Skin Disease Detection

T.E. mini-project report submitted in partial fulfillment of the requirements of the degree of

Information Technology

SUBMITTED BY:

NAME	CLASS	ROLL NO.
Bhavya B Gada	TEIT - 1	18
Shivangkumar Gandhi	TEIT - 1	19
Pruthvi Rathod	TEIT - 2	60

Under the guidance of

Prof.. Pranoti Nage

Department of Information Technology Atharva College of Engineering, Malad(W)

University of Mumbai

2020–2021

CERTIFICATE

This is to certify that the T.E. mini-project entitled **"Skin Disease Detection"** is a bonafide work of **Bhavya B Gada, Pruthvi Rathod, Shivangkumar Gandhi** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **"Information Technology"** during the academic year 2020–2021.

Ms. Pranoti Nage

Guide

Ms. Nileema Pathak

Co-Guide

Prof. Deepali Maste

Head of Department

Dr. S.P. Kallurkar

Principal

T.E. Mini-Project Report Approval

This mini-project synopsis entitled ***Skin Disease Detection*** by ***Bhavya B Gada, Pruthvi Rathod, Shivangkumar Gandhi*** is approved for the degree of ***Information Technology*** from the ***University of Mumbai***.

Examiners

1.

2.

Date:

Place: Atharva College of Engineering, Malad.

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Signature
Bhavya Gada (TEIT-1 18)

Signature
Shivangkumar Gandhi(TEIT-1 19)

Signature
Pruthvi Rathod (TEIT-2 60)

Date

Table of Contents

Abstract

In this work, we address the problem of skin cancer classification using convolutional neural networks. A lot of cancer cases early on are misdiagnosed leading to severe consequences including the death of the patient. Also, there are cases in which patients have other problems and doctors interpret it as skin cancer. This leads to unnecessary time and money spent for further diagnosis. In this work, we address both of the above problems using deep neural networks and transfer learning architecture. We have used publicly available ISIC databases for both training and testing our network. Our model achieves an accuracy of 0.925, precision 0.91, recall 0.92, F1 score 0.91, and ROC- AUC 0.915 which is better than the previous state-of-the-art approaches.

Deep learning is a set of machine learning methods that were inspired by information processing and distributed communication in networks of biological neurons. Deep learning predominantly involves the development, training, and utilization of artificial neural networks (ANNs). ANNs are networks of artificial neurons that are based on biological neurons. Every ANN has at least 3 layers: an input layer that takes the input, a hidden layer that trains on the dataset fed to the input layer, and an output layer that gives an output depending on the application.

Deep learning has been gaining wide acclaim because of the results it achieves that have never been seen in any other machine learning method. Convolutional Neural Networks (a type of ANNs), are extensively used for image-based applications and have achieved better results than humans in object detection and classification

List of Figures

List of Tables

Chapter 1

Introduction

Skin diseases are more common than other diseases. Skin diseases may be caused by a fungal infection, bacteria, allergy, or viruses, etc. Skin disease may change the texture or color of the skin. In general, skin diseases are chronic, infectious, and sometimes may develop into skin cancer. Therefore, skin diseases must be diagnosed early to reduce their development and spread. The diagnosis and treatment of a skin disease take a long time and causes the financial and physical cost to the patient. In general, most common people do not know the type and stage of skin disease. Some of the skin diseases show symptoms several months later, causing the disease to develop and grow further. This is due to the lack of medical knowledge in the public. Sometimes, a dermatologist (skin specialist doctor) may also find it difficult to diagnose the skin disease and may require expensive laboratory tests to correctly identify the type and stage of the skin disease. The advancement of lasers and photonics-based medical technology has made it possible to diagnose skin diseases much more quickly and accurately. But the cost of such diagnosis is still limited and very expensive. Therefore, we propose an image processing-based approach to diagnose skin diseases. This method takes the digital image of disease effect skin area then use image analysis to identify the type of disease. Our proposed approach is simple, fast, and does not require expensive equipment other than a camera and a computer.

1.1 Motivation

Dermatological or skin diseases are the most widespread and common diseases all over the globe. More than a quarter a million people are suffering from a type of skin disorder known as Psoriasis. People suffering from skin diseases report that the skin conditions or disorders that they suffer from negatively affect their quality of life. This in turn makes their day-to-day tasks harder. If these skin disorders are not caught and identified at an early stage, they may get worse and worse with time. It may further lead to spreading and worse conditions. They may spread from one part to another or may spread from one individual to another. The best bet in these cases is to identify them at the earliest of stages. The attributes that skin diseases show are so diverse, hence it becomes a very difficult job to come up with a system or algorithm that can correctly and efficiently detect them. Both, the color of the skin as well as skin itself color plays a vital role in the detection of skin diseases as they vary from person to person. The main objective of detecting sharp changes or sharp variations in pixel brightness or edges in images is to seize various features. It can be demonstrated that discontinuities in the brightness of pictures are probably due to the following reasons: Lack of symmetry of no continuation in-depth, Lack of continuity in surface orientation, Different materials pose different issues with their inherent properties Different images have different radiance or lighting which depends on how the image was captured.

1.2 Problem Statement

Now day's skin diseases have become a more common problem in human life. Most of these diseases are dangerous and harmful, particularly if not treated at an initial stage. It has become an important thing to treat these skin diseases properly at the earlier stages itself to prevent serious damage to the skin. This system would help to solve this problem to a great extent. Since the system would allow users to determine the skin diseases to provide treatments or advice to the patient by making use of images of skin infected with the disease and by obtaining information from the patient.

1.3 Objectives

The objective of this project is to identify skin lesions based on the input skin images texture analysis based on thresholding and neural network to detect and diagnose skin disease. Our objective of the project is to detect the type of skin disease easily with accuracy. The goal of this project or objective is to develop a system which recognizes skin diseases or rather classify with utmost accuracy among skin diseases.

Objectives are as follows:

1. Obtain a public/private image dataset for skin diseases.
2. Preprocess the images for standardization of the dataset.

   The aim of pre-processing an image is to convert the image into a more suitable and usable form. Pre-processing suppresses the unwanted noise and distortions, enhances colours along with other image features. This helps in further processing steps. Image pre-processing methods can also be for various plots. Neighbouring pixels of a single object in an image have essentially the same or similar brightness. This allows the system to recognize the object and separate it. Image is subjected to orientation correction, resize, noise removal, grayscale conversion, colour enhancement so as to segment it properly.

3. Train dataset using different CNN models and analyse accuracy to decide the preferred CNN model.

   Deep Learning being a part of machine learning procedure stimulated by assembly and gathering of the human brain usually identified as neural networks. CNN (Convolution Neural Networks) is a course of deep learning

procedure that is typically castoff for examining the clear substances such as imageries and videotapes. Through the improvement of CNN, around has remained affected development detected to solve numerous organization based glitches in medicinal images examination. The rudimentary procedure for CNN founded skin disease images organization is obtainable in Figure

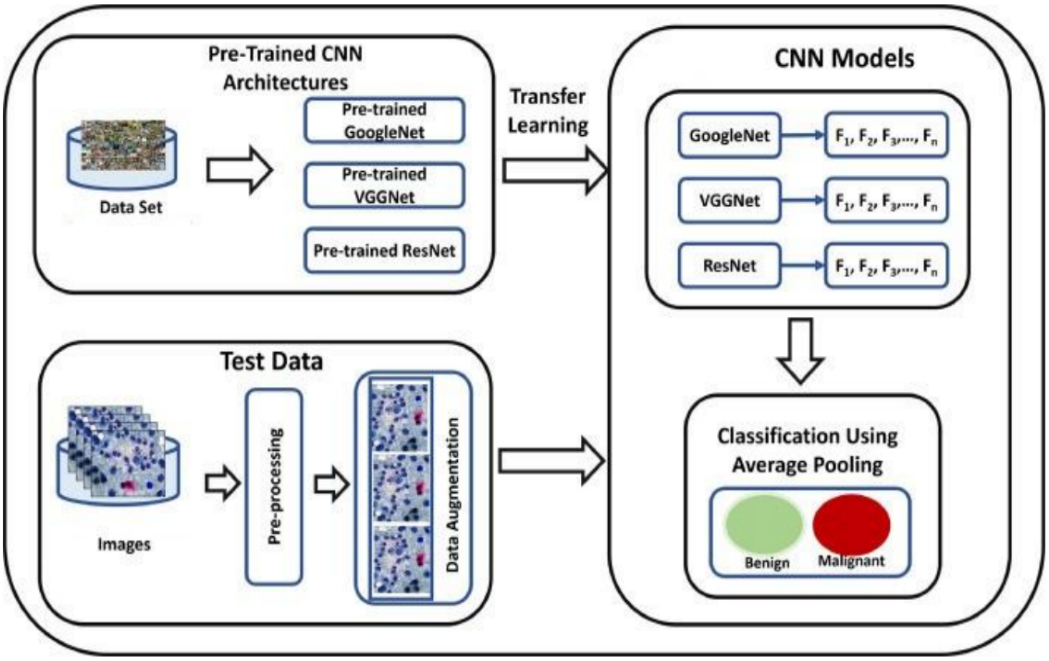


Fig. 1.1

4. Use test images and feed into CNN models to get accurate classification.
5. To show that the system is ready to predict the skin class as malignant or benign as per features of the image.

Our aim is to provide a comprehensive view on prediction of breast cancer through Machine Learning through both image and data analyses, which can play a pivotal role in prevention of misdiagnosis in future.

1.4 Scope

The scope of this project is the classification of benign and malignant breast Cancer using supervised machine learning algorithms based on image datasets.

Breast cancer has been identified as the second leading cause of death among women worldwide after lung cancer and hence, it becomes extremely crucial to identify it at an early stage, which can considerably increase the chances of survival. The most important part in cancer detection is to be able to differentiate between benign and malignant tumors and this is where the work of Machine Learning comes in. Taking all the dependent features upon consideration, Supervised Machine Learning methods allow for classification with higher degree of accuracy and improve upon the misdiagnosis of the physicians, which might occur almost 20% of the time. In our paper, we are focusing towards understanding the shortcomings of digital mammograms in detection of breast cancer and utilize Machine Learning classifiers for the classification of benign and malignant tumors using image analysis.

Chapter 2

Review of Literature

Sr no	Title of the paper	Methodologies used	Dataset with Performance measures	Drawbacks/Conclusion
1	Disease Classification based on Dermoscopic Skin Images Using Convolutional Neural Network in Teledermatology System IEEE Xplore 2020	Inception V3, MobileNet	MNIST HAM10000 - CNN -72% MobileNet- 58%	The model results from the learning process will be applied to a web classifier. The comparison of predictive accuracy shows that the web-classifier using the CNN Inception V3 model has an accuracy value of 72% while the web-classifier that uses the MobileNet v1 model has an accuracy value of 58%.
2	Automatic diagnosis of skin diseases using convolution neural network ELSEVIER 2020	AlexNet	Dermnet -72.1%	One of the key features of the proposed technique is the vast features generated by the convolutional layer of the network which is used by the final layers of the network for classification. This work can be extended to classify skin diseases into more classes.
3	Detection of Melanoma using Deep Learning Techniques IEEE Xplore 2020	Data Augmentation, Data Segmentation (UNET), Classification using Deep Network	ISBI 2017 dataset 84.2%	A two-stage model has been built for the classification of a skin lesion into melanoma and non-melanoma type. Results obtained for the segmentation stage using UNET are good enough for the targeted application. Good accuracy was obtained from the FCRN

				classification stage, but since the targeted application is for medical diagnosis of false negatives needed to be less. Therefore, increasing the recall factor had to be given more priority than increasing the overall accuracy.
4	Deep Learning in Skin Disease Image Recognition: A Review IEEE Access 2020	VGGNet, Inception, DenseNet	HAM10000 dataset- 80% to 89%	Few data may cause insufficient feature extraction, which will affect the diagnosis and recognition of lesions. One of the significant problems in skin disease image recognition are the difficulty in obtaining a large number of data sets due to the complications in medical image collection and the rare occurrence of individual diseases
5.	Discriminative Feature Learning for Skin Disease Classification Using Deep Convolutional Neural Network IEEE Access 2020	ResNet152, InceptionResNet-V2	Wuhan Union Hospital searched images and unified these images of 14 Classes.	Fine-tune all layers of ResNet152 and InceptionResNetV2 to address the problem of facial skin disease images. The performance of the proposed method can be improved by designing a dataset with the help of a dermatologist to visually organize the taxonomy.
6	Skin Lesion Classification using Convolutional Neural Network with Novel Regularizer	CNN MODEL WITH NOVEL REGULARIZER	ISIC Archives- 86.35%	When the dimensionality is very high and the number of instances is very low, the use of these regularizations is pointless. Likewise, regularization algorithms also

	IEEE Access 2017			have several limitations: 1) It cannot be used for feature selection or feature reduction. 2) Choosing a suitable value of λ is difficult, as it is a continuous value and attempting a million times to select a single suitable value is computationally expensive and time-intensive.

Table 2.1

Chapter 3

Requirement Analysis

Requirements analysis in systems engineering and software engineering encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product, taking account of the possibly conflicting requirements of the various stakeholders, such as beneficiaries or users. A software requirements specification (SRS) is a document that is created when a detailed description of all aspects of the software to be built must be specified before the project is to commence. It is important to note that a formal SRS is not always written. In fact, there are many instances in which effort expended on a SRS might be better spent in other software engineering activities.Requirements analysis is critical to the success of a development project. Requirements must be actionable, measurable, testable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design. By analyzing different hardware components and familiar software following are hardware & software used in our project:

3.1 Software Requirements

1. Operating System: Windows 7 or newer. (32-bit or 64-bit)
2. Browser: A chromium based web browser. Example: Google Chrome.
3. Any Integrated Development Environment (IDE) or Code Editor. Example: Microsoft Visual Studio.
4. Cloud platform: The cloud platform provides computation-intensive task processing services. The learning ability of the model in the algorithm load module is improved through receiving the skin image data from the edge nodes. The algorithm load module realizes the adaptation with the application

5. requirements by loading different learning models, and the updated model parameters are transmitted to the edge nodes. The resource cognition module cognizes the network resources, and the data cognition module cognizes the application context and network environment context. The two modules act upon each other to carry out the network resource management and allocation to meet service requirements of applications.

   Example: Kaggle Notebook.

3.2 Hardware Requirements

1. Ram: 4GB or more.
2. Processor: i3 7th gen or more.
3. Programming Language: Python 3.6.
4. Hard disk space: 10GB or more.
5. High end GPU for training: Nvidia 1060 6GB or more.

Chapter 4
Report On Present Investigation

4.1 Existing system

There has been a lot of work published in the domain of skin cancer classification using deep learning and computer vision techniques. These works use a lot of different approaches including classification only, segmentation and detection, image processing using different types of filters etc.

Some papers separately used AdaBoost to classify skin lesions. Some used different sets of features including type of lesion, texture, colour etc and neural networks for the making of a robust diagnosis system.

The examples till now only showed algorithms using traditional machine learning techniques, but lately deep learning have proved to be more accurate. The reason is that it automates the feature extraction process completely. It is upto the algorithms to find the better features and train the model accordingly.

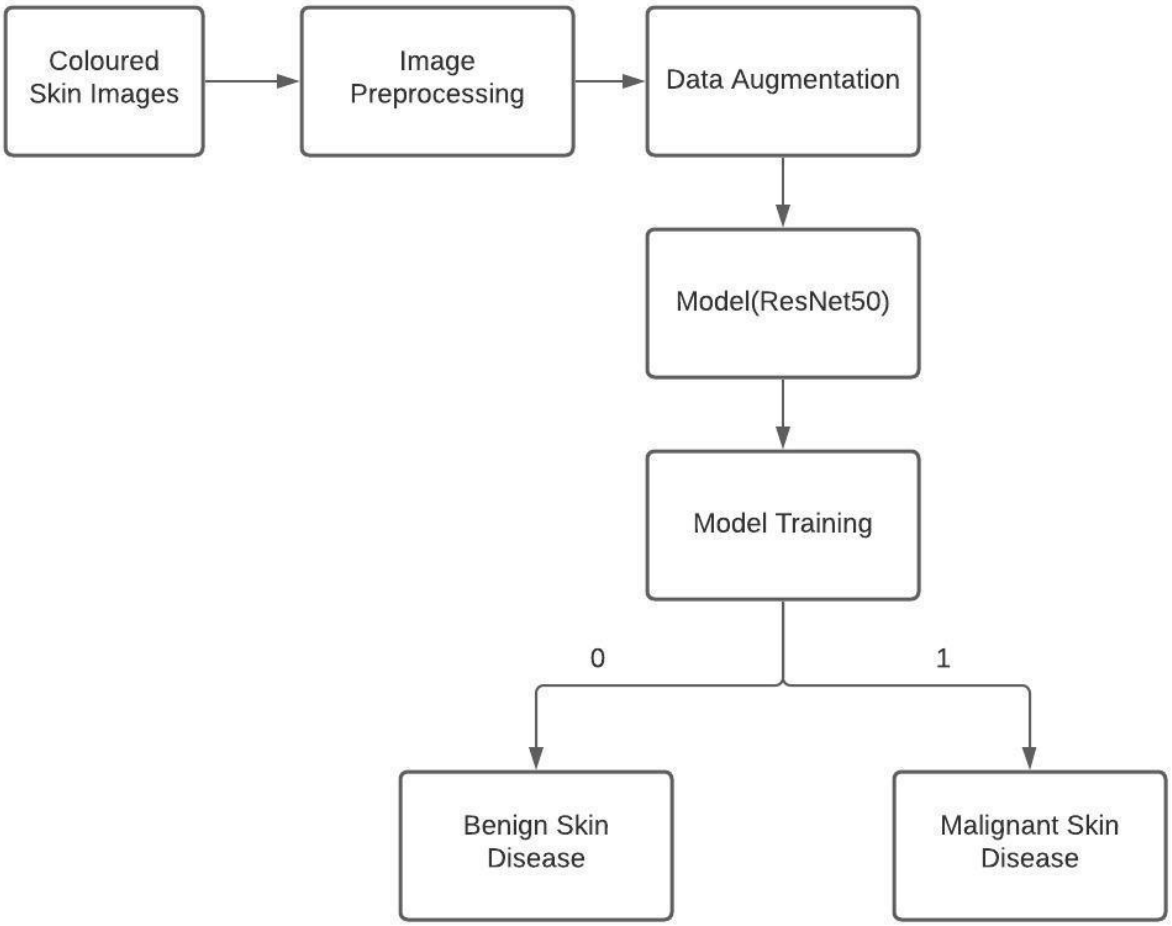
Some projects include Support Vector Machine (SVM), Decision Trees and ANN to classify skin images into Disease Classes.

There has been work done on skin image classification by building Neural Networks from scratch. The accuracy obtained is not so great, leading to miss-classification

The paper we have tackled skin cancer classification which is of binary type i.e. there are 2 classes present in the dataset benign and malignant. We have used the publicly available ISIC database for both training and testing the images. We have compared 5 backbone transfer learning architectures - VGG16, ResNet50, InceptionV3, MobileNet and DesneNet169

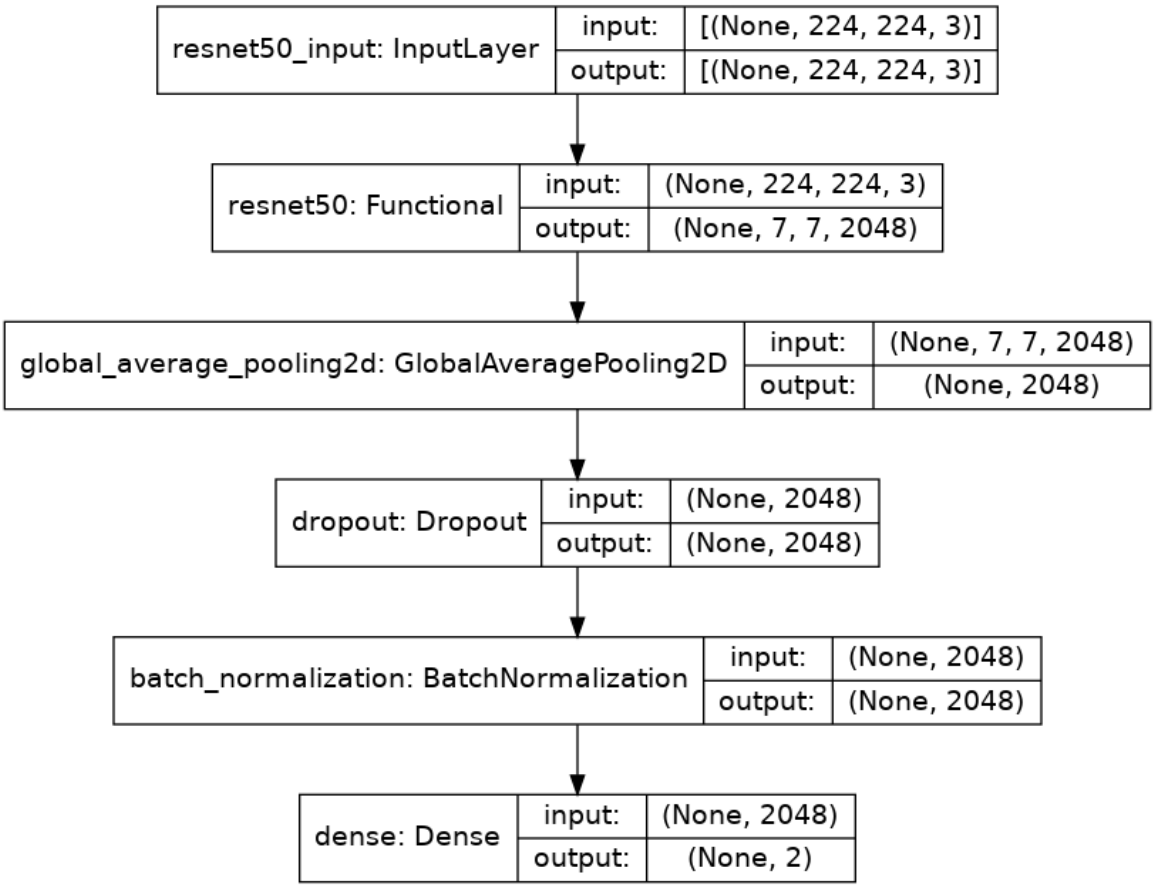
4.2 Implementation

4.2.1 Algorithm/Flowchart



1. Skin Coloured images are obtained from ISIC dataset.. Consisting of Malignant and Benign skin Images.
2. Image Pre-processing: The skin images obtained from the dataset are not of the size (226,226,3) and hence need to be resized accordingly.
3. Image Augmentation: The dataset consists of Training and testing images, total of 3,374 and need to be augmented to increase the accuracy.
   The augmentation process consists of rotating, flipping, resizing, etc.
4. Model: Model used here is a pretrained model of ResNet50.
5. The training data which is generated through augmentation is used to train and test the model.
6. The model is ready to predict the skin class as malignant or benign as per features of the image.

4.2.2 Block Diagram



4.2.3 Dataset

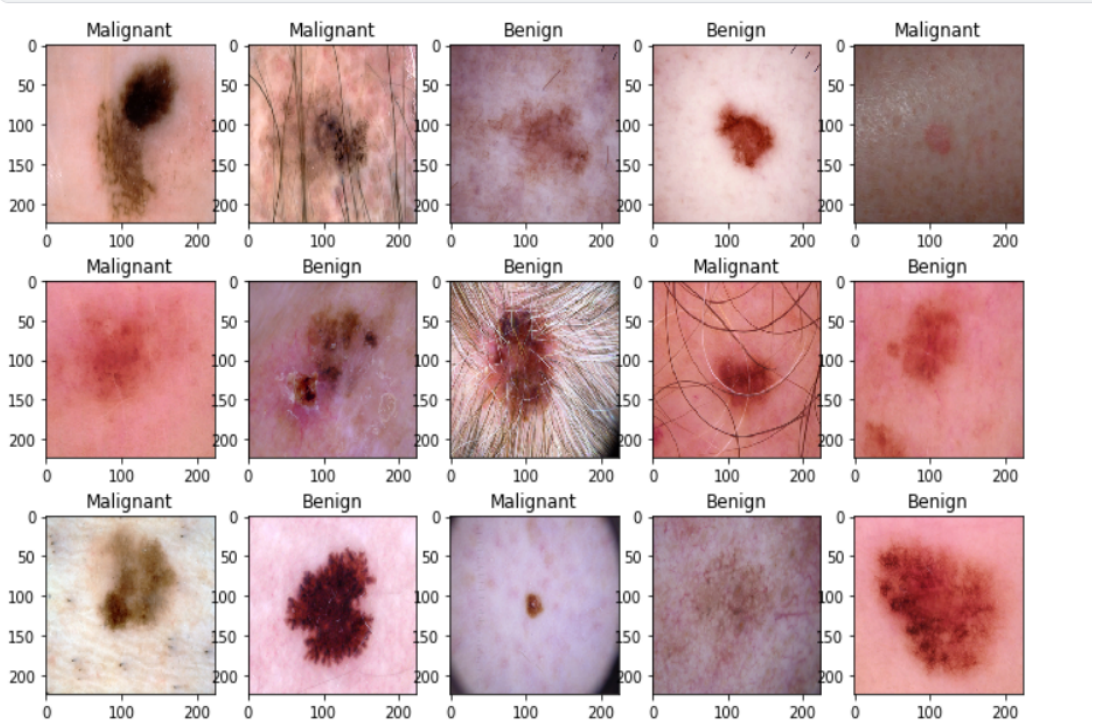


Fig 4.1

Dataset:

Data consist of images in 2 folders of train and test each having Malignant and Benign folders

Train:

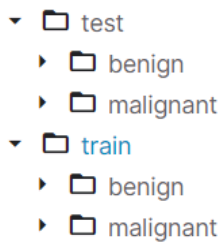
---Benign: 1197 images

---Malignant: 1440 images

Test:

---Benign: 360 images

---Malignant: 300 images



4.2.4 Pseudo Code

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

import os

import json
import math
import os
import cv2
from PIL import Image
import numpy as np
from keras import layers
from keras.applications import ResNet50,MobileNet, DenseNet201,
InceptionV3, NASNetLarge, InceptionResNetV2, NASNetMobile
from keras.callbacks import Callback, ModelCheckpoint, ReduceLROnPlateau,
TensorBoard
from keras.preprocessing.image import ImageDataGenerator
from keras.utils.np_utils import to_categorical
from keras.models import Sequential
from keras.optimizers import Adam
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.metrics import cohen_kappa_score, accuracy_score
import scipy
from tqdm import tqdm
import tensorflow as tf
from keras import backend as K
import gc
from functools import partial
from sklearn import metrics
from collections import Counter
import json
import itertools
%matplotlib inline

def Dataset_loader(DIR, RESIZE, sigmaX=10):
    IMG = []
    read = lambda imname: np.asarray(Image.open(imname).convert("RGB"))
    for IMAGE_NAME in tqdm(os.listdir(DIR)):
        PATH = os.path.join(DIR,IMAGE_NAME)
        _, ftype = os.path.splitext(PATH)
        if ftype == ".jpg":
            img = read(PATH)

            img = cv2.resize(img, (RESIZE,RESIZE))

            IMG.append(np.array(img))
    return IMG
```

```
benign_train =
np.array(Dataset_loader('../input/skin-cancer-malignant-vs-benign/train/be
nign',224))
malign_train =
np.array(Dataset_loader('../input/skin-cancer-malignant-vs-benign/train/ma
lignant',224))
benign_test =
np.array(Dataset_loader('../input/skin-cancer-malignant-vs-benign/test/ben
ign',224))
malign_test =
np.array(Dataset_loader('../input/skin-cancer-malignant-vs-benign/test/mal
ignant',224))
benign_train_label = np.zeros(len(benign_train))
malign_train_label = np.ones(len(malign_train))
benign_test_label = np.zeros(len(benign_test))
malign_test_label = np.ones(len(malign_test))

X_train = np.concatenate((benign_train, malign_train), axis = 0)
Y_train = np.concatenate((benign_train_label, malign_train_label), axis =
0)
X_test = np.concatenate((benign_test, malign_test), axis = 0)
Y_test = np.concatenate((benign_test_label, malign_test_label), axis = 0)

s = np.arange(X_train.shape[0])
np.random.shuffle(s)
X_train = X_train[s]
Y_train = Y_train[s]

s = np.arange(X_test.shape[0])
np.random.shuffle(s)
X_test = X_test[s]
Y_test = Y_test[s]

Y_train = to_categorical(Y_train, num_classes= 2)
Y_test = to_categorical(Y_test, num_classes= 2)

x_train, x_val, y_train, y_val = train_test_split(
    X_train, Y_train,
    test_size=0.2,
    random_state=11
)

# Display first 15 images of moles, and how they are classified
w=40
h=30
fig=plt.figure(figsize=(12, 8))
columns = 5
rows = 3
for i in range(1, columns*rows +1):
```

```
    ax = fig.add_subplot(rows, columns, i)
    if np.argmax(y_train[i]) == 0:
        ax.title.set_text('Benign')
    else:
        ax.title.set_text('Malignant')
    plt.imshow(X_train[i], interpolation='nearest')
plt.show()

BATCH_SIZE = 64


train_generator = ImageDataGenerator(

        zoom_range=2,

        rotation_range = 90,

        horizontal_flip=True,

        vertical_flip=True,

    )


def build_model(backbone, lr=5e-4):
    model = Sequential()
    model.add(backbone)
    model.add(layers.GlobalAveragePooling2D())
    model.add(layers.Dropout(0.5))
    model.add(layers.BatchNormalization())
    model.add(layers.Dense(2, activation='softmax'))


    model.compile(
        loss='binary_crossentropy',
        optimizer=Adam(lr=lr),
        metrics=['accuracy']
    )

    return model

K.clear_session()
gc.collect()

resnet = ResNet50(
    weights='imagenet',
    include_top=False,
    input_shape=(224,224,3)
)
model = build_model(resnet ,lr = 1e-4)
model.summary()

learn_control = ReduceLROnPlateau(monitor='val_acc', patience=5,
                                  verbose=1,factor=0.2, min_lr=1e-7)
```

```
filepath="weights.best.hdf5"
checkpoint = ModelCheckpoint(filepath, monitor='val_acc', verbose=1,
save_best_only=True, mode='max')
history = model.fit_generator(
    train_generator.flow(x_train, y_train, batch_size=BATCH_SIZE),
    steps_per_epoch=x_train.shape[0] / BATCH_SIZE,
    epochs=50,
    validation_data=(x_val, y_val),
    callbacks=[learn_control, checkpoint]
)
# list all data in history
# summarize history for accuracy
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
# summarize history for loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
Y_val_pred = model.predict(x_val)
accuracy_score(np.argmax(y_val, axis=1), np.argmax(Y_val_pred, axis=1))
Y_pred = model.predict(X_test)
tta_steps = 10
predictions = []
for i in tqdm(range(tta_steps)):
    preds = model.predict_generator(train_generator.flow(X_test,
batch_size=BATCH_SIZE, shuffle=False),steps = len(X_test)/BATCH_SIZE)
```

```
    predictions.append(preds)
    gc.collect()

Y_pred_tta = np.mean(predictions, axis=0)
from sklearn.metrics import confusion_matrix

def plot_confusion_matrix(cm, classes,
                          normalize=False,
                          title='Confusion matrix',
                          cmap=plt.cm.Blues):

    if normalize:
        cm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis]
        print("Normalized confusion matrix")
    else:
        print('Confusion matrix, without normalization')

    print(cm)

    plt.imshow(cm, interpolation='nearest', cmap=cmap)
    plt.title(title)
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=55)
    plt.yticks(tick_marks, classes)

    fmt = '.2f' if normalize else 'd'
    thresh = cm.max() / 2.
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, format(cm[i, j], fmt),
                 horizontalalignment="center",
                 color="white" if cm[i, j] > thresh else "black")
    plt.ylabel('True label')
```

```
    plt.xlabel('Predicted label')
    plt.tight_layout()

cm = confusion_matrix(np.argmax(Y_test, axis=1), np.argmax(Y_pred,
axis=1))
#Confusion Matrix
cm_plot_label =['benign', 'malignant']
plot_confusion_matrix(cm, cm_plot_label, title ='Confusion Metrix for Skin
Cancer')
cm = confusion_matrix(np.argmax(Y_test, axis=1), np.argmax(Y_pred_tta,
axis=1))

cm_plot_label =['benign', 'malignant']
plot_confusion_matrix(cm, cm_plot_label, title ='Confusion Metrix for Skin
Cancer')

#Classification Report
from sklearn.metrics import classification_report
print(classification_report( np.argmax(Y_test, axis=1),
np.argmax(Y_pred_tta, axis=1), target_names = ['Benign (Class
0)','Malignant (Class 1)']))

#ROC curve
from sklearn.metrics import roc_auc_score, auc
from sklearn.metrics import roc_curve
roc_log = roc_auc_score(np.argmax(Y_test, axis=1), np.argmax(Y_pred_tta,
axis=1))
false_positive_rate, true_positive_rate, threshold =
roc_curve(np.argmax(Y_test, axis=1), np.argmax(Y_pred_tta, axis=1))
area_under_curve = auc(false_positive_rate, true_positive_rate)
plt.plot([0, 1], [0, 1], 'r--')
plt.plot(false_positive_rate, true_positive_rate, label='AUC =
{:.3f}'.format(area_under_curve))
```

```
plt.xlabel('False positive rate')
plt.ylabel('True positive rate')
plt.title('ROC curve')
plt.legend(loc='best')
plt.show()
plt.close()

i=0
prop_class=[]
mis_class=[]

for i in range(len(Y_test)):
    if(np.argmax(Y_test[i])==np.argmax(Y_pred_tta[i])):
        prop_class.append(i)
    if(len(prop_class)==8):
        break

i=0
for i in range(len(Y_test)):
    if(not np.argmax(Y_test[i])==np.argmax(Y_pred_tta[i])):
        mis_class.append(i)
    if(len(mis_class)==8):
        break

w=60
h=40
fig=plt.figure(figsize=(18, 10))
columns = 4
rows = 2
def Transfername(namecode):
    if namecode==0:
        return "Benign"
    else:
```

```
        return "Malignant"

for i in range(len(prop_class)):
    ax = fig.add_subplot(rows, columns, i+1)
    ax.set_title("Predicted result:"+
Transfername(np.argmax(Y_pred_tta[prop_class[i]]))
                        +"\n"+"Actual result: "+
Transfername(np.argmax(Y_test[prop_class[i]])))
    plt.imshow(X_test[prop_class[i]], interpolation='nearest')
plt.show()

# save it as a h5 file
from tensorflow.keras.models import load_model
model.save('ResNet50_MalVBen93.h5')

from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
model=load_model('ResNet50_MalVBen93.h5')

img=image.load_img('../input/skin-cancer-malignant-vs-benign/test/malignan
t/338.jpg',target_size=(224,224))
plt.imshow(img)
plt.show()

img_array = image.img_to_array(img)
img_batch = np.expand_dims(img_array, axis=0)
prediction = model.predict(img_batch)

if(np.argmax(prediction) == 1):
    print("Malignant")
else:
    print("Benign")
```

We have used the concept of transfer learning for the classification. With transfer learning, instead of starting the learning process from scratch, the model starts from patterns that have been learned when solving a different problem. This way the model leverages previous learnings and avoids starting from scratch. In image classification, transfer learning is usually expressed through the use of pre-trained models. A pre-trained model is a model that was trained on a large benchmark dataset to solve a problem similar to the one that we want to solve. We used three pre-trained models- Inception v3, MobileNet, DenseNet and NasNet as the pre-trained weights for our work.

Pre Trained Model	Accuracy	Precision	Recall	F1 Score	ROC-AUC
ResNet50	0.923	0.915	0.926	0.916	0.915
InceptionV3	0.901	0.891	0.908	0.893	0.897
MobileNet	0.897	0.894	0.892	0.891	0.894
DenseNet	0.905	0.901	0.891	0.904	0.899
NasNet	0.579	0.643	0.599	0.548	0.593

Table 4.1

4.2.5 Screenshots with Output:

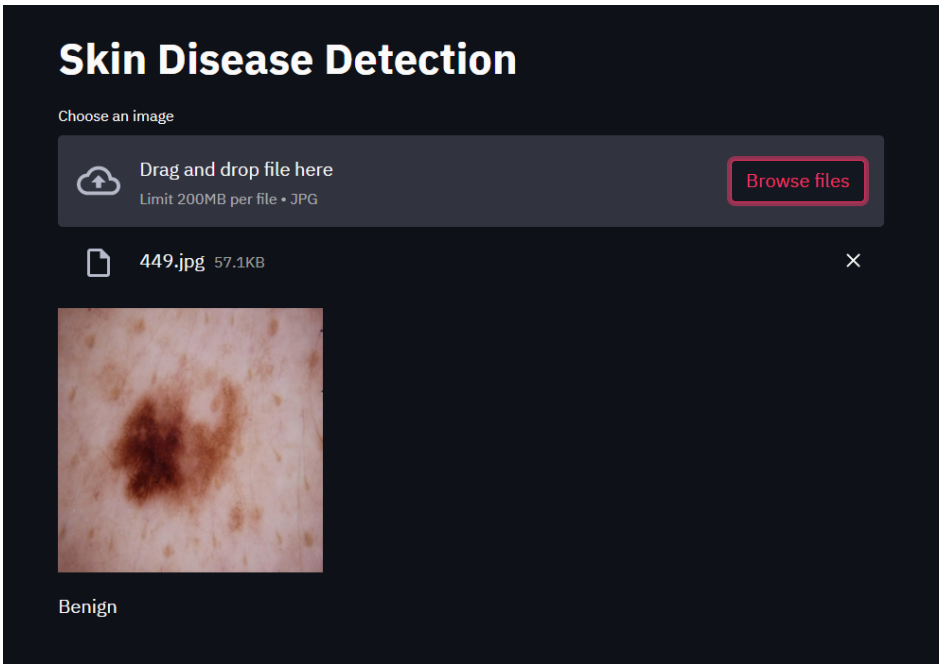


Fig 4.12

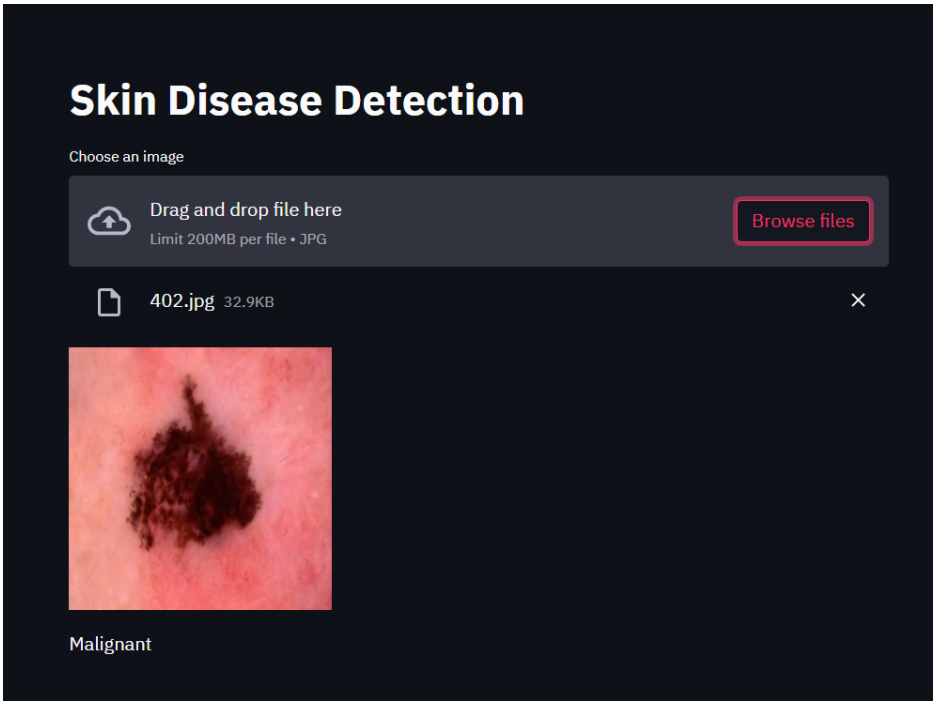


Fig 4.13

Chapter 5
Results and Discussion

Comparison of Accuracy:

Resnet50:

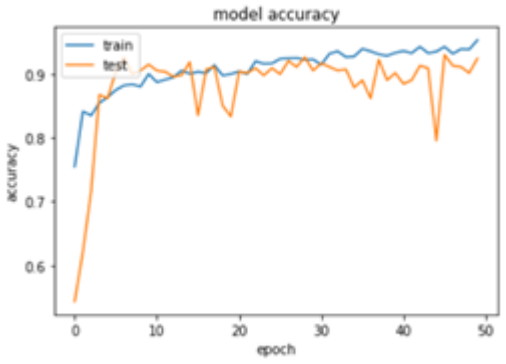


Fig 4.2

DenseNet

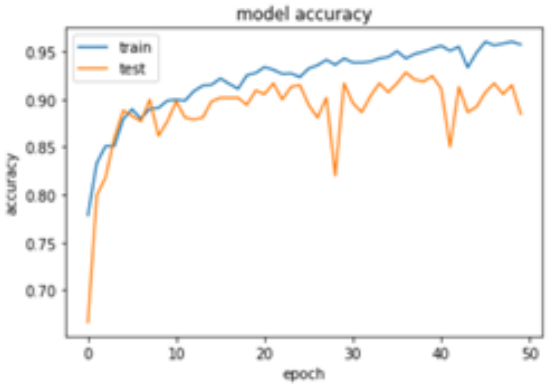


Fig 4.3

Inception V3:

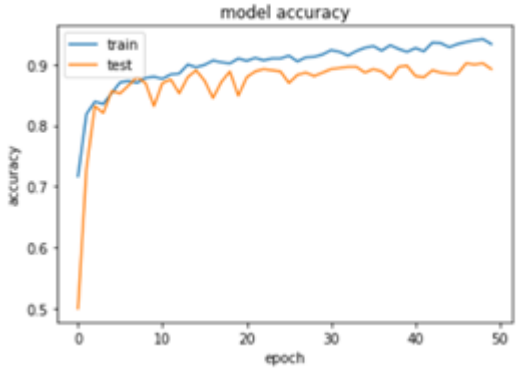


Fig 4.4

MobileNet:

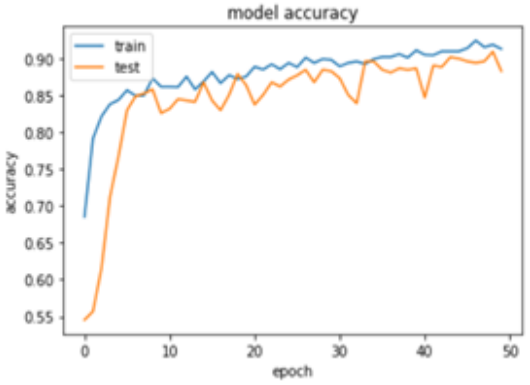


Fig 4.5

NasNet:

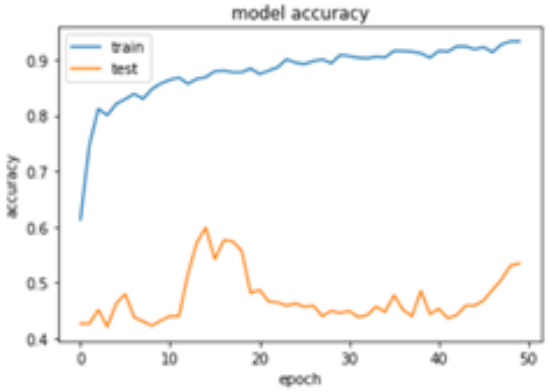


Fig 4.6

Comparison of Losses:

ResNet50:

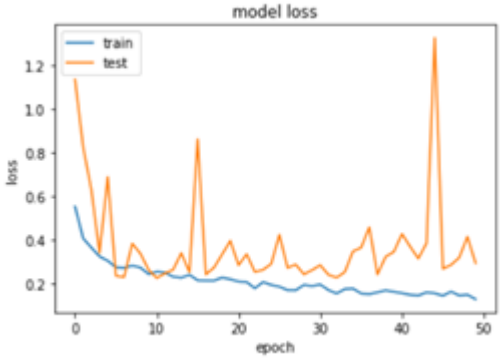


Fig 4.7

DenseNet:

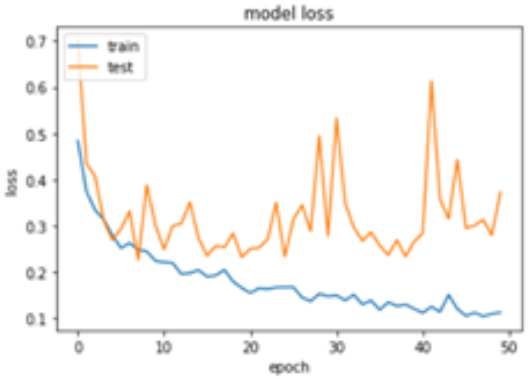


Fig 4.8

InceptionV3:

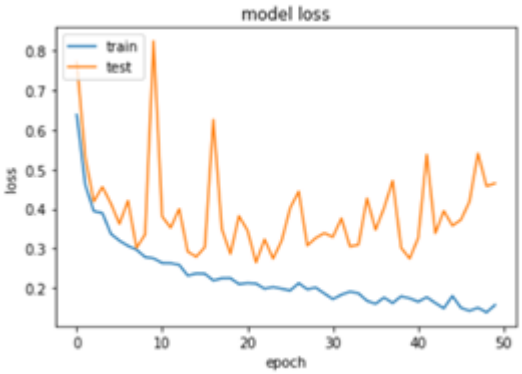


Fig 4.9

MobileNet:

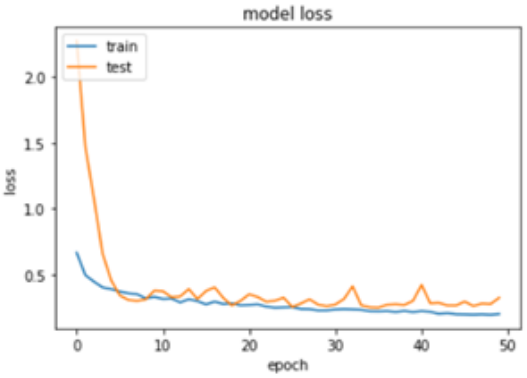


Fig 4.10

NasNet:

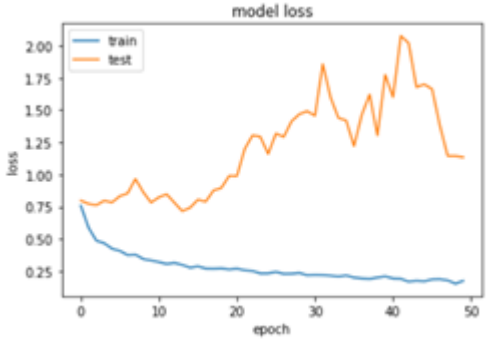


Fig 4.11

We found that ResNet50 had achieved the best accuracy among the 5 models that were experimented.

ResNet achieved accuracy of 92.3 %, followed by DenseNet which acquired accuracy of 90.5% and InceptionV3 had accuracy of 90.1%.
The NasNet model was the least accurate with just 57.9% accuracy.

Confusion Matrix of ResNet50 model:

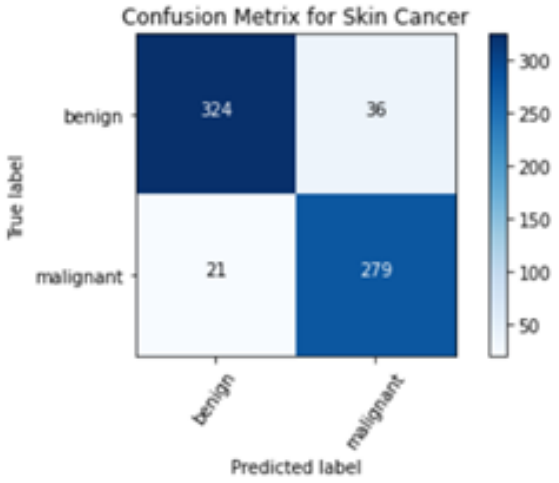


Fig 4.14

ROC-AUC curve of ResNet50 model:

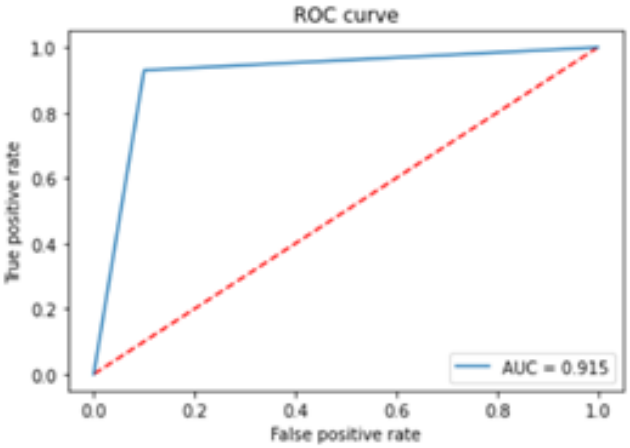


Fig 4.15

Classification Report of ResNet50 model:

	Precision	Recall	f1-score	support
Benign (Class 0)	0.94	0.90	0.92	360
Malignant (Class 1)	0.89	0.93	0.91	300
Accuracy			0.92	660
Macro avg	0.91	0.92	0.91	660
Weighted avg	0.91	0.91	0.91	660

Table 5.1

Chapter 6
Conclusion

In conclusion, this study investigated the ability of deep convolutional neural networks in the classification of benign vs malignant skin cancer. Our results show that state-of-the-art deep learning architectures trained on dermoscopy images (3600 in total composed of 3000 training and 600 validation) outperforms dermatologists. We showed that with use of very deep convolutional neural networks using transfer learning and fine-tuning them on dermoscopy images, better diagnostic accuracy can be achieved compared to expert physicians and clinicians. Although only 1 preprocessing step is applied in this paper, the experimental results are very promising. These models can be easily implemented in dermoscopy systems or even on smartphones in order to assist dermatologists. More diverse datasets (varied categories, different ages) with much more dermoscopy images and balance

References

1. K. E. Purnama et al., "Disease Classification based on Dermoscopic Skin Images Using Convolutional Neural Network in Teledermatology System," 2019 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM), Surabaya, Indonesia, 2019, pp. 1-5, doi: 10.1109/CENIM48368.2019.8973303.

2. T. Shanthi, R.S. Sabeenian, R. Anand, Automatic diagnosis of skin diseases using convolution neural network, Microprocessors and Microsystems, Volume 76, 2020,103074, ISSN 0141-9331, https://doi.org/10.1016/j.micpro.2020.103074.

3. V. B.N., P. J. Shah, V. Shekar, H. R. Vanamala and V. Krishna A., "Detection of Melanoma using Deep Learning Techniques," 2020 International Conference on Computation, Automation and Knowledge Management (ICCAKM), Dubai, United Arab Emirates, 2020, pp. 391-394, doi: 10.1109/ICCAKM46823.2020.9051495.

4. L. -F. Li, X. Wang, W. -J. Hu, N. N. Xiong, Y. -X. Du and B. -S. Li, "Deep Learning in Skin Disease Image Recognition: A Review," in IEEE Access, vol. 8, pp. 208264-208280, 2020, doi: 10.1109/ACCESS.2020.3037258.

5. B. Ahmad, M. Usama, C. Huang, K. Hwang, M. S. Hossain and G. Muhammad, "Discriminative Feature Learning for Skin Disease Classification Using Deep Convolutional Neural Network," in IEEE Access, vol. 8, pp. 39025-39033, 2020, doi: 10.1109/ACCESS.2020.2975198.

6. M. A. Al Bahar, "Skin Lesion Classification Using Convolutional Neural Network With Novel Regularizer," in IEEE Access, vol. 7, pp. 38306-38313, 2019, doi: 10.1109/ACCESS.2019.2906241.