

A project report on

Medical Analysis on Chronic and Brain Tumor Prediction

Submitted in partial fulfillment for the award of the degree of

Integrated MTech Software Engineering

by

Sai Sheshank Gaddam (17MIS7099)



AMARAVATI

MAY, 2020

CERTIFICATE

This is to certify that the Summer Project work titled “Medical Analysis on Chronic and Brain Tumour Prediction” that is being submitted by G. Sai Sheshank (17MIS7099) is in partial fulfillment of the requirements for the award of **Master of Technology (Integrated 5 Year) software engineering**, is a record of bonafide work done under my guidance. The contents of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for award of any degree or diploma and the same is certified.



(Dr. Bksp Kumar Raju)

Guide

The thesis is satisfactory

Approved by



PROGRAM CHAIR

M. Tech. SE



DEAN

School Of Computer Science and Engineering

Date: 13th May 2020

TO WHOMSOEVER IT MAY CONCERN

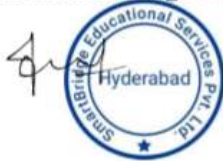
This is to certify that **Mr./Ms. Saisheshank Gaddam** pursuing M.Tech Integrated in Computer Science Engineering from Vellore Institute of Technology, Amaravathi has successfully completed his/her Summer Internship from **01st April 2020 to 13th May 2020**.

During this period he/she had learned the concepts of **Artificial Intelligence with Python & IBM Watson** and successfully completed a project "**Chronic Kidney Disease Analysis**".

Refer the enclosed **Certificate of Merit** for his/her performance during the tenure of Internship.

We wish him/her all the best for his/her future endeavours.

For SmartBridge Educational Services Pvt. Ltd.,



Jayaprakash.ch,
Program Manager

CERTIFICATE BY THE EXTERNAL GUIDE

This is to certify that the project report entitled “**Medical Analysis on Chronic and Brain Tumor Prediction**” submitted by G Sai Sheshank (17MIS7099) to VIT-AP in partial fulfilment of the requirement for the award of the degree of Integrated MTech Software Engineering for the bonafide work carried out by him/her under my guidance. The project fulfil the requirements as per the regulations of this Institute and in my opinion meets the necessary standards for submission. The contents of this report have 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.



(Nagarjuna Madluri)

Signature of the External Supervisor

G. Sai Sheshank

ABSTRACT

The Internship in the The SmartBridge company (In Collaboration with IBM) as Artificial Intelligence Intern. The process of the Internship is that first we will be getting trained on the Python for Data Science and then Deep Learning algorithms, where we have come across how AI will be leading the future and the types of Learnings in ML, Deep Learning algorithms such as ANN, CNN, NLP with which we worked on number of datasets and Pre-process the data later apply suitable AI algorithms to find the best accuracy of the model and predict the application necessary. Later, complete overview of IBM Watson Studio where we will be deploying and Integrate apps through Node-Red for the and then the final project development.

This project is aimed at developing a model which predicts whether a person is affected by kidney disease based on certain parameters, demonstrates a promising capability of reducing the uncertainties in the prediction model. Accurate and reliable disease predictions are necessary to the people in need.

ACKNOWLEDGEMENT

It is my pleasure to express with deep sense of gratitude to Professor BKSP. Kumar Raju, VIT-AP, for his constant guidance, continual encouragement, understanding; more than all, he taught me patience in my Endeavor. My association with him is not confined to academics only, but it is a great opportunity on my part of work with an intellectual and expert in the field of coding.

I would like to express my gratitude Dr. G. Viswanathan, Dr. Sankar Viswanathan, Dr. D. Subhakar, and Prof. Jagadish Mudiganti, Dean of School of Computer Science & Engineering, for providing with an environment to work in and for his inspiration during the tenure of the course.

In jubilant mood I express ingeniously my whole-hearted thanks to Dr. Pradeep Reddy CH, Associate Professor, all teaching staff and members working as limbs of our university for their not-self-centered enthusiasm coupled with timely encouragements showered on me with zeal, which prompted the acquirement of the requisite knowledge to finalize my course study successfully. I would like to thank my parents for their support.

It is indeed a pleasure to thank my friends who persuaded and encouraged me to take up and complete this task. At last but not least, I express my gratitude and appreciation to all those who have helped me directly or indirectly toward the successful completion of this project.

Place: AMARAVATHI

G. Sai Sheshank

Date: 05.07.2020

17MIS7099

CONTENTS

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION	11
1.2 PROBLEM STATEMENT	11
1.3 STUDY OF THE SYSTEM	11
1.4 MODULE DESCRIPTION	12-14
1.5 SCOPE OF THE PROJECT	14

CHAPTER 2

BACK GROUND

2.1 INTRODUCTION.....	3
2.2 DATASET.....	3
2.3 METHODOLOGY	5
2.3.1 IMPORTING LIBRARIES.....	6
2.3.2 INITIALIZE THE MODEL.....	7
2.3.3 ADDING DIFFERENT LAYERS	7
2.3.4 IMAGE PREPOROCESSING	9
2.3.5 IMPORT DATASET.....	10
2.3.6 TRAINING AND EVALUATING THE MODEL	11

2.4 BUILDING USER INTERFACE.....	14
2.4.1 PREDICTION.....	16
 CHAPTER 3	
CODE.....	27-41
 CHAPTER 4	
CONCLUSION AND FUTURE WORK.....	38
 CHAPTER 5	
REFERENCES.....	39

LIST OF FIGURES

Fig1: Kidney with stones.....	15
Fig2: Normal Kidney	15
Fig3: Brain Image.....	16
Fig4: Brain Tumor.....	16
Fig6: Import Libraries.....	17
Fig7: Initialize Model.....	17
Fig8: Applying dense layers.....	18
Fig9: Layers.....	18
Fig10: Compile.....	19
Fig11: Image Pre-processing.....	19
Fig12: Imported Dataset.....	20
Fig13: Fitting model.....	21
Fig14: Flask.....	22
Fig15: Prediction.....	23
Fig16: Using Open cv Prediction.....	24
Fig17: Prediction of no tumor in brain.....	25
Fig18: Prediction of brain tumor.....	25

LIST OF TABLES

Fig6: Flowchart.....	16
----------------------	----

CHAPTER-1

1.1 INTRODUCTION

A global health problem which is steadily growing is Chronic kidney disease (CKD), Brain tumor. It is a chronic condition associated with increased morbidity and mortality, a high risk of many other diseases including cardiovascular disease, and high health care costs, kidney stones, brain disease.

1.2 PROBLEM STATEMENT

- There is significant importance in the early detection, controlling, and managing of the disease. It is necessary to predict the progression of CKD with reasonable accuracy because of its dynamic and covert nature in the early stages, and patient heterogeneity. CKD is often described by severity stages. Similarly, there is an urgency to early tumor in the brain.
- Clinical decisions are influenced by the stage, whether a patient is progressing, and the rate of progression. Also, defining the disease stage is quite crucial as it gives several indications that support the determination of required intervention and treatments.

1.3 STUDY OF THE SYSTEM

Over two million people worldwide receive dialysis or kidney transplant treatment to stay alive, yet this number may represent only 10% of people who need treatment to live. The majority of the 2 million people who receive treatment for kidney failure are in only five relatively wealthy countries, which represent 12% of the global population. By comparison, only 20% of the world's population is treated in about 100 developing countries, and they represent almost half the global population. Annually, more than one million people in 112 lower-income countries die from untreated kidney failure, due to the huge financial burden of dialysis or kidney transplantation treatment.

1.4 MODULE DESCRIPTION

The system after careful analysis has been identified to be presented with the following modules:

MODULES INVOLVED

- Adding Convolution, Max Pooling, Flatten
- Adding dense layers
- Loading of data
- Dataset Pre-Processing
- Model Creation
- Integrating through Flask
- Designing UI

ADDING CONVOLUTION, MAX POOLING, FLATTEN LAYERS

- A convolutional layer that extracts features from a source image. Convolution helps with blurring, sharpening, edge detection, noise reduction, or other operations that can help the machine to learn specific characteristics of an image.
- A pooling layer that reduces the image dimensionality without losing important features or patterns.
- A fully connected layer also known as the dense layer, in which the results of the convolutional layers are fed through one or more neural layers to generate a prediction.

ADDING DENSE LAYERS

Dense Layer = Fullyconnected Layer = topology, describes how the neurons are connected to the next layer of neurons (every neuron is connected to every neuron in the next layer), an intermediate layer (also called hidden layer see figure)

Output Layer = Last layer of a Multilayer Perceptron.

LOADING OF DATA

The given data set will be loaded into the Jupyter notebook for data analyzation and further processes will be made to predict the correct output in the note book. Later, the data given is pre-processed and going to predict the output. Finally, this will we generating a model using CNN and integrated the model to flask using python where it interacts with HTML and CSS to make the final web App.

DATASET PRE-PROCESSING

The raw data set will be chosen to us which contains images with chronic disease and normal kidney images. Later we need to clean the data by using the commands. Where we need to rescale the images to 0-255. The given data consists of various features few kidneys with stones and few with some other diseases. Since, human can able to see the data with his naked eye but machine understands only using the matrix entry of pixels. By this machine understand what type of images we have taken and what are the two different categorical values we get.

MODEL CREATION

Once the pre-processed data is classified and trained / classified with CNN, there will be an accuracy check based on certain parameters like number of nodes in the input layer, number nodes in the hidden layer, the output node, activation parameter for both input and output layers, optimizer used and all. Finally, after the model we got the most accurate then it saved through .H5 file so as to integrate with web page using Flask.

INTEGRATION THROUGH FLASK

After the model is saved through .H5 file, we need to create a flask file in python where the .H5 file model is integrated here in Flask so as to connect with the web files to produce the final UI and get the correct prediction same as the accuracy we got in the model saving.

DESIGNING UI

The UI we created is nothing but the simple HTML file which includes the background colour, using this button we are able to predict whether a person is facing from the chronic kidney disease or a person with healthy.

1.5 SCOPE OF THE PROJECT

AI can plays a major role in extracting hidden data from the large patient medical and clinical dataset with images that physicians frequently collect from patients to obtain insights about the diagnostic information, and to implement precise treatment plans. Artificial Intelligence can be defined as the process of extracting hidden data from a large dataset. AI techniques are applied and used widely in various contexts and fields. With CNN technique we could predict, classify, filter and predict data.

Pros:

With the best accuracy we predict the Kidney Disease. You don't have to go to doctor for the confirmation. After you confirm with this prediction you can go to the doctor and get the medication from them. Instead if you don't have the disease then you don't have to go to doctor all through the way. Doctors can also recheck the patients and their analysis through this model.

Cons:

Few people do negligence having our model. Because they think that they are having a predictor so we can check at some time and we don't have to go to doctor. So, people through their negligence they might get harm themselves. This might cause danger to their life.

CHAPTER-2

2.1 INTRODUCTION

The main initiate of this project is to reduce human work. To develop this type of project the person should have well knowledge of python using deep learning techniques mainly convolution neural networks because the approach used for our project is CNN. We have taken two categories one with chronic disease and the other with normal kidney images. Similarly, we have two different types of images for brain tumor one with brain tumor and other with normal brain images

2.2 DATASET

DATASET OF KIDNEY CHRONIC DISEASE

Images of different resolutions and sizes were obtained from many resources, including photos taken by CT scan, a MRI scans obtained from the internet. These images were rescaled into 255×255 to increase the size of our dataset producing a total number of 100 images in our dataset. The data was labelled into two main categories: one is chronic dataset ckd, and non-chronic disease (no-ckd).



Fig1: Kidney with stones



Fig2: Normal Kidney

DATASET OF BRAIN TUMOR:

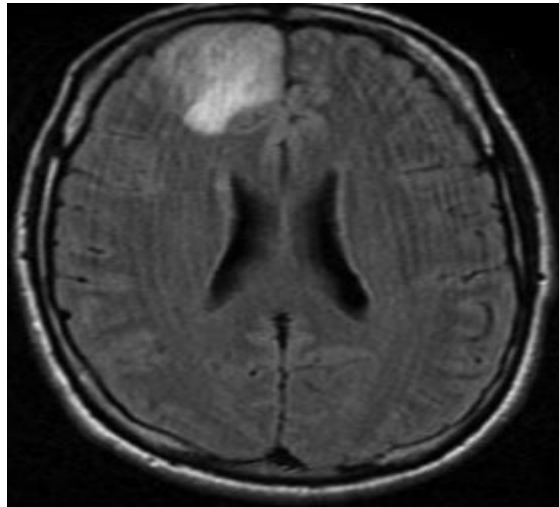


Fig3: Brain Image

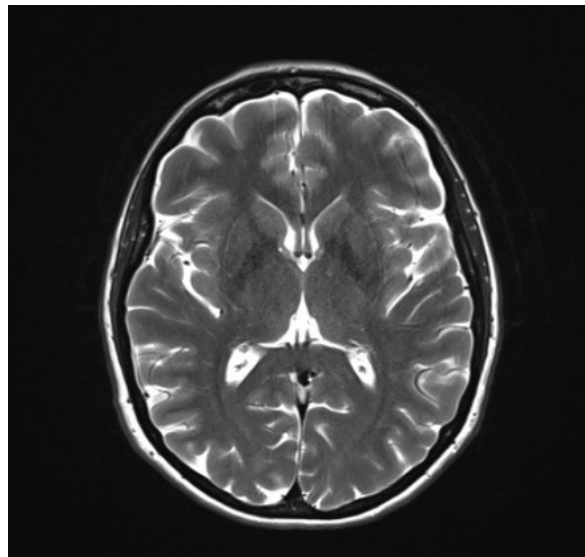


Fig4: Brain Tumor

2.3 METHODOLOGY

Here we have used a Deep Learning technique – Convolutional Neural Network (CNN). We used this technique because it makes the task effectively and also it plays a core part in a regular Artificial Neural Network (ANN) with pre-processing. CNN is combination of Convolutional Layers and Neural Network. Basically, any Neural Network which is used for image processing, consist of following layer - Convolutional Layer, Pooling Layer, Flatten Layer, Dense Layer. The code for both the brain tumor, chronic disease is

same. So, the following code explanation is for chronic disease which is same as Brain tumor.

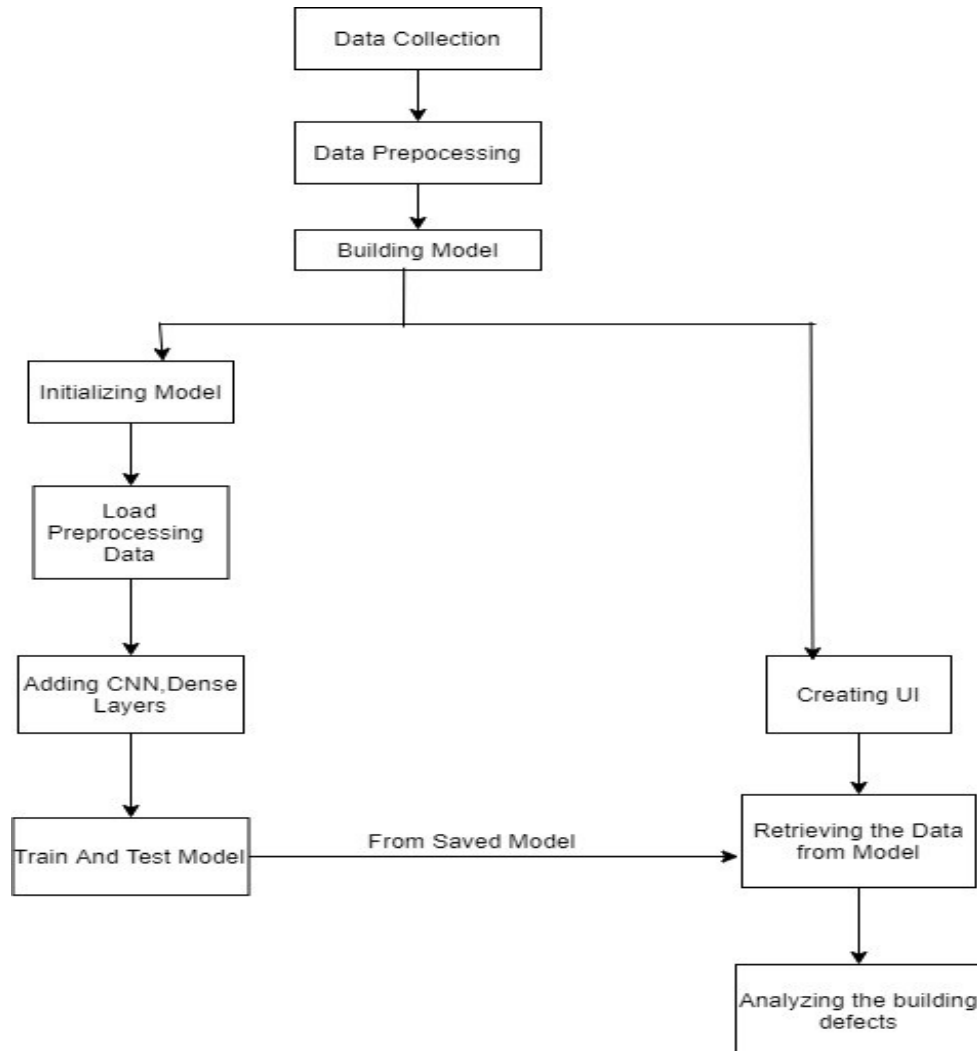


Fig5: Flowchart

2.3.1 IMPORT LIBRARIES

Importing the required libraries for our model. Where we are going to add different layers further like Dense, Convolution, Max pooling, Flatten. We will be defining our deep learning neural network using Keras packages. We import the Sequential, Dense, Dropout and Activation packages for defining the network architecture. We use load_model package for saving and retrieving our model.

```
In [1]: from keras.models import Sequential
        from keras.layers import Dense
        from keras.layers import Convolution2D
        from keras.layers import MaxPooling2D
        from keras.layers import Flatten

Using TensorFlow backend.
C:\Users\vyshn\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:516: FutureWarning: Passing (type, 1) or '1
type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
_np_qint8 = np.dtype [("qint8", np.int8, 1)]
C:\Users\vyshn\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:517: FutureWarning: Passing (type, 1) or '1
type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
_np_quint8 = np.dtype [("quint8", np.uint8, 1)]
C:\Users\vyshn\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:518: FutureWarning: Passing (type, 1) or '1
type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
_np_qint16 = np.dtype [("qint16", np.int16, 1)]
C:\Users\vyshn\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:519: FutureWarning: Passing (type, 1) or '1
type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
_np_quint16 = np.dtype [("quint16", np.uint16, 1)]
C:\Users\vyshn\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:520: FutureWarning: Passing (type, 1) or '1
type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
_np_qint32 = np.dtype [("qint32", np.int32, 1)]
C:\Users\vyshn\anaconda3\lib\site-packages\tensorflow\python\framework\dtypes.py:525: FutureWarning: Passing (type, 1) or '1
type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) / '(1,)type'.
_np_resource = np.dtype [("resource", np.ubyte, 1)]
C:\Users\vyshn\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:541: FutureWarning: Passing (type,
1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) /
'(1,)type'.
_np_qint8 = np.dtype [("qint8", np.int8, 1)]
C:\Users\vyshn\anaconda3\lib\site-packages\tensorboard\compat\tensorflow_stub\dtypes.py:542: FutureWarning: Passing (type,
1) or '1type' as a synonym of type is deprecated; in a future version of numpy, it will be understood as (type, (1,)) /
'(1,)type'.
_np_quint8 = np.dtype [("quint8", np.uint8, 1)]
```

Fig6: Import Libraries

2.3.2 INITIALIZE THE MODEL

There are two ways to build Keras models: *sequential* and *functional*. Here we are using sequential for our model. The sequential API allows you to create models layer-by-layer for most problems. It is limited in that it does not allow you to create models that share layers or have multiple inputs or outputs.

```
In [2]: model = Sequential()

WARNING:tensorflow:From C:\Users\vyshn\anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:68: The name tf.get_d
efault_graph is deprecated. Please use tf.compat.v1.get_default_graph instead.
```

Fig7: Initialize Model

2.3.3 ADDING DIFFERENT LAYERS

```
In [3]: model.add(Convolution2D(32,(3,3), input_shape = (64,64,3), activation = 'relu'))

WARNING:tensorflow:From C:\Users\vyshn\anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:504: The name tf.placeholder is deprecated. Please use tf.compat.v1.placeholder instead.

WARNING:tensorflow:From C:\Users\vyshn\anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:3828: The name tf.random_uniform is deprecated. Please use tf.random.uniform instead.

In [4]: model.add(MaxPooling2D(pool_size = (2,2)))

WARNING:tensorflow:From C:\Users\vyshn\anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:3652: The name tf.nn.max_pool is deprecated. Please use tf.nn.max_pool2d instead.

In [5]: model.add(Flatten())
```

Fig8: Convolution, Maxpooling, Flatten Layers

Arguments:

- filters: Denotes the number of Feature detectors.
- kernel_size: Denotes the shape of the feature detector. (3,3) denotes a 3 x 3 matrix.
- input_shape: standardises the size of the input image
- activation: Activation function to break the linearity
- pool_size: the shape of the pooling window.

```
In [6]: model.add(Dense(units = 128, init = 'uniform', activation = 'relu'))

C:\Users\vyshn\anaconda3\lib\site-packages\ipykernel_launcher.py:1: UserWarning: Update your `Dense` call to the Keras 2 AP
I: `Dense(units=128, activation="relu", kernel_initializer="uniform")`
    """Entry point for launching an IPython kernel.

In [7]: model.add(Dense(units = 128, init = 'uniform', activation = 'relu'))

C:\Users\vyshn\anaconda3\lib\site-packages\ipykernel_launcher.py:1: UserWarning: Update your `Dense` call to the Keras 2 AP
I: `Dense(units=128, activation="relu", kernel_initializer="uniform")`
    """Entry point for launching an IPython kernel.

In [8]: model.add(Dense(output_dim = 3, activation = 'softmax', init = 'uniform'))

C:\Users\vyshn\anaconda3\lib\site-packages\ipykernel_launcher.py:1: UserWarning: Update your `Dense` call to the Keras 2 AP
I: `Dense(activation="softmax", units=3, kernel_initializer="uniform")`
    """Entry point for launching an IPython kernel.
```

Fig9: Applying dense layers

Softmax Activation: The softmax activation function is used in neural networks when we want to build a multi-class classifier which solves the problem of assigning an instance to one class when the number of possible classes is more than two.

ReLU Activation: ReLU stands for Rectified Linear Unit. The main advantage of using this function over other activation functions is that it does not activate all the neurons at the same

time. If we don't know which activation function should be used then, using ReLU is a good choice.

```
entry point for launching an Ipython kernel.

In [9]: model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])

WARNING:tensorflow:From C:\Users\vyshn\anaconda3\lib\site-packages\keras\optimizers.py:744: The name tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.

WARNING:tensorflow:From C:\Users\vyshn\anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:3005: The name tf.log is deprecated. Please use tf.math.log instead.
```

Fig10: Compile

Optimizer: Adam is used to provide an optimization algorithm that can handle on noisy problems. This also increases the accuracy.

Loss: We have two different types of cross entropy one is binary and the other is categorical. Binary **cross-entropy** is for binary classification, whereas categorical cross entropy is for multi-class classification.

Metrics: Metrics is a function just to know the performance of the model.

2.3.4 IMAGE PREPROCESSING

```
from keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(rescale = 1./255, shear_range = 0.2, zoom_range = 0.2, horizontal_flip = True)
test_datagen = ImageDataGenerator(rescale = 1./255)
```

Fig11: Image Preprocessing

Importing the required Libraries and then changing the images to one particular dimension

Arguments:

- rescale: Rescaling factor. Defaults to None. If None or 0, no rescaling is applied, otherwise we multiply the data by the value provided
- shear_range: Shear Intensity. Shear angle in a counter-clockwise direction in degrees.
- zoom_range: Range for random zooming of the image.

2.3.5 IMPORTING THE DATASET

Importing the dataset from our destination which contains three different classes of images contains building defects – spall, flakes, cracks where we store them in three different folders combine to one folder.

```
In [8]: 1 from keras.preprocessing.image import ImageDataGenerator
        2 train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
        3 test_datagen=ImageDataGenerator(rescale=1./255)

In [9]: 1 x_train = train_datagen.flow_from_directory(r'C:\Users\Sai\Desktop\CKD Analysis\train',target_size = (64,64),batch_size = 32
        2 x_test = test_datagen.flow_from_directory(r'C:\Users\Sai\Desktop\CKD Analysis\test',target_size = (64,64),batch_size = 32,cl
        3

Found 155 images belonging to 2 classes.
Found 65 images belonging to 2 classes.

In [10]: 1 print(x_train.class_indices)

{'ckd': 0, 'no ckd': 1}
```

Fig12: Imported Dataset

Arguments:

- directory: Location of the x_train or x_test
- target_size: The dimensions to which all images found will be resized. Same as input size.
- Batch_size: Size of the batches of data (default: 32).
- Class_mode: Determines the type of label arrays that are returned. One of “categorical”, “binary”, “sparse”, “input”, or None.

2.3.6 TRAINING AND EVALUATING THE MODEL

Here we are training the model by fit_generator and later going to save the model. The arguments are as follows

- generator: A generator sequence used to train the neural network (x_train).
- steps_per_epochs: Total number of steps (batches of samples) to yield from generator before declaring one epoch finished and starting the next epoch. It should typically be equal to the number of samples of your dataset divided by the batch size.
- epochs: Total number of epochs. One complete cycle of predictions of a neural network is called an epoch.

- `Validation_data`: A generator sequence used to test and evaluate the predictions of the neural network (`x_test`).
- `Validation_steps`: Total number of steps (batches of samples) to yield from `validation_data` generator before stopping at the end of every epoch.

```
In [12]: 1 model.fit_generator(x_train,steps_per_epoch=5,epochs=100,validation_data=x_test,validation_steps=2)

WARNING:tensorflow:From C:\Users\Sai\anaconda3\lib\site-packages\tensorflow\python\ops\math_grad.py:1250: add_dispatch_support
t.<locals>.wrapper (from tensorflow.python.ops.array_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
WARNING:tensorflow:From C:\Users\Sai\anaconda3\lib\site-packages\keras\backend\tensorflow_backend.py:986: The name tf.assign_
add is deprecated. Please use tf.compat.v1.assign_add instead.

Epoch 1/100
5/5 [=====] - 10s 2s/step - loss: 1.4023 - acc: 0.4118 - val_loss: 0.9732 - val_acc: 0.4531
Epoch 2/100
5/5 [=====] - 3s 642ms/step - loss: 0.7344 - acc: 0.5800 - val_loss: 0.5673 - val_acc: 0.7273
Epoch 3/100
5/5 [=====] - 4s 809ms/step - loss: 0.6270 - acc: 0.7240 - val_loss: 0.5425 - val_acc: 0.7576
Epoch 4/100
5/5 [=====] - 3s 569ms/step - loss: 0.5442 - acc: 0.8274 - val_loss: 0.5040 - val_acc: 0.7812
Epoch 5/100
5/5 [=====] - 4s 797ms/step - loss: 0.4793 - acc: 0.8705 - val_loss: 0.4325 - val_acc: 0.7879
Epoch 6/100
5/5 [=====] - 3s 563ms/step - loss: 0.4334 - acc: 0.8391 - val_loss: 0.4005 - val_acc: 0.8485
Epoch 7/100
```

Fig13: Fitting model

2.4 BUILDING USER INTERFACE FOR THE MODEL:

Flask is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex applications. It began as a simple wrapper around Werkzeug and Jinja and has become one of the most popular Python web application frameworks.

It is an API of Python that allows us to build up web-applications. It was developed by Armin Ronacher. Flask's framework is more explicit than Django's framework and is also easier to learn because it has less base code to implement a simple web-Application.


```
opencvCKD.py X app.py X base.html X
1  from __future__ import division, print_function
2  # coding=utf-8
3  import os
4  import numpy as np
5  from keras.preprocessing import image
6
7  from keras.models import load_model
8  import tensorflow as tf
9  global graph
10 graph=tf.get_default_graph()
11
12 # Flask utils
13 from flask import Flask, request, render_template
14 from werkzeug.utils import secure_filename
15
16 app = Flask(__name__)
17 model = load_model('kidney.h5')
18 print('Model Loaded. Check http://127.0.0.1:5000/')
19
20 @app.route('/', methods=['GET'])
21 def index():
22     # Main page
23     return render_template('base.html')
24
25 @app.route('/predict', methods=['GET', 'POST'])
26 def upload():
27     if request.method == 'POST':
28         # Get the file from post request
29         f = request.files['image']
30
31         # Save the file to ./uploads
32         basepath = os.path.dirname(__file__)
33         file_path = os.path.join(
34             basepath, 'uploads', secure_filename(f.filename))
35         f.save(file_path)
36         img = image.load_img(file_path, target_size=(64, 64))
37         x = image.img_to_array(img)
38         x = np.expand_dims(x, axis=0)
39
40         with graph.as_default():
41             preds = model.predict_classes(x)
42             index = ['ckd', 'no ckd']
43             text = "prediction : "+index[preds[0]]
44
45             # ImageNet Decode
46             return text
47
48
49 if __name__ == '__main__':
50     app.run(debug=True, threaded = False)
51
52
53
```

Fig14: Flask

2.5 PREDICTION(UI):

PREDICTION OF CHRONIC KIDNEY DISEASE:

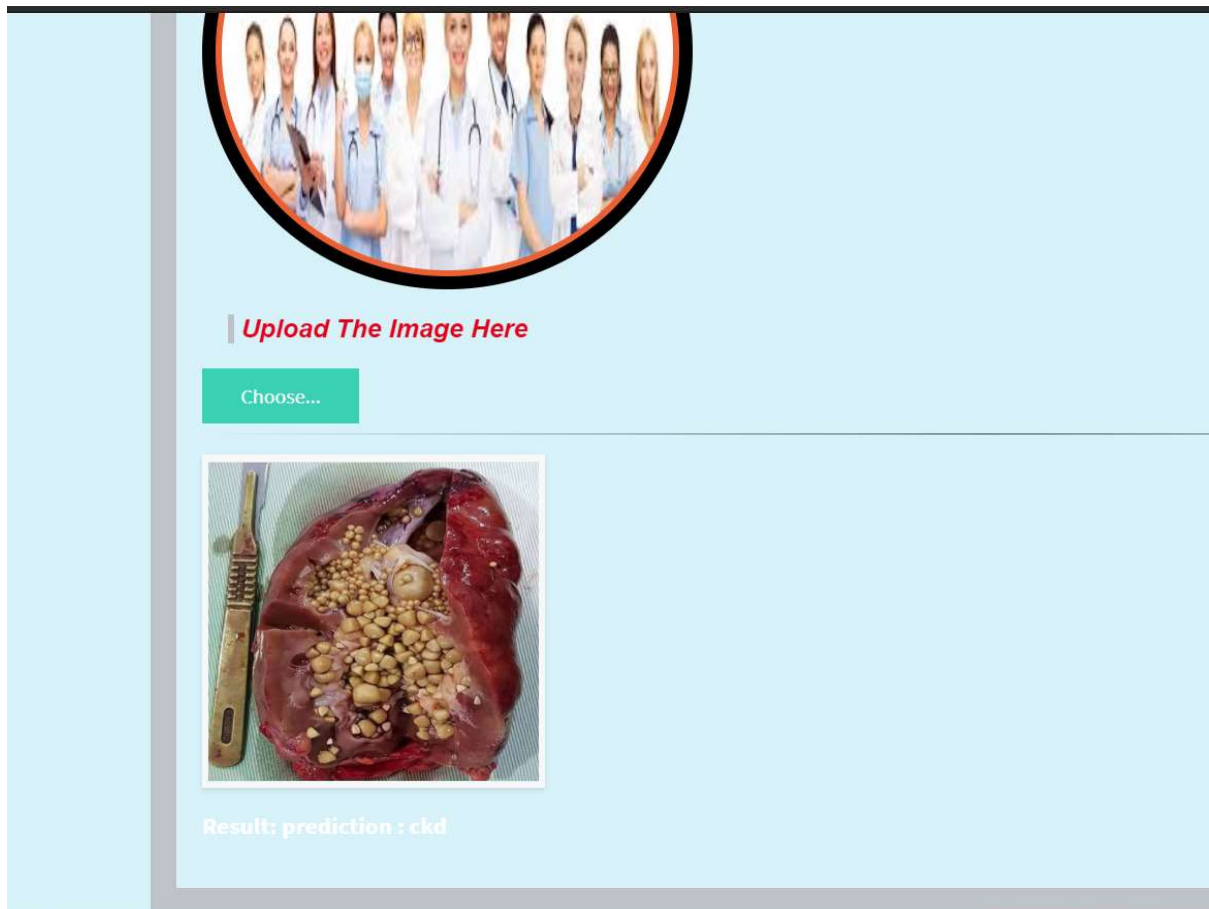


Fig15: Prediction

The predicted value is displayed beside.




Fig16: Using OpenCV Prediction

PREDICTION OF BRAIN TUMOR:

Brain Tumor prediction using CNN

Brain Tumor Prediction :

Brain tumor identification is really challenging task in the early stages of life. But now it became advanced with various machine learning algorithms. Now a day's issue of brain tumor automatic identification is of great interest. A tumor is the unusual growth of the tissues. A brain tumor is a number of unnecessary cells growing in the brain or central spine canal. It is the unrestrained progress of cancer cells in any portion of the body. In Order to detect the brain tumor of a patient, we consider the data of patients like MRI images of a patient's brain. Here our problem is to identify whether the tumor is present in the patient's brain or not. It is very important to detect the tumors at the starting level for a healthy life of a patient.



Upload Image Here

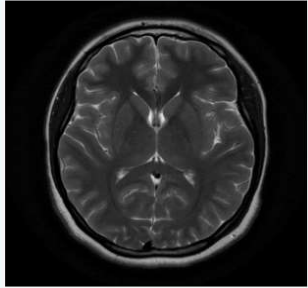


Result: Prediction : no

Fig17: Prediction of no tumor in brain

Brain Tumor prediction using CNN

Brain Tumor Prediction :

Brain tumor identification is really challenging task in the early stages of life. But now it became advanced with various machine learning algorithms. Now a day's issue of brain tumor automatic identification is of great interest. A tumor is the unusual growth of the tissues. A brain tumor is a number of unnecessary cells growing in the brain or central spine canal. It is the unrestrained progress of cancer cells in any portion of the body. In Order to detect the brain tumor of a patient, we consider the data of patients like MRI images of a patient's brain. Here our problem is to identify whether the tumor is present in the patient's brain or not. It is very important to detect the tumors at the starting level for a healthy life of a patient.



Upload Image Here

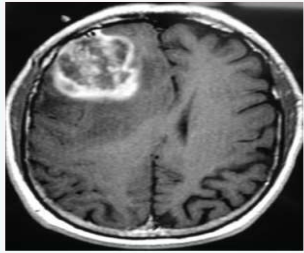

Result: Prediction : yes

Fig18: Prediction of brain tumor

CHAPTER-3

CODE

CNN Model (IPNBY):

```
#importing the libraries

from keras.models import Sequential

from keras.layers import Dense

from keras.layers import Convolution2D

from keras.layers import MaxPooling2D

from keras.layers import Flatten

from keras.models import load_model

import numpy as np

import cv2

from skimage.transform import resize

model = Sequential()

model.add(Convolution2D(32,(3,3),input_shape=(64,64,3),activation='relu'))

model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Dense(output_dim=128,init='uniform',activation='relu'))

model.add(Dense(output_dim=2,activation='softmax',init='uniform'))

from keras.preprocessing.image import ImageDataGenerator

train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)

test_datagen=ImageDataGenerator(rescale=1./255)

x_train = train_datagen.flow_from_directory(r'C:\Users\Sai\Desktop\CKD
Analysis\train',target_size = (64,64),batch_size = 32,class_mode = 'categorical')

x_test = test_datagen.flow_from_directory(r'C:\Users\Sai\Desktop\CKD
Analysis\test',target_size = (64,64),batch_size = 32,class_mode = 'categorical')
```

```

print(x_train.class_indices)

model.compile(loss = 'categorical_crossentropy',optimizer = "adam",metrics = ["accuracy"])

model.fit_generator(x_train,steps_per_epoch=5,epochs=100,validation_data=x_test,validation_steps=2)

model.save("kidney.h5")

from keras.models import load_model

import numpy as np

import cv2

model =load_model('combustion.h5')

model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])

from skimage.transform import resize

def detect(frame):

    try:

        img= resize(frame,(64,64))

        img = np.expand_dims(img,axis=0)

        if(np.max(img)>1):

            img =img/255.0

        prediction =model.predict(img)

        print (prediction)

        prediction_class = model.predict_classes(img)

        print(prediction_class)

        return prediction_class

    except AttributeError:

        print("shape not found")

frame= cv2.imread(r'C:\Users\Sai\Desktop\AI intern proj1\withfire.jpg')

data= detect(frame)

data

```

```

from playsound import playsound

if(data[0]==0):

    playsound(r'C:\Users\Sai\Desktop\AI intern proj1\Tornado_Siren.mp3')

#elif(data[0]==1):

    # playsound(r'C:\Users\DELL\Downloads\Annoying_Alarm_Clock-UncleKornicob-1.mp3')

```

Flask File (PY):

```

from __future__ import division, print_function

# coding=utf-8

import os

import numpy as np

from keras.preprocessing import image

from keras.models import load_model

import tensorflow as tf

global graph

graph=tf.get_default_graph()

# Flask utils

from flask import Flask, request, render_template

from werkzeug.utils import secure_filename

app = Flask(__name__)

model = load_model('kidney.h5')

```

```

print('Model loaded. Check http://127.0.0.1:5000/')

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

def index():

    # Main page

    return render_template('base.html')

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

def upload():

    if request.method == 'POST':

        # Get the file from post request

        f = request.files['image']

        # 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)

        img = image.load_img(file_path, target_size=(64, 64))

        x = image.img_to_array(img)

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

```

```

with graph.as_default():

    preds = model.predict_classes(x)

    index = ['ckd','no ckd']

    text = "prediction : "+index[preds[0]]

    # ImageNet Decode

    return text

if __name__ == '__main__':

    app.run(debug=True,threaded = False)

```

base.html

```

<!DOCTYPE html>

<html lang="en">

<body>

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <meta http-equiv="X-UA-Compatible" content="ie=edge">

    <title>Forest Combustion Recognition Using AI</title>

    <script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>

    <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>

```

```
<script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
```

```
<link href="{{ url_for('static', filename='css/main.css') }}" rel="stylesheet">
```

```
<link  
href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:400,700&display=swap"  
rel="stylesheet" >
```

```
<title>Chronic Kidney Disease Using AI CNN</title>
```

```
<style>
```

```
div.imageborder {
```

```
border-radius: 999em;
```

```
width: 350px;
```

```
height: 350px;
```

```
padding: 5px;
```

```
line-height: 0;
```

```
border: 10px solid #000;
```

```
background-color:#eb6134;
```

```
}
```

```
img {
```

```
border-radius:999em;
```

```
height: 100%;
```

```
width: 100%;
```

```

margin: 0;

}

body{

border : 20px solid #bdc3c7;

padding: 20px;

max-width:10000px;

width:80%;

margin: 20px auto;

font-family: source sans pro;

background-image:
url("https://i.pinimg.com/originals/01/da/f3/01daf317b1b314d85b9c5556ed39018b.png");

}

}

h2{

color: #2c3e50;

margin:1.66%;

font-size: 2.0rem


}

.one{

border-left: 5px solid #bdc3c7;

```



```

    font-size: 25px;

    <!--color: #ffbf00;-->

    color: #000000;

    font-family: Aharoni;

    padding-left: 5px;
}

.date{

    color: #3498db;

    margin: 1.66%;

    letter-spacing: 0.2rem;

}

p{

    margin: 1.66%;

}

hr{

    border: 0;

    height: 1px;

    background-image: linear-gradient(to right, rgba(0, 0, 0, 0), rgba(0, 0, 0, 0.75), rgba(0, 0, 0, 0));

}

.post{

```

```
        margin:20px;

    }

    h1{

        color: #3474db;

        margin:1.66%;

        font-size: 1.3rem;

        font-family:Arial Black Web Safe Font;

        text-align: right;

    }

    h1.two{

        color: #3474db;

        font-size: 25px;

        font-family:Arial Black Web Safe Font;

        text-align: right;

    }

    .post{

        font-size: 20px;

        font-style: oblique;

        font-family: sans-serif;

        margin:20px;
```

```
}

#result {

    color: #FFFFFF;

}

p.big {

    line-height: 1.8;

}

.lg

{

    color: #3474db;

}

.three

{

    color: #e6001a;

}

.upload-labe one

{

    color: #e6001a;

}

</style>
```

</head>

<div class="jumbotron">

<nav class="navbar navbar-expand-lg navbar-light bg-dark">

<div class="container">

<h1 class="two" ><i class="fa fa-free-code-camp fa lg" aria-hidden="true"></i>

</h1>

</div>

</div>

</nav>

<div class="container">

<div id="content" style="margin-top:2em">

<div class="container">

<div class="row">

<div class="col-sm-8 bd" >

<hr>

<h3 class="one"> Predict The Kidney Disease with CKD and NO CKD

</h3>

<p class="post"></p>

</div>

<div class="imageborder" >

</div>

<div class="col-sm-5">

<div>

<h4 class="one post three" ><i class="fa fa-file-image-o" aria-hidden="true">

</i>Upload The Image Here </h4>

<form action = "http://localhost:5000/" id="upload-file" method="post"
enctype="multipart/form-data">

<label for="imageUpload" class="upload-label">

Choose...

</label>

<input type="file" name="image" id="imageUpload" accept=".png, .jpg, .jpeg">

</div>

</div>

<hr>

</div>

<div class="image-section" style="display:none;">

<div class="img-preview">

<div id="imagePreview">

</div>

</div>

<div>

<button type="button" class="btn btn-info btn-lg " id="btn-predict">Predict!</button>

</audio>

</div>

</div>

<div class="loader" style="display:none;"></div>

<h3>

</h3>

</div>

</div>

</div>

</div>

</div>

</div>

</body>

<footer>

<script src="{ {url_for('static', filename='js/main.js') } }" type="text/javascript"></script>

</footer>

</body>

</html>

CHAPTER-4

CONCLUSION AND FUTURE WORK

- We developed this project using python3.
- By taking the dataset we evaluated with different models and checked the accuracy for which model we can get maximum percentage so that particular model can be sustained and taken as good model.
- Future work, we could take this project to higher levels after the good research and we'll prepare a dataset so that we can even give the prescription for the patient according to his health condition and level of disease spread in his body.

Chapter-5

REFERENCES:

<https://www.kaggle.com/mansoordaku/ckdisease>

<https://towardsdatascience.com/data-visualization-for-machine-learning-and-data-sciencea45178970be7>

<https://thesmartbridge.com/documents/spsaimldocs/Datapreprocessing.pdf>

<https://thesmartbridge.com/documents/spsaimldocs/Datapreprocessing.pdf>

<https://thesmartbridge.com/documents/spsaimldocs/Datapreprocessing.pdf>

<https://thesmartbridge.com/documents/spsaimldocs/FlaskML.pdf>