Volume: 8 Issue: VIII Month of publication: August 2020

DOI: https://doi.org/10.22214/ijraset.2020.31102

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue VIII Aug 2020- Available at www.ijraset.com

Dermatological disorder detecting Web Application using Deep Convolutional Neural Networks

Debaditya Shome
B. Tech (ETC), Department of Electronics Engineering, Kalinga Institute of Industrial Technology, Bhubaneshwar, India

Abstract: Skin diseases are one of the most common diseases in the world but still its accurate diagnosis is exceptionally challenging because of its complexities of skin tone, color, lesion size, presence of hair. This paper provides an approach to use various Deep Learning based computer vision based techniques to automatically predict the various kinds of skin diseases. The system uses a novel custom Convolutional Neural Network (CNN) model disease which successfully predicts the skin disease if it belongs to 7 classes of diseases the model has been trained on. The system consists of three phases- The feature extraction phase, the training phase and the testing /validation phase. The model makes use of deep learning technology to train itself with the various skin images from the HAM10000 open source dataset. The main objective of this model is to achieve maximum accuracy of skin disease prediction which it successfully achieves a training accuracy of 94.81% and test accuracy of 80.52%. This model is further deployed as open source software using a Tensorflow.js.

Keywords: Computer Vision, Deep Learning, Image Recognition, Skin Disease, Dermatology

I. INTRODUCTION

Dermatological disorder detection still remains one of the most error prone and complex branch in healthcare due to human skin being most uncertain and troublesome terrains due to existence of deviations, hair and many other mitigating factors. Diagnosis of skin diseases is a series of laboratory tests but generally people and doctors take things lightly and when the Skin lesion gets severe then only the tests start, eventually making very hard for the person to survive. We human beings in our busy lives don't care about a small skin lesion and start applying our own home remedies, but it would be great to have an online test for detection which predicts if the skin lesion is serious or not. This paper proposes a novel technology which would help diagnose and predict 7 classes of skin diseases in a few seconds.

The classes of lesion that our model can classify are Melanocytic nevi, Melanoma, Benign keratosis, Basal Cell Carcinoma, Actinic keratoses, Vascular lesions, Dermatofibroma.

Melanocytic Nevi is found one in every 100 infants being born with moles, this if found malignant can be really harmful, our model detects Malignant Melanocytic nevi to make sure the cancerous lesion gets treated at the first stage.

Melanoma is the deadliest form skin cancer which takes several lives every year. According to www.cancer.org, About 100,350 new melanomas will be diagnosed (about 60,190 in men and 40,160 in women). About 6,850 people are expected to die of melanoma (about 4,610 men and 2,240 women). Melanoma is curable once treated at an early stage hence our model's diagnosis would help a lot in this regard.

Seborrheic/Benign Keratosis are non-cancerous lesions on skin and our model can detect that for easy removal by doctors as early as possible.

Basal Cell Carcinoma starts with a small bump that looks like a pimple that doesn't go away, that is the reason most people neglect it at early stages. According to www.cancer.net, every year almost 2000 people die from Basal cell Carcinoma and most of the cases are due to being too late for treatment.

Actinic keratoses involves scaly spots/patches on topmost layer of skin generally caused by sun exposure damage. According to www.skincancer.org, 10% of these cases become cancerous without early detection and is usually removed as a precaution.

Vascular lesions on top layer of the human skin are not dangerous but can be detected by our model so as to make sure that it's not any other cancerous / risky lesion. They can mostly be cured.

Dermatofibroma are common to occur in adults in the age range 20 to 49. Mostly this is benign but one it's variant called cellular dermatofibroma has a high recurrence rate after initial surgical excision due to which sometimes it poses risks and thus in this case also early detection is necessary.

Computer Vision requires an effective segmentation algorithm and image data pre-processing to efficiently detect pixels of the skin lesions in those images. The images of 7 type of skin lesion disorders have been taken from the HAM10000 (Human Against Machine with 10000 training images) dataset available at Harvard dataverse v2 which consists of 10015 images [1].

It is also available at Kaggle as Skin MNIST: HAM10000. More than 50% of lesions are confirmed through histopathology (histo), the ground truth for the rest of the cases is either follow-up examination (follow-up), expert consensus (consensus), or confirmation by in-vivo confocal microscopy (confocal). The dataset includes lesions with multiple images, which can be tracked by the lesion_id-column within the HAM10000_metadata file.
The model proposed is a deep learning CNN model [2]. Deep learning is a small subset of machine learning in which the model is trained on a large amount of data and number of classifiers get reduced substantially.
Convolutional Neural Network (CNN/ConvNet) is a type of Deep learning architecture which takes in an input image, assigns weights and biases to different parts of the images. The architecture of the CNN is similar to that of visual cortex due to the confined visual field in between CNN layers resembling to the receptive fields of Biological neurons in the visual cortex [3].
A deep ConvNet/CNN is a model architecture where there is more than one hidden layer. As we go deeper, the model starts to learn more complex features thus increasing accuracy until it overfits the data available.

II. METHODOLOGY

A. Proposed Method

1) Custom deep-CNN Architecture

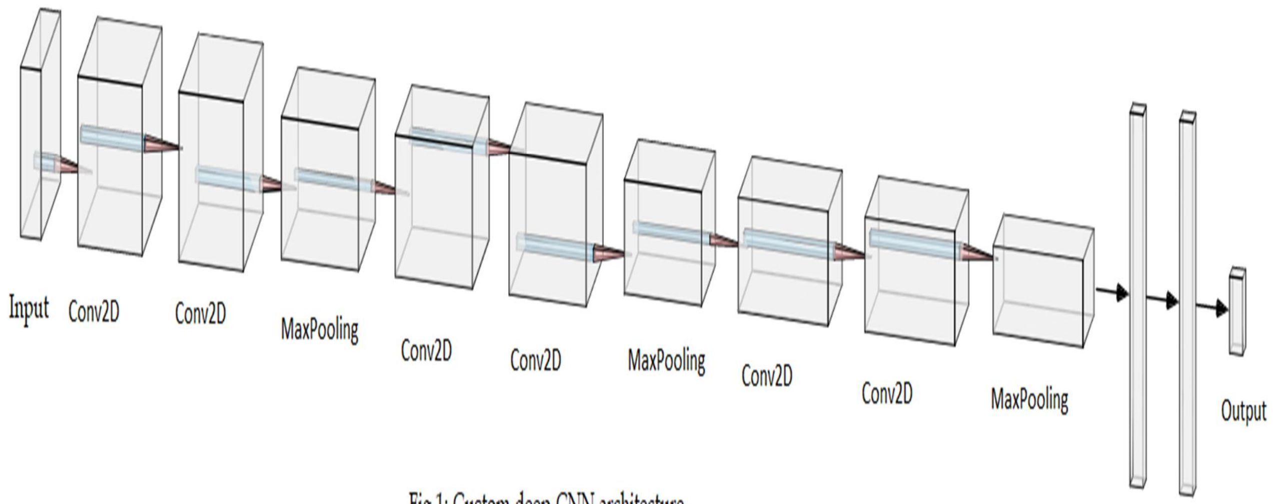

Fig 1: Custom deep-CNN architecture

Figure 1 illustrates our model architecture. With 385,575 parameters including 384,679 trainable parameters and 896 non-trainable parameters, our deep Convolutional Neural Network model consists of 6 Convolutional layers with 3 Max Pooling layers and 3 dropout layers in between followed by a Flatten and a Dense layer at the end.
The Input image size for this model is 75x100x3 and output consists of 7 classes of skin disorders.

a) *First Convolutional Block:* It consists of a 2D Convolutional layer with 896 parameters which has a ReLU (Rectified Linear Unit) activation function and Batch normalization layer 1.
b) Second Convolutional block: It consists of a 2D Convolutional layer with 9248 parameters which has a ReLU (Rectified Linear Unit) activation function and Batch normalization layer 2.
c) *First Max Pooling Block:* It has a 2D Max Pooling layer with pool size of 2x2 followed by a dropout layer with dropout rate of 20 %.
d) *Third Convolutional Block:* Starts with a 2D Convolutional layer with 18496 parameters which has a ReLU (Rectified Linear Unit) activation function and Batch normalization layer 3.

e) *Fourth Convolutional Block:* It has 2D Convolutional layer with 36928 parameters which has a ReLU (Rectified Linear Unit) activation function and Batch normalization layer 4.
f) *Second Max Pooling Block:* It has a 2D Max Pooling layer with pool size of 2x2 followed by a dropout layer with dropout rate of 50 %.
g) *Fifth Convolutional Block:* It consists 2D Convolutional layer with 73856 parameters which has a ReLU (Rectified Linear Unit) activation function and Batch normalization layer 5.
h) *Sixth Convolutional Block:* It starts with a 2D Convolutional layer with 147584 parameters which has a ReLU (Rectified Linear Unit) activation function and Batch normalization layer 6.
i) *Third Max Pooling Block:* It has a 2D Max Pooling layer with pool size of 2x2 followed by a dropout layer with dropout rate of 50 %.
j) *Fully Connected Layer:* This is the last layer which starts with a Flatten layer followed by a Dense layer with 96775 parameters which has a SoftMax activation function.

2) *Loss function and Optimizer:* In Deep learning, the goal is to bridge the gap between predictions and clinical real-world results. This gap or difference is referred to as Loss function. Our goal hence is to optimize the loss function by finding the optimized value of weights. Our model was compiled using the Categorical cross entropy loss, which is the loss function used in case of multi-class classification tasks. The optimizer we use here to compile our model is RMSprop optimizer which uses the magnitude of recent gradients for normalizing the gradients. In RMSprop we always keep a moving average over the root mean squared (hence RMS) gradients by which the current gradients get divided.

3) *Process of Training the Model*

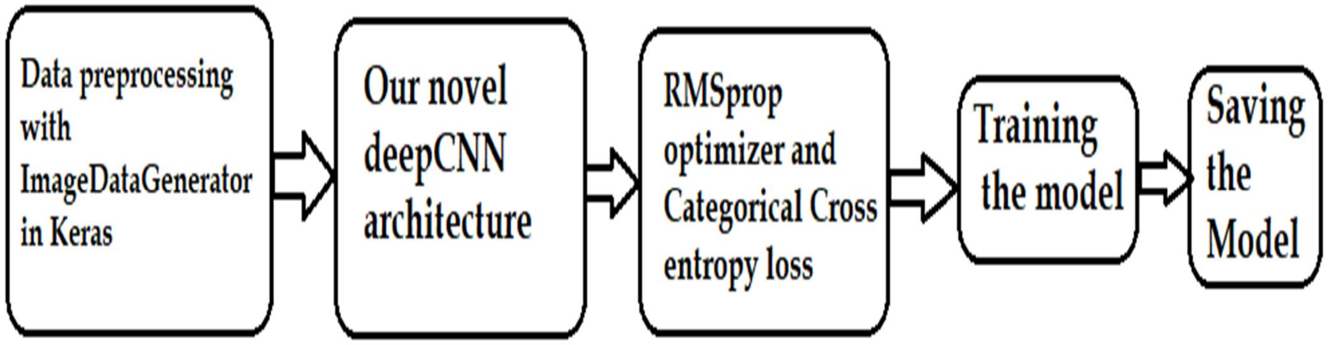
Data preprocessing with ImageDataGenerator in Keras
Our novel deepCNN architecture
RMSprop optimizer and Categorical Cross entropy loss
Training the model
Saving the Model

Figure 2: Training Process

III.SETUP AND RESULTS

A. Setup

The deep-CNN model for the multiclass dermatological disorder classification was done using custom made Novel architecture which is mentioned above.
Everything from data pre-processing to model training to inference was done in the Jupyter Notebook environment in a TensorFlow GPU kernel. The main packages used were OpenCV for image pre-processing, TensorFlow GPU and Keras GPU for training and inference.
Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, R, Theano, or PlaidML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible [4].
The images are pre-processed using the ImageDataGenerator from keras.preprocessing package. Then the model is compiled using RMSprop optimizer and Categorical Cross-entropy loss. We chose learning rate of 0.001 and weight decay of 1e-6 and
Batch size of 64.

The model summary is given below in Table 1:

TABLE I. Designed model summary

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 75, 100, 32)       896
_________________________________________________________________
activation_1 (Activation)    (None, 75, 100, 32)       0
_________________________________________________________________
batch_normalization_1 (Batch (None, 75, 100, 32)       128
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 75, 100, 32)       9248
_________________________________________________________________
activation_2 (Activation)    (None, 75, 100, 32)       0
_________________________________________________________________
batch_normalization_2 (Batch (None, 75, 100, 32)       128
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 37, 50, 32)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 37, 50, 32)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 37, 50, 64)        18496
_________________________________________________________________
activation_3 (Activation)    (None, 37, 50, 64)        0
_________________________________________________________________
batch_normalization_3 (Batch (None, 37, 50, 64)        256
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 37, 50, 64)        36928
_________________________________________________________________
activation_4 (Activation)    (None, 37, 50, 64)        0
_________________________________________________________________
batch_normalization_4 (Batch (None, 37, 50, 64)        256
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 18, 25, 64)        0
_________________________________________________________________
dropout_2 (Dropout)          (None, 18, 25, 64)        0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 18, 25, 128)       73856
_________________________________________________________________
activation_5 (Activation)    (None, 18, 25, 128)       0
_________________________________________________________________
batch_normalization_5 (Batch (None, 18, 25, 128)       512
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 18, 25, 128)       147584
_________________________________________________________________
activation_6 (Activation)    (None, 18, 25, 128)       0
_________________________________________________________________
batch_normalization_6 (Batch (None, 18, 25, 128)       512
_________________________________________________________________
```

```
max_pooling2d_3 (MaxPooling2 (None, 9, 12, 128)       0
_________________________________________________________________
dropout_3 (Dropout)          (None, 9, 12, 128)       0
_________________________________________________________________
flatten_1 (Flatten)          (None, 13824)            0
_________________________________________________________________
dense_1 (Dense)              (None, 7)                96775
=================================================================
```

B. Results and Inference

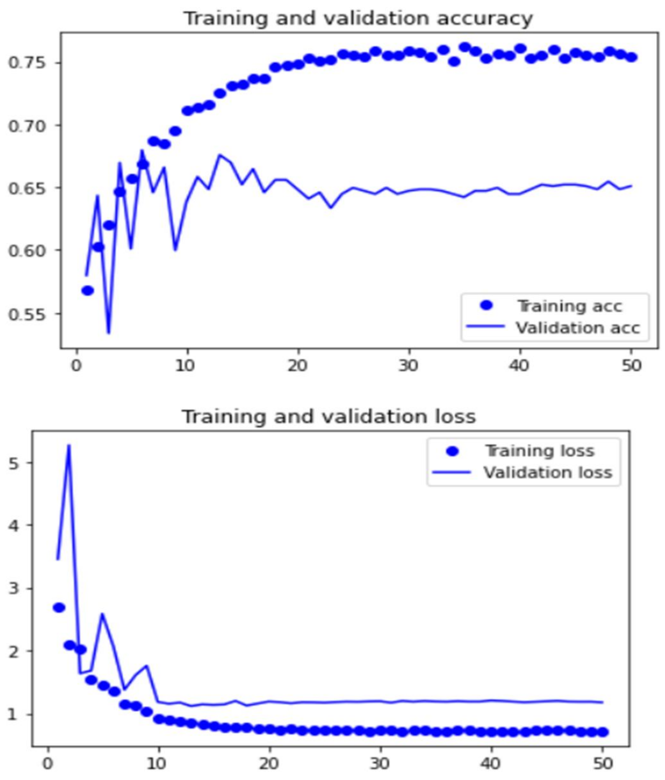

Figure 3: Results

Figure 3 shows our model's performance on training and validation sets which came out to be 94.81% and 80.52% of training and validation accuracy respectively.
The predictions are printed in form of an array of probabilities of the diseases in the order:
['Melanocytic nevi', 'Melanoma', 'Benign keratosis-like lesions ', 'Basal cell carcinoma', 'Actinic keratoses', 'Vascular lesions', 'Dermatofibroma'] along with the name of the disease with highest probability.

C. Deploying as a Web application using TensorFlow.js
Getting a Deep learning model to production and be of real value to the society is a challenging task. For this task TensorFlow.js remains one of the best choices. TensorFlow.js is an open-source hardware-accelerated JavaScript library for training and deploying machine learning models [5]. This library has a model converter called TensorFlow.js converter which converts your TensorFlow/keras saved models to a JSON model and some binary files which can would be used by TensorFlow.js to load and run the model in a browser. Next, we just need to an html page ready for our web application and add an JavaScript file which has all the TensorFlow.js script and a function to render the predictions to a specific HTML tag.

IV. CONCLUSIONS AND FUTURE SCOPE

Highly accurate detection of Dermatological disorders is a highly complex task. In this paper we showed a novel deep ConvNet model used to perform multiclass classification on 7 classes of skin disorders/lesions with training accuracy of 94.81% and validation accuracy as high as 80.52%.

This model performs great in research environment but after deployment the accuracy seems to get low, so Future scope of this paper would be training the model further on more versatile images of different sizes and different parameters so as be another step closer to making this available to be of healthcare department's use and of value to the people with skin disorders.

REFERENCES

[1] Tschandl, P., Rosendahl, C. & Kittler, H. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. Sci. Data 5, 180161 (2018).
[2] Yamashita, R., Nishio, M., Do, R.K.G. et al. Convolutional neural networks: an overview and application in radiology. Insights Imaging 9, 611–629 (2018).
[3] Cichy, R., Khosla, A., Pantazis, D. et al. Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence. Sci Rep 6, 27755 (2016).
[4] Chollet, F. & others, 2015. Keras. Available at: https://github.com/fchollet/keras.
[5] TensorFlow.js , Available at https://github.com/tensorflow/tfjs