SKIN CANCER DETECTION USING CNN

A CAPSTONE PROJECT REPORT

Submitted in partial fulfillment of the
requirement for the award of the
Degree of

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING

by

VIRGIL K (19BCD7158)
POTHAMSETTI SAMPATH (19BCE7443)
PEDIREDDY V SRI SAI CHARAN (19BCE7663)

Under the Guidance of

DR. RAJESH KOLLURI

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
VIT-AP UNIVERSITY
AMARAVATI- 522237

CERTIFICATE

This is to certify that the Capstone Project work titled "**SKIN CANCER DETECTION USING CNN**" that is being submitted by **VIRGIL K**

(19BCD7158), POTHAMSETTI SAMPATH (19BCE7443), PEDIREDDY V SRI SAI CHARAN (19BCE7663)

is in partial fulfillment of the requirements for the award of Bachelor of Technology, is a record of bonafide work done under my guidance. The contents of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for award of any degree or diploma and the same is certified.

Dr. Rajesh Kolluri

Guide

The thesis is satisfactory / unsatisfactory

Internal Examiner **External Examiner**

Approved by

PROGRAM CHAIR **DEAN**

B. Tech. CSE School Of Computer Science and Engineering

ACKNOWLEDGEMENTS

INDEX

ABSTRACT

Skin cancers are cancers that arise from the skin. They are due to the development of abnormal cells that can invade or spread to other parts of the body. There are three main types of skin cancers: basal-cell skin cancer (BCC), squamous-cell skin cancer (SCC) and melanoma. The first two, along with several less common skin cancers, are known as nonmelanoma skin cancer (NMSC). More than 90% of cases are caused by exposure to ultraviolet radiation from the Sun. This exposure increases the risk of all three main types of skin cancer. Exposure has increased, partly due to a thinner ozone layer. For melanomas and basal-cell cancers, exposure during childhood is particularly harmful. For squamous-cell skin cancers, total exposure, irrespective of when it occurs, is more important. Between 20% and 30% of melanomas develop from moles. People with lighter skin are at higher risk.as are those with poor immune function Diagnosis is by biopsy. In this Project we will be testing out different Convolutional Neural Networks Architectures which extract features of the images from the dataset and classify them. Whichever Architecture gets the best Accuracy we will deploy that model in a website and host it using an API service so that patients and doctors all around the world can use it and detect skin lesions.

INTRODUCTION

Skin cancer is the most generic form of cancer, globally accounting for at least 40% of cancer cases. The most common type is nonmelanoma skin cancer, which occurs in at least 2–3 million people per year. This is a rough estimate, however, as good statistics are not kept. Of nonmelanoma skin cancers, about 80% are basal-cell cancers and 20% squamous-cell skin cancers. Basal-cell and squamous-cell skin cancers rarely result in death. In the United States, they were the cause of less than 0.1% of all cancer deaths. Globally in 2012, melanoma occurred in 232,000 people and resulted in 55,000 deaths. White people in Australia, New Zealand and South Africa have the highest rates of melanoma in the world. The three main types of skin cancer have become more common in the last 20 to 40 years, especially regions where the population is predominantly White.

The Convolutional Neural Network is a subtype of Neural Networks that is mainly used for applications in image and speech recognition. Its built-in convolutional layer reduces the high dimensionality of images without losing its information. That is why CNNs are especially suited for this use case.

In CNN, we take an image as an input, assign importance to its various aspects/features in the image and be able to differentiate one from another. The pre-processing required in CNN is much lesser as compared to other classification algorithms.

We will be testing CNN Architectures such as Xception ,Renet152, DenseNet201 .

We will then deploy the best model (in terms of accuracy) in the website and host it using an API service.

Conventionally It takes about 2 to 3 weeks to get the results of a patient's biopsy which is a lot.

Doctors and Patients around the world can use our website to Upload the image of the patient's skin lesion and our CNN model will classify the lesion immediately which is quite useful rather than waiting weeks for diagnosis.

Problem Statement

Skin cancer is an alarming issue and it must be detected as early as possible. The diagnostic is a manual process that is time consuming as well as expensive. But, today’s world science has become advanced by using machine learning and it can be helpful in many ways. Hence, machine learning can make easy for detecting cancerous cells and that is why machine learning specially convolutional neural network is used to detect cancerous cell more quickly, and Efficiently

Requirements

Google Colab pro

Python 3.9

Deep learning

Tensorflow

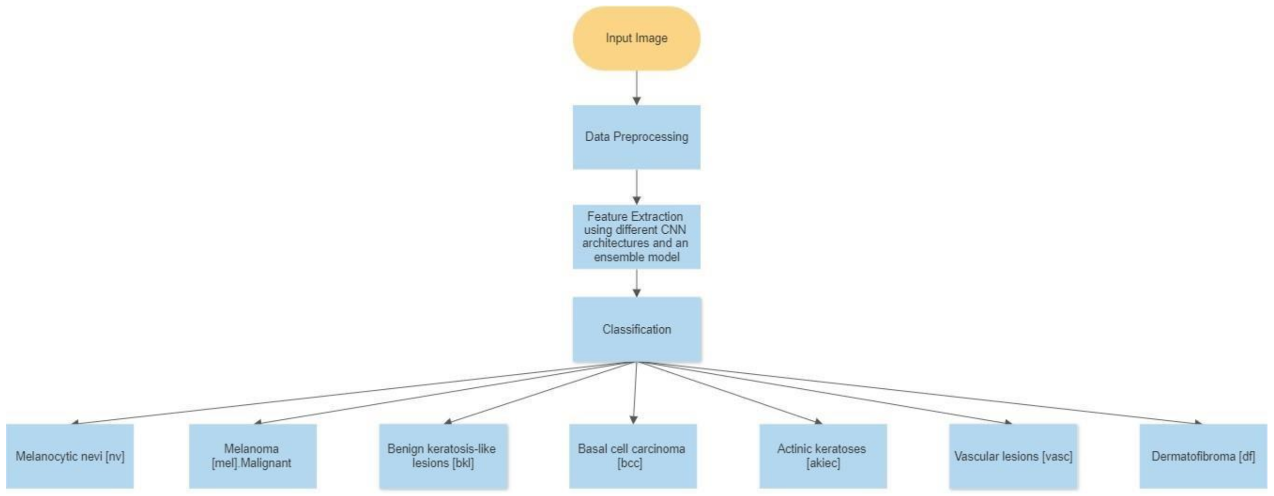
Ngrok Account

Ham10000 Dataset

Background and Literature Survey

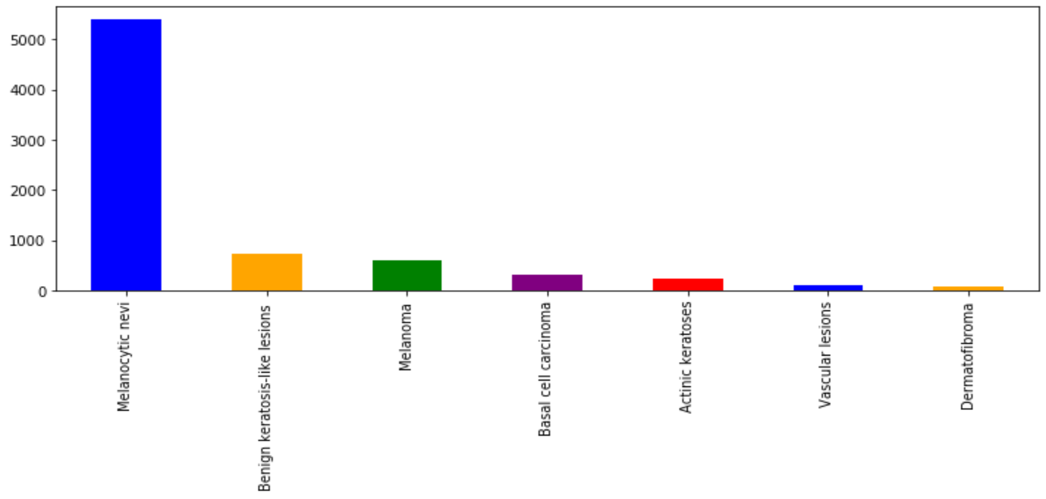
In the diagnosis of skin melanoma by analyzing histopathological images, the detection of the melanocytes in the epidermis area is an important step. However, the detection of melanocytes in the epidermis area is difficult because other keratinocytes that are very similar to the melanocytes are also present. This paper uses () computer-aided technique for segmentation of the melanocytes in the skin histopathological images. To reduce the local intensity variant, a mean-shift algorithm is applied for the initial segmentation of the image. A local region recursive segmentation algorithm is then proposed to filter out the candidate nuclei regions based on the domain prior knowledge. To distinguish the melanocytes from other keratinocytes in the epidermis area, a novel descriptor, named local double ellipse descriptor (LDED), is proposed to measure the local features of the candidate regions. The LDED uses two parameters: region ellipticity and local pattern characteristics to distinguish the melanocytes from the candidate nuclei regions. Experimental results on 28 different histopathological images of skin tissue with different zooming factors show that the proposed technique provides a superior performance.

Working Methodology


Input Image
Data Preprocessing
Feature Extraction using different CNN architectures and an ensemble model
Classification
Melanocytic nevi [nv]
Melanoma [mel] Malignant
Benign keratosis-like lesions [bkl]
Basal cell carcinoma [bcc]
Actinic keratoses [akiec]
Vascular lesions [vasc]
Dermatofibroma [df]

Dataset Details

- The CNN model is Trained using HAM10000.It consists of 10015 Images which will be converted as Training, testing and validation sets for our model.
- Exploratory Data Analysis (EDA)



- The 7 different diagnostic skin lesion categories to be predicted are:
- **1)Melanocytic nevi [nv]**
- it can be a benign or rarely Malignant lesion and there are about 6705 images
- **2)Melanoma [mel] Malignant**
- it is a Malignant lesion and there are about 1113 images
- **3)Benign keratosis-like lesions [bkl]**
- it is a benign lesion and there are about 1099 images
- **4)Basal cell carcinoma [bcc]**
- it is a Malignant lesion and there are about 514 images
- **5)Actinic keratoses [akiec]**
- it is a benign lesion and there are about 327 images • **6)Vascular lesions [vasc]** it is a benign lesion and there are about 142 images
- **7)Dermatofibroma [df]**

it is a benign lesion and there are about 115 images

- The focus has to be on the malignant pigment lesions, which include melanoma, the deadliest type of skin cancer, basal cell carcinoma, and some vascular lesions.

Dataset Preprocessing

Preprocessing refers to all the transformations on the raw data before it is fed to the deep learning algorithm. For instance, training a convolutional neural network on raw images will lead to bad classification performances.

a) Image Resizing
The size of the images in the dataset is 600x450, which is ridiculously huge, so we resize the images to 256x256 RGB images for normal CNN model, and 256x256 for Xception ,Resnet152, DenseNet201 .

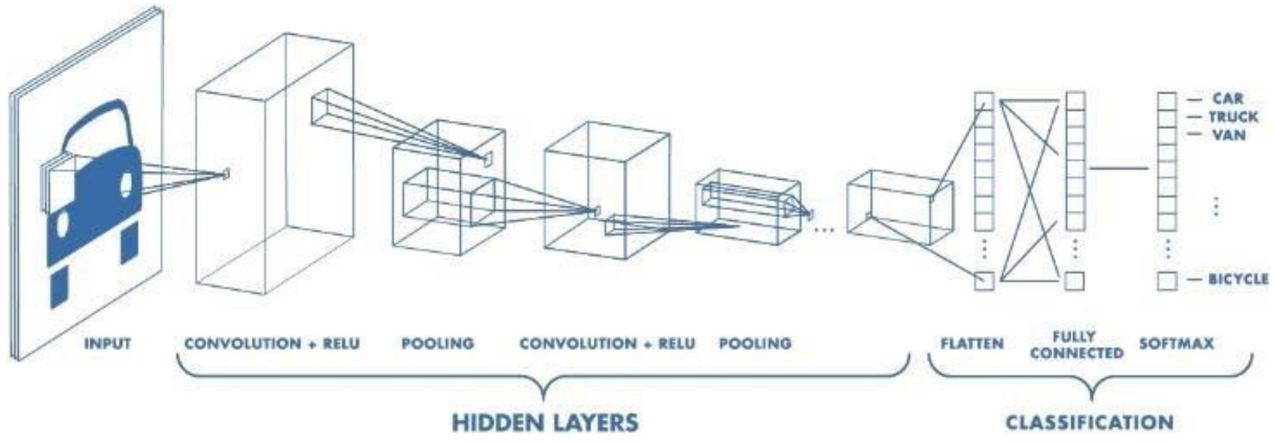
b) Data Augmentation
Data augmentation is a technique to artificially create new training data from existing training data. This is done by applying domain-specific techniques to examples from the training data that create new and different training examples.

Image data augmentation is the most well-known type of data augmentation and involves creating transformed versions of images in the training dataset that belong to the same class as the original image.

Transforms include a range of operations from the field of image manipulation, such as shifts, flips, zooms, and much more.

C) Training, Validation, and Test Split
The dataset will be split into 70% training examples, 10% validation examples, and 20% testing and variable epochs for different models.

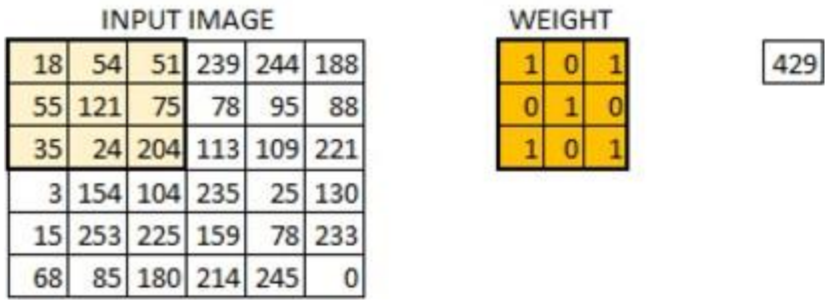
2)Feature Extraction and Classification Convolutional Neural Networks



A Convolutional Neural Network, also known as CNN is a class of neural networks that specializes in processing data that has a grid-like topology, such as an image. A digital image is a binary representation of visual data. It contains a series of pixels arranged in a grid-like fashion that contains pixel values to denote how bright and what color each pixel should be.

A CNN consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of convolutional layers, pooling layers, fully connected layers, and normalization layers.

What is Convolutional Layer?



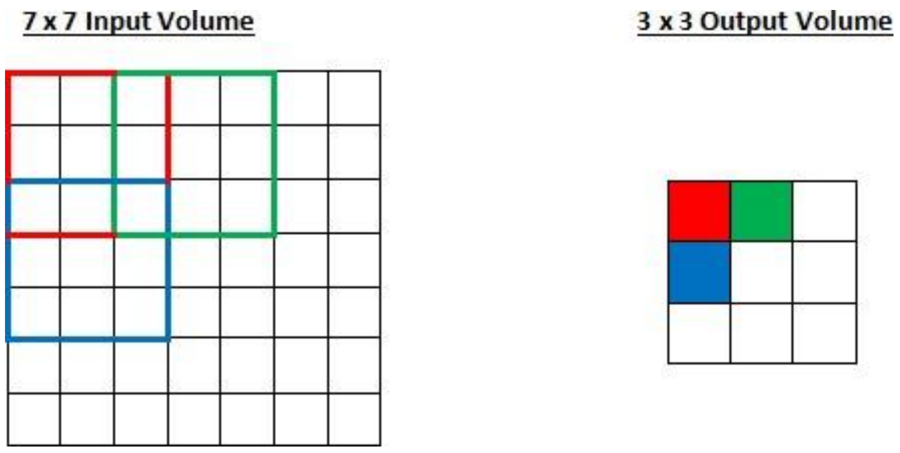
18	54	51	239	244	188
55	121	75	78	95	88
35	24	204	113	109	221
3	154	104	235	25	130
15	253	225	159	78	233
68	85	180	214	245	0

1	0	1
0	1	0
1	0	1

Convolutional Layers have a moving filter also called as the weight matrix. The filter slides over the input image (convolution operation) to produce a feature map.

Weight matrix and Input Image multiply (dot product) and summed to produce the feature map

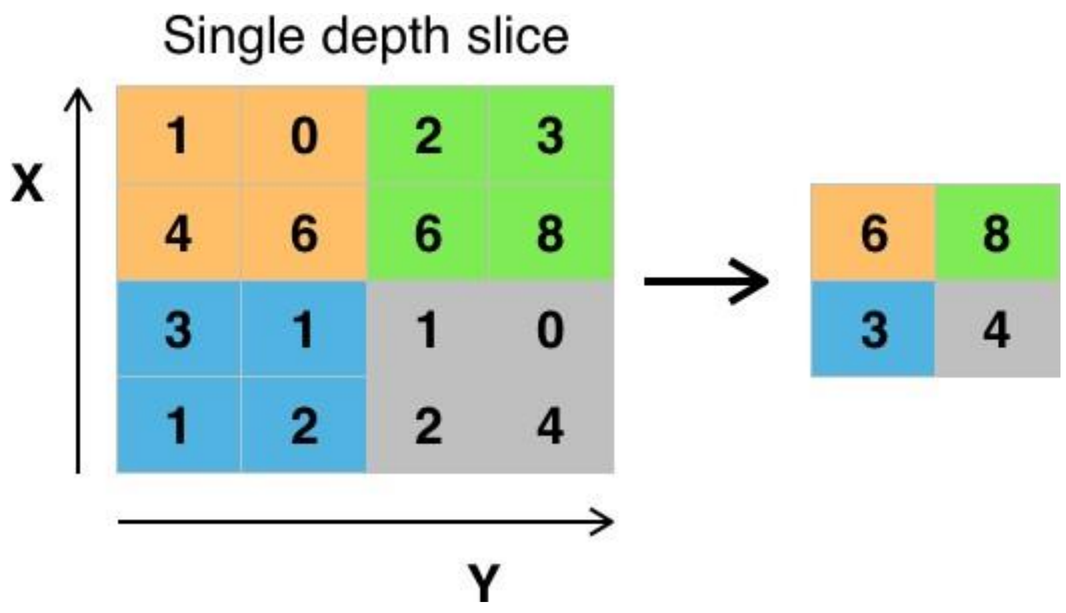
What are Strides?



Stride decide how our weight matrix should move in the input, i.e, jumping one step or two.

What is Max Pooling layer?

We move a window across a 2D input space, where the maximum value within that window is the output.



What is Fully Connected Layer?

In Fully Connected layer all neurons are connected to all neurons of the previous layer. After feature extraction we need to classify the data into various classes, this can be done using a fully connected neural network.

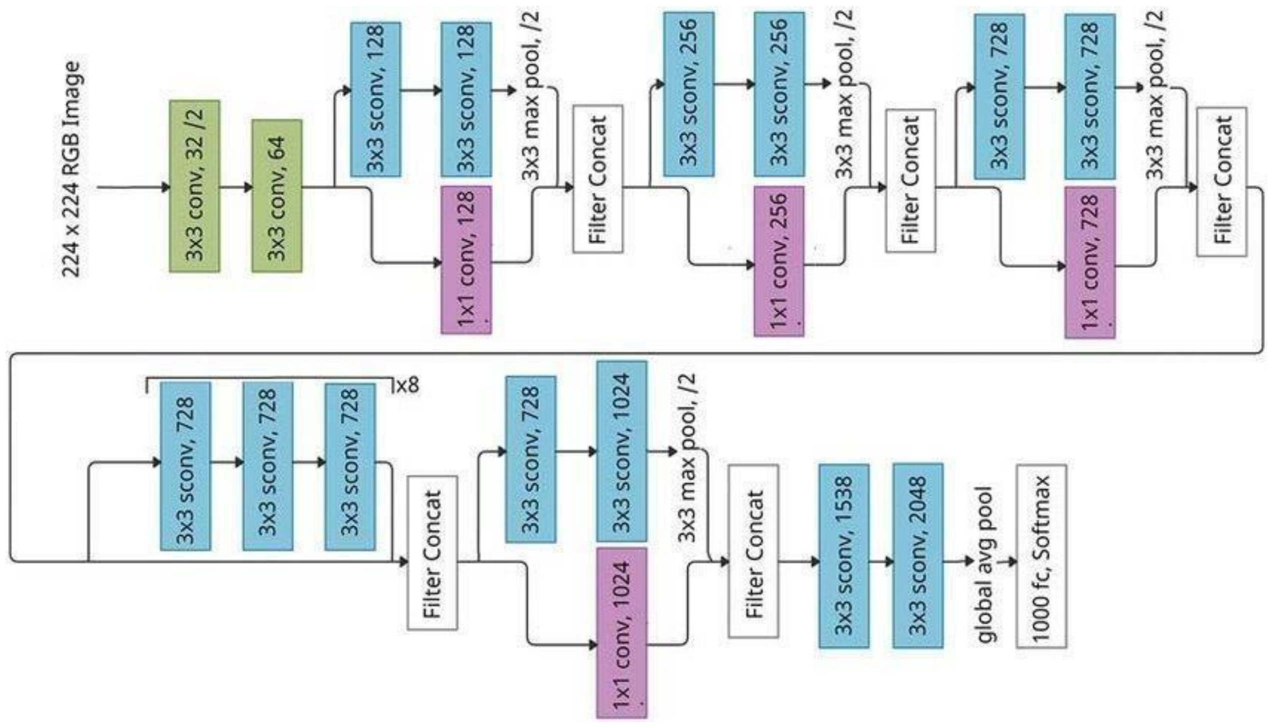
What is Softmax Layer?

Softmax Layer returns the probabilities of each class and the class with higher probability will be the target class.

CNN Architectures Xception

Xception stands for “extreme inception”, it takes the principles of Inception to an extreme. In Inception, 1x1 convolutions were used to compress the original input, and from each of those input spaces we used different type of filters on each of the depth space. Xception just reverses this step. Instead, it first applies the filters on each of the depth map and then finally compresses the input space using 1X1 convolution by applying it across the depth. This method is almost identical to a depthwise separable convolution, an operation that has been used in neural network design as early as 2014. There is one more difference between Inception and Xception. The presence or absence of a non-linearity after the first operation. In Inception model, both operations are followed by a ReLU non-linearity, however Xception doen't introduce any non-linearity.

Xception offers an architecture that is made of Depthwise Separable Convolution blocks +
Maxpooling, all linked with shortcuts as in ResNet implementations

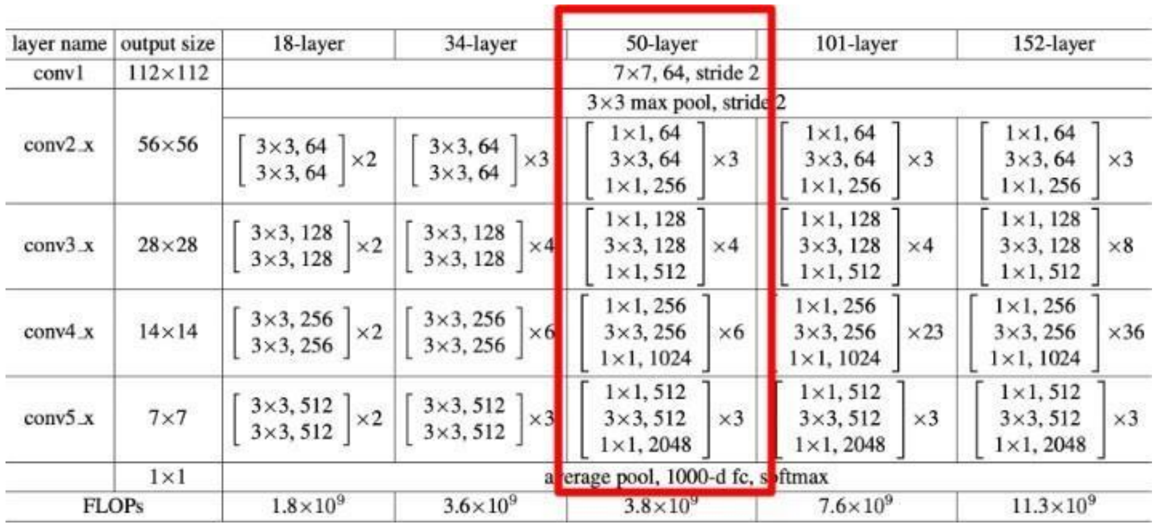


Resnet152

A residual neural network is an artificial neural network. It is a gateless or open-gated variant of the HighwayNet, the first working very deep feedforward neural network with hundreds of layers, much deeper than previous neural networks. Skip connections or shortcuts are used to jump over some layers.

ResNet can have a very deep network of up to 152 layers by learning the residual representation functions instead of learning the signal representation directly.

ResNet introduces skip connection (or shortcut connection) to fit the input from the previous layer to the next layer without any modification of the input. Skip connection enables to have deeper network



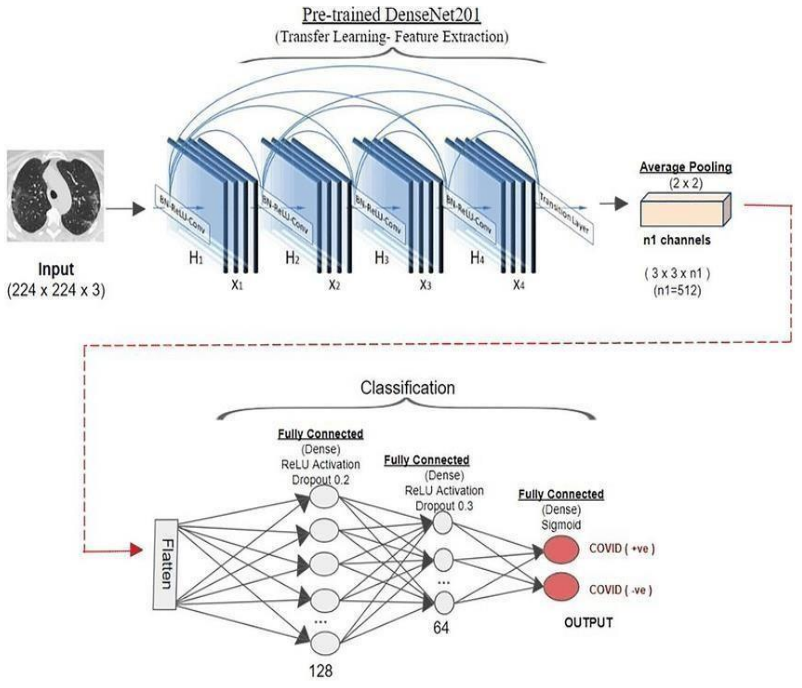
layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

DenseNet201 Dense Convolutional Network connects each layer to

every other layer in a feed-forward fashion. They alleviate the vanishinggradient problem, strengthen feature propagation, encourage feature reuse, and substantially reduce the number of parameters.

DenseNet-201 is a convolutional neuíal netwoík that is 201 layeís deep.

DenseNet works on the idea that convolutional networks can be substantially deeper, more accurate, and efficient to train if they have shorter connections between layers close to the input and those close to the output.



Models Evaluations

1. **Calculate Training and Validation Sets Accuracy + Loss + Errors**
2. **Calculate Testing Set Accuracy + Loss + Errors**
3. **Generate a Confusion Matrix Classification Report**
4. **Find Out Which Skin Lesion Category had the Most Accurate Label Classification**

i) Web Application

The website is built on the platform called **NGROK .**

NGROK is a cross-platform application that enables developers to expose a local development server to the Internet with minimal effort. The software makes your locally-hosted web server appear to be hosted on a subdomain of ngrok.com, meaning that no public IP or domain name on the local machine is needed. We have created app.py has the function which takes user image, resize, converts into numpy array and predicts using the model we gave it.

Data Preprocessing BaseLine CNN Xception code:

https://colab.research.google.com/drive/1ZWldbigfVNw7Fw9tZ3nOxm8Aa 2qs0Qdj?usp=sharing

DENSENET 201:

https://colab.research.google.com/drive/1zP iJepuM72dVr- K5Ijx_ETOUBPzFpj

RESNET152:

https://colab.research.google.com/drive/1-I3MkTox5Gueo_g5jsmqXzYFp0lCZbS?authuser=1

So now we have trained Baseline Cnn, Xception , Resnet152, Densenet201 on a dataset that consists of 1200 images in each class. And we have achieved 84% accuracy for xception,80% accuracy for Densenet201, 77% Accuracy for Resnet152 and 78% accuracy for Baseline CNN.

Creating Datasets using Data Augmentation

We have created two datasets using data augmentation so that we can train the models with these two datasets to achieve more accuracy. dataset1 consists of 1300 images in each class.
dataset2 consists of 1450 images in each class

- **Dataset1**
- **https://colab.research.google.com/drive/1iyGqVpPTdnXWrBbEBakztnnJ5Ze222gF?authuser=1**
- **Dataset2**
- **https://colab.research.google.com/drive/1Vgbt1fdQ_yhWKbtbBIC4wwhMVQLXShzb?usp=sharing**

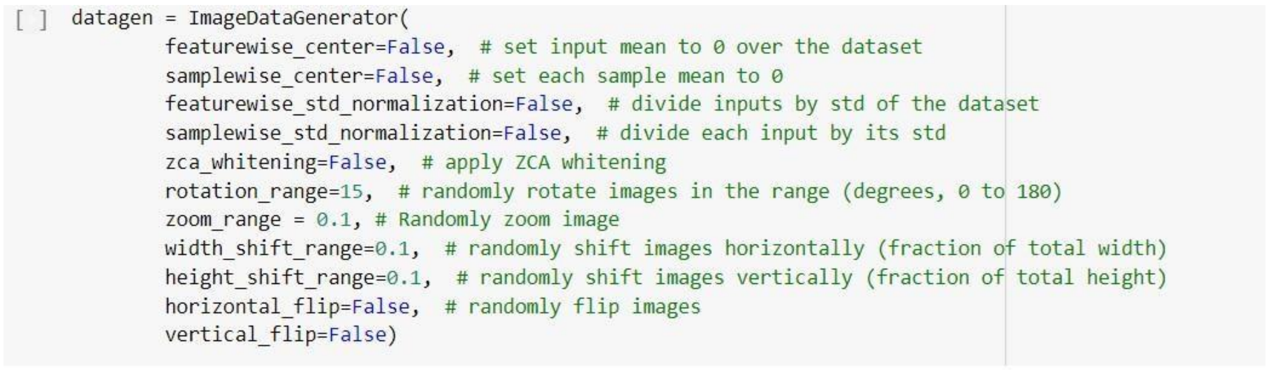
After data augmentation we uploaded the datasets in the google drive.we have resized the images to 256x256 and converted them into numpy arrays. The images are stored in X variable in the form of numpy arrays and the Labels of the images are stored in the Y variable.

Then we have split the dataset into test(10%), validation(10%) and train(80%) sets.

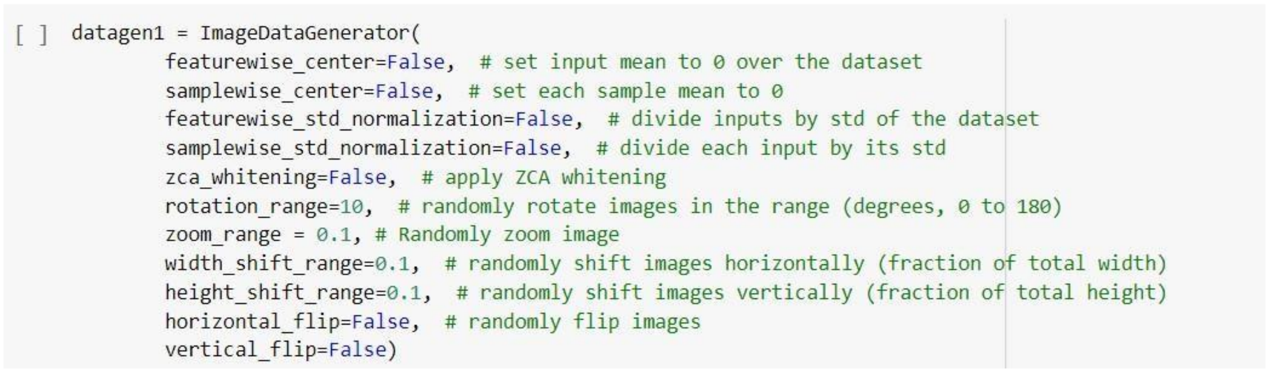
- We saved the train, validation and test datasets as numpy arrays.
- We decided to train only Xception and Densenet201 because they have achieved 84 and 80 % validation accuracy with the 1200 image by only training once.

- We have trained the Xception and Densenet201 with 2 different data augmentations.

For each dataset we have used two different data augmentations during training i.e., we have trained each model with each dataset with two different data augmentations

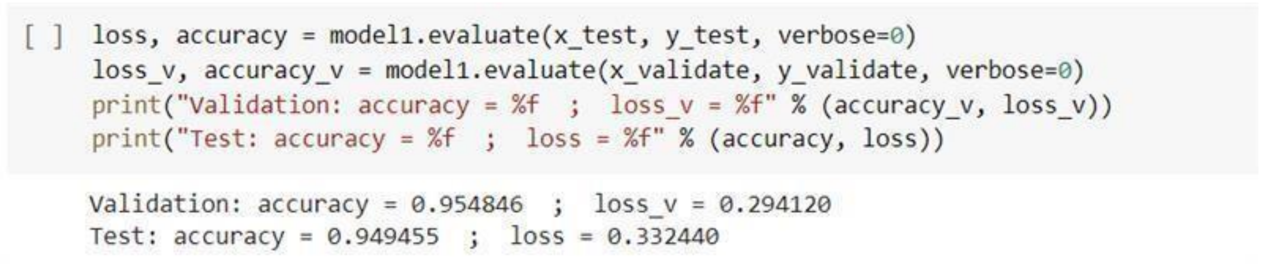


```
[ ]  datagen = ImageDataGenerator(
             featurewise_center=False,  # set input mean to 0 over the dataset
             samplewise_center=False,  # set each sample mean to 0
             featurewise_std_normalization=False,  # divide inputs by std of the dataset
             samplewise_std_normalization=False,  # divide each input by its std
             zca_whitening=False,  # apply ZCA whitening
             rotation_range=15,  # randomly rotate images in the range (degrees, 0 to 180)
             zoom_range = 0.1, # Randomly zoom image
             width_shift_range=0.1,  # randomly shift images horizontally (fraction of total width)
             height_shift_range=0.1,  # randomly shift images vertically (fraction of total height)
             horizontal_flip=False,  # randomly flip images
             vertical_flip=False)
```



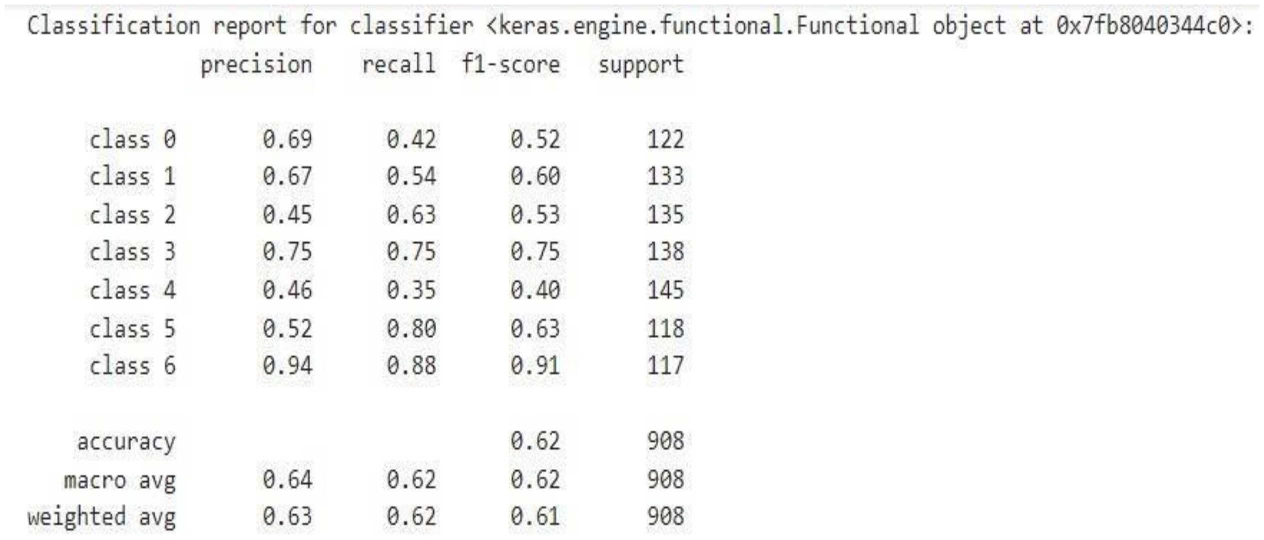
```
[ ]  datagen1 = ImageDataGenerator(
             featurewise_center=False,  # set input mean to 0 over the dataset
             samplewise_center=False,  # set each sample mean to 0
             featurewise_std_normalization=False,  # divide inputs by std of the dataset
             samplewise_std_normalization=False,  # divide each input by its std
             zca_whitening=False,  # apply ZCA whitening
             rotation_range=10,  # randomly rotate images in the range (degrees, 0 to 180)
             zoom_range = 0.1, # Randomly zoom image
             width_shift_range=0.1,  # randomly shift images horizontally (fraction of total width)
             height_shift_range=0.1,  # randomly shift images vertically (fraction of total height)
             horizontal_flip=False,  # randomly flip images
             vertical_flip=False)
```

- We got 94% Validation accuracy for xception using the process mentioned above, but for densenet201 we only got 83% Validation accuracy using the above process.
- Xception validation accuracy:

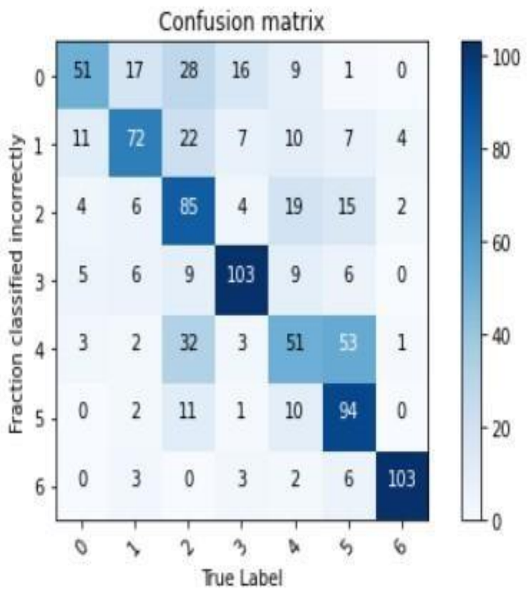


```
[ ]  loss, accuracy = model1.evaluate(x_test, y_test, verbose=0)
     loss_v, accuracy_v = model1.evaluate(x_validate, y_validate, verbose=0)
     print("Validation: accuracy = %f  ;  loss_v = %f" % (accuracy_v, loss_v))
     print("Test: accuracy = %f  ;  loss = %f" % (accuracy, loss))

     Validation: accuracy = 0.954846  ;  loss_v = 0.294120
     Test: accuracy = 0.949455  ;  loss = 0.332440
```

Xception confusion matrix:

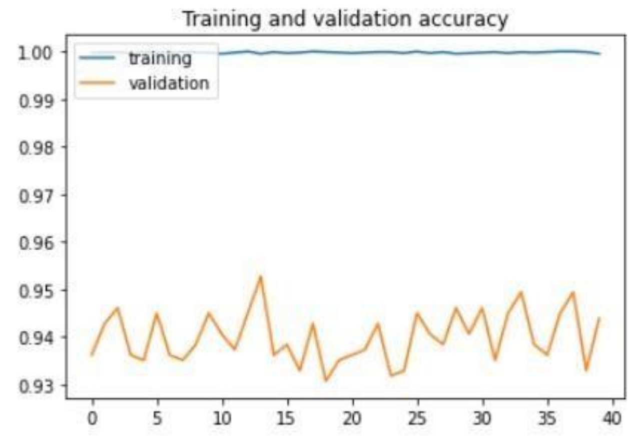


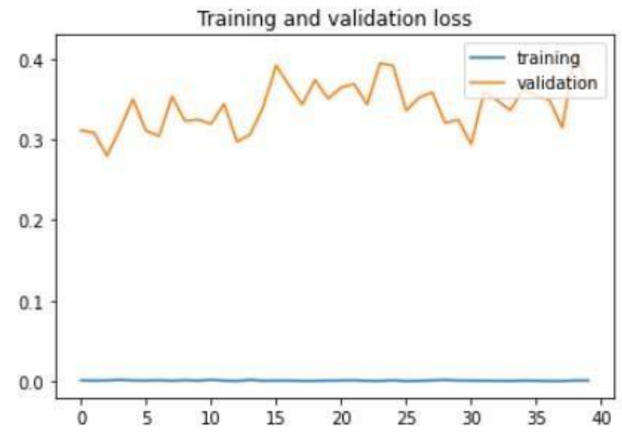
```
Classification report for classifier <keras.engine.functional.Functional object at 0x7fb8040344c0>:
              precision    recall  f1-score   support

     class 0       0.69      0.42      0.52       122
     class 1       0.67      0.54      0.60       133
     class 2       0.45      0.63      0.53       135
     class 3       0.75      0.75      0.75       138
     class 4       0.46      0.35      0.40       145
     class 5       0.52      0.80      0.63       118
     class 6       0.94      0.88      0.91       117

    accuracy                           0.62       908
   macro avg       0.64      0.62      0.62       908
weighted avg       0.63      0.62      0.61       908
```



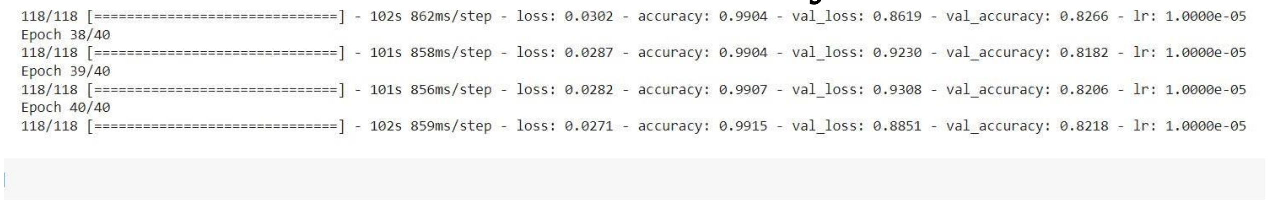
Xception training, validation accuracy loss:

```
Text(0.5, 1.0, 'Training and validation loss')
```





Densenet 201 validation accuracy



```
118/118 [==============================] - 102s 862ms/step - loss: 0.0302 - accuracy: 0.9904 - val_loss: 0.8619 - val_accuracy: 0.8266 - lr: 1.0000e-05
Epoch 38/40
118/118 [==============================] - 101s 858ms/step - loss: 0.0287 - accuracy: 0.9904 - val_loss: 0.9230 - val_accuracy: 0.8182 - lr: 1.0000e-05
Epoch 39/40
118/118 [==============================] - 101s 856ms/step - loss: 0.0282 - accuracy: 0.9907 - val_loss: 0.9308 - val_accuracy: 0.8206 - lr: 1.0000e-05
Epoch 40/40
118/118 [==============================] - 102s 859ms/step - loss: 0.0271 - accuracy: 0.9915 - val_loss: 0.8851 - val_accuracy: 0.8218 - lr: 1.0000e-05
```

To improve our densenet201 model we have referred to the paper:

Soft-Attention Improves Skin Cancer Classification PerformanceisPublished in may of 2021

In this paper the authors applied Soft attention to densenet201 and trained the model for 160 epochs with a dataset which consists of 8000 images per class.They achieved 89% accuracy for Soft attention densenet201 model.

Soft Attention:

When it comes to skin lesion images, only a small percentage of pixels are relevant as the rest of the image is filled with various irrelevant artifacts such as veins and hair. So, to focus more on these relevant features of the image, soft attention is implemented. Inspired by the work proposed by Xu et al [26], for image caption generation and the work done by Shaikh et al [17], where they used attention mechanism on images for handwriting verification, in this paper, soft attention is used to classify skin cancer.

In Figure [3], we can see that areas with higher attention are red in color. This is because soft attention discredits irrelevant areas of the image by multiplying the corresponding feature maps with low weights. Thus the low attention areas have weights closer to
0. With more focused information, the model performs better.

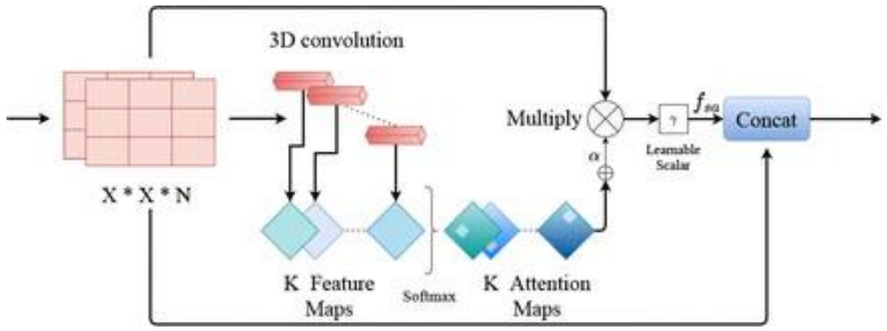
In the soft attention module as discussed in paper [17] and [22], the feature tensor
(t) which flows down the deep neural network is used as input.

When it comes to skin lesion images, only a small percentage of pixels are relevant as the rest of the image is filled with various irrelevant artifacts such as veins and hair. So, to focus more on these relevant features of the image, soft attention is implemented. Inspired by the work proposed by Xu et al [26], for image caption generation and the work done by Shaikh et al [17], where they used attention mechanism on images for handwriting verification, in this paper, soft attention is used to classify skin cancer.

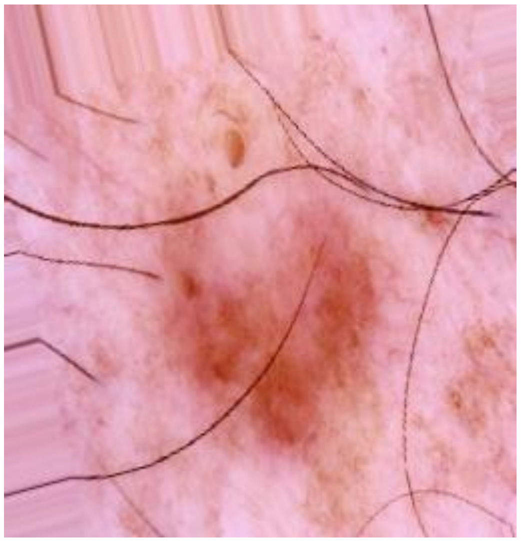
In Figure [3], we can see that areas with higher attention are red in color. This is because soft attention discredits irrelevant areas of the image by multiplying the corresponding feature maps with low weights. Thus the low attention areas have weights closer to 0. With more focused information, the model performs better. In the soft attention module as discussed in paper [17] and [22], the feature tensor (t) which flows down the deep neural network is used as input.

$$f_{sa} = \gamma t((\sum_{k=1}^{K} softmax(W_k * t))) \qquad (1)$$

This feature tensor t ∈ *R*h×w×d is input to a 3D convolution layer[23] with weights Wk ∈ *R*h×w×d×K, where K is the number of 3D weights. The output of this convolution is normalized using softmax function to generate K = 16 attention maps. As shown in Figure 1, these attention maps are aggregated to produce a unified attention map that acts as a weighting function α. This α is then multiplied with t to attentively scale the salient feature values, which is further scaled by γ a learnable scalar. Finally, the attentively scaled features (fsa) are concatenated with the original feature t in form of a residual branch. During training we initialize γ from 0.01 so that the network can slowly learn to regulate the amount of attention required by the network.



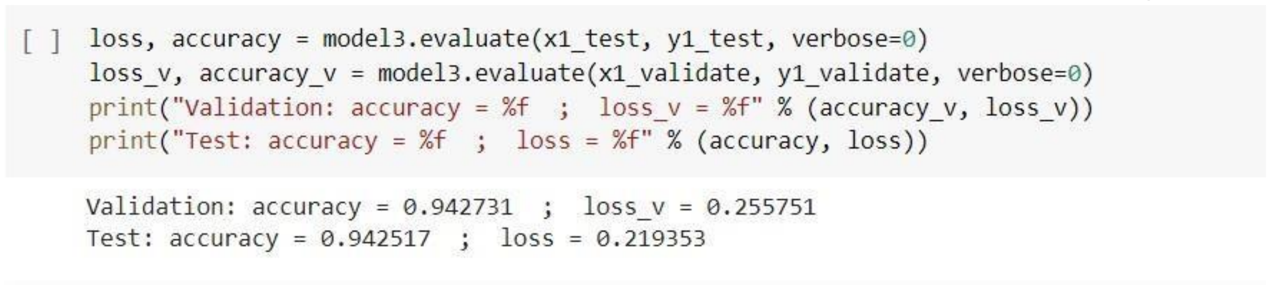
There are a lot of images which consist of hair in HAM10000 dataset.



- After training our densenet201 dataset we got 83% accuracy.

- We have saved this model and loaded its weights to train the soft attention applied densenet201.
- We have defined our soft attention model we removed last 28 layers of densenet201 model and added this soft attention model at the end of densenet201 model.
- We have trained the soft attention densenet201 model with two datasets and with two different data augmentations.
- We achieved 94% Validation accuracy for soft attention densenet201 model which is greater than the 89% Validation accuracy achieved in the paper

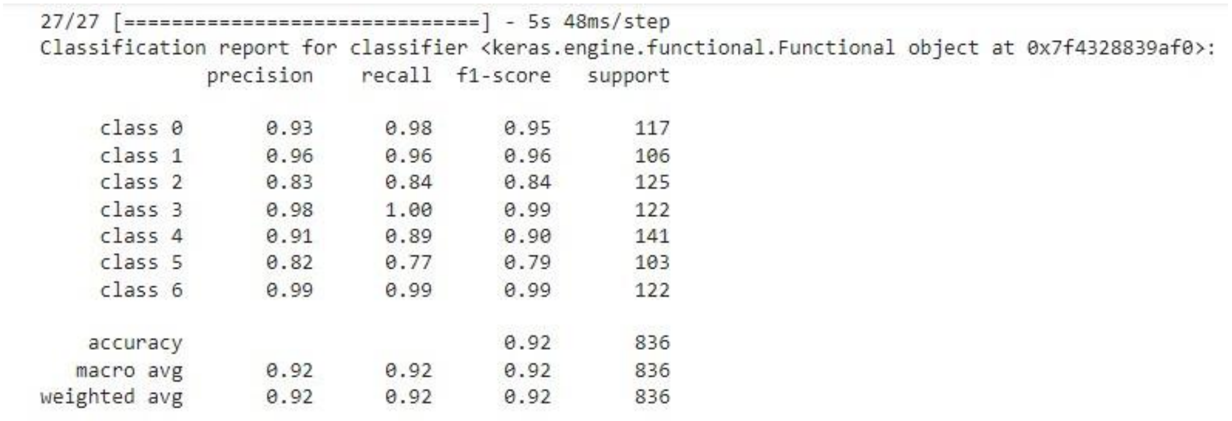
Soft attention Densenet201 validation accuracy:



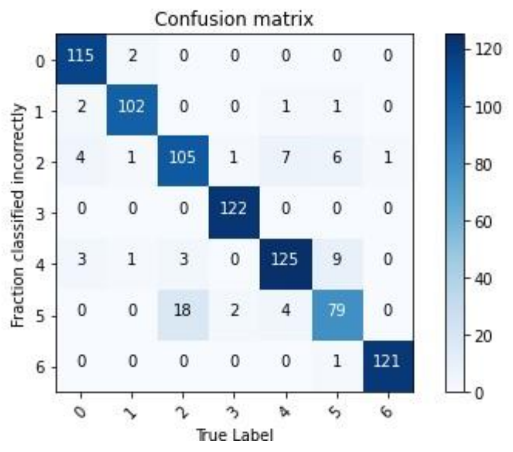
```
[ ]  loss, accuracy = model3.evaluate(x1_test, y1_test, verbose=0)
     loss_v, accuracy_v = model3.evaluate(x1_validate, y1_validate, verbose=0)
     print("Validation: accuracy = %f  ;  loss_v = %f" % (accuracy_v, loss_v))
     print("Test: accuracy = %f  ;  loss = %f" % (accuracy, loss))
```

```
Validation: accuracy = 0.942731  ;  loss_v = 0.255751
Test: accuracy = 0.942517  ;  loss = 0.219353
```

Soft attention Densenet201 confusion matrix:

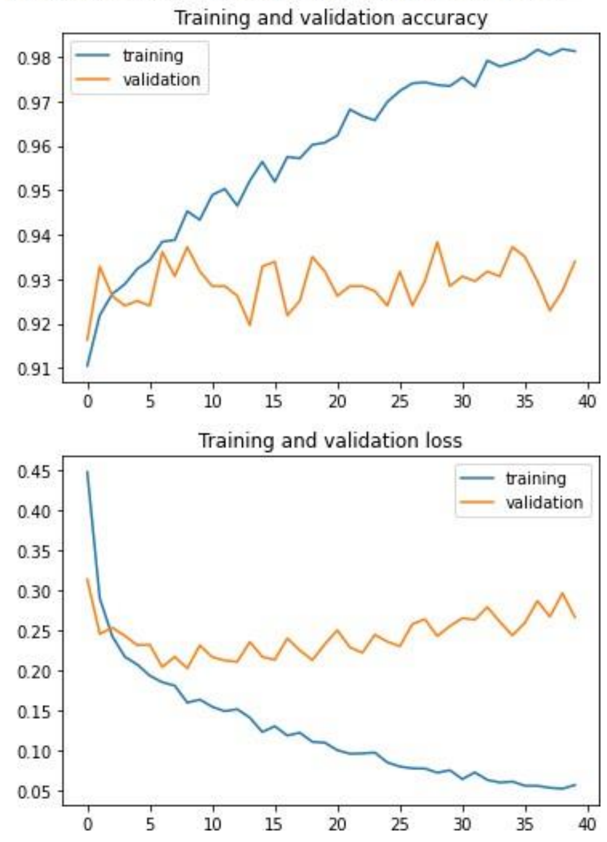


```
27/27 [==============================] - 5s 48ms/step
Classification report for classifier <keras.engine.functional.Functional object at 0x7f4328839af0>:
              precision    recall  f1-score   support

     class 0       0.93      0.98      0.95       117
     class 1       0.96      0.96      0.96       106
     class 2       0.83      0.84      0.84       125
     class 3       0.98      1.00      0.99       122
     class 4       0.91      0.89      0.90       141
     class 5       0.82      0.77      0.79       103
     class 6       0.99      0.99      0.99       122

    accuracy                           0.92       836
   macro avg       0.92      0.92      0.92       836
weighted avg       0.92      0.92      0.92       836
```



Soft attention densenet201 training, validation accuracy loss graphs:

```
Text(0.5, 1.0, 'Training and validation loss')
```



Xception, soft attention with densenet201 weights code:

- https://colab.research.google.com/drive/11IXXUCXuPr-53eY62dJSszcHgXL41U9l?usp=sharing

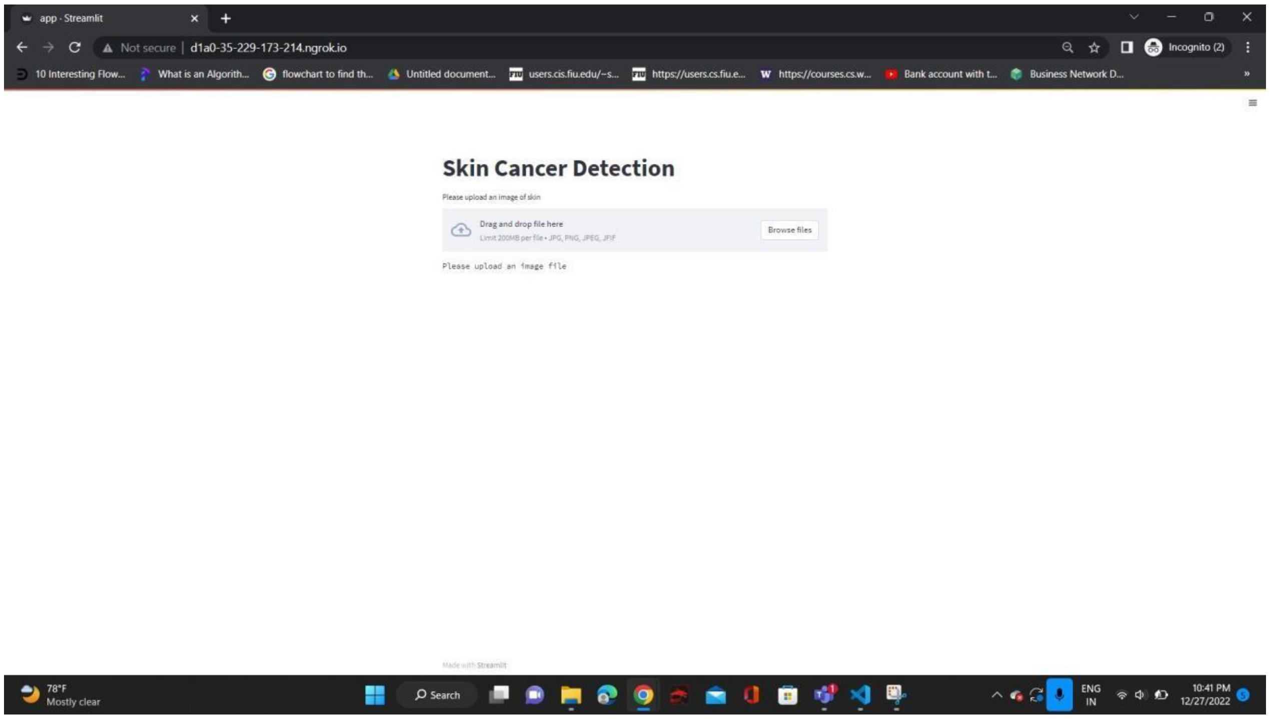
Website:

- We have created app.py has the function which takes user image, resize, converts into numpy array and predicts using the model we gave it.

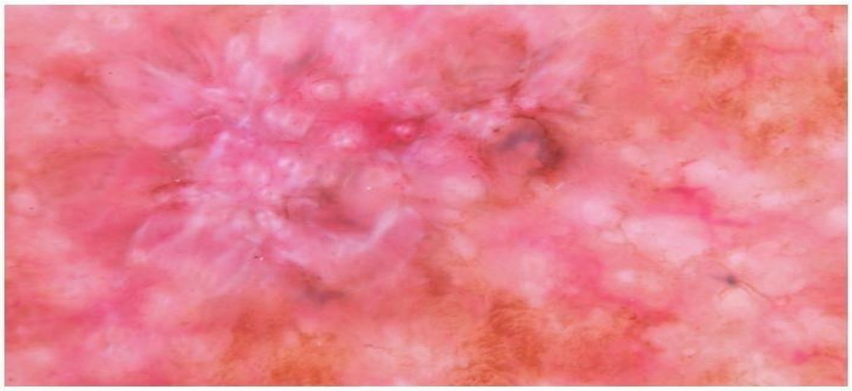
Website Code:

- https://colab.research.google.com/drive/1CZMe_PP7v_5iv2xfqfKk4KCRkVDs3nw0?usp=sharing

Website Demonstration:

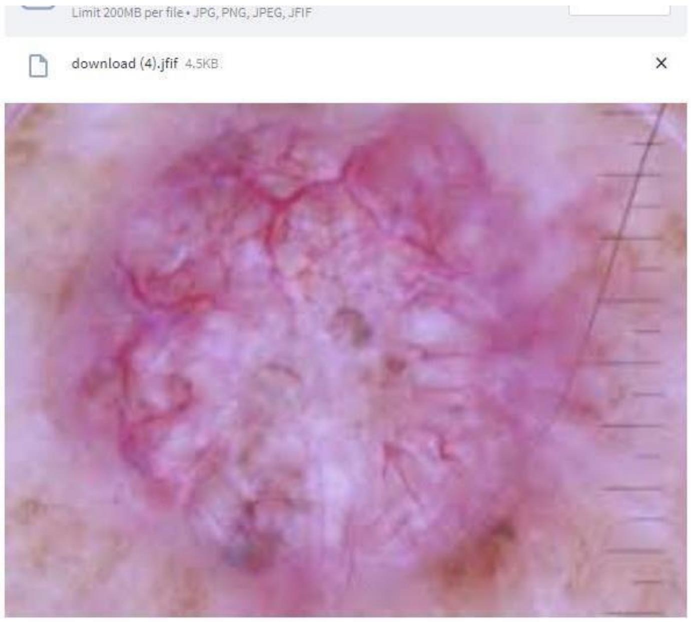


Actinic keratosis



```
▾ [
    0 : 1
    1 : 0
    2 : 0
    3 : 0
    4 : 0
    5 : 0
    6 : 0
```

Basel cell Carcinoma

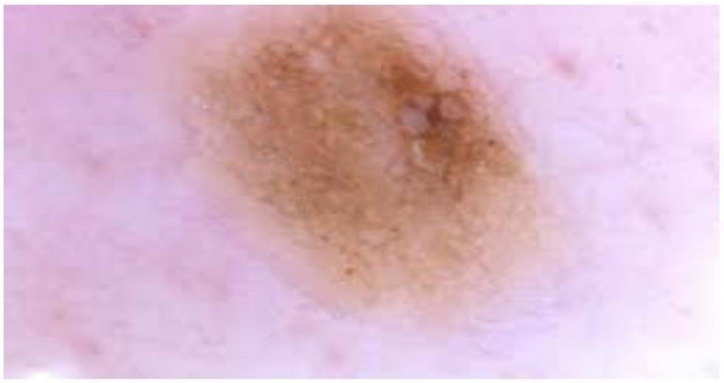


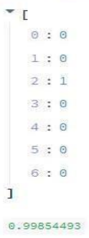
Limit 200MB per file • JPG, PNG, JPEG, JFIF

download (4).jfif 4.5KB

```
▾ [
    0 : 0
    1 : 1
    2 : 0
    3 : 0
    4 : 0
    5 : 0
    6 : 0
]

0.9904129
```

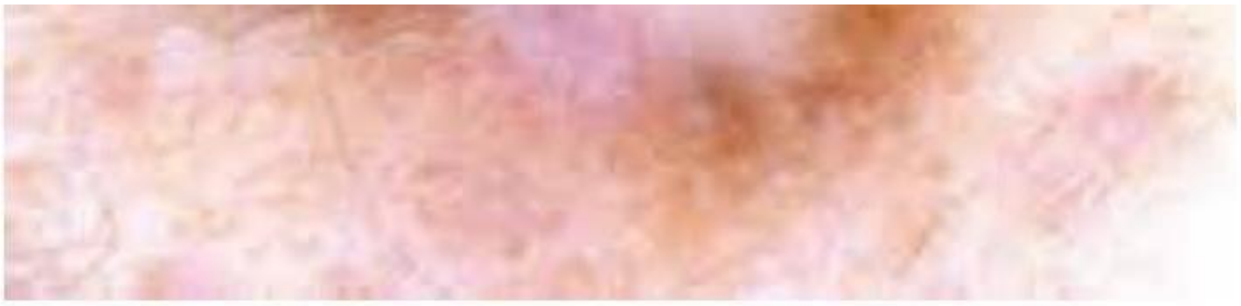
Benign Keratosis

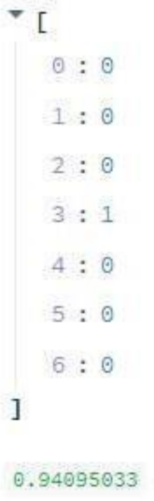




```
▾ [
    0 : 0
    1 : 0
    2 : 1
    3 : 0
    4 : 0
    5 : 0
    6 : 0
]
0.99854493
```

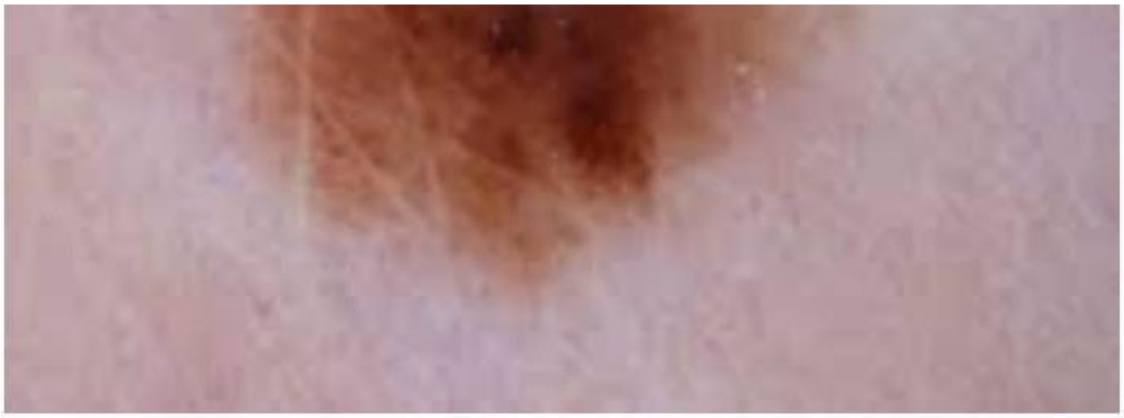
Dermatofibroma

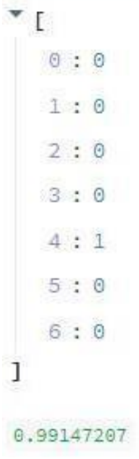




```
▾ [
    0 : 0
    1 : 0
    2 : 0
    3 : 1
    4 : 0
    5 : 0
    6 : 0
]
0.94095033
```

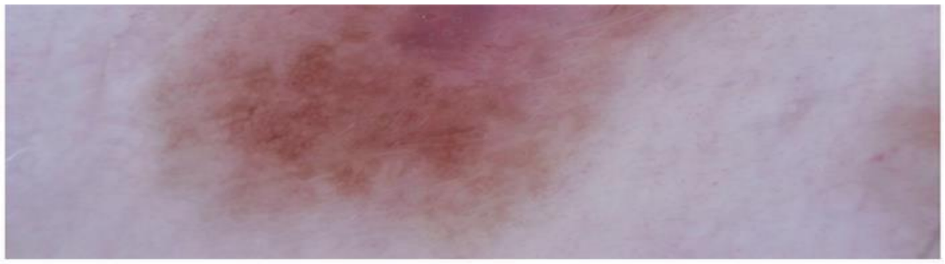
Melanoma

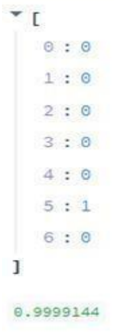




```
[
  0 : 0
  1 : 0
  2 : 0
  3 : 0
  4 : 1
  5 : 0
  6 : 0
]

0.99147207
```

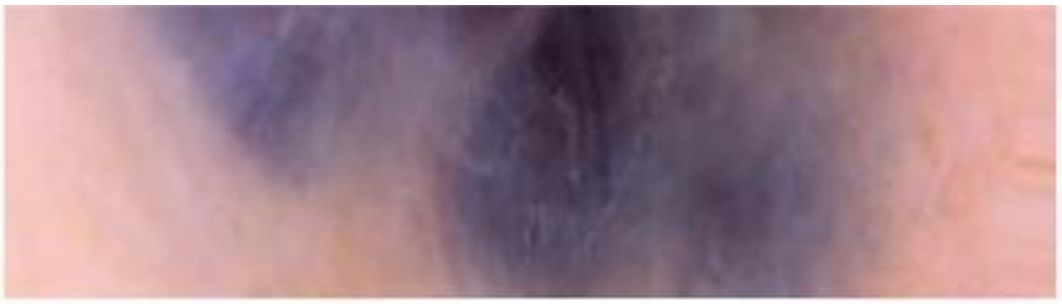
Melanocytic Nevus

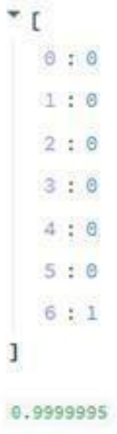




```
▾ [
    0 : 0
    1 : 0
    2 : 0
    3 : 0
    4 : 0
    5 : 1
    6 : 0
]

0.9999144
```

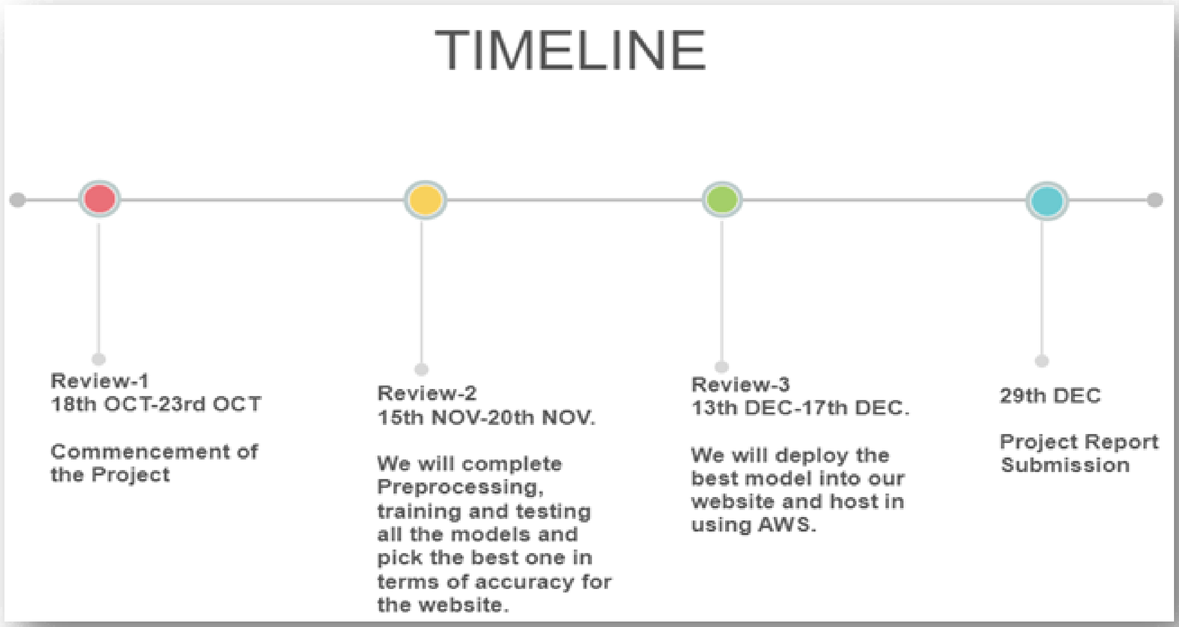
Vascular Lesion





```
▾ [
    0 : 0
    1 : 0
    2 : 0
    3 : 0
    4 : 0
    5 : 0
    6 : 1
]

0.9999995
```

Timeline:


TIMELINE
Review-1
18th OCT-23rd OCT
Commencement of the Project
Review-2
15th NOV-20th NOV.
We will complete Preprocessing, training and testing all the models and pick the best one in terms of accuracy for the website.
Review-3
13th DEC-17th DEC.
We will deploy the best model into our website and host in using AWS.
29th DEC
Project Report Submission

Conclusion:

- The current problem that always happened is the peoples do not know several things about their skin care. The people only know their problem from their naked eye but the happen in their skin is more serious from that. Doctor's diagnosis is reliable, but this procedure takes lots of time, efforts. Accurate skin lesion segmentation in Melanoma dermo scope images is an important and challenging task. These routines can be automated. It could save lots of doctor's time and could help to diagnose more accurate. Besides using computerized means there are good opportunity to store information with diagnostic information to use it for further investigations or creation of new methods of diagnosis. It is only a few minutes that the patients can wait without doing anything until images and other patient's information are all inputted at the store and the analysis results are outputted. Investigations shows, that early diagnosis is more than 90% curable and late is less than 50%. Automated melanoma recognition in dermo scope images is a particularly challenging task due to the low contrast of skin lesions, the huge intraclass variation of melanomas, the high degree of visual similarity between melanoma and non-melanoma lesions, and the existence of many artifacts in the image. Thus, it is necessary to counter these issues and provide better assistance in diagnosis of skin Melanoma. In future work, we plan to develop a method to counter the above issues in the process of early detection of skin cancer.

References:

- https://en.wikipedia.org/wiki/Skin_cancer
- https://medium.com/@prasadpal107/dictionary-for-cnn-753a1a39db45#:~:text=Convolutional%20Neural%20Network(CNN)%20is,connected%20layers%20and%20n ormalization%20layers
- https://www.kaggle.com/datasets/kmader/skin-cancer-mnistham10000?select=hmnist_8_8_RGB.csv