Skin Cancer Detection

Using

Transfer Learning

&

Ensemble Modeling

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

**DECLARATION**

We, hereby, declare that the thesis entitled “Skin Cancer Detection Using Transfer Learning and Ensemble Modeling" submitted by us, for the award of the degree of *Bachelor of Technology in Programme* to VIT is a record of bonafide work carried out by me under the supervision of Dr. Agilandeeswari L.

We further declare that the work reported in this thesis has not been submitted and will not be submitted, either in part or in full, for the award of any other Degree or Diploma in this Institute or any other Institute or University.

Place: Vellore Date:

**Signature of the Candidate**

Acknowledgements

Executive Summary

(Summary of the thesis

One page and not exceeding 200 words Times New Roman, 12)

table of Contents

list of Figures

List of tables

Abbreviations

Symbols and Notations

Table of Contents

[Abstract 5](#_Toc71240771)

[Introduction 5](#_Toc71240772)

[Objective 5](#_Toc71240773)

[Motivation 5](#_Toc71240774)

[Background 6](#_Toc71240775)

[Project Description and Goals 6](#_Toc71240776)

[Technical Specifications 6](#_Toc71240777)

[Design Approach and Details 6](#_Toc71240778)

[Design Approach/Materials and Methods 6](#_Toc71240779)

[Codes and Standards 6](#_Toc71240780)

[Constraints, Alternatives and Tradeoffs 6](#_Toc71240781)

[Proposed Architecture 6](#_Toc71240782)

[DATA SELECTION 6](#_Toc71240783)

[EXPLORATORY DATA ANALYSIS 6](#_Toc71240784)

[CHECKING THE TYPES OF DATA 6](#_Toc71240785)

[FINDING THE OUTLIERS 6](#_Toc71240786)

[DATA VISUALIZATION 7](#_Toc71240787)

[DATA PRE-PROCESSING 7](#_Toc71240788)

[SPLITTING THE DATA 7](#_Toc71240789)

[CHECKING FOR MISSING VALUES 7](#_Toc71240790)

[CHECKING CATEGORICAL FEATURES 7](#_Toc71240791)

[NORMALIZING DATASET 7](#_Toc71240792)

[FEATURE TRANSFORMATION 7](#_Toc71240793)

[WHY DO WE NEED FEATURE TRANSFORMATION AND SCALING? 7](#_Toc71240794)

[FEATURE TRANSFORMATIONS USED IN THE MODELS 7](#_Toc71240795)

[MODEL SELECTION 7](#_Toc71240796)

[MODEL TRAINING 7](#_Toc71240797)

[MODEL EVALUATION 7](#_Toc71240798)

[Schedule, Tasks and Milestones 7](#_Toc71240799)

[Project Demonstration 8](#_Toc71240800)

[Cost Analysis, Result and discussion 8](#_Toc71240801)

[Summary 8](#_Toc71240802)

[References 8](#_Toc71240803)

[Appendix 8](#_Toc71240804)

[Image acquisition 8](#_Toc71240805)

[Image Pre-processing: 8](#_Toc71240806)

[Feature Extraction 8](#_Toc71240807)

[Classification: 9](#_Toc71240808)

[Network Architecture 9](#_Toc71240809)

[Pre-processing 9](#_Toc71240810)

[Architecture and Process Flow Model for the Proposed Model 10](#_Toc71240811)

[Methodology and System Design 12](#_Toc71240812)

[Image Processing 13](#_Toc71240813)

[Convolutional Neural Network 13](#_Toc71240814)

[Proposed Simple Process Flow Diagram 13](#_Toc71240815)

Skin Cancer Detection using Transfer Learning and Ensemble Modeling

Proposed Architecture

# 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 analyze 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.

# Introduction

## Objective

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 analyze 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.

## Motivation

Skin cancer is an alarming disease for mankind. The necessity of early diagnosis of the skin cancer have been increased because of the rapid growth rate of Melanoma skin cancer, its high treatment costs, and death rate. This cancer cells are detected manually and it takes time to cure in most of the cases.

The features of the affected skin cells are extracted after the segmentation of the dermoscopic images using feature extraction technique.

## Background

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.

# Project Description and Goals

As mentioned before, 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.

# Technical Specifications

**INTERFACE REQUIREMENTS**

There are two interfaces namely – User Interface and Software Interface.

* User Interface:

The user interface will be implemented using any desktop running on Windows OS. This interface will be very user friendly so that people from different strata can use it to detect their disease without any difficulty by just uploading their medical test image.

* Software Interface:

A software interface running on Windows OS. It should have Python compiler.

# Design Approach and Details

## Design Approach/Materials and Methods

## Codes and Standards

## Constraints, Alternatives and Tradeoffs

# Proposed Architecture

## DATA SELECTION

## EXPLORATORY DATA ANALYSIS

### CHECKING THE TYPES OF DATA

 there are four types of EDA in all:

* **Univariate non-graphical**: This is the simplest form of data analysis among the four options. In this type of analysis, the data that is being analysed consists of just a single variable. The main purpose of this analysis is to describe the data and to find patterns.
* **Univariate graphical**: Unlike the non-graphical method, the [graphical](https://analyticsindiamag.com/whoever-said-machine-learning-is-all-about-statistics-is-wrong/) method provides the full picture of the data. The three main methods of analysis under this type are histogram, stem and leaf plot, and box plots. The histogram represents the total count of cases for a range of values. Along with the data values, the stem and leaf plot shows the shape of the distribution. The box plots graphically depict a summary of minimum, first quartile median, third quartile, and maximum.
* **Multivariate non-graphical**: The multivariate non-graphical type of EDA generally depicts the relationship between multiple variables of data through cross-tabulation or statistics.
* **Multivariate graphical**: This type of [EDA](https://analyticsindiamag.com/hands-on-guide-to-datatable-library-for-faster-eda/) displays the relationship between two or more set of data. A bar chart, where each group represents a level of one of the variables and each bar within the group represents levels of other variables.

### FINDING THE OUTLIERS

There exist three different options on how to treat non-error outliers:

1. Keep
2. Delete
3. Recode

Keep

When most of the detected outliers are non-error outliers and rightfully belong to the population of interest

Delete

The most straightforward option is to delete any outlying observation. However, this strategy bears a high risk of losing information. Especially if you find many outlying data points, try to avoid this. Also, deleting interesting and influential outliers (points that belong to the population of interest) can falsely impact any output, e.g., prediction or test result, you aim to achieve.

Recode

Recoding outliers is a good option to treat outliers and keep as much information as possible simultaneously. This option should always be accompanied by sound reasoning and explanation.

### DATA VISUALIZATION

* Univariate data analysis
* Bivariate data analysis
* Line plot
* Box plot
* Scatter matrix

## DATA PRE-PROCESSING

### SPLITTING THE DATA

divide the dataset into three parts to avoid overfitting or underfitting and model selection bias called -

1. Training set (Has to be the largest set)
2. Cross-Validation set or Development set or Dev set
3. Testing Set

Underfitting:

Underfitting mainly occurs when a machine learning algorithm is not able to capture the lower trend of data which is mainly when data is nor well fitted inside the model.

overfitting?

When machine learning algorithm is trained on very well data and very closely on a dataset which can lead to a negative impact on the performance of the system leading to the wrong system and prediction model.

### CHECKING FOR MISSING VALUES

there are three categories of missing data:

* **MCAR** (Missing Completely At Random) where the pattern of missingness is statistically independent of the data record. Example: you have a data set on a piece of paper and you spill coffee on the paper destroying part of the data.
* **MAR** (Missing At Random) where the probability distribution of the pattern of missingness is functionally dependent upon the observable component in the record. MCAR is a special case of MAR. Example: if a child does not attend an educational assessment because the child is (genuinely) ill, this might be predictable from other data we have about the child’s health, but it would not be related to what we would have measured had the child not been ill.
* **MNAR** (Missing Not at Random) which is defined as the case which is NOT MAR, or when the missingness is specifically related to what is missing. Example: a person does not attend a drug test because the person took drugs the night before.

### CHECKING CATEGORICAL FEATURES

Two major types of categorical features are

* **Nominal** – These are variables which are not related to each other in any order such as colour (black, blue, green).
* **Ordinal** – These are variables where a certain order can be found between them such as student grades (A, B, C, D, Fail).

### NORMALIZING DATASET

The goal of normalization is to change values to a common scale without distorting the difference between the range of values.

Using this technique, we are going to have a mean of 0 and a standard deviation of 1 in our dataset. We can either do it normally by combining different functions in numpy, i.e.,

z = (x.values-np.mean(x.values))/np.std(x.values)

where *x* is a data frame with all numerical indices. If we want to retain the values in a data frame, then we can simply remove. values in front of it.

## FEATURE TRANSFORMATION

### WHY DO WE NEED FEATURE TRANSFORMATION AND SCALING?

Oftentimes, we have datasets in which different columns have different units – like one column can be in kilograms, while another column can be in centimetres. Furthermore, we can have columns like income which can range from 20,000 to 100,000, and even more; while a benign column which can range from 0 to 100(at the most). Thus, Malignant is about 1,000 times larger than age. When we feed these features to the model as is, there is every chance that the Malignant values will influence the result more due to its larger value. But this doesn’t necessarily mean it is more important as a predictor. So, to give importance to both Benign, and Malignant, we need feature scaling.

### FEATURE TRANSFORMATIONS USED IN THE MODELS

## MODEL SELECTION

## MODEL TRAINING

## MODEL EVALUATION

# Schedule, Tasks and Milestones

# Project Demonstration

# Cost Analysis, Result and discussion

# Summary

# References

<https://www.analyticsvidhya.com/blog/2020/07/types-of-feature-transformation-and-scaling/>

<https://towardsdatascience.com/detecting-and-treating-outliers-in-python-part-3-dcb54abaf7b0>

<https://towardsdatascience.com/data-visualization-for-eda-exploratory-data-analysis-f001a1bf0087>

<https://analyticsindiamag.com/exploratory-data-analysis-functions-types-tools/>

<https://medium.com/mljcunito/the-last-step-in-data-preprocessing-handling-missing-values-ae8f19dae309>

<https://www.kdnuggets.com/2020/07/easy-guide-data-preprocessing-python.html>

# Appendix

## 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.

## 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](https://deepai.org/machine-learning-glossary-and-terms/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

### Pre-processing

#### Data Preprocessing

##### 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.

##### Splitting the Data

The dataset will be split as:

* Training Data: 70%
* Validation Data: 15%
* Testing Data: 15%

#### Image Preprocessing

##### Convert into Array and then into Tensors

The images are converted into arrays because the images will be processed as arrays and the images will be further converted into tensors because the tensors have the following adantages:

* It avoids multi-linear data structure loss
* Computations can be enhanced for classical data processing
* Multi-dimensional data image can be more efficient and effective

After converting, we will also obtain the respective labels for all the images

#### 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. The model consists of an Ensemble Model which consists of the following Pre-trained Models:

* MobileNet
* InceptionV1
* Xception
* Convolutional neural Network

# Architecture and Process Flow Model for the Proposed Model

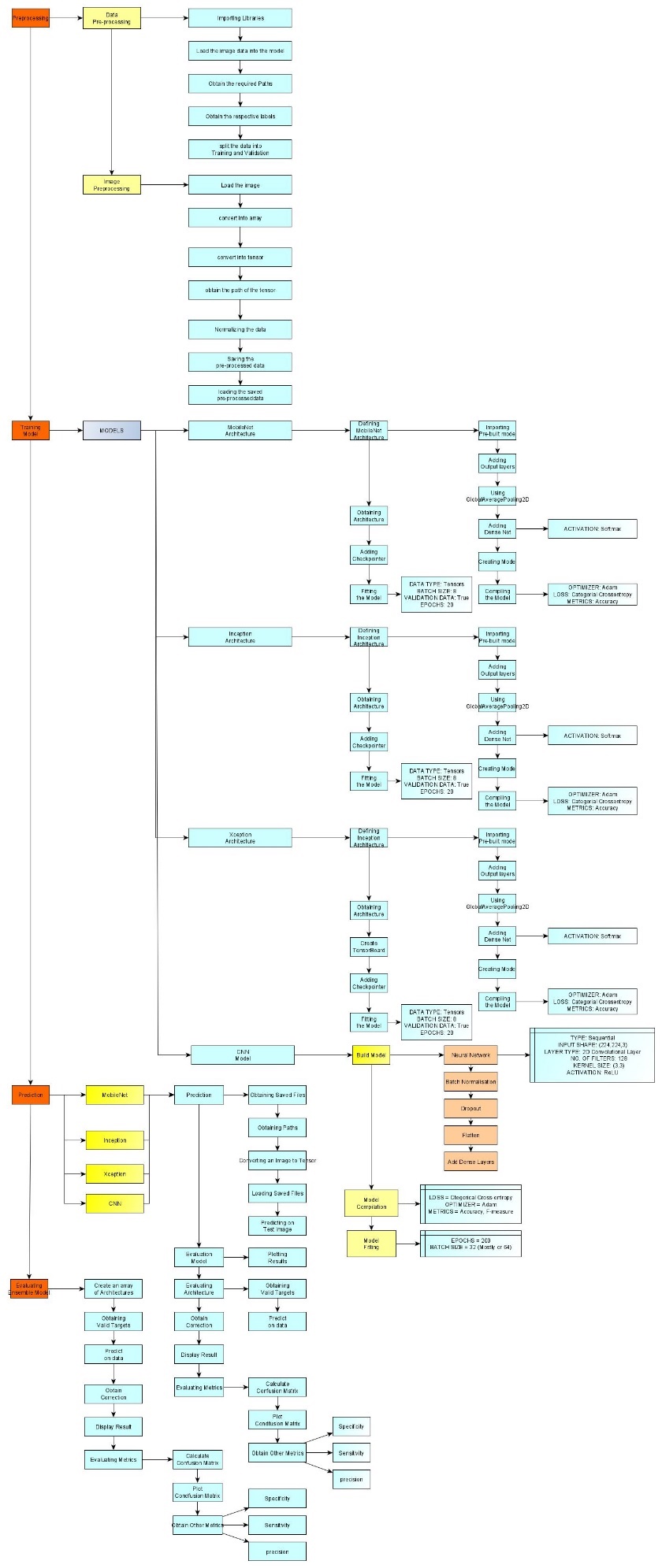


Figure Proposed Architecture

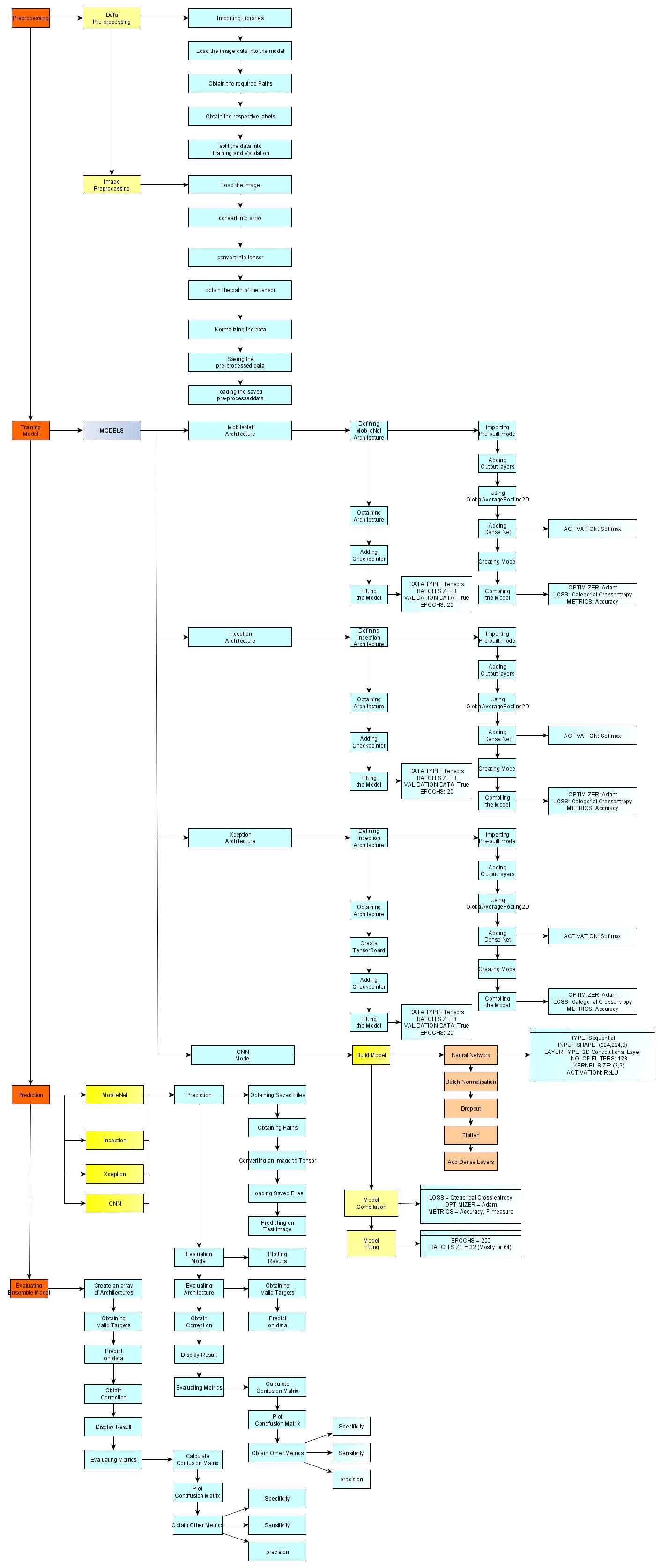


Figure : Proposed Architecture (Lower-half)

# Methodology and System Design

## Image Processing

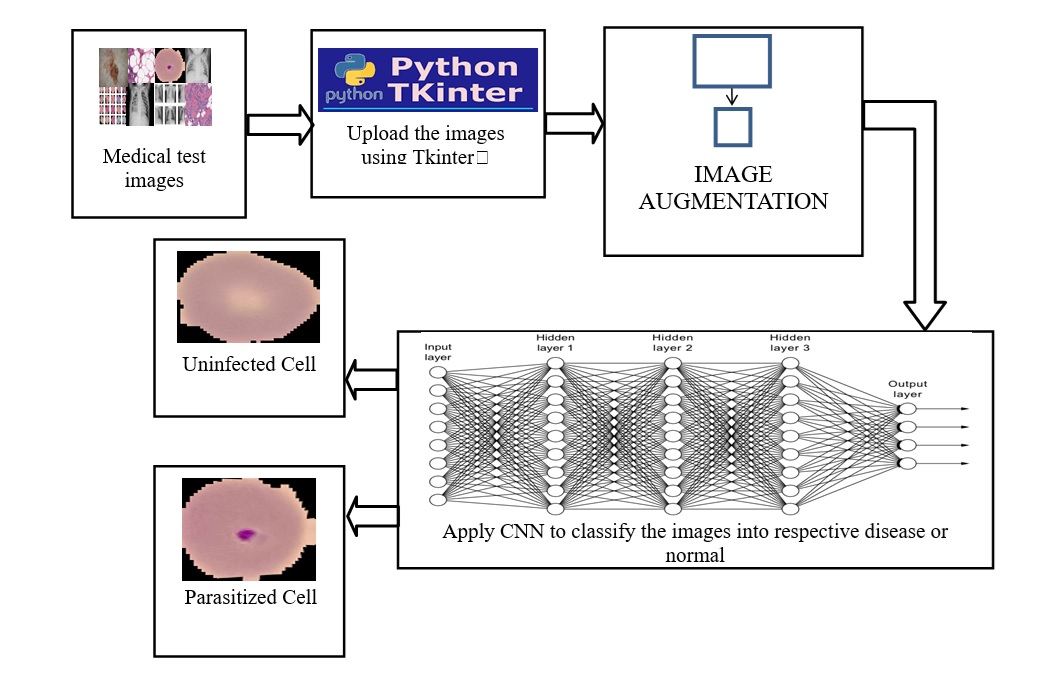
* Image processing can be defined as the technical analysis of an image by using complex algorithms.
* The purpose of early image processing was to improve the quality of the image. Its use has been increasing exponentially in the last decades.
* Its applications range from medicine to entertainment,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

## Proposed Simple 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.