SKIN CANCER DETECTION USING CONVOLUTIONAL NEURAL NETWORK

Under the Guidance of

Dr. Agilandeeswari L

(Associate Professor, School of Information Technology and Engineering)

For the Course

ITE1015 – Soft Computing

For Winter Semester 2020-21

Project Team:

Aashish Bansal 19BIT0346

Perumalla Sasank 19BIT0338

Keerthi Yasasvi 19BIT0335

TABLE OF CONTENTS

SKIN CANCER DETECTION USING CNN

ABSTRACT

The project is a CNN trained model which can predict whether the patient has a suffering from Cancer or not by checking the images of the infected areas on the body. The model has been trained on a variety of images through which it predicts the required.

In this project, the image file of the patient is upload into a software, which is GUI-based interface, developed with the help of Tkinter, and it consists of the model saved as a file and the software uses that to analyse the image and give the prediction which can help doctors to start with the medication way faster instead of waiting for the laboratory reports for the confirmation.

So basically,

- Skin cancer is an abnormal growth of skin cells. Most skin cancers are caused by exposure to ultraviolet (UV) light. When the skin is not protected, UV rays from sunlight or tanning beds can damage and alter skin's DNA that leads to the cancer.
- Deep learning model has been built to classify and identify the binary diagnostic group of melanocytic images obtained through dermoscopy.
- Based on the model, disease detection through dermal cell images has been investigated, and classifications on dermal cell images have been performed.

KETWORDS

- Model
- Convolutional Neural Network
- Cancer
- Malignant
- Benign
- Detection
- Tkinter
- Software
- Analysis
- Uploading
- Training
- Testing
- Validation
- prediction

INTRODUCTION

- The significant growth of medical images and techniques requires comprehensive and exhaustive efforts from a medical professional who is susceptible to human error and the result can also vary widely among various experts.
- In this project, we have used the above stated idea behind disease detection, to develop a system using convolutional neural network (CNN) that will help in detection of a particular disease.
- The system has been made user-friendly with the help of GUI, so that it can be used not only by the medical professionals but also by the population at large.

DATASET - **SIIM-ISIC MELANOMA CLASSIFICATION**

WHAT SHOULD I EXPECT THE DATA FORMAT TO BE?

The images are provided in DICOM format. This can be accessed using commonly-available libraries like `pydicom`, and contains both image and metadata. It is a commonly used medical imaging data format.

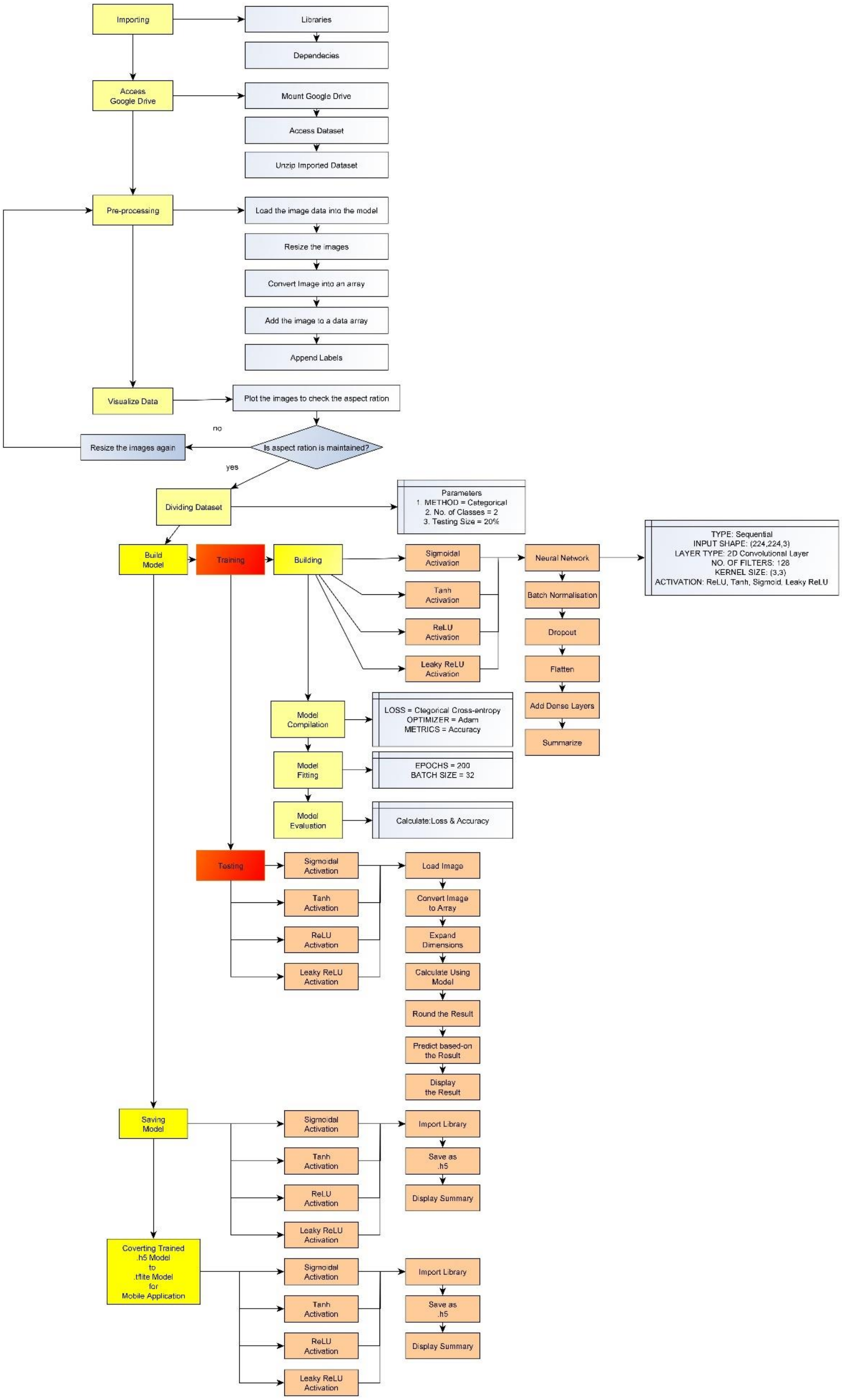
Images are also provided in JPEG and TFRecord format (in the `jpeg` and `tfrecords` directories, respectively). Images in TFRecord format have been resized to a uniform 1024x1024.

Metadata is also provided outside of the DICOM format.

WHAT ARE WE PREDICTING?

We are predicting a binary target for each image. The model should predict the probability (floating point) between 0.0 and 1.0 that the lesion in the image is malignant (the target). In the training data, the value 0 denotes benign, and 1 indicates malignant.

GENERAL ARCHITECTURE AND MODEL PROCESS FLOW



ABOUT THE DISEASES

Skin cancer is the most prevalent type of cancer. Melanoma, specifically, is responsible for 75% of skin cancer deaths, despite being the least common skin cancer. The American Cancer Society estimates over 100,000 new melanoma cases will be diagnosed in 2020. It's also expected that almost 7,000 people will die from the disease. As with other cancers, early and accurate detection—potentially aided by data science—can make treatment more effective.

Currently, dermatologists evaluate every one of a patient's moles to identify outlier lesions or “ugly ducklings” that are most likely to be melanoma. Existing AI approaches have not adequately considered this clinical frame of reference. Dermatologists could enhance their diagnostic accuracy if detection algorithms take into account “contextual” images within the same patient to determine which images represent a melanoma. If successful, classifiers would be more accurate and could better support dermatological clinic work.

As the leading healthcare organization for informatics in medical imaging, the Society for Imaging Informatics in Medicine (SIIM)'s mission is to advance medical imaging informatics through education, research, and innovation in a multi-disciplinary community. SIIM is joined by the International Skin Imaging Collaboration (ISIC), an international effort to improve melanoma diagnosis. The ISIC Archive contains the largest publicly available collection of quality-controlled dermoscopic images of skin lesions.

In this competition, you’ll identify melanoma in images of skin lesions. In particular, you’ll use images within the same patient and determine which are likely to represent a melanoma. Using patient-level contextual information may help the development of image analysis tools, which could better support clinical dermatologists.

Melanoma is a deadly disease, but if caught early, most melanomas can be cured with minor surgery. Image analysis tools that automate the diagnosis of melanoma will improve dermatologists' diagnostic accuracy. Better detection of melanoma has the opportunity to positively impact millions of people.

MOTIVATION

- Disease detection plays a very important role in the process of diagnosis. Therefore, the motivation lies in accurate classification and detection of the diseases based on medical images.
- The main aim is to minimize the chances of error that might happen due to the doctor's misjudgement.
- Developing a system that will not only help in detecting the diseases efficiently but will also save the time and effort of the medical practitioners.
- This will also save the patients from running to the doctor to get their medical reports verified.

METHODOLOGY AND SYSTEM DESIGN

IMAGE PROCESSING

- Image processing can be defined as the technical analysis of an image by using complex algorithms.
- The purpose of early image processing was to improve the quality of the image. Its use has been increasing exponentially in the last decades.
- Its applications range from medicine to entertainment, passing by geological processing and remote sensing.

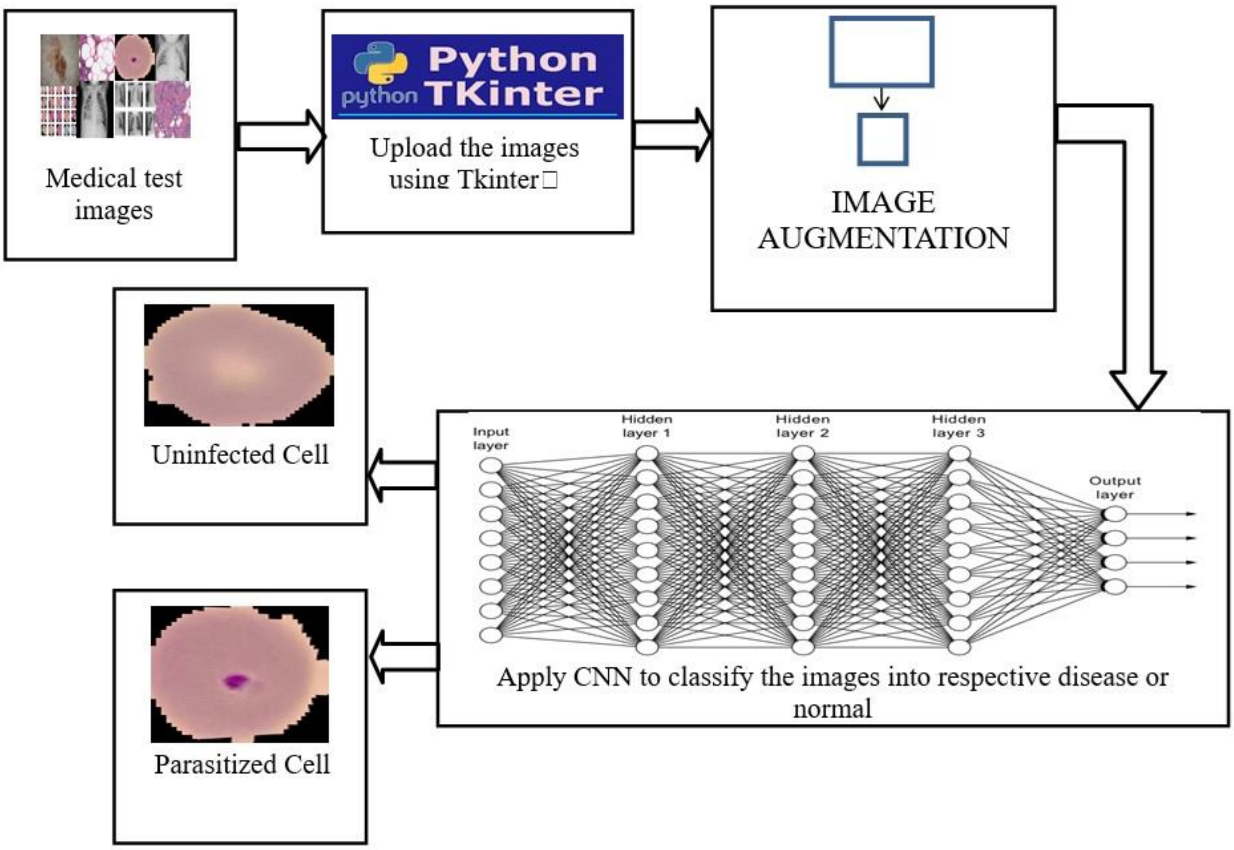
CONVOLUTIONAL NEURAL NETWORK

- A Convolutional Neural Network (CNN) is comprised of one or more convolutional layers (often with a subsampling step) and then followed by one or more fully connected layers as in a standard multilayer neural network.
- The architecture of a CNN is designed to take advantage of the 2D structure of an input image. This is achieved with local connections and tied weights followed by some form of pooling which results in translation invariant features.

- Another benefit of CNNs is that they are easier to train and have fewer parameters than fully connected networks with the same number of hidden units.

PROCESS FLOW DIAGRAM

The model has been described using the simple image given below:



This is the way the project is going to flow. First a medical test image will be taken which will be uploaded to GUI-based software (made with the help of Tkinter) and then the image will be processed using the model which has been saved as a file and has been integrated with the software for faster processing. The model will generate an output which will be returned to the software and the final output will be displayed to the user on the screen, based on which the medical diagnosis for the patient can be started at an early stage and help save many lives due to the delay in the provision of the medication.

FUTURE SCOPE

- Implementation of various other algorithms and using several optimization techniques. Also, more data will be collected in order to recognize the features more accurately.
- Major attention will be given to increase the accuracy such that our proposed system can be used to detect a large number of chronic and critical diseases.
- When these enhancements are done, the system can be integrated with an android application to make it more convenient and easily portable. This will allow people from all strata to use it effectively even if they do not have a personal computer.

REVIEW ON VARIOUS SCHEMES (MENTION APPROPRIATE REFERENCES)

IMAGE ACQUISITION:

In Image Processing, it is defined as the action of retrieving an image from some source, usually a hardware-based source for processing. It is the first step in the workflow sequence because, without an image, no processing is possible. The image that is acquired is completely unprocessed.

For this project, the original dataset has been obtained from the Kaggle and the dataset has been uploaded into the Google Drive as a zip file. With the help of a Python Library, the Google Drive can be authorised using which we can access the dataset by just importing the dataset into the model and then unzipping it.

IMAGE PRE-PROCESSING:

Image processing is divided into analogue image processing and digital image processing. **Digital image processing** is the use of computer algorithms to perform image processing on digital images. As a subfield of digital signal processing, **digital image processing** has many advantages over **analogue image processing**. It allows a much wider range of algorithms to be applied to the input data — the aim of digital image processing is to improve the image data (features) by suppressing unwanted distortions and/or enhancement of some important image features so that our **AI-Computer Vision** models can benefit from this improved data to work on. An image is nothing more than a two-dimensional array of numbers (or pixels) ranging between 0 and 255. It is defined by the mathematical function f(x,y) where x and y are the two co-ordinates horizontally and vertically. The value of f(x,y) at any point is giving the pixel value at that point of an image. The dataset is then divided into the train set and test set of 80% and 20% images respectively.

STEPS FOR IMAGE PROCESSING ARE

READ IMAGE

In this step, we store the path to our image dataset into a variable then we created a function to load folders containing images into arrays. But first, we need to import the libraries that we are going to use

RESIZE IMAGE

In this step-in order to visualize the change, we are going to create two functions to display the images the first being a one to display one image and the second for two images. We need to resize the images because some images captured by a camera and fed to our AI algorithm vary in size, therefore, we should establish a base size for all images fed into our AI algorithms.

REMOVE NOISE

we add this code to smooth our image to remove unwanted noise. But for the dataset which we obtained did not have noise.

ROI FROM NON-ROI

Return on investment is the performance measurement and evaluation metric expressed as a ratio or a percentage. There are several ways to calculate ROI, but one of the most common formulas divides net income (gains – cost of investment) by the cost of investment.

The equation is applicable to various industries and looks like this:

ROI= (Gains – Cost of Investment)/Cost of Investment

The calculation is easy if you know values for this formula. In reality, it will take some time to understand if predicted gains and actual gains are the same or at least close to each other. Cost of investment is also an estimate. So, it's about forecasting these values as accurately as possible.

FEATURE EXTRACTION

Feature extraction is a process of dimensionality reduction by which an initial set of raw data is reduced to more manageable groups for processing. A characteristic of these large data sets is a large number of variables that require a lot of computing resources to process. Feature extraction is the name for methods that select and /or combine variables into features, effectively reducing the amount of data that must be processed, while still accurately and completely describing the original data set.

The process of feature extraction is useful when you need to reduce the number of resources needed for processing without losing important or relevant information. Feature extraction can also reduce the amount of redundant data for a given analysis. Also, the reduction of the data and the machine's efforts in building variable combinations (features) facilitate the speed of learning and generalization steps in the machine learning process. The dataset comprising of RGB images of skin samples has been taken. The images are 1024x1024 which are resized to 224x224 pixels.

CLASSIFICATION:

We will be classifying the images as into two categories:

1. Malignant (Cancerous)
2. Benign (Non-cancerous)

NETWORK ARCHITECTURE

TRAINING

First, we need to train the model based on the Neural Network which we have built.

BUILDING

NEURAL NETWORK

We have built a Sequential Neural Network for which we have resized the images to the 224x224 with three channels. The layers of the network are 2D Convolutional Layers with 128 filters and kernel size as (3,3). The Activation functions used are ReLU, Tanh, Sigmoid and Leaky ReLU. Originally, only the ReLU activation function was used but later Tanh, Sigmoid and Leaky ReLU functions were used in order to compare the metrics obtained in order to obtain the best training results from the network.

BATCH NORMALISATION

Training deep neural networks with tens of layers is challenging as they can be sensitive to the initial random weights and configuration of the learning algorithm.

One possible reason for this difficulty is the distribution of the inputs to layers deep in the network may change after each mini-batch when the weights are updated. This can cause the learning algorithm to forever chase a moving target. This change in the distribution of inputs to layers in the network is referred to the technical name "*internal covariate shift*."

Batch normalization is a technique for training very deep neural networks that standardizes the inputs to a layer for each mini-batch. This has the effect of stabilizing the learning process and dramatically reducing the number of training epochs required to train deep networks. The batch normalization has been done over axis -1.

DROPOUT

Deep learning neural networks are likely to quickly overfit a training dataset with few examples.

Ensembles of neural networks with different model configurations are known to reduce overfitting, but require the additional computational expense of training and maintaining multiple models.

A single model can be used to simulate having a large number of different network architectures by randomly dropping out nodes during training. This is called dropout and offers a very computationally cheap and remarkably effective regularization method to reduce overfitting and improve generalization error in deep neural networks of all kinds.

FLATTEN

Flattening is converting the data into a 1-dimensional array for inputting it to the next layer. We **flatten** the output of the convolutional layers to create a single long feature vector. And it is connected to the final classification model, which is called a fully-connected layer.

DENSE LAYER

A dense layer has been added at the end of the network so that all the processed inputs are obtained and compiled together in order to generate an overall output and so that the output from any neuron is not missed which might lead to fatal results.

MODEL COMPILATION

We compile the model with the Categorical Cross entropy as the loss function, Adam as the optimizer and using Accuracy as the metrics.

MODEL FITTING

We fit the model on the training data and since we have a huge dataset, so we have used a greater number of epochs. In this case, we have fit the model with 200 epochs since we have more than 20000 images, so a large dataset will also require a greater number of epochs.

MODEL EVALUATION

After fitting the model, we evaluated the computed results for the loss and accuracy.

TESTING

The process of testing is also similar to the process which we have used for training out model.

LOAD IMAGE

An individual image is loaded from the testing dataset and it tested using the model trained on the data.

CONVERT IMAGE TO ARRAY

We convert the image into an array of numbers because every image is made up of a array of numbers and it is also processed in the form of numbers and not as an image as such, even though it is a single image.

EXPAND DIMENSIONS

We insert a new axis that will appear at the *axis* position in the expanded array shape.

CALCULATE USING MODEL

The result is calculated using the model calculations for the above processed image.

ROUND THE RESULT

We round off the result to 0 or 1 which would make it clear whether the result is malignant or benign for the patient image.

DISPLAY THE RESULT

Store the output in a variable which can be accessed by the software to obtain the result which will be sent for display and output to the patient.

SAVING THE MODEL

The model is saved in the form of a file which is saved with the .h5 extension.

CONVERTING THE MODEL FOR APPLICATION USE

It makes it easier to convert models as part of a model development pipeline. TensorFlow Lite converter takes a TensorFlow or Keras model and generates a .tflite file.

COMPARATIVE STUDY ON VARIOUS SUBTITLES (USING TABLE):

The Literature Survey done for the accomplishment of the project is:

Authors &Year	Methodology or Techniques used	Advantages	Issues	Metrics used
May-20	CNN, AlexNet, ResNet-18, VGG16, SVM, Black-hat filter, Inpaint Algorithm, Median Filter, Otsu's Methodology	SVM Accuracy = 86.21%, ResNet Accuracy = 87%	Accuracy Original Data = 80%, Accuracy Augmented Data = 98.61%	ReLU, CNN with Data Augmentation = 88.87%, CNN without Data Augmentation = 78.96%
2020	CNN, Inception-v3, Keras, TensorFlow, DCNN, LeakyReLU, Adamax optimizer, TPR is similar to the positive predictive value	0.86 AUROC for BKL	0.78 AUROC for MEL	Accuracy
2020	MVSM classifier, CNN, feature extraction, GLCM, SVM, ABCD	dataset which consists of eight different classes is compressed into 800 images and applied, the accuracy achieved is about 96.25%.	accuracy is lowered if minute amounts of foreign elements are found on the sample	Accuracy

2020	CNN SENet154, WSL, Adam, weighted loss-entropy	Efficient Architecture	Not much improved with ensemble strategy	EfficientNet, SENet (T1 = 67.2%, T2 = 70.0%), ResNeXt WSL (T1 = 65.9%, T2 = 68.1%)
2020	GLCM, HOG, GAC	Feature extraction for early detection	Not enough/adequate dataset	ABCD Rule, SVM Classifier, Accuracy, Sensitivity, Specificity using KNN
2019	Multiclass SVM, AlexNet, ReLU	Accuracy – 94.016%	Model used is a pre-trained model, robust	GOPS, L1D miss rate
2019	CNN, pooling layers, dense network, SVM	AlexNet, VGG16, ResNet-18	Deep Network but Small Dataset – 3000 images	Accuracy = 74%
Apr-19	CNN, pooling layer, dense network	Accuracy – 89.5%	Time consuming	Accuracy = 89.5%, Recall = 0.84, Specificity, Precision = 0.8325, F-measure = 0.8325
Mar-19	CNN, Inception V2 Net, K-means Cluster, Max-pooling, Sonification Algorithms	No. of K-means Epochs = 100	F2-score +ve Prediction = 59.9%, High Sensitivity, Low Specificity	F2-score = 81.8%, Sensitivity = 91.7%, Specificity = 41.8%, Precision = 57.3%
2019	CNN, McNemar Test, ResNet50, Bonferroni Correction	MATLAB	Small dataset (11,444), training may be inefficient, class imbalance	Accuracy = 100%, detection rate = 100%
2018-2019	CNN, VGG16, ImageNet	Accuracy – 92.5%, Max-pooling fetches maximum pixel	F1-score = 0.77, VGG16 Accuracy = 78%	Random Forest = 65.9%, XGBoost = 65.15%, SVM = 65.86%, ReLU, Sigmoid

2018	MatConvNet & GoogLeNet Inception V3 CNN, GoogLeNet, AlexNet, ResNet, VGGNet, Simple Majority Voting, SMP	1.28 million NATURAL images 500 epochs, pre-trained models used, MatConvNet provides pre-trained CNN models and some functions to create and initialize new neural networks	Limited computational resources, time-consuming procedures	GoogLeNet Error Rate = 0.1, ResNet Error Rate= 0.02, AlexNet Error Rate = approx. 0.001, VGGNet Error Rate = approx. 0.001
Mohammad Ashraf Ottom 2019	CNN	2000 images provided by ISIC (International Skin Imaging Collaboration)	accuracy 74%	CNN; melanoma; skin cancer; image pre-processing
Abhinav Sagar 2020	CNN, ANN, Fuzzy Measures, SVM	CNN, ANN, Fuzzy Measures, SVM		Accuracy and pression
Muhammad Rukunuddin Ghalib 2017	SVM, KNN, Decision Tree, Boosted Tree	Trained on different models	Improper accuracy	KNN is 92.70%, SVM is 93.70%, Decision tree (DT) is 89.5%, boosted tree (BT) is 84.30%.
Hema Rajini Narayanan 2017	KNN, SVM and CNN	Trained on different models	Very small dataset	KNN 85%, SVM 96% CNN 98%
Dr. SHAILAJA K 2017	SVM	spot analysis and guides for the direction of spread of the cancer	Improper pixel results errors	Edge and line detection
Mohammad AliKadampur 2020	CNN	classify dermal cell images and detect skin cancer.	accuracy of less than 76%	Accuracy Image classification

Titus Josef Brinker 2018	skin lesions using CNNs	large dataset and large variety	dermoscopic patterns are not considered here	Good dataset
Andre Esteva and Brett Kuprel 2017	CNN	129,450 clinical images of dataset and 2,032 different diseases	No pre-processing	Large variety
Noel B. Linsangan 2018	KNN	86.67% accuracy in determining the classification.	Small dataset	Accuracy but small dataset
Daniel J Kadouch 2016	K-Means	Faster identification	Melanoma can be fatal if not diagnosed at early stage.	accuracy of identification of skin cancer from dermoscopic images is directly proportional to the accuracy of the skin lesion segmentation
2017	GANs, CNN, AlexNet, StyleGANs, InceptionV3-StyleGANs, ResNet50-StyleGANs, VGG16BN-StyleGANs	size of 600×600 as input dataset,	sets the weight coefficient w in the SoftMax loss function	Accuracy
2018	CNN, SciKit, Keras, TensorFlow, OpenCV, ReLU	90% accuracy, Convolution maintains the spatial interrelation of the pixels, values of the pixels ranging from 0 - 255 i.e., 256 pixels.	Rectified Linear Unit is a non-linear operation. ReLU acts on an elementary level.	Accuracy
2019	AlexNet, Ordinary CNN, VGG-16, LIN, Inception-v3, and ResNet. Lévy flight, ReLU	size of input images in the input is considered 28×28 pixel.	doesn't give the best global solution	Accuracy

2019	STM32, ROC, CNN, ReLU, NLSC	Accuracy - 99%, F1-Score - 99%	computing and index loss, poor lesion skin discrimination specificity	Accuracy
2019	CNN, Feature Extraction, HSV format	Accuracy of 98%. for melanoma skin cancer detection and 93% for melanoma type, TPR of 94.25%, FPR of 3.56%, and EP of 4%, average accuracy of 91.66%	high error rates, 25.6% Caucasian error and 23.2 Xanthous error, validation loss of 57.56%	Accuracy
2019	CNN, keras, AlexNet, VGG16, SGD optimiser,	trained on more than 126k images, higher image augmentation (24x) and image resolution (1k), the same performances can be achieved using less than 5000 images, no impact of image resize filters	Experiments at 277x277 pixel resolution, Experiments without transfer learning	Accuracy
2019	CNN, grad-CAM, TensorFlow, Inception-ResNet-v2, DenseNet121, Xception	consists of 150,223 clinical images from 543 different skin diseases, achieved an accuracy of 87.25 ± 2.24% on the dermoscopic images for four common skin diseases, including SK, BCC, psoriasis and melanocytic nevus.	highest average precision (77.0%)	Accuracy
2019	CNN, keras, TensorFlow, Inception V3, ResNet50, VGG16, MobileNet and InceptionResnet	7 types of skin lesion diseases identification namely: Benign Keratosis, Dermatofibroma, Vascular Lesion, Melanoma, Melanocytic Nevus, Basal Cell Carcinoma and Actinic Keratosis., InceptionResnet achieved an average	low F1 score	Accuracy

		accuracy of 91%, Accuracies of 90 and 91%		

CONCLUSIONS:

- A “health discernment system” has been proposed for medical image classification that will work in real-life scenarios.
- The proposed method is based on ***Convolutional Neural Network*** architecture.
- Different sub-models pertaining to the two diseases (skin cancer: Melanoma, Benign) have been designed using convolutional neural network (CNN) and they have all been tested separately.

REFERENCES:

- ResearchGate.com
- ScienceDirect.com
- GeeksforGeeks.com
- TutorialsPoint.com
- cs.stanford.edu
- Springer.com
- towardsdatascience.com
- academictorrents.com
- kaagle.com