A REPORT ON CLASSIFICATION AND DETECTION OF OSTEOARTHRITIS IN KNEE MRI IMAGES USING DEEP LEARNING

Major Project-II (CS891) progress report submitted to

Central Institute of Technology Kokrajhar

in partial fulfilment for the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

Anil Kumar Chaudhary (Gau-C-15/054)
Chandan Kumar Patel (Gau-C-15/066)
Manisha Debnath (Gau-C-15/073)

Under the supervision of

Dr. Pankaj Pratap Singh

Department of Computer Science and Engineering

Central Institute of Technology Kokrajhar

May 10, 2019

DECLARATION

We certify that

(a) The work contained in this report has been done by us under the guidance of our supervisor.

(b) The work has not been submitted to any other Institute for any degree or diploma.

(c) We have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

(d) Whenever we have used materials (data, theoretical analysis, figures, and text) from other sources, we have given due credit to them by citing them in the text of the thesis and giving their details in the references. Further, we have taken permission from the copyright owners of the sources, whenever necessary.

Anil Kr. Chaudhary.
Date: May 10, 2019 Anil Kumar Chaudhary (Gau-C-15/054)

Chandan Kumar Patel
Place: Kokrajhar Chandan Kumar Patel (Gau-C-15/066)

Manisha Debnath.
Manisha Debnath (Gau-C-15/073)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CENTRAL INSTITUTE OF TECHNOLOGY KOKRAJHAR

KOKRAJHAR, ASSAM - 783370

CERTIFICATE

This is to certify that the project report entitled "**Classification and Detection of Osteoarthritis in knee MRI images using Deep Learning**" submitted by **Anil Kumar Chaudhary(Gau-C-15/054), Chandan Kumar Patel(Gau-C-15/066), Manisha Debnath(Gau-C-15/073)** towards partial fulfilment of requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering is a record of bonafide work carried out by them under my supervision and guidance during year 2018-19.

Dr. Pankaj Pratap Singh
Department of Computer Science and Engineering
Central Institute of Technology Kokrajhar
Kokrajhar, Assam - 783370

Date: May 10, 2019
Place: Kokrajhar

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CENTRAL INSTITUTE OF TECHNOLOGY KOKRAJHAR

KOKRAJHAR, ASSAM - 783370

BONAFIDE CERTIFICATE

This is to certify that the project titled "**A REPORT ON CLASSIFICATION AND DETECTION OF OSTEOARTHRITIS IN KNEE MRI IMAGES USING DEEP LEARNING**" is a bonafide record of the work done by **Anil Kumar Chaudhary (Gau-C-15/054), Chandan Kumar Patel (Gau-C-15/066), Manisha Debnath (Gau-C-15/073)** towards partial fulfilment of requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering of **CENTRAL INSTITUTE OF TECHNOLOGY, KOKRAJHAR**, during the year 2018-19.

14/05/19

Mr. Ranjan Maity
Assistant Professor
Project Coordinator

Dr. Pankaj Pratap Singh
Head of the Department

Examiner______________

Abstract

Names of the students: Roll No:

Anil Kumar Chaudhary **(Gau-C-15/054)**

Chandan Kumar Patel **(Gau-C-15/066)**

Manisha Debnath **(Gau-C-15/073)**

Degree for which submitted: **Bachelor of Technology**

Department: **Department of Computer Science and Engineering**

Thesis title: **A report on Classification and Detection of Osteoarthritis in knee MRI images using Deep Learning**

Thesis supervisor: **Dr. Pankaj Pratap Singh**

Month and year of thesis submission: **May 10, 2019**

Osteoarthritis is one of the prime causes of infirmity in elderly and overweight people. Osteoarthritis is a joint disease that mostly affects the cartilage. Cartilage helps the easy glide of bones and obstructs them from rubbing each other. In Osteoarthritis cartilage is ruptured due to which bones rub each other causing severe pain. The current strategy for the evaluation of Osteoarthritis includes clinical investigation and medical imaging techniques. In this project, we want to detect and classify Osteoarthritis in knee from medical images using Deep Features to categorize the images. In this project, the problem of noises can impact on the detection and classification of target area in images and cause of it the irrelevant features could be selected from the medical images. This project will also focus on handling the huge amount of image data by utilizing some High Performance Computing. This paper also discovers the Magnetic Resonance imaging (MRI) techniques for detection and classification of Osteoarthritis in descriptive and comparative manner. Thus an integrated discussion of various detection techniques, feature extraction techniques and classification schemes regarding Osteoarthritis is done in a scientific way.

Keywords: Medical Imaging, MRI, Osteoarthritis, OA detection, classification, Deep Learning

Acknowledgements

We would like to thank the following people for their support and guidance without whom the completion of this project in fruition would not be possible.

Dr. Pankaj Pratap Singh (HoD, Computer Science and Engineering, Central Institute of Technology, Kokrajhar), our project guide, for helping us and guiding us in the course of this project .

Our departmental reviewers, **Dr. Hemanta Kumar Kalita**, **Mr. Ranjan Maity**, **Mr. Pranab Kumar Singh**, **Dr. Apurbalal Senapati**, **Mr. Sanjib Narzary**, **Mr. Dharani Kanta Roy**, **Mrs. Tania Sarkar**, **Mr. Prasanta Baruah**, **Mr. Mithun Karmakar** for their insight and advice provided during the review sessions.

We gratefully acknowledge the sources **National Institute of Health** (https://oai.nih.gov) and **Invectus Innovation Pvt. Ltd., Noida** for support in providing the MRI data for this project.

We would also like to express our gratitude to **Dr. H. Baishya, MD Radiodiagnosis, MRM, Kokrajhar** for his support in this project.

A REPORT ON CLASSIFICATION AND DETECTION OF OSTEOARTHRITIS IN KNEE MRI IMAGES USING DEEP LEARNING

Major Project-II (CS891) progress report submitted to

Central Institute of Technology Kokrajhar

in partial fulfilment for the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

by

Anil Kumar Chaudhary (Gau-C-15/054)
Chandan Kumar Patel (Gau-C-15/066)
Manisha Debnath (Gau-C-15/073)

Under the supervision of

Dr. Pankaj Pratap Singh

Department of Computer Science and Engineering

Central Institute of Technology Kokrajhar

May 10, 2019

DECLARATION

We certify that

(a) The work contained in this report has been done by us under the guidance of our supervisor.

(b) The work has not been submitted to any other Institute for any degree or diploma.

(c) We have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

(d) Whenever we have used materials (data, theoretical analysis, figures, and text) from other sources, we have given due credit to them by citing them in the text of the thesis and giving their details in the references. Further, we have taken permission from the copyright owners of the sources, whenever necessary.

Date: May 10, 2019

Place: Kokrajhar

Anil Kumar Chaudhary (Gau-C-15/054)

Chandan Kumar Patel (Gau-C-15/066)

Manisha Debnath (Gau-C-15/073)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CENTRAL INSTITUTE OF TECHNOLOGY KOKRAJHAR

KOKRAJHAR, ASSAM - 783370

CERTIFICATE

This is to certify that the project report entitled "**Classification and Detection of Osteoarthritis in knee MRI images using Deep Learning**" submitted by **Anil Kumar Chaudhary(Gau-C-15/054), Chandan Kumar Patel(Gau-C-15/066), Manisha Debnath(Gau-C-15/073)** towards partial fulfilment of requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering is a record of bonafide work carried out by them under my supervision and guidance during year 2018-19.

Date: May 10, 2019
Place: Kokrajhar

Dr. Pankaj Pratap Singh
Department of Computer Science and Engineering
Central Institute of Technology Kokrajhar
Kokrajhar, Assam - 783370

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CENTRAL INSTITUTE OF TECHNOLOGY KOKRAJHAR

KOKRAJHAR, ASSAM - 783370

BONAFIDE CERTIFICATE

This is to certify that the project titled "**A REPORT ON CLASSIFICATION AND DETECTION OF OSTEOARTHRITIS IN KNEE MRI IMAGES USING DEEP LEARNING**" is a bonafide record of the work done by **Anil Kumar Chaudhary (Gau-C-15/054), Chandan Kumar Patel (Gau-C-15/066), Manisha Debnath (Gau-C-15/073)** towards partial fulfilment of requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering of **CENTRAL INSTITUTE OF TECHNOLOGY, KOKRAJHAR,** during the year 2018-19.

Mr. Ranjan Maity	Dr. Pankaj Pratap Singh
Assistant Professor	Head of the Department
Project Coordinator	

Examiner_________________

Abstract

Names of the students: Roll No:

Anil Kumar Chaudhary **(Gau-C-15/054)**

Chandan Kumar Patel **(Gau-C-15/066)**

Manisha Debnath **(Gau-C-15/073)**

Degree for which submitted: **Bachelor of Technology**

Department: **Department of Computer Science and Engineering**

Thesis title: **A report on Classification and Detection of Osteoarthritis in knee MRI images using Deep Learning**

Thesis supervisor: **Dr. Pankaj Pratap Singh**

Month and year of thesis submission: **May 10, 2019**

Osteoarthritis is one of the prime causes of infirmity in elderly and overweight people. Osteoarthritis is a joint disease that mostly affects the cartilage. Cartilage helps the easy glide of bones and obstructs them from rubbing each other. In Osteoarthritis cartilage is ruptured due to which bones rub each other causing severe pain. The current strategy for the evaluation of Osteoarthritis includes clinical investigation and medical imaging techniques. In this project, we want to detect and classify Osteoarthritis in knee from medical images using Deep Features to categorize the images. In this project, the problem of noises can impact on the detection and classification of target area in images and cause of it the irrelevant features could be selected from the medical images. This project will also focus on handling the huge amount of image data by utilizing some High Performance Computing. This paper also discovers the Magnetic Resonance imaging (MRI) techniques for detection and classification of Osteoarthritis in descriptive and comparative manner. Thus an integrated discussion of various detection techniques, feature extraction techniques and classification schemes regarding Osteoarthritis is done in a scientific way.

Keywords: Medical Imaging, MRI, Osteoarthritis, OA detection, classification, Deep Learning

Acknowledgements

We would like to thank the following people for their support and guidance without whom the completion of this project in fruition would not be possible.

Dr. Pankaj Pratap Singh (HoD, Computer Science and Engineering, Central Institute of Technology, Kokrajhar), our project guide, for helping us and guiding us in the course of this project .

Our departmental reviewers, **Dr. Hemanta Kumar Kalita**, **Mr. Ranjan Maity**, **Mr. Pranab Kumar Singh**, **Dr. Apurbalal Senapati**, **Mr. Sanjib Narzary**, **Mr. Dharani Kanta Roy**, **Mrs. Tania Sarkar**, **Mr. Prasanta Baruah**, **Mr. Mithun Karmakar** for their insight and advice provided during the review sessions.

We gratefully acknowledge the sources **National Institute of Health** (https://oai.nih.gov) and **Invectus Innovation Pvt. Ltd., Noida** for support in providing the MRI data for this project.

We would also like to express our gratitude to **Dr. H. Baishya, MD Radiodiagnosis, MRM, Kokrajhar** for his support in this project.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Recent advances in artificial intelligence have led to fully automated work-flows that often exceed human performance. State of the art neural networks can detect the objects in the images and classify them into thousands of categories more accurately and magnitudes faster than humans[3]. They translate texts from multiple languages[4], drive cars autonomously through cities[5], and detect malware in computer systems[6]. In most of these cases, they have been trained on tens of thousands, or even millions, of data samples. Neural networks have also found great success in the field of medical image analysis where data sets are often much smaller. Although the same techniques can be applied, one is often confronted with a different set of challenges.

Medical imaging is the process of creating visual representations of the internal structures hidden by the skin and bones. It is the technique where we can reveal the interior of the body for clinical diagnosis and medical intervention. It is the part of biological imaging and incorporates radiology which uses the imaging technologies of X-ray, Magnetic Resonance Imaging (MRI), Ultrasound, Computed Tomography (CT) etc. Osteoarthritis is one of the most common form of arthritis disease that is seen mostly in females, overweight and elderly people. Osteoarthritis (OA) is a joint disease that mostly affects cartilage. Cartilage is the protective connective tissue that covers the end of bones in a joint. Healthy cartilage allows easy glide of bone in the joint and prevents them from rubbing each other. In Osteoarthritis the top layer of cartilage breaks down and wears away. This allows the bones to rub each other causing pain. It commonly affects the joint in the knee, hip, spine and feet. The most common cause of osteoarthritis of the knee is age. There are two types of OA, Primary OA seen in aged people due to genetic reasons or aging. Secondary OA tends to show up earlier in life due to some injury, diabetes, obesity, athletics or patients with rheumatoid arthritis. The sample of normal and affected Osteoarthritis knee image is shown in Figure 1.1.

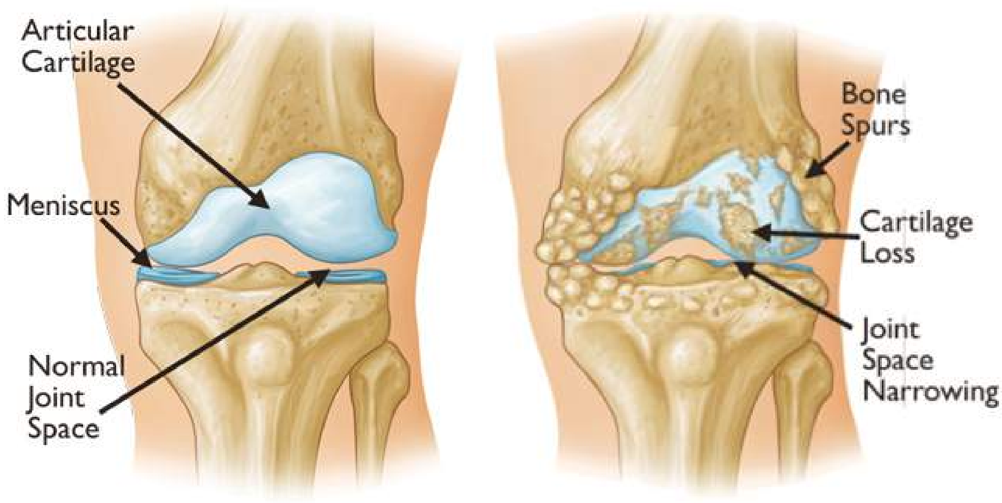

FIGURE 1.1: Sample of Normal and Osteoarthritis Knee

The main symptoms of OA are pain and difficulty in joint motion, reduced function and participation restriction, Joint stiffness in the morning or after rest. Currently evaluation of OA is based on clinical examination, symptoms and simple radiographic assessment techniques (X-ray), MRI, CT etc. While several other methods have been proposed, Kellgren-Lawrence (KL) system is validated method of classifying individual joints into 5 grades. Table 1.1 below shows the different grades of OA disease.

KL Grades	OA Analysis
Grade 0	No Radio-graphic features of OA present
Grade 1	Doubtful OA (narrowing of joint space)
Grade 2	Mild OA (definite narrowing of joint space)
Grade 3	Moderate OA (multiple osteophytes, sclerosis)
Grade 4	Severe OA (large osteophytes, sever sclerosis, bone deformity)

TABLE 1.1: Different grades of OA disease

The common X-ray findings of OA include destruction of joint cartilage; joint space is diminished between adjoining bones and bone spur formation. MRI scans may be ordered when x- rays do not give clear reason for joint pain or when the x- ray suggests that other type of joint tissues can be damaged. The current methods used for clinical diagnosis of Osteoarthritis are not accurate enough to efficiently measure the quality and evolution of Osteoarthritis. Thus we require more significant methods and algorithms which are multifactoral to access the parameters and progression of Osteoarthritis.

Detection and Classification of Osteoarthritis in knee from medical images is one of the active fields in computer vision. Image classification is one of the most commonly studied problems in computer vision. The goal of this field is detecting all the objects of a given image. Since the Convolution Neural Networks (CNN) can detect the objects with more than 90% of accuracy, we can use a fine-tuned neural network to detect an object. In the respective for above-mentioned issues, this work will implement Deep Features techniques to detect and classify the Osteoarthritis from medical images and also incorporate a large database for better accuracy. This project focuses on the various machine learning techniques for the classification of OA disease from the medical images.

Musculoskeletal diseases and articular disorders are one of the major health problems in recent years and affect especially the aging population. The human knee joint is commonly affected by osteoarthritis (OA), a degenerative disease that is the primary cause of chronic disability. It is a highly prevalent joint disease, causing pain and disability in older people, and it can be characterized by progressive degradation of the di-arthrodial joint tissue. OA can affect the joints of the spine, fingers, thumbs, hips, knees, and toes as depicted in Figure 1.2, but it is most prevalent in knee and hip joints. Detection and progress monitoring of knee OA can be done by measuring biochemical and anatomical changes associated with the tissues such as articular cartilage, ligaments, meniscus, synovial fluid, and subchondral bones.

FIGURE 1.2: Locations of various human body joints affected by osteoarthritis

An early detection and monitoring of osteoarthritis is possible by measuring pre-structural and structural changes associated with the tissues such as articular cartilage, meniscus, synovial fluid, and subchondral bones. Soft tissues within the di-arthrodial joint have been successfully used to detect and monitor knee OA, changes associated with the subchondral bones are a promising imaging biomarker for assessing knee OA conditions.

Figure 1.3 The worldwide prevalence estimate for symptomatic OA is 9.6 % among men and 18 % among women. In addition, progression of early OA can take place within 10 years of a major injury. In case of injury at age 15, young people may have OA as early as age 25.

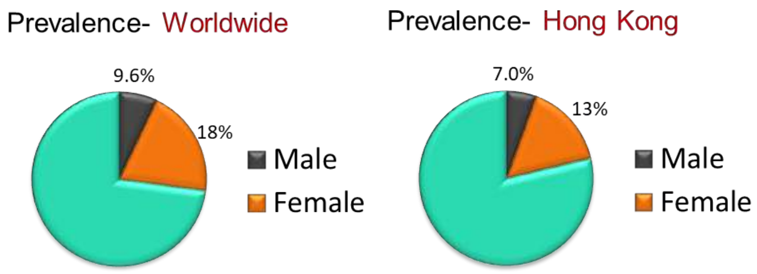

FIGURE 1.3: Prevalence of Osteoarthritis Worldwide and Hong Kong

According to the studies by World Health Organization for the year 2000, knee OA is most prevalent in developing countries in Europe (EU BC) (13.3%), Eastern Mediterranean Region (30% aged above 60) and South-East Asia Region (SEA) (7.9%) for the age group between 30 – 80 years. In Hong Kong, prevalence of osteoarthritis is showing similar trend as other part of world. A study in 2000 reported that among Hong Kong people aged 50 and above, 7 percent of men and 13 percent of women suffered from osteoarthritis.

1.1 Problem Definition

Object detection is a general term for computer vision techniques for locating and labeling objects. Object detection techniques can be applied both to static images or moving images. Computer vision techniques are already in wide use in medical areas today, as they can offer valuable insight about various diseases to detect efficiently. In this project we will be solving an object detection and image classification problem, where the goal will be to detect the OA affected area in the OA knee MRI images. Detection is the process of finding out the particular objects in the images, in this case finding out the OA affected area from knee MRI images. Whereas, classification is a process of classifying whether the output image is OA affected or Normal.

Object detection is widely used in the world of medical to detect the diseases. Neural network based classifiers are used together with other object detection techniques. During this project we have explored the modern open source based solutions for object detection in medicals, in this case for detecting OA disease. TensorFlow Object Detection API[7][1][2][8], an open source framework for object detection related tasks, was used for training and testing an SSD (Single-Shot Multibox Detector) with Mobilenet- model. The model was tested as pre-trained and with fine-tuning with a dataset consisting of OA MRI images. The sample detected OA region in knee MRI is shown in Figure 1.4.

FIGURE 1.4: Sample of detected OA region in knee MRI

Open source library such as TensorFlow Object Detection API offers easy-to-use, open source frameworks where pre-trained models for object detection (such as ones downloadable from the TensorFlow model zoo) reach high accuracy in detecting various object from humans to tv monitors. However, the dataset of MRI images used in this project made it important to train the models to work with the specific data. For this project, TensorFlow Object Detection API was chosen due to its popularity in object detection and for its easy-to-approach nature. In this project, TensorFlow Object Detection API was tested for detection of OA disease. The model used within the API is a SSD model with Mobilenet. The pretrained model is trained and tested with our own data which consisted of MRI images of knee OA.

1.1.1 Our Previous Work

In the previous semester, we worked on an image classification problem, where the goal was to tell which class the input image belongs to. Classification is a process of classifying whether the output image is OA affected or Normal. The way we achieved it is by training an artificial neural network on few thousand images of normal knee MRI and OA affected MRI and made the NN(Neural Network) learn to predict which class the input image belongs to, next time it seen an image having a normal knee MRI or OA affected MRI in it. In the previous work we developed a fully automated non-contrast MRI application for classification and detection of knee Osteoarthritis. The approach we used in the previous work (for classification problem) is based on Convolution Neural Network. Convolutional neural networks is the base tool for such kind of study because they have been setting state of the art results in a vast majority of image segmentation and classification tasks. CNN to achieve a robust and accurate classification of even knee MRI, where we used Deep learning approach based on U-net Architecture and for implementing and performing this approach we used Keras framework in python Language. The sample of normal and affected Osteoarthritis knee MRI is shown in Figure 1.5.

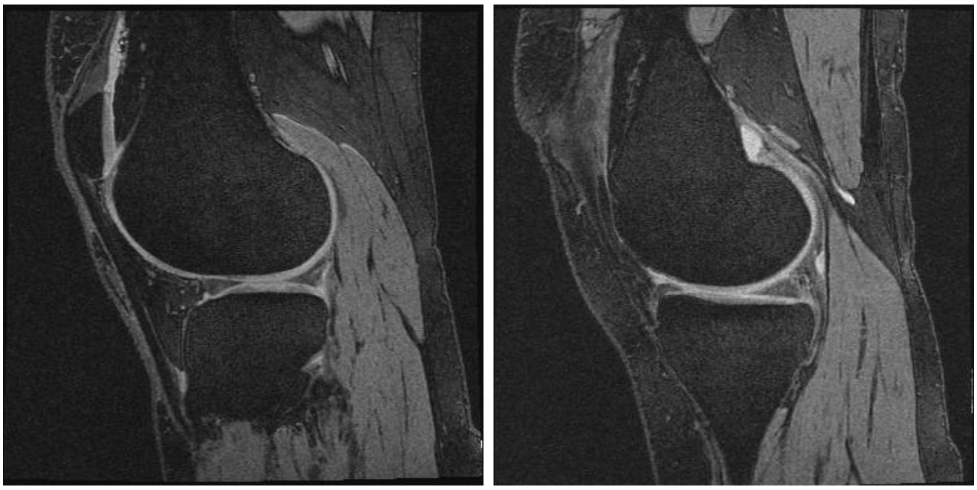

FIGURE 1.5: Sample of Normal and Osteoarthritis Knee MRI

1.1.2 Magnetic Resonance Imaging

Magnetic Resonance Imaging (MRI)[9] uses magnetic fields and radio frequen-cies to probe tissue structure. In contrast to X-ray and CT which require the exposure of ionizing radiation, MRI is a noninvasive imaging method. As such it has become an essential diagnostic imaging modality in the medical field[10][9].

96% of the human body is made up of hydrogen, oxygen, carbon, and nitrogen, all of which are referred to as MR active. These elements have an odd atomic mass number giving the nucleus a spin. Due to the laws of electromagnetic induction, the motion of unbalanced charge produces a magnetic field around itself. Hydrogen is the element used in MRI because its solitary proton in the nucleus gives it a relatively large magnetic moment[10].

The positively charged hydrogen particles in water produce a signal when exposed to a strong external magnetic field. This field is supplied by a magnet in the MRI machine aligning the magnetic field of the hydrogen atoms to its own. Gradient coils are used to cause a linear change in the strength of this field. By alternating the current of these coils on the x, y and z axes, it is possible to calculate a three-dimensional image of the tissue[10].

1.1.3 Osteoarthritis

The development of OA in the knee is due to a heterogeneous group of conditions involving all the components in diarthrodial joint but mainly associated with defective integrity of articular cartilage (AC), meniscus damages and related changes in the underlying bone at the joint margins. An early detection and monitoring of osteoarthritis is possible by measuring pre-structural and structural changes associated with the tissues such as articular cartilage, meniscus, synovial fluid, and subchondral bones.[11] In the assessment of osteoarthritis, pre-structural/biochemical compositions (water content, proteoglycan content and collagen) are mainly measured in the articular cartilage tissue to diagnose the disease at early stages followed by the measurement of anatomical (thickness, volume, surface area) changes that occurs either at early or during the progression stages. Similarly, quantitative MR of meniscus has been used to measure biochemical (collagen-PG) changes during the early onset of osteoarthritis followed by its morphological (flap, or complex tears; meniscal maceration; or destruction) changes to monitor progression. Although soft tissues within the diarthrodial joint have been successfully used to detect and monitor knee OA, changes associated with the subchondral bones are a promising imaging biomarker for assessing knee OA condition. Researchers have demonstrated that the changes in subchondral bone shape morphometry, osteophytes, surface area, and bone marrow lesions are important to assess OA progression, and they can effectively be measured using magnetic resonance (MR) imaging[12].

Quantitative MRI can be used to diagnose, assess, and monitor diseases such as osteoarthritis by identifying morphologic changes in the knee and calculating quantitative values such as T1 rho, T1/T2 relaxation times. However, the clinical application of quantitative MRI has been hindered by the requirement for time-consuming post-processing of images, particularly for segmentation of the joint and musculoskeletal tissue. Generally, measurement of physiological changes in AC using quantitative MR techniques requires segmentation of cartilage region followed by generating a technique specified map which is being overlaid to measure specific values in knee tissue. In most cases, the segmentation of AC (either the whole or different compartments) is performed either by an expert or using semi-automated methods which are tedious and time-consuming that may also lead to reliability issues. If it's done manually, it can take a skilled technician around three hours to do an image, depending on the quality of the image and what actually is being segmented. Thus, segmentation of knee joint tissues such as cartilage, bone and meniscus from medical images is an important perspective to researchers and clinicians for the development of biomarkers, knee implant procedures, modeling of the knee to predict kinematics and understanding the physiological phenomenon in healthy knee joint tissues. As a result, the team sought to utilize advanced computation approaches including image/pattern analysis techniques in combination with deep learning to develop a fully automatic segmentation technique for bone, cartilage and meniscus. The developed technique would accurately extract quantitative measurements in terms of cartilage volume and thickness, bone area/shape & attrition, meniscus damages as well as biochemical compositions by estimating Spin-Lock relaxation times on knee MRI.

1.2 Existing Method

There has been a lot work in detection and classification of knee Osteoarthritis X-ray images using image processing techniques[13] and traditional computer vision techniques. However, there proposed approach was lacking the accuracy in compare to Deep Learning based techniques. Among the deep learning techniques, a broad class of method i.e., two stage detection (CNN) which acts as feature extractor and various other techniques for image detection and classification. The current strategy for the detection and classification of Osteoarthritis from medical images includes medical expertise verification & medical imaging techniques. There has been a lot work in object detection and classification using traditional computer vision techniques (sliding windows, deformable part models). As per the preceding survey, researchers have investigated various methods on the classification and detection of Osteoarthritis using knee images like X-ray.

1.3 Summary

This project report consists of six chapters. The first chapter reviews the introduction, problem statement, our previous semester work and existing methods related to this project. The second chapter discusses the state of art on related works. Chapter three discusses some background study on deep learning and also the theories of modern methods in object detection. Chapter four discusses the method and tool that we have used in this project. Results and discussion will be discussed in chapter five and lastly chapter six includes the conclusion and future scope of this work.

Chapter 2

Literature Review

This chapter includes the state of art by various researchers and defines the problem of this project and also provides an overview of prerequisites that led to the main task of this project. It features a section for the necessary medical knowledge as well as discusses the theories and some background study on modern methods in object detection[14].

2.1 State of Art

As per the preceding survey[15], researchers have investigated various methods on the detection and classification of Osteoarthritis using different knee images like X-ray, MRI etc. Image segmentation method has a great importance in most medical imaging applications. In medical imaging the segmentation methods are classified into two basic groups: pixel based (includes thresholding, region growing region merging etc) and geometry based (includes deformable models, active contours active appearance models). Therefore to know the proper progression & extremity of the disease medical imaging is carried out in terms of X-ray imaging or Magnetic Resonance imaging. As per the survey,

Lior Shamir et al.[15], have used joint detection algorithm and feature extraction technique to analyze the knee X-ray image. The algorithm finds the joint and separates it from rest of the images. The features were computed by Zernike features, Multi-scale histograms, first four moments, Tamura texture features etc. The features extracted were classified using weighted Nearest Neighbor rule. The authors concluded that 95% of moderate OA was differentiated from normal and 80% of minimal OA was differentiated from normal.

M.Subramonium et al.[15], have used edge detection algorithm, to detect edges of Knee X-ray images in osteoarthritis. The features computed were femur, tibia and patella cartilage. The authors have concluded that the proposed algorithm gives sufficiently good results and is very effective in noisy and blur images.

Samir K.Bandyopadhyay et al.[15], have used Local Binary Pattern (LBP) based classification system for the detection of OA using Knee X-ray images. The classification is achieved by computing the histograms of LBP of knee X-ray images using K- Nearest Neighbor classifier. The authors have concluded with 95.24% of accuracy in detecting normal or abnormal knee image and 97.37% of accuracy in detecting medium or worst cases.

Prafull Sharma et al.[15],have used different Image segmentation algorithms on X-ray bone images. The discrete step, Watershed segmentation and Otsu's segmentation are used to analyze the abnormalities and problems associated with bone structures. The authors have concluded that discrete step algorithm provides quick and efficient results.

Bindushree R et al.[15],have used different image processing techniques to measure the joint space width in knee x-ray images. Different techniques used are contrast enhancement, histogram equalization, canny edge detection algorithm and thresholding for extraction & computation of features. The authors have concluded that the Joint Space Width of the knee x-ray image is compared with the standard Width (4.8 for women and 5.7 for men) and that image is said to be normal case or osteoarthritis case.

Subromoniam M et al.[15],have used computer aided diagnosis for the detection of OA using X-ray images. Haralick feature extraction technique for computation & SVM classifier with the kernel functions are used for the detection & classification of OA. They have concluded that the algorithm had a sufficient good result with accuracy of 99% in the diagnosis of bone disorders caused by OA.

Dian Pratiwi ,have used artificial neural back propagation method for measuring the severity of Osteoarthritis disease[15]. In their work the whole processing is divided into three steps, image processing, feature extraction and classification using artificial neural network process. They have concluded with 66.6% of classification rate.

Jessie Thomson et al.[15], features, simple pixel features etc. The experiment showed that combining shape & texture based classifiers resulted in 84.9% of accuracy as compared to shape alone.

Tati L.Mengko et al.[15], have used machine vision system for osteoarthritis assessment. In their proposed method image segmentation uses edge detection method to determine the region of joint space. The feature computed is the distance between femur & tibia bone. The classification of normal & affected OA is obtained from radiographic image using neural network that is later examined with the predefined diagnosis managed by the physician. They concluded with 50% sensitivity, 100% specificity & positive predictive value & 91.84% negative predictive value.

From the survey[15] it is observed that a good number of researchers have worked on Knee X-ray imaging for detection and classification of OA using different approaches and forecasted some promising results with their own datasets. But still there is no such research work done on Knee MRI (Magnetic Resonance Imaging). There is also a scope for using such robust algorithm using different parameters on Knee MRI as MRI, also known as nuclear magnetic resonance imaging, is a scanning technique that uses strong magnetic fields and radio waves to generate a detailed image of the body's soft tissue and bones. Different robust features are required to be extracted from Knee MRI also for the detection of Osteoarthritis. The comparative analysis with promising results were made on Knee x-ray by researchers as predicted and discussed in the related work. But still these algorithms & methods are to be applied using Knee MRI.

For the classification and detection of Knee Osteoarthritis, the segmentation of knee cartilage and the surrounding bones is a problem which has gained considerable importance in recent years. Segmentation of knee joint tissues such as cartilage, bone and meniscus from medical images is an important perspective to researchers and clinicians for the development of biomarkers, knee implant procedures, modeling of the knee to predict kinematics and understanding the physiological phenomenon in healthy knee joint tissues. Segmentations of bones are required for computer-based surgical planning of knee implants. Other applications include modeling of the knee by finite elements to predict joint kinematics or the understanding of natural variation and physiological effects for healthy joints.

Neural networks are known to be feature selectors, meaning that they will learn to extract information that is relevant to the task. This assumes that the size of the data set and the complexity of the problem enable the network to find correlating features. Based on a series of tests, it was not possible to create a stable algorithm that would predict the OA of an individual based on their knee MRI and hence the classification between normal Knee MRI and OA Knee would also be predicted.

2.2 Summary

In this chapter we have described the survey of various researcher's works, and studied the related research problems. The background study related to this project and an overview of prerequisites that led to the main task of this project is described in chapter three. It features a section for the necessary medical knowledge as well as a hierarchical derivation to the appropriate machine learning methods. However, the methodology that we have used in doing this project is described in chapter four.

Chapter 3

Background Study

This chapter discusses the theories of neural network and some background study on deep learning[14] for image classification and object detection.

3.1 Classification

Classification is a method to extract information from data sets. This is done by dividing the data into categories based on some features. The idea is to derive a model which can perform the sorting process by training it on data objects where the category, or label, is known. The model should then be able to classify unlabeled data with sufficient accuracy. There are many different models that are used for classification, e.g. neural networks.

3.2 Machine Learning Methods

The concept of classical programming is that an engineer defines a set of rules, called an algorithm, as shown in Figure 3.1 which uses input data to calculate some form of output data.

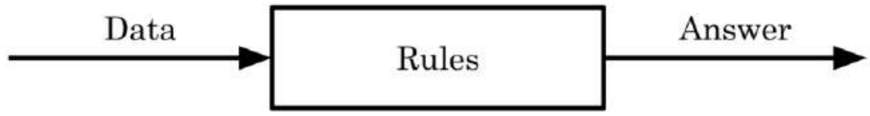

FIGURE 3.1: Classical Programming Pipeline

A machine learning algorithm is an algorithm that can learn from data (shown in Figure 3.2). It can be used to calculate these rules automatically, so they do not have to be specified by hand. Three components are needed for such an approach:

- Input data the algorithm is supposed to transform
- Output data the algorithm is supposed to predict
- A measurement to validate the performance of a prediction

It works by feeding input and output data into a pipeline, which will learn to transform one into the other. With the advantage that no explicit programming is needed to generate the rules, comes the disadvantage that prior input and output data is required for the initial learning process. Machine learning may be applied as an

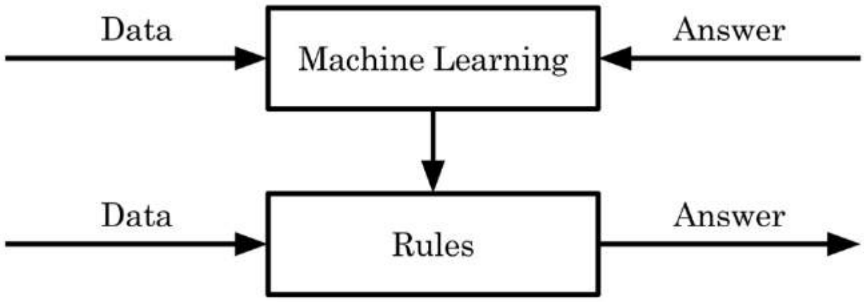

FIGURE 3.2: Machine Learning Pipeline

effective method if it is not feasible or possible to define an algorithm by hand and sufficient data is available for training. How much "sufficient" is depends on factors like the type of task, the complexity of the data, the uniformity of the data, the type of machine learning algorithm and others. There are different subparts to machine learning like supervised and unsupervised learning. Supervised learning is used when it is clear what the output data looks like, whereas unsupervised learning can help to find unknown patterns in the data. Examples of supervised learning techniques include linear regression, naive Bayes, support vector machines, decision trees, random forests, gradient boosting and artificial neural networks (ANNs).

3.2.1 Artificial Neural Networks

Machine learning is a field in computer science aiming to imitate the human learning process. Artificial neural networks, or just neural networks, is a kind of machine learning technique where the structure of the human brain is the inspiration.

The artificial neural network (ANN) is a network built of a number of interconnected neurons. The neurons are simple processing units that change their internal state, or activation, based on the current input and produces an output that depends on both the input and current activation. Such a neuron and its biological counterpart is shown in Figure 3.3. The ANN is constructed by having a large number of these neurons working in parallel and connecting some neurons to others through weighted connections, creating a weighted and directed network of different layers. It is by adjusting these weighted connections and the internal activations of the neurons the ANN can be improved, or trained. Usually the network cycle through a set of training data sufficiently many times, until the weights have been adjusted enough to produce the desired output. A common way of learning for a neural network is the process of back propagation. This is a method of dynamic adjustment based on providing feedback to the network, initiated from a difference between the desired output and the current output. The weights of the interconnected neurons are adjusted depending on the degree they contribute to the error and this process is repeated in cycles until the network achieves a desired accuracy of classification.

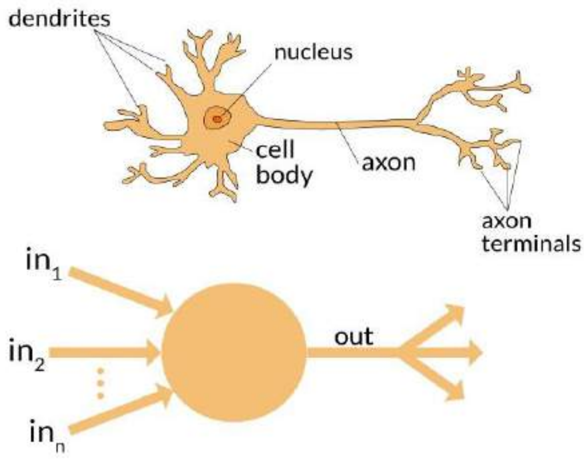

FIGURE 3.3: A comparison between a biological and an artificial neuron

3.2.2 Convolution Neural Networks

CNNs have wide applications in image and video recognition, recommender systems and natural language processing. Convolutions are frequently used in image processing, which is also why they were introduced to visual tasks in the field of deep learning. They allow learning local patterns in the data instead of treating the input features in a global manner like dense layers do. Convolutional neural networks (CNNs) are a specific type of ANN that uses an operation called convolution in at least one of their layers. The first CNN was introduced by Yann LeCunn in 1990 at which time its popularity was limited.

A convolution is a mathematical operation on two functions of real-valued argument. In imaging terminology, the first function refers to the input and the second function describes the kernel. The output of this operation is called a feature map. CNN models stand for one of the oldest deep neural networks hierarchy that have hidden layers among the general layers to help the weights of the system learn more about the features found in the input image. Another type of layer that is frequently used in CNN architectures perform a pooling operation. By doing so, the spatial resolution is reduced, and only the most relevant features are kept. This is important to maintain a manageable network size.

A general CNN architecture looks like the one shown in Figure 3.4 and consist of

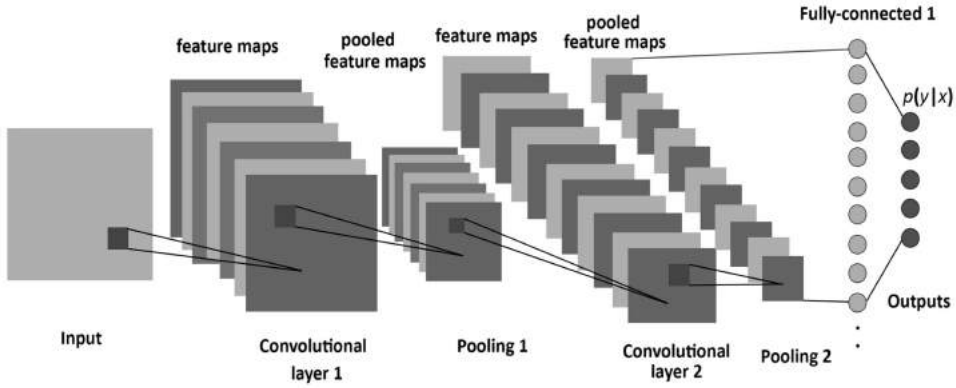

FIGURE 3.4: A General CNN layer hierarchy

distinct types of layers. The process of building a Convolutional Neural Network always involves four major steps.

Step - 1 : Convolution

Step - 2 : Pooling

Step - 3 : Flattening

Step - 4 : Full connection

The basic foundation of every Convolutional Neural Network is made up of these operations, so to develop a sound understanding of the working of these ConvNets, we need to comprehend thoroughly the working of these operations.

3.2.2.1 Representation of an Image:

An Image is classified as a matrix of pixel values. Any image can be represented as a matrix made up of pixel values. Any component of an image is conventionally referred to as a Channel. Images obtained from a standard digital camera will consist of 3 channels – green, blue and red. These can be pictured as 3 two-dimensional matrices, one for each color, with pixel values between 0 to 255 and stacked over one another. In this project we will use only grayscale images so as to have a single 2D matrix depicting an image. Pixel value range for the matrix is between 0 to 255, with 0 indicating black and 255 showcasing white.

3.2.2.2 The Convolution Step:

The name ConvNets had been derived from an operator called 'Convolution'. The primary objective of this operator is the extraction of input image features. Convolution learns image features and works in coordination with pixels by using small squares of input data.

Consider a 5 x 5 image with pixel values as 0 and 1 only as shown in Figure 3.5. Pixel values vary from 0 to 255 in grayscale images:

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

FIGURE 3.5: 5 x 5 image

Consider one more 3 x 3 matrix as shown below in Figure 3.6 :
Convolution of the 5 x 5 image and the 3 x 3 matrix can be calculated, for each position, we perform element-wise multiplication among the two matrices and summate the outputs of multiplication to obtain the final single element (an integer) of the output matrix. Also, note here that only a part of the input image is seen by the 3 x 3 matrix in every stride. The 3 x 3 matrix is generally called 'kernel' or 'filter' or a 'feature detector' in the CNN terminology. And the matrix obtained due to sliding of filter on the image and the computation of the dot product is referred as the 'Convolved Feature' or 'Activation Map' or the 'Feature Map'. Filters serve as feature detectors for the input image (original). Three parameters control the size of Convolved Feature. They are required to be set before performing the convolution step.

1	0	1
0	1	0
1	0	1

FIGURE 3.6: Convolved Feature

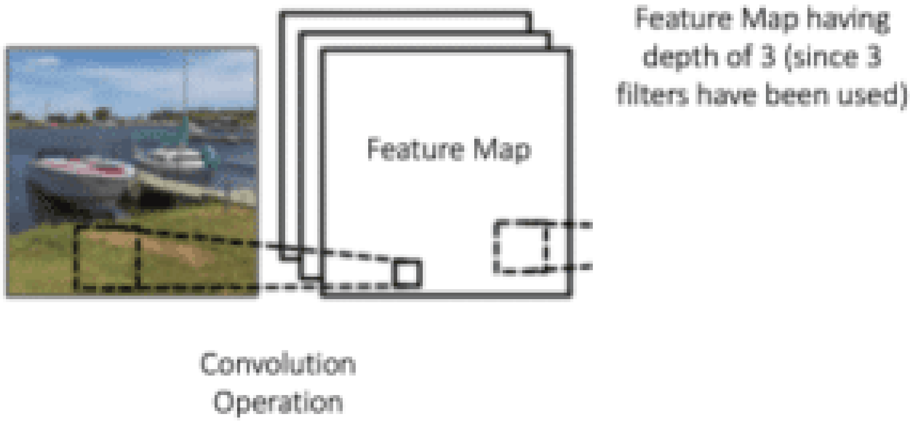

FIGURE 3.7: Convolution Operation

Depth: It defines the exact number of filters needed for the convolution operation. In the figure below, the convolution of the original boat image is performed using 3 distinct filters, thereby, generating 3 distinct feature maps. These feature maps can be thought of as stacked 2D matrices. So, the 'depth' of our feature map here would be 3.

Stride: It tells about the quantity of pixels needed to slide the filter matrix over the input matrix. A filter is moved one pixel at a time for a stride 1 and jumps by 2 pixels simultaneously at stride 2. However, note that using larger strides will result in small feature maps.

Zero-padding: It becomes necessary often to pad the input matrix using Zero around the borders, as that filter can be applied to bordering elements of input image matrix easily. The advantage of Zero-padding is that it permits monitoring of the feature map size. This phenomenon of adding zero-padding is known as 'wide convolution' and that of no zero-padding is 'narrow convolution'.

3.2.2.3 Introducing Non-Linearity (ReLU):

In a typical ConvNet, a supplementary operation known as ReLU is performed after every convolution operation. ReLU is the abbreviation for Rectified Linear Unit and to our surprise, is itself a non-linear operation. ReLU is an operation working element-wise i.e. is applied per pixel and substitutes every negative pixel value by 0 in the feature map. It serves the purpose of introducing non-linearity in ConvNet, because the maximum real-life data we will want to feed into our ConvNet is non-linear. (Convolution is a linear operation i.e. possesses element-wise matrix type multiplication and addition and so, there's a need for the introduction of a nonlinear function such as ReLU to account for non-linearity). Non-linear functions such as 'sigmoid' can be put to use for operation in the place of ReLU. However, ReLU is most widely used due to its consistently better performance in most situations.

3.2.2.4 The Pooling Step

If we wish to reduce the dimensionality of individual feature maps and yet sustain the crucial information, we use Spatial Pooling commonly known as downsampling or subsampling. This type of Pooling is of varied types: Average, Sum, Maximum, etc. For Max Pooling, we first specify a spatial neighborhood and then pick out the largest element of that feature rectified map within that window. Instead of largest element, if we pick the Average one it's called Average Pooling and when the summation of elements in the window is taken, we call it Sum Pooling. Max Pooling has proven itself as the best. The Figure 3.8 below shows a Pooling operation to every feature map separately and results 3 output maps from the 3 input maps.

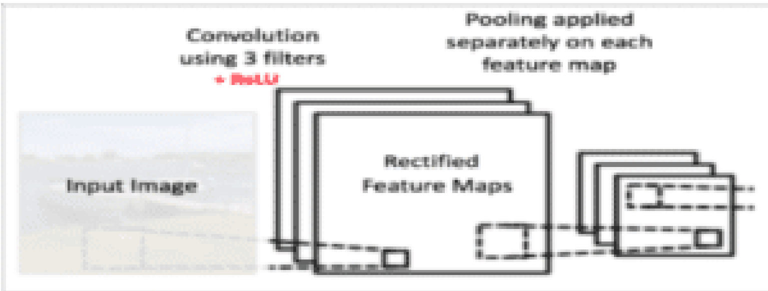

Figure 3.8: Pooling Operation

The basic purpose of pooling is to tremendously reduce the structural size of the input given. More specifically, the following are achieved through pooling:

- Generates input representations i.e. feature dimensions that are small and so, more manageable.

- Lessens the parameters used and the computations in the network to efficiently control overfitting.
- Makes the network resistant to distortions, translations in the input image and to small transformations. Though minute distortions in the input shall not distort the pooling output as average/maximum value is used in the local neighborhood.
- Helps in arriving at an equivariant representation of the input image. This is extremely useful as it enables us to detect small objects inside our image irrespective of where they are placed.

The working of Convolution, ReLU and Pooling till layers form the basic structure of any CNN. Generally, there are 2 sets of Convolution, ReLU and Pooling layers; convolution is performed on the output of the first Pooling layer by the second convolution layer employing six filters and so, producing six feature maps as well. ReLU layer is implemented on all these six feature maps individually. Having done this, we're left with the task of Max Pooling, which we do on each of the six rectified feature maps separately. Coordination among these layers helps in the extraction of useful features from the images fed, then it adds non-linearity in the network and cuts down feature dimension thereby, making the features equivariant to scale and translate. Input to the Fully Connected Layer is provided by the output of the second pooling layer.

3.2.2.5 Fully Connected Layer

This layer is described as a Multilayer Perceptron which utilizes a softmax activation function present in the output layer. The words "Fully Connected" in the fully connected layer indicate that every neuron in the next layer is connected to every individual neuron in the previous layer. Pooling and convolutional layer outputs depict high-resolution features of the input image. The fully connected layer on the basis of the training dataset utilizes these features for categorizing input images into different classes. A more general way of learning non-linear combinations of such features is by inserting a fully-connected layer instead of using classification. For the task of classification, features of the convolutional and pooling layers serve the purpose but using a combination of these is the best choice. Summation of all the output possibilities of the Fully Connected Layer comes as 1. To ensure this, we operate the Softmax activation function in the output layer. This function works on any arbitrary real valued vector and transforms it into a vector valued between one and zero, so as to acquire a sum of 1.

3.2.2.6 Softmax Function

Softmax function calculates the probabilities distribution of the event over 'n' different events. In general way of saying, this function will calculate the probabilities of each target class over all possible target classes. Later the calculated probabilities will be helpful for determining the target class for the given inputs.

3.3 Object Detection

An image classification or image recognition model simply detect the probability of an object in an image. In contrast to this, object localization refers to identifying the location of an object in the image. An object localization algorithm will output the coordinates of the location of an object with respect to the image. In computer vision, the most popular way to localize an object in an image is to represent its

FIGURE 3.9: Sample of bounding box representation

location with the help of bounding boxes. Figure 3.9 shows an example of a bounding box.

3.3.1 Modern Methods in Object Detection

Object detection is a common term for computer vision techniques classifying and locating objects in an image. Modern object detection is largely based on use of convolutional neural networks. Some of the most relevant system types today are Faster R-CNN, R-FCN, Multibox Single Shot Detector (SSD) and YOLO (You Only Look Once)[16][1] .

Original **R-CNN** method worked by running a neural net classifier on samples cropped from images using externally computed box proposals (samples cropped with externally computed box proposals; feature extraction done on all the cropped samples). This approach was computationally expensive due to many crops; **Fast R-CNN** reduced the computation by doing the feature extraction only once to the whole image and using cropping on the lower layer (feature extraction only once on the whole image; samples cropped with externally computed box proposals). **Faster R-CNN** goes a step further and used the extracted features to create class-agnostic box proposals (feature extraction only once on the whole image; no externally computed box proposals). **R-FCN** is like Faster R-CNN but the feature cropping is done in a different layer for increased efficiency.

YOLO (You Only Look Once) works on different principle than the aforementioned models: it runs a single convolutional network on the whole input image (once) to predict bounding boxes with confidence scores for each class simultaneously. The advantage of the simplicity of the approach is that the YOLO model is fast (compared to Faster R-CNN and SSD) and it learns a general representation of the objects. This increases localization error rate (also, YOLO does poorly with images with new aspect ratios or small object flocked together) but reduces false positive rate.

Single Shot Multibox Detector (SSD) differs from the R-CNN based approaches by not requiring a second stage per-proposal classification operation. This makes it fast enough for real-time detection applications. However, this comes with a price of reduced precision. Note that **"SSD with MobileNet"** refers to a model where model meta architecture is SSD and the feature extractor type is MobileNet.

3.3.2 Speed-Accuracy Tradeoff[1]

Many modern object detection applications require real-time speed. Methods such as YOLO or SSD work that fast, but this tends to come with a decrease in accuracy of predictions, whereas models such as Faster R-CNN achieve high accuracy but are more expensive to run. Overall mean average precision (mAP) vs CPU Time for some commonly used model types, including the model type in this project (SSD with Mobilenets) tested on low resolution images (red marker) is illustrated in Figure 3.10. The cost in model speed depends on the application: With larger images (e.g. 600x600) SSD works comparable to more computationally expensive models such Faster R-CNN, even as on smaller images its performance is considerably lower. Note that mAP metric is discussed in chapter four.

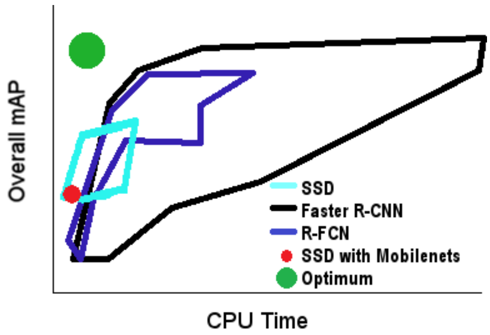

FIGURE 3.10: Comparison of different models with mAP vs CPU

3.3.3 Object Detection in Medical Images

Computer vision techniques are widely used in medicals today. Medical object detection is the task of identifying medical-based objects within an image. Applications include use in terms of detection and classification of various diseases (in our case Osteoarthritis). Medical image processing tools are playing an increasingly important role in assisting the clinicians in diagnosis, therapy planning and image-guided interventions. Accurate, robust and fast tracking of deformable anatomical objects such as the heart, is a crucial task in medical image analysis. One of the main challenges is to detect the actual location of the target area. To solve such challenges there has been a lot modern object detection methods are used in medical field. Now a days, various medical image analysis applications are demonstrated with focus on detection and classification of several diseases. The modern object detection methods can be implemented for different medical imagine methods also, like for Ultrasound (US), Cardiac Computed Tomography (CT), Magnetic Resonance Imaging (MRI), and X-ray fluoroscopy.

3.4 Object Detection Approach

The network used in this project is based on Single shot detection (SSD)[17]. The architecture is shown in Figure 3.11. The SSD normally starts with a VGG model, which is converted to a fully convolutional network. Then we attach some extra convolutional layers, that help to handle bigger objects. The output at the VGG network is a 38x38 feature map (conv4 3). The added layers produce 19x19, 10x10, 5x5, 3x3, 1x1 feature maps. All these feature maps are used for predicting bounding boxes at various scales (later layers responsible for larger objects). Thus the overall idea of SSD is shown in Figure 3.12. Some of the activations are passed to the sub-network that acts as a classifier and a localizer. Anchors[17] (collection of boxes overlaid on image at different spatial locations, scales and aspect ratios) act as reference points on ground truth images as shown in Figure 3.13. A model is trained to make two predictions for each anchor:

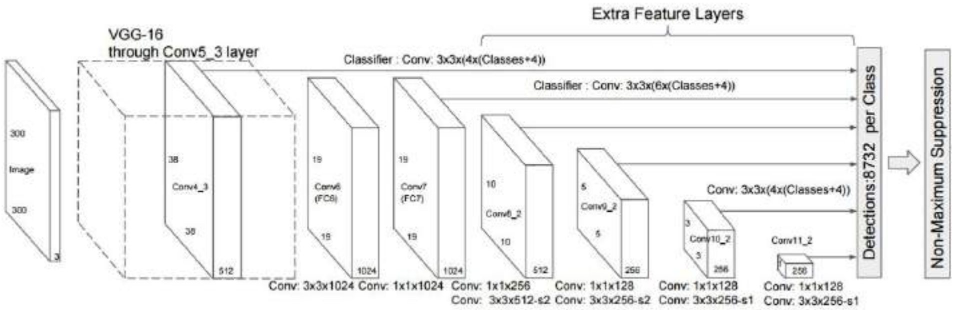

FIGURE 3.11: SSD Architecture

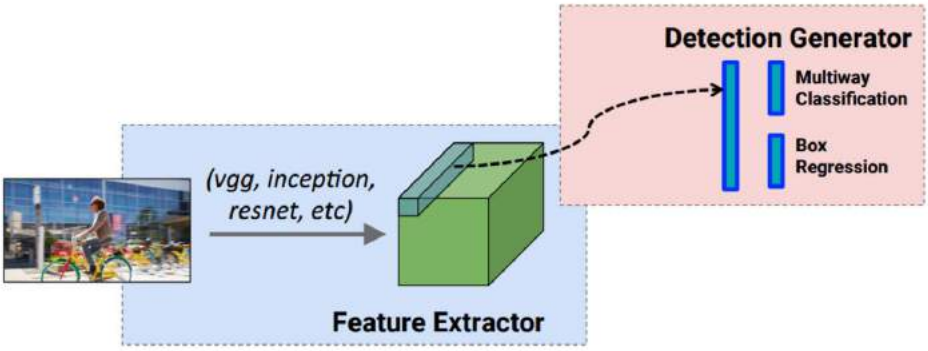

FIGURE 3.12: SSD Overall Idea

- A discrete class
- A continuous offset by which the anchor needs to be shifted to fit the ground-truth bounding box

During training SSD matches ground truth annotations with anchors. Each element of the feature map (cell) has a number of anchors associated with it. Any anchor with an IoU (jaccard distance) greater than 0.5 is considered a match. Consider the case as shown in Figure 3.14, where the cat has two anchors matched and the dog has one anchor matched. Note that both have been matched on different feature maps

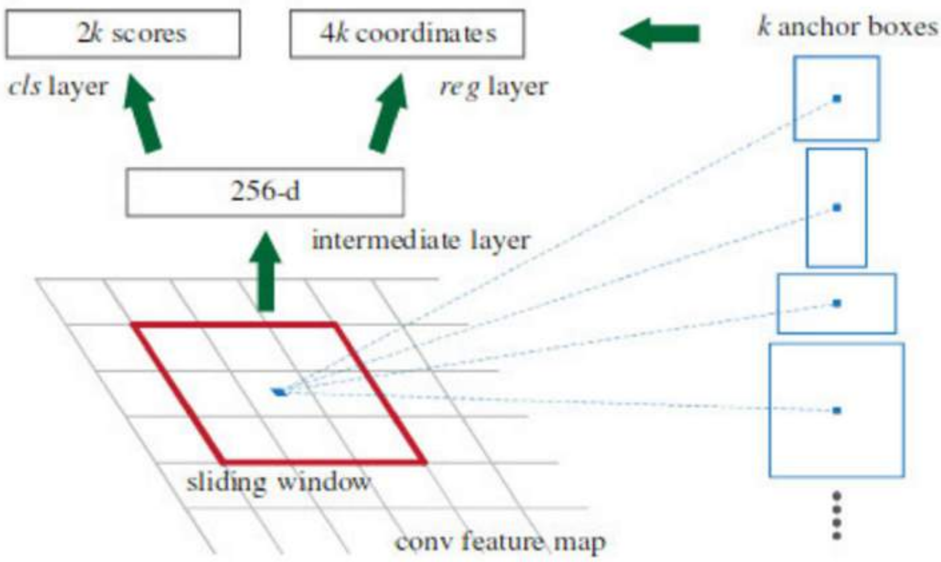

FIGURE 3.13: Anchors

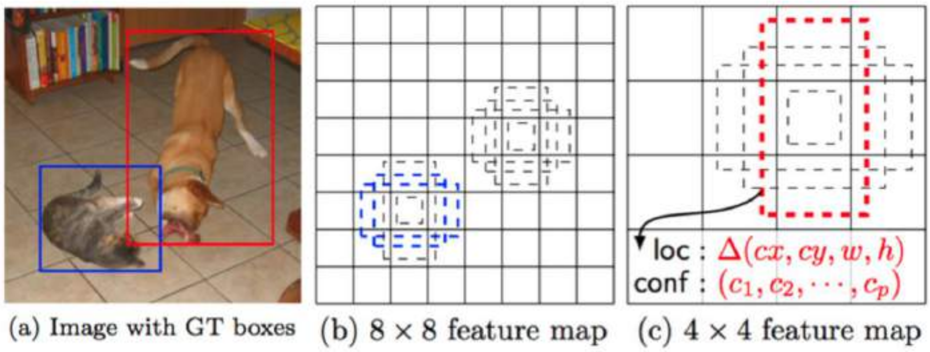

FIGURE 3.14: Matches

The loss function used is the multi-box classification and regression loss. The classification loss used is the softmax cross entropy and, for regression the smooth L1 loss is used. During prediction, non-maxima suppression[17] is used to filter multiple boxes per object that may be matched as shown in Figure 3.15.

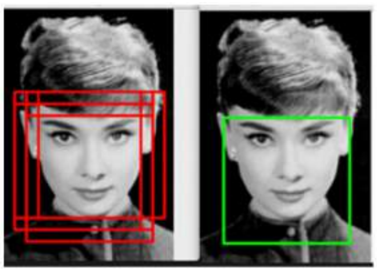

FIGURE 3.15: Non-maxima suppression

3.5 Summary

In this chapter we discussed the theories of neural network and some background study on deep learning including convolution layer, pooling layer, Flatten layer and fully connected layer. In this chapter we also discussed our study related to image classification and various modern methods and approaches for object detection in an image including overall study of SSD model. The method and tool that we have used in this project is discussed in the next chapter.

Chapter 4

Methodology

This chapter describes the methodology that we have used for the classification and detection of OA affected region in knee MRI. It will discuss the design and processing decisions that were made for the development of this project, while also giving critical insight into how these applied technologies work. The sections for setup and data set describe the working environment in this project. Preprocessing, architecture, training and postprocessing works for the development of this project are addressed in order in this chapter.

4.1 Classification Methodology

In this project we are solving an image classification problem using deep learning[18] [19], where our goal is to tell which class the input image belongs to.

The key thing to understand while image classification in this project is that the model we are building is trained on two classes of normal knee MRI and OA affected knee MRI. The way we are going to achieve this classification by training an artificial neural network on image datasets and make the neural network learn to predict which class the input image belongs to, next time it sees an image the model will be able to predict if the input contains having a normal knee MRI or a OA affected knee MRI. A simple work flow of classification model is shown in Figure 4.1 shows the prediction of input MRI image after training the neural network on the image datasets of OA affected knee MRI and normal knee MRI.

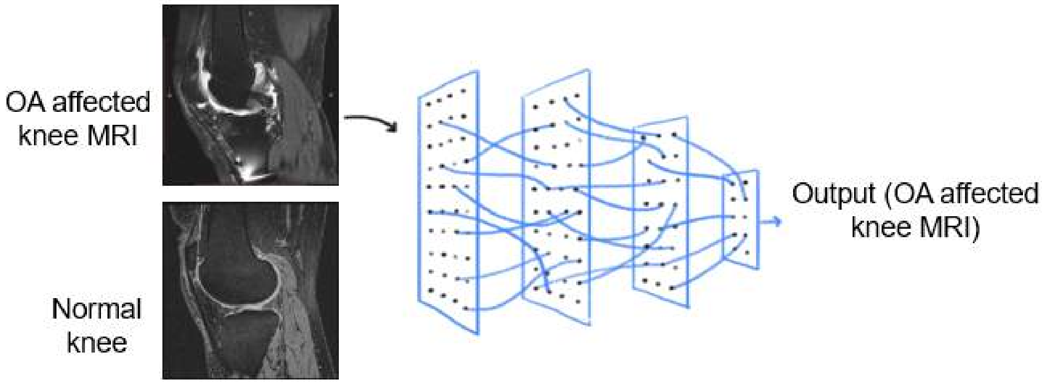

FIGURE 4.1: OA knee or normal knee prediction

4.2 Object Detection Methodology

In this project we are solving an object detection problem using deep learning, where our goal is to localize the actual region of OA in knee MRI images. The key thing to understand while detection of OA affected region are that the model we are building is trained on dataset of OA affected knee MRI. The way we are going to achieve this by exploring the modern open source based solutions for object detection in medicals, in this case detection of Osteoarthrotis disease. In this project we are using TensorFlow Object Detection API, an open source framework for object detection related tasks, for training and testing an SSD (Single-Shot Multibox Detector) with Mobilenet- model. The model is tested as pre-trained and with fine-tuning with a dataset consisting of OA knee MRI images.

4.2.1 TensorFlow Object Detection API[2][1] with SSD model with Mobilenet

TensorFlow Object Detection API is "an open source framework built on top of TensorFlow" that aims to make it easy to "design, train and implement object detection models". To achieve this, TensorFlow Object Detection API provides the user with multiple pre-trained object detection models with instructions and example codes for fine-tuning and using the models for object detection tasks. TensorFlow Object Detection API can be used with different pre-trained models. In this work, a SSD model with Mobilenet (ssd_mobilenet_v1_coco) was chosen. Figure 4.2 depicts how the pretrained, trained using MSCOCO Dataset, which consists of labeled instances in images dataset, containing object types such as "person", "cat" or "dog", able to recognize the cat (98%) and the potted plant (57%) but less sure about the identity of the dog (86% dog, 75% cat).

FIGURE 4.2: The pre-trained SSD model, able to recognize the cat (98%) and the potted plant (57%) but less sure about the identity of the dog (86% dog, 75% cat).

4.3 Setup

The workstation included an Intel i7 processor, 8GB of RAM and the computer ran Ubuntu 16.04 As the primary programming language Python 2 was chosen due to its simple syntax and popularity in the deep learning field. The code was briefly tested with Python 3 as well and seemed to work. was used as the framework for training models because its high level syntax allows fast prototyping and testing. The development environment was a Jupyter Notebook, which allowed a flexible execution of code snippets instead of running the entire program for every single change. The processing of medical image data needed a library that could handle these formats. PyDicom, SikitLearn is a Python libraries written in C++. It includes many tools for image processing and is especially popular in the medical field. Other libraries were also used for smaller tasks.

4.4 Data set

The data set is a collection MRIs of Right and Left Knees. The number of available samples grew during the project. For most of the development time, it included data samples that came from multiple MRIs sources. Data we have used consisted of knee MRI images provided by Invectus Innovation Pvt. Ltd., Noida and also downloaded from NIH (National Institute of Health) web-site - https://oai.nih.gov. The seggital view of the knee MRI images has been used for this project. All images were provided in the Dicom file format, which is very common in the medical field. It features lossless compression, as well as technical meta information about each image. Data featured a sagittal perspective. For our classification problem we have used 330 samples of knee MRI and for our OA detection problem we have used 400 samples. The images shown in Figure 4.3 move through the slices of one 2D MRI.

Source	Prospective	View	Samples	Maps	Resolution
MRIs	No	Segittal	330	0	384

Table 4.1: Details of the data set separated by source

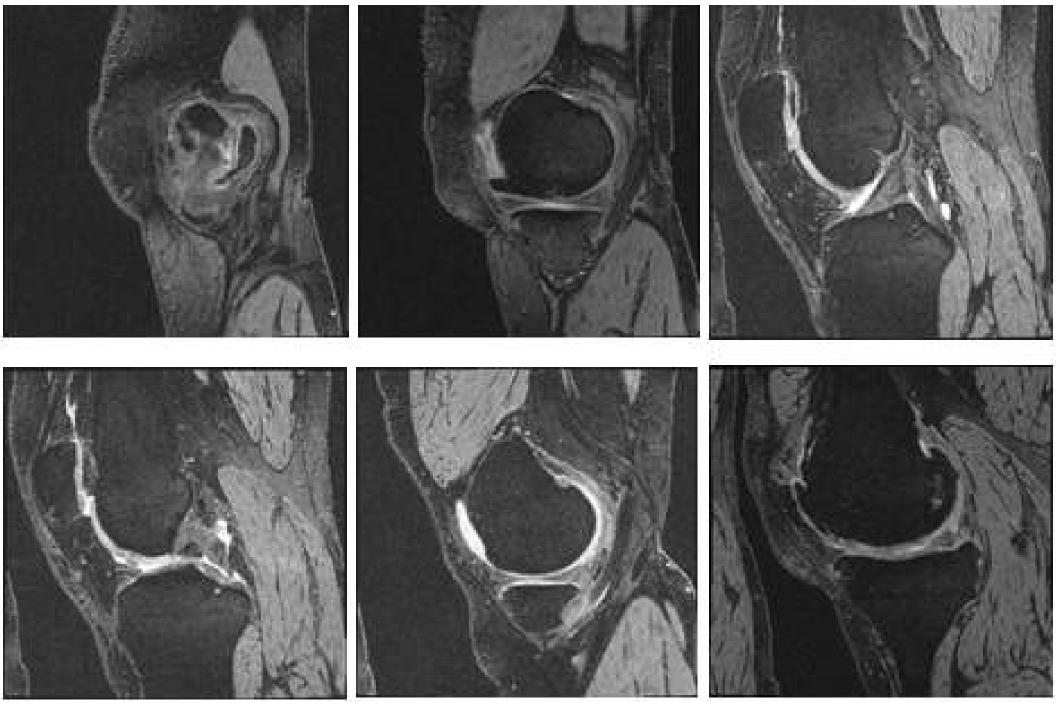

Figure 4.3: Six different slices of one 2D MRI with seggital view

The Femur already becomes visible after few frame, whereas the Tibia starts to show in the second frame. The manual labeling process took 6 hours per 2D image, and each map showed Femur, Tibia, and Patela separately. Each 2D MRI have 160 slices. In 160 slices starting 22-25 and ending 10-17 slices are not useful because of they are not showing any bone.

4.5 Preprocessing

The parameters in a Neural Network commonly range from tens of thousands to hundreds of millions. This complexity allows the model to learn on its own what features of an image are relevant for any given task. It works in conjunction with the fact that high volumes of data are available for the training. Because of the small data set that was available for this study, several types of preprocessing were applied to the images. For the most part, these techniques remove irrelevant information and reduce variance between multiple samples. Other preparation methods experimented.

4.5.1 Training, Validation, and Testing Sets

The data was split into three subsets of which each was applied for learning process of the model (training, validation, and testing)[1]. The training set is commonly the largest of the three and contains the data that is applied to the actual learning process. It's the only portion of the data the network will try to recognize the pattern and learn the target image. The validation data is used to regularly measure the performance of the model and check whether overfitting occurs or not. If the accuracy of the validation set drops below the results of the training data, the network is starting to memorize the data it knows rather than generalizing on the concept. The third subset is referred to as the testing data. In contrast to the validation set, it's only used once in the very end, to give a final score. The idea is that by building a model based on the validation results a certain amount of information bleed occurs, where the network will implicitly learn from the validation data. In order to prevent biased results, the testing data is used as a last performance reference.

- Training Set: 70% of the data
- Validation Set: 20% of the data
- Test Set: 10% of the data

In this project, for detection of OA disease, the training data base is generated by manually tagging the OA affected regions in the knee MRI images using LabelImg tool (Figure 4.4). The regions within the line markings were tagged as OA disease affected regions but any other regions outside the line markings were not tacked.

LabelImg saves the annotations as xml-files in PASCAL VOC format, so a ready-made script was available for creating TFRecords (Tensor Flow record format). The size of the MRI images (and the corresponding annotation files) were later resized to 384x384 to increase model training efficiency.

FIGURE 4.4: Typical training data image being labeled using labelImg tool.

4.5.2 Augmentation

Image augmentation is a popular approach to virtually increase the size of the data set. The general idea is that a neural network will overfit more when learning one sample n times, in comparison to learning n alternations of this sample just once. It helps to generalize on new images instead of memorizing patterns in the training data. Lossless augmentation techniques are those that don't change the values and relative localities in a sample. In 2D these include vertical and horizontal flips, as well diagonal flips if width and height are equal. Note that any multiples of 90-degree rotations can also be created using a combination of these flips. Although the samples are changed as a whole, they do not vary concerning their contained values. The term lossless only refers to the technical change and not necessarily to the semantic change. Vertically flipping a slice of this data set switches the absolute positions of Femur and Tibia. This might make it harder for the network to distinguish between the two. Lossy augmentation methods include a variety of image transformations. Common choices include:

- Horizontal shifts
- Vertical shifts
- Rotations
- Shear mapping
- Brightness adjustments
- Contrast adjustments
- Gamma adjustments

For this study horizontal flips were implemented to give the impression that images from both knees were available. In addition to this, these images were shifted 24 pixels on the horizontal axis to add another type of augmentation. Interestingly, these methods did not improve the accuracy of the model. The horizontal flips even hurt the performance and were therefore removed. The horizontal shifts were kept in the training set, so the network would perform better on unseen data that was not perfectly aligned.

4.6 Classification Work Flow

Figure 4.5 shows the work flow of the model for classification of OA affected knee MRI and normal knee MRI[18]. First we have prepared the data sets by selecting the MRIs of OA affected knee and MRIs of normal knee. Then the prepared data sets are labeled into two parts, training set and test set. The labeled data sets are then re-scaled to $n \times n$ pixel and read as grayscale. The data sets are splitted as 1442 images for training purpose and 272 images for testing and then data sets are reshaped appropriately for Tensor Flow to build the model.

After building the model, the loss function is calculated (generally we take the cross entropy loss function). Adam is a replacement optimization algorithm for stochastic gradient descent for training deep learning models which as optimizer sets learning rate to 0.001. The model is then trained for n epochs and finally the prediction of the input image is achieved as a OA affected knee MRI or a normal knee MRI.

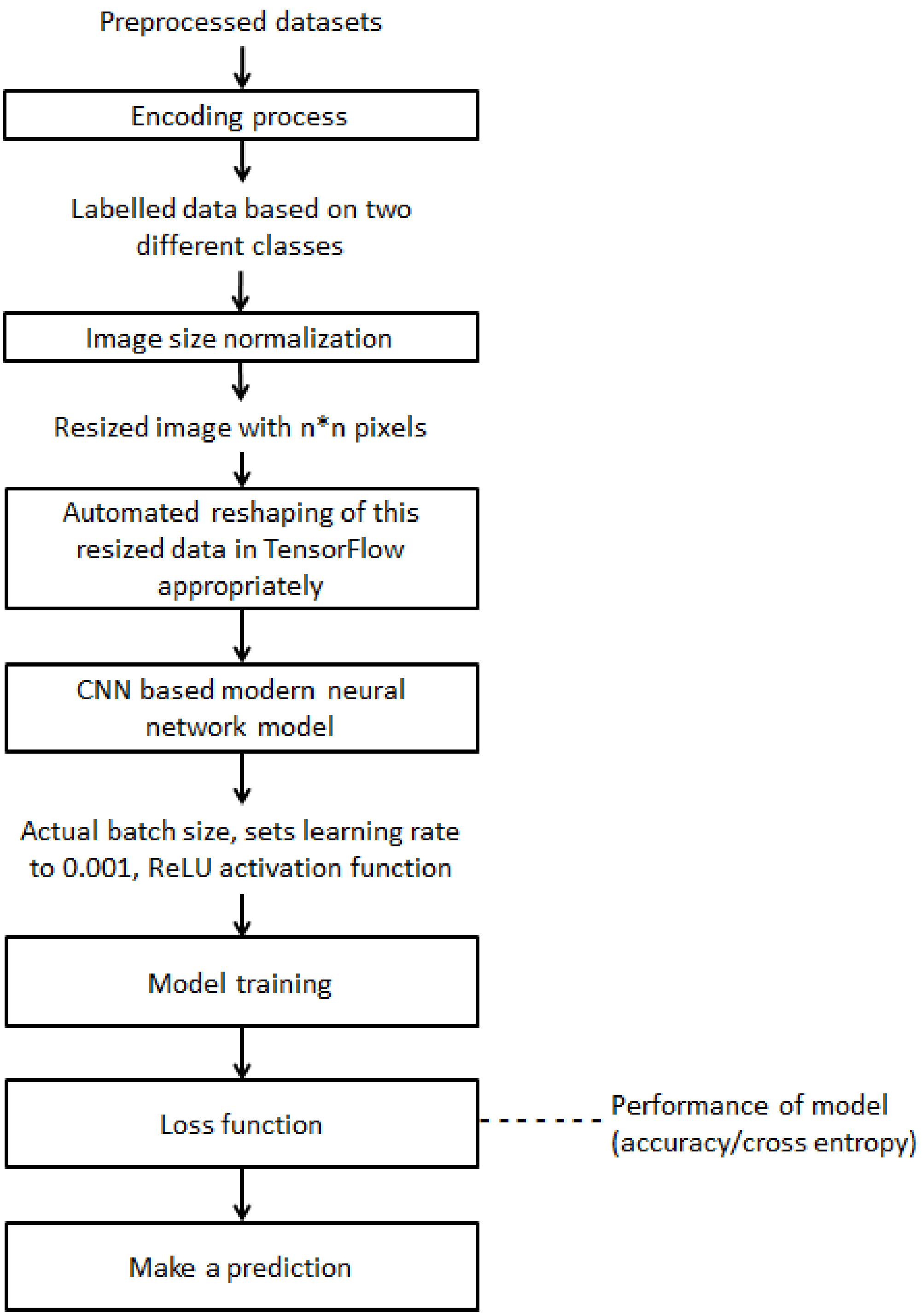

FIGURE 4.5: Flow Chart of Classification Model

4.6.1 Labelling the image datasets

The MRI datasets are labelled based on two different classes. One class contains the normal knee MRI images and the other class contains the OA affected knee MRI images.

4.6.2 Image Transformation

All the MRI images of the datasets are in different pixel values. So, the image datasets with different pixel values are resized to $n \times n$ pixel values and made all images with similar pixel values.

4.6.3 TensorFlow based automated reshaping of data

After the image transformation step, the data sets are appropriately reshaped for TensorFLow.

4.6.4 CNN based modern neural network model

A simple deep learning based classification model is shown in Figure 4.6 In neu-

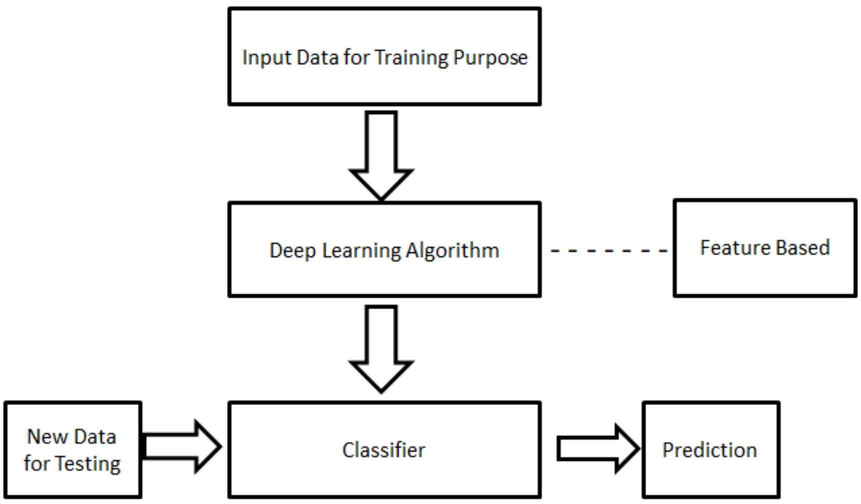

FIGURE 4.6: A Classification Model

ral networks, Convolutional neural network (ConvNets or CNNs) is one of the main categories to do images recognition, images classifications, Objects detections, recognition faces etc., are some of the areas where CNNs are widely used. In this project, the classification model is also based on CNN for image classification.

The key thing to understand while image classification in this project is that the model we are building is trained on two classes of normal knee MRI and OA affected knee MRI. The way we are going to achieve this classification by training an artificial neural network on image datasets and make the neural network learn to predict which class the input image belongs to, next time it sees an image the model will be able to predict if the input contains having a normal knee MRI or a OA affected knee MRI.

4.6.4.1 Activation Functions

Activation function decides, whether a neuron should be activated or not by calculating weighted sum and further adding bias with it. The purpose of the activation function is to introduce non-linearity into the output of a neuron. We know, neural network has neurons that work in correspondence of weight, bias and their respective activation function. In a neural network, we would update the weights and biases of the neurons on the basis of the error at the output. This process is known as back-propagation. Activation functions make the back-propagation possible since the gradients are supplied along with the error to update the weights and biases.

Activation functions sit between layers in a network to introduce a non-linearity. Otherwise, the operations could be to a simple linear transformation and remove the benefits of building a deep model. The rectified linear unit (ReLU) is the default recommendation for an activation function in modern neural networks. It's defined as the maximum of 0 and the input value and can be described as a nonlinear function made up of two linear pieces. Because of this, it preserves many of the properties that make models easy to optimize and generalize well on new data.

4.6.4.2 Batch Size

The number of random samples per training step is referred to as the batch size. In the past, it was believed that larger batches led to something called the generalization gap, where the accuracy of a model would drop if it was trained on unusually large batches. Recent work suggests other reasons for this decrease in accuracy. While common batch sizes range from 32 to 256. Using batches smaller than 32 samples can introduce a different kind of problem. Having too few data points that don't represent the mean of the data well, may lead to slow and unstable training. Based on hardware limitations the largest possible batch sizes ranged from 24 to 48 samples depending on the current architecture. Whatever could be fit into memory was used for these experiments. Exceptions occurred when working with 2D convolutions in which case the batch size had to be limited to just 4 samples.

4.6.4.3 Learning Rate Policy

One full iteration over the training samples is referred to as an epoch, and the learning rate policy describes how the learning rate is changed from one epoch to another. With the introduction of adaptive optimizers like Adam, there has been a lower emphasize on this topic because the learning rate is modified during training. Even though this reduces the number of possible defects, training time can often be saved with the right initial learning rate. Ten epochs were run at different learning rates to compare initial results and to examine the point at which the model wouldn't converge at all. 0.002 was the highest rate at which the model started training, but 0.001 resulted in the best score. Adding a decay that reduces the learning rate manually over time did not improve the results with the use of Adam.

4.6.4.4 Metrics and Loss Functions

Metrics are used in deep learning to measure the performance of a model. For example, the accuracy is often chosen to describe how well a neural network is doing on a classification task. An accuracy of 0.9 indicates that 9 out of 10 samples are classified correctly.

$$Accuracy = \frac{TP + TN}{(TP + TN + FP + FN)}$$

In the formula above T and F indicate whether a prediction was true or false. P and N stand for a positive or negative outcome.The result of a loss function is a metric that will be minimized during the backward propagation process. It needs to be a differentiable function, which is why the accuracy cannot be used as a loss function. It is a binary metric that works with true or false values and not with probabilities. In situations like these, a surrogate function is used that has a high correlation with the target metric. For classification problems, this is often the cross entropy. Because a segmentation can be seen as a classification for every output pixel, it was also chosen as a candidate for this study.

$$CrossEntropy(p, q) = -\sum p(x) log q(x)$$

4.6.4.5 Optimizer

The previous section provided an overview of the loss function, which measures the performance of a prediction during the training process. This section discusses how the result can be used to execute the actual optimization step. The derivative of a single variable function defines the slope of this function at any given point. Knowing this, one can tell in which direction the original function declines. Adjusting the independent variable along the descent of a loss function will minimize the error in respect to its dependent variable. The gradient is the derivative for functions of multi-dimensional inputs, such as the loss functions used in deep learning.

A process that minimizes its result is called gradient descent. While it is possible to determine its minimum analytically, it is intractable for artificial neural networks due to the high number of parameters. Instead, the stochastic gradient descent (SGD) is used which will take a random batch of the training data and iteratively adjust the parameters in small steps. SGD is the basis for all common ANNs, but over the years different variants were introduced to the field. The Adam optimizer is such a variant, which enhances SGD amongst other things by using what's called an adaptive momentum estimation. This is also where its name derives from. It adaptively adjusts the learning rate which defines how much the parameters will be changed in one training step. By incorporating the previous and current shape of the slope, Adam can tremendously speed up the training.

4.7 Osteoarthritis Detection Work Flow

Image classification can perform some pretty amazing feats, but a large drawback of many image classification applications is that the model can only detect one class per image. With an object detection model, not only we can classify multiple classes in one image, but we can specify exactly where that object is in an image with a bounding box framing the object. The TensorFlow Models GitHub repository has a large variety of pre-trained models for various machine learning tasks, and one excellent resource is their object detection API[16][8]. The object detection API makes it extremely easy to train your own object detection model for a large variety of different applications. Whether you need a high-speed model to work on live stream high-frames-per-second (fps) applications or high-accuracy desktop models, the API makes it easy to train and export a model.

Figure 4.7 shows the work flow of the model for detection of OA affected region in knee MRI. First we have prepared the data sets by selecting the MRIs of OA affected knee. Then the prepared data sets are labeled into two parts, training set and test set. The labeled data sets are then re-scaled to n x n pixel and read as grayscale. The data sets are splitted as 303 images for training purpose and 46 images for testing along with respective .xml files and then data sets are reshaped appropriately for TensorFlow object detection API to build the model. The steps for building a custom object detection model using TensorFlow API is discussed below.

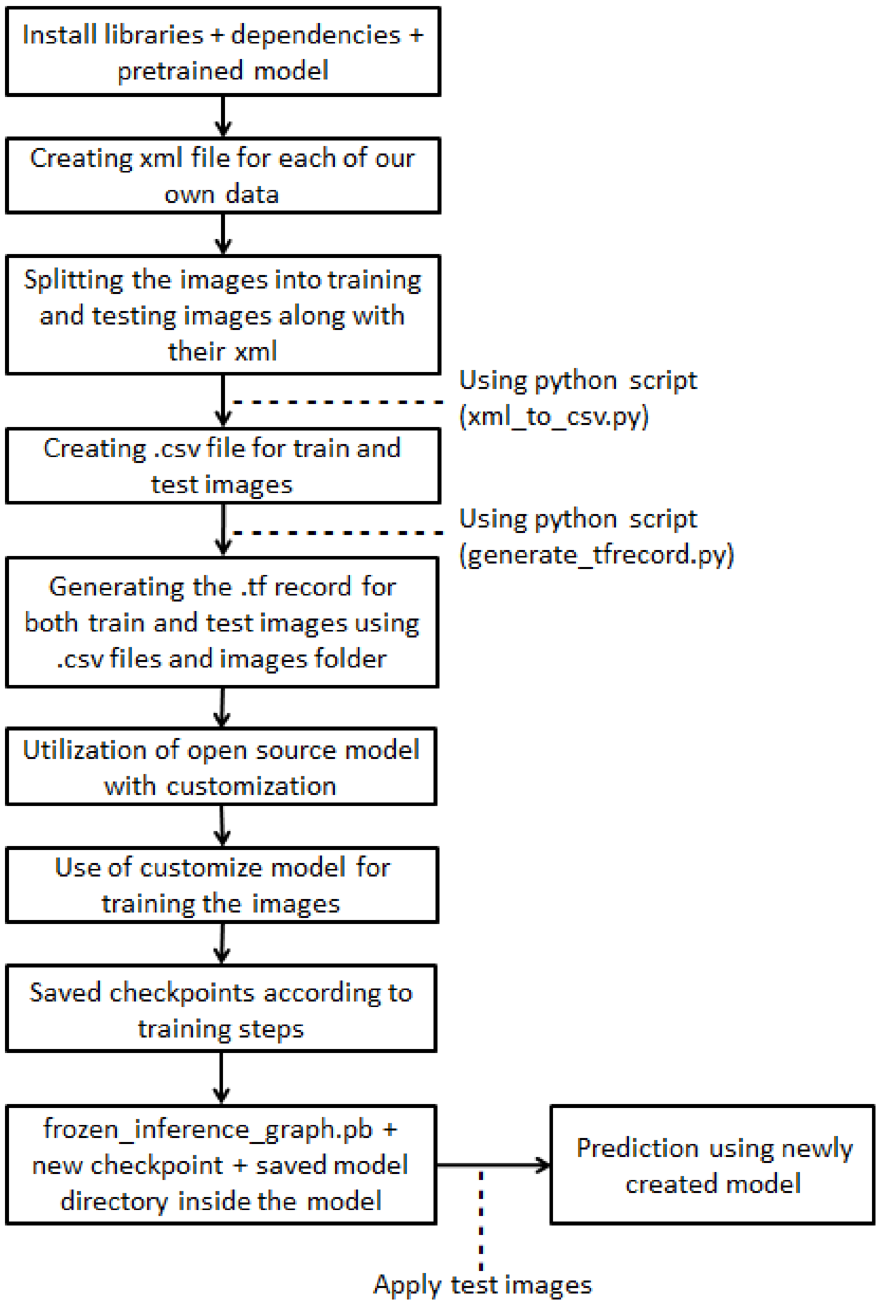

FIGURE 4.7: Flow Chart of Object Detection Model

4.7.1 Data Set Collection

Some very large detection data sets, such as Pascal and COCO, exist already, but to train a custom object detection class, it is required to create and label own data set.

For own data set, images of OA affected knee MRI is collected from the website of National Institute of Health (https://oai.nih.gov) and also data provided by Invectus Innovation Pvt. Ltd., Noida. Only the OA affected knee MRI images are used for this project. Ideally, got at least 300-400 OA knee MRI images for this project. Then the data set splitted as 303 images for training; for the two possibilities of OA affected regions namely, effusion and cartilage erosion (as per the medical expert of this area). Due to the limited amount of data, test files are splitted to 10%; of the total MRI images. For convenience, it is decided to resize all the images to 384×384 pixels before saving them so to create the required bounding boxes and not worry about having to resize the images down the line.

4.7.2 Creating Bounding Boxes

In order to train the object detection model, for each image the image's width, height, and each class with their respective xmin, xmax, ymin and ymax bounding box is needed. For this project, the total number of MRI images labeled are 345 which include both training and test MRI images. Some of the MRIs from above mentioned labeled dataset with corresponding xml files showing details of their respective xmin, ymin, xmax and ymax bounding box values are shown by Table 4.2. Simply, bounding box is the frame that captures exactly where the possibilities of OA affected area is in the MRI image. Figure 4.8 is a labeled MRI image sample showing bounding boxes. These labels are created using LabelImg tool, an excellent open source free software that makes the labeling process much easier. It will save individual xml labels for each image, which we will convert into a csv table for training.

FIGURE 4.8: Sample showing bounding boxes

MRIs	xml files	xmin	ymin	xmax	ymax
IMG-0000-0001001.jpg	IMG-0000-0001001.xml	117	137	138	181
IMG-0000-0001002.jpg	IMG-0000-0001002.xml	121	135	136	175
IMG-0000-0001003.jpg	IMG-0000-0001003.xml	124	138	138	177
IMG-0026-00032.jpg	IMG-0026-00032.xml	98	133	117	179
IMG-0026-00033.jpg	IMG-0026-00033.xml	93	128	115	179
IMG-0026-00034.jpg	IMG-0026-00034.xml	94	122	112	186
trcartilage001.jpg	trcartilage001.xml	110	212	130	239
trcartilage002.jpg	trcartilage002.xml	105	199	131	237
trcartilage003.jpg	trcartilage003.xml	103	159	124	233
tscartilage001.jpg	tscartilage001.xml	119	185	157	219
tscartilage002.jpg	tscartilage002.xml	113	172	149	219
tscartilage003.jpg	tscartilage003.xml	112	176	147	215

Table 4.2: MRIs data with corresponding xml files showing details of their respective xmin, ymin, xmax and ymax bounding box values

4.7.3 Convert Labels to the TFRecord Format

After converting xml files into csv files then it is needed to generate a TFRecord file to optimize the data feed.

4.7.4 Model Selection

There are models in the TensorFlow API which can be used depending on the needs. If a high-speed model that can work on detecting image feed at high fps is needed, the single shot detection (SSD) network works best. Some other object detection networks detect objects by sliding different sized boxes across the image and running the classifier many times on different sections of the image; this can be very resource consuming. As its name suggests, the SSD network determines all bounding box probabilities in one go; hence, it is a vastly faster model. However, with single shot detection, speed is gained at the cost of accuracy. For this project, single shot detection as the bounding box framework is used, but for the neural network architecture, the MobileNet model is used, which is designed to be used in mobile applications. The config file for SSD MobileNet has already configured and depending on the computer, the batch size in the config file is lowered to stop running out of memory.

4.7.5 Retrain the Model with Collected Data

Now trained the entire SSD MobileNet model has been trained on collected data from scratch which took a lot of training time. The easier solution is to take a model already trained on a large data set and clip off the last layer, which has the classes from the trained model, and replace it with required classes. By doing this, all the feature detectors trained in the previous model will be used and these features will be used to try to detect the new classes. Since only the last layer of this mobilenet model is retrained, a high-end GPU is not required (but it can certainly speed things up). Once the loss is consistently around the value of 1 or starts rising, TensorFlow training can be stopped by pressing ctrl+c. To train, simply run the ‘train.py‘ file in the object detection API directory.

4.7.6 Implementation of New Model With TensorFlow

Before starting an experiment with the newly trained model, exported the graphs for inference from TensorFlow. The latest ckpt from the data directory is used. Various graphs showing the development of classification loss, localization loss and total loss obtained while training the model is depicted in Figure 4.9 and Figure 4.10
After this, validated the model's performance, in which those images are used which the model has never seen before; for validation images, some MRI images of different patient's knee MRI are used. About 10% of the total images are assigned to be validation images. It is to be noted that with a good number of validation images, multiple checkpoints can be tested to see which one performs best.

It is very clear to see that the very limited amount of data, about 100 images per class (for this case, effusion and cartilage erosion), was not enough to get a robust model. Though the detector was able to detect all the OA affected regions, but it was not able to detect the regions, may be due to absence of OA in knee MRI or due to limited amount of data for this project. But, We believe this OA detection problem showcases the API's capabilities well. It was the goal to gather all the steps to create such an object detection model which will be capable of detecting OA affected regions in knee MRI images.

FIGURE 4.9: Graph showing development of classification loss and localization loss during training

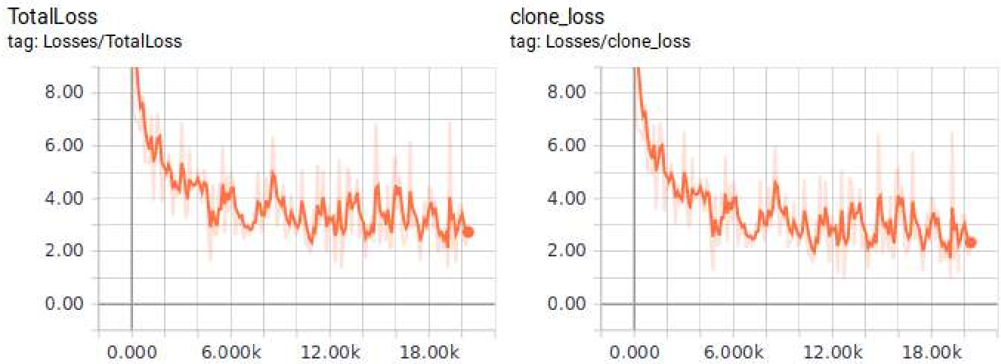

FIGURE 4.10: Graph showing development of total loss during training our model

4.7.7 Object Detection with Customized Model

The pre-trained SSD model (ssd_mobilenet_v1_coco) was fine-tuned for the dataset using manually labeled images of OA knee MRI. A provided configuration file (ssd_mobilenet_v1_coco.config) was used as a basis for the model configuration (after testing different configuration settings, the default values for configuration parameters were used). The provided checkpoint file for ssd_mobilenet_v1_coco was used as a starting point for the fine tuning process. The training was stopped after 12550 time steps when the mean average precision (mAP) some what leveled out. The mAP values kept fluctuating even after increased value of mAP, which raised suspicion that even longer training might improve the detection results. Training the model with CPU took a lot of time. The total loss value was reduced rapidly for this model due to starting from the pre-trained checkpoint file.

The fine-tuned model is fed to TensorFlow Object Detection API and tested on test data from OA knee MRI images, and in addition on test data taken from different patients OA knee MRI to see how model's specific factors affect the result. The model was then tested to see if the model specific fine-tuning improved the detection results.

4.7.8 Evaluation Metrics[1]

The most relevant evaluation metrics for this application are precision (correctness), recall, sensitivity and mAP.

Precision is the ratio of the correctly positive labeled by our program to all positive labeled and describes how relevant the detection results are.

Recall is the ratio of the correctly labeled by our program to all who are diabetic in reality and describes the percentage of relevant objects that are detected with the detector.

Generally, when precision increases, recall decreases and vice versa: a very sensitive model is able to catch large percentage of objects in an image, but it also generates high number of false positives, where as a model with high threshold for detection only produces few false positives but it also leaves a higher percentage of objects undetected. The right balance between these two depends on the application.

Sensitivity means percentage of diseased pixels properly excluded from the segmentation results out of all the pixels from the segmentation results outside of the ground truth disease [20][21].

Mean Average Precision (mAP) is a commonly used metrics for comparing model performance in object detection[1]. mAP summarizes the information in "precision-recall" curve to one number. Precision-recall curve is formed by sorting all the predicted bounding boxes (over all images in the evaluation set) assigned to a object class by their confidence rating, and then for each prediction, a recall value and a precision value are calculated. Recall is defined as proportion of TPs above the given rank out of all user tagged objects. Precision is defined as proportion of TPs in predictions above the given rank. From these precision-recall-pairs a precision-recall curve is formed. Average Precision (AP) is calculated as the "area under the precision-recall-curve", i.e. it approximates precision averaged across all values of recall between 0 and 1 by summing precision values multiplied by the corresponding change in recall from one point in curve to the next. However, what VOC metrics call "AP" is actually interpolated average precision, which is defined "as the mean precision at a set of equally spaced recall levels", the precision value corresponding to each recall interval being the maximum precision value observed in the curve within the interval. mAP of a model is then the mean of AP values of all object classes. The interpolated AP is, of course, higher than the basic AP; this must be taken into consideration when comparing mAP values.

4.8 Summary

In this chapter we described the methodologies that we have used for the classification of normal knee MRI and OA affected knee MRI as well as for detection of knee OA (Osteoarthritis) disease. We discussed the design and processing decisions that were made for the development of this project. The sections for setup and data set is also described for the working environment for this study. The development of image classification and object detection models and the work flow of all preprocessing, architecture, training and postprocessing was also discussed in chronological order. The result and discussion of our classification problem is discussed in the next chapter.

Chapter 5

Result and Discussion

The results in this chapter are based on data the network has been trained with. For classification, results are obtained from different OA patient's knee MRIs and normal knee MRIs. The methodology is applied on 1714 MRIs from which 1442 MRIs are used for training and 272 MRIs are used for testing. Whereas for detection of OA(Osteoarthritis) disease, results are obtained from different OA patient's knee MRIs and the methodology for detection is applied on

5.1 Classification Result Analysis

Table 5.1 shows the performance of the classification model.

S.No.	Epoch	Val. Accuracy	Val. Loss	Training Accuracy	Training Loss
1	1	0.9561	0.1391	0.9890	0.0335

Table 5.1: Performance of Model

Figure 5.1 shows the input MRI selected for prediction to our classification model. After the successful classification Figure 5.2 shows the the result obtained as OA knee.

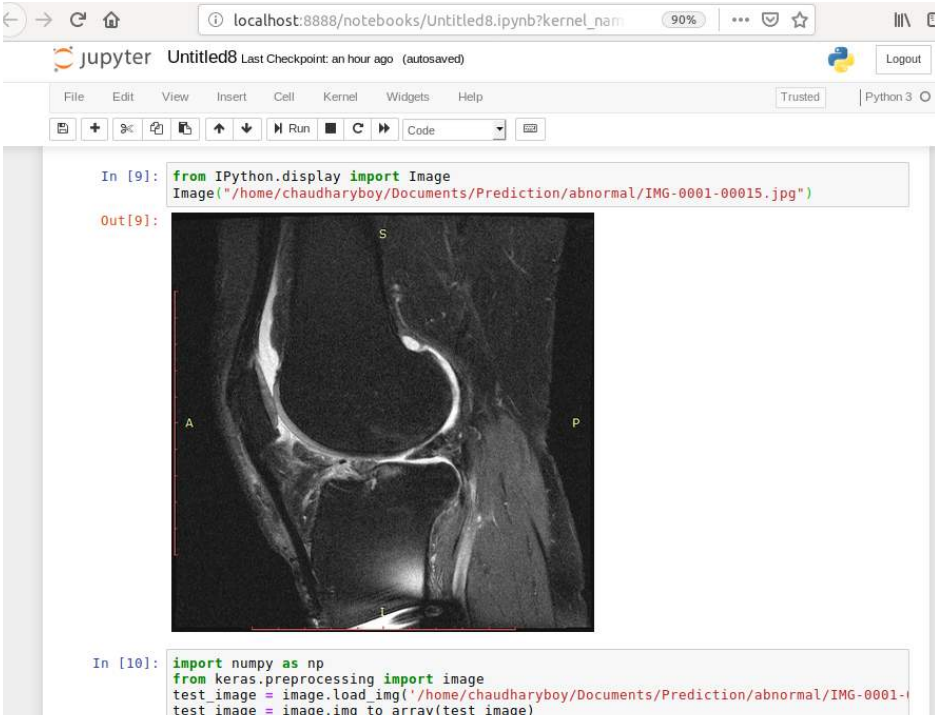

FIGURE 5.1: Input Knee MRI

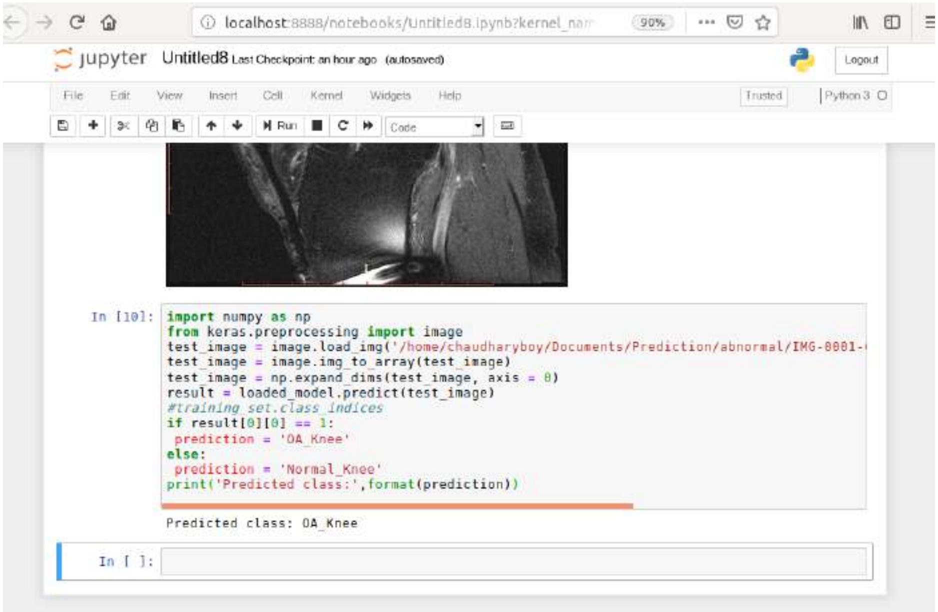

FIGURE 5.2: Output Predicted as OA Knee MRI

5.2 OA Detection Result Analysis

The detected results are based on data the network has been trained with. For this detection of OA disease, results are obtained from different OA patient's knee MRIs and the methodology for detection is applied on 349 MRIs from which 303 MRIs along with their .xml files are used for training and 46 MRIs along their .xml files are used for testing.

Figure 5.3 shows the input MRIs selected for detecting presence of OA disease

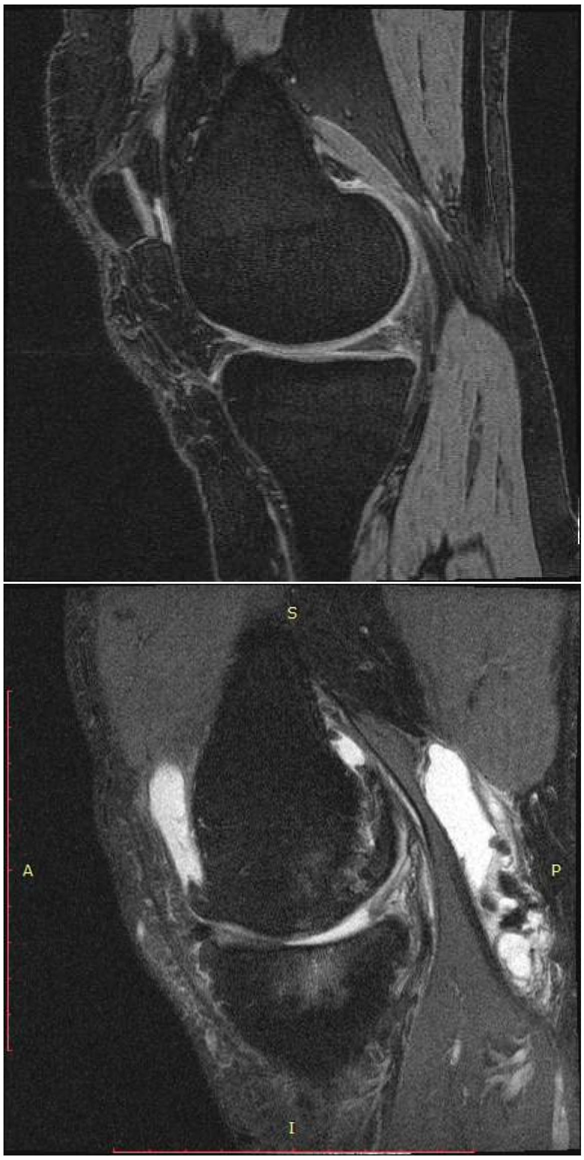

FIGURE 5.3: Input Knee MRIs selected for detecting OA

to the detection model. After the successful detection Figure 5.4 shows the result obtained as OA disease due to presence of cartilage erosion and Figure 5.5 shows the result obtained as OA disease due to presence of effusion.

Figure 5.6 shows an another input MRI selected for detecting presence of OA disease to the detection model and in this case the result obtained is shown by the Figure 5.7 which shows the result obtained as OA due to presence of both the cases of cartilage erosion and effusion.

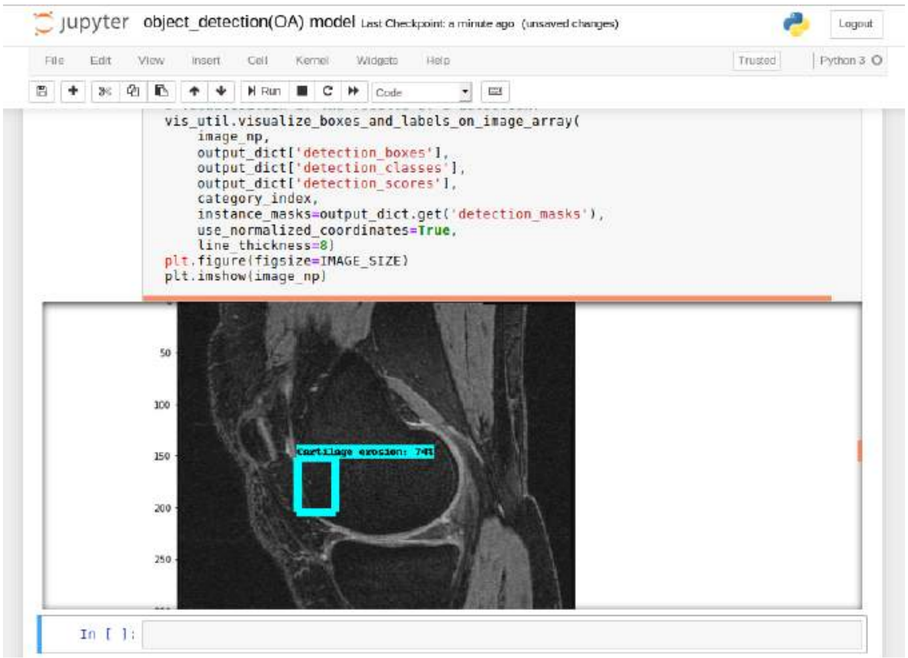

FIGURE 5.4: Output detected as OA due to cartilage errosion

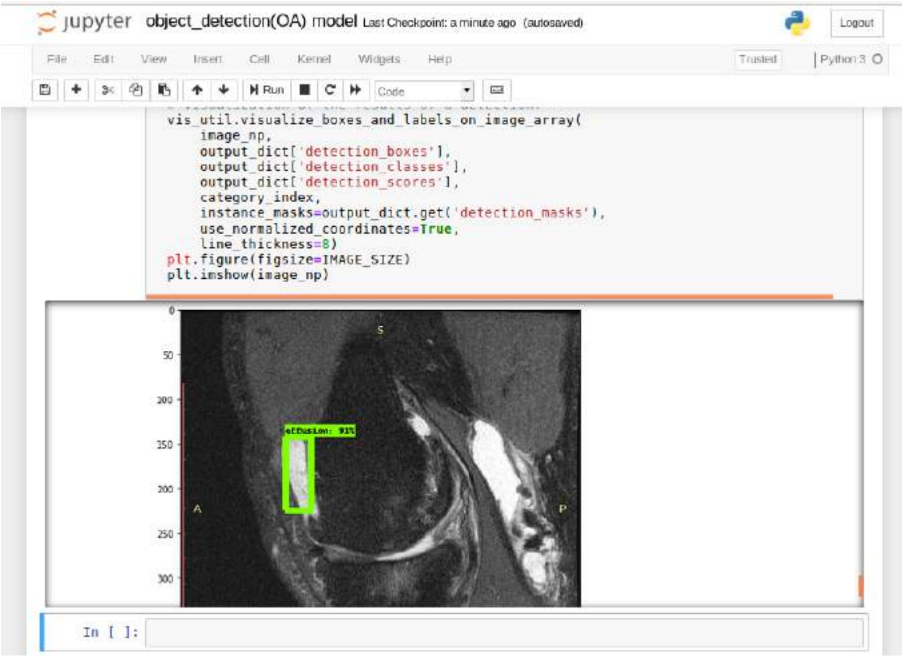

FIGURE 5.5: Output detected as OA due to effusion

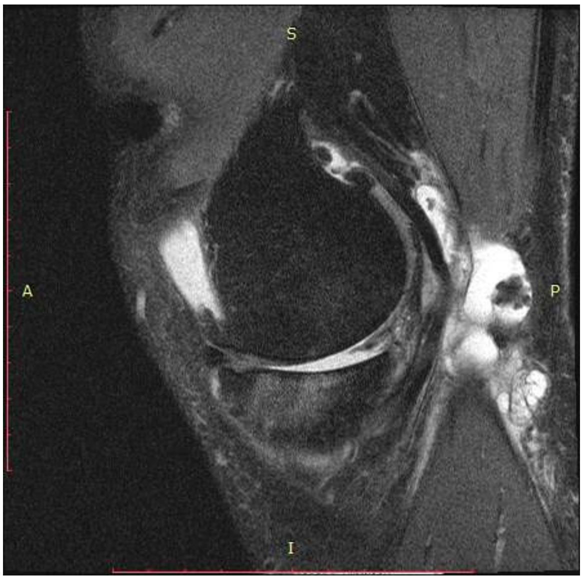

FIGURE 5.6: Input Knee MRIs selected for detecting OA

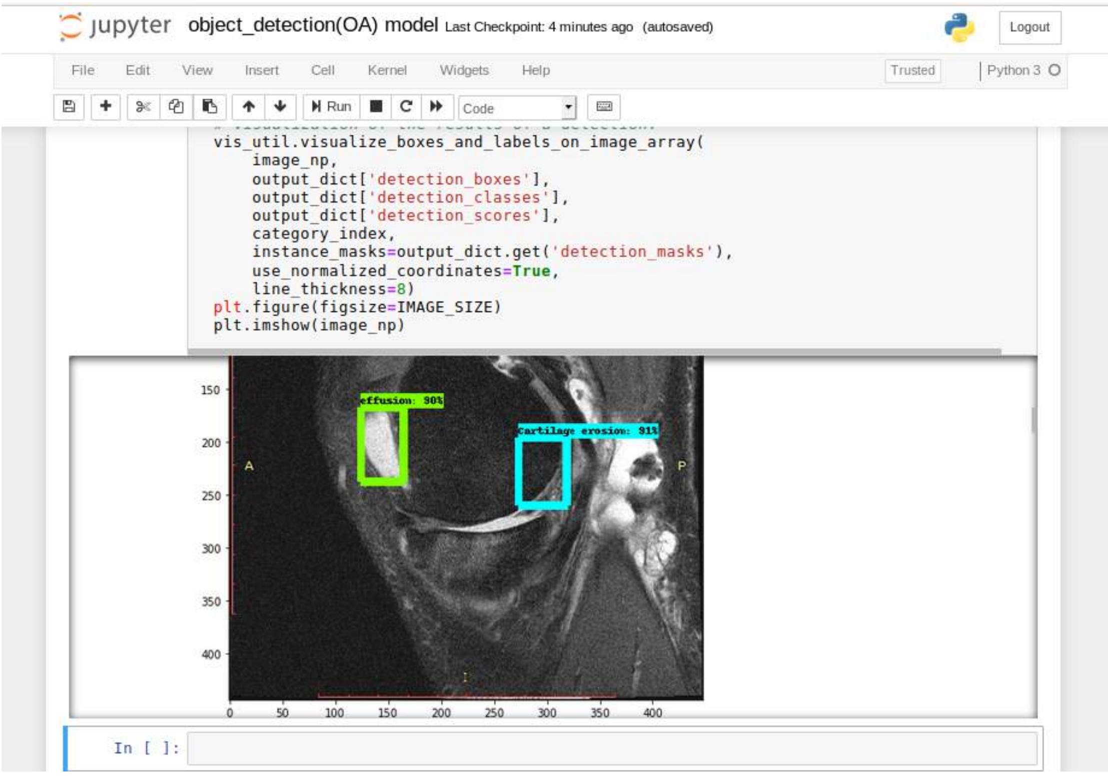

FIGURE 5.7: Output detected as OA due to presence of both cartilage erosion and effusion

Chapter 6

Conclusion and Future Scope

It was very interesting and adopting learning kind of experience to work with medical images in this project as it is playing a very big role in the medical field. This project took us through the various phases of project development and gave us the real insight into the world of machine learning techniques. The joy of working and the thrill involve while tackling the various problems and challenges gave us a feel of developers industry. It was due to this project we came to know how different machine learning techniques and methods can be designed in order to achieve a state of art solution to the problems like image classification and object detection. In this project, based on our created features from medical MRI images, later we were able to classify the normal knee MRI images and the disease affected knee MRI images. Also we were able to detect the actual location of the disease affected region in the knee MRI images.

Some different adaptations, tests, and experiments have been left for the future due to lack of data and due to lack of time as well. The experiments require a large amount dataset to achieve better accuracy results. Also the experiments with real data are usually very time consuming, requiring even days to finish a single run. Future work for this project concerns with solving this object detection and localization problem with considering some more possibilities of OA cases like edema and osteophytes. The future scope of this project would be to use a large dataset of OA knee MRIs and work on other two views such as axial and coronal. This would help this project in classifying and detecting the location of disease even in minor OA cases and with better accuracy. Other future scopes would also include the experiments to be done on 3-D images, which may help to detect the presence disease region from whole knee. Another future scope for this project would be to work on the medical images of other body joints also. After achieving so much of work on this project, the model then would be sufficient enough to predict the cases of OA as well as would also be able to detect the cause of OA presence in a patient's knee as well as other body joints.

Appendix A

LabelImg Tool

Appendix B

The Full Code

```python
import numpy as np
import os
import six.moves.urllib as urllib
import sys
import tarfile
import tensorflow as tf
import zipfile

from distutils.version import StrictVersion
from collections import defaultdict
from io import StringIO
from matplotlib import pyplot as plt
from PIL import Image

# This is needed since the notebook is stored in the object_detection folder.
sys.path.append("..")
from object_detection.utils import ops as utils_ops

# print(tf.__version__)
# if StrictVersion(tf.__version__) < StrictVersion('1.12.0'):
#   raise ImportError('Please upgrade your TensorFlow installation to v1.12.*.')

# This is needed to display the images.
%matplotlib inline

from utils import label_map_util

from utils import visualization_utils as vis_util

# What model to download.
MODEL_NAME = 'chandanKaEffusion_inference_graph'

# Path to frozen detection graph. This is the actual model that is used for the object detection.
```

```python
PATH_TO_FROZEN_GRAPH = MODEL_NAME + '/frozen_inference_graph.pb'

# List of the strings that is used to add correct label for each box.
PATH_TO_LABELS = os.path.join('training', 'object-detection.pbtxt')

NUM_CLASSES = 2

detection_graph = tf.Graph()
with detection_graph.as_default():
  od_graph_def = tf.GraphDef()
  with tf.gfile.GFile(PATH_TO_FROZEN_GRAPH, 'rb') as fid:
    serialized_graph = fid.read()
    od_graph_def.ParseFromString(serialized_graph)
    tf.import_graph_def(od_graph_def, name='')

category_index = label_map_util.create_category_index_from_labelmap(PATH_TO_LABELS, use_display_name=True)

def load_image_into_numpy_array(image):
  (im_width, im_height) = image.size
  return np.array(image.getdata()).reshape(
      (im_height, im_width, 3)).astype(np.uint8)

# For the sake of simplicity we will use only 2 images:
# image1.jpg
# image2.jpg
# If you want to test the code with your images, just add path to the images to the TEST_IMAGE_PATHS.
PATH_TO_TEST_IMAGES_DIR = 'test_images'
TEST_IMAGE_PATHS = [ os.path.join(PATH_TO_TEST_IMAGES_DIR, 'image{}.jpg'.format(i)) for i in range(1, 11) ]

# Size, in inches, of the output images.
IMAGE_SIZE = (14, 8)
def run_inference_for_single_image(image, graph):
  with graph.as_default():
    with tf.Session() as sess:
      # Get handles to input and output tensors
      ops = tf.get_default_graph().get_operations()
      all_tensor_names = {output.name for op in ops for output in op.outputs}
      tensor_dict = {}
      for key in [
          'num_detections', 'detection_boxes', 'detection_scores',
          'detection_classes', 'detection_masks'
      ]:
        tensor_name = key + ':0'
        if tensor_name in all_tensor_names:
          tensor_dict[key] = tf.get_default_graph().get_tensor_by_name(
              tensor_name)
      if 'detection_masks' in tensor_dict:
        # The following processing is only for single image
```

```
   detection_boxes = tf.squeeze(tensor_dict['detection_boxes'], [0])
   detection_masks = tf.squeeze(tensor_dict['detection_masks'], [0])
   # Reframe is required to translate mask from box coordinates to image coordinates and fit the image size.
   real_num_detection = tf.cast(tensor_dict['num_detections'][0], tf.int32)
   detection_boxes = tf.slice(detection_boxes, [0, 0], [real_num_detection, -1])
   detection_masks = tf.slice(detection_masks, [0, 0, 0], [real_num_detection, -1, -1])
   detection_masks_reframed = utils_ops.reframe_box_masks_to_image_masks(
       detection_masks, detection_boxes, image.shape[1], image.shape[2])
   detection_masks_reframed = tf.cast(
       tf.greater(detection_masks_reframed, 0.5), tf.uint8)
   # Follow the convention by adding back the batch dimension
   tensor_dict['detection_masks'] = tf.expand_dims(
       detection_masks_reframed, 0)
  image_tensor = tf.get_default_graph().get_tensor_by_name('image_tensor:0')

  # Run inference
  output_dict = sess.run(tensor_dict,
                         feed_dict={image_tensor: image})

  # all outputs are float32 numpy arrays, so convert types as appropriate
  output_dict['num_detections'] = int(output_dict['num_detections'][0])
  output_dict['detection_classes'] = output_dict[
      'detection_classes'][0].astype(np.uint8)
  output_dict['detection_boxes'] = output_dict['detection_boxes'][0]
  output_dict['detection_scores'] = output_dict['detection_scores'][0]
  if 'detection_masks' in output_dict:
   output_dict['detection_masks'] = output_dict['detection_masks'][0]
return output_dict

for image_path in TEST_IMAGE_PATHS:
image = Image.open(image_path)
# the array based representation of the image will be used later in order to prepare the
# result image with boxes and labels on it.
image_np = load_image_into_numpy_array(image)
# Expand dimensions since the model expects images to have shape: [1, None, None, 3]
image_np_expanded = np.expand_dims(image_np, axis=0)
# Actual detection.
```

```
    output_dict = run_inference_for_single_image(image_np_expanded,
      detection_graph)
   # Visualization of the results of a detection.
    vis_util.visualize_boxes_and_labels_on_image_array(
        image_np,
        output_dict['detection_boxes'],
        output_dict['detection_classes'],
        output_dict['detection_scores'],
        category_index,
        instance_masks=output_dict.get('detection_masks'),
        use_normalized_coordinates=True,
        line_thickness=8)
    plt.figure(figsize=IMAGE_SIZE)
    plt.imshow(image_np)
```

References

[1] Pirkko Mustamo. Object detection in sports: Tensorflow object detection api case study. *Bachelor's Thesis Degree Programme in Mathematical Sciences*, January 2018.

[2] Daniel Stang. Step by step tensorflow object detection api tutorial-selecting a model. October 2007.

[3] Shaoqing Ren Kaiming He, Xiangyu Zhang and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. Feb 2015.

[4] Zhifeng Chen Quoc V. Le Mohammad Norouzi Wolfgang Macherey Maxim Krikun Yuan Cao Qin Gao Klaus Macherey Je Klingner Apurva Shah Melvin Johnson Xiaobing Liu Łukasz Kaiser Stephan Gouws Yoshikiyo Kato Taku Kudo Hideto Kazawa Keith Stevens George Kurian Nishant Patil Wei Wang Cli Young Jason Smith Jason Riesa Alex Rudnick Oriol Vinyals Greg Corrado Macdu Hughes Yonghui Wu, Mike Schuster and Jerey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. Sep 2016.

[5] Anna Choromanska Krzysztof Choro-manski Bernhard Firner Lawrence Jackel Mariusz Bojarski, Philip Yeres and Urs Muller. Explaining how a deep neural network trained with end-to-end learning steers a car. Apr 2017.

[6] Joshua Saxe and Konstantin Berlin. Deep neural network based mal-ware detection using two dimensional binary program features. Aug 2015.

[7] Justin Francis. How to create your own custom object detection model. *O'reilly link- https://www.oreilly.com/*, October 2017.

[8] Kurt. Object detection tutorial in tensorflow: Real-time object detection. *edureka*, 7, April 2019.

[9] Neeraj Sharma Lalit M. Aggarwal. Automated medical image segmentation techniques. *National Center for Biotechnology Information*, Jan-Mar 2010.

[10] Catherine Westbrook. Mri at a glance. 2016.

[11] Dr KM Ravikumar Sanjeevakumar Kubakaddi. Measurement of cartilage thickness for early detection of knee osteoarthritis (koa). *IEEE Point-of-Care Healthcare Technologies (PHT) Bangalore*, Jan 2013.

[12] Min-Sun Son Marc Levenston Brian A. Hargreaves Lauren M. Shapiro, Emily J. McWalter and Garry E. Gold. Mechanisms of osteoarthritis in the knee: Mr imaging appearance. Feb 2014.

[13] Pooja U. Patravali Ramesh R. Manza, Shivanand S. Gornale. Classification of osteoarthritis using image processing techniques. *International Journal of Scientific Engineering Research*, 7, June 2016.

[14] Andrew Ng. "deep learning specialization,coursera.

[15] Pooja U. Patravali Ramesh R. Manza, Shivanand S. Gornale. A survey on exploration and classification of osteoarthritis using image processing techniques. *International Journal of Scientific Engineering Research*, 7, June 2016.

[16] Introduction and use - tensorflow object detection api tutorial, pythonprogramming.net.

[17] Pratik Kalshetti Ashish Jaiswal Naman Rastogi Prafull Gangawane. Object detection. *Department of Computer Science and Engineering, Indian Institute of Technology Bombay, India.*

[18] Venkatesh Tata. Simple image classification using convolution neural network — deep learning in python. Dec 2017.

[19] Geoffrey E. Hinton Alex Krizhevsky, Ilya Sutskever. Imagenet classification with deep convolutional neural networks.

[20] C Metz. Basic principles of of roc analysis. *Seminars in Nuclear Medicine*, 7, 1978.

[21] B. Sirmacek and C Unsalan. Road network extraction using edge detection and spatial voting. *International Conference on Pattern Recognition, Turkey, Istanbul*, August 2010.