

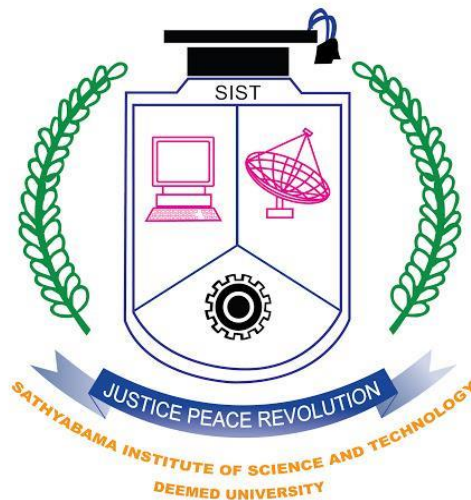
SKIN CANCER IMAGE PROCESSING

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in Computer Science and engineering

by

A. VAMSI ANIRUDH (37110045)

T. AGASSI DAN (37110021)



**DEPARTMENT OF COMPUTER SCIENCE AND
ENGINEERING SCHOOL OF COMPUTING**

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,

CHENNAI - 600119

APRIL-2020

SATHYABAMA



INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with “A” grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai –
600119



www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **A. VAMSI ANIRUDH (3611262)** who carried out the project entitled “**SKIN CANCER IMAGE PROCESSING**” under my supervision from November 2019 to April 2020.

Internal Guide

Dr. P. SARDAR MARAN, M.E., Ph.D.,

Head of the Department

Submitted for Viva voce Examination held on_____

Internal Examiner

External Examiner

DECLARATION

I **A. VAMSI ANIRUDH (37110045)** hereby declare that the Project Report entitled “**SKIN CANCER IMAGE PROCESSING**” done by me under the guidance of Assistant Prof. **Dr. P. SARDAR MARAN, M.E., Ph.D.**, is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY**.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

The satisfaction and intense happiness on the successful completion of any task would be incomplete without mentioning the people who made it possible whose constant guidance and encouragement made my effort with success.

I am very grateful to **SATHYABAMA** management for encouraging the students to do the projects after the completion of second year itself.

I convey my thanks to **Dr. T. SASIKALA, M.E., Ph.D.**, Dean, School of Computing, **Dr. S. VIGNESHWARI, M.E., Ph.D.**, and **Dr. L. LAKSHMANAN, M.E., Ph.D.**, Heads of the Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. P. SARDAR MARAN, M.E., Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved the way for the successful completion of my project work.

This acknowledgement would not be complete without expressing my sincere gratitude to my parents for their love, patience, encouragement and understanding which are the source of my motivation and inspiration throughout my work.

ABSTRACT

Dermatological diseases are determined to result in a severe effect on the fitness of thousands and thousands of human beings as everyone is suffering from nearly all sorts of skin issues every year. For the reason that human analysis of such illnesses takes some time and effort, and present day methods are simplest used to analyze singular styles of pores and skin illnesses, there is a need for a more excessive-level pc-aided know-how within the analysis and diagnosis of multi-type skin illnesses. Our proposed approach is simple, fast and does not require expensive equipment other than a camera and a computer. The approach works on the inputs of a color image. This paper proposes a technique to apply PC-aided strategies in deep studying neural networks together with Convolutional neural networks (CNN).

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ACKNOWLEDGEMENT	iv
	ABSTRACT	v
	TABLE OF CONTENTS	vi
	LIST OF FIGURES	viii
	LIST OF ABBREVIATIONS	ix
1.	INTRODUCTION	1
	1.1 CNN KEY FEATURES	2
	1.2 CNN ADVANTAGES	3
	1.3 INTRODUCTION	4
2.	AIM AND SCOPE	6
	2.1 AIM OF THE PROJECT	6
	2.2 REAL TIME APPLICATIONS	6
	2.3 LIMITATIONS	6
	2.4 ADVANTAGES	7
	2.5 SCOPE	7
3.	DESIGN METHODOLOGY	8
	3.1 SKIN DETECTION	8
	3.2 MACHINE LEARNING	9

	3.3 DEEP LEARNING	10
	3.4 CONVOLUTIONAL NEURAL NETWORKS	10
	3.5 DATASET	12
	3.6 LITERATURE SURVEY	13
4.	METHODOLOGY	13
	4.1 PROPOSED SYSTEM	13
	4.2 PROPOSED SOLUTION	14
	4.3 SYSTEM FUNCTIONAL SPECIFICATION	15
	4.4 SYSTEM ARCHITECTURE	18
	4.5 MODULES	19
	4.6 LIBRARIES	19
5.	RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS.	21
	5.1 ANALYSIS	21
	5.2 FUTURE DEVELOPMENT	23
	5.3 CONCLUSION	23
	REFERENCES	24
	APPENDIX	25
	A. SCREEN SHOTS	25
	B. SOURCE CODE	28
	C. PUBLICATION REPORT	34

LIST OF FIGURES

FIGURE No.	FIGURE NAME	PAGE No.
3.5.1	EXAMPLE IMAGE OF BENIGN	12
3.5.2	EXAMPLE IMAGE OF MALIGNANT	12
4.4	FLOW DIAGRAM OF THE PROJECT	19
5.1.1	GRAPHICAL REPRESENTATION OF LOSS	21
5.1.2	GRAPHICAL REPRESENTATION OF ACCURACY	21
5.1.3	ANALYSIS OUTPUT	22
A.1	TEST DATA ACCURACY VALUES	25
A.2	TEST DATA ACCURACY VALUES	26
A.3	TEST DATA PREDICTION	26
A.4	CLASSIFICATION	27
A.5	IMAGE DATABASE	27

LIST OF ABBREVIATIONS

ABBREVIATION	EXPANSION
CNN	CONVOLUTIONAL NEURAL NETWORKS
SIFT	SCALE-INVARIANT FEATURE TRANSFORMATION
HOG	HISTOGRAM OF ORIENTED GRADIENTS
NN	NEURAL NETWORK
VGG	VISUAL GEOMETRY GROUP
ML	MACHINE LEARNING
DRN	DILATED RESIDUAL NETWORK
FCRN	FULLY CONNECTED RESIDUAL NETWORK
GPU	GRAPHICAL PROCESSING UNIT

CHAPTER 1

INTRODUCTION

To pick out a skin disease, a ramification of visual clues can be used which includes the character lesion morphology, the body size distribution, shade, scaling and association of lesions. by way of analyzing the man or woman components separately, the complexity of the recognition method is pretty improved and the human-engineered characteristic extraction technique is not relevant for its category.

On the opposite, hand- crafted features are just committed to a restricted kind of skin sicknesses due to its numerous nature and aren't suitable to be implemented to instructions and datasets. One way to clear up this trouble is to apply characteristic studying which removes the need for characteristic engineering and allows the machine to decide which feature to apply through itself. even though many class structures that use feature-mastering were developed, most of them are restrained to derma reproduction or histopathology snap shots. they are specially used for detecting mitosis, that is a cancer indicator.

For maximum of the cases, switch getting to know can be utilized to teach a deep Convolutional Neural community (CNN). also in switch getting to know, instead of education the network from randomly initialized parameters, a pre-skilled network can be used by best-tuning its weights by continuing the lower back propagation. The cause is that the outcomes of some of the initial layers of a nicely-skilled community contain certain usual functions like blobs, edges which might be used in lots of obligations and such features may be carried out directly to a brand new dataset. For the proposed skin diagnosis gadget, switch getting to know is executed by means of satisfactory- tuning ImageNet, that's a pre-trained model together with Caffey, that is a framework in deep gaining knowledge that is used for green and expressive CNN training.

Dermatological diseases are one of the complex branches of technology due to its complexities within the prognosis of illnesses and their version inside the

changing surroundings. skin illnesses are the most not unusual among them mainly susceptible to unfold and can prove to be fatal main to pores and skin cancer if now not treated in its earlier ranges. studies induce that one-5th of the people are likely to be affected by pores and skin cancer of their lifetime and therefore making it analysis greater challenging.

The maximum time-honored generation used for the prediction of sicknesses is artificial Intelligence using Deep learning wherein AI can be used to broaden algorithms to research the behavior and styles of the disorder and in flip, the machines make use of these mastering methods to manner the pictures and predict.

Synthetic Intelligence stimulates human intelligence in machines to make them think like humans to inherit the learning and hassle- fixing developments that the human brains showcase and thereby taking the nice moves that permit them to gain its goal in minimum time. It makes the machines mimic it and execute responsibilities, from the best to those which are even greater complicated the usage of its goals which include gaining knowledge of, reasoning, and notion.

A number of the preceding benchmarks that used artificial intelligence aren't taken into consideration to encompass them since they're now made as an inherited function in computers. Machines are stressed out using a cross-disciplinary method based on mathematics, laptop technological know-how, linguistics, psychology, and extra.

The computers are consequently provided with the capacity to understand even the complicated theories without being specifically programmed the usage of deep learning which humans locate hard to understand. they are used to process massive complex datasets inside the healthcare enterprise to analyze them into medical insights. in contrast to gadget gaining knowledge of, deep learning uses huge datasets and less classifiers ensuing in greater education time.

CNN models have been proved to have much higher modeling capacity, compared to the previous image recognition mainstream pipelines, e.g., HAAR, SIFT, HOG image features followed by spatial feature encoding, then random forest or support vector classifiers.

Given millions of parameters to fit during model training (much more than previous pipelines), CNN representation empowers and enables computerized image recognition models, with a good possibility to be able to handle more challenging imaging problems.

The primary risk is overfitting since model capacity is generally high in deep learning but often very limited datasets are available. Let us now introduce deep convolutional networks. As their name suggest, CNN apply convolutional filters to compute image features that are useful for classification.

The convolutional filter kernel elements are learned from the raw image training data in a supervised fashion. This is important as it avoids the “handcrafting” of features for classification, while learning features that are useful for the classification problem to be solved.

1.2 CNN ADVANTAGES

The usage of CNN is motivated by the fact that they can capture / are able to learn relevant features from an image /video (sorry I don't know about speech / audio) at different levels similar to a human brain. This is feature learning! Conventional neural networks cannot do this.

Another main feature of CNNs is weight sharing. Let's take an example

to explain this. Say you have a one layered CNN with 10 filters of size 5x5. Now you can simply calculate parameters of such a CNN, it would be $5*5*10$ weights and 10 biases i.e. $5*5*10 + 10 = 260$ parameters. Now let's take a simple one layered NN with 250 neurons, here the number of weight parameters depending on the size of images is '250 x K' where size of the image is P X M and $K = (P * M)$. Additionally, you need 'M' biases. For the MNIST data as input to such a NN we will have $(250*784+1 = 19601)$ parameters. Clearly, CNN is more efficient in terms of memory and complexity. Imagine NNs and CNNs with billions of neurons, then CNNs would be less complex and saves memory compared to the NN.

In terms of performance, CNNs outperform NNs on conventional image recognition tasks and many other tasks. Look at the Inception model, Resnet50 and many others for instance.

For a completely new task / problem CNNs are very good feature extractors. This means that you can extract useful attributes from an already trained CNN with its trained weights by feeding your data on each level and tune the CNN a bit for the specific task.

1.3 INTRODUCTION

In the past 10-year amount, the annual variety of malignant melanoma cases has raised by 53 percent, partially due to raised ultraviolet exposure. Though malignant melanoma is one in all the foremost deadly varieties of carcinoma, a quick diagnosing will cause a really high likelihood of survival. CNN methods used to classify skin cancers are presented. A basic requirement for the successful training of deep CNN models is that sufficient training data labelled with the classes are available. For most of the cases, transfer learning can be utilized to train a deep Convolutional Neural Network (CNN).

Also in transfer learning, instead of training the network from randomly initialized parameters, a pre-trained network can be used by fine-tuning its weights

by continuing the back propagation. The reason is that the results of some of the initial layers of a well-trained network contain certain generic features like blobs, edges that are used in many tasks and such features can be applied directly to a new dataset. For the proposed skin diagnosis system, transfer learning is done by fine-tuning ImageNet, which is a pre-trained model along with Caffe, which is a framework in deep learning that is used for efficient and expressive CNN training.

AIM AND SCOPE

2.1 AIM OF THE PROJECT

Skin problems not only injure physical health but also induce psychological problems, especially for patients whose faces have been damaged or even disfigured. Using smart devices, most of the people are able to obtain convenient clinical images of their face skin condition. On the other hand, the convolutional neural networks (CNNs) have achieved near or even better performance than human beings in the imaging field. Therefore, this paper studied different CNN algorithms for face skin disease classification based on the clinical images.

2.2 REAL TIME APPLICATIONS

- This technique can be used to determine the skin disease just with the help of an image.
- It can be used in medical requirements by dermatologists.
- It can be used for daily check by a user to know about their skin.
- It can be used to detect changes in the skin and determine the change.
- It can be used in dermatological researches done by common man to dermatologists.

2.3 LIMITATIONS

These are the limitations regarding my project. This may not affect for others because I have used some different components.

- The position and orientation of the image should be placed correctly or else the prediction may be effected.
- The lesser the trained dataset causes less accuracy in correct predictions.

2.4 ADVANTAGES

- The most advantage of this project is that the human requirement to detect and analyze the skin disease in a particular region just with an image.
- The fast and machine accuracy provides most accurate results so that human efforts can be concentrated on other necessary works.

2.5SCOPE

The main objective is observing the skin for dubious changes will facilitate discover Benign and Malignant at its earliest stages. Early detection of offers you the best likelihood for fortunate treatment. To automate the process of skin cancer detection, this project aims to develop a deep convolutional neural network. The deeper CNN models will give much better performance compared to shallower models.

CHAPTER 3

DESIGN METHODOLOGY

In this section, we will present the main concepts that this project is concerned with. Our main aim of this project is to showcase human emotions by taking both front features and side features, beside these we are taking color images as datasets or from real time live video. We are taking seven basics emotions like happiness, sad, anger, surprise, fear and disgust after that We train CNN and CNN is used to extract features from input data, it captures them, test them and then trained by CNN and at last its classified into different emotions as stated above.

3.1 SKIN DETECTION

The first CNN, consisting of 20 convolution layers with 3×3 filters, is a kind of VGG network. The second is composed of 20 network-in-network (NiN) layers which can be considered a modification of Inception structure. When training these networks for human skin detection, we consider patch-based and whole-image-based training. The first method focuses on local features such as skin color and texture, and the second on the human-related shape features as well as color and texture. Experiments show that the proposed CNNs yield better performance than the conventional methods and also than the existing deep-learning based method. Also, it is found that the NiN structure generally shows higher accuracy than the VGG-based structure. The experiments also show that the whole-image-based training that learns the shape features yields better accuracy than the patch-based learning that focuses on local color and texture only.

3.2 MACHINE LEARNING

Machine Learning (ML) is a subfield of Artificial Intelligence. A simple ML explanation is the one coined by Arthur Samuel in 1959: "... field of study that gives computers the ability to learn without being explicitly programmed". This statement provides a powerful insight in the particular approach of this field. It completely differs from other fields where any new feature has to be added by hand. For instance, in software development, when a new requirement appears, a programmer has to create software to handle this new case. In ML, this is not exactly the case.

The ML algorithms create models, based on input data. These models generate an output that is usually a set of predictions or decisions. Then, when a new requirement appears, the model might be able to handle it or to provide an answer without the need of adding new code.

ML is usually divided into 3 broad categories. Each category focuses on how the learning process is executed by a learning system. These categories are: supervised learning, unsupervised learning, and reinforcement learning.

Supervised learning is when a model receives a set of labeled inputs, which means that they also contain the corresponding belonging class. The model tries to adapt itself in a way that can map every input with the corresponding output class. On the other hand, unsupervised learning receives a set of inputs without them being labeled. In that sense, the model tries to learn from the data by exploring patterns on them. Finally, reinforcement learning is when an agent is rewarded or punished accordingly the decisions it took in order to achieve a goal.

3.3 DEEP LEARNING

Deep learning is a subset of machine learning in artificial intelligence that has networks capable of learning unsupervised from data that is unstructured or unlabeled. Also known as deep neural learning or deep neural network. Deep learning, a form of machine learning, can be used to help detect fraud or money laundering, among other functions.

Deep learning has evolved hand-in-hand with the digital era, which has brought about an explosion of data in all forms and from every region of the world. This data, known simply as big data, is drawn from sources like social media, internet search engines, e-commerce platforms, and online cinemas, among others. This enormous amount of data is readily accessible and can be shared through fintech applications like cloud computing.

However, the data, which normally is unstructured, is so vast that it could take decades for humans to comprehend it and extract relevant information. Companies realize the incredible potential that can result from unraveling this wealth of information and are increasingly adapting to AI systems for automated support.

Deep learning unravels huge amounts of unstructured data that would normally take humans decades to understand and process.

3.4 Convolutional Neural Networks

The study of the visual cortex is closely related to the development of the convolutional neural networks. Back in 1968, Hubel and Wiesel presented a study focused on the receptive fields of the monkey's visual cortex. This study was

relevant because of the striate cortex (primary visual cortex) architecture description and the way that cells are arranged on it. Moreover, it also presented two different type of cells: simple, and complex. The simple ones are focused on edge-like shapes; while the complex ones cover a broader spectrum of objects and they are locally invariant. Therefore, the different sets of cell arrangements in the cortex are able to map the entire visual field by exploiting the correlation of objects and shapes in local visual areas. One of the first implementations inspired by Hubel and Wiesel ideas was one called Neocognitron. Neocognitron is a neural network model developed by Kuniyiko Fukushima in 1980.

The first layer of the model is composed by units that 14 represent simple cells; while the second layer unit's represent complex cells. The implementation of the local invariance property of the visual cortex is the greatest Neocognitron's achievement. Furthermore, the output mapping is one to one. Each complex cell maps to one and only one specific pattern. This paper underlines the idea that in order to solve pattern recognition problems, it might be a better idea to use automatic learning solutions instead of hand designed ones. Since addressing all the different cases that input data could have on a natural way is quite a complex task, machine learning suits better for this purpose. Additionally, a simple pattern recognition system is described. This system consists of two main modules: a feature extractor, which transforms the input data into low dimensional vectors; and a classifier, which most of the time is general purpose and trainable.

A **Convolutional 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.

3.5 DATASET

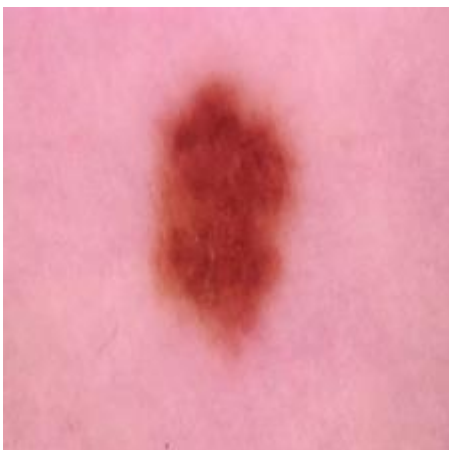
For our project we have collected data from Kaggle, Kaggle is the world's largest data science community with powerful tools and resources to help you achieve your data science goals.

Kaggle allows users to find and publish data sets, explore and build models in a web-based data-science environment, work with other data scientists and machine learning engineers, and enter competitions to solve data science challenges.

Kaggle is open resource, we had training dataset, public test dataset (which is then used as validation dataset for our project), and further a private dataset (same size with public test dataset and will be used as data for evaluating the prediction performance). From Kaggle we took seven basic emotions which are stated below. Using these dataset we train CNN.

While not all the subjects have a corresponding sequence for each label, the labeled emotions on the sequences are:

1. BENIGN
2. MALIGNANT



3.6 LITERATURE SURVEY

An Overview of Melanoma Detection in Dermoscopy Images Using Image Processing and Machine Learning, 2016

The task is divided into 3 stages.

1. Lesion Segmentation
2. Feature Segmentation
3. Classification.

Different methods for each stage are listed.

Melanoma Detection by Analysis of Clinical Images Using CNN, 2017

1. Illumination removal,
Skin texture normalization.
2. K-means algorithm is used for edge detection.
3. For classification purpose, fully connected layers where final diagnosis is done.

Automated Melanoma Recognition in Dermoscopy Images via Very Deep Residual Networks, 2019

1. In this paper, two NN models are used FCRN which is a CNN used for image processing.
2. DRN which is a Fully connected layer.
3. Both FCRN and DRN are residual networks with very deep neural networks.

CHAPTER 4

METHODOLOGY

4.1 PROPOSED SYSTEM

Melanoma, a serious type of skin cancer which can further grow to other body parts if not cured at an early stage. It occurs when DNA is damaged due to UV radiation which can result in uncontrolled cellular growth of melanocytes cells. It is the 19th most identified cancer in women and men. There are almost 3, 00,000 new cases in 2018. The symptoms and signs of melanoma can be identified using ABCDE (Asymmetry, Border, Color, Diameter and Evolving) and the Ugly Duckling Sign. Benign is a noncancerous form. In general, Skin Cancer is tested using a biopsy method, which is a slow process. Thus, an automatic skin cancer detection system can be used to overcome the above stated problem. There are some challenges in analyzing the skin lesion images such as presence of hair, skin lines and ruler marker. These challenges can cause an error in classifying an image. So, an appropriate pre-processing technique is used to overcome the above stated problems. Image Segmentation algorithm is used to acquire ROI by removing the healthy part of skin from the image. Then the CNN algorithm is implemented for classification of images. CNN algorithm automatically extracts features from skin lesion image and classifies it as melanoma or benign.

4.2 PROPOSED SOLUTION

To overcome the existing drawbacks, comparing the traditional machine learning approaches, deep-learning based methods have shown better performance in terms of accuracy and speed of processing in image recognition. We have used a modified Convolutional Neural Network (CNN).

CNN is mostly used in image and face recognition. CNN is a kind of artificial neural network that employs convolutional methodology to extract features from the input data to increase the number of features from live video streaming.

That captures each frame and tests them and is trained by CNN models and later classified into different emotions. With the computational power of Graphical Processing Unit's (GPU's), CNN has achieved remarkably cutting-edge results in image recognition.

4.3 SYSTEM FUNCTIONAL SPECIFICATION

4.3.1 DATA PREPARATION

In this we obtain the dataset from the Kaggle competition. From Kaggle open resource, we had a training dataset, public test dataset (which is then used as validation dataset for our project), and further a private dataset (same size with public test dataset and will be used as data for evaluating the prediction performance). It is noteworthy that in the original provided dataset (either in training dataset or in public test dataset), we have actually in total 2 categories. However, the main problem will be that we have a quite unbalanced distribution of Disgust label in provided data, it accounts for a unusually low percentage, the obvious unbalance in data distribution will influence further neural network training.

4.3.2 Data Preprocess

In this stage we detect face & crop, resize, add noise, and normalizations. It aims to remove background and non- face areas, then crop the face area. Next pre-processing is down-sampling (resize) the resolution to the image to be 32x32, 64x64, 128x128.

IMAGE DATA PRE-PROCESSING-

In the last few years, deep learning has truly come into the mainstream. Deep learning uses neural networks with a lot of hidden layers and requires huge quantities of training data (dozens in today's state of the art). These models have proven especially useful in gaining insight and approaching human-level precision in perceptual tasks such as vision, voice, and language processing. Theory and Methodology. Constructing an efficient neural network model involves close consideration of both the design of the network and the format of input data. This article is concerned with the latter. Number of images, image height, image width, number of channels, and number of levels per pixel are the most widely used image data input parameters. The colours Red, Green, and Blue are usually represented by three data channels.

4.3.3 Feature extraction

We have reduced the redundant data and extracted the implicit features of facial expressions. Feature extraction is a step in the dimensionality reduction process, which separates and reduces a wide collection of raw data into smaller classes. When you have a broad data set, the technique of extracting the characteristics is useful and you need to minimise the number of resources without missing any significant or relevant information. The image's dimensions are 28 x 28. And if you want to check, then you can check by counting the number of pixels. But, in the case of a color image, we have three channels or matrices. The colors are red, green, and blue. Thus, in these three matrices, each matrix has values between 0-255, reflecting the strength of the pixel's color. If you have a color picture, such as the dog image we have, please send it to us.

4.3.4 MODEL ARCHITECTURE

Here, we first approached this project by specifying clearly our primary purpose--classify provided greyscale pictures into one out of six labels. The picture classification task naturally leads us to the well-known Convolutional Neural Network (CNN). After researching how CNN usually performs for greyscale

pictures, we decided to start from three sequential convolutional layers followed by a maxpooling layer, a common activation function for convolutional layer "relu" is used, as well as the same padding pattern. Then, from the basic structure, we further added more and more convolutional layers with different features captured. The features to be captured from the convolutional layer increased from 32 to 128, it is suggested that such hierarchical structure (with increasing layer nodes) performs better for deep neural networks. Finally, the convolved layer is first flattened and then go through two more dense layers to reach the output layer in which softmax activation function is used for multiclass classification (six classes in total).

CNN is a kind of artificial neural network that employs convolutional methodology to extract features from the input file to increase the amounts of features from live video streaming. It's also a Deep Learning algorithm that can take an input image, assign importance (learnable weights and biases) to various aspects/objects in the image, and discriminate between them. Due to its high precision, CNNs are used for image classification and recognition. CNN is a Deep Learning algorithm that can take in an image as input, assign importance (learnable weights and biases) to various aspects/objects in the image, and distinguish one from the other.

4.3.5 CLASSIFICATION

In the classification step we classify the images and predict the disease as -

- benign
- malignant

4.4 SYSTEM ARCHITECTURE

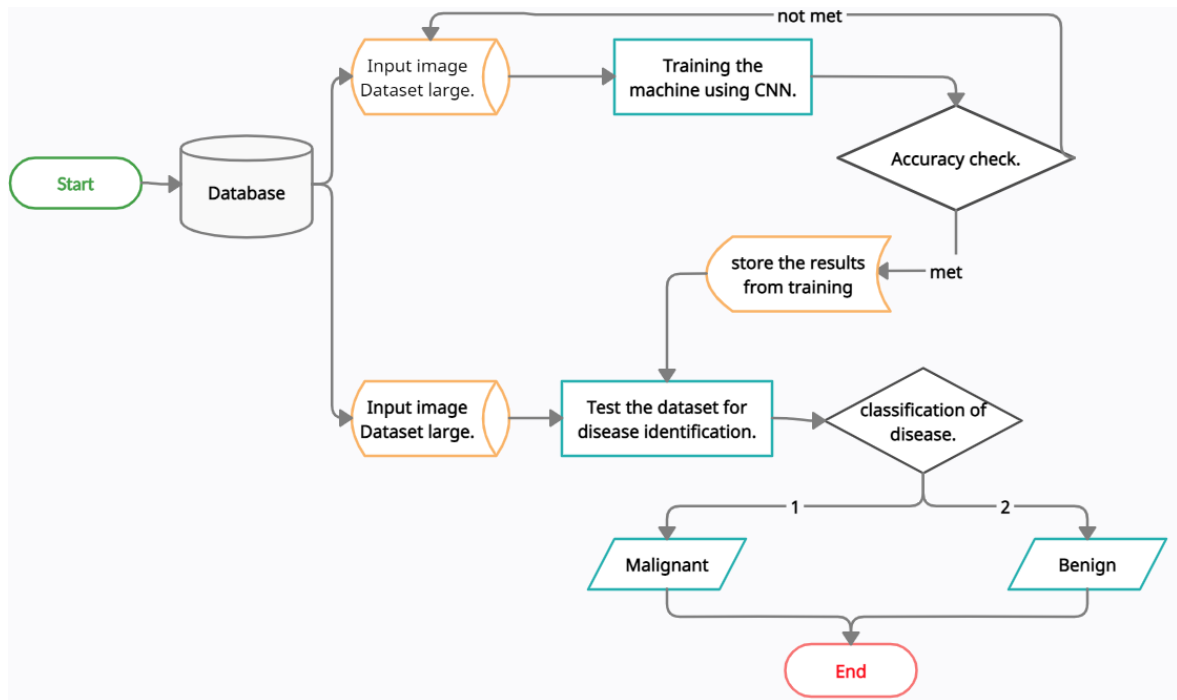
The elements of a convolutional neural network, such as convolutional and pooling layers, are relatively straightforward to understand.

The challenging part of using convolutional neural networks in practice is how to design model architectures that best use these simple elements.

A useful approach to learning how to design effective convolutional neural network architectures is to study successful applications. This is particularly straightforward to do because of the intense study and application of CNNs through 2012 to 2016 for the ImageNet Large Scale Visual Recognition Challenge, or ILSVRC. This challenge resulted in both the rapid advancement in the state of the art for very difficult computer vision tasks and the development of general innovations in the architecture of convolutional neural network models.

We will begin with the LeNet-5 that is often described as the first successful and important application of CNNs prior to the ILSVRC, then look at four different winning architectural innovations for the convolutional neural network developed for the ILSVRC, namely, AlexNet, VGG, Inception, and ResNet.

By understanding these milestone models and their architecture or architectural innovations from a high-level, you will develop both an appreciation for the use of these architectural elements in modern applications of CNN in computer vision, and be able to identify and choose architecture elements that may be useful in the design of your own models.



4.5 MODULES

Modules are -

1. data analysis
2. prediction
3. classification

4.6 LIBRARIES

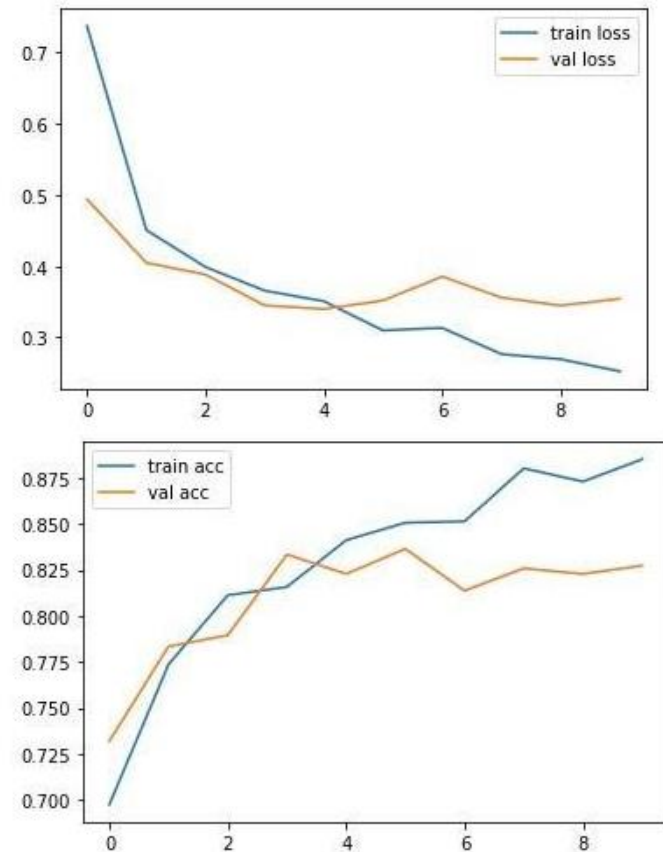
- from tensorflow.keras.layers - Input, Lambda, Dense, Flatten, Conv2D, MaxPooling2D for taking values, to analyse the image and their conversions.
- from tensorflow.keras.preprocessing - image from preprocessing the image for noise and other disturbances.
- from tensorflow.keras.preprocessing.image - ImageDataGenerator, load_img to generate data from the image and also to load the images in specific requirements for training and testing.

- from tensorflow.keras.models - Sequential to follow in an ordered algorithm to load images while testing and training the images by the machine.
- numpy as np to work with the array data while classification.
- from glob - glob to retrieve files/path names matching a specified pattern.
- matplotlib.pyplot as plt used for 2D graphics in python programming language.

CHAPTER-5

RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS

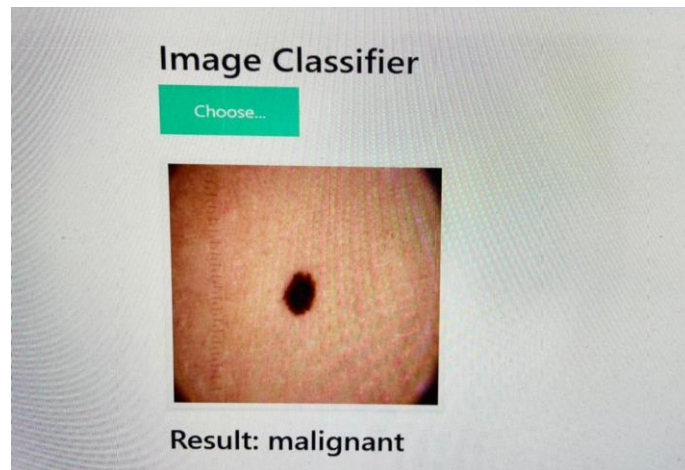
5.1 ANALYSIS



<Figure size 432x288 with 0 Axes>

- The graph shows the accuracy of the trained data.
- The above graph shows the accuracy in the trained data set.
- The below graph shows the error in the trained data set.
- The use of data analysis is we'd be able to know what kind of data we have in the train data set.

As the machine predicts the disease from the image using CNN the disease is now classified and displays the name of it. The image tells whether an image is Benign or Malignant by selecting the path of an image by using CNN algorithm.



5.1.3 Analysis Output

CONCLUSION

5.2 FUTURE DEVELOPMENT

In this project we are able to concentrate on two main diseases for the future development. We would like to develop this to analyze and classify not only these two diseases but also other kinds.

And in future purposes we would like to develop to detect and classify not only diseases but also to several skin based medical support. So that right medical treatment is given with faster analysis and classification of the disease.

Sometimes with lack of right analysis knowledge people hesitate to visit doctors considering a minor issue developing an application for mobiles may help people get their condition cleared and also the machine learns more cases with this information database.

5.3 CONCLUSION

Here by we came to an end of the project with Convolutional neural networks (CNNs) have accomplished astonishing achievements across a variety of domains, including medical research, and an increasing interest has emerged in radiology. Although deep learning has become a dominant method in a variety of complex tasks such as image classification and object detection. Using this the machine analyses the image and finally classifies into the respective disease.

REFERENCES

1. Nabin K. Mishraa and M. Emre, "An Overview of the Melanoma Detection of Dermoscopic Images Using Image Processing and Machine Learning", 2019
2. E. Nasr-Esfahani, S. Samavi, N. Karimi, "Melanoma Detection by Analysis of Clinical Images Using CNN", IEEE,2018
3. Lequan Yu, Hao Chen, Qi Dou, "Automated Melanoma Recognition in Dermoscopy Images via Very Deep Residual Networks", IEEE, 2016.
4. dria Romero Lopez, Xavier Giro-i-Nieto, "SKIN LESION CLASSIFICATION FROM DERMOSCOPIC IMAGES USING DEEP LEARNING TECHNIQUES ", IEEE,2017.
5. Danilo Barros Mendes, "Skin Lesions Classification Using Convolutional Neural Networks in Clinical Images", arXiv, 2018.

APPENDIX

A. SCREEN SHOTS

```
[ ] y_pred = model.predict(test_set)
```



y_pred

```
array([[5.60033619e-01, 4.39966381e-01],  
       [4.81014848e-01, 5.18985152e-01],  
       [1.25508867e-02, 9.87449050e-01],  
       [5.01491308e-01, 4.98508632e-01],  
       [2.28653818e-01, 7.71346152e-01],  
       [6.77788854e-01, 3.22211206e-01],  
       [7.76692808e-01, 2.23307192e-01],  
       [1.22023389e-01, 8.77976596e-01],  
       [7.64577270e-01, 2.35422686e-01],  
       [6.09069884e-01, 3.90930116e-01],  
       [9.73289480e-06, 9.99990225e-01],  
       [2.07629755e-01, 7.92370200e-01],  
       [6.69044405e-02, 9.33095634e-01],  
       [3.21743965e-01, 6.78256094e-01],  
       [4.14117458e-05, 9.99958634e-01],  
       [3.15875486e-02, 9.68412519e-01],  
       [6.32001221e-01, 3.67998749e-01],  
       [8.05722952e-01, 1.94277048e-01],  
       [2.86496431e-01, 7.13503659e-01],  
       [1.00790849e-03, 9.98992145e-01].
```

Test data accuracy values(part 1)

```
[2.83813007e-04, 7.99714170e-01],
[9.20528964e-06, 9.99990821e-01],
[2.36342862e-01, 7.63657153e-01],
[7.02754706e-02, 9.29724574e-01],
[7.39229560e-01, 2.60770410e-01],
[6.75513506e-01, 3.24486494e-01],
[2.01435417e-01, 7.98564613e-01],
[6.23378515e-01, 3.76621425e-01],
[1.73856914e-01, 8.26143086e-01],
[2.47759432e-01, 7.52240598e-01],
[8.13637972e-01, 1.86362088e-01],
[4.48667270e-04, 9.99551356e-01],
[4.42770213e-01, 5.57229757e-01],
[6.22837007e-01, 3.77163023e-01],
[7.82049060e-01, 2.17950940e-01],
[5.50354898e-01, 4.49645132e-01],
[5.17908158e-03, 9.94820952e-01],
[6.24737978e-01, 3.75261992e-01],
[9.09389317e-01, 9.06107053e-02],
[8.93039763e-01, 1.06960312e-01],
[7.03334212e-01, 2.96665758e-01],
[5.44862568e-01, 4.55137461e-01],
[4.70748931e-01, 5.29251039e-01],
[5.79282343e-02, 9.42071736e-01],
[9.32975113e-02, 9.06702459e-01],
[4.18470234e-01, 5.81529737e-01],
[7.12964311e-02, 9.28703606e-01]], dtype=float32)
```

Test Data Accuracy Values(part 2)

+ Code + Text

```
[ ] import numpy as np
    y_pred = np.argmax(y_pred, axis=1)
```

▶ y_pred

```
array([0, 1, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0,
       0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1, 0, 0, 1,
       0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0,
       0, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1,
       1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0,
       1, 1, 1, 1, 1])
```

Fig: Test Data Prediction.

```
+ Code + Text

[ ]
import numpy as np
x=np.expand_dims(x,axis=0)
img_data=preprocess_input(x)
img_data.shape

(1, 224, 224, 3)

model.predict(x)

array([[0., 1.]], dtype=float32)

[ ] a=np.argmax(model.predict(x), axis=1)

[ ] if a==0:
    print("benign")
else:
    print('malignant')

malignant
```

Fig: Classification.

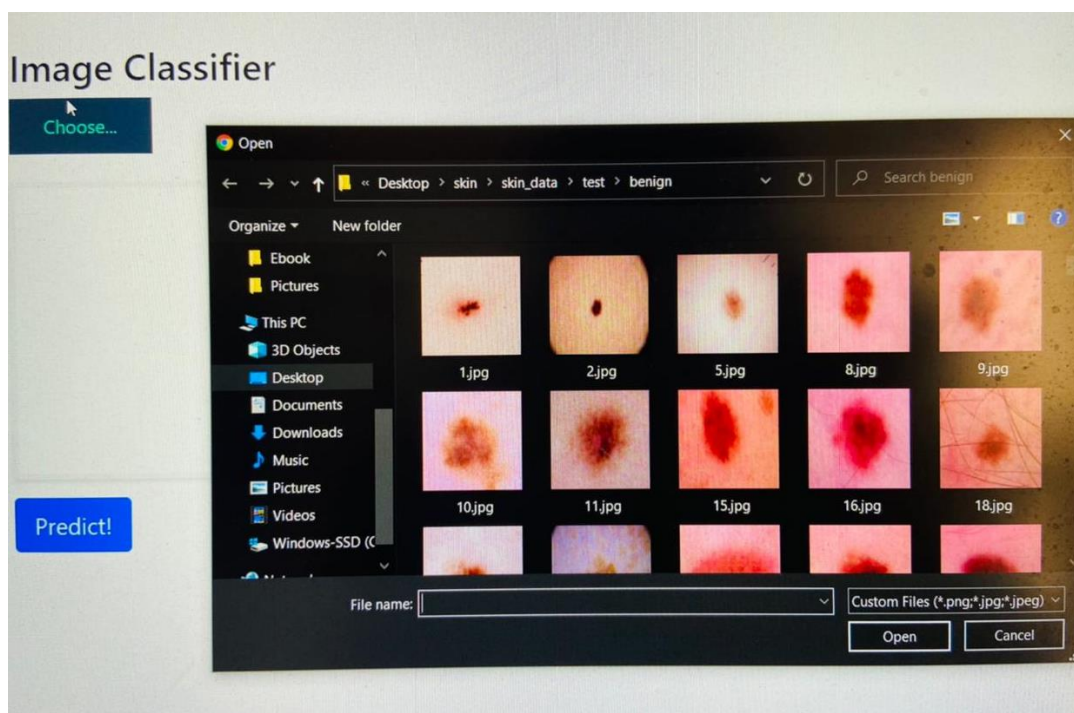


Fig: Image Database.

B. SOURCE CODE

Pylab code:

```
# import the libraries as shown below
from tensorflow.keras.layers import Input, Lambda, Dense, Flatten, Conv2D,
MaxPooling2D
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import ImageDataGenerator, load_img
from tensorflow.keras.models import Sequential
import numpy as np
from glob import glob
import matplotlib.pyplot as plt

#resize the images
img_width= 224
img_height = 224

train= '/content/drive/MyDrive/data/train'
test = '/content/drive/MyDrive/data/test'

folders = glob('/content/drive/MyDrive/data/train/*')
folders

### Create Model from scratch using CNN
model=Sequential()
model.add(Conv2D(filters=16,kernel_size=2,padding="same",activation="relu",input_
shape=(224,224,3)))
model.add(MaxPooling2D(pool_size=2))
model.add(Conv2D(filters=32,kernel_size=2,padding="same",activation = "relu"))
model.add(MaxPooling2D(pool_size=2))
model.add(Conv2D(filters=64,kernel_size=2,padding="same",activation="relu"))
model.add(MaxPooling2D(pool_size=2))
model.add(Flatten())
model.add(Dense(500,activation="relu"))
model.add(Dense(2,activation="softmax"))
model.summary()

28

model.compile(loss='categorical_crossentropy',
```

```

optimizer='adam',
metrics=['accuracy'])
datagen = ImageDataGenerator(rescale = 1./255, validation_split= 0.2)

train_datagen = datagen.flow_from_directory('/content/drive/MyDrive/data/train/',
                                           target_size = (224, 224),
                                           batch_size = 16,
                                           class_mode = 'categorical')

test_set = datagen.flow_from_directory('/content/drive/MyDrive/data/test/',
                                       target_size = (224, 224),
                                       batch_size = 16,
                                       class_mode = 'categorical')

history = model.fit_generator(generator= train_datagen,
                              steps_per_epoch=len(train_datagen),
                              epochs = 10,
                              validation_data= test_set,
                              validation_steps=len(test_set))

# plot the loss
plt.plot(history.history['loss'], label='train loss')
plt.plot(history.history['val_loss'], label='val loss')
plt.legend()
plt.show()
plt.savefig('LossVal_loss')

# plot the accuracy
plt.plot(history.history['accuracy'], label='train acc')
plt.plot(history.history['val_accuracy'], label='val acc')
plt.legend()
plt.show()
plt.savefig('AccVal_acc')

from tensorflow.keras.models import load_model

model.save('/content/drive/MyDrive/Colab Notebooks/model.h5')

y_pred = model.predict(test_set)

```

```
y_pred
```

```
import numpy as np
```

```
y_pred = np.argmax(y_pred)
```

```
y_pred
```

```
from tensorflow.keras.models import load_model
```

```
from tensorflow.keras.preprocessing import image
```

```
from keras.applications.resnet50 import preprocess_input
```

```
model=load_model('/content/drive/MyDrive/Colab Notebooks/model.h5')
```

```
img=image.load_img('/content/ISIC_0000086.jpg',target_size=(224,224))
```

```
x=image.img_to_array(img)
```

```
x
```

```
x.shape
```

```
x=x/255
```

```
x.shape
```

```
import numpy as np
```

```
x=np.expand_dims(x,axis=0)
```

```
img_data=preprocess_input(x)
```

```
img_data.shape
```

```
model.predict(x)
```

```
a=np.argmax(model.predict(x), axis=1)
```

```
if a==0:
```

```
    print("benign")
```

```
else:
```

```
    print('malignant')
```

```

from __future__ import division, print_function
# coding=utf-8
import sys
import os
import glob
import re
import numpy as np

# Keras
from keras.applications.resnet50 import preprocess_input
from tensorflow.keras.applications.imagenet_utils import preprocess_input,
decode_predictions
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image

# Flask utils
from flask import Flask, redirect, url_for, request, render_template
from werkzeug.utils import secure_filename
#from gevent.pywsgi import WSGIServer

# Define a flask app
app = Flask(__name__)

# Model saved with Keras model.save()
MODEL_PATH = 'model1.h5'

# Load your trained model
model = load_model(MODEL_PATH, compile=False)

```

```

def model_predict(img_path, model):
    img = image.load_img(img_path, target_size=(224, 224))

    # Preprocessing the image
    x = image.img_to_array(img)
    # x = np.true_divide(x, 255)
    ## Scaling
    x=x/255
    x = np.expand_dims(x, axis=0)

    # Be careful how your trained model deals with the input
    # otherwise, it won't make correct prediction!
    x = preprocess_input(x)

    preds = model.predict(x)
    preds=np.argmax(preds, axis=1)
    if preds==1:
        preds="malignant"
    else:
        preds="malignant"

    return preds

@app.route('/', methods=['GET'])
def index():
    # Main page
    return render_template('index.html')

```

```

@app.route('/predict', methods=['GET', 'POST'])

```

```

def upload():
    if request.method == 'POST':
        # Get the file from post request
        f = request.files['file']

        # Save the file to ./uploads
        basepath = os.path.dirname(__file__)
        file_path = os.path.join(
            basepath, 'uploads', secure_filename(f.filename))
        f.save(file_path)

        # Make prediction
        preds = model_predict(file_path, model)
        result=preds
        return result
    return None

if __name__ == '__main__':
    app.run(debug=True)

```

C. PUBLICATION REPORT

- Paper was uploaded to the journal.
- Accepted by journal.
- Paper will Publish on the end of MAY.