REGULAR ARTICLE

Detection of Malignant Melanoma Presence by Skin Lesion Analysis using Convolutional Neural Network

Krishna Sudhir*

Correspondence:
sudhir20k@ncssm.edu
North Carolina School of Science and Mathematics, 1219 Broad St., 27705 Durham, NC
Full list of author information is available at the end of the article
*NCSSM Online Program

Abstract

Skin cancer remains one of the most common cancer types in the United States, affecting approximately every one out of five people. With significantly high incidence rates, and aggressive growth of some malignant melanomas, early detection can greatly improve the prognosis of afflicted patients. Convolutional Neural Networks have been increasingly utilized in various image processing and computer vision applications. This research project endeavors to develop a Convolutional Neural Network model that will predict a patient's chances of having skin cancer. The model was trained and tested using a dataset containing 1,802 dermoscopy images of benign cases of skin cancer and 1,499 malignant cases. The image dimensions were scaled down and gray scaled. The model was trained using 100 epochs and the cross entropy loss function. The model percent accuracy and percent loss were observed to be approximately 75% and 55% respectively. The probabilities and predictions were computed and further evaluated by visually inspecting upon 32 known cases of both benign and malignant dermoscopy images.

Keywords: Skin Cancer; Convolutional Neural Network; Malignant; Benign; Melanoma; Dermoscopy; Keras; Tensorflow; Deep learning

1 Introduction

Skin cancer is considered to be the most prevalent form of cancer in the United States. This study focuses on using Convolutional Neural Networks (CNN) as a computational approach to detect Skin cancer by using thousands of labeled images as inputs to the model.

1.1 Skin Cancer

Skin cancer is an abnormal growth of skin cells that develops most often due to over exposure to ultraviolet (UV) rays, affecting approximately 3 million Americans yearly and over 9000 patients diagnosed daily [1]. Current studies estimate that statistically, one in every five people will develop skin cancer. The following are the major types of skin cancer most commonly seen:

- Basal cell carcinoma
- Squamous cell carcinoma
- Melanoma
- Nonmelanoma Skin Cancer

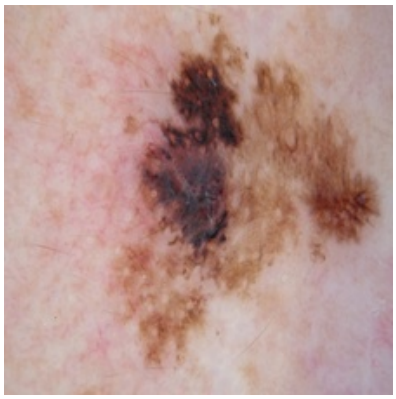

Figure 1 An Sample Image of Malignant Case Used

In general, skin cancer develops primarily on areas exposed to ultraviolet rays emitted by the sun; this includes the scalp, face, ears, neck, lips, chest, hands, arms, and on the legs. It can also form on areas that are not exposed to ultraviolet rays regularly, especially on darker toned patients, such as the palms, beneath the nails, and the genital area. Types of basal cell carcinomas usually develop in sun-exposed areas of the body. They are known to appear as flat brown bumps. Squamous cell carcinoma, similar to basal cell carcinomas, develop in sun-exposed areas of the body. Individuals with a darker complexion are at a higher risk of developing squamous cell carcinoma in areas that are not exposed to the sun. Melanomas may develop in any region of the body, and people of any complexion are susceptible to them.

Skin cancer incidence rates have continued to increase dramatically in the last three decades, and is now the most commonly diagnosed form of cancer in the United States. Skin cancer can be severely impairing with the substantial health care costs and extensive treatment plans.

Each year, nearly 5 million people are diagnosed and treated for skin cancer, inclusive of all types, and annual costs have proven to be as high as $8.1 billion, with about $3.3 billion being directed towards melanomas. Of all types of skin cancer, melanoma is responsible for most deaths from skin cancer in the United States. It is estimated that approximately 9,500 people are diagnosed with skin cancer every day, making it the most common type of cancer long adolescents in the United States [2].

Current diagnosis procedures consist of a series of examinations by a medical professional and a skin biopsy. Superficial skin cancers such as basal cell carcinomas rarely spread, and therefore are easily removed by a biopsy. However large carcinomas may require further testing to evaluate the extent of the cancer. Additional testing may include imaging tests and examinations of lymph nodes nearby in order to test for signs of cancer. This may include CT-scans, MRI scans and X-ray [3].

Early detection greatly improves the prognosis of patients with malignant melanomas, as aggressive local growth and metastasis causes approximately 75 percent of all deaths by skin cancer [4].

1.2 Types of Skin Cancer

- **Basal cell carcinomas** usually develop in sun-exposed areas of the body. They are known to appear as flat brown bumps.

- **Squamous cell carcinoma**, similar to basal cell carcinomas, develop in sun-exposed areas of the body. People with a darker complexion are more likely to develop squamous cell carcinoma on areas that aren't often exposed to the sun.

- **Melanomas** may develop in any region of the body, and people of any complexion are susceptible to this.

1.3 Artificial Neural Networks (ANN)

Neural Networks contain one input layer, which takes a set of inputs as a vector, one or more hidden layers and an output layer. The hidden layers can be fully connected, aka dense or sparsely connected. The output layer provides the prediction output and thus will have an appropriate number of levels. For example, a binary classification problem will have an output layer with just one node that would be either "zero" or "one". In case of multi-class problems with n classes will have n distinct units. Figure 6 shows typical architecture of an artificial neural network. [5]

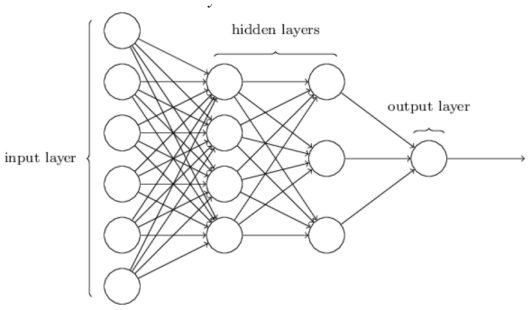

Figure 2 Typical Artificial Neural Network

1.4 Activation Function

Each neuron can be viewed as a multi-input switch which will turn on depending on the input values and the weights. How the inputs and weights trigger this is defined by a mathematical function called activation function. This activation function is analogous to the action potential of the biological neuron. Two of the commonly used activation functions are sigmoid function and the Rectified Linear Unit (ReLu).

1.5 Weights

Each neuron (perceptron) acts as a switch which operates according to a mathematical function. It can be defined as in equation 1

$$z = b + \sum_{i=1}^{m} W_i.x_i \tag{1}$$

The coefficients are known as the weights of the linear function [6].

1.6 Loss Function

In mathematical optimization and decision theory, a loss function or cost function is a function that maps an event or values of one or more variables onto a real number intuitively representing some "cost" associated with the event.

1.7 Training

During the training phase of a neural network, the weights are fine tuned through each pass or otherwise known as epoch. During the training phase, each layer creates an output which feeds forward to the next layer as its input. In the end, the output layer predicts the output, which is compared against the actual output. The variance of the predicted output to the actual outcome is defined by a loss function. The purpose of loss function is to optimize the loss function. This will lead the learning function to a local minima, where the loss function yields the lowest loss.

For each pass, the loss is used to readjust the weights using an algorithm known as backpropagation, which optimizes the weights as it optimizes the loss.

1.8 Overfitting and Regularization

One of the problems with optimization of loss during the training stage is overfitting. The model will tend to have a much higher predictive power for the training set compared to the test set. One of the ways to control overfitting is to introduce noise or error into the model itself. This method is called regularization. One of the popular methods of regularization in neural networks is introducing a penalty or LASSO method. A neural network could use L1 or L2 regularization depending on the level of constraints on weights required for preventing overfitting [7].

1.9 Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNN) are a type of deep neural networks that are applied to processing visual images. Deep learning, a subset of machine learning, utilizes a hierarchical level of artificial neural networks to carry out the process of machine learning. The artificial neural networks are built like the human brain, with neuron nodes connected together like a web. While traditional programs build analysis with data in a linear way, the hierarchical function of deep learning systems enables machines to process data with a nonlinear approach. The deep neural network has multiple layers of interconnected “neurons”, which are functions that take a set of input and turn on or off based on an activation function.

CNNs utilize feedforward artificial neural networks (ANN), or multilayer perceptrons with fully connected network of layers and “neurons”. However, they are

regularized in order to reduce the likelihood of overfitting the data. By use of image classification, CNN will analyze input images for low level features in order to develop more abstract concepts through a series of convolutional layers.

There are many different types of hidden layers used in CNN architecture:

- Convolution Layer
- Pooling Layer
- Fully Connected Layer

Convolution layer does what is knows as convolution on the input data. This layer captures essential features of the input image such as edges and corners through the process of convolution. Here, the image is scanned through a filter with specified weights and the element-wise product is summed up to convolve into a smaller set of data. The convolution layer converts an input image to a stack of convolved images [8].

Pooling is a process where the only a selected candidate from the input pixel is used. This is employed to down-sampling of the convolution layers. There are two types of pooling, namely, Max Pooling and Average Pooling. In the case of Max Pooling, only the input with the maximum value is pooled or selected within the kernel. A kernel is a small matrix (2x2 or 3x3 for example) that is moved across the image in steps. The number of steps that are moved in each step is known as stride. The kernel dimension, the stride length and the pooling type are all tuning parameters for the Pooling Layer.

Flattening is the process of converting the last convolution layer into a one dimensional layer.

Fully connected layer is also known as a dense layer. In this layer, all inputs are fully connected to all the perceptrons of the subsequent layer [6], [9].

1.10 Overfitting and Dropout

Dropout is one of the popular methods in CNN to reduce overfitting. As the name suggests, the dropout randomly eliminates the nodes such that the model doesn't closely fit the training data. Keras allows the users to choose how much dropout is required as specified by a dropout rate (a value between 0 and 1).

2 Methods

1 The data was collected from ISIC Archives and contains 1800 images of benign melanomas and malignant melanomas taken from patients. The data contains 2640 images in the training set, and 663 images in the testing set. Of the testing set, there were 361 images of benign melanomas and 301 images of malignant melanomas. In the training set, there were 1,441 images of benign melanomas and 1,198 images of malignant melanomas. The original images were of the size 224 224 with three channels (RGB). The following libraries

in the R programming language were imported and used in to conduct this study:

- ggplot2
- readr
- rlist
- keras
- EBImage
- stringr
- pbapply
- dplyr
- tensorflow

2 For this study, the images were scaled down to size 50 by 50 with one channel (Gray-scale) to reduce the computational complexity (Figure 3). For this purpose, an "extract-feature" function was created. Within the function, the images were listed in the path with only pictures of malignant and benign melanomas. The malignancy of the melanoma was used as the binary predictor variable such that malignant == 1, and benign == 0. The images were resized, and gray-scaled. The images were retrieved as matrices, coerced to a row-wise vector, and bound from a list of vectors to a matrix.

3 The function was used to process both training and testing datasets by passing the corresponding directories.

4 The training and testing data frames were rearranged into 4D tensors with dimensions of (observations x height x width x channels). A binary classifier was constructed with a single-unit output layer with sigmoid activation such that the value retrieved is the probability of a malignant melanoma P(y = Malignant).

5 Prior to training the model, the 4D tensors were rearranged by dimensions. The model was then defined and trained, while model accuracy and loss were monitored closely.

6 The model training history of accuracy and loss by each epoch were plotted and visualized.

7 32 random images from the test data set from both the benign and malignant classifications were taken and fed into the model. The predictions were then visually inspected along with the classes and associated prediction probabilities. The model was stored as an R object for future use.

3 Dataset

The data used in this research study is publicly available at Kaggle [10]. The data was originally curated from the archives of dermoscopic images from The International Skin Imaging Collaboration (ISIC) [11]. The dataset contains 1,802 images of benign cases of skin cancer and 1,499 malignant cases. This data was divided into training dataset and test dataset with 2,644 and 663 images respectively. The training set contains 1,441 benign and 1,198 malignant images whereas the test set contains 361 images of benign and 301 images of malignant samples. Each of the images is of size 224 x 224 pixels and they are of 3 channels (RGB). The images were originally captured by dermatologists using dermoscopy [12].

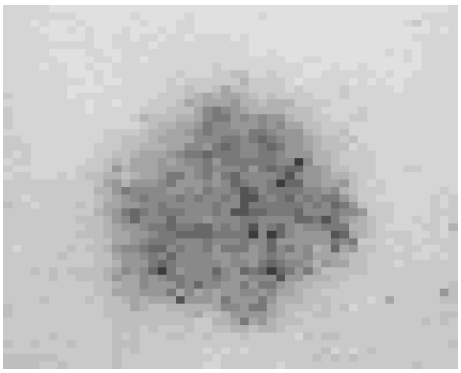

Figure 3 Scaled down version of the image (50x50x1)

4 Data Preparation

The original images are of size 224 x 224 pixels with 3 color channels. Running CNN on these images with 224x224x3 (150,528) features introduced computational complexity. To reduce the computational requirements, the images were scaled down to 50x50 pixels in size and were reduced to gray scale. Thus the number of features were reduced to 50x50x1 (2,500). Dataset was divided into training dataset with 3301 images and test set with 662 images. The images were loaded from the folders named 'malignant' and 'benign' and the folder names were transcribed into the label of the dataset as '1' for 'malignant' and '0' for 'benign'.

5 Modeling

In the CNN modeling using Keras [13], there are three steps involved, namely:

1. Decide the network architecture which determines various layers required.
2. Compiling the model with an appropriate loss function, optmizer and metrics for model fit.
3. Determime the hyperparameters such as batch size, number of epochs and validation split. Batch size determines the number of images used in each iteration. The epoch is one forward pass through the network. The validation split will be used to validate the model in each epoch. By adjusting the parameters, one can optimize the model for accuracy and loss.

5.1 Network Architecture

This study was conducted using a variant of CIFAR10 [14] network architecture implemented in Keras examples. The R code was adapted from the GitHub repository of the CNN / Keras tutorial in R [15]. The images were scaled down from 224 x 224 x 3 to 50 x 50 x 1 size. The architecture uses a series of convolutional, maxpooling and dropout layers as well as flatten and dense layers. The various layers and their parameters are given in the figure 4. The output layer is a dense layer with one output unit (binary classification) and the sigmoid activation function.

6 Results and Discussions

Convolutional Neural Networks are very computationally complex machine learning methods. Running a large number of image files in a consumer laptop can be

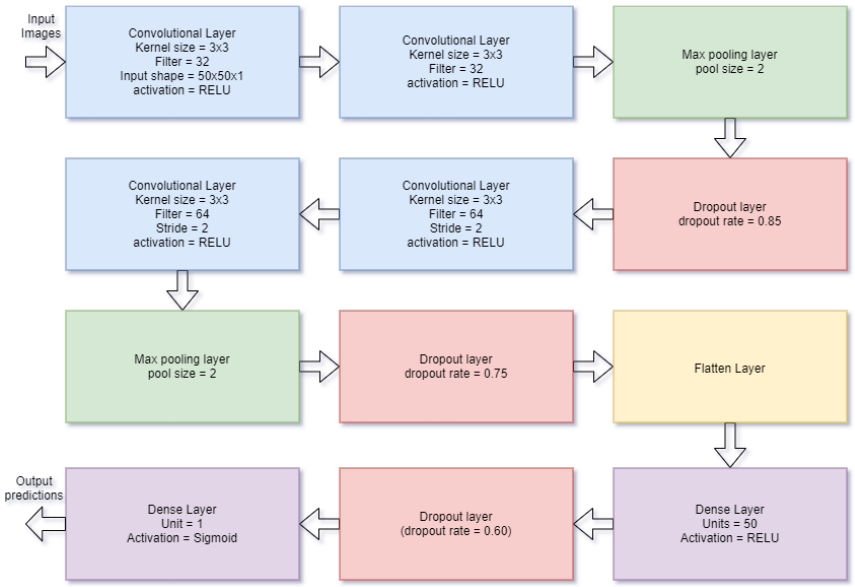

Figure 4 Convolutional Neural Network Architecture

daunting. In this study, 3,303 images collected from patient dermoscopy scans were used to train and test the model. Running for a total of 100 epochs with a batch size of 100 took approximately one hour of computational time in a consumer laptop.

The CNN training procedure runs the validation concurrently in each epoch. Until it reached around epoch 50, the accuracy was flat at around 50%. As the model training passes through each epoch, the model gradually improved accuracy and the loss was reduced. The model accuracy and loss were concurrently plotted by the Keras library. One of the benefits of using Keras is that the accuracy and the loss are tracked in parallel for the training as well as validation sets. The plots for model loss showed that the model performed equally well in training as well as validation and thus demonstrating that there was no significant overfitting. However, the accuracy plot showed significant overfitting until it reached around epoch count of 50 and there after the overfitting problem gradually disappeared. Overall after 100 epochs, the model accuracy saturated to 75% (Figure 5).

In addition to the validation model training process, a visual validation of a small sample was conducted to verify the results. For this, 32 images of benign cases and 32 images of malignant cases were separately selected and used to predict the outcome of those images using the model. The prediction results as well as the predicted probabilities were superimposed to the image grid to show how well the model performed. Figure 7 shows the outcomes predicted by the model on malignant images and Figure 8 shows the outcomes predicted by the model on the benign images. It was observed that the model accurately predicted benign melanomas with an accuracy of 75%. When the visual representation of the model accuracy in predicting malignant melanomas was generated, it was observed that the model accurately predicted benign melanomas with an accuracy of approximately 68.75%.

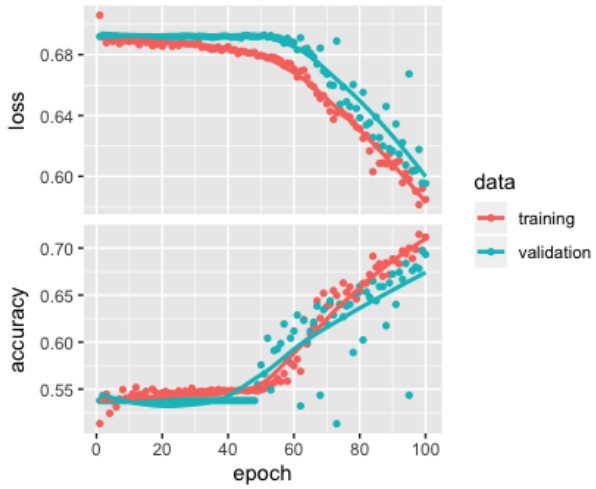

Figure 5 CNN Accuracy and Loss Plot

The greater percent accuracy may be attributed to an increased number of benign observations. It is also possible that inconsistencies in the extent of quality in the images may have varied slightly, which may also be a determinant of the model accuracy. However, because model prediction accuracy is only marginally greater for benign melanoma images, it can be considered negligible.

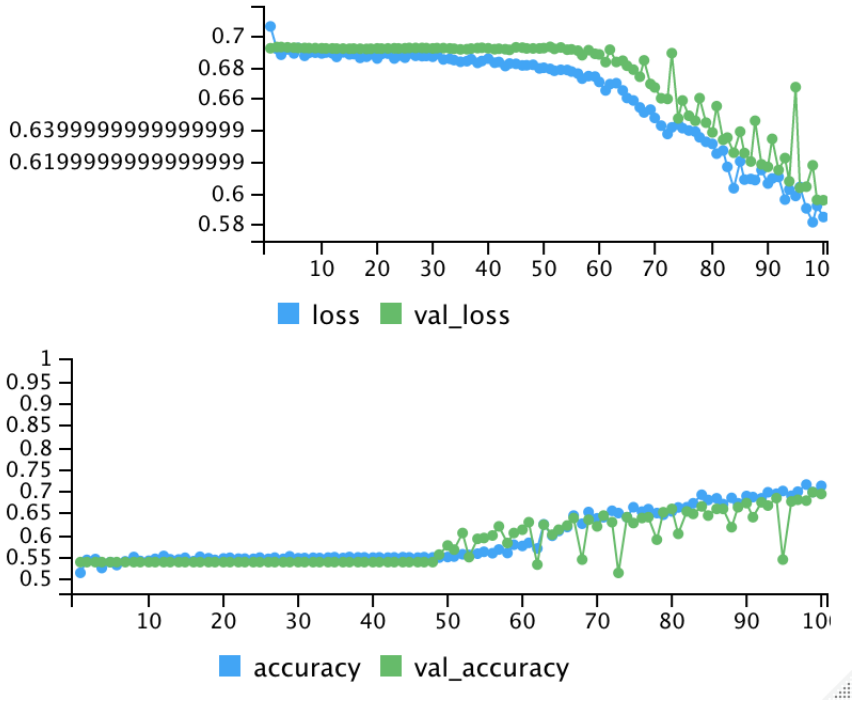

Figure 6 CNN Training History Plot

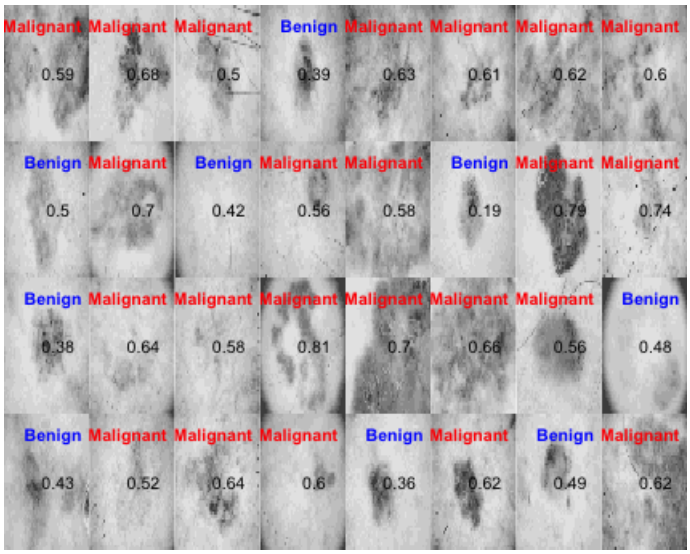

Figure 7 Prediction Results on Images of Malignant Cases

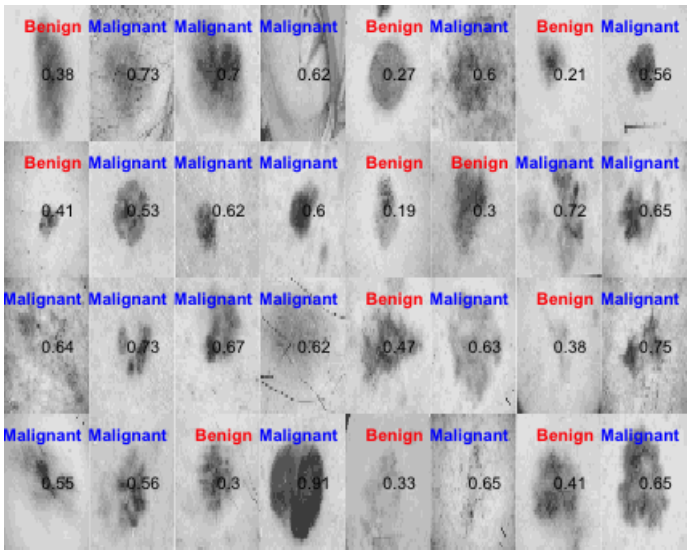

Figure 8 Prediction Results on Images of Benign Cases

There are many hyperparameters that can be adjusted to optimize the run time as well as the accuracy. A binary cross entropy loss function was used, as well as the relu activation function. With training the model, the number of epochs used had to be adjusted accordingly to the resulting model accuracy. The model percent accuracy and percent loss were reportedly approximately 75% and 60% respectively. With further hyperparameter tuning of the filtered image dimensions, activation and loss function types, number of epochs, and dropout rate, the model accuracy may have varied as well. Further investigation is highly recommended in order to identify the necessary hyperparameter tuning for the most ideal model accuracy and loss.

7 Conclusion

Convolutional Neural Network has been increasingly used in image recognition, face tagging and other image applications. This study has looked at a small set of skin cancer dermoscopy images which were labeled prior to the analysis using CNN in an attempt to classify them as 'malignant' and 'benign. The model was developed and trained. After fine-tuning the hyperparameters, the model showed an accuracy of 71% with a loss of less than 50%. The model was finetuned and the overfitting was minimised.

While the results are promising, the study needs to be extended for improving accuracy to a level so that it can be utilized for healthcare applications. This model can be improved by tweeking the model and adding more images to the dataset. In general the following steps may find useful to improve the accuracy of the model:

- Add more layers to the model architecture. Increasing the depth may potentially improve the accuracy.
- Deep learning with CNN requires far more than a few thousands of images. Adding hundreds of thousands of images can substantially improve accuracy. However, this will be computationally intensive.
- Use a higher quality image than the scaled down version of 50x50x1. It can be investigated if increasing the channels to 3, .i.e., RGB would improve the model. However this can exponentially increase the computational complexity.
- Utilize high power machines. Using Graphical Processing Units (GPU) have been found to be very effective in deep learning. This will help running the model with hundreds of thousands of images.

It can be safely concluded that Convolutional Neural Network can be effectively employed in image analysis of dermoscopy images to accurately diagnose the presence of skin cancer. It holds great promise in early and accurate detection and therefore, better healthcare outcomes.

Availability of data and materials
The data for this work was obtained from
https://www.kaggle.com/fanconic/skin-cancer-malignant-vs-benign.

Competing interests
The author declares that they have no competing interests.

Author's contributions
This paper is solely the work of the author. All references are included in the bibliography and are cited appropriately.

Acknowledgements
The author thanks Mr. Robert Gotwals for giving an opportunity and assistance to work on this interesting problem, as part of the NCSSM coursework, "Data Science for Scientists", which in itself introduced the class a wide variety of tools and practices that immensely helped to conduct this study. Appreciation is also extended to the NCSSM (North Carolina School of Science and Mathematics) for facilitating various computational sciences courses as part of the curriculum. The author is also grateful to the creators of Keras, TensorFlow, the libraries which made running CNN accessible as well as to Kaggle and ISIC for providing the dataset publicly for conducting this study.

References
1. Skin Cancer Overview. https://www.aad.org/media/stats-skin-cancer
2. Skin Cancer. https://www.aad.org/media/stats-skin-cancer
3. Diagnosing Skin Cancer.
   https://www.cancercenter.com/cancer-types/skin-cancer/diagnosis-and-detection
4. Saladi RN1, P.A.: The causes of skin cancer: a comprehensive review. Drugs Today **14**, 37–53 (2005)
5. Nielson, M.: Neural Networks and Deep Learning. neuralnetworksanddeeplearning.com, Online Book (Dec 2019)
6. Convolutional Neural Networks in R.
   https://www.r-bloggers.com/convolutional-neural-networks-in-r/
7. L1 and L2 Regularization Methods.
   https://towardsdatascience.com/l1-and-l2-regularization-methods-ce25e7fc831c
8. A Beginner's Guide To Understanding Convolutional Neural Networks. https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks/
9. Convolutional Neural Network Tutorial. https://youtu.be/FmpDIaiMIeA
10. Kaggle Dataset: Skin Cancer Malignant Vs. Benign.
    https://www.kaggle.com/fanconic/skin-cancer-malignant-vs-benign
11. The International Skin Imaging Collaboration.
    https://www.isic-archive.com/#!/topWithHeader/tightContentTop/about/isicArchive
12. Principles of Dermoscopy. https://dermoscopedia.org/Principles_of_dermoscopy
13. R Interface to 'Keras'. https://keras.rstudio.com/
14. CIFAR10 CNN Keras Example.
    https://github.com/keras-team/keras/blob/master/examples/cifar10_cnn.py
15. CNN Tutorial in R. https://github.com/monogenea/CNNtutorial