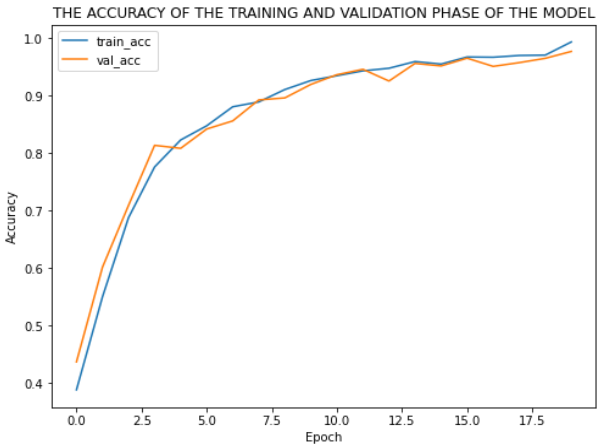

Fig. 4. Accuracy of the Training and Validation Phase of Our Model.

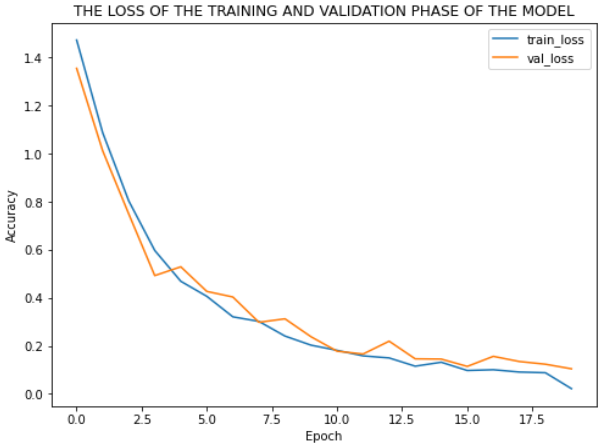

Fig. 5. Loss of the Training and Validation Phase of Our Model.

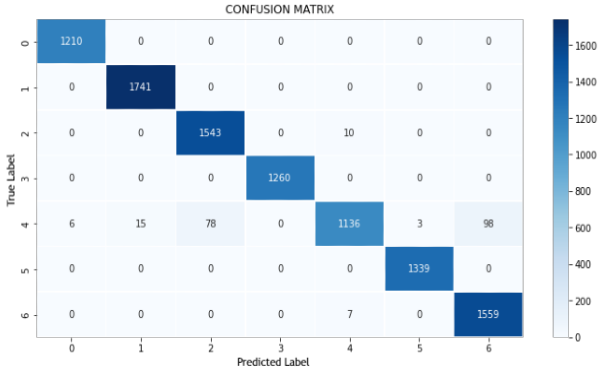

Fig. 6. Confusion Matrix of Our Model using Testing Data

TABLE I
PRECISION, RECALL AND F1 SCORE OF OUR MODEL

	precision	recall	f1-score	support
nv	1.00	1.00	1.00	1210
mel	0.99	1.00	1.00	1741
bkl	0.95	0.99	0.97	1553
bcc	1.00	1.00	1.00	1260
akiec	0.99	0.85	0.91	1336
vasc	1.00	1.00	1.00	1339
df	0.94	1.00	0.97	1566
Accuracy	-	-	0.98	10005
Macro avg.	0.98	0.98	0.98	10005
Weighted avg,	0.98	0.98	0.98	10005

TABLE II
COMPARISON OF VALIDATION OF OUR MODEL AGAINST OTHERS

	CNN (Keras Sequential	DenseNet121	ResNet50	VGG11
Accuracy	97%	90%	87%	85%

REFERENCES

[1] Skin cancer - symptoms and causes, Dec. 2020. [Online]. Available:https://www.mayoclinic.org/diseases-conditions/skin-cancer/symptoms-causes/syc-20377605.

[2] Radiation: Ultraviolet (uv) radiation and skin cancer, Oct. 2017. [Online].Available:https://www.who.int/news-room/q-a-detail/radiation-ultraviolet-(uv)-radiation-and-skin-cancer.

[3] Skin cancer survival rate, Jan. 2018. [Online]. Available:http://moffittcloud.aviddesign.com/cancers/skin-cancer-nonmelanoma/survival-rate/. networks[j],"Nature, vol. 542, no. 7639, pp. 115–118, 2017.

[4] H. Mhaske and D. Phalke, "Melanoma skin cancer detection and classificationbased on supervised and unsupervised learning," in2013 International con-ference on Circuits, Controls and Communications (CCUBE), IEEE, 2013,pp. 1–5.

[5] F. Ercal, A. Chawla, W. V. Stoecker, H.-C. Lee, and R. H. Moss, "Neural net-work diagnosis of malignant melanoma from color images,"IEEE Transactionson biomedical engineering, vol. 41, no. 9, pp. 837–845, 1994.52

[6] S. E. Umbaugh, R. H. Moss, and W. V. Stoecker, "Applying artificial in-telligence to the identification of variegated coloring in skin tumors,"IEEEengineering in medicine and biology magazine, vol. 10, no. 4, pp. 57–62, 1991.

[7] D. Piccolo, A. Ferrari, K. Peris, R. Daidone, B. Ruggeri, and S. Chimenti,"Dermoscopic diagnosis by a trained clinician vs. a clinician with minimaldermoscopy training vs. computer-aided diagnosis of 341 pigmented skin le-sions: A comparative study,"British Journal of Dermatology, vol. 147, no. 3,pp. 481–486, 2002.

[8] P. Tschandl, C. Rosendahl, and H. Kittler, "The ham10000 dataset, a largecollection of multi-source dermatoscopic images of common pigmented skinlesions,"Scientific data, vol. 5, no. 1, pp. 1–9, 2018.

[9] M. Binder, A. Steiner, M. Schwarz, S. Knollmayer, K. Wolff, and H. Peham-berger, "Application of an artificial neural network in epilumi-nescence mi-croscopy pattern analysis of pigmented skin lesions: A pilot study,"BritishJournal of Dermatology, vol. 130, no. 4, pp. 460–465, 1994.

[10] T. Mendonça, P. M. Ferreira, J. S. Marques, A. R. Marcal, and J. Rozeira,"Ph 2-a dermoscopic image database for research and benchmarking," in201335th annual international conference of the IEEE engineering in medicine andbiology society (EMBC), IEEE, 2013, pp. 5437–5440.

[11] H. Mujtaba,What is resnet or residual network: How resnet helps?Oct. 2020.[Online]. Available:https://www.mygreatlearning.com/blog/resnet/.

[12] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely con-nected convolutional networks," inProceedings of the IEEE conference oncomputer vision and pattern recognition, 2017,

pp. 4700–4708.[21] S. Gupta,Vgg-11 architecture, May 2020. [Online]. Available:https://iq.opengenus.org/vgg-11/.53

[13] S. Gupta,Vgg-11 architecture, May 2020. [Online]. Available:https://iq.opengenus.org/vgg-11/.53