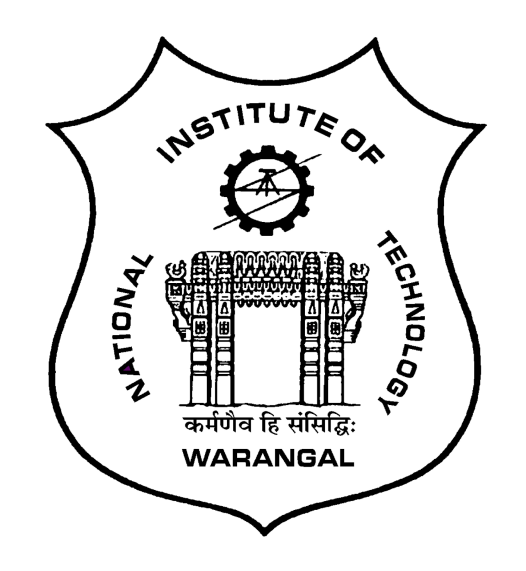
**NATIONAL INSTITUTE OF TECHNOLOGY, WARANGAL**

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING



**INTRODUCTION TO ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING (EC204)**

**PROJECT REPORT**

**Skin Cancer Classification Using Image Processing and Deep Learning**

**TEAM MEMBERS:**

* Aditya Mishra (21ECB0B03)
* Taras Rajan Kerketta (21ECB0B61)

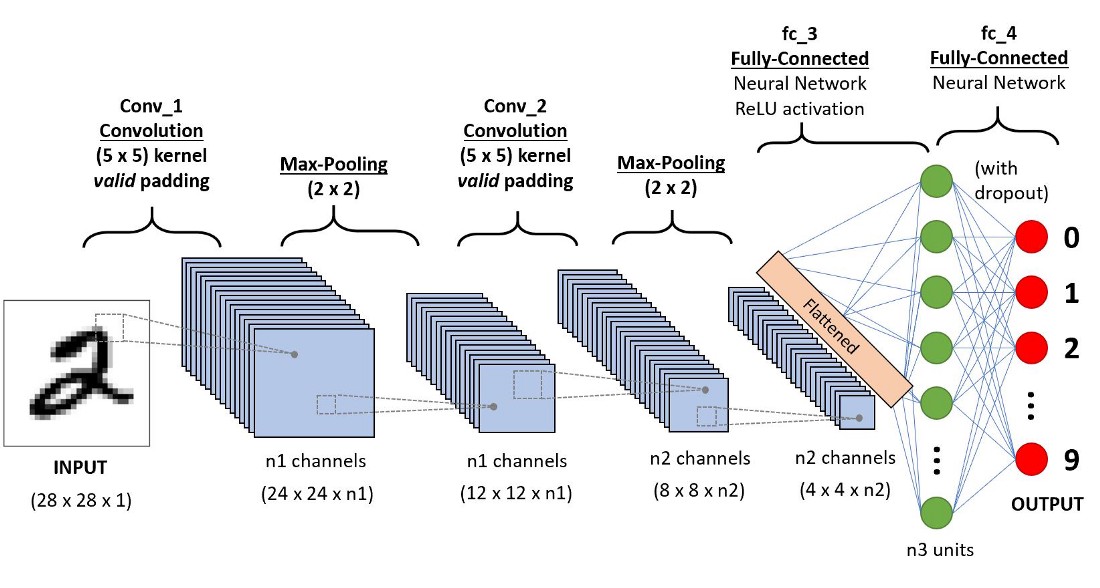
**SUBMITTED TO:**

**Professor J. Ravi Kumar**

**INTRODUCTION:**

Convolutional Neural Networks (CNN) is a type of Deep Learning architecture inspired by the visual cortex of a mammal. They possess a huge potential and have found their applications in the field of image classification, natural language processing, image caption generator and many more. In the recent past, due to the lack of computation power CNNs failed to solve complex problems. But with the advent of Graphical Processing Units(GPU) and their uses in machine learning, CNNs have re-emerged and outperformed other architectures for computer vision tasks. CNN is attracting interest in variety of domains and medical diagnosis is no exception. The use of CNNs in the field of medical research have already been demonstrated in detecting diabetic retinopathy, detection of lymph node in women with breast cancer and many more. Image Classification plays a key role in computer vision. It includes preprocessing the image data, image segmentation, extracting key features and classifying the image to its corresponding class. With the help of CNNs images are classified efficiently and accurately. This technique could be applied in medical diagnosis, facial recognition, security and many different fields.

**CONVOLUTION NEURAL NETWORK:**



A **Convolution Neural Network (ConvNet/CNN)** is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlaps to cover the entire visual area.

1. **Input Layer:** This layer holds the raw input of the image with width 64, height 64 and depth 3.
2. **Convolution Layer:** This layer computes the output volume by computing the dot product between all filters and image patches. If we use total of 12 filters for this layer, we’ll get output volume of dimension 32 x 32 x 12.
3. **Activation Function Layer:** This layer will apply an element-wise activation function to the output of the convolution layer. Some common activation functions are -

RELU: max (0, x), SIGMOID: 1/(1+e^-x) etc.

4. **Pool Layer:** This layer is inserted in the convnets and its main function is to reduce the size of volume which makes the computation fast reduces memory and also prevents overfitting. Two common types of pooling layers are **max pooling** and **average pooling**. If we use a max pool with 2 x 2 filters and stride 2, the resultant volume will be of dimension 30,30,3.

**TENSORFLOW:**

* TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks.
* TensorFlow was developed by the Google Brain team for internal Google use in research and production. The initial version was released under the Apache License 2.0 in 2015. Google released the updated version of TensorFlow, named TensorFlow 2.0, in September 2019.

**SKIN CANCER:**

Skin cancers are cancers that arise from the skin. They are due to the development of abnormal cells that have the ability to 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 a number of less common skin cancers, are known as no melanoma skin cancer (NMSC. Basal-cell cancer grows slowly and can damage the tissue around it but is unlikely to spread to distant areas or result in death. It often appears as a painless raised area of skin that may be shiny with small blood vessels running over it or may present as a raised area with an ulcer.Squamous-cell skin cancer is more likely to spread. It usually presents as a hard lump with a scaly top but may also form an ulcer.Melanomas are the most aggressive. Signs include a mole that has changed in size, shape, color, has irregular edges, has more than one color, is itchy or bleeds.

**METHODOLOGY (Algorithm) ML:**

**Import Library:**

import numpy as np

import pandas as pd

import os

from glob import glob

import keras

from keras.utils.np\_utils import to\_categorical

from keras.models import Sequential

from keras.layers import Dense, Dropout, Flatten, Conv2D, MaxPooling2D, BatchNormalization, Activation

from PIL import Image

np.random.seed(123)

from keras.preprocessing.image import ImageDataGenerator

from sklearn.model\_selection import train\_test\_split

* **Glob**(short for global) is used to return all file paths that match a specific pattern.
* **Sequential API**is used to create models’ layer-by-layer. Functional API is an alternative approach of creating more complex models. Functional model, you can define multiple input or output that share layers.
* **Python Imaging Library**is a free and open-source additional library for the Python programming language that adds support for opening, manipulating, and saving many different image file formats.
* The**seed ()**method is used to initialize the random number generator. The random number generator needs a number to start with (a seed value), to be able to generate a random number.
* The**random\_state** is an integer value which implies the selection of a random combination of train and test. When you set the test\_size as 1/4 the there is a set generated of permutation and combination of train and test and each combination has one state.

2)3"#<! #

# 2&3

0       # $    !    4

#!

%#

$$#!

+-.////!,! ,  !, !

A ,#,#,#!,#

2)3"#<! #

# 2&3

0       # $    !    4

#!

%#

$$#!

+-.////!,! ,  !, !

A ,#,#,#!,#

**Methods**

This document presents the CNN classification for targeted disease. The disease is 7 types of skin lesions that was existed in HAM10000 dataset,

'nv': 'Melanocytic nevi',

'mel': 'Melanoma',

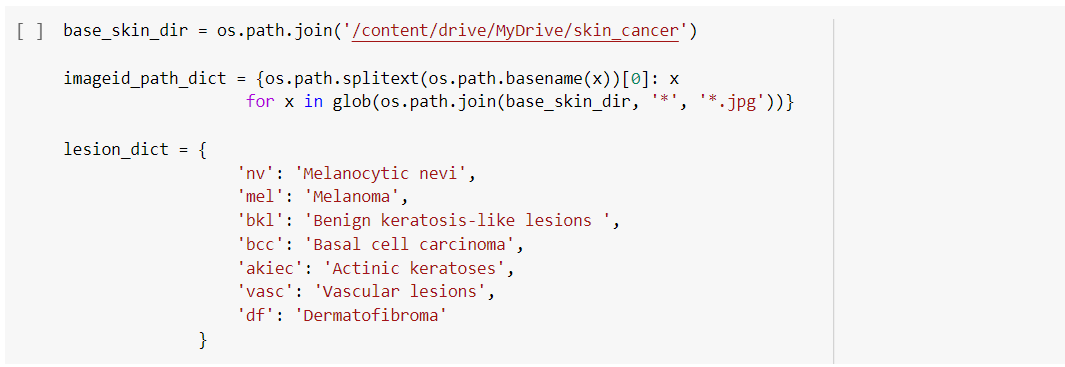
'bkl': 'Benign keratosis-like lesions ',

'bcc': 'Basal cell carcinoma',

'akiec': 'Actinic keratoses',

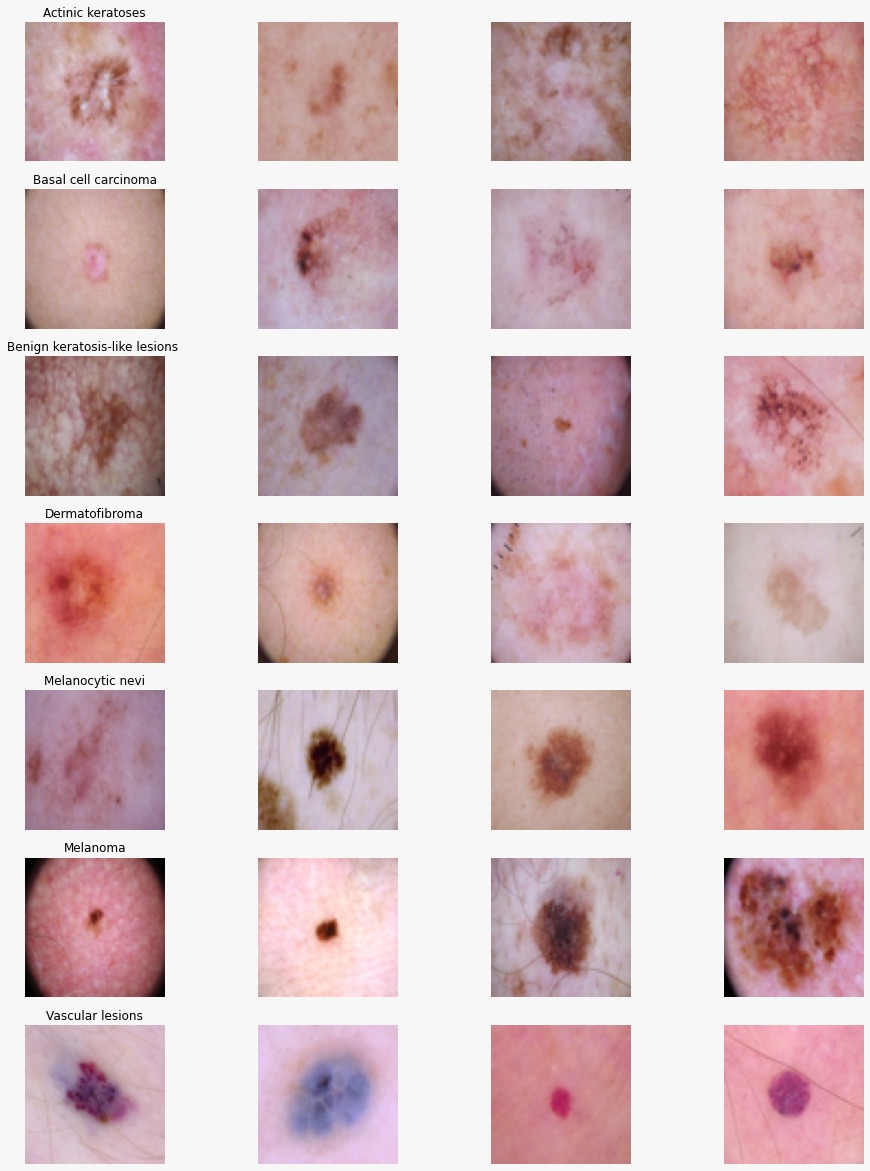
'vasc': 'Vascular lesions',

'df': 'Dermatofibroma'

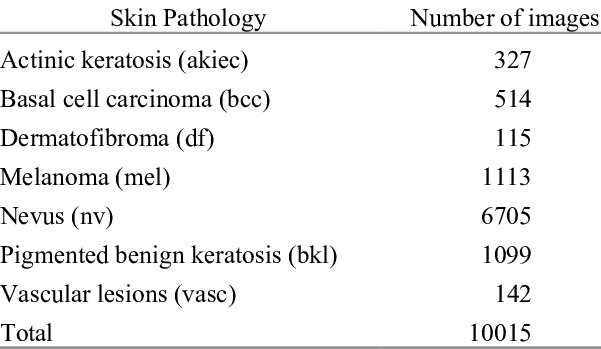


**Data collection**

* The dataset that was available on Kaggle.

****

HAM10000 dataset - MNIST-HAM10000 is the dataset used for training the model. The dataset consists of around 10015 labelled images for 7 different types of skin lesions. The 7 different types of skin lesions and their distribution is shown. This table shows sample dermoscopy pictured used in this system.

****

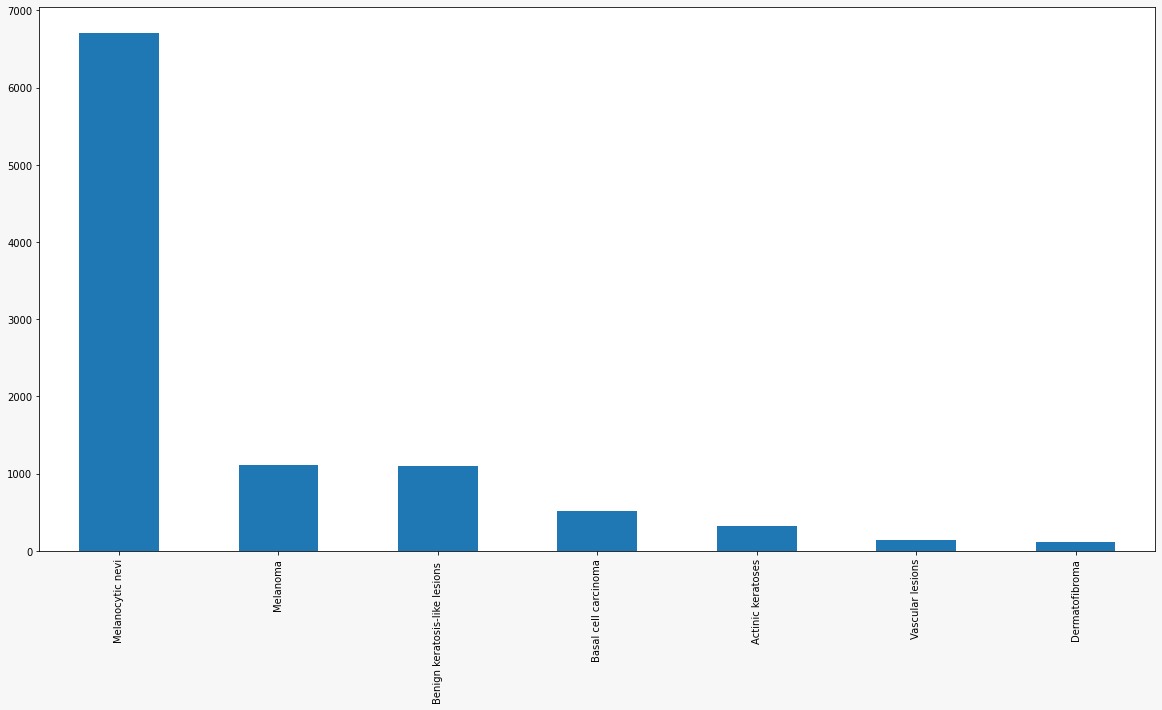
**Data Pre-processing:**

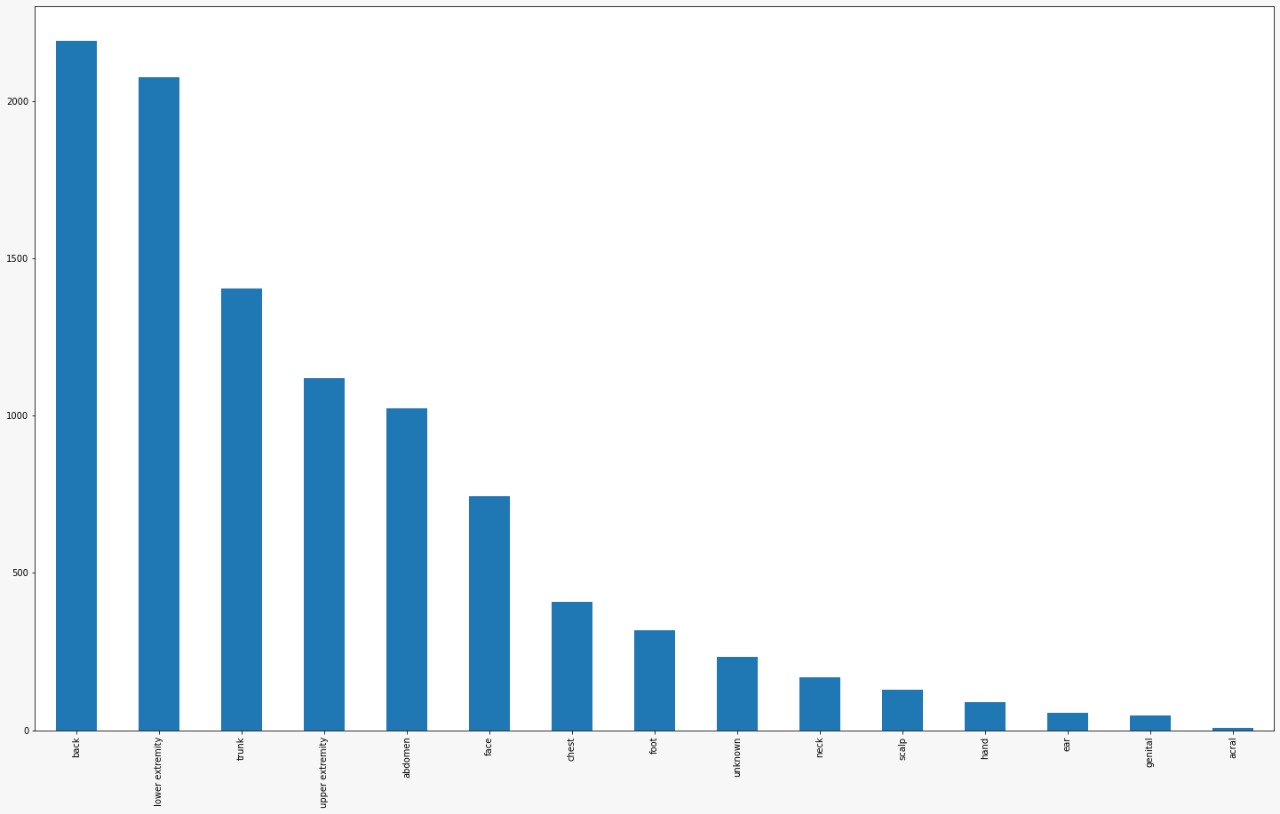
This stage sows the data preparation, this is needed to make the data ready for use as an input. First, the image size is equalised into 64 by 64 pixels.Then normalization is applied, this is done to modify values so that can measure them on the global scale.

The data must be divided into 3 parts, namely training data, validation data and test data. This is needed to build and test the system. Sequentially each part of the data amounts to 8012, 2003 and 789 then data augmentation applied to the training data, this is used to prevent overfitting and to get the better accuracy.

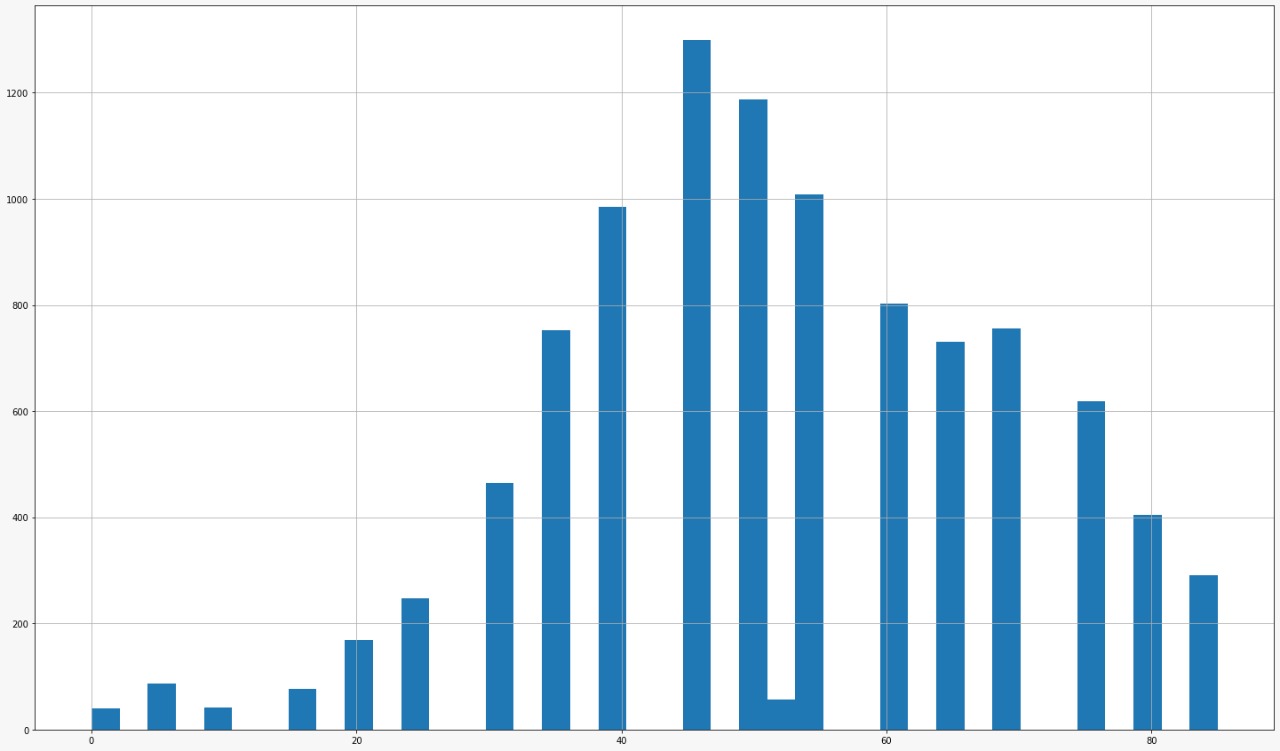
**Data Analyzation**

No. of images for each type of skin cancer

****

Localization of cancer

Images of each age group



**Algorithm**

**Building CNN Model**

**Convolution Layer:**

Convolution layer performs a convolution operation on the output of other function repeatedly. This layer applies the output function as a feature map from the image. Convolution is used to produce the extracted feature from the inputted image. The layer calculates the output of the neurons that connected to the input to local area in the input, the calculation used is dot product between their weights and the small area connected to the input. Kernel, stride and padding is used as the hyperparameter in this layer. The size of kernel and stride that used for each convolution layer are 3 by 3and one. Further, the type of padding used are same which means the output dimension is same as the input dimension.

**Activation function**

This, that used is Rectified Linear unit ReLU. The ReLU will eliminate vanishing gradient by applying the activation function element as f(x)=max(0, x) which means element activation will be done when at the threshold of 0.

**Softmax**

The softmax function is used as the activation function in the output layer of neural network models that predict a multinomial probability distribution. That is, softmax is used as the activation function for multi-class classification problems where class membership is required on more than two class labels.

**Pooling**

This layer is inserted in the convnets and its main function is to reduce the size of volume which makes the computation fast reduces memory and also prevents overfitting. Two common types of pooling layers are max pooling and average pooling. If we use a max pool with 2 x 2 filters and stride 2, the resultant volume will be of dimension 30,30,3.

**Flattening**

The features map generated from the previous stage is still in the multidimensional form array, so we must to flattening or reshaping the feature map. It is needed to convert feature map into a vector so it can be used as input from the fully connected layer.

**Dense layer**

Dense layer is the layer that contains all the neurons that are deeply connected within them. This means that every neuron in the dense layer takes the input from all the other neurons of the previous layer. We can add as manydense layers as required. It is one of the most commonly used layers.

**CNN Architecture:**

* **Model: "sequential"\_ Layer (type) Output Shape Param # =================================================================**
* **conv2d (Conv2D) (None, 62, 62, 32) 896**
* **conv2d\_1 (Conv2D) (None, 60, 60, 32) 9248 max\_pooling2d (MaxPooling2D )(None, 30, 30, 32) 0 dropout (Dropout) (None, 30, 30, 32) 0**
* **conv2d\_2 (Conv2D) (None, 28, 28, 64) 18496**
* **conv2d\_3 (Conv2D) (None, 26, 26, 64) 36928 max\_pooling2d\_1 (MaxPooling2D)(None, 13, 13, 64) 0 dropout\_1 (Dropout) (None, 13, 13, 64) 0**
* **flatten (Flatten) (None, 10816) 0**
* **dense (Dense) (None, 128) 1384576**
* **dropout\_2 (Dropout) (None, 128) 0**
* **dense\_1 (Dense) (None, 64) 8256**
* **dense\_2 (Dense) (None, 32) 2080**
* **dropout\_3 (Dropout) (None, 32) 0**

**dense\_3 (Dense) (None, 7) 231 =================================================================**

**Total params: 1,460,711**

**Trainable params: 1,460,711**

**Non-trainable params: 0**

**CNN Sequence**

* **shape = (64, 64, 3)classifier = Sequential()**
* **classifier.add(Conv2D(32, (3, 3), activation='relu',input\_shape=shape))**
* **classifier.add(Conv2D(32, (3, 3), activation='relu'))**
* **classifier.add(MaxPooling2D(pool\_size = (2, 2)))**
* **classifier.add(Dropout(0.05))**
* **classifier.add(Conv2D(64, (3, 3), activation='relu'))**
* **classifier.add(Conv2D(64, (3, 3), activation='relu'))**
* **classifier.add(MaxPooling2D(pool\_size=(2, 2)))**
* **classifier.add(Dropout(0.10))**
* **classifier.add(Flatten())**
* **classifier.add(Dense(128, activation='relu'))**
* **classifier.add(Dropout(0.15))**
* **classifier.add(Dense(64, activation='relu'))**
* **classifier.add(Dense(32, activation='relu'))**
* **classiier.add(Dropout(0.20))**
* **classifier.add(Dense(7, activation='softmax'))**
* **classifier.summary()**

**Compilation:**

**The code is now compiled**

1. **Optimiser: Adam**

Optimizers are algorithms or methods used to change the attributes of your neural network such as weights and learning rate in order to reduce the losses. Optimizers help to get results faster.

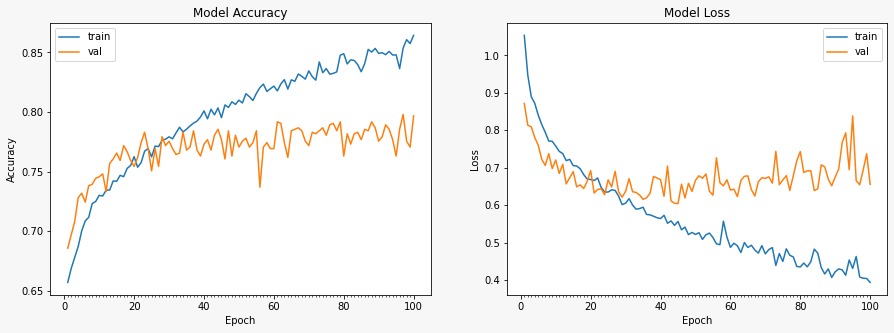
**Adam** is an alternative optimization algorithm that provides more efficient neural network weights by running repeated cycles of “adaptive moment estimation.” Adam extends on stochastic gradient descent to solve non-convex problems faster while using fewer resources than many other optimization programs.

1. **Loss: categorical\_crossentropy –** Loss: A scalar value that we attempt to minimize during our training of the model. The lower the loss, the closer our predictions are to the true labels.
2. **Matrices accuracy**:Used as a loss function for multi-class classification model where there are two or more output labels. The output label is assigned one-hot category encoding value in form of 0s and 1. The output label, if present in integer form, is converted into categorical encoding using keras.

**Outputs**

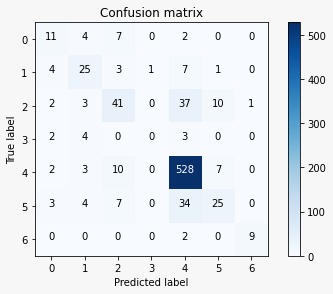
**Model Accuracy and Model loss graph**

On training the model for 100 epochs it was observed that a test accuracy of 85 percent was observed. It was observed that the accuracy for both training and test data kept increasing up to 30 epochs and gradually started saturating for the later epochs. Similarly, the training and test loss decreased exponentially for the first 30 epochs and remained unchanged for the next epochs. The plot for the accuracy and loss obtained during the training and testing process is shown:



**Confusion Matrix**

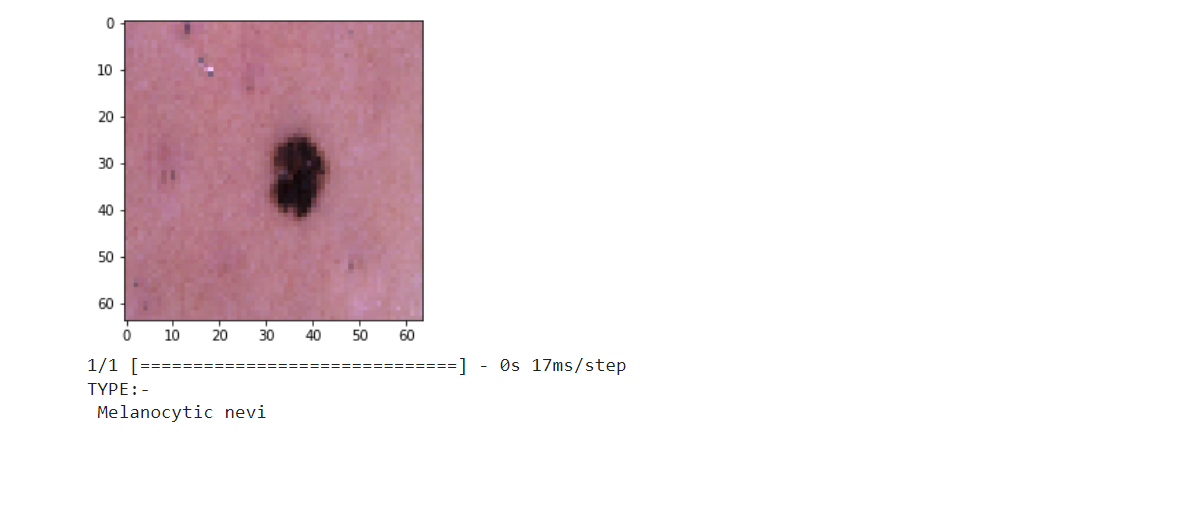
A confusion matrix is a table that is used to define the performance of a classification algorithm. A confusion matrix visualizes and summarize the performance of classification algorithm.



**SIMULATION AND RESULT:**

**Software Used: Google Colab**





**CONCLUSION:**

The result of CNN model that use input shape with 64x64 pixel. Adam optimizer, learning rate 0.001 and number of epochs 100 show that the level of training accuracy is 84% for testing accuracywith the HAM10000 dataset. Bcc disease is the most difficult to identified by the system and melanoma is the most likely identify by the system.CNNs performbetter than human vision and predicting the highest of probabilitiescan reduce the efforts of a dermatologist or a specialistto a considerable extent.We also tested the same with randomly generated augmented images and achievednearly the same accuracy and precision. The performance indexes here are Accuracy, Precision, Recall. Final results showed that using the Standard CNN method gives the best achievement for the Skin cancer diagnosis.

$<&/+6/+<///

 %/#)/\*()\*0

 !! 

$<&/+6/+<///

 %/#)/\*()\*0

 !! 

**REFERENCE:**

* <https://ieeexplore.ieee.org/abstract/document/9294814>
* <https://aip.scitation.org/doi/epdf/10.1063/1.5141652>
* <https://ieeexplore.ieee.org/abstract/document/9377155>
* https://www.kaggle.com/datasets/kmader/skin-cancer-mnist-ham10000