

**AUTOMATED DETECTION OF KIDNEY STONES USING CORONAL CT
IMAGES WITH DEEP LEARNING IN THE FIELD OF AI**

submitted in partial fulfillment of the requirements for the award of Bachelor of
Technology degree in Biomedical engineering

by

P. MONICA (38240018)

W. SHARON BENETTA (38240034)



DEPARTMENT OF BIOMEDICAL ENGINEERING

SCHOOL OF BIO AND CHEMICAL ENGINEERING

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade “A” by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI -600 119

MAY - 2022

DEPARTMENT OF BIOMEDICAL ENGINEERING

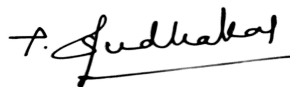
BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **P. MONICA (38240018), W. SHARON BENETTA (38240034)** who carried out the project entitled " **Automated Detection Of Kidney Stones Using Coronal CT Images With Deep Learning In The Field Of AI**" under my supervision from **September 2021 to March 2022**.



Dr. J. BETHANNEY JANNEY, M.E., Ph.D.,

Guide



Dr.T. SUDHAKAR, M.Sc., Ph.D.,

Head of the Department

Submitted for Viva-voce Examination held on 02.05.2022

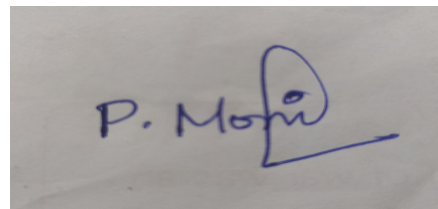
Internal Examiner

External Examiner

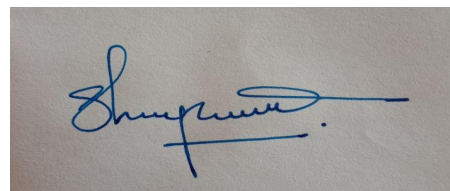
DECLARATION

We **P. MONICA (38240018)** and **W. SHARON BENETTA (38240034)** hereby declare that the project entitled "**Automated Detection of Kidney Stones Using Coronal CT Images With Deep Learning In The Field Of AI**" done by us under the guidance of **Dr. J. BETHANNEY JANNEY** is submitted in partial fulfilment of the requirements for the award of Bachelor of Technology in Biomedical Engineering.

1.

A photograph of a handwritten signature in blue ink on a light-colored surface. The signature appears to be 'P. Monica' with a stylized flourish at the end.

2.

A photograph of a handwritten signature in blue ink on a light-colored surface. The signature is more complex and stylized, appearing to be 'Sharon Benetta'.

DATE: 23.04.2022

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATES

ACKNOWLEDGEMENT

We are pleased to acknowledge our sincere thanks to the **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. We are grateful to them.

We convey our thanks to **Dr. T Sudhakar, Head of the Department, Dept. Biomedical Engineering** for providing me necessary support and details at the right time during the progressive reviews.

We would like to express our sincere and deep sense of gratitude to our Project Guide **Dr. J. Bethanney Janney, ME., Ph.D., Department of Biomedical Engineering** for her valuable guidance, suggestions, and constant encouragement paving the way for the successful completion of our project work.

We wish to express our thanks to all **Teaching and Non-teaching** staff members of the **Department of Biomedical Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Various imaging technique are used for the diagnosis of kidney stone disease. Specialists are needed for the interpretation and full diagnosis of these images. Computer-aided diagnosis systems are the practical approaches that can be used as auxiliary tools to assist the clinicians in their diagnosis. In this project, an automated detection of kidney stones using coronal computed tomography(CT) images was proposed with deep learning (DL) technique which has made significant progress in the field of artificial intelligence. So, to detect the stone and that too precisely paves the way to image processing because through image processing there is a tendency to get the precise results and its an automatic method of detecting the stone. Doctor generally uses the manual method to detect the stone from the Computed Tomography image but our technique is fully automated so it is advantageous as the time reduced and with the chances of error also reduces.

Keywords- *Kidney stone, medical image, Deep Learning, Computed tomography*

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	
	LIST OF FIGURES	
	LIST OF TABLES	
1	INTRODUCTION	1
1.1	Imaging Modalities	2
1.2	Outline of the project	3
2	LITERATURE REVIEW	4
3	AIM AND SCOPE	6
3.1	Aim	6
3.2	Proposed System	6
3.3	Motivation	6
4	METHODOLOGY AND MATERIALS	7
4.1	Block Diagram	7
4.1.2	Collection of datasets	7
4.1.3	pre-processsing	8
4.2	Classification of CNN layer	8
4.2.1	CNN	9
4.2.2	Background of CNN s	9
4.2.3	Working	10
4.2.4	What's a pooling layer?	14
4.2.5	Limitations	16
4.3	Deep learning	17
4.4	Python	17
4.4.1	Numpy	17

4.4.2	Uses of Numpy	17
4.4.3	Keras	17
4.4.4	Tensor Flow	18
4.4.5	Tensorflow Ecosystem	18
4.4.6	Advantages of Tensorflow	18
4.4.7	SKlearn	19
4.4.8	Uses of SKlearn	19
4.4.9	Sklearn in Python	19
4.5	Resnet50	19
4.5.1	Resnet50 Architecture	20
4.5.2	Xception in Python	20
4.5.3	CNN Xception	20
4.5.4	SGD Optimiser	21
4.5.5	Uses	21
4.5.6	Dropout	21
4.5.7	Flatten	21
4.5.8	Dense	22
4.5.9	Dense Layer	22
4.6	Maxpooling 2D	22
4.6.1	Working of Max Pool	22
4.6.2	Conv2D	22
4.6.3	Matplotlib	23
4.6.4	Uses	23
4.6.5	Keras Image Data Generator	23
4.6.6	Pillow	23

4.6.7	Uses	23
4.7	OpenCV	24
4.7.1	Uses	24
5	RESULT AND DISCUSSION	23
5.1	Result	23
5.1.1	Streamlit	25
5.1.2	Graphical Representation of Loss and Accuracy	27
6	SUMMARY AND CONCLUSION	32
6.1	Summary	32
6.2	Conclusion	32
	REFERENCES	33

LIST OF FIGURES

FIGURE NO	TITLE	PAGE NO
1.1	Kidney stone	1
4.1	Block diagram of kidney stone detection	7
4.2	Dataset of kidney stone absent	7
4.3	Dataset of Kidney stone present	8
4.4	Block diagram of CNN layer	8
4.5	A CNN Sequence to classify handwritten digits	10
4.6	4x4x3 RGB Image	11
4.7	Types of Pooling	15
4.8	Block diagram of Resenet50	20
4.9	Architecture of Resenet50	20
5.1	Output Images of kidney stone present	26
5.2	Output Images of kidney stone absent	26
5.3	Training process for epochs 3	27
5.4	Model accuracy and model loss graph	27
5.5	Training process for epochs 10	28
5.6	Model accuracy and model loss graph	28
5.7	Training process for epochs 20	29
5.8	Model accuracy and model loss graph	29
5.9	Training process for epoch 500	30
5.10	Model accuracy and model loss graph	31

LIST OF TABLES

TABLE NO.	TABLE OF CONTENT	PAGE NO.
5.1	Model accuracy and model loss	28
5.2	Model accuracy and model loss	29
5.3	Model accuracy and model loss	30
5.4	Model accuracy and model loss	31
5.5	Final tabulation for loss and accuracy	32

CHAPTER 1

INTRODUCTION

Kidney stones are also known as renal calculi, nephrolithiasis or urolithiasis. These are hard deposits which are made up of minerals and salts. It will not cause any symptoms until it moves. When it passes into the ureters it may create a block in the ureters which may stop the flow of urine causing swelling of the kidney. Some times small stones will come out of the body through urine. The pain caused by the kidney stone may differ because of movement of stones in different locations. The types of kidney stones are as follows:

- Calcium oxalate: This is the common type of kidney stone and found in majority of patients. It is created by combination of calcium with oxalate in the urine. When we have more oxalate or less urine it can cause the oxalate to crystallize and clump together into stones.
- Uric acid: This is caused for the people who lose more fluid because of chronic diarrhea or malabsorption. This stone is a combination of carbon, nitrogen, oxygen and hydrogen.
- Struvite: This stone is caused due to urinary tract infection. It is a combination of magnesium, ammonium and phosphate.
- Cystine: This is found with the people suffering from hereditary disorder called cystinuria. It is formed due to amino acid and these stones are rare.

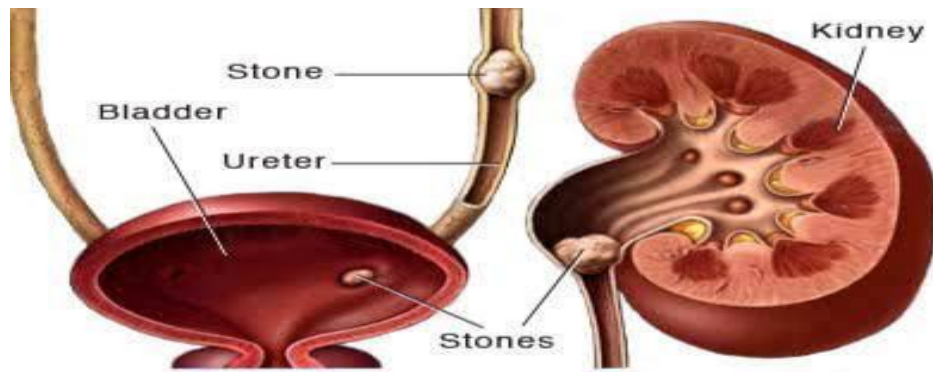


Fig:1.1:Kidney stone

1.1 Imaging Modalities:

Imaging techniques allow to get a picture of the internal parts of the body which can be used to detect any abnormalities. There are different imaging modalities to detect kidney stones. The most commonly used to detect kidney stones are

X-rays

A abdominal x- ray will give the picture of the abdomen it can identify many stones .But through abdominal X-ray not all stones are visible due to gas in digestive tract and overlying structures.

Ultrasound

The ultrasound is also called as sonography it is one of the imaging method which uses high frequency of sound waves to view internal parts of the body. For kidney stone stone detection it create a image of kidney and baldder. The image which captured will provided a information for treatment and diagnosing various type of diseases.

Computed Tomography (CT)

Computed tomography is used to examine inside of the body structures. It uses X-ray and computer to produce cross sectional image of the body. It is a noninvasive method, Painless and very fast. It can also reveal internal injuries and bleeding. The CT scans for kidney stone detection will give more detailed information of the urinary tract and it is useful to detect tumors, stones and other lesions. The 3D image of the stone is generated by the CT which is rebuild into multiple viewing planes. Comparing to other modalities the CT have high sensitivity for detecting kidney stones.

1.2 Outline Of The Project

Nowadays, kidney stone has become a major problem and if not detected at an early stage, then it may cause complications and sometimes surgery is also needed to remove the stone. So, to detect the stone and that too precisely paves the way to image processing because through image processing there is a tendency to get the precise results and it is an automatic method of detecting the stone. Doctor generally uses the manual method to detect the stone from the Computed Tomography image but our technique is fully automated so it is advantageous as the time is reduced and with that the chances of error also reduces. In the past few decades, Deep Learning has proved to be a very powerful tool because of its ability to handle large amounts of data. The interest to use hidden layers has surpassed traditional techniques, especially in pattern recognition. One of the most popular deep neural networks is Convolutional Neural Networks.

Since the 1950s, the early days of AI, researchers have struggled to make a system that can understand visual data. In the following years, this field came to be known as Computer Vision. In 2012, computer vision took a quantum leap when a group of researchers from the University of Toronto developed an AI model that surpassed the best image recognition algorithms and that too by a large margin. The deep learning in which imagenet has been used.

CHAPTER - 2

LITERATURE REVIEW

Vineela et al. (2020) reported that kidney stones are hard collection of salt and minerals often made up of calcium and uric acid. Majority of people with stones in kidney at initial stage do not notice and it damages the organs slowly. It is very important to detect the exact and accurate position of kidney stone for surgical operations. Ultrasound images normally consist of speckle noise which cannot be removed by mankind. Hence we preferred automated techniques in detection of kidney stones in ultrasound images using median filter instead of Gabor filter.

G. Santhoshi et al. (2020) vol 5 reported that kidney stones are not a new subject to being one of the major health concerns of today's day and age, if not detected at early stages might also become life threatening. Detecting a kidney stone might require a technique that ensures precision and also is in wide use. Hence the technique none other than image processing has a tendency to encompass an automatic method to detect stones precisely leading itself to being one of the more popular methods for performing detecting, sensing, or forecasting processes through images. The speckle noise and low contrast of images produced using ultrasound could pose a considerable challenge to detect merely any stones. Therefore is deployed the befitting image processing technique to overcome the challenge. The image restoration which uses Gaussian filter process helps get rid of the speckle noise by pre-processing the produced ultrasound image. Which is also used to smoothen the ultrasound image thus we obtain restored image. Image segmentation is done to the already pre-processed image for the detection of any possible stones using a Canny edge detection technique thereby retaining much prominent edges. Further wavelet transformation and CNN classification is done to the segmented image to identify the presence of stones in a kidney.

Vinayagam.P, Sreemathi.M, (2019) vol 14 developed that due to the presence of noise, there are inaccuracies in the classification of kidney stone. Kidney stone has become common nowadays due to various factors. In the proposed methodology nephrolithiasis in the MRI image is preprocessed using DWT (Discrete Wavelet Transform). Key features are extracted using Gray level co-occurrence matrix (GLCM). A data set of 20 test data containing normal and abnormal kidney MRI

images are classified using Back Propagation Method of Neural Network (BPNN). The output of BPN classification is displayed in LCD board which is interfaced with an Arduinouno board. A Fuzzy Clustering Mean Algorithm (FCM) is used for successful segmentation of kidney stone.

Pooja Gohel (2018) reported has This paper presents a technique for detection of kidney stones through different steps of image processing. The first step is the image pre-processing using filters in which image gets smoothed as well as the noise is removed from the image. Image enhancement is a part of preprocessing which is used to enhance the image which is achieved with power law transformation. Next, the image segmentation is performed on the preprocessed image using thresholding technique. The imaging modality used is CT because it has low noise compared to other modalities such as x-ray and ultrasound.

M.Akshaya,R.Nithushaa, founded as Back-Propagation Network is the most commonly used algorithm in training neural networks. It is employed in processing the image and data to implement an automated kidney stone classification. The conventional technique for medical resonance kidney images classification and stone detection is by human examination. This method is not accurate since it is impractical to handle large amount of data. Magnetic Resonance (MR) Images may inherently possess noise caused by operator errors. This causes earnest inaccuracies in classification features/ diseases in image processing. However, usage of artificial intelligent based methods along with neural networks and feature extraction has shown great potential in extracting the region of interest using back propagation network algorithm in this field. In this work, the Back Propagation Network was applied for the objective of kidney stone detection. Decision making is carried out in two stages: 1.Feature extraction 2.Image classification. The feature extraction is done using the principal component analysis and the image classification is done using Back Propagation Network (BPN). This work presents segmentation method using Fuzzy C-Mean (FCM) clustering algorithm. The performance of the BPN classifier was estimated in terms of training execution and classification accuracies. Back Propagation Network gives precise classification when compared to other methods based on neural network

CHAPTER 3

AIM &SCOPE

3.1 AIM

The aim of this project is to detect kidney stones using coronal CT images with deep learning in the field of Artificial Intelligence. kidney stones are common worldwide. Nowadays, even childrens are found with kidney stones. The prevention of this remains to be a serious problem. As it says " Early detection save lives" the earlier the stones are predicted the lower the chance of high risk. We commonly know doctors by viewing the scan report says the result as, whether a patient is detected with kidney stone or not. But the question is how the computer can view an image and give the exact result as doctors. So by using deep learning technique with image recognition algorithms a computer can recognize the given dataset and can give the exact result. Our technique is fully automated, time consuming will be less and no need to wait for doctor to check the scan reports individually as the dataset of the imgaes are already updated in the computer the results will be faster and many sets of images result can be viewed within a short period of time.

3.2 Proposed system

The development of technology, a lot of computer-aided systems has been evolved for diagnosing the presence of disease with an important emphasis on system accuracy. In proposed system, we develop an automated detection of kidney stones using coronal computed tomography (CT) images was proposed with deep learning (DL) technique which has made significant progress in the field of artificial intelligence.

3.3 Motivation

The development of imaging techniques, diagnosis using computed Tomography (CT) images has become widespread because of its low cost, reliable and non-invasive procedure. feature extraction, analysis and pattern recognition techniques for these images are used for finding the abnormality like a tumor, cyst, stone etc.

CHAPTER 4

METHODOLOGY AND MATERIAL

4.1 Block diagram

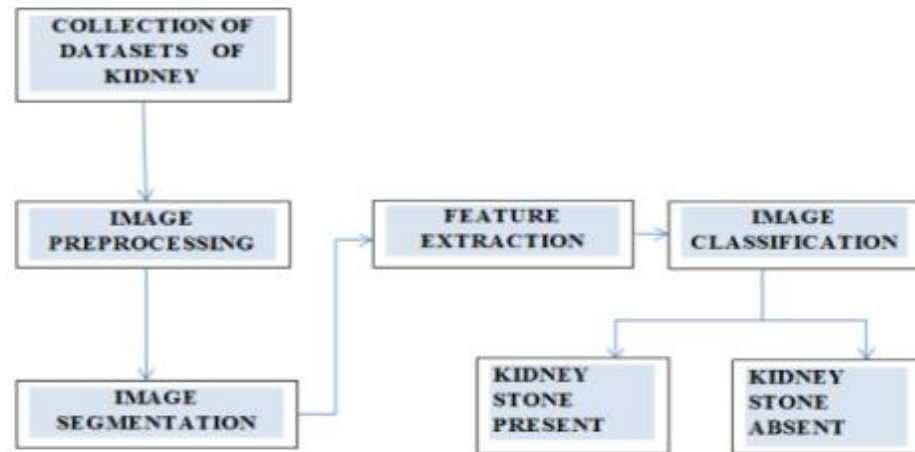


Fig:4.1: Block Diagram of Kidney stone detection

4.1.2 Collection of datasets





Fig:4.2: Dataset of kidney stone absent

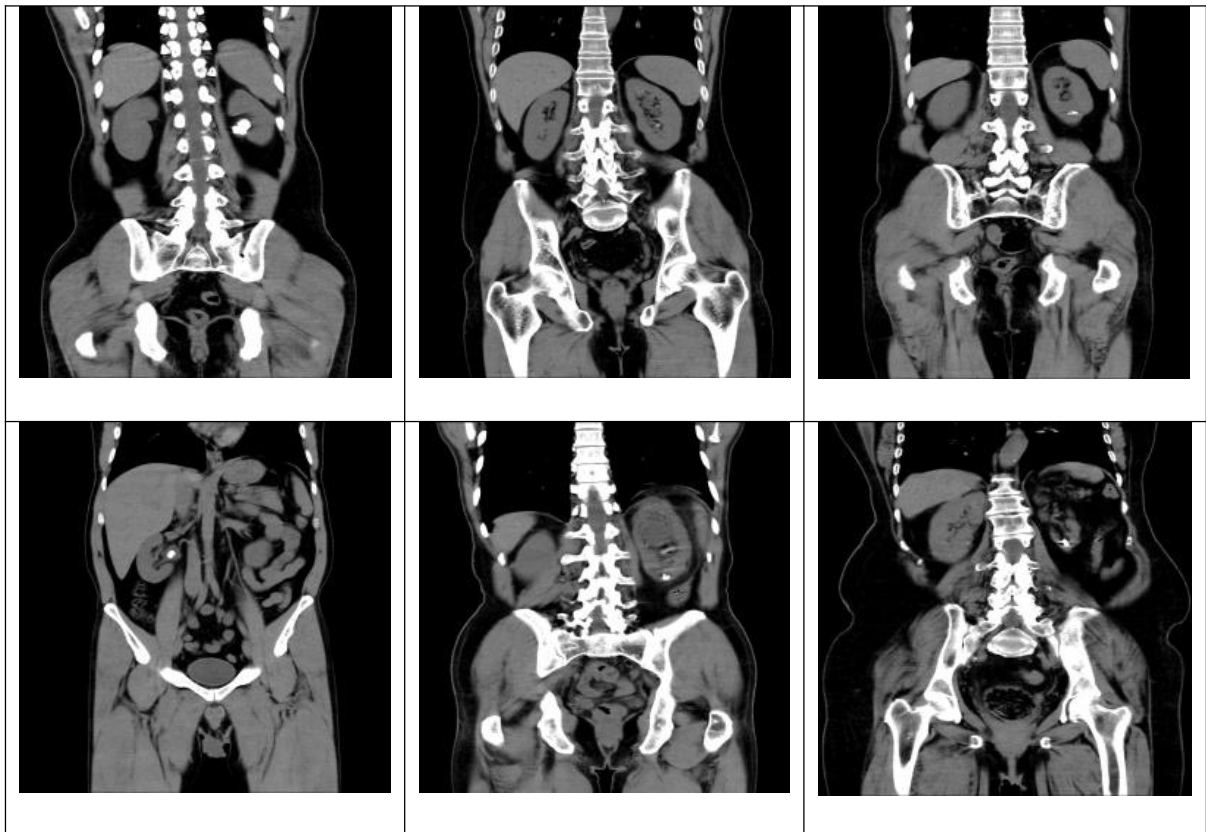


Fig:4.3: Dataset of kidney stone present

4.1.3 Pre- Processing

The images in the database are exposed to the pre-processing stage where the resizing task is carried out to maintain the uniform dimensions among the images.

4.2 Classification of CNN Layer

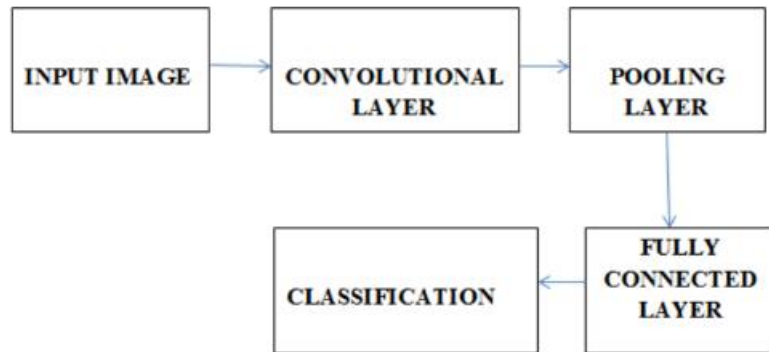


Fig:4.4: Block diagram of CNN Layer

4.2.1 Convolutional Neural Networks (CNN)

In deep learning, a convolutional neural network (CNN/ConvNet) is a class of deep neural networks, most commonly applied to analyse visual imagery. Now when we think of a neural network, we think about matrix multiplications but that is not the case with ConvNet. It uses a special technique called Convolution. Now in mathematics convolution is a mathematical operation on two functions that produces a third function that expresses how the shape of one is modified by the other.

But we don't really need to go behind the mathematics part to understand what a CNN is or how it works. Bottom line is that the role of the ConvNet is to reduce the images into a form that is easier to process, without losing features that are critical for getting a good prediction.

4.2.2 Background of CNNs

CNNs were first developed and used around the 1980s. The most that a CNN could do at that time was recognize handwritten digits. It was mostly used in the postal sectors to read zip codes, pin codes, etc. The important thing to remember about any deep learning model is that it requires a large amount of data to train and also requires a lot of computing resources. This was a major drawback for CNNs at that period and hence CNNs were only limited to the postal sectors and it failed to enter the world of machine learning.

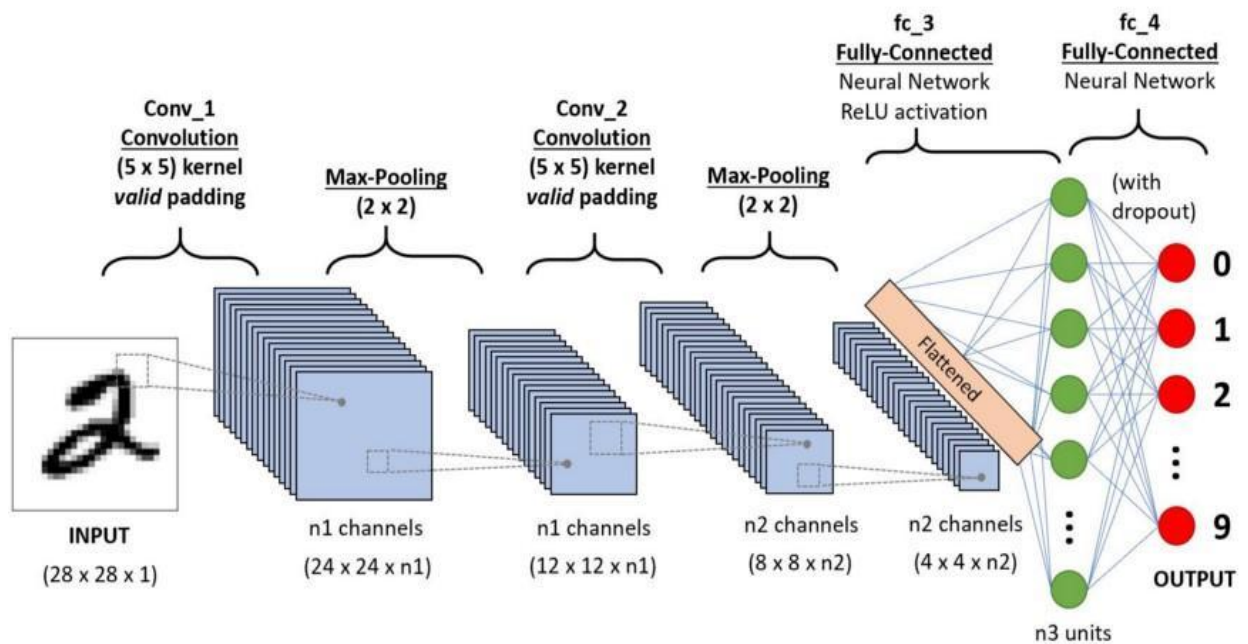


Fig:4.5: A CNN sequence to classify handwritten digits

In 2012 Alex Krizhevsky realized that it was time to bring back the branch of deep learning that uses multi-layered neural networks. The availability of large sets of data, to be more specific ImageNet datasets with millions of labeled images and an abundance of computing resources enabled researchers to revive CNNs.

4.2.3 Working

Before we go to the working of CNN's let's cover the basics such as what is an image and how is it represented. An RGB image is nothing but a matrix of pixel values having three planes whereas a grayscale image is the same but it has a single plane. Take a look at this image to understand more.

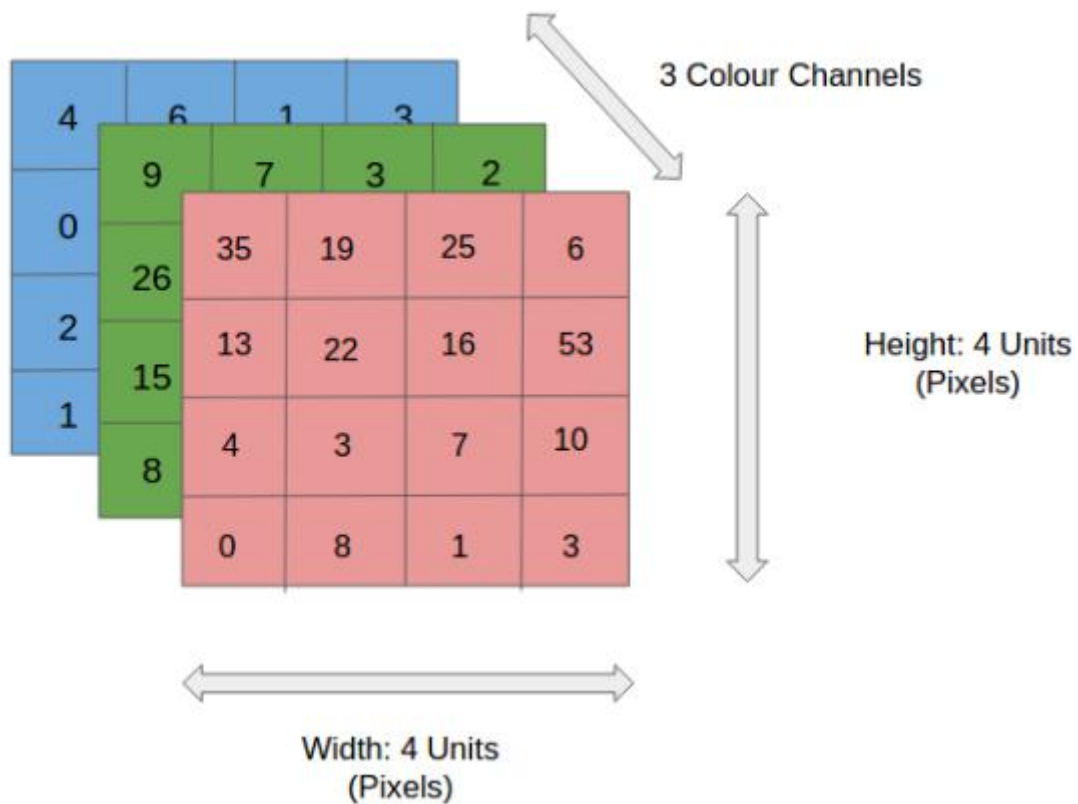
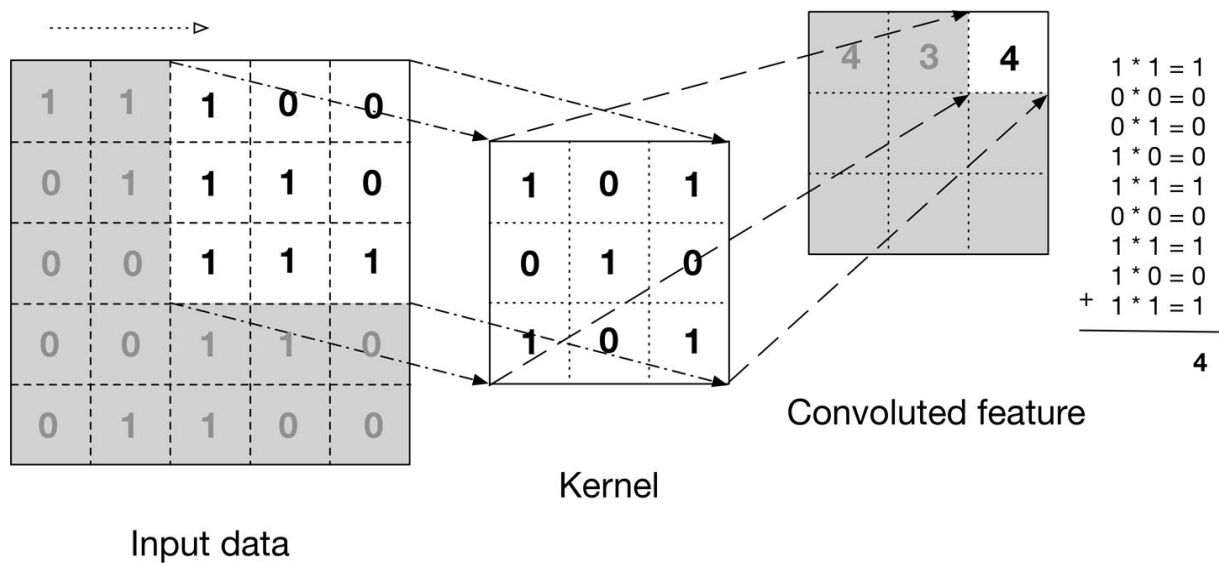
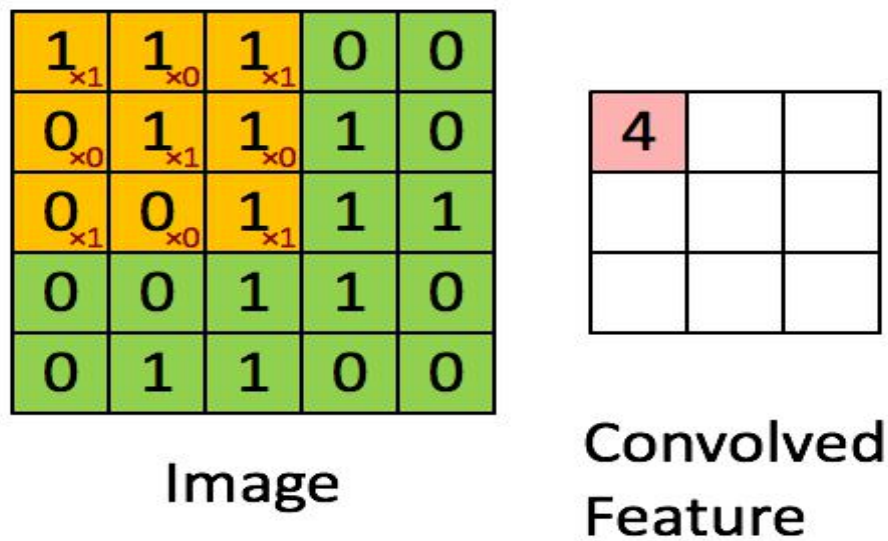


Fig:4.6: 4x4x3 RGB Image

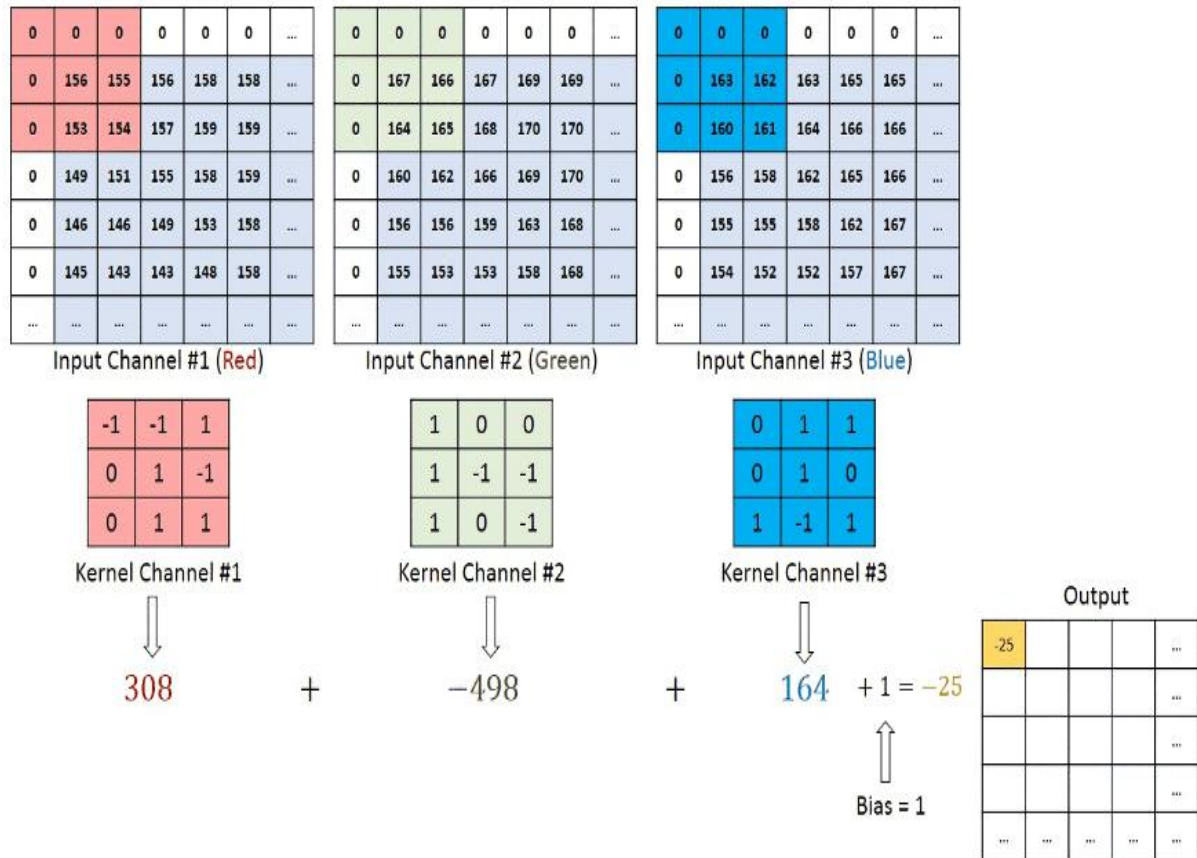
For simplicity, let's stick with grayscale images as we try to understand how CNNs work.



The above image shows what a convolution is. We take a filter/kernel(3×3 matrix) and apply it to the input image to get the convolved feature. This convolved feature is passed on to the next layer.



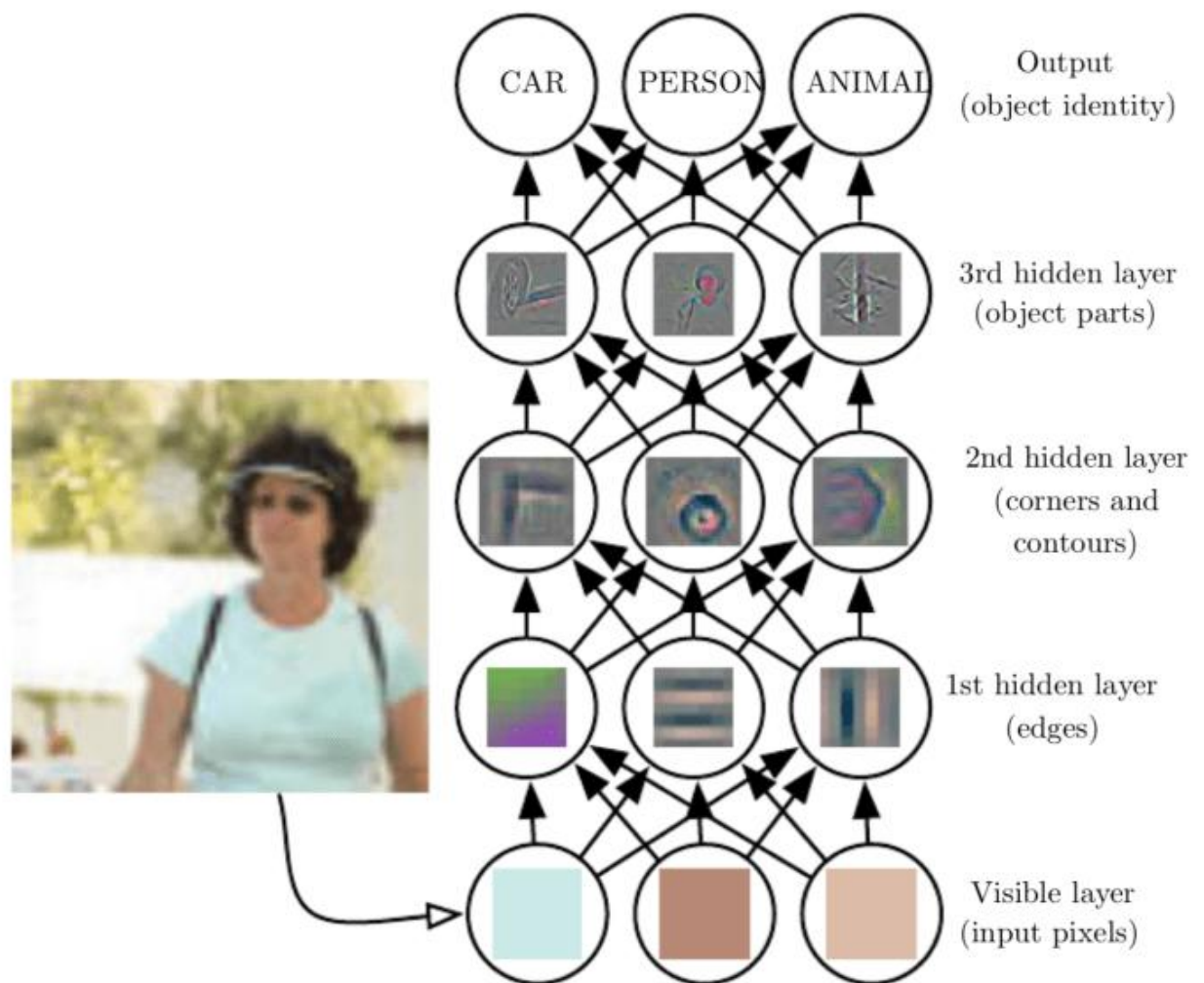
In the case of RGB color, channel take a look at this animation to understand its working



Convolutional neural networks are composed of multiple layers of artificial neurons. Artificial neurons, a rough imitation of their biological counterparts, are mathematical functions that calculate the weighted sum of multiple inputs and outputs an activation value. When you input an image in a ConvNet, each layer generates several activation functions that are passed on to the next layer. The first layer usually extracts basic features such as horizontal or diagonal edges. This output is passed on to the next layer which detects more complex features such as corners or combinational edges. As we move deeper into the network it can identify even more complex features such as objects, faces, etc.



Based on the activation map of the final convolution layer, the classification layer outputs a set of confidence scores (values between 0 and 1) that specify how likely the image is to belong to a “class.” For instance, if you have a ConvNet that detects cats, dogs, and horses, the output of the final layer is the possibility that the input image contains any of those animals.



4.2.4 What's a pooling layer?

Similar to the Convolutional Layer, the Pooling layer is responsible for reducing the spatial size of the Convolved Feature. This is to decrease the computational power required to process the data by reducing the dimensions. There are two types of pooling average pooling and max pooling. I've only had experience with Max Pooling so far I haven't faced any difficulties.

3.0	3.0	3.0
3.0	3.0	3.0
3.0	2.0	3.0

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

So what we do in Max Pooling is we find the maximum value of a pixel from a portion of the image covered by the kernel. Max Pooling also performs as a Noise Suppressant. It discards the noisy activations altogether and also performs de-noising along with dimensionality reduction. On the other hand, Average Pooling returns the average of all the values from the portion of the image covered by the Kernel. Average Pooling simply performs dimensionality reduction as a noise suppressing mechanism. Hence, we can say that Max Pooling performs a lot better than Average Pooling.

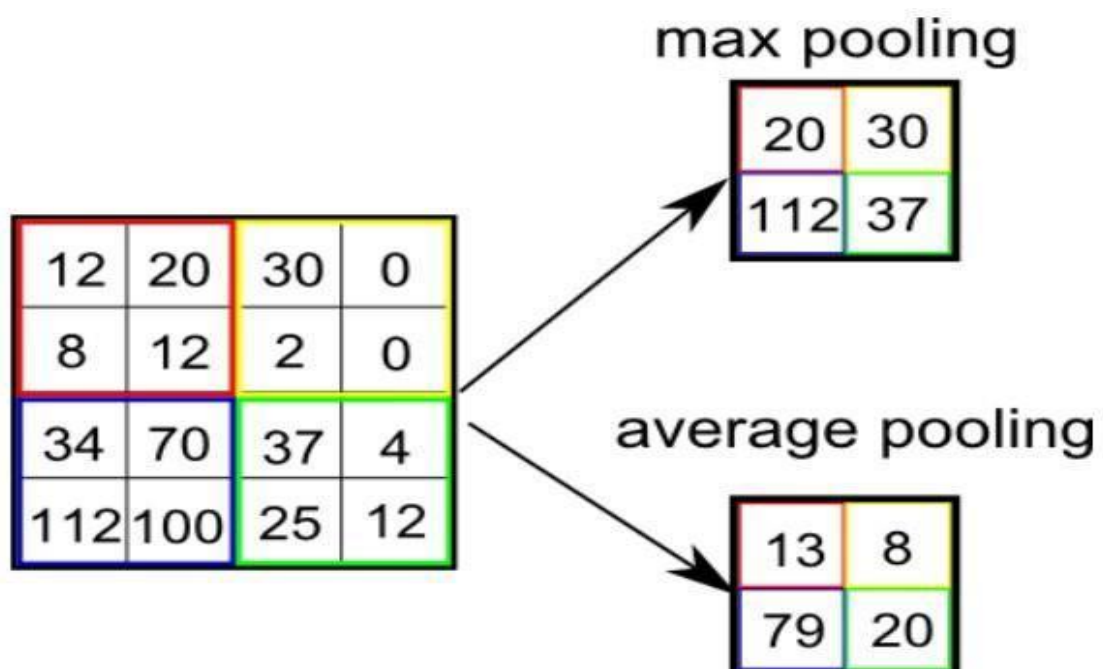


Fig:4.7: Types of Pooling

4.2.5 Limitations

Despite the power and resource complexity of CNNs, they provide in-depth results. At the root of it all, it is just recognizing patterns and details that are so minute and inconspicuous that it goes unnoticed to the human eye. But when it comes to understanding the contents of an image it fails. Let's take a look at this example. When we pass the below image to a CNN it detects a person in their mid-30s and a child probably around 10 years.

But when we look at the same image we start thinking of multiple different scenarios. Maybe it's a father and son day out, a picnic or maybe they are camping. Maybe it is a school ground and the child scored a goal and his dad is happy so he lifts him.

These limitations are more than evident when it comes to practical applications. For example, CNNs were widely used to moderate content on social media. But despite the vast resources of images and videos that they were trained on it still isn't able to completely block and remove inappropriate content. As it turns out it flagged a 30,000-year statue with nudity on Facebook.

Several studies have shown that CNNs trained on ImageNet and other popular datasets fail to detect objects when they see them under different lighting conditions and from new angles. Despite the limits of convolutional neural networks, however, there's no denying that they have caused a revolution in artificial intelligence.

Today, CNNs are used in many computer vision applications such as facial recognition, image search, and editing, augmented reality, and more. As advances in convolutional neural networks show, our achievements are remarkable and useful.

4.3 Deep learning

Deep learning is a subfield of machine learning concerned with algorithms inspired by the structural and functional of the brain called artificial neural network. It is used in Image Recognition, Fraud Detection, News Analysis, Stock Analysis, Self-driving cars, Healthcare like cancer image analysis, etc. By inputting more data in the network the layers get trained very well. They can be classified into Supervised, Semi-Supervised and Unsupervised categories. Each layer is known for extracting information specifically. For example in Image recognition, the first layer will find the edge, lines, etc, second layer like eye, ear, nose, etc. In Image processing and deep learning may identify the concepts relevant to human such as digital image.

4.4 PYTHON

4.4.1 Numpy

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

4.4.2 Uses of Numpy

NumPy can be used to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.

4.4.3 Keras

Keras is a powerful and easy-to-use free open-source Python library for developing and evaluating deep learning models. It wraps the efficient numerical computation libraries Theano and TensorFlow and allows you to define and train neural network models in just a few lines of code. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result as fast as possible is key to

doing good research. Keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear and actionable feedback upon user error.

4.4.4 Tensor Flow

TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks.

4.4.5 TensorFlow Ecosystem

TensorFlow provides a collection of workflows to develop and train models using Python or JavaScript, and to easily deploy in the cloud, on-prem, in the browser, or on-device no matter what language you use. The tf. data API enables you to build complex input pipelines from simple, reusable pieces. It is a symbolic math library that uses dataflow and differentiable programming to perform various tasks focused on training and inference of deep neural networks.

4.4.6 Advantages of TensorFlow

- Open-source platform. It is an open-source platform that makes it available to all the users around and ready for the development of any system on it.
- Data visualization.
- Keras friendly.
- Scalable.
- Compatible.
- Parallelism.
- Architectural support.

- Graphical support.
- Difference Between Keras and Tensor Flow
- Keras is a neural network library while TensorFlow is the open-source library for a number of various tasks in machine learning. TensorFlow provides both high-level and low-level APIs while Keras provides only high-level APIs.

4.4.7 SKlearn

Scikit-learn (formerly scikits.learn and also known as sklearn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support-vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. Scikit-learn is a NumFOCUS fiscally sponsored project.

4.4.8 Uses of Sklearn

Scikit-learn is probably the most useful library for machine learning in Python. The sklearn library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction.

4.4.9 Sklearn in Python

Scikit-learn is a free machine learning library for Python. It features various algorithms like support vector machine, random forests, and k-neighbours, and it also supports Python numerical and scientific libraries like NumPy and SciPy .

4.5 Resnet50

ResNet-50 is a convolutional neural network that is 50 layers deep. You can load a pretrained version of the network trained on more than a million images from the ImageNet database. The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals.

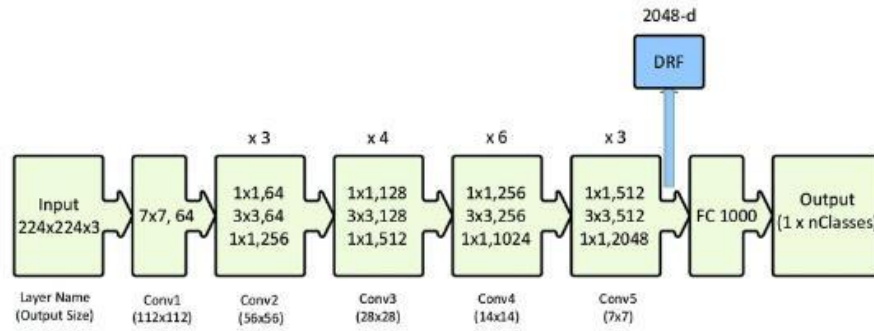


Fig:4.8: Block diagram Resnet50

4.5.1 Resnet50 Architecture

ResNet50 is a variant of ResNet model which has 48 convolution layers along with 1 Max pool and 1 Average pool layer. It has 3.8×10^9 floating points operations. It is a widely used ResNet model and we have explored ResNet50 architecture in depth.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112 × 112	7 × 7, 64, stride 2				
conv2_x	56 × 56	3 × 3 max pool, stride 2				
		$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28 × 28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14 × 14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7 × 7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1 × 1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Fig:4.9: Architecture of Resnet50

4.5.2 Xception in Python

Xception is an extension of the inception Architecture which replaces the standard Inception modules with depthwise Separable Convolutions. The architecture of Xception (Entry Flow>Middle Flow>Exit FLOW)

4.5.3 Cnn Xception

Xception is a deep convolutional neural network architecture that involves Depthwise Separable Convolutions. ... This observation leads them to propose a novel deep convolutional neural network architecture inspired by Inception, where Inception modules have been replaced with depthwise separable convolutions.

4.5.4 SGD Optimizer

Stochastic gradient descent is an iterative method for optimizing an objective function with suitable smoothness properties. It can be regarded as a stochastic approximation of gradient descent optimization, since it replaces the actual gradient by an estimate there

4.5.5 Uses

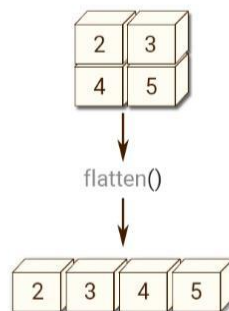
Stochastic gradient descent (often abbreviated SGD) is an iterative method for optimizing an objective function with suitable smoothness properties (e.g., differentiable or subdifferentiable).

4.5.6 Dropout

Dropout is a technique used to prevent a model from overfitting. Dropout works by randomly setting the outgoing edges of hidden units (neurons that make up hidden layers) to 0 at each update of the training phase.

4.5.7 Flatten

The `flatten()` function is used to get a copy of an given array collapsed into one dimension. syntax: `ndarray.flatten(order='C')`



4.5.8 Dense

Dense is the only actual network layer in that model. A Dense layer feeds all outputs from the previous layer to all its neurons, each neuron providing one output to the next layer. It's the most basic layer in neural networks.

4.5.9 Dense Layer

Dense Layer is simple layer of neurons in which each neuron receives input from all the neurons of previous layer, thus called as dense. Dense Layer is used to classify image based on output from convolutional layers.

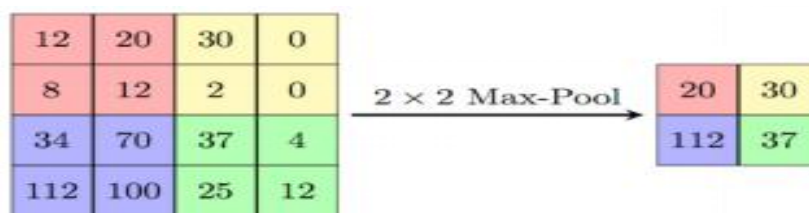
4.6 Maxpooling 2D

Max pooling operation for 2D spatial data. Down samples the input along its spatial

dimensions (height and width) by taking the maximum value over an input window (of size defined by `pool_size`) for each channel of the input.

4.6.1 Working of Max Pool

Max Pooling is a pooling operation that calculates the maximum value for patches of a feature map, and uses it to create a down sampled (pooled) feature map. It is usually used after a convolutional layer.



4.6.2 Conv2D

Conv2D class. 2D convolution layer (e.g., spatial convolution over images). This layer creates a convolution kernel that is convolved with the layer input to produce a tensor of outputs. If `use_bias` is True, a bias vector is created and added to the outputs.

4.6.3 Matplotlib

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK.

4.6.4 Uses

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible. Create publication quality plots. Make interactive figures that can zoom, pan, update.

4.6.5 Keras Image Data Generator

Keras ImageDataGenerator is a gem! It lets you augment your images in real-time while your model is still training! You can apply any random transformations on each training image as it is passed to the model. This will not only make your model robust but will also save up on the overhead memory.

4.6.6 Pillow

Python Imaging Library is a free and open-source additional library for the Python programming language that adds support for opening, manipulating, and saving many different image file formats. It is available for Windows, Mac OS X and Linux.

4.6.7 Uses

Pillow is a Python Imaging Library (PIL), which adds support for opening, manipulating, and saving images. The current version identifies and reads a large number of formats. Write support is intentionally restricted to the most commonly used interchange and presentation formats.

4.7 Opencv

OpenCV is a library of programming functions mainly aimed at real-time computer vision. Originally developed by Intel, it was later supported by Willow Garage then Itseez. The library is cross-platform and free for use under the open-source Apache 2 License.

4.7.1Uses

OpenCV is a great tool for image processing and performing computer vision tasks. It is an open-source library that can be used to perform tasks like face detection, objection tracking, landmark detection, and much more. It supports multiple languages including python, java C++

CHAPTER -5

RESULT AND DISCUSSION

5.1 RESULT

The collection of datasets of kidney is around 1,799 images and they are divided into training and testing set. The images in the database are exposed to the pre-processing stage where the resizing task is carried out to maintain the uniform dimension among the images. Opencv is a great tool for image processing and performing computer vision tasks. To initialize training method and training rate we used xception and set optimizer as SGD, We trained to 500 epochs and we achieved 98% accuracy. This can detect a image with two classes that is kidney stone is present and kidney stone is absent and the result will be shown in grey colour. We have used streamlit to execute the results as it is an open source python library for machine learning and data science.

5.1.1 Streamlit

Streamlit is an open-source Python library that makes it easy to create and share beautiful, custom web apps for machine learning and data science. In just a few minutes you can build and deploy powerful data apps. It is the fastest and easiest way. It is the one of the network which works under the server of googlenet to give specific output.

By using streamlit run app.py we can upload the images in the imagenet and we get the output has whether the stone is present or absent.

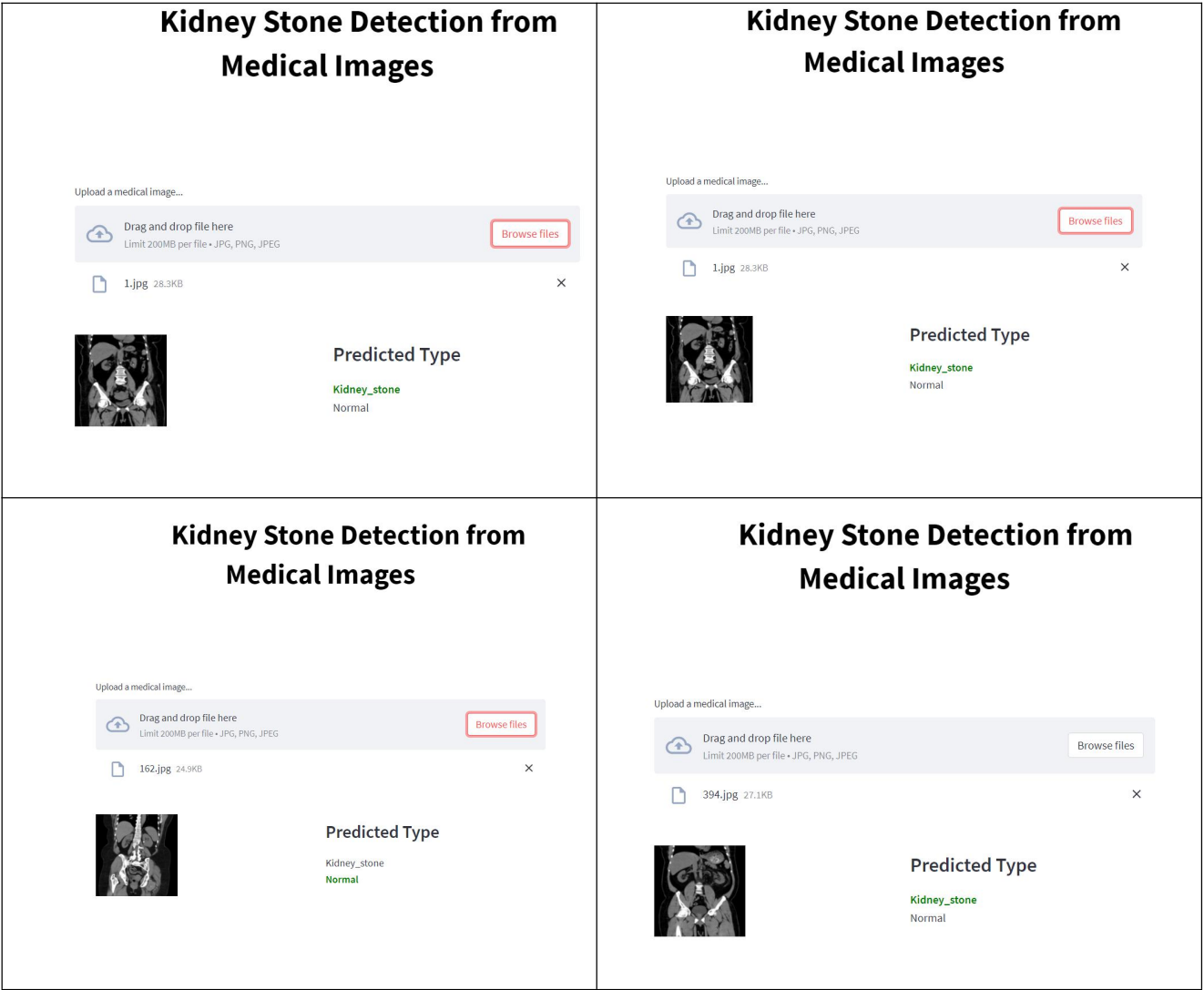


Fig:5.1: Output Images of Kidney stone present

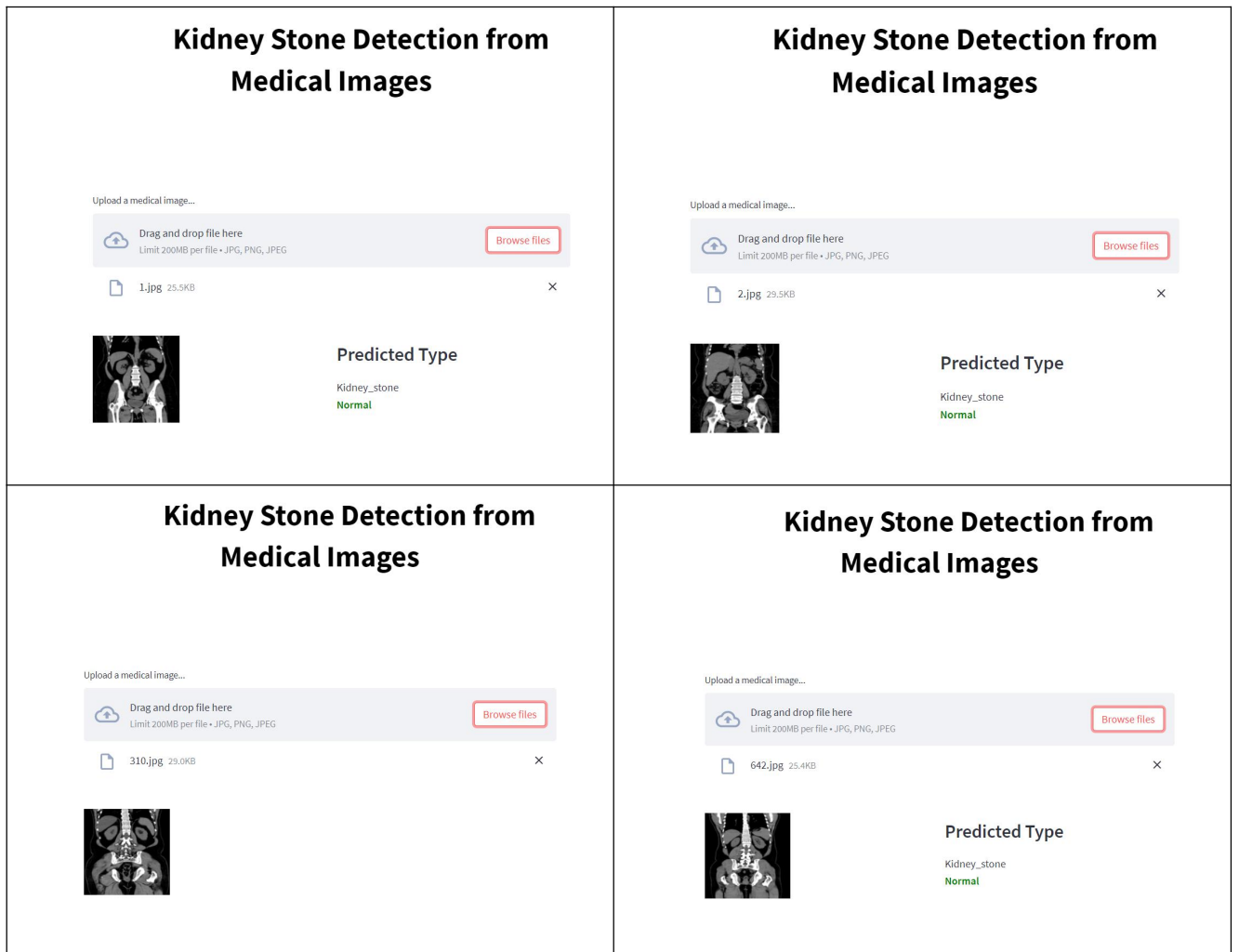


Fig:5.2: Output Images of Kidney stone absent

5.1.2 Graphical Representation of Loss and Accuracy Values

Xception tf.keras.applications...

jupyter train_cnn Last Checkpoint: 11/30/2021 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel) C

Run

To Start Training Process

```
In [7]: # Training
Training_model = model.fit(training_set,
                           steps_per_epoch= 32,
                           epochs= 3,
                           validation_data= testing_set,
                           validation_steps = 32)
```

Epoch 1/3
32/32 [=====] - ETA: 0s - loss: 0.9367 - accuracy: 0.5195WARNING:tensorflow:Your input ran out of data; interrupting training. Make sure that your dataset or generator can generate at least `steps\_per\_epoch \* epochs` batches (in this case, 25 batches). You may need to use the repeat() function when building your dataset.
32/32 [=====] - 170s 5s/step - loss: 0.9367 - accuracy: 0.5195 - val_loss: 1.4282 - val_accuracy: 0.5116
Epoch 2/3
32/32 [=====] - 148s 5s/step - loss: 0.8192 - accuracy: 0.5410
Epoch 3/3
32/32 [=====] - 148s 5s/step - loss: 0.8522 - accuracy: 0.5010

Fig:5.3: Training process for epochs 3

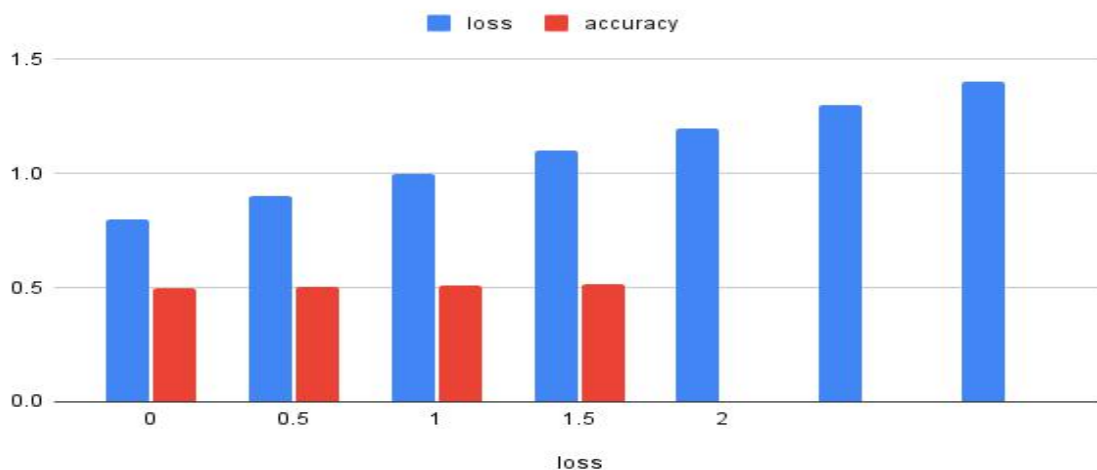


Fig:5.4: Model accuracy and model loss graph

Epochs	Accuracy	Loss
3	0.5010	0.8522

Table:5.1: Model accuracy and model loss

Xception  tf.keras.applications...

jupyter train_cnn Last Checkpoint: 16 hours ago (unsaved changes)  Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

To Start Training Process

```
In [10]: # Training
Training_model = model.fit(training_set,
                           steps_per_epoch= 32,
                           epochs= 10,
                           validation_data= testing_set,
                           validation_steps = 32)
```

```
Epoch 1/10
32/32 [=====] - ETA: 0s - loss: 0.8300 - accuracy: 0.5206WARNING:tensorflow:Your input ran out of data;
interrupting training. Make sure that your dataset or generator can generate at least `steps_per_epoch * epochs`
batches (in this case, 32 batches). You may need to use the repeat() function when building your dataset.
32/32 [=====] - 163s 5s/step - loss: 0.8300 - accuracy: 0.5206 - val_loss: 1.0418 - val_accuracy: 0.4942
Epoch 2/10
32/32 [=====] - 139s 4s/step - loss: 0.8088 - accuracy: 0.5312
Epoch 3/10
32/32 [=====] - 141s 4s/step - loss: 0.7957 - accuracy: 0.5488
Epoch 4/10
32/32 [=====] - 142s 4s/step - loss: 0.8198 - accuracy: 0.5312
Epoch 5/10
32/32 [=====] - 145s 5s/step - loss: 0.8069 - accuracy: 0.5410
Epoch 6/10
32/32 [=====] - 144s 4s/step - loss: 0.7456 - accuracy: 0.6016
Epoch 7/10
32/32 [=====] - 142s 4s/step - loss: 0.7761 - accuracy: 0.5737
Epoch 8/10
32/32 [=====] - 142s 4s/step - loss: 0.7410 - accuracy: 0.6035
Epoch 9/10
32/32 [=====] - 142s 4s/step - loss: 0.7467 - accuracy: 0.5697
Epoch 10/10
32/32 [=====] - 141s 4s/step - loss: 0.7362 - accuracy: 0.6012
```

Fig:5.5: Training process for epochs 10

loss and accuracy

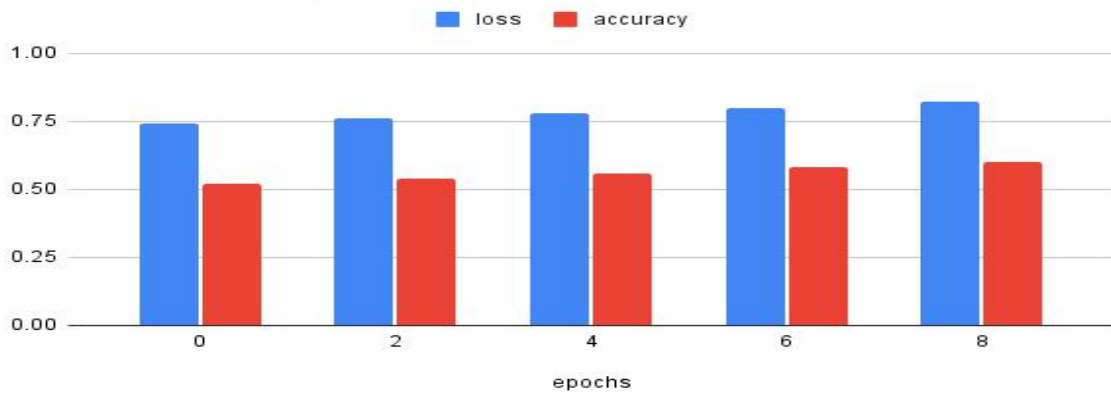


Fig:5.6: Model accuracy and model loss graph

Epochs	Accuracy	Loss
10	0.6012	0.7362

Table:5.2: Model accuracy and model graph

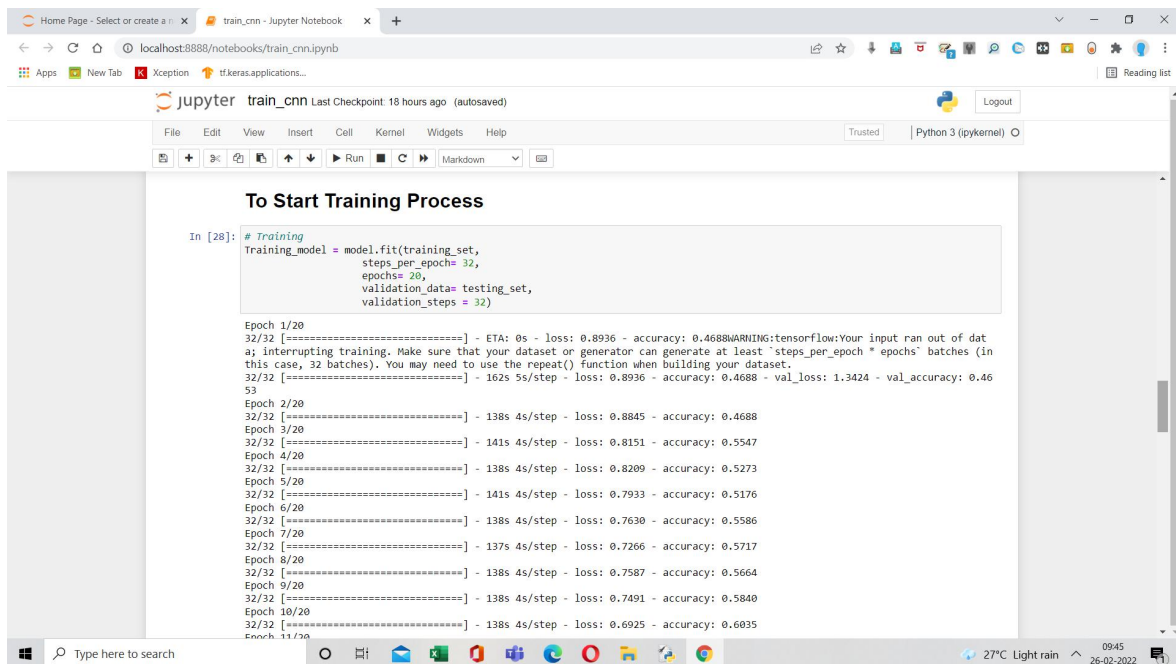


Fig:5.1.5: Training process for epochs 20

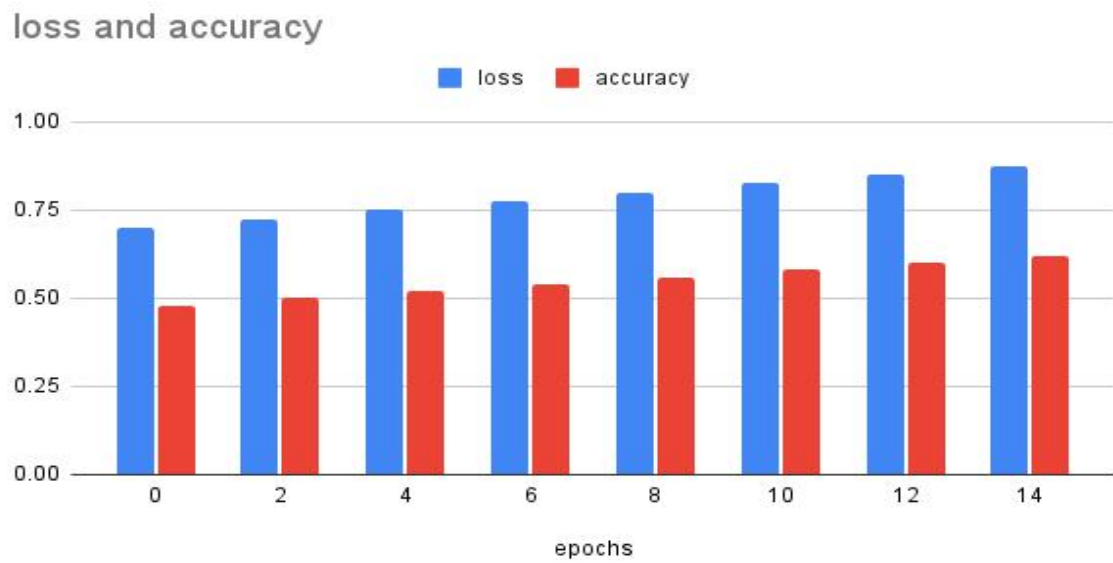


Fig :5.7: Model accuracy and model loss graph

Epochs	Accuracy	Loss
20	0.6640	0.6343

Table:5.3:Model accuracy and model loss

To Start Training Process

```
In [6]: # Training
Training_model = model.fit(training_set,
                           steps_per_epoch= 20,
                           epochs= 500,
                           validation_data= validation_set,
                           validation_steps = 20)

20/20 [=====] - 84s 4s/step - loss: 0.0496 - accuracy: 0.9968
Epoch 492/500
20/20 [=====] - 84s 4s/step - loss: 0.0558 - accuracy: 0.9905
Epoch 493/500
20/20 [=====] - 84s 4s/step - loss: 0.0521 - accuracy: 0.9969
Epoch 494/500
20/20 [=====] - 84s 4s/step - loss: 0.0437 - accuracy: 0.9969
Epoch 495/500
20/20 [=====] - 85s 4s/step - loss: 0.0448 - accuracy: 0.9906
Epoch 496/500
20/20 [=====] - 85s 4s/step - loss: 0.0448 - accuracy: 0.9906
Epoch 497/500
20/20 [=====] - 85s 4s/step - loss: 0.0617 - accuracy: 0.9906
Epoch 498/500
20/20 [=====] - 85s 4s/step - loss: 0.0560 - accuracy: 0.9844
Epoch 499/500
20/20 [=====] - 85s 4s/step - loss: 0.0424 - accuracy: 1.0000
Epoch 500/500
20/20 [=====] - 85s 4s/step - loss: 0.0480 - accuracy: 1.0000
```

Fig:5.8: Training process for epochs 500

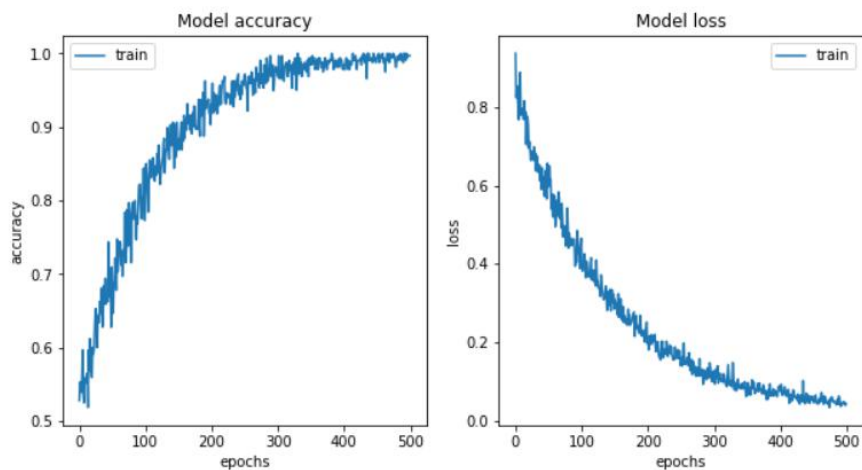


Fig:5.9: Model accuracy and model loss graph

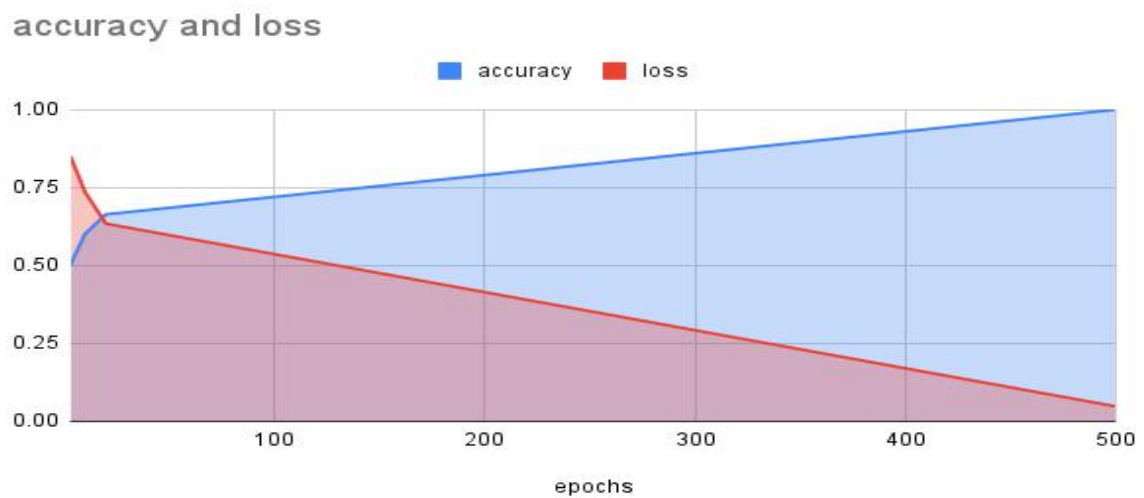
Epochs	Accuracy	Loss
500	1.0000	0.0480

Table:5.4:Model accuracy and model loss

Table:5.5:Final tabulation for loss and accuracy

Epochs	Accuracy	Loss
3	0.5010	0.8522
10	0.6012	0.7362
20	0.664	0.6343
500	1.0000	0.0480

Fig:5.10:Final graph for loss and accuracy



CHAPTER - 6

SUMMARY AND CONCLUSION

6.1 SUMMARY

This model developed is related to the detection of kidney stones using coronal CT images with deep learning technique. we trained the model by using openCV, keras, Tensor flow and achieved an accuracy. The images were resized successfully. By CNN Image recognition is done. The dataset of the input images are classified as Kidney stone present and Kidney stone absent. With streamlit the required results are obtained. As the model developed gives instantaneous results, time consuming will be less hence it can be used to find various abnormalities.

6.2 CONCLUSION

As the computer- aided diagnosis systems are practical approaches, image processing paves the way for kidney stone detection because of the tendency to get precise results. The Artificial Intelligence is an emerging trend and it is used to describe machine learning algorithms. so in the field of AI with deep learning technique the automated of kidney stone detection is achieved with coronal CT images .By the development of automated method to find kidney stones it is helpful to find the abnormalities like stone, cyst etc.

REFERENCES

1. Riya Mishr, Avik Bhattacharjee, M.Gayathri, C.Malathy, "Kidney Stone Detection with CT Images Using Neural Network,"International Journal of Psychosocial Rehabilitation, Vol. 24, Issue,pp:2492-2497, August,2020.
2. Venkatasubramani.K, K.Chaitanya Nagu, P. Karthik, A.Lalith Vikas," Kidney Stone Detection Using Image Processing and Neural Networks," Vol.25,pp.13112-13119,May.2021.
3. Alok Hasoriya, Shrikant D. Zade, Leena H.Patil, "Identification of Kidney stone Disease with CT scan images using CNN Algorithm,"(IRJET),Vol.8, June. 2021.
4. Kadir Yildirim, Pinar Gundogan Bozdog, Muhammed Talo, Ozal Yildirim, Murat Karabatak, U.Rajendra Acharya, " Deep learning model for automated Kidney Stone Detection Using Coronal CT Images," vol.135, August, 2021.
5. T. Vineela¹, R. V. G. L. Akhila², "Kidney stone analysis using digital image processing " by International Journal of Research in Engineering, Science and Management Volume-3, Issue-3,pp275-277, March-2020.
- 6.Nvagireddi Amrutha Lakshmi,Bodasakurti Jyothirmayi,"Kidney Stone Detection From Ultrasound Images By UsingCanny Edge Detection And CNN Classification"by International Journal for Research in Engineering Application & Management (IJREAM) ISSN : 2454-9150 Vol-05, Issue-12,pp228-232, Mar 2020.

7 .Vinayagam,"et.al Kidney stone detection using neural network " by International Journal of Applied Engineering Research ISSN 0973-4562 Volume 14, Number 6, 2019 (Special Issue).

8. M.Akshaya,R.Nithushaa,N.Sri Madhava Raja,S.Padmapriya,Kidney Stone Detection Using Neural Networks "by Auckland University of Technology,IEEE ICSCAN December 23,2020.

9. Annameti Rohith ,S. Premkumar,Detection of Kidney Stones in Ultrasound Images Using Median Filter Compared with Rank Filter" by ISSN: 2237-0722Vol. 11 No. 4 (2021).

10.Stalina ,Adithi,"Kidney Stone Detection Using Image Processing and Neural Networks" by International Journal of Management, Technology And Engineering Volume 8, Issue XI,pp:2257-2260, NOVEMBER/2018,

11. Fuzhe ma ,lingyun liu, Hongyu jing, "Detection and diagnosis of chronic kidney disease using deep learning-based heterogeneous modified artificial neural network" by vol 11 October 2020.

12. P. Archana; M. V Chiranth; S. Chethan; K. N. Jeevan Reddy; A. Sanketha Gowda, "AUTOMATIC KIDNEY LESION DETECTION FOR CT IMAGES USING MORPHOLOGICAL CNN" by International Journal of Advanced Research in Computer Science . 2021 Special Issue, Vol. 12, p112-117. 6p. vol 12,2021.

13. Yinsheng li, Ke Li , John W.Garett ,Guang -Hong Chen,"Generation of virtual non-contrast (VNC image) from dual energy CT scans using deep learning" by SPIE Medical Imaging, 15 february 2021.

14. Guy Hidas, MD Ruth Eliahou, MD Mordechai Duvdevani,"Determination of Renal Stone Composition with Dual-Energy CT: In Vivo Analysis and Comparison

with X-ray Diffraction” by Radiology: Volume 257: Number 2,pp:394-398,November 2020.

15. James C. Williams Jr, Allen Rodgers ,”Urine and stone analysis for the investigation of the renal stone former a consensus conference” , by Epub,pp:1-16, february 2021.

16.MehmetBaygina,OrhanYamanb,”Exemplar Darknet19 feature generation technique for automated kidney stone detection with coronal CT images” by Artificial Intelligence in Medicine ,volume 127, May 2022.

17.Jorge F. Lazo,Aldo Marzullo, “A Lumen Segmentation Method in Ureteroscopy Images based on a Deep Residual U-Net architecture”by arXiv:2101.05021v1 [eess.IV] ,May 6,2021.

18. Anushri Parakh,et.al."Urinary Stone Detection on CT Images Using Deep Convolutional Neural Networks: Evaluation of Model Performance and Generalization" by Radiology Artificial Intelligence Volume 1: Number 4,pp:1-10, August 6,2019.

19. S.Suganyadevi,dr.M Renukadevi, Segmentation of kidney stone region in ultrasound image by using region partition and mounting segmentation algorithm (RPM) by IT industry in vol 9,no 1,pp:512-515,2021.

20.S. Selvarani, P. Rajendran, Detection of Renal Calculi in Ultrasound Image Using Meta-Heuristic Support Vector Machine" by Journal of Medical system,pp:3-9, 31 july 2019.